Design Methods for Networked Control Systems with Unreliable Channels focusing on Packet Dropouts

ISABEL JURADO FLORES

Doctoral Thesis

–

Universidad de Sevilla
Departamento de Ingeniería de Sistemas y Automática
Supervisors: Dr. Francisco Rodríguez Rubio
Dr. Manuel Gil Ortega Linares

Escuela Técnica Superior de Ingenieros
Dept. de Ingeniería de Sistemas y Automática
Camino de los Descubrimientos s/n,
41092 Sevilla
ESPAÑA

UNIVERSIDAD DE SEVILLA
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA

DOCTORAL THESIS

Design Methods for Networked Control Systems with Unreliable Channels focusing on Packet Dropouts

Thesis submitted to the Departament of Ingeniería de Sistemas y Automática,
Escuela Técnica Superior de Ingeniería, in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the
University of Sevilla.

By

Isabel Jurado Flores

Sevilla, February 2013

To my parents

Acknowledgements

First of all, I would like to thank my parents, Miguel and María Luisa, for their support, love and education. I would also like to thank my sisters Eva, Ana and Irene, for their friendship and confidence.

I am deeply grateful to my thesis directors Francisco R. Rubio and Manuel G. Ortega, for all their patience, support and teachings. They have helped me with all the problems and inconvenients during my thesis.

In addition, I am gratefull to all my collegues and friends in the departament. Specially, I would like to thank all the PhD students in our department, in particular my friends Pepe, Luis, Antonio, Amalia, Rachad, Jorn, Carolina, Alicia, Alejandro, Guilherme, Filiberto, David, José Antonio, Guillermo, Cristina and Ramón.
I specially want to thank to Pablo Millán, for his friendship and his help during all this years. He has been an unvaluable friend and collegue.

Moreover, I would like to thank to Ministry of Education and Science for the doctorate grant (FPI) and the DPI projects that supported me during the develpment of my PhD. Besides, I thank the European Commission for founding part of my work through the Feed-Netback Project of the 7th Framework Programme.

Also, I am grateful to all the people I met in Australia, during the seven months I spent working there. I thank to my supervisor Daniel Quevedo for all his attention. He included me in his researchs and I did a lot of work collaborating with him. I am grateful for all the friends I made in Australia like Pierre, Diego, Kash, Boris, Alejandro, Ricardo, Katherine, Eduardo, Hope, Ramón, Rocío, Juan Carlos, Mauricio and Carmen.

Isabel Jurado Flores
Sevilla, June 2013.

Contents

List of Figures

List of Tables

Notation

$\mathbb{N}_0$ is used for $\{0,1,2,\ldots\}$; $\mathbb{R}$ are the real numbers, whereas $\mathbb{R}_{\geq 0} \triangleq [0,\infty)$. The ℓ-th unit row-vector in Euclidean space is denoted e_ℓ, for example, $e_1 = [1 \quad 0 \quad \ldots \quad 0]$, $e_2 = [0 \quad 1 \quad 0 \quad \ldots \quad 0]$; I_n is the $n \times n$ unit matrix, $0_n \triangleq 0 \cdot I_n$; $\otimes$ refers to the Kronecker product. The convention $\sum_{j=1}^{0} a_j = 0$ is adopted, for all $a_0, a_1 \in \mathbb{R}$. To denote the probability of an event Ω, we write $\mathbf{Pr}\{\Omega\}$. The expected value of a random variable μ given Δ, is denoted via $\mathbf{E}\{\mu \,|\, \Delta\}$, whereas for the unconditional expectation we write $\mathbf{E}\{\mu\}$. A real random variable μ, which is zero-mean Gaussian with covariance Γ is denoted by $\mu \sim \mathcal{N}(0,\Gamma)$. Mathematical variance is denoted as σ^2.

The trace of a matrix A is denoted by $\mathbf{tr}(A)$. The ∞-norm and the 2-norm are represented by $\|\cdot\|_\infty$ and $\|\cdot\|_2$, respectively. Given any matrix M, M^T denotes its tranpose; $M > 0$ and $M \geq 0$ denote that the matrix M is positive definite and positive semi-definite, respectively. The argument of the z-transform is represented as z. And $\mathbb{R}_{sp}$ is the subset of real rational discrete-time strictly proper transfer functions.

Σ is used to denote a nominal state-space model of a plant and Σ^* for a state-space model of a plant with uncertainties.

In the case of transfer funtions, $G(z)$ denotes a nominal discrete model of a plant while $G^*(z)$ is a discrete transfer function model of a plant with uncertainties.

x represents the state of a plant, u is the control signal and y denotes the output.

N is the number of consecutive dropouts induced by the network.

The probability of a successful communication is represented by p.

N_u is the prediction horizon.

$\mathbb{W} = \{x \in \mathbb{R}^n / \|x\| < \delta\}$.

$\delta_{i,j}$ is the Kronecker delta symbol.

$\varnothing$ represents a null matrix: $\begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$

Chapter 1

Introduction

Networked control systems have become a very important field in the control community due to its cost-effective and flexible applications. Networked control systems (NCSs) comprise sensors, actuators, and controllers, the operation of which is coordinated via a communication network. Typically, these systems are spatially distributed, may operate in an asynchronous manner, but have their operation coordinated to achieve desired overall objectives.

This chapter presents a summary of NCSs, and in particular, the specific cases in which this thesis is focused. The main issues related with NCSs, with the problems and advantages associated, are described in this section. Lastly, an outline of the present thesis together with its most relevant contributions are given.

1.1 Introduction to Networked Control Systems

Networked Control Systems (NCSs) are spatially distributed systems wherein the communication between plants, sensors, actuators and controllers occurs through a communication network. This kind of systems and their characteristics are extensively described in [J. P. Hespanha, P. Naghshtabrizi and Y. Xu, 2007], [W. Zhang, M. S. Branicky and S. M. Phillips, 2001], [R. A. Gupta, 2010] and [J. Chen, K. H. Johansson, S. Olariu, I. Ch. Paschalidis and I. Stojmenovic, 2011]. The complexity of system design and realization,

the wiring cost, installation and maintenance can be reduced drastically with the insertion of the communication network. However, the communication network in the system also brings some inconvenients, such as communication delays, data dropouts, codification errors and so on, which could degrade the system performance and even destabilize the system.

Nowadays, there is a large number of practical situations in where the use of communication networks for control is needed for the application or process advises control engineers. For example, they are specially needed in places where space and weight are limitated, when the distances under consideration are large and in control applications where the wiring is not possible.

There are also some generic advantages when using digital communication networks:

1. The complexity in point-to-point wiring connections are very reduced, as well as the costs of media. Therefore, installation costs can be also drastically reduced.
2. The reduction of the wiring complexity makes easier the diagnosis and maintenance of the system, providing cost savings because of the installation and higher operation efficiency.
3. NCSs are flexible and re-configurable.
4. Reliability, redundancy and robustness to failure.
5. NCSs provide modularity, control decentralisation and integrated diagnostics.

All these advantages suggest that NCSs will play a central role in the near future, being a very challenging and promising research field.

1.2 Objectives of the Thesis

The general idea of this thesis is to proposed some novel solutions to different problems related with NCSs. All the considered problems are very typical in the frame of control

over networks, mainly considering packet dropouts.

Within the context of systems with packet dropouts, different problems will be studied. In order to obtain different solutions for this kind of systems, the following objectives will be considered:

- Controller designs.

  H_∞ controllers, achieving the robustification of systems with uncertainties.

  MPC controllers, combined with buffer strategies.

- Filter designs.

  H_∞ filters for systems with uncertainties, using frecuency techniques and also Markov chains.

- Algorithm designs.

  Dynamic placement of a distributed control in a network formed by a matrix structure of nodes.

  Dynamic placement of the output estimator in a network formed by a line structure of nodes.

- Distributed cooperative estimation.

  Based on local Luenberger observers.

One of the objectives of this thesis will be to analyze the stability and performance of controlled systems. In some cases, the design will be done by means the stability constraints.

The robustification of the systems, in particular those ones with uncertainties, will be also taken into account. With respect to the analysis and design of controlled system, H_∞ techniques will be used.

Another important objective of this thesis will be the design of algorithms for a dynamic network, which will be composed by certain structure of nodes. The algorithm will be able to decide which node will be the controller or the output estimator in the network. Also stability and performance of controlled system will be analyzed.

The design of distributed estimation and schemes is also addressed. Networks with induced time-delays are considered, together with random dropouts. The reduction of energy consumption will be an important objective in this part of the thesis. In this case, an event-based communication policy between agents will be examined, providing a trade-off between performance and communication savings.

1.3 Related literature

There are a lot of studies in the literature about the main problems associated with NCSs. One way to approach time delay issues is to resort to Lyapunov-Krasovskii functionals (see, for example, [P. Millan, L. Orihuela, C. Vivas and F. R. Rubio, n.d.] and [D. Yue, Q. H. and J. Lam, 2005]). Another important topic to be studied in NCSs is the impact of network-induced data losses. Data dropouts occur, for instance, due to collisions or low SNR (signal to noise ratio) in wireless channels. There are different ways to deal with this kind of NCSs. One way is the use of predictive control, which makes it possible to calculate future model-based data and to use them to compute the control actions. Some examples of networked control based on MPC for linear and non-linear systems can be found in [P. Millan, I. Jurado, C. Vivas and F. R. Rubio, n.d.], [D. Muñoz and P. D. Christofides, 2008], [D. Quevedo, J. Østergaard and D. Nešić, 2011] and [D. Quevedo and D. Nešic, 2011].

A different way to deal with NCSs subject to data dropouts consists in modeling the

dropouts by means of a switched system, i.e., a Markov jump linear system (MJLS), see [O. L. V. Costa, M. D. Fragoso, and R. P. Marques, 2005]. Related with this approach, [Q. Ling and M. Lemmon, 2004] presents a result which shows that, for a specific NCS architecture subject to data dropouts, the resulting MJLS is equivalent to a linear loop with an external noise source. This noise has the particularity of having a variance that is proportional to the variance of another signal within the original control loop. This result is used in [E. I. Silva and S. A. Pulgar, 2011] to show that there exists a *second order moments equivalence* between the considered NCS and an auxiliary control system. In this auxiliary control system, the unreliable control channel has been replaced by an additive i.i.d. noise channel that has a Signal to Noise Ratio (SNR) constraint. In that paper, the probability of data losses is fixed and is used in the control synthesis. The objective in [E. I. Silva and S. A. Pulgar, 2011] is to minimize the error covariance designing the controller via a Youla parametrization. However, in that work only the ideal case of having a perfect LTI nominal model is considered. Therefore, robustness properties in the presence of plant model uncertainties are not guaranteed.

Some works on NCSs that take into account system uncertainties are [N. Elia, 2005], [H. Ishii, 2008] and [P. Seiler and R. Sengupta, 2005]. In [N. Elia, 2005] the modelling of the plant and the controller as deterministic time invariant discrete-time systems connected to zero-mean stochastic structured uncertainty is proposed. The variance of the stochastic perturbation is a function of the Bernoulli parameters, and the controller design is posed as an optimization problem to maximize mean-square stability margins of the closed loop system. H_∞ control approaches were proposed in [H. Ishii, 2008] and [P. Seiler and R. Sengupta, 2005]. In these works, the considered uncertainty comes from the unreliability of the network. In contrast, the present work considers both dropouts of the network and some structural uncertainties of the plant separately.

In a Networked Control System (NCS), sensor, controller and actuator links are not

transparent, but are affected by bit-rate limitations, packet dropouts and/or delays, leading to performance degradation; see, e.g., papers in the special issues [Antsaklis and Baillieul, 2004, 2007, Franceschetti et al., 2008, J. Chen, K. H. Johansson, S. Olariu, I. Ch. Paschalidis and I. Stojmenovic, 2011]. Ideally, communication artifacts can be alleviated by transmitting at high bit-rates and with high transmission powers [Pantazis and Vergados, 2007, Park et al., 2008, Quevedo et al., 2010, Cardoso de Castro et al., 2012]. However, if network nodes are wireless and connected to a finite power source, then energy efficiency becomes an important issue. This makes the design of NCSs often a challenging task.

An interesting aspect which has not been explored sufficiently is that of architectural freedom in the design of NCSs. When compared to traditional hard-wired control loops, wireless NCSs offer architectural flexibility and additional degrees of freedom. In particular, there is no need to pre-assign in a static fashion which nodes carry out control calculations, and which nodes merely relay data. Intuitively, and in relation to the packet dropout issue, the roles of individual nodes should depend upon the information available, thus, upon transmission outcomes.

As background, [Goodwin et al., 2008] studies performance of three static NCS architectures by adopting an additive signal-to-noise ratio constrained channel model. The results in that paper suggest that, in the absence of coding, placing the controller at the actuator node will give better performance than placing it at the sensor node. It is worth noting that [E. I. Silva and S. A. Pulgar, 2011] showed that the channel model in [Goodwin et al., 2008] can be used to describe erasure channels where dropouts are independent and identically distributed (i.i.d.). Viewed from that perspective, it was implicitly assumed in [Goodwin et al., 2008] that communication acknowledgments are not available at the transmitter. The work [C. L. Robinson and P. R. Kumar, 2008] examines NCSs with stochastic packet dropouts using optimal control techniques. The work shows that optimal control performance can be achieved if all nodes aggregate their entire history of received data

and relay it to the controller which is located at the actuator node. Depending upon the information available at each node, various optimal control problems can be analyzed, see [C. L. Robinson and P. R. Kumar, 2008] and also [Gupta et al., 2009] for a formulation where the controller is pre-allocated to a fixed node having perfect access to previous plant inputs. More recently, [Pajic et al., 2011] investigates a distributed control strategy wherein the network itself acts as a controller for a MIMO plant. Each node (including the actuator nodes) perform linear combinations of internal state variables of neighboring nodes. In the case of analog erasure channels with i.i.d. dropouts (without acknowledgments), in [Pajic et al., 2011] the resulting NCS is then cast, analyzed and designed as a jump-linear system.

In the area of distributed estimation, the problem typically involves limited processing capabilities of the agents, locally sensed information, and inter-component communications which are typically carried out asynchronously, wirelessly, and subject to limitations such as energy or physical constraints in the environment. Examples of active areas of application include sensor networks, transportation systems, network congestion control and routing, and autonomous vehicle systems among others [Akyildiz et al., 2002, Briñón Arranz et al., 2009, Cortes et al., 2004, Estrin et al., 1999, Lu et al., 2011, Xiao et al., 2005].

Regardless of the application, the common challenge remains the same. That is, to derive collective behaviors through the design of individual agent estimation and control algorithms. The primary distinguishing feature of this distributed approach is the distribution of information. As opposed to ‘centralized’ solutions, no single agent has access to the information gathered by all the agents. Since sensor networks are usually large scale systems, it is not advisable or even impossible to employ a centralized processor to gather all system data implementing classical centralized estimation techniques. Furthermore, there is typically a communication cost in distributing locally gathered information. A secondary distinguishing feature is complexity. Conventional decentralized estimation schemes can also be unattractive in many situations, provided that all-to-all communication is involved,

and scalability for high number of nodes is compromised, see [Olfati-Saber, 2005].

There exists a vast literature related to the problem of distributed estimation in sensor networks. The most common approach has been the distributed Kalman filter (DKF) based on consensus strategies. The methodology implies correcting the local estimations performed at each node based on the information received from their neighbors. See, for instance, [Alriksson and Rantzer, 2006, Maestre et al., 2010, Olfati-Saber and Shamma, 2005, Olfati-Saber, 2007].

Apart from the techniques employing Distributed Kalman Filter (DKF), there exists a number of works that propose different approaches. For example, distributed moving horizon schemes are employed in [Farina et al., 2009], where the solution requires each sensor to solve a quadratic optimization problem at every sampling time. A finite-horizon paradigm is proposed in [Dong et al., 2012] and [Shen et al., 2010] to design distributed observers that take into account quantization errors and successive packet dropouts. Another significant work from the same authors lying in this filed is that in [Shen, Wang and Xiaohui, 2011], in which a stochastic sampling between nodes is considered. A very interesting research direction considers more general models of the plant, including nonlinearities, inner delays, or different Markov-chain-driven dynamical modes. See, for instance, [Shen, Wang, Hung and Chesi, 2011], [Liang et al., 2011] or [Liang et al., 2012].

Despite the great deal of effort developed in distributed estimation, there is much room for research in the topic. Specifically, network-induced problems have historically received little attention. When a communication network is used to close estimation or control loops in real time, the assumption of perfect communication channels does not hold. Hence, reliable designs must be aware of network-induced constraints, significantly, delays and packet dropouts. These effects degrade the performance of a given estimation scheme or control implementation, even resulting in unstable behaviors, see [J. P. Hespanha, P. Naghshtabrizi and Y. Xu, 2007].

Furthermore, all the aforementioned works assume a time-driven scenario where the agents are required to broadcast their states at every sampling time. Event-based methods are somehow more efficient from the point of view of bandwidth use, as communications are invoked only when significant information requires to be transmitted, see [Dormido et al., 2008, Lunze and Lehmann, 2010] and [Tabuada, 2007]. This approach becomes specially beneficial in the context of distributed estimation over networks, as the limitations imposed by the network render the frequency at which the system communicates. A reduction in the transmission frequency implies bandwidth saving but also an improvement in average transmission delays and packet collisions, for back-off retransmission algorithms are reduced. Moreover, in wireless sensor networks, the battery life span is of great importance, and it is mainly related to the number of transmissions of the device. These two facts motivate the use of aperiodic communication policies, which allow to avoid the transmission of irrelevant data, reducing network traffic and energy expenditure. From the best of our knowledge, this is the first approach that considers the distributed estimation through non-reliable networks using an event-based sampling policy.

1.4 Contributions of the Thesis

In this section, a brief summary of the contributions of each chapter is presented.

In all chapters, an NCS wherein a communication channel introduces data dropouts is considered.

In Chapter 2, the plant model has structural uncertainties. Therefore, the main goal of that chapter is to find a robust controller for the plant with uncertainties and with data losses in the transmission; also minimizing the variance of the error signal. *Mean square stability (MSS)* and robustness properties also have to be guaranteed. An H_∞ control approach is proposed in such a way that both structural uncertainties in the plant and data losses can be tolerated, while optimizing the performance of the system. To deal with the plant uncertain-

ties, a central LTI controller will be calculated. This central controller will be combined with another transfer function, $Q(z)$, that will be in charge of the minimization of the error variance. In order to find $Q(z)$, an algorithm is proposed which gives a solution while satisfying the constraints. The union of these two transfer functions provides the proposed controller.

In Chapter 3 the same structure of NCS than in Chapter 2 is used. The difference here is that a mixed H_2/H_∞ control technique is proposed in this chapter (while in Chapter 2 only the H_∞ technique is used). The H_2 part is designed in such a way that the NCS is stabilized, taking into account the probability of data dropouts, while the H_∞ approach is used to make the closed-loop system robust enough against structural uncertainties of the nominal model.

Furthermore, the problem of the glucose control for diabetic patients subject to sensor errors constraints has been presented to illustrate the perfomance provided by this technique. To represent the uncertanties, different characteristics of patients have been considered for the synthesis of the controller.

In Chapter 4, structural uncertainties in the plant are considered also. One goal of this chapter is to find a robust controller for the plant with uncertainties, which will be carried out by means of an H_∞ control approach. Another important objective is to design a filter that calculates an estimation of the output of the plant. This estimation will be used when a packet dropout occurs, so the feedback will not become zero. *Mean square stability (MSS)* and robustness properties also have to be guaranteed. The filter design will be done with a technique based on the location of the unstable poles of the model of the plant. Further information can be found in [J. E. Normey-Rico and E. F. Camacho, 2009].

Also in Chapter 4, the networked control system is modelled as a MJLS and an LMI is derived in order to find a robust filter and controller by means of H_∞ techniques (see [S. Skogestad, and I. Postlethwaite, 2005]). The designed filter will calculate an estimation of

the state of the plant. This estimation will be used when a packet dropout occurs, so the feedback will not become zero.

Chapter 5 proposes a predictive control scheme focussing on the sensor/actuator vs. controller information interchange policy. Another concern in this chapter is the design of a strategy for networked linear systems with disturbances, with large data dropouts, retaining good performance. Additionally, limiting the amount of information transmitted in a Networked control system is a major concern. In this chapter, the effect of reducing the number of data packet exchanges between the controller and the actuator, while keeping an error threshold for the actuator control signals, is explored. This threshold allows to limit the amount of information through the network, transmitting only when relevant information for control is needed.

The network model considered allows for packet dropouts in both links, controller-to-actuator and sensors-to-controller. This motivates the inclusion of detection and compensation of missing packets resorting to buffering and state estimator respectively. To show the behaviour of the proposed compensation strategy, simulation results are provided on the level control problem of a three-tank system.

It has been also proposed to send from the controller a sequence of control signals that, appropriately buffered and scheduled at the actuator end, become a safeguard in case of delays or eventual packet dropouts. This concept naturally fits the model predictive control paradigm (MPC), and so has been reported in the literature.

In Chapter 5, it is supposed that the statistics of the time delays and dropouts can be measured or estimated with enough precision, exploiting this fact to design a stochastic packetized MPC to improve the control performance.

Chapter 6 studies NCS in which the network is composed of a certain number of nodes forming a matrix strucure. These nodes follow an algorithm, that decides which node will calculate the control input. This node will solve a cooperative MPC communicating with its

neighbors. Each node knows a part of the whole system model and it shares its information with a group of neighbor nodes, so they cooperate in order to exchange their information about the system. At each sampling time, a different group of nodes is chosen to calculate the control signal. This group of nodes will be chosen depending on the particular network outcomes for that sampling time. Chapter 6 extends the recent conference contribution [D. E. Quevedo, K. H. Johansson, A. Ahlén and I. Jurado, 2012] to encompass NCSs with parallel links and the use of cooperative MPC. The idea is motivated by the fact that the link transmission outcomes may change at each sampling instant, so one particular node is not always the best suited to perform the control calculation.

In the network under consideration in this work, the only node that receives the state of the plant without any dropouts is the sensor node, which is located next to the plant. The actuator node is directly connected to the plant input, therefore this data is received without problems. The actuator node is also the only node that provides transmission acknowledgments.

It is supposed that the model of the plant is divided into a certain number of incomplete subsystems. The control policy to be used will be a cooperative MPC.

With the algorithm proposed, transmission outcomes and their acknowledgments will determine, at each time instant, whether the control input will be calculated at the actuator node, or closer to the sensor node.

Chapter 6 also studies a single-loop NCS topology which uses a series connection of analog erasure channels. Thus, transmissions are affected by random packet dropouts.

Another flexible NCS architecture where the role played by individual nodes depends upon transmission outcomes is presented. With this algorithm, transmission outcomes determine, at each instant, which node will calcultate the state estimate.

In Chapter 7 a distributed cooperative estimation framework is discussed based on local Luenberger-like observers in combination with consensus strategies. Remarkably,

network-induced delays and packet dropouts are considered. The efficient use of the network resources receives important attention in both, time-driven periodic and event-based communication between the agents. The former approach reduces the amount of information communicated over the network resorting to two different ideas: on the one hand, only neighbors are allowed to communicate, reducing transmissions with respect to all-to-all communication schemes. On the other hand, the design of the observers contemplates the possibility of sharing only a part of the estimated state between neighbors, instead of communicating the whole estimated vector state. This economy in the use of network resources is, by its own nature, further improved with the event-driven communication approach.

1.5 List of publications

The following articles have been issued or submitted for publication during the elaboration of this thesis:

Journal Papers:

1. Pablo Millán, Luis Orihuela, Isabel Jurado Carlos Vivas and Francisco R. Rubio. *Distributed Estimation in Networked Systems Under Periodic and Event-Based Communication Policies*. International Journal of System Science.

2. Isabel Jurado, Manuel G. Ortega, Daniel Quevedo and Francisco R. Rubio. *An H_∞ suboptimal robust control approach for Networked Control Systems with uncertainties and data dropouts*. Submitted to IET Control Theory Applications.

3. P. Millán, L. Orihuela,I. Jurado, F.R. Rubio. *Control of autonomous underwater vehicle formations subject to inter-vehicle communication problems*. Submitted to IEEE Transactions on Control Systems Technology.

4. Daniel Quevedo, Karl H. Johansson, Anders Ahlén and Isabel Jurado. *An Adaptive Architecture for Control using Erasure Channels*. Submitted to Automatica.

5. Daniel E. Quevedo and Isabel Jurado. *Stability of Sequence-Based Control with Random Delays and Dropouts*. Submitted to IEEE Transactions on Automatic Control.

Book Chapters:

1. Isabel Jurado, Daniel E. Quevedo, Karl H. Johansson and Anders Ahlén. *Cooperative Dynamic MPC for Networked Control Systems*.

   Book: *Distributed MPC Made Easy*

   Editors: Dr. José M. Maestre and Dr. Rudy R. Negenborn.

Conference Papers:

1. Daniel E. Quevedo , Karl H. Johansson, Anders Ahlén and Isabel Jurado. *Dynamic Controller Allocation for Control over Erasure Channels*. 3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys), 2012.

2. Isabel Jurado, Daniel E. Quevedo, Pablo Millán, Francisco R. Rubio. *Stochastic packetized model predictive control for networked control systems subjects to time-delays and dropouts*. 20th International Symposium on Mathematical Theory of Networks and Systems, Melbourne (Australia), 2012.

3. Isabel Jurado, Manuel G. Ortega and Francisco R. Rubio. *Networked Mixed H_2/H_∞ robust control approach for NCS with uncertainties and data dropouts*. Proceedings of the 18th World Congress of the International Federation of Automatic Control (IFAC), 2011.

4. Isabel Jurado, Manuel G. Ortega, Francisco R. Rubio and Julio E. Normey-Rico. *A robust filter and controller design for NCS with uncertainties and data dropouts*.

Proceedings the 9th IEEE International Conference on Control Automation (IEEE ICCA), 2011.

5. Isabel Jurado, Manuel G. Ortega, Francisco R. Rubio y Julio E. Normey-Rico. *Diseño de un filtro y un controlador robusto para NCS con incertidumbres y pérdidas de datos*. Actas de las XXXII Jornadas de Automática, Sevilla, España, Septiembre 2011.

6. Isabel Jurado, Zaui Mahmud, Manuel G. Ortega and Antonia Ortega Linares. *Adaptación para el control a través de red de un sistema de levitación neumática*. Actas de las XXXI Jornadas de Automática, Jaén, España, Septiembre 2010.

7. Javier Arriaga, Pablo Millán, Isabel Jurado, Carlos Vivas and Francisco R. Rubio. *Application of network-based robust control to a personal pendulum vehicle*. European Control Conference (ECC), Budapest (Hungary), p. 400-405, 2009.

8. Isabel Jurado, Manuel G. Ortega and Francisco R. Rubio. *Comparación de estrategias de control para la orientación de un quadrotor controlado a través de red*. Actas de las XXX Jornadas de Automática, Valladolid, España, Septiembre 2009.

9. Pablo Millán, Isabel Jurado, Carlos Vivas and Francisco R. Rubio. *Networked predictive control of systems with large data dropouts*. 47th IEEE Conference on Decision and Control (CDC'08), Cancun (Mexico), 2008.

10. Pablo Millán, Isabel Jurado, Carlos Vivas and Francisco R. Rubio. *An algorithm to compensate for large data dropouts in networked control systems*. Proceedings of the 13th IEEE international Conference on Emerging Technologies and Factory Automation (ETFA'08), Hamburg (Germany), 2008.

11. Pablo Millán, Isabel Jurado, Manuel G. Ortega, Francisco R. Rubio. *Estudio comparativo de diferentes técnicas de cuantificación*. Actas de las XXVIII Jornadas de Automática, Tarragona (España), 2008.

12. Javier Arriaga, Pablo Millán, Isabel Jurado, Carlos Vivas, Francisco R. Rubio. *Técnicas de control a través de red inalámbrica de un vehículo basado en péndulo invertido*. Actas de las XXVIII Jornadas de Automática, Tarragona (España), 2008. (Best paper award in Control Engineering)

13. Pablo Millán, Isabel Jurado, Carlos Vivas, Francisco R. Rubio. *Modelado y control de una instalación para ensayos de sistemas de intercambio de presión*. Actas de las XXVII Jornadas de Automática, Huelva (España), 2007.

Chapter 2

An H_∞ suboptimal robust control approach for Networked Control Systems with uncertainties and data dropouts

This chapter studies the design of Networked Control Systems subject to plant uncertainties and data losses. The controller design has two main objectives. The first one is to robustify the control law against plant uncertainties. The other one is to achieve good performance by minimizing the variance of the error signal. Data losses are modelled as an independent-identically distributed (i.i.d.) sequence of Bernoulli random variables. For analysis and design, this random variable is replaced by an additive noise plus gain channel, which is equal to the successful transmission probability in the feedback loop. Also, structural uncertainties in the model of the plant are considered. To cope with the latter, an H_∞ control technique is proposed. The controller is synthesized in order to make the closed-loop system robust against structural uncertainties of the nominal model, while achieving optimal performance of the system in the presence of dropouts.

Therefore, the main goal of this chapter is to find a robust controller for the plant with uncertainties and with data losses in the transmission; also minimizing the variance of the error signal. *Mean square stability (MSS)* and robustness properties also have to be guaran-

teed. An H_∞ control approach is proposed in such a way that both structural uncertainties in the plant and data losses can be tolerated, while optimizing the performance of the system. To deal with the plant uncertainties, a central LTI controller will be calculated. This central controller will be combined with another transfer function, $Q(z)$, that will be in charge of the minimization of the error variance. In order to find $Q(z)$, an algorithm is proposed which gives a solution while satisfying the constraints. The union of these two transfer functions provides the proposed controller.

2.1 H_∞ control problem

In this section, a brief summary of H_∞ control is presented. Further information can be found in [K. Zhou, J. C. Doyle, and K. Glover, 1996], [J. C. Doyle, K. Zhou, K. Glover and B. Bodenheimer, 1994] and [S. Skogestad, and I. Postlethwaite, 2005]. The control system described in Figure 2.1 is considered, where the generalized plant $P(z)$ and the controller $C(z)$ are both assumed to be real-rational and proper. The signals involved in the diagram are the following: $w \in \mathbb{R}^{m_2}$ represents the disturbance vector, $u \in \mathbb{R}^{m_1}$ is the control input, $z_\infty \in \mathbb{R}^{p_1}$ is the error vector used in quantifying H_∞ performance. The measurement supplied to the controller is represented by $m \in \mathbb{R}^{p_2}$.

The synthesis problem considered in this approach consists in finding an LTI controller $C(z)$ that minimizes the following H_∞ criterion:

$$Min \quad \gamma, \quad subject \quad to: \quad \|T_\infty\|_\infty < \gamma \quad \gamma \in \mathbb{R}^+, \tag{2.1}$$

where $T_\infty(z)$ denotes the closed-loop transfer function from w to z_∞. It is given by the lower linear fractional transformation:

$$T_\infty(z) = P_{11}(z) + P_{12}(z)C(z)(I - P_{22}(z)C(z))^{-1}P_{21}(z),$$

with

$$P(z) = \begin{bmatrix} P_{11}(z) & P_{12}(z) \\ P_{21}(z) & P_{22}(z) \end{bmatrix}.$$

The transfer function $T_\infty(z)$ is chosen to represent a mixed-sensitivity H_∞ control problem, as explained in [S. Skogestad, and I. Postlethwaite, 2005]. For closed loop transfer functions, $S(z)$ is used to denote the ouput sensitivity of the system, and $T(z)$ to denote the output complementary sensitivity. Thus, $S(z) = (1+L(z))^{-1}$ and $T(z) = L(z)(1+L(z))^{-1}$, where $L(z)$ is the transfer function around the loop as seen from the output. Then, two weighting functions may be chosen: $W_s(z)$ to weight the sensitivity function $S(z)$ and $W_t(z)$ to weight the complementary sensitivity function $T(z)$. These weighting functions allow to specify the range of frequencies of relevance for the corresponding closed-loop transfer matrix. As it is known, an appropriate shaping of $T(z)$ is desirable for tracking problems, noise attenuation and for robust stability with respect to multiplicative output uncertainties. On the other hand, an appropiate shaping of $S(z)$ will allow to improve the performance of the system avoiding steady state errors. Thereby, this approach is useful to have an appropriate performance on tracking problems, as well as for the system robustification against noises and uncertainties.

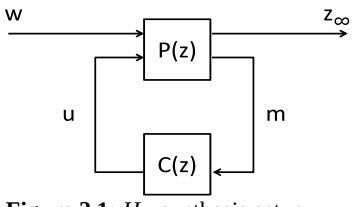

Figure 2.1: H_∞ synthesis setup

Given the above, in order to design a networked controller by means of this control technique, it is necessary to put the original system under consideration into the form shown in Figure 2.1.

2.2 Problem definition

This chapter is focused on a Robust NCS (RNCS) wherein the main problems are the uncertainties in the plant model and the packet dropouts. Therefore, the aim is to design a controller that stabilizes a system subject to these two issues. Also, it is required to achieve a small variance of the error signal.

The plant model under consideration is represented by:

$$G^*(z) = G(z)(I + W_I(z)\Delta(z)), \tag{2.2}$$

where $G^*(z)$ represents the family of all the possible plants, $G(z)$ is the nominal plant model and $W_I(z)\Delta(z)$ is the multiplicative uncertainty, with $\|\Delta(z)\|_\infty < 1$.

In the following, the way to deal with the information losses is presented.

The packet dropouts imply that there is an unreliable channel in the feedback path. This situation is illustrated in Figure 2.2, where $C(z)$ is the controller, r is the reference and y is the plant output. The relation between the channel input v and the channel output w_p is:

$$w_p(k) \doteq (1 - d_r(k))v(k), \quad \forall k \in \mathbf{N}_0, \quad \forall v(k) \in \mathbf{N}, \tag{2.3}$$

where d_r models data losses, $d_r(k) \in \{0,1\}, \quad \forall k \in \mathbf{N}_0$.

The plant under consideration is a random one, due to the existence of an unreliable channel with random data dropouts. In this case, the probability of a successful communication is given by $p \in (0,1)$. The plant model is fixed but unknown, because of the uncertainties considered, see (2.2).

The following notion of stability is adopted:

Definition 1 (Mean square stability) [O. L. V. Costa, M. D. Fragoso, and R. P. Marques, 2005] Consider a system described by $x(k+1) = f(x(k), w(k))$, where $k \in \mathbb{N}_0$, $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, $x(k) \in \mathbb{R}^n$ is the system state at time instant k, $x(0) = x_0$, where x_0 is a second order random variable, and the input w is a second order *wss* (wide-sense stationary)

process independent of x_0. The system is said to be mean square stable (MSS) if and only if there exist finite $\mu \in \mathbb{R}^n$ and finite $M \in \mathbb{R}^{n\times n}$, $M \geq 0$, such that

$$\begin{aligned} \lim_{k\to\infty} \mathbf{E}\{x(k)\} &= \mu, \\ \lim_{k\to\infty} \mathbf{E}\{x(k)x(k)^T\} &= M, \end{aligned} \tag{2.4}$$

regardless of the initial state x_0.

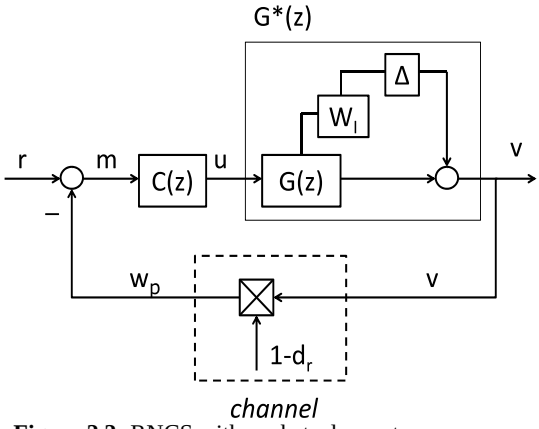

Figure 2.2: RNCS with packets dropouts

Assumption 1.

- The process d_r is an independent sequence of i.i.d. Bernoulli random variables with $\mathbf{P}\{d_r(k) = 1\}$ =$1 - p$ for all $k \in \mathbb{N}_0$.

- The plant transfer function $G(z)$ belongs to $\mathbb{R}_{sp}$, is SISO, non-zero, has no zeros or poles on the unit circle, and has a stabilizable and detectable underlying realization.

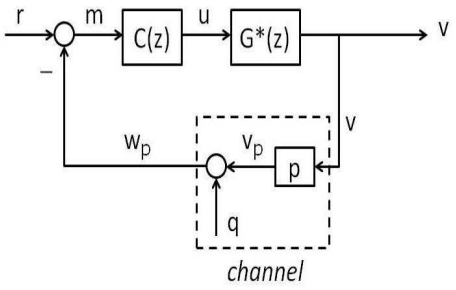

Figure 2.3: Equivalent RNCS with packets dropouts

- The initial states of both the plant and the controller are jointly second order variables.

The following result establishes an equivalence between the system with dropouts and an associated LTI one.

Theorem 1. (Equivalence, [Q. Ling and M. Lemmon, 2004], [E. I. Silva and S. A. Pulgar, 2011]) Consider the feedback loops in Figures 2.2 and 2.3. The signal q in Figure 2.3 is an independent sequence of i.i.d. random variables having zero mean and a variance σ_q^2 that satisfies

$$\sigma_q^2 = p(1-p)\sigma_v^2,$$

provided the stationary variance of v exists and is finite.

Let us suppose that $p \in (0,1)$ and that Assumptions 2 and 8 from [E. I. Silva and S. A. Pulgar, 2011] hold. Then:

1. If the feedback system depicted in Figure 2.2 is MSS and the feedback system in Figure 2.3 is internally stable, then the stationary PSDs (Power Spectral Densities)

of the error ($e \doteq r - y$) and of all the corresponding signals in the loops are the same in both situations.

2. The networked system in Figure 2.2 is MSS if and only if the feedback loop in Figure 2.3 is asymptotically stable and

$$\frac{p}{1-p} > \left\| T_p(z) \right\|_2^2, \tag{2.5}$$

where $T_p(z)$ is the transfer function from q to v_p in Figure 2.3, namely

$$T_p(z) \doteq -pG^*(z)C(z)(1 + pG^*(z)C(z))^{-1}. \tag{2.6}$$

As a consequence of Theorem 1, ensuring MSS of the system in Figure 2.2 is equivalent to achieving stability of the system in Figure 2.3, while at the same time satisfying condition (2.5). Also, minimizing the variance of a signal is equivalent in both systems. Thereby, the problem can be posed as the searching of a controller $C(z)$ that stabilizes the system in Figure 2.3, satisfies (2.5) and minimizes the variance of the error, taking into account that the plant $G(z)$ is the nominal plant model and that the closed-loop system must be robust against the uncertainties in the plant model.

As mentioned before, a contribution of the current work is that structural uncertainties are considered in the model of the plant $G^*(z)$. Due to this fact, the mixed sensitivity approach within the H_∞ scope allows to impose robust performance by means of appropriate design of weighting functions. In particular, as stated in Section II, robust stability can be guaranteed by weighting the complementary sensitivity function if structural multiplicative uncertainty is considered ([M. G. Ortega and F. R. Rubio, 2004], [M. G. Ortega, M. Vargas, L. F. Castaño and F. R. Rubio, 2006]), while performance characteristics can be imposed by means of a reasonable weight on the sensitivity function. Using this approach a *central* H_∞ controller will be obtained.

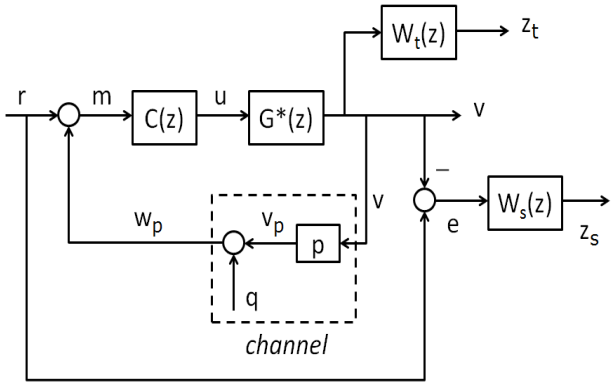

Figure 2.4: RNCS and the weighting transfer functions

To incorporate condition (2.5), an additional transfer function ($Q(z)$) is designed. Therefore, combining the central H_∞ controller with this transfer function, the final H_∞ controller is obtained. The above leads to the following control design problem:

Problem 1 Consider the RNCS in Figure 2.2 where the plant $G(z)$ has bounded structural multiplicative uncertainties. Then, the problem consists in finding a robust controller $C(z)$, using the RNCS in Figure 2.3, that achieves the following conditions simultaneously:

- Minimize $\|T_\infty\|_\infty$ to achieve a good performance on tracking problems and the system robustification against the plant uncertainties (see Figure 2.4).

- Minimize the variance of the error signal e.

The first objective will be achieved by means of the H_∞ central controller. To tackle the second one, the parametrization in (2.8) will be used. The parameter will be found solving an algorithm that seeks to minimize the variance of the error signal and satisfies an appropriate constraint.

2.3 Controller design

In this section, the controller synthesis procedure is described. The central controller is calculated by means of the described H_∞ control technique. Some weighting transfer functions will be considered in the system to deal with the uncertainties of the plant model. The augmented system is represented in Figure 2.4. The weighting transfer functions $W_s(z)$ and $W_t(z)$ weight the sensitivity function ($S(z)$) and the complementary sensitivity function ($T(z)$), respectively. The outputs of these weighting transfer functions are the signals z_s and z_t respectively. The latter represent the components of the vector $z_\infty = \begin{bmatrix} z_s \\ z_t \end{bmatrix}$ in Figure 2.1.

One of the considered weighting functions is the sensitivity transfer function, that is the one from the reference to the error signal. The other used transfer function $(T(z))$ depends on the open-loop transfer function of the system, $L(z)$. Thus, the output signal v is taken as the input of the weighting transfer function $W_t(z)$, as represented in Figure 2.4.

The final H_∞ controller is built by a linear fractional transformation of the central controller and a transfer function $Q(z)$ (see Figure 2.5). $Q(z)$ is calculated in order to deal with the optimization problem of minimizing the error variance. Therefore, the problem can be written as:

$$Q(z)_{opt} = arg \quad inf \quad [\sigma_e^2]_{Q(z)}, \tag{2.7}$$

where σ_e^2 represents the variance of the error.

It is important to note that the resulting controller has the following structure:

$$C(z) = K_{c11}(z) + K_{c12}(z)(1 - Q(z)K_{c22}(z))^{-1}Q(z)K_{c21}(z) \tag{2.8}$$

where $Q(z)$ has to satisfy the constraint:

$$\|Q(z)\|_\infty < \gamma$$

where γ is the H_∞ perturbation attenuation parameter in (3.1), and

$$K_c(z) = \begin{bmatrix} K_{c11}(z) & K_{c12}(z) \\ K_{c21}(z) & K_{c22}(z) \end{bmatrix}$$

In Figure 2.5, the structure of the global controller is represented. In the present work, a fixed structure has been considered for $Q(z)$. This is because it is necessary for the search method described below to know how many parameters have to be found with the iterations, so the best possible performance is achieved. In this case, it is supposed that $Q(z)$ is a first order transfer function, therefore the parameters are a gain, a pole and a zero. Therefore, a first order transfer function is chosen for $Q(z)$. With this structure, there are three parameters to take into account to minimize the error of the system, and each one is affecting in a different way. A first order system is also chosen in order to not increase excessively the computational cost of the algorithm. Thereby the structure is:

$$Q(z) = \frac{K_Q(z - c_Q)}{(z - p_Q)}$$

where $c_Q \in (-1,1)$ and $p_Q \in (-1,1)$, this way, the $Q(z)$ is stable and with minimun phase. Imposing the condition $\|Q(z)\|_\infty < \gamma$, it can be seen that $K_Q < \gamma\frac{1+p_Q}{1+c_Q}$ if $c_Q > p_Q$ and $K_Q < \gamma\frac{1-p_Q}{1-c_Q}$ if $c_Q < p_Q$. If $c_Q > p_Q$, then the H_∞ norm is localized when $z = -1$, and if $c_Q < p_Q$ it is localized when $z = 1$.

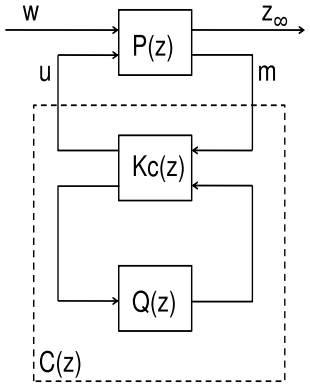

Figure 2.5: Controller structure

To find $Q(z)_{sub}$, which is an approximation of $Q(z)_{opt}$, the following numerical search over a grid is used, for any fixed and known $p \in (0,1)$:

---

1. Define a grid for K_Q, c_Q and p_Q:
   $\tilde{c}_Q = \{c_{Q_1}, ..., c_{Q_{ic}}, ..., c_{Q_{nc}}\}$, $c_{Q_1} \geq -1$ and $c_{Q_{nc}} \leq 1$, $\forall ic \in [1, nc]$.
   $\tilde{p}_Q = \{p_{Q_1}, ..., p_{Q_{ip}}, ..., p_{Q_{np}}\}$, $p_{Q_1} \geq -1$ and $p_{Q_{np}} \leq 1$, $\forall ip \in [1, np]$.
   $\tilde{K}_Q = \{K_{Q_1}, ..., K_{Q_{iK}}, ..., K_{Q_{nK}}\}$.
2. Find the variance of the error ([E. I. Silva and S. A. Pulgar, 2011]), $\left[\sigma_e^2\right]_{Q_i(z)}$ for every $Q_i(z) \in \tilde{Q}(z)$, where $\tilde{Q}(z) = \frac{\tilde{K}_Q(z-\tilde{c}_Q)}{(z-\tilde{p}_Q)}$.
3. Define i_{K^*}, i_{c^*} and i_{p^*} as the indexes i_K, i_c and i_p associated to the smallest $\left[\sigma_e^2\right]_{Q_i(z)}$. Approximate the optimal variance of the error by $\left[\sigma_e^2\right]_{Q_i^*(z)}$.

Therefore, with $K_{Q_{iK^*}}, c_{Q_{ic^*}}, p_{Q_{ip^*}}$, it is possible to approximate $Q(z)_{opt}$ with the desired precision. In [E. I. Silva and S. A. Pulgar, 2011] a similar procedure is used but the global controller is designed with a Youla parametrization. This leads to a 2-norm minimization problem of certain matrix transfer functions, which is solved in a specific way that, in general, can not be extrapolated to another controller parametrization. The advantage of the method presented in the present work is that it is not necessary to know exactly the model of the plant, since structural uncertainties are considered.

As mentioned before, the H_∞ control problem will be solved to find an optimal controller which achieves the system robustification against the plant uncertainties and gives optimal performance. To carry out the synthesis, the system in Figure 2.4 has to be expressed, by means of a *lower linear fractional transformation*, in the form as in Figure 2.1. It is easy to see that, by identifying the terms, the followings expressions hold:

$$w = [r \quad q]^T,$$

$$P(z) = \left[\begin{array}{cc|c} W_s(z) & 0 & -W_s(z)G(z) \\ 0 & 0 & W_t(z)G(z) \\ \hline I & -I & -pG(z) \end{array}\right]$$

With respect to the minimization problem in (3.1), $T_\infty(z)$ is chosen as follows:

$$\|T_\infty(z)\|_\infty = \left\| \begin{bmatrix} W_s(z)S(z) \\ W_t(z)T(z) \end{bmatrix} \right\|_\infty$$

2.4 Stability analysis

For the stability analysis, the system in Figure 2.4 is represented as a state space system. As a consequence of Theorem 1, if the stability of the system in Figure 2.4 is proven, the system in Figure 2.2 will be also stable.

Notice that constraint (2.5) is imposed in the design of the controller.

The system under consideration is:

$$\begin{aligned} x(k+1) &= A'x(k) + B'_w w(k) \\ z_\infty(k) &= C'x(k) \end{aligned} \tag{2.9}$$

where A' and B' are the state space realization of the system $PC(z)$ shown in 2.6. The controller $C(z)$ is obtained following the equation (2.8), as explained in Section 2.3.

Theorem 2. Given $\gamma > 0$ (obtained from the controller synthesis in Section 2.3) and matrices A' and B', the system (3.5.2) is stable and $\frac{\|z_\infty\|_2}{\|w\|_2} < \gamma$ if and only if there exists a matrix X, such that the following matrix inequality holds:

$$\begin{bmatrix} -X & 0 & X^T A'^T & X^T C'^T \\ 0 & -\gamma I & {B'_w}^T & 0 \\ A'X & B'_w & -X & 0 \\ C'X & 0 & 0 & -I \end{bmatrix} < 0 \tag{2.10}$$

where $X = P^{-1}$, with $P > 0$.

Proof:

Consider

$$V(k) = x^T(k)Px(k)$$

with $P > 0$, and seek:

$$V(k+1) - V(k) = x^T(k+1)Px(k+1) - x^T(k)Px(k) < 0$$

By incorporating the H_∞ condition: $\frac{\|z_\infty\|_2}{\|w\|_2} < \gamma$, the following expression is obtained:

$$\begin{aligned} x^T(k)A'^T PA'x(k) + x^T(k)A'^T PB'_w w(k) + w^T(k){B'_w}^T PA'x(k) + \\ w^T(k){B'_w}^T PB'_w w(k) - x^T(k)Px(k) + z_\infty^T(k)z_\infty(k) - \gamma w^T(k)w(k) < 0 \end{aligned} \tag{2.11}$$

Defining $\xi = [x \quad w]^T$, the equation (2.11) is equivalent to:

$$\begin{aligned} \xi^T \begin{bmatrix} -P & 0 \\ 0 & -\gamma I \end{bmatrix} \xi + \xi^T \begin{bmatrix} A'^T \\ {B'_w}^T \end{bmatrix} P \begin{bmatrix} A' & {B'_w}^T \end{bmatrix} \xi + \\ \xi^T \begin{bmatrix} C'^T \\ 0 \end{bmatrix} \begin{bmatrix} C' & 0 \end{bmatrix} \xi < 0 \end{aligned} \tag{2.12}$$

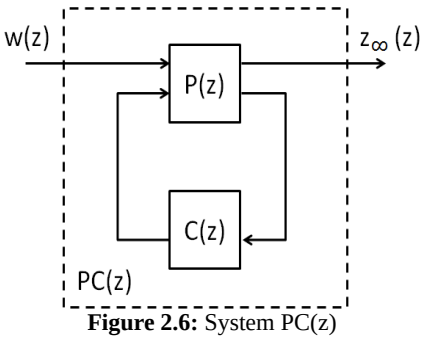

Figure 2.6: System PC(z)

Using Schur complement and with the following multiplication,

$$\begin{bmatrix} P^{-1} & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \begin{bmatrix} -P & 0 & A'^T & C'^T \\ 0 & -\gamma I & B_w'^T & 0 \\ A' & B_w' & -P^{-1} & 0 \\ C' & 0 & 0 & -I \end{bmatrix} \begin{bmatrix} P^{-1} & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} < 0$$

the linear matrix inequality (2.10) is obtained.

The result follows upon noting that, for linear systems, the existence of a matrix $P > 0$ such that (2.10) holds, is a necessary and sufficient condition for the stability of the system, [J. Daafouza and J. Bernussoub, 2001].

2.5 Numerical example

To illustrate the methodology proposed in this paper, the following unstable nominal plant is considered:

$$G(z) = \frac{z-0.5}{z(z-1.1)},$$

with sampling time $t_m = 0.05s$.

Let us consider a transmission channel with a successful probability $p = 0.7$. Later, it will be checked if this probability is enough to keep the systems equivalence and, consequently, the mean square stability of the system.

The plant under consideration has parametrical uncertainties, in the gain and in the dynamics as well. Also structural uncertainties are considered, as unmodelled dynamics. In this example, two non-nominal model structures are considered. To obtain these models, two high frequency poles are included and a percentage of uncertainty in the model gain and pole has been considered. From these two systems and the nominal plant, multiplicative uncertainties can be derived. The frequency response of these uncertainties have been plotted in Figure 2.7.

The two non-nominal models have the following expressions:

$$G_1(z) = \frac{0.884(z-0.5)}{z(z-1.111)(z-0.1^{10})}$$

$$G_2(z) = \frac{1.075(z-0.5)}{z(z-1.09)(z-0.2^{15})}$$

From this estimation of the uncertainty bound, the weighting transfer function $W_t(z)$ for the complementary sensitivity function is designed in such a way that its modulus is greater than the modulus of the uncertainties at all frequencies (see Figure 2.7):

$$W_t(z) = \frac{2.438z - 1.995}{z + 0.9802} \tag{2.13}$$

$$W_s(z) = \frac{0.07012z - 0.04207}{z - 0.995} \tag{2.14}$$

W_s is chosen taking into account that at low frequencies the gain of W_s^{-1} has to be very small, in order to avoid steady state errors.

By solving the H_∞ control problem for this case, using some functions of the $\mu-$*Analysis and Synthesis Toolbox* for Matlab, and considering a success probability $p = 0.7$, a robust controller is obtained yielding the following performance measure:

$$\|T_\infty\|_\infty = 0.7893 < 1$$

The fact that $\|T_\infty\|_\infty < 1$ assures that the sensitivity functions are staying below their bounds.

To obtain the global controller, $Q(z)$ can be calculated by means the algorithm presented in the previous section, finding the following result:

$$Q(z)_{sub} = \frac{0.2(z-0.4)}{z-0.3}$$

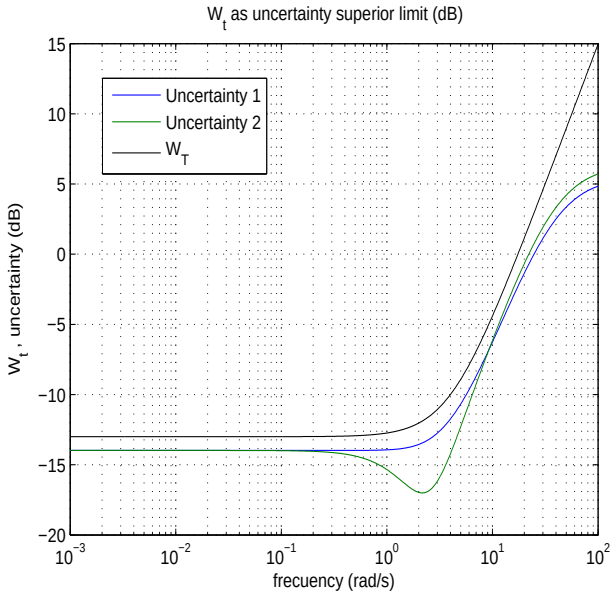

Figure 2.7: Uncertainties and W_t

$$\|Q(z)_{sub}\|_\infty = 0.2154 < \gamma = 0.7893$$

Once the controller is obtained, it is checked that all the sensitivity functions are below the inverse of W_s, as it is shown in Figure 2.8.

Figure 2.8 illustrates the sensitivity functions of the nominal and non-nominal plant models and the inverse of the weighting transfer function $W_s(z)$. This figure shows how all the sensitivity functions, the one of the nominal system and the other non-nominal systems, are bounded (in magnitude) by the inverse of the weighting function $W_s(z)$ due to the fact that the achieved value of γ is lesser than one. This fact indicates that the output v can follow the reference r for all the plant models under consideration, that is, a tracking problem can be solved although the plant model is not exactly known.

Figure 2.9 represents the complementary sensitivity functions of the nominal and non-

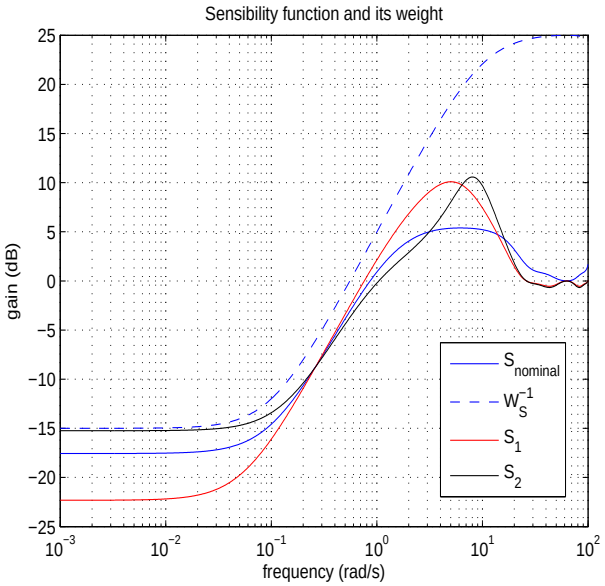

Figure 2.8: $S(z)$ of the nominal plant model and $W_s^{-1}(z)$

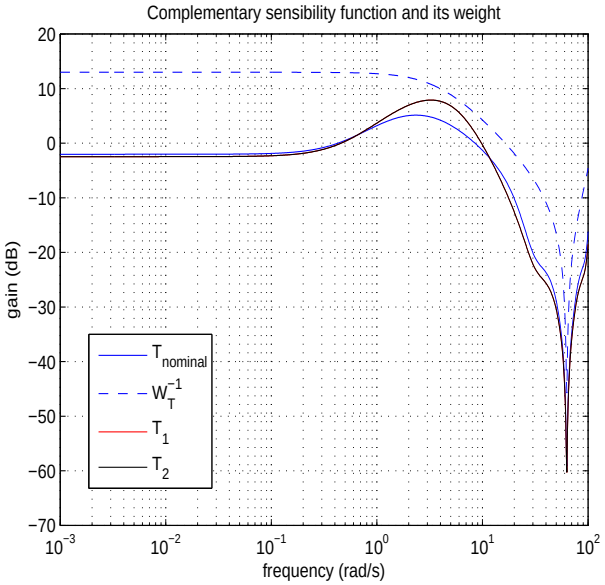

Figure 2.9: $T(z)$ of the nominal plant model and $W_t^{-1}(z)$

nominal plant models and the inverse of the weighting transfer function $W_t(z)$. Clearly, all the complementary sensitivity functions, the one of the nominal system and the other non-nominal systems, lie below the inverse of the weighting function $W_t(z)$. Therefore, the obtained controller is robust against the uncertainties in the plant model.

To verify these results, some simulations have been carried out. These simulations have been also obtained for the different $Q(z)$. Figure 2.10 shows how the system follows the reference with a successful transmission probability $p = 0.7$, which is greater than the minimal p needed to achieve MSS and robustness properties for this system, which is obtained from (2.5):

$$p_{min} = \frac{\left\|T_p(z)\right\|_2^2}{1+\left\|T_p(z)\right\|_2^2},$$

yielding $p_{min} = 0.65$, since $\left\|T_p(z)\right\|_2^2 = 1.36$ in the case of $Q_{opt}(k)$, and (2.5) has to be satisfied. This figure illustrates output trajectories of the closed-loop system with the nominal plant $G(z)$, with plant $G_1(z)$ and with the plant $G_2(z)$. The results are very similar because of the robustness of the system. However, there exist some differences between the different outputs. For example, the output with $G_1(z)$ has an overshoot that is greater than the overshoot when the nominal model is used. With respect to the output with $G_2(z)$, the overshoot is reduced with respect the other cases, but the stationary performance is worse.

Illustrative outputs of the different systems for a success probability of $p = 0.9$ are presented in Figure 2.11. In this case, the probability of success in the transmission has been increased, although the controller used in these simulations is the one calculated for $p = 0.7$. Clearly, the results are better than the ones presented in Figure 2.10, but the differences between the performance with the different systems is the same as in the case of $p = 0.7$.

Finally, Figure 2.12 presents the outputs of all systems with $p = 0.4$, while using the

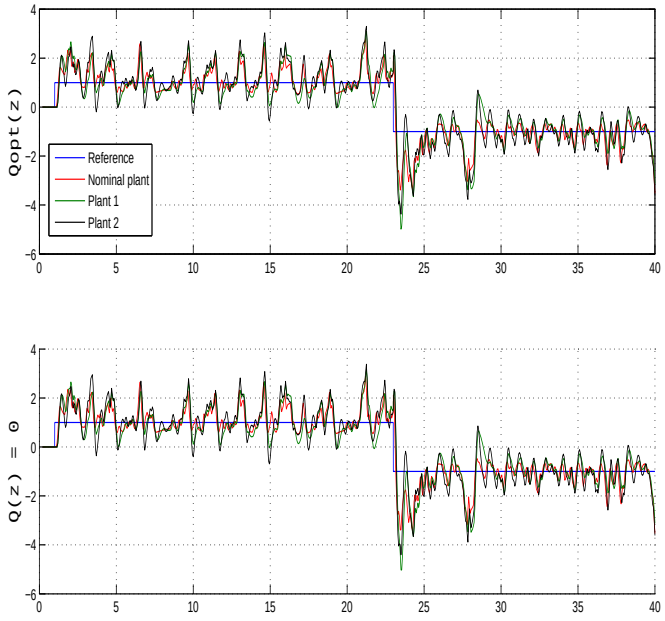

Figure 2.10: Simulation results with p = 0.7

same controller as in the preceding simulations. Obviously, the performances get worse for all systems, and in the case of the plant with the uncertainties 1, the closed-loop system becomes unstable. Therefore, with $p = 0.4$, the robust stability is lost.

Some representative results (error variances) are shown in Table 2.1, where two different cases are considered. In one case $Q(z)_{opt}$ is used and in the other the central controller is used, thereby $Q(z) = 0$. In the cases, the constraint $\|Q(z)\|_\infty \leq \gamma$ holds. The error variances are worse in the case of the non-optimal $Q(z) = 0$. Clearly, as p increases, performance improves.

The case of a traditional H_∞ controller has been also considered. This controller does not take into account the data losses in the system. As expected, the closed loop system becomes unstable for all scenarios considered in this work.

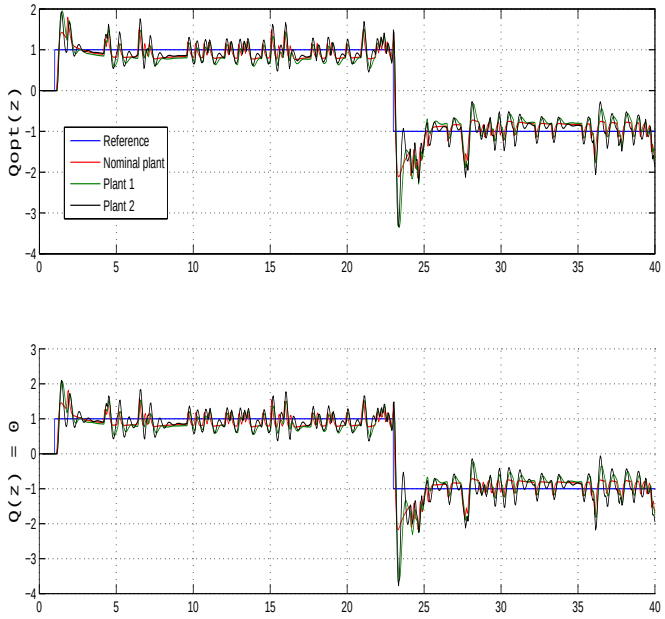

Figure 2.11: Simulation results with p = 0.9

Table 2.1: Error variances

$p = 0.7$	Plant		
$Q(z)$	$G(z)$	$G_1(z)$	$G_2(z)$
Q_{opt}	1.71	1.84	2.04
0	1.738	1.867	2.09
$p = 0.9$			
Q_{opt}	0.995	1.01	1.09
0	0.9998	1.023	1.127
$p = 0.4$			
Q_{opt}	5.96	29.47	8.98
0	6.155	34.09	9.093

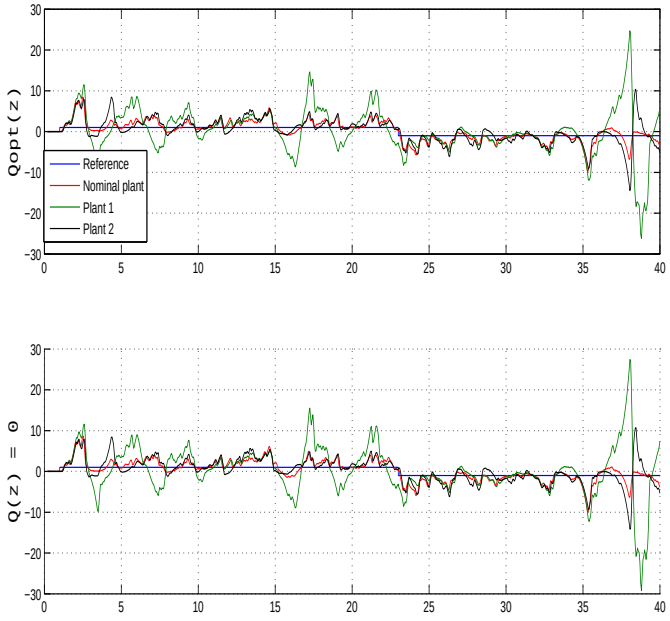

Figure 2.12: Simulation results with p = 0.4

2.6 Conclusions

The chapter has focused on control loops for SISO LTI plants, where the feedback path comprises a communication channel affected by Bernoulli data losses. This system has been studied as an equivalent one wherein the unreliable channel has been replaced by an additive i.i.d. noise channel, plus a gain. The objective of the chapter has been the synthesis of a controller that compensates model uncertainties and failed transmissions. To perform this task, an H_∞ control problem has been proposed. Numerical examples have illustrated closed-loop system performance benefits of our approach.

Future works could consider different structures for the parameter $Q(z)$, non-linear systems and also to include delays in the communication channel.

2.7 Related publications

- Isabel Jurado, Manuel G. Ortega, Daniel Quevedo and Francisco R. Rubio. *An H_∞ suboptimal robust control approach for Networked Control Systems with uncertainties and data dropouts*. Submitted to IET Control Theory Applications.

Chapter 3

Mixed H_2/H_∞ robust control approach for NCS with uncertainties and data dropouts

In this chapter, a Robust Networked Control System (RNCS) subject to data losses constraints is again considered. These data losses are modelled as an independent sequence of i.i.d. Bernoulli random variable. This random variable is replaced by an additive noise plus a gain, which is equal to the successful transmission probability in the feedback loop. Also, structural uncertainties in the model of the plant are considered.

To cope with this problem, a mixed H_2/H_∞ control technique is proposed in this chapter. In the previous chapter (Chapter 2), only the H_∞ technique is used while in this one the H_2 approach is added. In this way, the H_2 technique is used to stabilize the NCS taking into account the probability of data dropouts, while the H_∞ approach is in charge of making the closed-loop system robust enough against structural uncertainties of the nominal model.

3.1 Mixed H_2/H_∞ control problem

In this section, a brief mixed H_2/H_∞ control approach is described. Further information can be found in [K. Zhou, J. C. Doyle, and K. Glover, 1996] and [J. C. Doyle, K. Zhou, K. Glover and B. Bodenheimer, 1994]. The control system described in Figure 3.1 is con-

sidered, where the generalized plant $P(z)$ and the controller $C(z)$ are both assumed to be real-rational and proper. The signals involved in the diagram are the following: $w \in \mathbb{R}^{m_2}$ represents the disturbance vector, $u \in \mathbb{R}^{m_1}$ is the control input, $z_\infty \in \mathbb{R}^{p_1}$ and $z_2 \in \mathbb{R}^{p_3}$ are the error vectors, the first one for the measurement of the H_∞ performance, and the second one for the H_2 performance. The measurement supplied to the controller is represented by $m \in \mathbb{R}^{p_2}$.

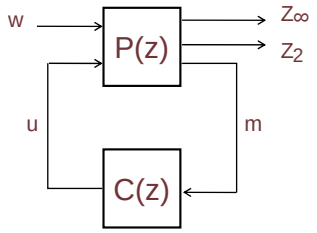

Figure 3.1: Mixed H_2/H_∞ synthesis

The synthesis problem considered in this approach consists in finding a suboptimal LTI controller $C(z)$ that minimizes the following mixed H_2/H_∞ criterion:

$$Min \quad \alpha \|T_\infty\|_\infty^2 + \beta \|T_2\|_2^2, \tag{3.1}$$

subject to:

- $\|T_\infty\|_\infty < \gamma_0$
- $\|T_2\|_2 < \nu_0$

where $T_\infty(z)$ and $T_2(z)$ denote the closed-loop transfer functions from w to z_∞ and z_2, respectively; and γ_0, $\nu_0 \in \mathbb{R}^+$.

As will be shown, the minimization of $\|T_2\|_2$ implies the minimization of the lower bound of the success probability in the data transmission.

In order to find out a controller by means of this control technique, it is necessary to put the original system into the form of the block diagram shown in Figure 3.1. To do this, the original system is changed with a *lower linear fractional transformation.*

In this case, T_∞ is chosen to represent a mixed-sensitivity H_∞ control problem, which is widely explained in [S. Skogestad, and I. Postlethwaite, 2005]. So two weighting functions are again chosen: $W_s(z)$ to weight the sensitivity function $S(z)$ and $W_t(z)$ to weight the complementary sensitivity function $T(z)$. These weighting functions allow to specify the range of frequencies of relevance for the corresponding closed-loop transfer matrix.

3.2 Problem definition

As Chapter 2, this one is focused on a RNCS wherein the main problems are the uncertainties in the model of the plant and the packets dropouts. So, the aim is to design a controller that stabilize a system subject to these two problems together.

The uncertainties under consideration were presented in Chapter 2, as well as the way to deal with the information losses, the *Mean square stability* definition and the *Equivalence* theorem.

In this chapter, condition (2.5) is imposed by solving an H_2 control problem, to find the minimal probability of success in the transmission (p). Therefore, by mixing the H_∞ technique from Chapter 2 and this one, a mixed H_2/H_∞ control problem is formulated, with the following cost function to minimize: $\alpha \|T_\infty\|_\infty^2 + \beta \|T_2\|_2^2$, where $\|T_\infty\|_\infty$ includes some weighting functions to achieve the system robustification and $\|T_2\|_2$ will be $\left\|T_p(z)\right\|_2$, to impose condition (2.5).

Problem 2 Consider the RNCS in Figure 2.2 where the plant $G(z)$ has bounded structural multiplicative uncertainties. Then, the problem consists in finding a robust controller $C(z)$, using the RNCS in Figure 2.3, that achieves the following conditions simultaneously:

- Minimize $\|T_\infty\|_\infty$ to achieve a good performance on tracking problems and the system robustification against the plant uncertainties.
- Minimize $\|T_2\|_2$ to calculate the minimal successful probability of data losses possible for the NCS, imposing condition (2.5), so the systems in the Figures 2.2 and 2.3 are equivalents.

3.3 Controller design

In this section the controller synthesis will be performed by means of the described mixed H_2/H_∞ control technique. Some weighting transfer functions will be introduced in the system to deal with the uncertainties of the plant model. The augmented system is represented in Figure 3.2. The weighting transfer functions $W_s(z)$ and $W_t(z)$ weight the sensitivity function ($S(z)$) and the complementary sensitivity function ($T(z)$), respectively. The outputs of these weighting transfer functions are the signals z_s and z_t respectively, and they represent the components of the vector z_∞ in Figure 3.1.

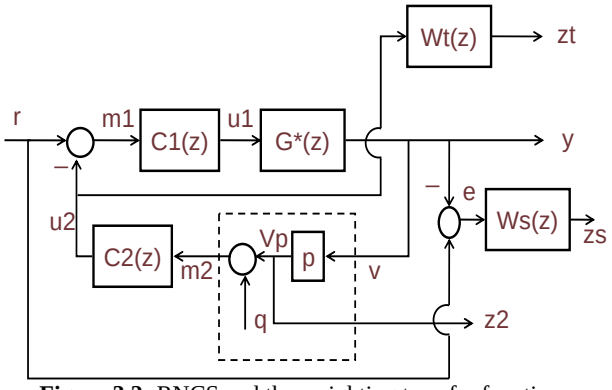

Figure 3.2: RNCS and the weighting transfer functions

It is important to note that the system under consideration, is a non-unitary feedback system. Thereby, in order to eliminate the steady state errors, a two-degrees-of-freedom controller is proposed. Therefore, the controller will be formed by two transfer functions,

$C_1(z)$ and $C_2(z)$. Also, the sensitivity function ($S(z)$) and the complementary sensitivity function ($T(z)$) expressions will change. These expressions will be:

$$S(z) = \frac{1 + C_1(z)G(z)\left(C_2(z)p - 1\right)}{1 + C_1(z)G(z)C_2(z)p}$$

$$T(z) = \frac{C_1(z)C_2(z)G(z)p}{1 + C_1(z)C_2(z)G(z)p}$$

The sensitivity function ($S(z)$) represents the transfer function from the reference to the error signal. The complementary sensitivity function ($T(z)$) depends on the open-loop transfer function of the system, which is: $L(z) = C_1(z)C_2(z)G(z)p$, so the control signal u_2 should be the input of the weighting transfer function $W_t(z)$, as it is represented in Figure 3.2.

The objectives of the controller are the following:

1. Minimize the H_∞ norm of the closed loop from the exogenous disturbances vector to the vector z_∞.

2. Minimize the H_2 norm of the closed loop signal from that vector to the signal z_2.

So, as mentioned before, the mixed H_2/H_∞ control problem will be solved to find a suboptimal controller which achieves a trade-off between the minimum of the two norms under consideration. To carry out the synthesis, the system in Figure 3.2 has to be expressed, by means of a *lower linear fractional transformation*, as in Figure 3.1. It is easy to see that, by identifying the terms, the followings equations hold:

$$z_\infty = [z_s \quad z_t]^T, \quad w = [r \quad q]^T,$$

$$P(z) = \left[\begin{array}{cc|cc} W_s(z) & 0 & -W_s(z)G(z) & 0 \\ 0 & 0 & 0 & W_{ks}(z) \\ \hline 0 & 0 & pG(z) & 0 \\ \hline I & 0 & 0 & -I \\ 0 & I & pG(z) & 0 \end{array}\right]$$

With respect to the minimization problem in (3.1), $T_\infty(z)$ and $T_2(z)$ are chosen as follows:

$$\|T_2(z)\|_2 = \left\|T_p(z)\right\|_2$$

$$\|T_\infty(z)\|_\infty = \left\| \left[\begin{array}{c} W_s(z)S(z) \\ W_t(z)T(z) \end{array}\right] \right\|_\infty$$

The parameters will be chosen in such a way that the condition (2.5) holds. This means that:

$$v_0 = \frac{p}{1-p}$$

At this point, it is worth mentioning some comments in relation to the choice of the others parameters. It is interesting to note that, if the priority is to achieve the minimal possible p, it is important to obtain a controller that provides an H_2 norm of $T_2(z)$ very close to its minimum. Then, for this case, the parameter β should be greater than α. On the contrary, if the interest lies on achieving the best performance and robustness against noises and uncertainties, it is better to choose the parameter α greater than β. This means that the resulting controller will provide a very small H_∞ norm of $T_\infty(z)$.

The probability of success in the transmission p is assumed to be fixed in the controller synthesis. This is possible if the network requirements are well-known. In any case, if the value of p changes, the stability of the closed-loop system is guaranteed if p is greater than the minimal probability of success in the transmission obtained.

3.4 Numerical results

To illustrate the methodology proposed in this paper, this section shows the obtained results when the control strategy is applied to a particular example. In this example the following unstable nominal plant will be considered:

$$G(z) = \frac{z-0.5}{z(z-1.1)}$$

The sampling time will be $t_m = 0.05s$.

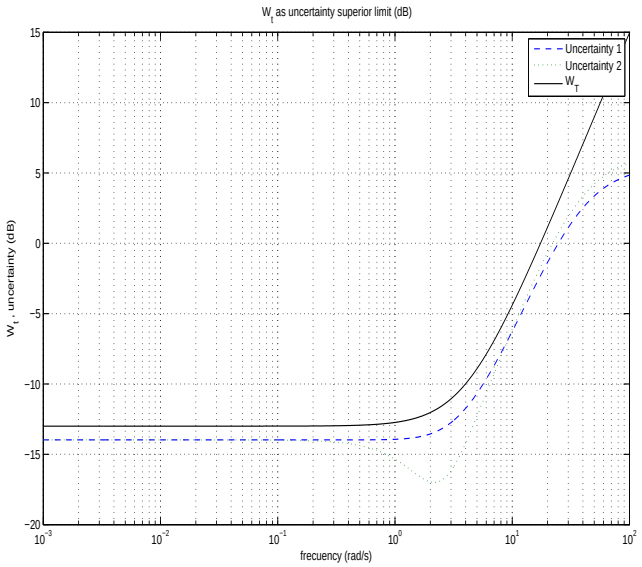

Figure 3.3: Uncertainties and W_t

To take into account the uncertainties in the plant, two non-nominal models have been also considered. To obtain these two other models, the real plant is supposed to have unmodelled dynamics, so, high frequency poles are include. Also a percentage of uncertainty in the model gain has been considered. From these two systems and the nominal plant, the multiplicative uncertainties can be computed. The frequency response of these uncertainties have been plotted in Figure 3.3.

From this estimation of the uncertainty, the weighting transfer function $W_t(z)$ for the complementary sensitivity function is designed in such way that its modulus must be greater than the modulus of the uncertainties for all frequency. The frequency response of $W_t(z)$ has been also represented in Figure 3.3.

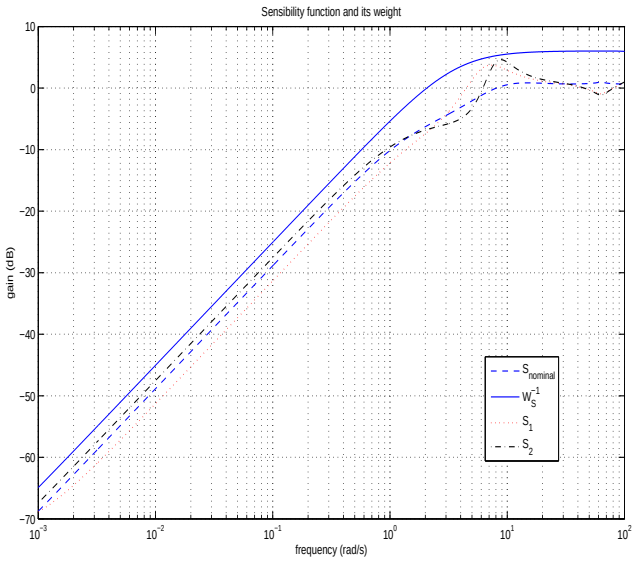

Figure 3.4: $S(z)$ of the nominal plant model and $W_s(z)$

By solving the mixed H_2/H_∞ control problem for this case using some functions of the $\mu-$*Analysis and Synthesis Toolbox* for Matlab and considering a success probability $p = 0.7$, a robust controller is obtained yielding the following results:

$$\|T_\infty\|_\infty = 0.8441, \quad \|T_2\|_2 = 1.3615$$

By imposing equation (2.5), this means that the system can afford a success probability p equal to or greater than 0.65, to guarantee MSS and to preserve the demanded robustness properties.

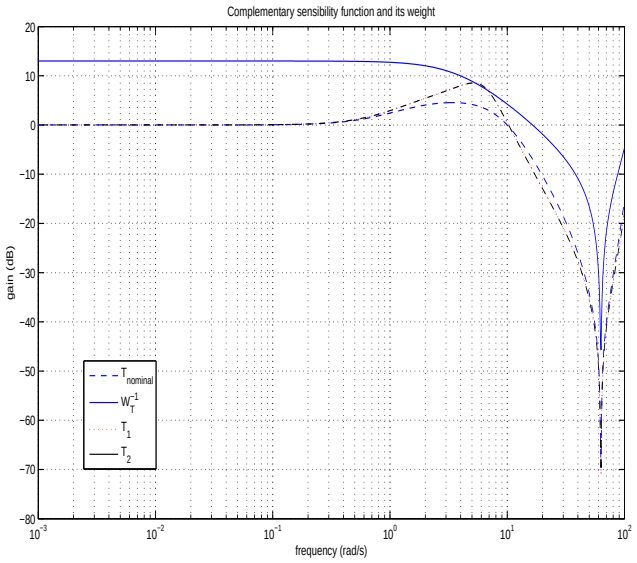

Figure 3.5: $T(z)$ of the nominal plant model and $W_t(z)$

In Figure 3.4 the sensitivity functions of the nominal and non-nominal plants models and the inverse of the weighting transfer function $W_s(z)$ are represented. This graphic shows how all the sensitivity functions, of the nominal system and systems with uncertainties, are below the inverse of the weighting function $W_s(z)$. This fact indicates that the output y can follow the reference r for all the plant models under consideration, that is, a tracking problem can be solved although the plant model is not exactly known.

Figure 3.5 represents the complementary sensitivity functions of the nominal and non-nominal plants models and the inverse of the weighting transfer function $W_t(z)$. From this graphic it is possible to see that all the complementary sensitivity functions, of the nominal system and systems with uncertainties, are below the inverse of the weighting function $W_t(z)$, so the obtained controller is robust against the uncertainties in the plant model.

To corroborate these results, some simulations have been carried out with the proposed example. Figure 3.6 shows how the system follows the reference with a successful trans-

mission probability $p = 0.7$, which is greater than the minimal p that can provide MSS and robustness properties for this system. This graphic represents the outputs of the closed-loop system with the nominal plant, with the plant with the uncertainties 1 and with the plant with the uncertainties 2. The results are very similar because the robustness of the system. However, there exist some differences between the differents outputs. For example, the output with the uncertainties 1 has an overshoot that is greater than the overshoot when the nominal model is used. With respect to the output with the uncertainties 2, the overshoot is reduced with respect the other cases, but the stationary performance is worse.

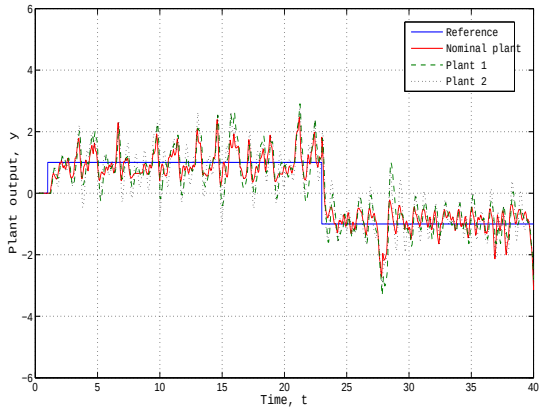

Figure 3.6: Simulation results with p = 0.7

The outputs of the different systems for a value of $p = 0.9$ are shown in Figure 3.7. In this case, the probability of success in the transmission has been increased, although the controller used in these simulations is the one calculated for $p = 0.7$. Obviously, the results are better than in the ones presented in Figure 3.6, but the differences between the performance with the different systems is the same as in the case of $p = 0.7$. Also, there are steady state errors because the controller is the calculated for $p = 0.7$ so the feedback is non-unitary. These steady state errors might be avoid by calculating the controller using $p = 0.9$, but the objective is to compare the results with the same controller, supposing that

p have changed in the network.

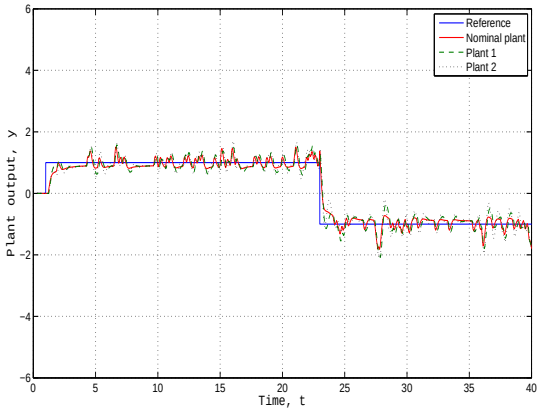

Figure 3.7: Simulation results with p = 0.90

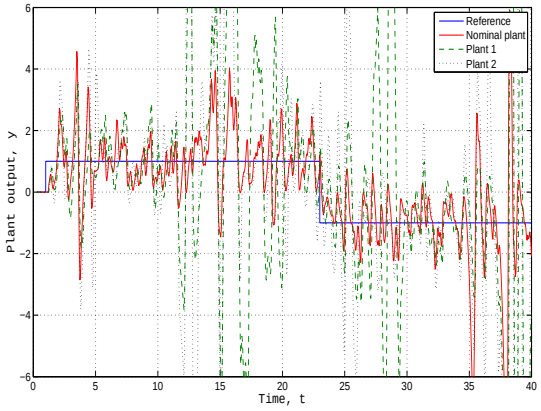

Figure 3.8: Simulation results with p = 0.40

Finally, Figure 3.8 presents the outputs of all the systems imposing $p = 0.4$, while using the same controller as in the precedings simulations. Obviously, the performances get worse for all the systems, and in the case of the plant with the uncertainties 1 and 2, the closed-loop system becomes unstable. Therefore, with $p = 0.4$, the robust stability is lost.

3.5 Application to glucose robust control in diabetes with sensor failures

In this section, the glucose control for diabetic patients is considered. The problem of possible sensor failures, that leads into an absence of glucose measurement, is treated. This sensor errors are taking into account by means of modelling its failures as an independent sequence of random variable. This random variable is replaced by an additive noise plus a gain, which is equal to the successful transmission probability in the feedback loop. Also, to take into account different patients, structural uncertainties in the model of the plant are considered.

To solve this problem, the mixed H_2/H_∞ robust control technique presented above is used. In this way, the H_2 approach is used to stabilize system taking into account the probability of sensor errors, while the H_∞ approach is in charge of making the closed-loop system robust enough against structural uncertainties of the nominal model.

3.5.1 Introduction to glucose problem in diabetes

Insulin is the most important factor for the digestion process, in where the food is decomposed to create glucose, the principal source of energy for the body. This glucose passes to the blood, where the cells absorb it thanks to the insulin. As the glucose concentration raises, the insulin secretion is stimulated from the pancreas. This makes the insulin level in blood increases, inducing the glucose absortion into the cells. In a diabetic person, the insulin deficiency causes the glucose concentration in blood, so the body is deprived of its main energy source. Furthermore, high glucose levels in blood may damage the blood vessels, the kidneys and the nerves. Since there is no cure for the diabetes yet, the people affected by this metabolic disease have to control the glucose levels in blood, keeping them close enough to the normal ones by the external insulin supply. An appropiate control can help preventing diseases related with heart and circulatory system, eyes, kidneys and

nerves.

Some biological mathematical models provide characteristics of the glucose regulation behavior in the human body, such the *Sorensen Model*, described in [Kovacs and Kulcsár, 2007], and the one used in this work, the *Bergman Model*, described in [R. N. Bergman and Cobelli, 1981],[R. N. Bergman and Ader, 1985],[Lynch and Bequette, 2001].

Diabetes is a metabolism disease that is characterised by the increase of the glucose levels in blood (hyperglycemia), caused by the insulin secretion deficiency. Normal limits for the glucose levels in blood are around 130 mg/dl, even when some greasy and sugar food have been ingested. This gyclemia stability is achieved by means of a regulator mecanism extraordinary exact and sensitive.

When a non-diabetic person ingests food, the sugars in it are absorbed and they pass to the blood, tending the glucose levels to raise. This tendency is detected by the insulin producer cells, which respond with a quick insulin secretion. This makes the cells to absorb the glucose, that way its levels in blood decrease. For a diabetic person, the insulin production is so low that alters all the regulator mecanism: the glucose rise in blood is not followed by the sufficient increase of the insulin, thereby the glucose cannot be absorbed by the cells and its level keeps increasing. As a consequence, the cells cannot produce enough energy and their functions are altered.

In this work, type I diabetes is considered. In this kind of diabetes, the pancreas does not produce insulin, and the patient is totally dependent on insulin from an external source to be infused at a rate to maintain blood sugar levels at normal ones (72-145 mg/dl).

Since the normal body has a natural feedback regulation system for the insulin production, the goal of this paper is to regulate blood sugar level in a type I diabetic by controlling the insulin infusion rate.

This kind of problem has been treated by means of Model Predictive Control (MPC) ([Lynch and Bequette, 2001, 2002, P. Dua and Pistikopoulos, 2005, 2006]), and also with

Robust Control approaches ([S. Kamath, 2006, R. S. Parker and Peppas, 2000]).

In this work, it will be considered the possible sensor errors, that leads into an absence of glucose measurement. Some of the most common sensors are: implantable sensors (MiniMed), sensors based on blood samples (Accu-Chek) or non-invasive glucometer (GlucoWatch). Usually, the sensors samples are taking every 5-10 min. Therefore, it is important to have proper sensors that measure the glucose precisely. However, sometimes the sensors are not totally reliable.

Several causes of failures have been reported for glucose sensors [Zilic and Radecka, 2011]. One cause can be because the body rejects them during the early phase or after longer use. On the other hand, their sensitivity might degrade. Also, the fluid that flows to the sensors could be stopped. It was also observed that for some data points, the readings might be occasionally off, i.e., they drop-out [B. Kovatchev and Clarke, 2008]. Finally, sensors can completely fail.

To deal with that trouble, the robust controller introduced in this chapter is used for controlling the glucose levels. The controller synthesis will take into account the model uncertainties, because the resulting controller must to be accurate for different patients. Also it will be considered the previously described sensor errors.

3.5.2 System description

The model used to define the gluco-regulatory system is the *Bergman Model* [R. N. Bergman and Cobelli, 1981]. In the first place the model formed by the differential non-linear equations is presented. Then, that model is symplified by means its linearization around an equilibrium point.

Non-lineal model definition

The Bergman Model, used in this work, is described by means the following differential non-linear equation described in [R. N. Bergman and Cobelli, 1981]:

$$\frac{dG}{dt} = -P_1 G - X(G + G_b) + D(t) \tag{3.2}$$

$$\frac{dI}{dt} = -n(I + I_b) + \frac{U(t)}{V_1} \tag{3.3}$$

$$\frac{dX}{dt} = -P_2 X + P_3 I \tag{3.4}$$

$$D(t) = \frac{D_G A_G t e^{-t/t_{max,I}}}{V_G t_{max,G}^2} \tag{3.5}$$

where:

- $G(t)$ is the plasma glucose concentration (mmol/L) above basal value.
- $I(t)$ is the plasma insulin concentration (mU/L) above basal value.
- $X(t)$ is proportional to $I(t)$ in remote compartment (mU/L).
- $D(t)$ is the meal glucose disturbance (mmol/Lmin).
- $U(t)$ is the manipulated insulin infusion rate (mU/min).
- G_b and I_b are the basal values of glucose and insulin concentration (mmol/L and mU/L).
- D_G is the carbohidrate ingestion (g CHO).

The model parameters for a tipical patient are:

- $P_1 = 0.028\ min^{-1}$
- $P_2 = 0.025\ min^{-1}$
- $P_3 = 0.000013\ mU/L$
- $V_1 = 12$ L and $n = 5/54\ min$
- $A_G = 0.8$

- $t_{max,G} = 40\ min$
- $t_{max,I} = 55\ min$
- $V_G = 13.79$ L
- $I_b = 15$ mU/L
- $G_b = 4.5$ mmol/L

Linear model

In order to carry out the controller synthesis, a linear model is going to be obtained from the differential non-linear equations. Therefore, from the linearization of the model around the equilibrium point $X_0 = G_0 = 0$, the linear model obtained is the following:

$$\begin{bmatrix} \dot{G} \\ \dot{I} \\ \dot{X} \end{bmatrix} = \begin{bmatrix} -P_1 & 0 & -G_b \\ 0 & -n & 0 \\ 0 & P_3 & -P_2 \end{bmatrix} \begin{bmatrix} G \\ I \\ X \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 1/V_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{3.6}$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} G \\ I \\ X \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{3.7}$$

where

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} U - U_b \\ D - 0 \end{bmatrix} \tag{3.8}$$

Where $U_b = 16.66667$ mU/min is the insulin necessary to keep the equilibrium around X_0, G_0.

3.5.3 Problem definition

Once the gluco-regulatory model is identified, it is possible to apply different control structures to control the glucose level in blood.

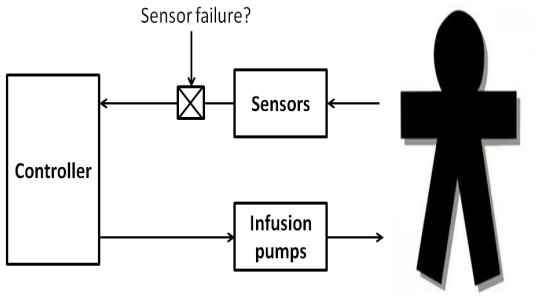

Figure 3.9: Control scheme with sensor failures

Therefore, the plant under consiteration will be a patient, and the model will be given by the Bergman Model equations. A sensor will be measuring the glucose level in blood. There will be also an insulin pump as actuator, injecting insulin intravenously in order to control the glucose level in blood (see Figure 3.9). The designed controller will be in charged of providing the quantity of insulin needed by the patient.

There are two main problems considered in this work:

- The uncertainties in the plant model, because the parameters change from one patient to another.
- The sensor failures. Each sampling time, the sensor has a probability of failing its measurement.

The aim of this work is to control the glucose levels by means of the insulin injections, taking into account the explained problems.

The uncertainties are modelled as structural ones, and they are represented by $G^*(z)$.

The information errors occur due to some sensor faliure. That means that the glucose measurement is not available at some instants. This situation is illustrated in Figure 3.10, where the different patientes are represented by $G^*(z)$, $C(z)$ is the controller, r is the reference that provides the proper glucose levels, y is the glucose level in blood, u is the insulin injection that will control the glucose in blood, and d_r models the sensor failures, that is

$d_r \in \{0,1\}$.

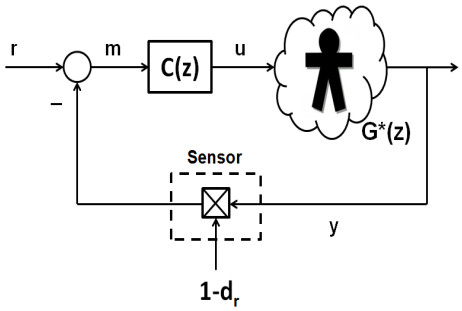

Figure 3.10: Robust controlled system with feedback failures

3.5.4 Simulation results

In this section, the simulation results obtained from the robust control techniques application are presented and analyzed.

The presented simulations take into account the following considerations:

1. Basal equilibrium point: the system is going to be stabilized around the equilibrium point given by the glucose basal level, that is, 81 mg/dl (4.5 mmol/L).

2. Glucose levels in blood: the used limits for a diabetic patient are given by 60 and 180 mg/dl.

3. Controller saturation: the applied control action has to be inside certain boundaries, defined by (3.9). This is the real situation when the insulin injections are dosed by means an external actuator (insulin pump).

$$0 \quad (mU/min) \leq U + U_{basal} \leq 100 \quad (mU/min) \tag{3.9}$$

It is possible to see how, with these boundaries, the control action (injected insulin), is never negative or bigger than 100 mU/min.

4. Sampling time: because of the real limitation of the existing glucose sensors, $t_m = 5$ min.

5. Meals: the external disturbances added are the ingestion of meals at different times during the day. They are defined by equation (3.5) with the parameters presented in Section 3.5.2.

   Four daily meals are considered ingested in the following timetable and quantities:

   - $08:00 \rightarrow 55.1 \quad g \quad CHO$
   - $13:00 \rightarrow 87.9 \quad g \quad CHO$
   - $18:00 \rightarrow 69.0 \quad g \quad CHO$
   - $22:00 \rightarrow 45.3 \quad g \quad CHO$

   Taking into account a conversion factor 1000/180 g CHO, the disturbances graphic (daily meals) is represented in Figure 3.11.

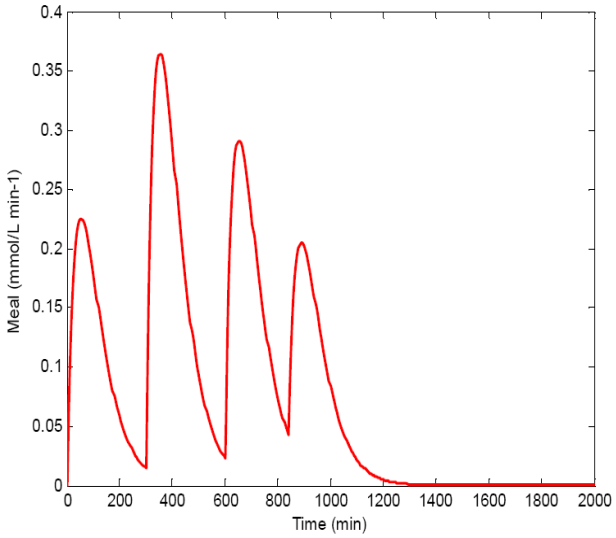

Figure 3.11: Daily meals ingestion

The designed H_2/H_∞ robust controller provides the following attenuation values:

$$\gamma_0 = 0.589$$

$$\nu_0 = 0.136$$

This controller has been obtained assuming a sensor measurement succesful probability of 70% ($p = 0.7$), that way equation (2.5) holds.

Simulations for different patients

Now, different patient models are considered for the simulations, which are defined by the parameter variations P_1, P_2 and P_3, which identify them.

The selected patients' parameters for this work are the following, which vary depending on their diabetic condition:

- Nominal patient (GN):

  P_1 = 0.028 min^{-1}

  P_2 = 0.025 min^{-1}

  P_3 = 1.3 e^{-5} mU/L

- Patient 1 ($G1$):

  P_1 = 0.026 min^{-1}

  P_2 = 0.024 min^{-1}

  P_3 = 1.1 e^{-5} mU/L

- Patient 2 ($G2$):

  P_1 = 0.030 min^{-1}

  P_2 = 0.027 min^{-1}

  P_3 = 1.4 e^{-5} mU/L

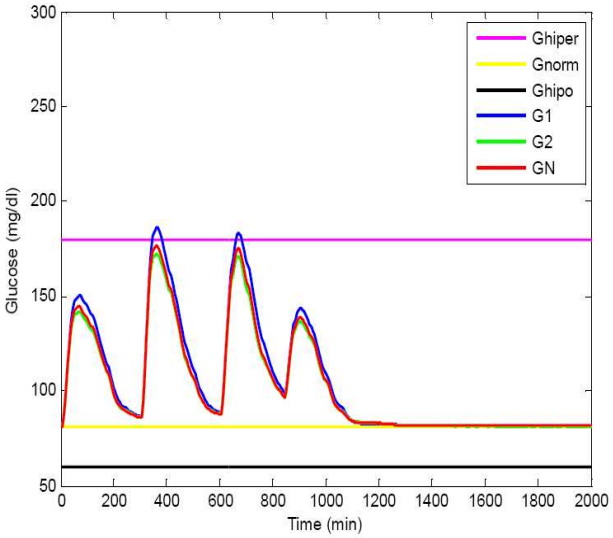

Figure 3.12: Glucose trajectory for different patients and $p = 0.9$

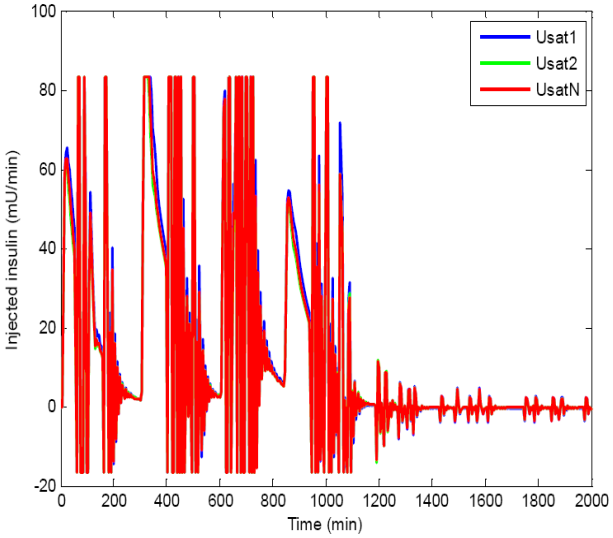

Figure 3.13: Injected insulin for different patients and $p = 0.9$

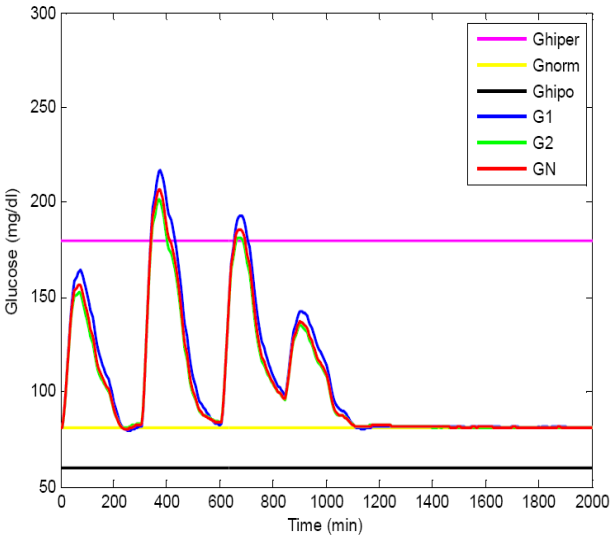

Figure 3.14: Glucose trajectory for different patients and $p = 0.7$

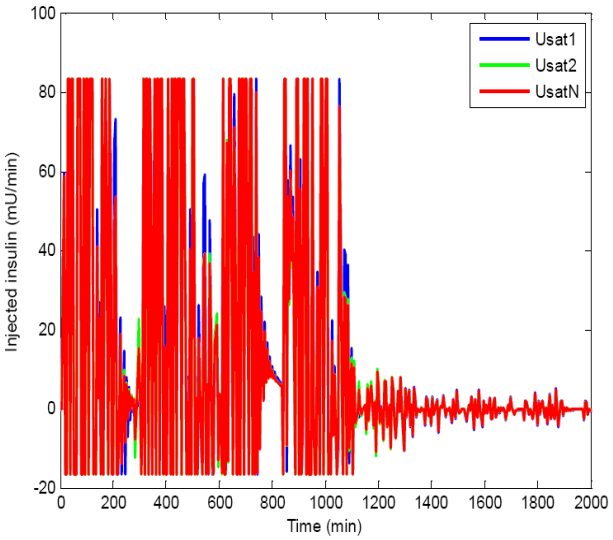

Figure 3.15: Injected insulin for different patients and $p = 0.7$

The robustness of the system can be appreciated for the different patient models, since the obtained glucose evolution is quite similar for all of them, for the same probability p.

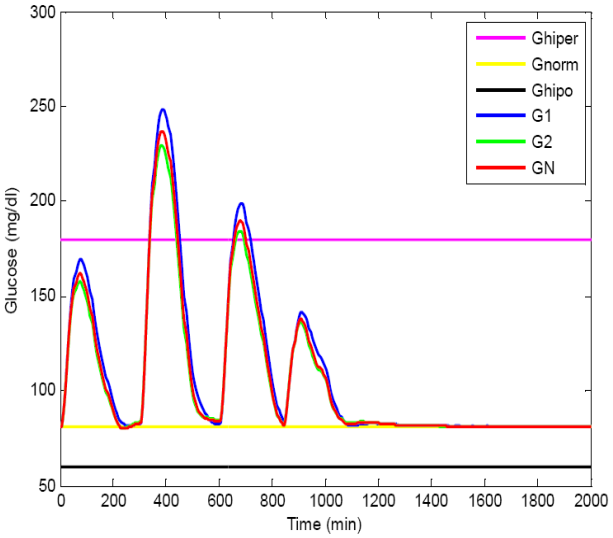

Figure 3.16: Glucose trajectory for different patients and $p = 0.5$

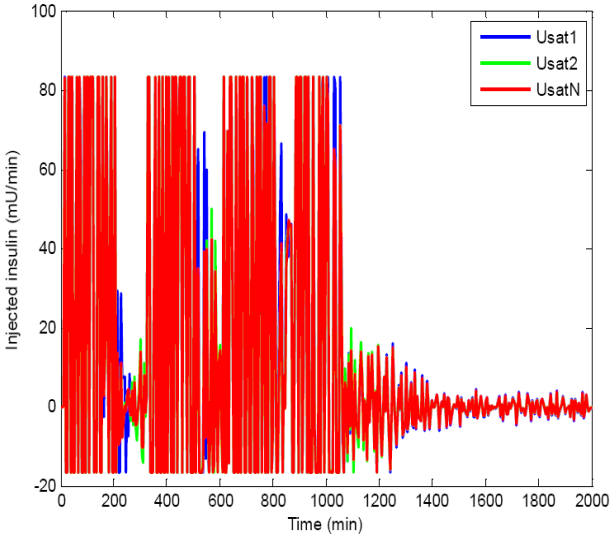

Figure 3.17: Injected insulin for different patients and $p = 0.5$

It can be seen in Figure 3.12 how the glucose is always less than 200 mg/dl for the probability $p = 0.9$, and only patient 1 enters in hyperglycemia.

Figure 3.14 shows how the performance gets worse for $p = 0.7$. Now, the glucose levels exceed 200 mg/dl, being all the patients at hyperglycemia levels during some minutes.

With $p = 0.5$, Figure 3.16, the maximum glucose level increases, being all the patients more time in hyperglycemia.

The glucose level never enters in the hypoglycemia zone, which can cause the death of the patient.

Figures 3.13, 3.15 and 3.17 show the injected insulin, which gets more violent when p decreases. Also, the times when it reaches the saturation levels increase with smaller p.

3.6 Conclusions

This chapter has focused on a NCS subject to data dropouts constraints. In particular, control loops for SISO LTI plants, where the feedback path comprises a communication channel that produces data losses, are considered. This system has been studied as an equivalent one wherein the unreliable channel has been replaced by an additive i.i.d. noise channel, plus a gain.

The objective of this chapter has been the synthesis of a controller that avoid the model uncertainties and support the failed transmissions. Also, the lower bound of the success probability in the transmission has been found. To perform this task, a mixed H_2/H_∞ control problem has been proposed. To obtain a robust controller, some functions have been chosen to weight some sensitivity functions. Moreover, from this control problem, the minimal successful transmission probability is obtained such that MSS and robustness properties for the closed-loop system are guaranteed.

Finally, an application has been exposed to obtain some numerical results that illustrates the closed-loop system performance. These simulation results corroborated that ro-

bust performance is achieved if the successful probability transmission is higher than the minimum computed, while the differents systems performances get worse, until the robust stability is lost, as the successful probability transmission decreases.

Also, an application of this technique to the problem of the glucose control for diabetic patients subject to sensor errors constraints has been presented.

Different patients have been considered, thereby the synthesis of the controller has been such that avoid the model uncertainties and support the sensor failures. Also, the lower bound of the success probability in the sensors has been found.

Some simulations have been carried out for different patients to illustrate the closed-loop system performance. These simulation results corroborated that robust performance is achieved if the successful sensor measurement probability is higher than the minimum computed.

3.7 Related publications

- Isabel Jurado, Manuel G. Ortega and Francisco R. Rubio. *Networked Mixed H_2/H_∞ robust control approach for NCS with uncertainties and data dropouts*. Proceedings of the 18th World Congress of the International Federation of Automatic Control (IFAC), 2011.

Chapter 4

An H_∞ Filter and Controller Design for Networked Control of Markovian Systems with Uncertainties and Data Dropouts

This chapter considers a Robust Networked Control System (RNCS) subject to data losses constraints and modelled as a Markovian Jump Linear System. A filter and a controller will be designed together by means of H_∞ techniques. This design will provide robustness to the closed-loop system against plant uncertainties, as well as disturbances attenuation.

The network will introduce data losses which will be modelled as a sequence of independent and identically distributed (i.i.d.) Bernoulli random variable. It is considered a maximum number of consecutive packet dropouts. Moreover, uncertainties in the model of the plant are included, as well as unknown disturbances.

To cope with this problem, a robust H_∞ filter and controller are designed by developing a Linear Matrix Inequality (LMI).

There are also structural uncertainties in the plant and unknown disturbances. The system is modelled as a Markovian Jump Linear System (MJLS) and an LMI is derived in order to find a robust filter and controller by means of H_∞ techniques (see [S. Skogestad,

and I. Postlethwaite, 2005]). The designed filter will calculate an estimation of the state of the plant. This estimation will be used when a packet dropout occurs, so the feedback will not become zero.

In the first section of this chapter, a filter and a controller will be synthesized in order to make the closed-loop system robust against structural uncertainties of the nominal model. The filter will be also able to deal with network data losses. This will be done by means of frecuency techniques.

4.1 Filter design using frecuency techniques

This result considers an NCS with data dropouts source as well as structural uncertainties in the plant. Therefore, one goal of this section is to find a robust controller for the plant with uncertainties, which will be carried out by means of an H_∞ control approach. Another important objective is to design a filter that calculates an estimation of the output of the plant. This estimation will be used when a packet dropout occurs, so the feedback will not become zero. Mean square stability (MSS) and robustness properties also have to be guaranteed. The filter design will be carried out with a technique based on the location of the unstable poles of the plant model. Further information can be found in [J. E. Normey-Rico and E. F. Camacho, 2009].

To illustrate the situation, a control structure is presented that depends on whether there is a packet dropout or not. This structure is represented in Figure 4.1. It can be seen that the feedback gets lost when a packet dropout occurs. Therefore, one of the main goal of this section is to design a filter to estimate the plant output when a packet dropout is detected, so the feedback is not zero when a data is lost. Figure 4.1 shows the system including the output estimation.

As mentioned before, structural uncertainties will be considered in the model of the plant $G^*(z)$. So, the controller $C(z)$ will be designed as a mixed sensitivity H_∞ controller,

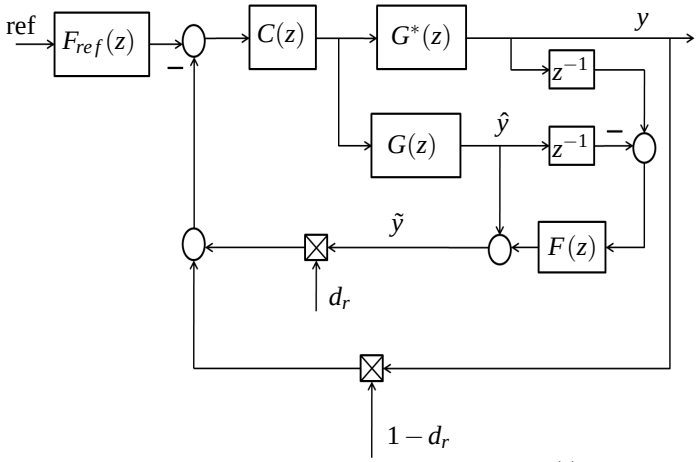

Figure 4.1: RNCS structure with the filter $F(z)$.

using a similar procedure as in Chapter 2. This approach makes possible to impose robust performance by means of appropriate design of weighting functions. In particular, it is well known that robust stability can be imposed by weighting the complementary sensitivity function if structural multiplicative uncertainty is considered ([M. G. Ortega and F. R. Rubio, 2004], [M. G. Ortega, M. Vargas, L. F. Castaño and F. R. Rubio, 2006]), while performance can be imposed by means of a reasonable weight on the sensitivity function. Using this approach, the H_∞ controller will be obtained.

On the other hand, the filter has to be able to deal with the uncertainties of the plant in order to give an appropriate estimation of the plant output. Therefore, it is necessary to obtain the conditions for the robustness of the closed-loop system. These conditions are given by the stability of the the characteristic equation for $G^*(z)$, yielding [J. E. Normey-Rico and E. F. Camacho, 2009]:

$$1+C(z)G(z)+F(z)z^{-1}C(z)G(z)W_m(z)\Delta(z)=0$$

Under the assumption of the nominal closed-loop system is stable, the robust stability con-

dition for the filter is given by the following expression:

$$\left|W_m(e^{jp})\right| < \frac{\left|1+C(e^{jp})G(e^{jp})\right|}{|F(e^{jp})e^{-jp}C(e^{jp})G(e^{jp})|} \quad \forall\, 0 < p < \frac{\pi}{2}$$

where p represents the non dimensional frequency, i.e., $p = \omega t_m$, being t_m the sampling time.

4.1.1 Filter design

The proposed structure for the design is shown in Figure 4.1. It can be seen that the structure is very similar to the Smith predictor with two additional filters. $F_{ref}(z)$ is a traditional reference filter to improve the set-point response and $F(z)$ is a predictor filter used to estimate the system output when a packet dropout occurs. $G(z)$ is the nominal model of the uncertain plant with uncertainties $G^*(z)$.

In the proposed structure, the estimation of the output, $\tilde{y}$, is only used when a packet is lost in the feedback channel, i.e., $d_r = 1$. Otherwise, the measured output is provided to the controller.

This structure has been particularized for the case in which the number of consecutive packet dropouts cannot more than one. Thus, the estimate output $\tilde{y}$ at the sampling time n is given by the followings equations:

$$\tilde{y}(n) = \hat{y}(n) + F[y(n-1) - \hat{y}(n-1)]$$

where $F[\tilde{y}(n-1) - \hat{y}(n-1)]$ is the correction factor of the estimation.

The filter, $F(z)$, design is done by means of frecuency techniques, taking into account the unstable dynamics of the plant. Figure 4.2 shows an equivalent part of the original system which is used in the filter design procedure. Making some blocks operations it is straightforward to see that $S_F(z) = G(z)(1 - z^{-1}F(z))$ and it is needed to be stable. Filter $F(z)$ is calculated in such a way that $(1 - z^{-1}F(z))$ cancels the unstable poles of $G(z)$ in

the expression of S(z). Further explanations can be found in [J. E. Normey-Rico and E. F. Camacho, 2009].

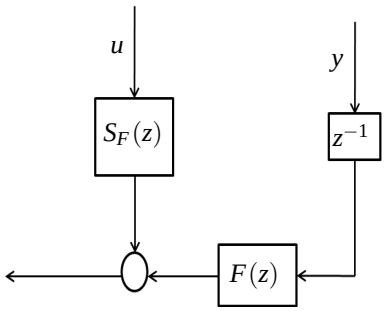

Figure 4.2: Structure for the filter design.

4.1.2 Numerical results

In this section some simulation results are presented for an unstable uncertain system. The nominal model of the plant is represented by the following transfer function, which is the same of that in the previous chapter, in section 2.5:

$$G(z) = \frac{z-0.5}{z(z-1.1)}$$

The sampling time is $t_m = 0.05s$ and the probability of success in the transmission is equal to $p = 0.7$.

The frequency response of these uncertainties have been depicted in Figure 4.3.

From this estimation of the uncertainty bound, the weighting transfer function $W_t(z)$ for the complementary sensitivity function is designed in such way that its module is greater than the modulus of the uncertainties for all frequencies. The frequency response of $W_t(z)$ has been also represented in Figure 4.3.

By solving the H_∞ control problem for this case using some functions of the $\mu-$*Analysis and Synthesis Toolbox* for Matlab, a robust controller is obtained yielding the following re-

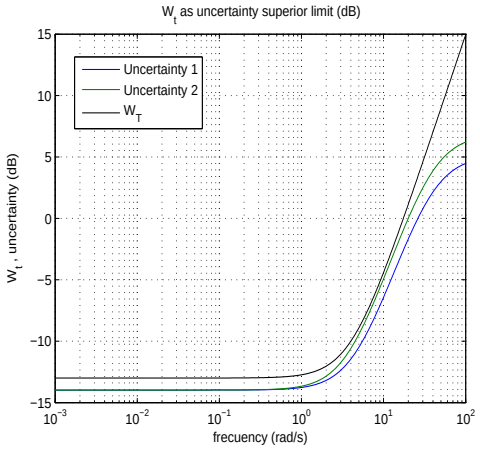

Figure 4.3: Uncertainties and W_t

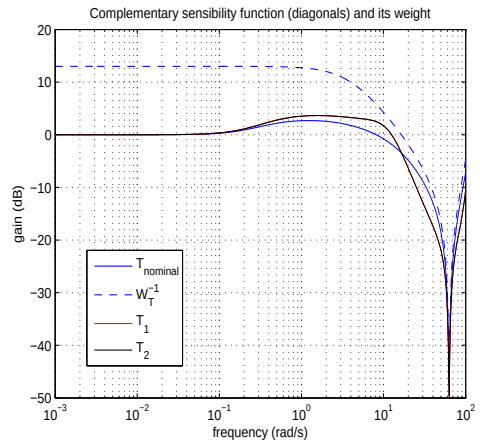

Figure 4.4: $T(z)$ of the nominal plant model and $W_t(z)$

sults:

$$\|T_\infty\|_\infty = 0.3243$$

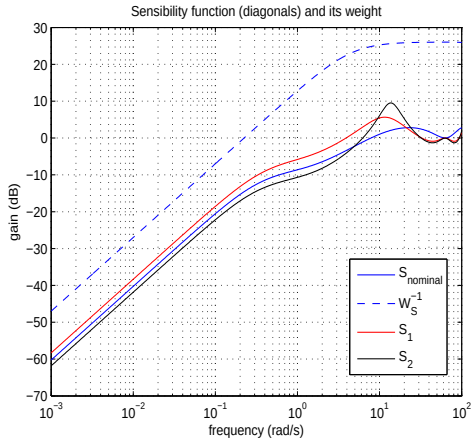

Figure 4.5: $S(z)$ of the nominal plant model and $W_S(z)$

It is important to note that the control problem here is not the same than the one in Chapter 2. In contrast to Chapter 2, here the *Equivalence* theorem is not used, so the gain p associated with communication success is not neccessary in the scheme. Another difference is that this chapter is not trying to minimize the error variance, while Chapter 2 was.

In Figure 4.5 modulus of the frequency response of the sensitivity functions of the nominal and non-nominal plants models and the inverse of the weighting transfer function $W_s(z)$ are represented. This graphic shows how all the sensitivity functions, of the nominal system and systems with uncertainties, are below the inverse of the weighting function $W_s(z)$. This fact indicates that the output y can follow the reference r for all the plant models under consideration, that is, a tracking problem can be solved although the plant model is not exactly known.

Figure 4.4 represents the complementary sensitivity functions of the nominal and non-nominal plants models and the inverse of the weighting transfer function $W_t(z)$. From this graphic it is possible to see that all the complementary sensitivity functions, of the nominal system and systems with uncertainties, are below the inverse of the weighting function $W_t(z)$, so the obtained controller is robust against the uncertainties in the plant model.

The expressions of $W_t(z)$ and $W_s(z)$ are definied by the following equations:

$$W_t(z) = \frac{2.438z - 1.995}{z + 0.9802}$$

$$W_s(z) = \frac{0.05558z - 0.04442}{z - 1}$$

As it was explained before, the filter $F(z)$ is chosen in such a way that the transfer function $S_F(z) = G(z)(1 - z^{-1}F(z))$ is stable. So the filter $F(z)$ is chosen in the following manner:

$$F(z) = \frac{0.8971z - 0.8195}{z^2 - 1.355z + 0.4326}$$

In Figure 4.6 some simulation results are presented. As can be seen, all the systems achieve stability. Logically, the best results are obtained with the nominal system. In order to reduce the overshoot, the reference filter $F_{ref}(z)$ could be used. Simulations in Figure 4.6 show the results with $F_{ref}(z) = 1$.

In Figure 4.7 the control signals for the different uncertainties and for the nominal systems are shown. It can be seen that the signal oscillates when the considered system is affected by uncertainties. When the system is the nominal model of the plant, the signal does not oscillate and goes to the reference without steady state errors. This is because the feedback changes when there is a data loss. This occurs when the plant has uncertainties, so the output estimated and the real output are different.

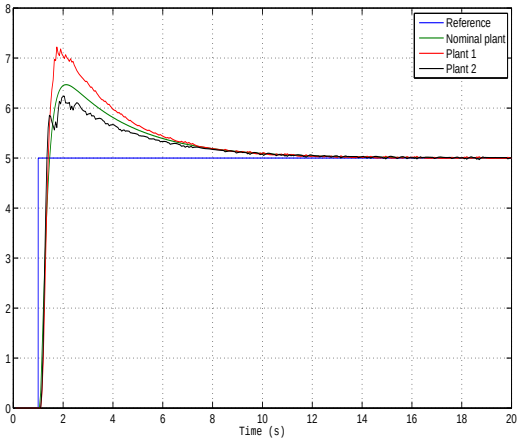

Figure 4.6: Simulations results.

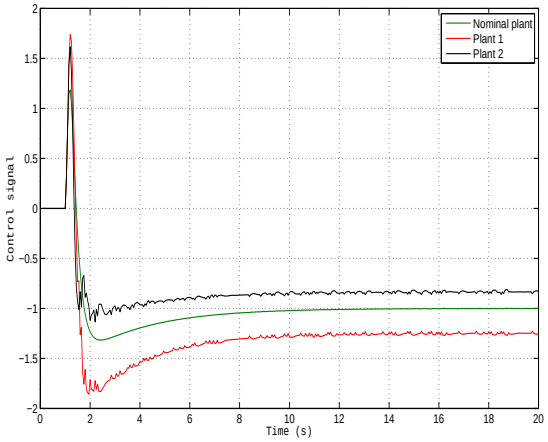

Figure 4.7: Control signals.

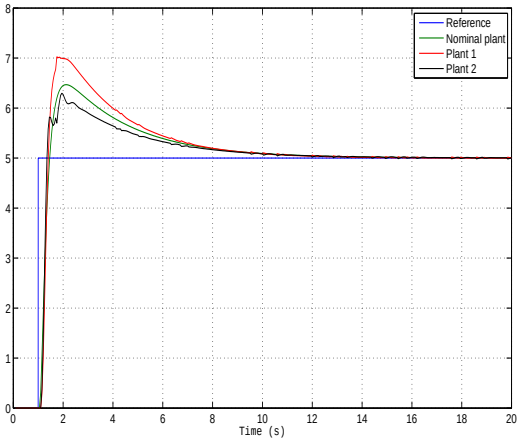

Figure 4.8: Simulations results with $p = 0.9$.

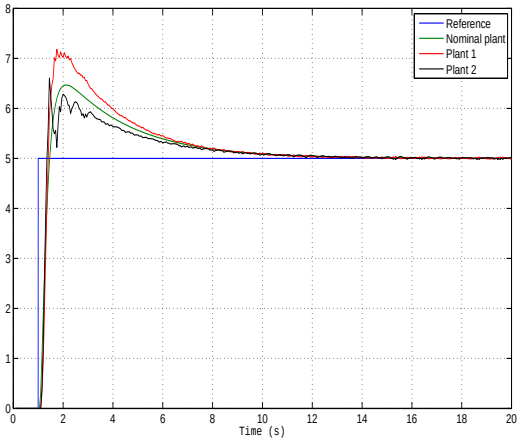

Figure 4.9: Simulations results with $p = 0.5$.

Table 4.1: Error variances

	Plant		
	Nominal plant	Plant 1	Plant 2
$p = 0.7$	0.1523	0.3689	0.1931
$p = 0.9$	0.1523	0.3566	0.1923
$p = 0.5$	0.1523	0.3706	0.1984

Another simulations results have been obtained when varying the probability of successful in the transmission. In Figure 4.8, simulations results when this value is set to $p = 0.9$ are shown; and it can be seen how the amplitude of the oscillations decreases. This is because the feedback doesn't change so often between the real output and the estimated output than in the case of $p = 0.7$.

The opposite case is shown in Figure 4.9. The oscillations of the signal increase because the feedback commute more often than before.

Logically, in the cases when it is considered that the plant does not have uncertainties the signals are the same because there are no differences between the real output and the filter estimation of the output.

To compare the results, the variances of the error signals are shown in the table 7.1. As it was said before, the results are equal in the case of the nominal plant. The variances get worse as the probability of success in the transmission decreases. This happens because as the probability decreases the estimation of the output is used more and more, and the estimation is not accurate when the plant is affected by uncertainties.

4.2 Problem statement for the Markovian Jump Control System

The aim of this part of the chapter is the design of a filter and a controller that ensure the stability of the uncertain system subject to packets dropouts. To deal with this problem, the system will be represented as a state-state model, and it will be solved by means of Linear

Matrix Inequalities (LMI). That way, in contrast to the previous section, it is possible to have a control structure that takes into account the possibility of a random number of dropouts.

In this part of the chapter, state-space techniques are used, whereas in the previous part frequency techniques were used, which are more useful to analyze the stability and synthesized the filter and controller.

In particular, the system structure is posed as a linear Markov system with $N+1$ modes, where N is the maximum number of consecutive dropouts. Each mode represents a structure of the system depending on the number of consecutive packet dropouts. The controller and the filter will be time-varying depending on the system mode.

The Markov process under consideration is: $\{\theta_k, k \geq 0\}$, with state space $\mathscr{S} = \{1, ..., N+1\}$ and state transition matrix $P = \{p_{ij}\}_{i,j \in \mathscr{S}}$, i.e., the transition probabilities are:

$$P\{\theta_{k+1} = j | \theta_k = i\} = p_{ij}, \quad \forall i, j \in \mathscr{S}$$

with $p_{ij} \geq 0$, $\forall i, j \in \mathscr{S}$ and $\sum_{j=1}^{N+1} p_{ij} = 1$, $\forall i \in \mathscr{S}$.

This kind of systems satisfies the property given by the following definition:

Definition: **Markovian property.** (See [O. L. V. Costa, M. D. Fragoso, and R. P. Marques, 2005]) A stochastic process has the Markov property if the conditional probability distribution of future states of the process depends only upon the present state, not on the sequence of events that preceded it, that is:

$$\mathbf{Pr}\{x(k)|x(k-1), x(k-2), ..., x(0)\} = \mathbf{Pr}\{x(k)|x(k-1)\} \tag{4.1}$$

The system dynamics is described by means of the following equations:

$$\Sigma^*(\theta_k) := \begin{cases} x(k+1) = A(\theta_k, k)x(k) + B(\theta_k, k)u(k) + B_w(\theta_k)w(k) \\ z(k) = C(\theta_k, k)x(k) + D(\theta_k, k)u(k) + D_w(\theta_k)w(k), \end{cases} \tag{4.2}$$

where $x(k) \in \mathbb{R}^n$ is the state of the system, $u(k) \in \mathbb{R}^{m_1}$ is the control signal, $z(k) \in \mathbb{R}^{p_1}$ is the error vector used for quantifying the H_∞ performance and $w(k) \in \mathbb{R}^{m_2}$ are the external perturbances, for each $\theta_k \in \mathscr{S}$.

Uncertainties under consideration are represented by the following equations:

$$A(\theta_k,k) = A(\theta_k) + \Delta_a(\theta_k,k)$$

$$B(\theta_k,k) = B(\theta_k) + \Delta_b(\theta_k,k)$$

$$C(\theta_k,k) = C(\theta_k) + \Delta_c(\theta_k,k)$$

$$D(\theta_k,k) = D(\theta_k) + \Delta_d(\theta_k,k)$$

where $A(\theta_k)$, $B(\theta_k)$, $C(\theta_k)$ and $D(\theta_k)$ are matrices with appropriate dimensions and represent the nominal model of the system. The uncertainties of the system are denoted by $\Delta_a(\theta_k,k)$, $\Delta_b(\theta_k,k)$, $\Delta_c(\theta_k,k)$ and $\Delta_d(\theta_k,k)$, which are unknown matrices satisfying the following:

$$\begin{pmatrix} \Delta_a(\theta_k,k) & \Delta_b(\theta_k,k) \\ \Delta_c(\theta_k,k) & \Delta_d(\theta_k,k) \end{pmatrix} = \begin{pmatrix} G_1(\theta_k) \\ G_2(\theta_k) \end{pmatrix} \Delta(\theta_k,k) \begin{pmatrix} H_1(\theta_k) & H_2(\theta_k) \end{pmatrix}$$

with $\Delta^T(\theta_k,k)\Delta(\theta_k,k) \leq I$, $\forall \theta_k \in \mathscr{S}$.

It is assumed that $\Sigma^*(\theta_k,k)$ is the real system and, as represented in Fig. 4.10, $\Sigma(\theta_k)$ is the nominal model of the system (without uncertainties), shown in equation (4.3). This nominal model will be useful when there is a dropout in the feedback channel. That way, it will be possible to estimate the state of the plant $\hat{x}$.

Furthermore, a buffer $b(\theta_k)$ will be used. This buffer will store the system states of $\Sigma^*(\theta_k,k)$ and also the states of the nominal system $\Sigma(\theta_k)$. When one or more dropouts occur, the buffer will provide the last available state from $\Sigma^*(\theta_k)$ and the state estimation from $\Sigma(\theta_k)$ corresponding to the same instant.

The filter to be designed will be splitted in two matrices: $F_1(\theta_k)$ and $F_2(\theta_k)$. Both together will provide an accurate correction for the estimation of the state $\hat{x}$.

$$\Sigma(\theta_k) := \begin{cases} \hat{x}(k+1) = A(\theta_k)\hat{x}(k) + B(\theta_k)u(k) \\ \hat{z}(k) = C(\theta_k)\hat{x}(k) + D(\theta_k)u(k) \end{cases} \tag{4.3}$$

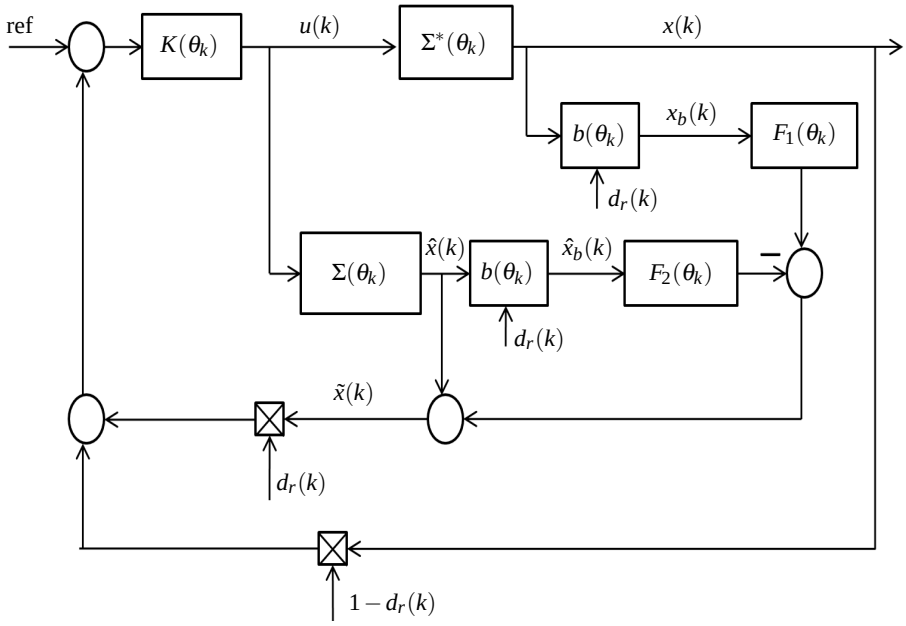

Figure 4.10: Scheme of the networked control system

The controller $K(\theta_k)$ is also represented in Fig. 4.10, and it will be designed, as the filters, depending on the mode θ_k.

4.3 Buffer policy

In order to store the states of the systems $\Sigma^*(\theta_k)$ and $\Sigma(\theta_k)$ and count the number of consecutive dropouts so the filters inputs are from the appropriate time instant, the buffer $b(\theta_k)$ is added to the system structure.

The buffer is represented by $b(\theta_k)$ with $N+1$ positions. Besides the system state, as can be seen in Fig. 4.10, it has also $d_r(k)$ as input. That will be useful to perfome the buffer policy. Using the input $d_r(k)$, the following variable can be computed:

$$d_c(k) = d_r(k)(1+d_c(k-1)),$$

which is a buffer inner variable that counts the number of consecutive dropouts at time instant k. Taking that into account, it will be possible to know which is the first non-empty

position in the buffer, that is, the most recent instant when there was not a dropout.

Thus, the output of the buffer will be

$$x_b(k) = \left(\begin{bmatrix} \delta_0(k) & \delta_1(k) & \cdots & \delta_N(k) \end{bmatrix} \otimes \mathbf{1}_n \right) \times \begin{bmatrix} x(k) \\ x(k-1) \\ \vdots \\ x(k_N) \end{bmatrix}, \tag{4.4}$$

where

$$\delta_0(k) = 1 - d_r(k),$$

$$\delta_1(k) = d_r(k)(1 - d_r(k-1))...(1 - d_r(k-N)),$$

$$\delta_2(k) = d_r(k)d_r(k-1)(1 - d_r(k-2))...(1 - d_r(k-N)),$$

$$\vdots$$

$$\delta_N(k) = d_r(k)d_r(k-1)...(1 - d_r(k-N)).$$

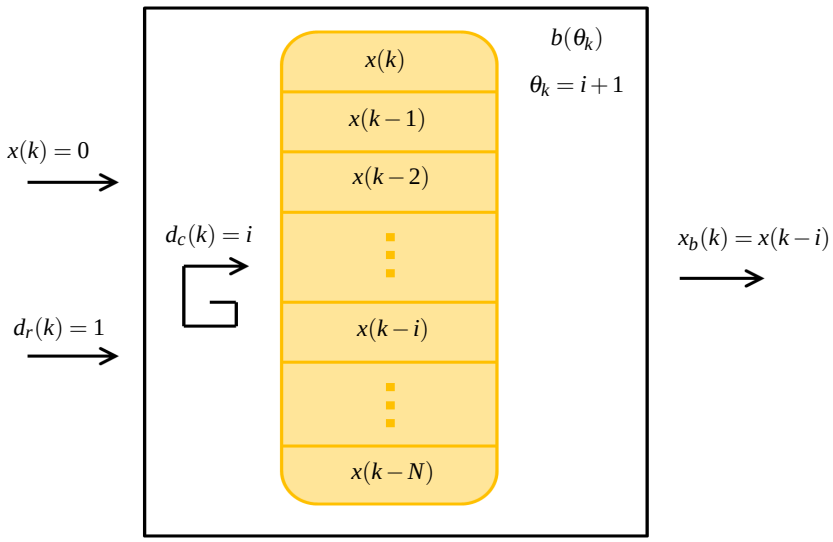

Figure 4.11: Buffer example

Fig. 4.11 represents an example wherein there is a dropout at instant k, that implies $x(k) = 0$, and there have been i consecutive dropouts, so the output of the buffer is $x_b(k) = x(k-i)$.

4.4 Model description

First of all, the different modes, depending on the droputs situation, are presented in the following:

$$\begin{aligned}
\theta_k = 1 &\Rightarrow \begin{cases} d_c(k) = 0 \\ \delta_0(k) = 1, \delta_1(k) = \delta_2(k) = ... = \delta_N(k) = 0 \end{cases} \\
\theta_k = 2 &\Rightarrow \begin{cases} d_c(k) = 1 \\ \delta_1(k) = 1, \delta_0(k) = \delta_2(k) = ... = \delta_N(k) = 0 \end{cases} \\
&\vdots \\
\theta_k = N+1 &\Rightarrow \begin{cases} d_c(k) = N \\ \delta_N(k) = 1, \delta_0(k) = ... = \delta_{N-1}(k) = 0 \end{cases}
\end{aligned} \tag{4.5}$$

Mode $\theta_k = 1$ represents the situation where there is not dropout at k, so the number of consecutive dropouts is zero. In mode $\theta_k = 2$ there is a dropout at k and there was no dropout at $k-1$, therefore in this case the number of consecutive dropouts is one. Following the same reasoning, the number of consecutive dropouts increase with the mode until the maximum is reached.

In order to make the system a Markov system, the Markovian property (4.1) has to hold. To ensure that, the state has to be augmented with the state of the previous time instants and their estimations, as it is shown in equation (4.6).

$$\mathbf{x}(k) = \begin{bmatrix} x^T(k) & x^T(k-1) & \cdots & x^T(k-N) & \hat{x}^T(k) & \hat{x}^T(k-1) & \cdots & \hat{x}^T(k-N) \end{bmatrix}^T \tag{4.6}$$

with $\mathbf{x}(k) \in \mathbb{R}^{2n(N+1)}$.

Therefore, the closed-loop *augmented model* of the system can be represented as:

$$\mathbf{x}(k+1) = (\underbrace{\mathbb{A}(\theta_k,k)+\mathbb{B}(\theta_k,k)\mathbb{K}(\theta_k)}_{\overline{\mathbb{A}}(\theta_k,k)})\mathbf{x}(k)+\mathbb{B}_w(\theta_k)w(k) \tag{4.7}$$

$$\mathbf{z}(k) = (\underbrace{\mathbb{C}(\theta_k,k)+\mathbb{D}(\theta_k,k)\mathbb{K}(\theta_k)}_{\overline{\mathbb{C}}(\theta_k,k)})\mathbf{x}(k)+\mathbb{D}_w(\theta_k)w(k), \tag{4.8}$$

With the uncertainties

$$\overline{\mathbb{A}}(\theta_k,k) = \overline{\mathbb{A}}(\theta_k)+\mathbb{G}_1(\theta_k)\Delta(\theta_k,k)\overline{\mathbb{H}}_1(\theta_k) \tag{4.9}$$

$$\overline{\mathbb{C}}(\theta_k,k) = \overline{\mathbb{C}}(\theta_k)+\mathbb{G}_2(\theta_k)\Delta(\theta_k,k)\overline{\mathbb{H}}_1(\theta_k), \tag{4.10}$$

with $\overline{\mathbb{H}}_1(\theta_k) = \mathbb{H}_1(\theta_k)+\mathbb{H}_2(\theta_k)\mathbb{K}(\theta_k)$

where

$$\mathbb{A}(\theta_k,k) = diag\{\begin{bmatrix} A(\theta_k,k) & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ \mathbf{0} & \ddots & & & \vdots \\ \vdots & & \ddots & & \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix}, \begin{bmatrix} A(\theta_k) & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ \mathbf{0} & \ddots & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix}\},$$

$$\mathbb{B}(\theta_k,k) = \begin{bmatrix} \mathbb{B}_{11}(\theta_k,k) & \mathbb{B}_{12}(\theta_k,k) \\ \mathbb{B}_{21}(\theta_k) & \mathbb{B}_{22}(\theta_k) \end{bmatrix},$$

$$\mathbb{B}_{11}(\theta_k,k) = \begin{bmatrix} \delta_0(k)B(\theta_k,k) & \delta_1(k)B(\theta_k,k) & \cdots & \delta_N(k)B(\theta_k,k) \\ & \varnothing & & \end{bmatrix},$$

$$\mathbb{B}_{12}(\theta_k,k) = \begin{bmatrix} d(k)B(\theta_k,k) & -\delta_1(k)B(\theta_k,k) & \cdots & -\delta_N(k)B(\theta_k,k) \\ & \varnothing & & \end{bmatrix},$$

$$\mathbb{B}_{21}(\theta_k) = \begin{bmatrix} \delta_0(k)B(\theta_k) & \delta_1(k)B(\theta_k) & \cdots & \delta_N(k)B(\theta_k) \\ & \varnothing & & \end{bmatrix},$$

$$\mathbb{B}_{22}(\theta_k) = \begin{bmatrix} d(k)B(\theta_k) & -\delta_1(k)B(\theta_k) & \cdots & -\delta_N(k)B(\theta_k) \\ & \varnothing & & \end{bmatrix},$$

The controller is composed by the following matrices:

$$\mathbb{K}(\theta_k) = diag\{\mathbb{K}_1(\theta_k), \mathbb{K}_2(\theta_k)\},$$

$$\mathbb{K}_1(\theta_k) = diag\{\delta_0(k)K(\theta_k), \delta_1(k)K(\theta_k)F_1(\theta_k), \ldots, \delta_N(k)K(\theta_k)F_1(\theta_k)\},$$

$$\mathbb{K}_2(\theta_k) = diag\{d_r(k)K(\theta_k), \delta_1(k)K(\theta_k)F_2(\theta_k), \ldots, \delta_N(k)K(\theta_k)F_2(\theta_k)\},$$

As it can be seen in Fig. 4.10, the filter is composed by two matrices: $F_1(\theta_k)$ and $F_2(\theta_k)$.

$$\mathbb{B}_w(\theta_k) = \begin{bmatrix} B_w(\theta_k) \\ \varnothing \end{bmatrix},$$

$$\mathbb{C}(\theta_k, k) = diag\{\begin{bmatrix} C(\theta_k,k) & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}, \begin{bmatrix} C(\theta_k) & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}\},$$

$$\mathbb{D}(\theta_k,k)=\left[\begin{array}{cc}\mathbb{D}_{11}(\theta_k,k) & \mathbb{D}_{12}(\theta_k,k)\\ \mathbb{D}_{21}(\theta_k) & \mathbb{D}_{22}(\theta_k)\end{array}\right],$$

$$\mathbb{D}_{11}(\theta_k,k)=\left[\begin{array}{cccc}\delta_0(k)D(\theta_k,k) & \delta_1(k)D(\theta_k,k) & \cdots & \delta_N(k)D(\theta_k,k)\end{array}\right],$$

$$\mathbb{D}_{12}(\theta_k,k)=\left[\begin{array}{cccc}d(k)D(\theta_k,k) & -\delta_1(k)D(\theta_k,k) & \cdots & -\delta_N(k)D(\theta_k,k)\end{array}\right],$$

$$\mathbb{D}_{21}(\theta_k,k)=\left[\begin{array}{cccc}\delta_0(k)D(\theta_k) & \delta_1(k)D(\theta_k) & \cdots & \delta_N(k)D(\theta_k)\end{array}\right],$$

$$\mathbb{D}_{22}(\theta_k,k)=\left[\begin{array}{cccc}d(k)D(\theta_k) & -\delta_1(k)D(\theta_k) & \cdots & -\delta_N(k)D(\theta_k)\end{array}\right]$$

and

$$\mathbb{D}_w(\theta_k)=\left[\begin{array}{c}D_w(\theta_k)\\ \varnothing\end{array}\right].$$

The uncertainties are now:

$$\mathbb{A}(\theta_k,k)=\mathbb{A}(\theta_k)+\Delta_{\mathbb{A}}(\theta_k,k)$$
$$\mathbb{B}(\theta_k,k)=\mathbb{B}(\theta_k)+\Delta_{\mathbb{B}}(\theta_k,k)$$
$$\mathbb{C}(\theta_k,k)=\mathbb{C}(\theta_k)+\Delta_{\mathbb{C}}(\theta_k,k)$$
$$\mathbb{D}(\theta_k,k)=\mathbb{D}(\theta_k)+\Delta_{\mathbb{D}}(\theta_k,k)$$

The unknown matrices satisfy the following:

$$\begin{pmatrix} \Delta_{\mathbb{A}}(\theta_k,k) & \Delta_{\mathbb{B}}(\theta_k,k) \\ \Delta_{\mathbb{C}}(\theta_k,k) & \Delta_{\mathbb{D}}(\theta_k,k) \end{pmatrix} = \begin{pmatrix} \mathbb{G}_1(\theta_k) \\ \mathbb{G}_2(\theta_k) \end{pmatrix} \Delta(\theta_k,k) \begin{pmatrix} \mathbb{H}_1(\theta_k) & \mathbb{H}_2(\theta_k) \end{pmatrix}$$

with

$$\mathbb{G}_1(\theta_k) = \begin{bmatrix} G_1(\theta_k) \\ \varnothing \end{bmatrix}, \quad \mathbb{G}_2(\theta_k) = \begin{bmatrix} G_2(\theta_k) \\ \varnothing \end{bmatrix},$$

$$\mathbb{H}_1(\theta_k) = \begin{bmatrix} H_1(\theta_k) & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}$$

and

$$\mathbb{H}_2(\theta_k) = \begin{bmatrix} H_2(\theta_k) & H_2(\theta_k) & \cdots & H_2(\theta_k) \end{bmatrix}.$$

4.5 H_∞ filter and controller synthesis

First, let us introduce an useful lemma for the main result:

Lemma 4.5.1. *(see [L. Xie, 1996]) Let Z, E, Δ, F be matrices with appropriate dimensions. Suppose Z is symmetric and $\Delta^T\Delta \leq \boldsymbol{I}$, then*

$$Z + E\Delta F + F^T\Delta^T E^T < 0$$

if and only if there exists scalar $\varepsilon > 0$ satisfying

$$Z + \varepsilon EE^T + \frac{1}{\varepsilon}F^T F < 0.$$

Next, we introduce the definition of robust MSS.

Definition: **Robust mean square stability.** System (4.3) is said to be robustly MSS if

$$\sum_{k=0}^{\infty} \mathbf{E}\{\|z(k)\|^2\} \leq \gamma^2 \sum_{k=0}^{\infty} \|w(k)\|^2$$

for any noise disturbance $w(\cdot) \in \ell^2$.

The following theorem is the main design result of this section.

Theorem 4.5.2. *If there exist matrices*

$$Y = (Y_1, ..., Y_N) > 0,$$

diagonal matrices

$$X = (X_1, ..., X_N) > 0,$$

scalars $\varepsilon_{\theta_k} > 0$ *and* $\mu > 0$ *such that the following LMI is feasible for all admissible uncertainties:*

$$\Psi < 0, \tag{4.11}$$

where

$$\Psi = \begin{bmatrix} -X_{\theta_k} & \boldsymbol{0} & J_{13}(\theta_k) & J_{14}(\theta_k) & J_{15}(\theta_k) \\ \boldsymbol{0} & -\mu \boldsymbol{I} & \mathbb{D}_w^T(\theta_k) & \mathbb{B}_w^T(\theta_k) W(\theta_k) & \boldsymbol{0} \\ J_{13}^T(\theta_k) & \mathbb{D}_w(\theta_k) & -\boldsymbol{I} + \varepsilon_{\theta_k} \mathbb{G}_2(\theta_k) \mathbb{G}_2^T(\theta_k) & \varepsilon_{\theta_k} \mathbb{G}_2(\theta_k) \mathbb{G}_1^T(\theta_k) W(\theta_k) & \boldsymbol{0} \\ J_{14}^T(\theta_k) & W(\theta_k)^T \mathbb{B}_w(\theta_k) & \varepsilon_{\theta_k} W(\theta_k)^T \mathbb{G}_1(\theta_k) \mathbb{G}_2^T(\theta_k) & J_4(\theta_k) & \boldsymbol{0} \\ J_{15}^T(\theta_k) & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\varepsilon_{\theta_k} \boldsymbol{I} \end{bmatrix}$$

and

$$J_{13}(\theta_k) = X_{\theta_k} \mathbb{C}^T(\theta_k) + Y_{\theta_k}^T \mathbb{D}^T(\theta_k),$$

$$J_{14}(\theta_k) = (\mathbb{A}(\theta_k) X_{\theta_k} + \mathbb{B}(\theta_k) Y_{\theta_k})^T W(\theta_k),$$

$$J_{15}(\theta_k) = X_{\theta_k} \mathbb{H}_1^T(\theta_k) + Y_{\theta_k}^T \mathbb{H}_2^T(\theta_k),$$

$$J_4(\theta_k) = -\mathscr{X} + \varepsilon_{\theta_k} W(\theta_k)^T \mathbb{G}_1(\theta_k) \mathbb{G}_1^T(\theta_k) W_{\theta_k}$$

with

$$\mathscr{X} = diag\{X_1, ..., X_{N+1}\},$$

then system (4.7) is robustly stochastically stable (MSS) with a noise attenuation level of the closed-loop system equal to $\sqrt{\mu}$.

Proof: Define

$$G(\theta_k) = W(\theta_k)\mathscr{P}W^T(\theta_k)$$

where

$$W(\theta_k) = \left(\begin{array}{ccc} \sqrt{p_{\theta_k,1}}\mathbf{I} & \cdots & \sqrt{p_{\theta_k,N+1}}\mathbf{I} \end{array} \right)$$

$$\mathscr{P} = diag\{P(1),...,P(N+1)\}$$

Define also the Lyapunov functional $V(\mathbf{x}(k),\theta_k)$ with the augmented state $\mathbf{x}(k)$

$$V(\mathbf{x}(k),\theta_k) = \mathbf{x}^T(k)P(\theta_k)\mathbf{x}(k)$$

$$\Delta V(\mathbf{x}(k),\theta_k) = \mathbf{E}\{V(\mathbf{x}(k+1),\theta_{k+1})|\mathbf{x}(k),\theta_k\} - V(\mathbf{x}(k),\theta_k)$$

$$\Delta V(\mathbf{x}(k),\theta_k) \leq \mathbf{x}^T(k)[\overline{\mathbb{A}}^T(\theta_k,k)G(\theta_k)\overline{\mathbb{A}}(\theta_k,k) - P(\theta_k)]\mathbf{x}(k) +$$
$$2\mathbf{x}^T(k)\overline{\mathbb{A}}^T(\theta_k,k)G(\theta_k)\mathbb{B}_w(\theta_k)w(k) + w^T(k)\mathbb{B}_w^T(\theta_k)G(\theta_k)\mathbb{B}_w(\theta_k)w(k)$$

In order to ensure the noise attenuation level $\gamma = \sqrt{\mu}$ the following terms are added:

$$\Delta V(\mathbf{x}(k),\theta_k) + \mathbf{z}^T(k)\mathbf{z}(k) - \gamma^2 w^T(k)w(k) = \xi^T(k)\Xi(\theta_k,k)\xi(k),$$

with

$$\xi^T(k) = \left[\begin{array}{cc} \mathbf{x}^T(k) & w^T(k) \end{array} \right]$$

and

$$\Xi(\theta_k,k) = \left[\begin{array}{cc} \Xi_{11}(\theta_k,k) & \Xi_{12}(\theta_k,k) \\ \Xi_{12}^T(\theta_k,k) & \Xi_{22}(\theta_k,k) \end{array} \right],$$

with

$$\Xi_{11}(\theta_k,k) = \overline{\mathbb{A}}^T(\theta_k,k)G(\theta_k)\overline{\mathbb{A}}(\theta_k,k) - P(\theta_k) + \overline{\mathbb{C}}^T(\theta_k,k)\overline{\mathbb{C}}(\theta_k,k),$$

$$\Xi_{12}(\theta_k,k) = \overline{\mathbb{A}}^T(\theta_k,k)G^T(\theta_k)\mathbb{B}_w(\theta_k) + \overline{\mathbb{C}}^T(\theta_k,k)\mathbb{D}_w(\theta_k)$$

and

$$\Xi_{22}(\theta_k,k) = -\gamma^2\mathbf{I} + \mathbb{D}_w^T(\theta_k)\mathbb{D}_w(\theta_k) + \mathbb{B}_w^T(\theta_k)G(\theta_k)\mathbb{B}_w(\theta_k).$$

And now a performance function J_T is introduced:

$$J_T = \mathbf{E}\{\sum_{k=0}^{T}[\mathbf{z}^T(k)\mathbf{z}(k) - \gamma^2 w^T(k)w(k)]|\mathbf{x}_0,\theta_0\}$$

$$J_T = \mathbf{E}\{\sum_{k=0}^{T}[\mathbf{z}^T(k)\mathbf{z}(k) - \gamma^2 w^T(k)w(k) + \Delta V(\mathbf{x}(k),\theta_k)]\} - \mathbf{E}\{\sum_{k=0}^{T}\Delta V(\mathbf{x}(k),\theta_k)\} =$$
$$= \sum_{k=0}^{T}\xi^T(k)\Xi(\theta_k,k)\xi(k) + V(\mathbf{x}(0),\theta_0) - V(\mathbf{x}(T+1),\theta_{T+1}) \leq V(\mathbf{x}(0),\theta_0)$$

therefore, when $T \to \infty \Rightarrow J_\infty \leq V(\mathbf{x}(0),\theta_0)$.

$$\Xi(\theta_k,k) = \underbrace{\begin{bmatrix} -P(\theta_k) & \mathbf{0} \\ \mathbf{0} & -\gamma^2\mathbf{I} \end{bmatrix}}_{\Xi(\theta_k)} +$$

$$\begin{bmatrix} \overline{\mathbb{A}}^T(\theta_k,k)W(\theta_k)\mathscr{P}W^T(\theta_k)\overline{\mathbb{A}}(\theta_k,k) + & \overline{\mathbb{A}}^T(\theta_k,k)W^T(\theta_k)\mathscr{P}W(\theta_k)\mathbb{B}_w(\theta_k) + \\ +\overline{\mathbb{C}}^T(\theta_k,k)\overline{\mathbb{C}}(\theta_k,k) & +\overline{\mathbb{C}}^T(\theta_k,k)\mathbb{D}_w(\theta_k) \\ \mathbb{B}_w^T(\theta_k)W(\theta_k)\mathscr{P}W^T(\theta_k)\overline{\mathbb{A}}(\theta_k,k) + & \mathbb{D}_w^T(\theta_k)\mathbb{D}_w(\theta_k) + \\ +\mathbb{D}_w^T(\theta_k)\overline{\mathbb{C}}(\theta_k,k) & +\mathbb{B}_w^T(\theta_k)W^T(\theta_k)\mathscr{P}W(\theta_k)\mathbb{B}_w(\theta_k) \end{bmatrix} =$$

$$= \Xi(\theta_k) + \begin{bmatrix} \overline{\mathbb{C}}^T(\theta_k,k) \\ \mathbb{D}_w^T(\theta_k) \end{bmatrix}\begin{bmatrix} \overline{\mathbb{C}}(\theta_k,k) & \mathbb{D}_w(\theta_k) \end{bmatrix} +$$
$$\begin{bmatrix} \overline{\mathbb{A}}^T(\theta_k,k)W(\theta_k)\mathscr{P}W^T(\theta_k)\overline{\mathbb{A}}(\theta_k,k) & \overline{\mathbb{A}}^T(\theta_k,k)W^T(\theta_k)\mathscr{P}W(\theta_k)\mathbb{B}_w(\theta_k) \\ \mathbb{B}_w^T(\theta_k)W(\theta_k)\mathscr{P}W^T(\theta_k)\overline{\mathbb{A}}(\theta_k,k) & \mathbb{B}_w^T(\theta_k)W^T(\theta_k)\mathscr{P}W(\theta_k)\mathbb{B}_w(\theta_k) \end{bmatrix} \leq 0$$

Applying Schur complement

$$\begin{bmatrix} -P(\theta_k) & \mathbf{0} & \overline{\mathbb{C}}^T(\theta_k,k) \\ \mathbf{0} & -\gamma^2\mathbf{I} & \mathbb{D}_w^T(\theta_k) \\ \overline{\mathbb{C}}(\theta_k,k) & \mathbb{D}_w(\theta_k) & -\mathbf{I} \end{bmatrix} +$$
$$+\begin{bmatrix} \overline{\mathbb{A}}^T(\theta_k,k)W(\theta_k) \\ \mathbb{B}_w^T(\theta_k)W(\theta_k) \\ \mathbf{0} \end{bmatrix}\mathscr{P}\begin{bmatrix} W^T(\theta_k)\overline{\mathbb{A}}(\theta_k,k) & W^T(\theta_k)\mathbb{B}_w(\theta_k) & \mathbf{0} \end{bmatrix} \leq 0$$

Taking into account the uncertainties,

$$\begin{bmatrix} -P(\theta_k) & \mathbf{0} & \overline{\mathbb{C}}^T(\theta_k) & \overline{\mathbb{A}}^T(\theta_k)W(\theta_k) \\ \mathbf{0} & -\gamma^2\mathbf{I} & \mathbb{D}_w^T(\theta_k) & \mathbb{B}_w^T(\theta_k)W(\theta_k) \\ \overline{\mathbb{C}}(\theta_k) & \mathbb{D}_w(\theta_k) & -\mathbf{I} & \mathbf{0} \\ W^T(\theta_k)\overline{\mathbb{A}}(\theta_k) & W^T(\theta_k)\mathbb{B}_w(\theta_k) & \mathbf{0} & -\mathscr{P}^{-1} \end{bmatrix} +$$

$$+ \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbb{G}_2(\theta_k) \\ W^T(\theta_k)\mathbb{G}_1(\theta_k) \end{bmatrix} \Delta(\theta_k,k) \begin{bmatrix} \overline{\mathbb{H}}_1(\theta_k) & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} +$$

$$+ \begin{bmatrix} \overline{\mathbb{H}}_1^T(\theta_k) \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \Delta^T(\theta_k,k) \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbb{G}_2^T(\theta_k) & \mathbb{G}_1^T(\theta_k)W(\theta_k) \end{bmatrix} \leq 0$$

And applying *Lemma* 4.5.1,

$$\begin{bmatrix} -P(\theta_k) & \mathbf{0} & \overline{\mathbb{C}}^T(\theta_k) & \overline{\mathbb{A}}^T(\theta_k)W(\theta_k) \\ \mathbf{0} & -\gamma^2\mathbf{I} & \mathbb{D}_w^T(\theta_k) & \mathbb{B}_w^T(\theta_k)W(\theta_k) \\ \overline{\mathbb{C}}(\theta_k) & \mathbb{D}_w(\theta_k) & -\mathbf{I} & \mathbf{0} \\ W^T(\theta_k)\overline{\mathbb{A}}(\theta_k) & W^T(\theta_k)\mathbb{B}_w(\theta_k) & \mathbf{0} & -\mathscr{P}^{-1} \end{bmatrix} +$$

$$+\varepsilon_{\theta_k} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbb{G}_2(\theta_k) \\ W^T(\theta_k)\mathbb{G}_1(\theta_k) \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbb{G}_2^T(\theta_k) & \mathbb{G}_1^T(\theta_k)W(\theta_k) \end{bmatrix} +$$

$$+\frac{1}{\varepsilon_{\theta_k}} \begin{bmatrix} \overline{\mathbb{H}}_1^T(\theta_k) \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} \overline{\mathbb{H}}_1(\theta_k) & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \leq 0$$

$$\begin{bmatrix} -P(\theta_k) & \mathbf{0} & \overline{\mathbb{C}}^T(\theta_k) & \overline{\mathbb{A}}^T(\theta_k)W(\theta_k) \\ \mathbf{0} & -\gamma^2\mathbf{I} & \mathbb{D}_w^T(\theta_k) & \mathbb{B}_w^T(\theta_k)W(\theta_k) \\ \overline{\mathbb{C}}(\theta_k) & \mathbb{D}_w(\theta_k) & -\mathbf{I}+\varepsilon_{\theta_k}\mathbb{G}_2(\theta_k)\mathbb{G}_2^T(\theta_k) & \varepsilon_{\theta_k}\mathbb{G}_2(\theta_k)\mathbb{G}_1^T(\theta_k)W(\theta_k) \\ W^T(\theta_k)\overline{\mathbb{A}}(\theta_k) & W^T(\theta_k)\mathbb{B}_w(\theta_k) & \varepsilon_{\theta_k}W^T(\theta_k)\mathbb{G}_1(\theta_k)\mathbb{G}_2^T(\theta_k) & \hat{J}_4(\theta_k) \end{bmatrix} +$$

$$\frac{1}{\varepsilon_{\theta_k}} \begin{bmatrix} \overline{\mathbb{H}}_1^T(\theta_k) \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} \overline{\mathbb{H}}_1(\theta_k) & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \leq 0,$$

$\hat{J}_4(\theta_k) = -\mathscr{P}^{-1} + \varepsilon_{\theta_k} W^T(\theta_k)\mathbb{G}_1(\theta_k)\mathbb{G}_1^T(\theta_k)W(\theta_k)$

Applying Schur complement, doing $X_{\theta_k} = P^{-1}(\theta_k)$ and pre- and post-multiplying by $\begin{bmatrix} X_{\theta_k} & \\ & \mathbf{I} \end{bmatrix}$,

$$\begin{bmatrix} -X_{\theta_k} & \mathbf{0} & X_{\theta_k}\overline{\mathbb{C}}^T(\theta_k) & X_{\theta_k}\overline{\mathbb{A}}^T(\theta_k)W(\theta_k) & X_{\theta_k}\overline{\mathbb{H}}_1^T(\theta_k) \\ \mathbf{0} & -\gamma^2\mathbf{I} & \mathbb{D}_w^T(\theta_k) & \mathbb{B}_w^T(\theta_k)W(\theta_k) & \mathbf{0} \\ \overline{\mathbb{C}}(\theta_k)X_{\theta_k} & \mathbb{D}_w(\theta_k) & -\mathbf{I}+\varepsilon_{\theta_k}\mathbb{G}_2(\theta_k)\mathbb{G}_2^T(\theta_k) & \varepsilon_{\theta_k}\mathbb{G}_2(\theta_k)\mathbb{G}_1^T(\theta_k)W(\theta_k) & \mathbf{0} \\ W^T(\theta_k)\overline{\mathbb{A}}(\theta_k)X_{\theta_k} & W^T(\theta_k)\mathbb{B}_w(\theta_k) & \varepsilon_{\theta_k}W^T(\theta_k)\mathbb{G}_1(\theta_k)\mathbb{G}_2^T(\theta_k) & J_4(\theta_k) & \mathbf{0} \\ \overline{\mathbb{H}}_1(\theta_k)X_{\theta_k} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\varepsilon_{\theta_k}\mathbf{I} \end{bmatrix} \leq 0$$

Letting $Y_{\theta_k} = \mathbb{K}(\theta_k)X_{\theta_k}$ and $\gamma^2 = \mu$, the LMI (4.11) is obtained. This completes the proof of Theorem 4.5.2.

□

4.5.1 Simulations results

This section presents some simulations results. The plant and the setup are described, providing all the considerations related to the scheme.

Plant description

The considered plant is a variant of the quadruple-tank process originally proposed in [Johansson, 2000], see [FeedBack, 2012]. Water is delivered to the four tanks by two independently controlled, submerged pumps. Notation related to the plant is given in Table 4.2.

For the simulationes, the following configuration is chosen (see Figure 4.12):

- Input water is delivered to the upper tanks. Pump 1 feeds tank 1 and pump 2 feeds tank 3.
- Tanks 1 and 3 are coupled by opening the corresponding valve.

Figure 4.13 shows a block diagram of the whole system. In tank 1 (respectively 3) the water level is measured and the control signal is applied to pump 1 (2). In the tanks 2 and 4

	Description
h_i	Water level of tank i
v_i	Voltage of pump i
h_i^0	Reference level of tank i
v_i^0	Reference voltage of pump i
Δh_i	Increment of h_i with respect to h_i^0
Δv_i	Increment of v_i with respect to v_i^0
s	Output to be tracked
r	Output reference for s
Δh_r	Reference level with respect to h^0
Δv_r	Reference voltage with respect to v^0

Table 4.2: Notation related to the plant

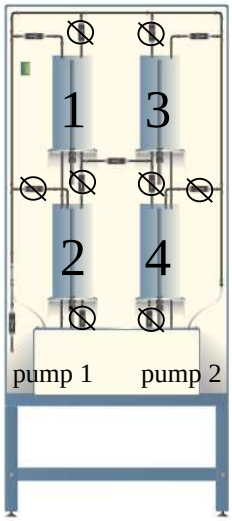

Figure 4.12: Schematic configuration of the coupled tanks

the water level is measured. The tanks are linked by means of the topology $2 \Leftrightarrow 1 \Leftrightarrow 3 \Leftrightarrow 4$.

The objective is to control the water level of the two lower tanks.

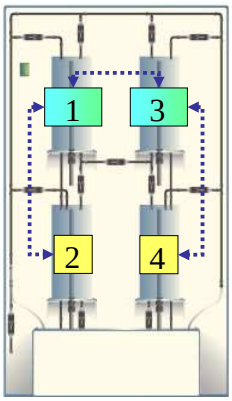

Figure 4.13: Control scheme with 4 tanks. Tanks 1 and 3 have sensors and actuators; tanks 2 and 4 have sensors. Blue dotted lines represent the communication links.

Plant modelling

The coupled tanks can be easily modelled by means of the following nonlinear equations:

$$\begin{aligned}
\frac{dh_1(t)}{dt} &= -\frac{a_1}{A}\sqrt{2gh_1(t)} + \eta v_1(t) - \frac{a_{13}}{A}\sqrt{2g(h_1(t)-h_3(t))},\\
\frac{dh_2(t)}{dt} &= \frac{a_1}{A}\sqrt{2gh_1(t)} - \frac{a_2}{A}\sqrt{2gh_2(t)},\\
\frac{dh_3(t)}{dt} &= -\frac{a_3}{A}\sqrt{2gh_3(t)} + \eta v_2(t) + \frac{a_{13}}{A}\sqrt{2g(h_1(t)-h_3(t))},\\
\frac{dh_4(t)}{dt} &= \frac{a_3}{A}\sqrt{2gh_3(t)} - \frac{a_4}{A}\sqrt{2gh_4(t)},
\end{aligned}$$

where $h_i(t)$ $(i = 1,...4)$ denotes the water level in the corresponding tank, v_i $(i = 1,2)$ are voltage applied to the pumps. a_i $(i = 1,...4)$ are the outlet area of the tanks, a_{13} is the outlet area between tanks 1 and 3; η is a constant relating the control voltage with the water flow from the pump, A is the cross-sectional area of the tanks, and g is the gravitational constant.

This system is linearized around the equilibrium point given by h_i^0 and u_i^0, yielding

$$\dot{\Delta h}(t) = A\Delta h(t) + B\Delta v(t), \tag{4.12}$$

where $\Delta h(t) = \left[h_1(t) - h_1^0 \ \dots \ h_4(t) - h_4^0\right]^T$ and $\Delta v(t) = \left[v_1(t) - v_1^0 \quad v_2(t) - v_2^0\right]^T$. Matrices A and B are obtained by using a Taylor expansion of the nonlinear equations of the model (4.13).

4.5.2 Results

In this section the simulation results are presented for the system with uncertainties. Taking into account the described plant and with a sampling time $t_m = 2s.$, the numerical values are:

$$A = \begin{bmatrix} -\frac{a_1 g}{A\sqrt{2gh_1^0}} - \frac{a_{13} g}{A\sqrt{2g(h_1^0 - h_3^0)}} & 0 & \frac{a_{13} g}{A\sqrt{2g(h_1^0 - h_3^0)}} & 0 \\ \frac{a_1 g}{A\sqrt{2gh_1^0}} & -\frac{a_2 g}{A\sqrt{2gh_2^0}} & 0 & 0 \\ \frac{a_{13} g}{A\sqrt{2g(h_1^0 - h_3^0)}} & 0 & -\frac{a_3 g}{A\sqrt{2gh_3^0}} - \frac{a_{13} g}{A\sqrt{2g(h_1^0 - h_3^0)}} & 0 \\ 0 & 0 & \frac{a_3 g}{A\sqrt{2gh_3^0}} & -\frac{a_4 g}{A\sqrt{2gh_4^0}} \end{bmatrix},$$

$$B = \begin{bmatrix} \eta & 0 \\ 0 & 0 \\ 0 & \eta \\ 0 & 0 \end{bmatrix} \tag{4.13}$$

	Value	Unit	Description
h_i	0-25	cm	Water level of tank i
v_i	0-5	V	Voltage level of pump i
A	0.01389	m^2	Cross-sectional area
a_i	50.265e-6	m^2	Outlet area of tank i
a_{13}	50.265e-6	m^2	Outlet area between tanks 1 and 3
η	0.22	$\frac{cm}{V.s}$	Contant relating voltage and flow
h_1^0	9.55	cm	Reference level of tank 1
h_2^0	16.9	cm	Reference level of tank 2
h_3^0	7.6	cm	Reference level of tank 3
h_4^0	14.1	cm	Reference level of tank 4
v_1^0	3.3	cm	Voltage level of pump 1
v_2^0	2.6	cm	Voltage level of pump 2

Table 4.3: Parameters of the plant.

$$A(\theta_k) = \begin{bmatrix} 0.9106 & 0 & 0.05119 & 0 \\ 0.0295 & 0.9736 & 0.0004058 & 0 \\ 0.03298 & 0 & 0.9284 & 0 \\ 0.000184 & 0 & 0.02085 & 0.9706 \end{bmatrix},$$

$$B(\theta_k) = \begin{bmatrix} 0.6585 & 0.008969 \\ 0.005122 & 0 \\ 0.005796 & 0.6595 \\ 0 & 0.003594 \end{bmatrix},$$

$$C(\theta_k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \textit{and} \quad D(\theta_k) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \forall \theta_k.$$

For this example, the chosen uncertainties are the following:

$$G_1(\theta_k) = G_2(\theta_k) = \begin{bmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix}, \quad \forall \theta_k,$$

$$H_1(\theta_k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \textit{and} \quad H_2(\theta_k) = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \forall \theta_k.$$

The external noise is: $|w(k)| \leq 1$, with

$$B_w = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T$$

and

$$D_w = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T.$$

Three different modes are considered, therefore: $\theta_k = \{1,2,3\}$. That means that the maximum number of consecutive dropouts is two. If $\theta_k = 1$ no dropout has occurred at k, if $\theta_k = 2$ a dropout has ocurred at k but the packet arrived at $k-1$, finally, if $\theta_k = 3$ there was a dropout at $k-1$ and also at k.

Given these modes, the trasition probabilities are:

$$p_{ij} = \begin{bmatrix} 0.4 & 0.2 & 0.3 \\ 0.5 & 0.2 & 0.3 \\ 0.2 & 0.5 & 0.3 \end{bmatrix}.$$

The water levels for the controlles systems are depicted in Fig. 4.14. In these simulations, the initial water levels are 20 cm for all the tanks.

The results for the Markovian structure presented in this chapter is compared in Fig. 4.14 with a classical structure, in where the controller input is null when a dropout occurs. The controller used in this structure is the calculated for $\theta_k = 1$, which is the mode that represents the no dropout situation.

It can be seen how the performance is improved with the Markovian structure, that provides a quicker response.

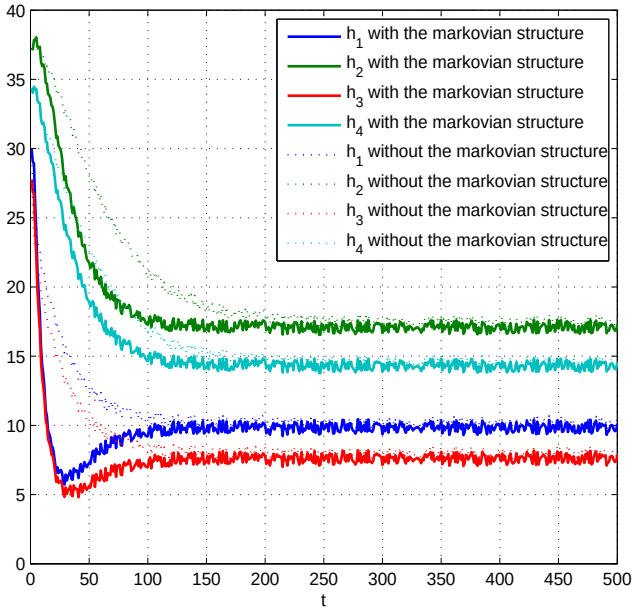

Figure 4.14: Water levels for the Markovian Jump Linear System.

The empirical performance measures are adopted

$$J_i \triangleq \sum_{k=1}^{500} h_i^2, \quad i = \{1,2,3,4\} \tag{4.14}$$

Table 4.4 shows how the performance improves by using the proposed method instead a classical structure.

Table 4.4: Performance indices J_i when controlling with the Markovian Jump structure with the filter and with the classical structure.

	Markovian structure	Classical structure
J_1	4.92×10^3	5.54×10^3
J_2	9.31×10^3	10.01×10^3
J_3	3.86×10^3	4.43×10^3
J_4	7.89×10^3	8.30×10^3

4.6 Conclusions

The paper has focused on a NCS subject to data dropouts constraints. In particular, the feedback path comprises a communication channel that produces data losses, are considered. It is considered that the maximum number of consecutive dropouts is known.

The NCS is modelled as a Markov Jump Linear System, with modes depending on the network situation.

The objective of this paper has been the synthesis of a controller and a filter that avoid the model uncertainties and support the failed transmissions. When a data dropout occurs, the sytem uses an estimated output given by the filter to do the feedback. To perform this task, a H_∞ control problem has been proposed in order to calculated the controller. The filter is calculated with an H_∞ technique together with the controller.

Finally, a plant has been chosen to obtain some numerical results that illustrate the closed-loop system performance. These simulations corroborated that robust performance is achieved.

4.7 Related publications

- Isabel Jurado, Manuel G. Ortega, Francisco R. Rubio and Julio E. Normey-Rico. *A robust filter and controller design for NCS with uncertainties and data dropouts*. Proceedings the 9th IEEE International Conference on Control Automation (IEEE ICCA), 2011.

- Isabel Jurado, Manuel G. Ortega, Francisco R. Rubio y Julio E. Normey-Rico. *Diseño de un filtro y un controlador robusto para NCS con incertidumbres y pérdidas de datos*. Actas de las XXXII Jornadas de Automática, Sevilla, España, Septiembre 2011.

Chapter 5

Packetized Model Predictive Control for Networked Control Systems subjects to time-delays and dropouts

In this chapter, two predictive packetized predictive control techniques are introduced in order to deal with time-delays and packet dropouts.

Therefore, this chapter presents a model predictive control formulation for Networked Control Systems subject to independent and identically distributed (i.i.d.) delays and packet dropouts. The system takes into account the presence of a communication network in the control loop, resorting to a buffer in the actuator to store and consistently apply delayed control sequences when fresh control inputs are not available.

The first approach presents a practical algorithm to design networked control systems able to cope with high data dropout rates. The algorithm is intended for application in packet based networks protocols (Ethernet-like) where data packets typically content large data fields. The key concept consists in the use of packets to transmit not only the current control signal, but predictions on a finite horizon without significantly increasing traffic load. Thus, predictive control is used together with buffered actuators and a state estimator to compensate for eventual packet dropouts. Additionally, some ideas are proposed to decrease traffic load, limiting packet size and media access frequency. Simulation results on

the control of a three-tank system are given to illustrate the effectiveness of the method.

The second approach presents a stochastic model predictive controller. It has been proposed to send from the controller a sequence of control signals that, appropriately buffered and scheduled at the actuator end, becomes a safeguard in case of delays or eventual packet dropouts. Although a significant body of research has developed different strategies, combining MPC and buffering strategies there is still room for further research and improvements. On the one hand works such as [D. Quevedo, J. Østergaard and D. Nešić, 2011] or [D. Muñoz and P. D. Christofides, 2008] neglect the effect of the network induced delays focusing the attention on the problem of packet dropouts, while in [G.P. Liu, J.X. Mu, D.Rees, S.C. Chai, 2006] only delays are considered. Further, in many works on MPC for NCS a deterministic approach is considered, yielding a worst-case approach.

This second technique considers both packet dropouts and random delays. A stochastic approach is adopted which allows to improve the control performance provided that the statistical distribution of the delays are known.

5.1 Networked Predictive Control of Systems with Data Dropouts

5.1.1 Problem statement

This technique is focused on the design of a predictive control structure for a networked control system with packet dropouts.

Systems to be considered are unconstrained discrete-time linear multiple-inputs plants, under the effect of bounded disturbances as:

$$x(k+1) = Ax(k) + Bu(k) + B_w w(k) \tag{5.1}$$

with $k \in \mathbb{N}_0 \triangleq \mathbb{N} \cup \{0\}$ and

$$u(k) \in \mathbb{U} \subseteq \mathbb{R}^{m_1}, \quad x(k) \in \mathbb{X} \subseteq \mathbb{R}^n, \quad \forall k \in \mathbb{N}_0$$

disturbances, $w(k)$ are considered to be bounded as

$$w(k) \in \mathbb{W}, \quad \mathbb{W} = \{x \in \mathbb{R}^n / \|x\| < \delta\}$$

In this setup, the plant and controller are assumed to be linked through a communication network (see Figure 5.1). Our interest lies in clock-driven Ethernet-like networks linking both, controller outputs to plant inputs, and plant outputs (sensors) to controller inputs. Data are sent in large packets, so that the relevant phenomena for control purposes are transmission delays and packet dropouts.

More precisely, only the problem of packet dropouts is addressed. Random delays are not a concern in this section, since small round-trip communication delays (the sum of delays from the sensor to the controller and from this to the actuator) are assumed, that is, delays are considered negligible with respect to sample periods. Thus, in the event that data packets do not arrive, or arrive later than a certain threshold, they are considered as missing packets.

This approach does not assume secured links in neither end of the communications chain. That is, packets can be dropped either in sensor to controller path, or in the controller to actuator one. This feature is particularly remarkable since usually dropouts are only considered in the controller to actuator path.

To this end, acknowledgment is assumed as part of the network protocol (TCP-like protocols), so that at any time instant k, the controller knows whether a control packet arrived at destination or not. Packets are also assumed to be time-stamped so they can be correctly sequenced at any point of the control loop.

To summarize, for the proposed control algorithm to work, all elements in the control loop are assumed to behave in a time-driven manner. Thus, the network model operates at the same sampling rate as the plant-controller model, with the following rules:

1. Time-driven sensors periodically sample plant outputs and states.

2. A time-driven predictive controller computes a control sequence at each sampling time.

3. A time-driven buffered actuator applies control signals at each sampling time.

4. Network is affected by dropouts at any point.

5. Delayed packets are taken as dropouts.

In order to achieve an appropriate performance level, it is proposes the use of a finite horizon predictive optimal control framework.

The predictive controller has access to the plant states $x(k)$, and computes at every time instant k a finite horizon optimal control sequence $U_k \in (\mathbb{U})^{N_u}$ of length N_u, such that the following functional is minimized

$$V(U(k),k) = \sum_{i=k}^{k+N_u-1} \ell(x'(i),u'(i)) + F(x'(k+N_u)) \tag{5.2}$$

where $x'(\cdot)$ and $u'(\cdot)$ denote predicted plant states and outputs respectively. Also in (5.2), $\ell(\cdot)$ denotes the stage cost and $F(\cdot)$ is the terminal cost.

Assuming this setup, it is shown next how this predictive control structure can be combined with an appropriate buffering and queuing strategy providing remarkable robustness to packet dropouts.

5.1.2 Packetized control and buffering strategy

In order to compensate for eventual packet dropouts and delays, one key feature of this first proposed predictive control scheme is buffering control signals in the actuator side.

In this scheme, also exploited for instance in [Yang et al., 2006], [Wenshan et al., 2007], [D. Muñoz, C. Panagiotis D., 2007], the buffered signals act as a safeguard against packet dropouts. Thus, as depicts the proposed control structure in Figure 5.1, the buffer stores a

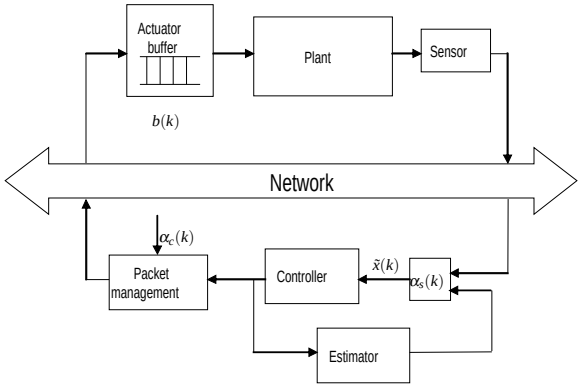

Figure 5.1: NCS proposed scheme

number of model-based predictions on future control actions, so the actuator can provide appropriate control in the event of dropouts.

The buffering policy is designed such that whenever new packets arrive, buffer is overwritten. The actuator is then sequentially feeded with the information in the buffer until new packets arrive. This corresponds to the intuitively appealing idea of "Use the most recent control sequence if available. If not, use predictions from the buffer."

The amount of consecutive dropouts this strategy can compensate for, is obviously equal to the buffer length. In this sense, the buffer can be reasonably dimensioned to store as many control actions as the prediction horizon N_u[1], and so is the maximum consecutive dropouts allowed by the proposed control structure.

This simple idea can be formalized as: Let us represent the state of the buffer at a given time instant k as $b(k) \in (\mathbb{U})^{N_u}$. Then, the dynamics of the buffer can be expressed as the recursive rule

$$b(k) = \alpha_c(k)U_k + (1-\alpha_c(k))Sb(k-1) \tag{5.3}$$

[1]Note that a larger buffer size is useless as the buffer receives at most N_u control predictions. There is also little point in using a smaller buffer since the last few predictions of every received sequence would be lost.

where matrix $S \in \mathbb{R}^{m_1 N_u \times m_1 N_u}$ is a shift matrix defined as the block matrix

$$S_{i,j} = \delta_{i+1,j} \cdot I_{m_1}; \quad i,j = 1,...,N_u.$$

In (5.3), $\alpha_c(k) \in \{0,1\}$ is a signal accounting for reception acknowledgment in the controller to actuator link, such that

$$\alpha_c(k) = \begin{cases} 1 & \text{if packet } U(k) \text{ arrives to buffer at time } k \\ 0 & \text{if packet } U(k) \text{ does not arrive to buffer at time } k \end{cases}$$

With this description the control action $u(k)$ released from the buffer at instant k can be expressed as

$$u(k) = \begin{bmatrix} I_{m_1} & 0_{m_1} & ... & 0_{m_1} \end{bmatrix} b(k).$$

This basic mechanism implicitly assumes that the controller computes and sends a whole control sequence of length N_u at every sampling time. This, as it has been discussed, does not significantly increase network load, as information is bundled in rather lengthy packets.

Nonetheless, specific situations in networked control systems suggest to reduce network access to its minimum. That is the case for instance of wireless sensor networks where typically energy saving is a major concern. In this kind of systems, it is advisable to design network protocols that avoid unnecessary network use, for example transmitting data packets of minimum length and only when relevant information for control is available.

In this sense, a further refinement can be introduced in this scheme in order to alleviate network load to a greater extent. The key idea here is comparing at every time instant k the control sequence in the buffer, $\beta(k)$, and the current controller sequence in the actuator, $U(k)$. This comparison is performed in the controller, so if both sequences match up to a certain degree, only the relevant changes are sent, or even no sequence might need to be sent at all.

Note that, as an acknowledgment signal $\alpha_c(k)$ is assumed part of the protocol, the controller has full access to the buffer state, $b(k)$, at every time instant k. That is, the buffer dynamics can be accurately reproduced at the controller side.

It is a natural assumption that the buffer's most distant predictions in the future, should be those differing to a greater extent with the more recently computed sequences in the controller. This intuitive idea suggests reducing packet size to form a trimmed packet by sending just the last few components that differ more than a certain threshold from the buffer state.

This packet management policy can be formalized as:

Consider $b(k)$ the buffer state, and $U(k)$ the computed control sequence at time instant k. Denote $b_j(k)$ and $U_j(k)$ as the j-th component of the corresponding sequences at time instant k.

The length of a trimmed packet N_T can be determined according to buffer at instant k as

$$N_T = N_u - \arg \min_{j \in \{1,\dots,N_u\}} \|U_j(k) - b_j(k)\| > \varepsilon.$$

That is, only the last N_T components of sequence $U(k)$ need to be sent to the buffer, that is with the actuator as represented in Figure 5.1, as the first $N_u - N_T$ match those in the buffer up to a certain tolerance ε.

With this definition, a trimmed packet of length $N_T \leq N_u$, $U^*(k) \in (\mathbb{R})^{N_T}$, can be built as

$$U^*(k) = \begin{bmatrix} 0_{m_1 N_T \times m_1 (N_u - N_T)} & I_{m_1 N_T} \end{bmatrix} U(k).$$

Thus, the buffer dynamics in (5.3) can be trivially modified to deal with trimmed packets $U^*(k)$ as

$$b(k) = \alpha_c(k) \left[b_1^T(k-1), \dots, b_{N_u - N_T}^T(k-1), (U_1^*(k))^T, \dots, (U_{N_T}^*(k))^T \right]^T$$

$$+ (1 - \alpha_c(k)) S b(k-1) \tag{5.4}$$

The proposed networked control structure in this work also considers the possibility of missing data packets in the sensor to actuator path. This issue, not treated in most previous

works, is specially relevant to take into account realistic networked control problems, as plant and controller are usually physically distributed.

To deal with eventual missing state measures, this work resorts to a model-based estimator that approximates plant states when no updated information from the sensors is received. The estimator takes the form

$$\hat{x}(k+1) \quad = \quad \alpha_s(k)x(k) + (1-\alpha_s(k))\, f(\hat{x}(k), u(k)) \tag{5.5}$$

where $\hat{x}(k) \in \mathbb{R}^n$ is the estimated plant state at instant k, and $f(\hat{x}(k), u(k))$ is an open-loop approximation of the plant dynamics. Considering the plant model (5.1), $f(\hat{x}(k), u(k)) = A\hat{x}(k) + Bu(k)$ can be taken.

As in equation (5.3), the estimator (5.5) makes use of a signal $\alpha_s(k)$ accounting in this case for the acknowledgment of reception of packets sent in the link from sensors to controller. In a similar fashion

$$\alpha_s(k) = \begin{cases} 1 & \text{if packet } U(k) \text{ arrives to controller at time } k \\ 0 & \text{if packet } U(k) \text{ does not arrive to controller at time } k \end{cases}$$

where $\alpha_s(k) = d_r(k) - 1$.

As can be easily interpreted from the estimator equation (5.5), the estimated state $\hat{x}(k)$ is updated with the measured state $x(k)$ when data packet from the sensor arrives, otherwise the plant state is estimated from the plant model.

It is worth to mention that signal $\alpha_s(k)$ is not directly provided by the network protocol, as packet reception is acknowledged at the controller side. Nonetheless, $\alpha_s(k)$ can be synthesized from the packet time stamps arriving from the sensor. Since clock-driven networking protocol is assumed, a simple procedure consists in checking the arrival time of every packet, so that only those arriving within the current sample period, are considered as valid states measures, otherwise dropout is assumed.

The addition of the estimator in the control scheme allows the controller to be feeded with the plant states at each sampling time, regardless of packet dropouts. This input to the controller can be measured or estimated depending on the arrival of the most recent sensor packet.

This estimation procedure at the controller together with the packet management policy above discussed, constitute the basic predictive networked control scheme proposed in this work. It is worth to remember that the network predictive controller is not required to satisfy network constraints of any kind, as it is designed regardless of the underlying network structure. From this point of view, the proposed methodology can be regarded as network compensation technique rather than a control methodology by itself.

Remark (Stability)
Stability of the proposed compensation methodology can be ensured as far as a number of mild requirements are satisfied.

Let $u(k)$ be the stabilizing predictive control action for system (5.1) computed at time instant k without network. As it has been discussed, packet dropouts associated to the inclusion of a network in the control structure, implies that there is no guaranty that signal $u(k)$ is accurately applied at every time instant k. Instead, the presented methodology computes a compensated control, $u_c(k)$, based on buffered predictions and state estimations.

From this point of view, the inclusion of the network, together with the proposed compensation scheme, amounts to introducing an additional disturbance term on system (5.1), as the following decomposition suggest

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + B(u_c(k) - u(k)) + B_w w(k) \\ &= Ax(k) + Bu(k) + Bw_u(k) + B_w w(k) \end{aligned} \tag{5.6}$$

where $w_u(k) = u_c(k) - u(k)$ represents the network effect on the predictive control structure.

Moreover, it can be checked that this additional term $w_u(k)$ is bounded. Notice that, as new packets arrive, the buffer and estimator are reset to match the computed sequence, hence $w_u(k) = 0$. As by assumption the number of consecutive network dropouts is limited,

the difference between the compensated control, $u_c(k)$, and the computed control, $u(k)$, can only grow between valid packets, thus it is bounded.

Also by assumption the term $w(k)$ is bounded, thus the overall disturbance term $\Omega(k) = Bw_u(k) + B_w w(k)$ is also bounded.

As discussed in [D.Q. Mayne, J.B. Rawlings, C.V. Rao, 2000] a stabilizing predictive controller can always be found under appropriate conditions for the unperturbed system (5.6).

Recalling results in [D. Limon, T. Alamo, E.F. Camacho, 2002]:

(A1) Let $x(k+1) = F(x(k))$ be the closed loop dynamics of the unperturbed system (5.6), with the origin being a fixed point.

(A2) Let $V(x)$ a Lyapunov function of the system Lipschitz in a neighborhood of the origin $\Lambda_r = \{x \in \mathbb{R}^n / V(x) \leq r\}$ such that

$$a \cdot \|x\|^p \leq V(x) \leq b \cdot \|x\|^p$$

$$V(F(x)) - V(x) \leq -c \cdot \|x\|^p$$

where a, b, c are positive constants and $p > 1$.

Then there exits a constant $\mu > 0$ such that for all disturbances $\Omega(k) \in B_\mu = \{\Omega(k) \in \mathbb{R}^n / \|\Omega(k)\| < \mu\}$ the perturbed system $x(k+1) = F(x(k)) + \Omega(k)$ is asymptotically ultimately bounded $\forall x(0) \in \Lambda_r$.

To conclude stability of the proposed control methodology, notice that conditions (A1) and (A2) are satisfied for system (5.6) taking $p = 2$, and considering a Lyapunov function of the form $V(x) = x^T P x$, which is trivially Lipschitz in a neighborhood of the origin Λ_r for arbitrarily large values of r.

5.1.3 Simulation results

Process modelling

The proposed algorithm has been tested on a level control model as depicted in Figure 5.2. The system is composed of three tanks, with the control problem consisting in tracking a reference level in the last one, acting on the flow poured in the first one. The model of the process can be easily obtained from a mass balance as:

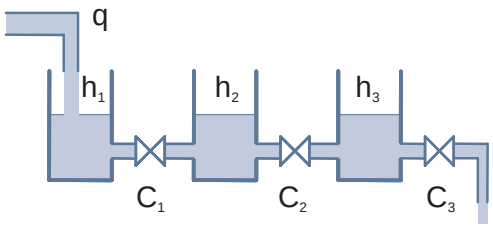

Figure 5.2: Three tank system

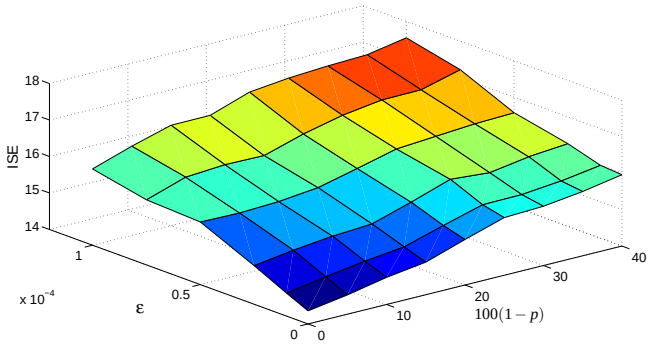

Figure 5.3: Influence of allowed error

$$\begin{aligned} \frac{dh_1}{dt} &= \frac{1}{S}\,q - \frac{1}{S}\,C_1\sqrt{h_1 - h_2} \\ \frac{dh_2}{dt} &= \frac{1}{S}\,C_1\sqrt{h_1 - h_2} - \frac{1}{S}\,C_2\sqrt{h_2 - h_3} \\ \frac{dh_3}{dt} &= \frac{1}{S}\,C_2\sqrt{h_2 - h_3} - \frac{1}{S}\,C_3\sqrt{h_3} \end{aligned} \tag{5.7}$$

where h_i represent the level of tank i.

The system is linearized to apply the proposed control structure around a trimming point H_1, H_2, H_3 and Q. Thus:

$$h_1 = H_1 + \Delta H_1, h_2 = H_2 + \Delta H_2$$

$$h_3 = H_3 + \Delta H_3, q = Q + \Delta Q$$

yielding the linear equation:

$$\Delta \dot{H} = L\,\Delta H + M\,\Delta Q \tag{5.8}$$

where:

$$\Delta H = \begin{bmatrix} \Delta H_1 & \Delta H_2 & \Delta H_3 \end{bmatrix}^T$$

$$L = \begin{bmatrix} \frac{-C_1}{2S\sqrt{H_1 - H_2}} & \frac{C_1}{2S\sqrt{H_1 - H_2}} & 0 \\ \frac{C_1}{2S\sqrt{H_1 - H_2}} & \frac{-C_1}{2S\sqrt{H_1 - H_2}} - \frac{C_2}{2S\sqrt{H_2 - H_3}} & \frac{C_2}{2S\sqrt{H_2 - H_3}} \\ 0 & \frac{C_2}{2S\sqrt{H_2 - H_3}} & \frac{-C_2}{2S\sqrt{H_2 - H_3}} - \frac{C_3}{2S\sqrt{H_3}} \end{bmatrix}$$

$$M = \begin{bmatrix} \frac{1}{S} & 0 & 0 \end{bmatrix}^T$$

A discrete model is then easily obtained from (5.8) as

$$x(k+1) = Ax(k) + Bu(k)$$

Application to a three-tank system

A number of simulations for different network operational conditions have been performed, taking as system parameters $S = 0.16\,m^2$, $C_1 = C_2 = 0.0256\,\frac{m^3}{s\,m^{1/2}}$ and $C_3 = 0.0251\,\frac{m^3}{s\,m^{1/2}}$, with an operation point $H_1 = 1\,m$, $H_2 = 0.7\,m$, $H_3 = 0.4\,m$, $Q = 0.014\,m^3/h$.

As an standard tool to compare performance results, the integral square error (ISE) measure has been employed.

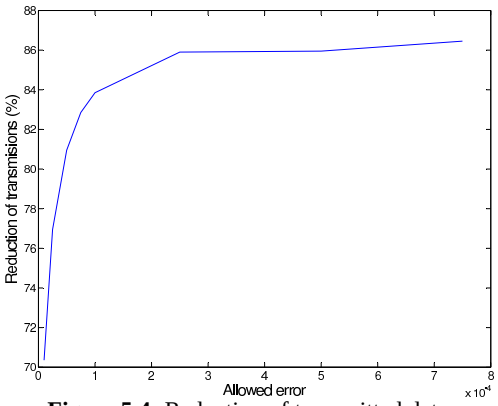

Figure 5.4: Reduction of transmitted data

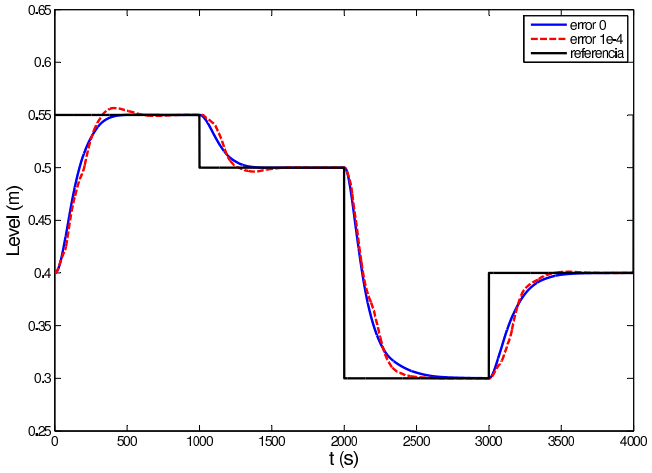

Figure 5.5: Step Response

First, the proposed strategy for reducing network traffic is compared with the conventional case where the entire control sequence is sent over the network at each sampling time. The influence of a successful transmission probability p, and allowed error ε is shown in Figure 5.3. This graphic represents the average ISE performance index for a number of experiments taking a step-like sequence with period $T = 2000\,s$ as reference. and a simulation time of 5000 s.

In Figure 5.4 the percentage reduction of controller-to-actuator transmissions is shown. This reduction is computed as the amount of information transmitted with the proposed

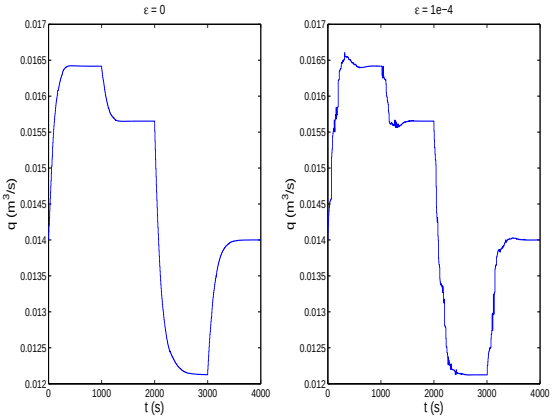

Figure 5.6: Tracking of references

queueing/buffering scheme with respect to the full information transmission case. It can be observed that savings above 85% can be obtained for sufficiently high allowed error ε. Nonetheless, from Figure 5.3, it is clear that there is little point in taking excessively high values of ε, as ISE performance starts degrading faster than transmission saving. For instance, in view of Figure 5.3 and Figure 5.4, by selecting $\varepsilon = 0.2 \cdot 10^{-4}$ a reduction of 70% is reached without worsening significantly the system response.

Figures 5.5 and 5.6 show the system response with a remarkable 30% packet dropout probability. It can be observed that the algorithm retains good performance even when with high dropout probability. Nonetheless, in some cases, depending on the random dropouts, the response may exhibit small overshoot.

Not surprisingly, performance degrades as either the allowed error ε, or the data dropout rate, increase. Remarkably, the controller can cope with data dropout rates above 40%.

In Figure 5.7, the transmission profile for a step tracking experiment with different values of ε are shown. It can be observed that, as expected, an intense transmission pattern is observed for the first instants of simulation, corresponding to the transient regime. As the system approaches steady state, traffic load is drastically reduced.

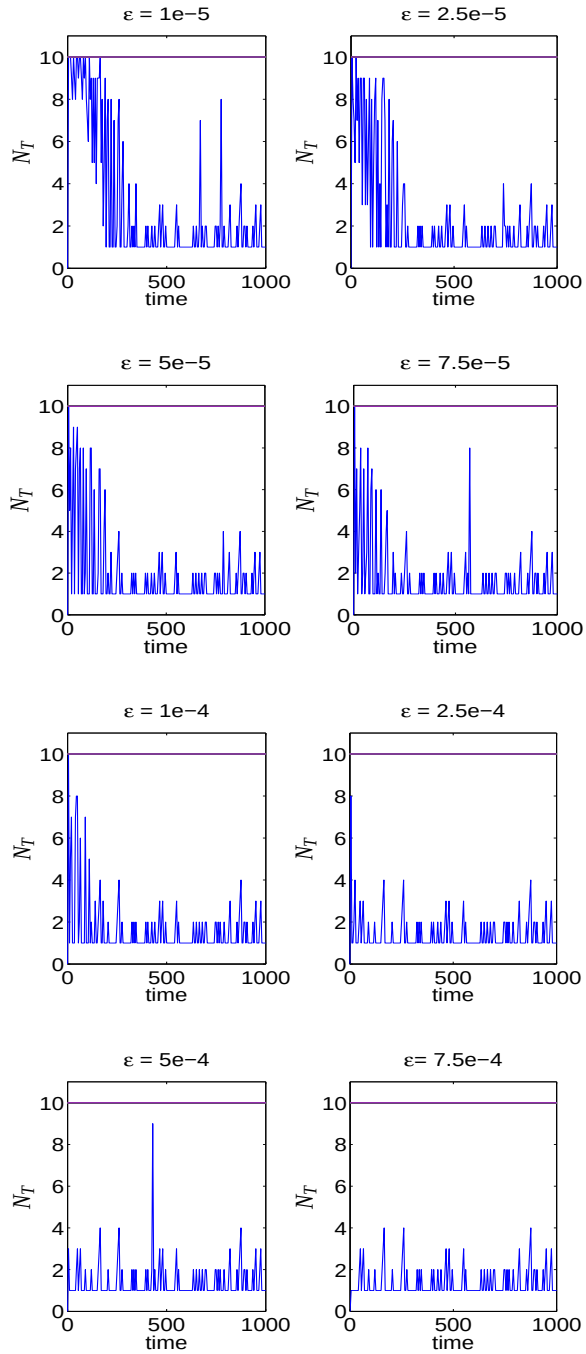

Figure 5.7: Number of components of $U^*(k)$ sent

5.2 Stochastic Packetized Model Predictive Control for Networked Control Systems subjects to time-delays and dropouts

In this section both packet dropouts and random delays are considered . A stochastic approach is adopted, which allows to improve the control performance provided that the statistical distribution of the delays are known.

5.2.1 Problem formulation

Consider the following discrete linear system:

$$x(k+1) = Ax(k)+Bu(k)+B_w w(k), \tag{5.9}$$

$$x(0) = x_0. \tag{5.10}$$

where $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^{m_1}$ and are the state vector and control input vector respectively and $w(k) \in \mathbb{R}^{m_2}$ is an exogenous disturbance affecting to the plant.

In this setup, the plant and controller are assumed to be linked through a communication network (see Fig. 5.8). The relevant phenomena to consider in this section are transmission delays and packet dropouts, which can degrade the control performance or even destabilize the plant. The random nature of both effects in real-time communication networks motivates the stochastic approach taken in this work. Delays and dropouts are assumed to be stochastic i.i.d. processes with known statistical distributions.

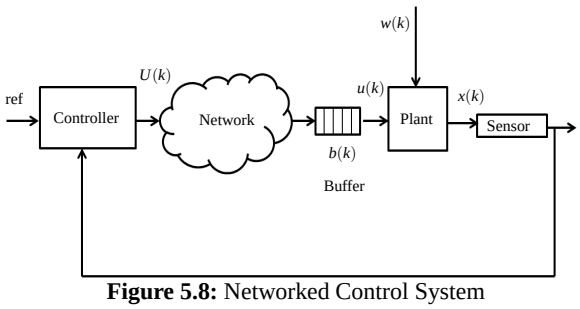

Figure 5.8: Networked Control System

To summarize, for the proposed control scheme to work, all elements in the control loop are assumed to behave in a time-driven manner, with the following elements:

1. Sensors periodically sample the plant state $x(k)$ and send it to the controller.

2. A stochastic predictive controller computes a control sequence $U(k) = [u(k|k)\ u(k+1|k)\ ...\ u(k+N_u|k)]$ at each sampling time and sends it through the network.

3. At the actuator side, control inputs are applied to the plant according to the last signal stored in the buffer. The buffer is updated discarding old control sequences whenever a newer one arrives.

4. Network is affected by i.d.d. dropouts and i.d.d delays $\tau(k)$. Where

$$\tau(k) = \begin{cases} i & \text{if } U(k) \text{ is received at time } k+i \\ & \quad \text{at the actuator node,} \\ \infty & \text{if } U(k) \text{ is lost} \end{cases} \tag{5.11}$$

Assumption 1: The process $\{\tau(k)\}_{k\in\mathbb{N}_0}$ is i.i.d., with delay distribution,

$$\textbf{Prob}\{\tau(k) = i\} = p_i, \quad i \in \mathbb{N}_0, \tag{5.12}$$

and dropout probability $\textbf{Prob}\{\tau(k) = \infty\} = p_\infty$.

In order to achieve an appropriate performance level, this work proposes the use of a stochastic predictive controller framework. That way, the controller will try to minimize the expected value of the following cost function:

$$V(x(k), \mathscr{U}_d(k), \mathscr{T}(k), U(k)) = \sum_{i=k}^{k+N_u-1} \ell\big(x'(i), u'(i)\big) + F\big(x'(k+N)\big), \tag{5.13}$$

where N_u is the prediction horizon, $x(k)$ is the measured state of the plant in k, $\mathscr{U}_d(k)$ is the set of optimal control sequences sent between $k-1$ and $k-\tau^{max}$,

$$\mathscr{T}(k) \triangleq \tau(i), \quad \forall i \in [k, k-1, ..., k-\tau^{max}]$$

is the set of possible delays of those control sequences, $U^*(k)$ is the new control sequence to be computed by the controller at time k, $\ell(\cdot)$ denotes the stage cost and $F(\cdot)$ is the

terminal cost. Moreover, $x'(i)$ and $u'(i)$ are state and control input open-loop predictions according to the buffer policy and the delay and dropouts statistical distribution:

$$\text{Open loop predictions} \begin{cases} x'(k) = x(k), \\ x'(k+1) = Ax(k) + Bu'(k), \\ x'(k+2) = Ax'(k+1) + Bu'(k+1), \\ \quad \vdots \end{cases} \tag{5.14}$$

where $u'(k)$, $u'(k+1)$, ... is the predicted control sequence.

When random time-varying delays and dropouts are taken into account, one of the main difficulties is the impossibility of predicting the system trajectory in a deterministic way, as the inputs actually applied to the plant are unknown by the controller. Different approaches, including min-max or worst-case approaches can be taken to deal with this difficulty.

In this work it is exploited the fact that the statistics of time delays and dropouts can be measured or estimated to improve the control performance. That way, the open-loop predictions described above depend on future delay and dropout realizations, so that the control inputs applied to the plant can be predicted by explicit enumeration of the realizations. For instance, when considering the case $\tau(k) = t$, then $u'(k+t) = u(k+t|k)$, $u'(k+t+1) = u(k+t+1|k)$ and so on.

The actual control inputs applied to the plant depends on the arrival of the control sequences sent by the controller and the buffer policy.

Let us represent the state of the buffer at a given time instant k as $b(k) \in \mathbb{R}^{m_1 N}$ and denote

$$\hat{k} = max\{k-l : \tau(k-l) = l\}$$

It easy to see that $\tau(k-l) = l$ indicates that the optimal control sequence computed in $k-l$, that is $U(k-l)$, arrives at time k to the buffer. Then, the dynamics of the buffer can be expressed as the recursive rule:

$$b(k) = \alpha(\mathscr{T}(k))U(\hat{k}) + (1 - \alpha(\mathscr{T}(k))\tilde{S}b(k-1) \tag{5.15}$$

where $\tilde{S} \in \mathbb{R}^{m_1 N \times m_1 N}$ is a shift matrix defined as the block matrix:

$$\tilde{S} = \begin{bmatrix} 0_{m_1\times m_1} & I_{m_1} & 0_{m_1\times m_1} & 0_{m_1\times m_1} & 0_{m_1\times m_1} & \cdots & 0_{m_1\times m_1} \\ 0_{m_1\times m_1} & 0_{m_1\times m_1} & I_{m_1} & 0_{m_1\times m_1} & 0_{m_1\times m_1} & \cdots & 0_{m_1\times m_1} \\ 0_{m_1\times m_1} & 0_{m_1\times m_1} & \cdots & I_{m_1} & 0_{m_1\times m_1} & \cdots & 0_{m_1\times m_1} \\ 0_{m_1\times m_1} & 0_{m_1\times m_1} & 0_{m_1\times m_1} & \cdots & I_{m_1} & \cdots & 0_{m_1\times m_1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0_{m_1\times m_1} & 0_{m_1\times m_1} & 0_{m_1\times m_1} & \cdots & \cdots & I_{m_1} & 0_{m_1\times m_1} \\ 0_{m_1\times m_1} & 0_{m_1\times m_1} & 0_{m_1\times m_1} & \cdots & \cdots & 0_{m_1\times m_1} & 0_{m_1\times m_1} \end{bmatrix}$$

In (5.15), $\alpha(\mathscr{T}(k)) \in \{0,1\}$ is a signal accounting for reception of control sequences at the buffer computed by the controller subsequent to those received before, such that:

$$\alpha(k) = \begin{cases} 1 & \text{if} \quad \hat{k} \in \{k, k-1, ..., k-\tau^{max}\} \\ 0 & \text{if} \quad \hat{k} \equiv \emptyset \end{cases}$$

With this description the control action $u(k)$ provided by the buffer at instant k can be expressed as

$$u(k) = \begin{bmatrix} I_{m_1} & 0_{m_1} & \dots & 0_{m_1} \end{bmatrix} b(k) \tag{5.16}$$

From equations (5.9)-(5.10) and (5.16) one can easily see that the state of the buffer is involved in the state of the NCS. However, the controller does not have access to the state of the buffer at any time k entailing a non standard MPC problem. Every sampling time, the controller has access to the plant states $x(k)$ and finds a finite horizon optimal control sequence $U(k) \in \mathbb{R}^{m_1 N}$ by solving the following optimization problem:

$$\min_{U(k)\in\mathbb{R}^{m_1 N}} \mathbb{E}\{V(x(k), \mathscr{U}_d(k), \mathscr{T}(k), U(k) | x(k), \mathscr{U}_d(k), \mathscr{T}(k))\} \tag{5.17}$$

where expectation is taken with respect to the discrete distribution of $\mathscr{T}(k)$. This can be done by explicit enumeration of the realization of $\mathscr{T}$ weighting all these realization with the corresponding probability.

As a consequence of *Assumption 1*, the minimization problem becomes:

$$\min_{U(k)\in\mathbb{R}_1^m} \sum_{i\in\mathbb{N}_0}^{\infty} p_i V(x(k), \mathscr{U}_d(k-i), i, U(k)) \tag{5.18}$$

Assuming this setup, it will be next illustrated how this stochastic predictive controller combined with a buffer provides robustness to packet delays and dropouts.

5.2.2 Simulation results

In this section the control strategy described above is applied to the following unstable system:

$$x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} w(k)$$

Delays are discrete uniformly distributed between 0 and 4, while the disturbance are random bounded disturbances with $|w(k)| < 0.5$.

The results obtained applying the proposed method in this section will be compared with the results from the method described in [D. Quevedo, J. Østergaard and D. Nešić, 2011], assuming no quantization issues.

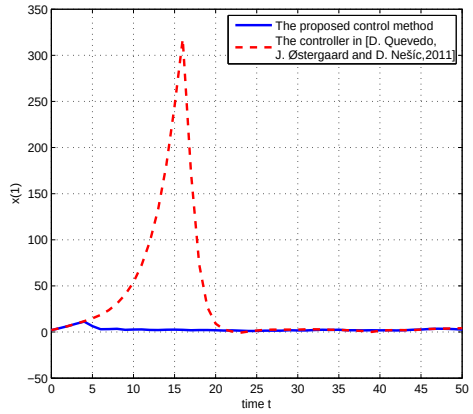

Figure 5.9: x(1) evolution with N=15

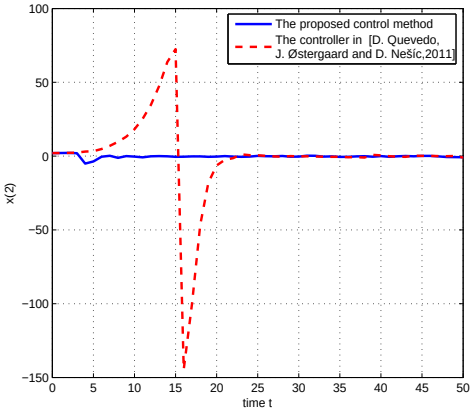

Figure 5.10: x(2) evolution with N=15

Figures 5.9 and 5.10 represent the evolution of the state by using the approach described in this paper and also the one in [D. Quevedo, J. Østergaard and D. Nešić, 2011]. It can be seen that the proposed controller improves the results because it takes into account in the objective function the different delay realizations.

In Figures 5.11 and 5.12 are shown the values of function V_t, which is defined in the following: $V_t = \sum_{i=0}^{ts} l(x(i), u(i))$, where $t_S = 100s$ is the simulation time. This funtion is represented with differents values of the control horizon and the initial value of x. In both figures it is possible to see how the value of V_t decreases with larger control horizons, as well when $x(0)$ is decreased,

$$x(0) = x_0 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

being x_0 a scalar that gives the initial state and will be varying in Figure 5.11.

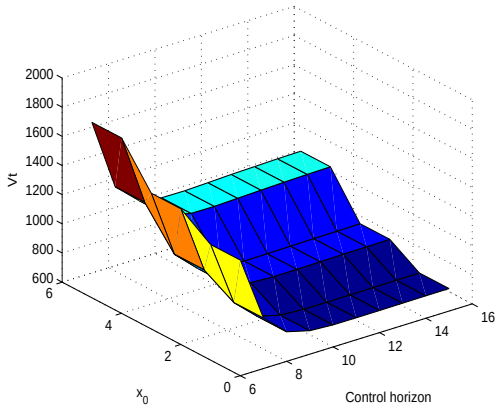

Figure 5.11: V_t evolution with the proposed control method.

5.2.3 Experimental results

This section presents an application of the proposed scheme to test its performance in a real system. The plant is the same one described in section 4.5.1 of Chapter 4. In this section, the experimental setup is described, providing all the considerations related to the scheme. After this, experimental results are presented.

Plant description

The plant, as in Chapter 4 is a variant of the quadruple-tank process, originally proposed in [Johansson, 2000], see [FeedBack, 2012]. A picture of the platform is given in Figure 5.13.

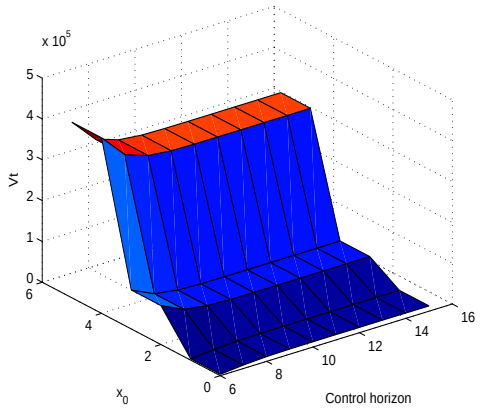

Figure 5.12: V_t evolution with the controller in [D. Quevedo, J. Østergaard and D. Nešić, 2011]

This educational plant is a model of a fragment of a chemical plant. It has 4 tanks, each one with a pressure sensor to measure the water level. The couplings between the tanks can be modified using seven manual valves to change the dynamics of the system.

For the experiments, the chosen configuration is the same than in section 4.5.1 of Chapter 4 (see Figure 4.12).

The objective of the experiments is twofold. First, the state of the plant must be monitored from every tank. Secondly, the water level of the two lower tanks is to be controlled.

Plant modelling

The non-linear and linearized model were described in Chapter 4.

The objective here is not only to stabilize the plant around the linearization point, but also to track references. In order to do that, the output of the system is set as, $s \triangleq C_r \Delta h$, where C_r is a matrix that selects the water level of tanks 2 and 4. The references are given by the vector r. To perform the tracking task, the equilibrium points $(\Delta h_r, \Delta v_r)$ associated with references r are found as follows.

$$\Delta \dot{h}_r = 0 = A\Delta h_r + B\Delta v_r,$$

$$s = r = C_z \Delta h_r.$$

Rewriting in blocks the equation above yields

$$\begin{bmatrix} 0 \\ r \end{bmatrix} = \begin{bmatrix} A & B \\ C_z & 0 \end{bmatrix} \begin{bmatrix} \Delta h_r \\ \Delta v_r \end{bmatrix},$$

Figure 5.13: Plant of four coupled tanks.

so that the equilibrium point associated with r can be easily obtained

$$\begin{bmatrix} \Delta h_r \\ \Delta v_r \end{bmatrix} = \begin{bmatrix} A & B \\ C_z & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ r \end{bmatrix}.$$

It is assumed that the references are reachable by the system, that is, the inverse does exist. Finally, to track references, the following system must be stabilized.

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{5.19}$$

where $x(t) \triangleq \Delta h(t) - \Delta h_r$ and $u(t) \triangleq \Delta v(t) - \Delta v_r$.

Results

In this section the experiments results are presented, using the described plant.

Delays are discrete uniformly distributed between 0 and 4, while the disturbance are random bounded disturbances with $|w(k)| < 0.5$.

	Value	Unit	Description
h_i	0-25	cm	Water level of tank i
v_i	0-5	V	Voltage level of pump i
A	0.01389	m^2	Cross-sectional area
a_i	50.265e-6	m^2	Outlet area of tank i
a_{13}	50.265e-6	m^2	Outlet area between tanks 1 and 3
η	0.22	$\frac{cm}{V.s}$	Contant relating voltage and flow
h_1^0	9.55 (12.6)	cm	Reference level of tank 1
h_2^0	16.9 (12.6)	cm	Reference level of tank 2
h_3^0	7.6 (11)	cm	Reference level of tank 3
h_4^0	14.1 (11)	cm	Reference level of tank 4
v_1^0	3.3 (3.5)	cm	Voltage level of pump 1
v_2^0	2.6 (1.5)	cm	Voltage level of pump 2

Table 5.1: Parameters of the plant. The terms in parentheses are related to the simulation experiments.

Fig. 5.14 and 5.15 show the outputs of tanks 2 and 4, with their respective references. Fig. 5.14 compares the performance of a classical MPC when the network under consideration is perfect and when it introduces dropouts. It can be seen how the dropouts make the performance much worse.

Fig. 5.15 considers the network with dropouts. It compares the classical MPC with the stochastic MPC presented in this chapter. It can be seen how the proposed stochastic MPC improves the perfomance.

5.3 Conclusions

This chapter has presented two model predictive control strategies in order to deal with time-delays and packet dropouts introduced by a communication network.

The first approach has presented a methodology to compensate for data dropouts and delays in networked control systems. The methodology takes advantage of the intrinsic computation of future control signals in predictive control, to cope with eventual data dropouts. A key aspect is the inclusion of a buffering strategy together with a model based plant estimator that, under certain conditions, ensure stability of the controlled system.

Simulation results show that remarkable data dropout rates up to 40% can be achieved without significant performance degradation, as well as traffic load alleviation up to 85% with respect to conventional buffered predictive control systems.

In the second approach, a stochastical model predictive controller has been designed, showing how statistical information on packet delays and dropouts can be used in the

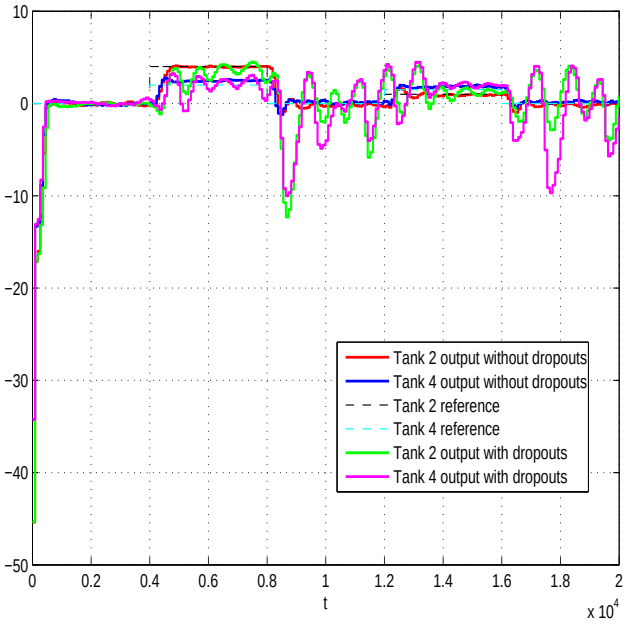

Figure 5.14: Tanks 2 and 4 outputs with a classical MPC

design of a networked control system. Also some simulations have been presented.

Future works may include studying closed loop stability and performance issues.

5.4 Related publications

- Isabel Jurado, Daniel E. Quevedo, Pablo Millán, Francisco R. Rubio. *Stochastic packetized model predictive control for networked control systems subjects to time-delays and dropouts*. 20th International Symposium on Mathematical Theory of Networks and Systems, Melbourne (Australia), 2012.

- Pablo Millán, Isabel Jurado, Carlos Vivas and Francisco R. Rubio. *Networked predictive control of systems with large data dropouts*. 47th IEEE Conference on Decision and Control (CDC'08), Cancun (Mexico), 2008.

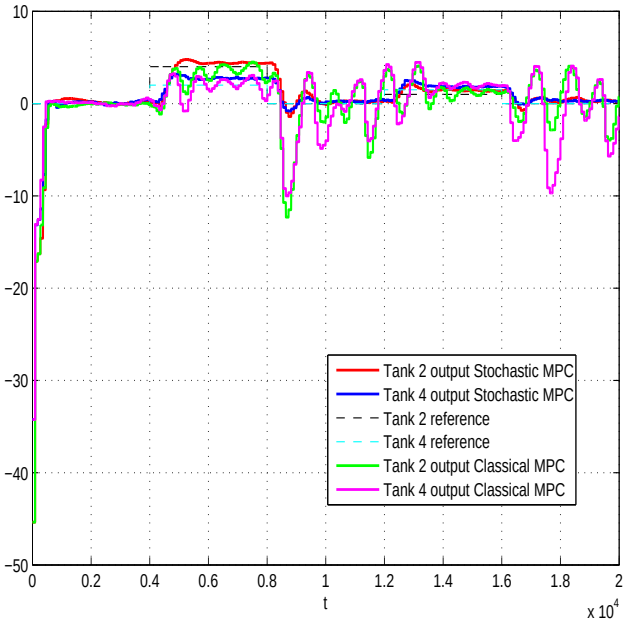

Figure 5.15: Tanks 2 and 4 outputs with a classical MPC and the presented stochastic MPC

Chapter 6

Dynamic Controller Placement for Networked Control Systems

This chapter presents two algorithms for Networked Control Systems with multiple wireless nodes. Communication between them is affected by random packet dropouts. Two algorithms are presented to decide which nodes will calculate the control input and which will only relay data. These algorithms make the architecture flexible by adapting the control to the possible changes in the network conditions.

Wireless sensor-actuator networks offer flexibility in the design of networked control systems. One novel element when using networks with multiple nodes is that the role of individual nodes does not need to be fixed. Stochastic models for transmission outcomes and characterize the distribution of controller location and the covariance of system states are adopted. Simulation results illustrate that the proposed architecture has the potential to give significantly better performance than limiting control calculations to be carried out at a fixed node.

In the first section of this chapter, the network is composed of a certain number of nodes in matrix formation. These nodes follow an algorithm, that decides which node will calculate the control input. This node will solve a cooperative MPC communicating with one of its closet neighbors. A survey in NCSs, dealing as well with interactions between network components is presented in [J. A. Giraldo and N. Quijano, 2011]. In the topic of coopera-

tive NCS, [R. Olfati-Saber, J. A. Fax and R. M. Murray, 2007] introduced some consensus protocols, where the main objective is to reach an agreement regarding an interaction rule that specifies the information exchange between an agent and all of its neighbors. Furthermore, [F. Xiao and L. Wang, 2008] developed a consensus algorithm for discrete systems with delays. In [U. Münz, A. Papachristodoulou and Frank Allgöwer, 2008], the authors study the stability of multi-agent system formations with delayed exchange of information between the agents.

In the network under consideration in this work, the only node that receives the state of the plant without any failure probability is the sensor node, which is located next to the plant. The actuator node is directly connected to the plant input, therefore this data is received without problems. The actuator node is also the only node that provides transmission acknowledgments. It is assumed as well that there is an array of nodes between the sensor and the actuator nodes, as shown in Fig. 6.1. The nodes in a column send the information to the following column of nodes. Also, it is considered that the nodes, except the sensor and the actuator nodes, can communicate with its closest neighbors in the same column, and many thereby cooperate and exchange information. The sensor and actuator nodes cannot calculate control values, they only are able transmit information. The communication between nodes is limited and subject to dropouts.

It is supposed that the model of the plant is divided in two incomplete subsystems. This can be easily generalized for the case of any number of incomplet subsystems. Each node will know only a part of the model of the system, that is why it has to collaborate with its neighbors, which know the other part of the system. Therefore, each node will estimate just a part of the state.

The control policy to be used will be a cooperative MPC. Within this context, a flexible NCS architecture where the role played by cooperative nodes depends upon transmission outcomes and their acknowledgments is presented. With the algorithm proposed, trans-

mission outcomes and their acknowledgments will determine, at each instant, whether the control input will be calculated at the actuator node, or closer to the sensor node.

An example of an application for this algorithm can be a situation where the network of nodes have moving obstacles in between, difficulting the communication. This way, depending on the position of the obstacles, the best pair of nodes to calculate the control value is varying.

In the second section of this chapter, a flexible architecture for a single-loop NCS topology using multiple nodes connected in series over analog erasure channels is presented. The control architecture proposed adapts to changes in network conditions, by allowing the role played by individual nodes to depend upon transmission outcomes. The proposed algorithm will decide which nodes assume estimator functions and which ones merely relay data.

In this last section it is assumed that the control policy consists of a pre-designed state feedback-gain combined with a state observer, which, in the absence of network effect, would lead to the desired performance. Within this context, a flexible NCS architecture where the role played by individual nodes depends upon transmission outcomes is presented. With the algorithm proposed, transmission outcomes determine, at each instant, whether the state estimation will be calculated at the actuator node, at the sensor node or at one of the intermediate nodes. It turns out that, if individual dropout processes are i.i.d., then the estimator location has a stationary distribution, which can be easily characterized. To analyze the performance of the dynamic NCS architecture in the presence of correlated dropouts, a jump-linear system model is derived and the network model recently introduced in [Quevedo et al., 2011] is adopted.

The first section of the chapter is a extension of the conference contribution [D. E. Quevedo, K. H. Johansson, A. Ahlén and I. Jurado, 2012] to consider NCSs with parallel links and the use of cooperative MPC. Also, in the second section, the contribution [D.

E. Quevedo, K. H. Johansson, A. Ahlén and I. Jurado, 2012] is extended by allowing the nodes to transmit their state estimates.

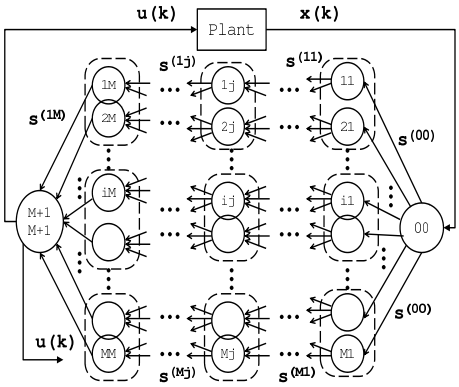

Figure 6.1: Control over a graph with dropouts and unreliable acknowledgments of actuator values.

6.1 Cooperative MPC for Networked Control Systems

6.1.1 NCS Setup

A MIMO LTI plant model of the following form is considered

$$x(k+1) = Ax(k) + Bu(k) + w(k) \tag{6.1}$$

where $x(0) \sim \mathcal{N}(0, P_0)$, $P_0 > 0$. In (6.1), $u(k) \in \mathbb{R}^{m_1}$ is the plant input, $x(k) \in \mathbb{R}^n$ is the state, and $w(k) \sim \mathcal{N}(0, D)$, $D > 0$ is driving noise.

The model described in (6.1) represents the whole plant. But, as foreshadowed in the introduction, individual nodes do not have knowlegde of this whole model. Thus, nodes have to interact with their neighbors to get all the information about the plant. It will be considered that, between two neighbor nodes, they have all the information about the plant model.

The composite model

For each node, the *composite model (CM)* [A. N. Venkat, J. B. Rawlings and S. J. Wright, 2007], is the combination of the decentralized model and all the interaction models (in the present work, it will be considered that the cooperation is done pair wise between two nodes in the same column, i. e., one interaction model). The decentralized state vector in node (ij), $x_{(ij)}$, is augmented with the state from the influence of the neighbor node (i^*j).

Therefore, the augmented state $x_{(ij)} = [x^T_{(ij)(ij)}, x^T_{(ij)(i^*j)}]^T$ represents the CM states for the node (ij), (i^*j) being the neighbor node interacting, pairwise, with (ij), which is in the same column and $i^* \in \{(i-1),(i+1)\}$. In this augmented state, $x_{(}ij)(i^*j)$ is the influence of the node (i^*j) on the node (ij), and $x_{(}ij)(ij)$ is the part of the state that take into account just the part of the model that the node (ij) knows, so it is a decentralized state. In this case, the CM for the node (ij) is written as

$$x_{(ij)}(k+1) = A_{(ij)}x_{(ij)}(k) + B_{(ij)}u_{(ij)}(k) + W_{(ij)(i^*j)}u_{(i^*j)}(k) \tag{6.2}$$

where

$$A_{(ij)} = \begin{bmatrix} A_{(ij)(ij)} & \\ & A_{(ij)(i^*j)} \end{bmatrix}, \quad B_{(ij)} = \begin{bmatrix} B_{(ij)(ij)} \\ \mathbf{0} \end{bmatrix}, \quad W_{(ij)(i^*j)} = \begin{bmatrix} \mathbf{0} \\ B_{(ij)(i^*j)} \end{bmatrix}.$$

The interactive matrices are written depending on i^* because

$$A_{(ij)((i-1)j)} = A_{(ij)((i+1)j)} \quad \textit{and} \quad B_{(ij)((i-1)j)} = B_{(ij)((i+1)j)},$$

since between two neighbor cooperative nodes there is the whole information about the system. Therefore, the node (ij) has one part of the model, and the neighbor nodes $((i-1)j)$ and $((i+1)j)$ have the other part of the model. This could be extended to the case in where the global state vector is divided in more than two parts, so the cooperation between all the nodes that have access to different information about the system will have to be considered.

The augmented control signal is denoted as

$$u_{(ij)}^{(i^*j)}(k) = [u_{(ij)}(k)^T, u_{(i^*j)}(k)^T]^T, \quad k \in \mathbb{N}_0, \tag{6.3}$$

and will be calculated with MPC techniques employing pair wise cooperation among the neighboring nodes. Thus,

$$u_{(ij)}^{(i^*j)}(k) = Cooperative \quad MPC\left(x_{(ij)}(k), x_{(i^*j)}(k)\right), \quad k \in \mathbb{N}_0, \tag{6.4}$$

where $x_{(ij)}(k)$ and $x_{(i^*j)}(k)$ represent the CM states for the nodes (ij) and (i^*j), respectively.

Network issues

Sensor and actuator nodes are connected via a wireless network, characterised via a graph having $M \times M + 2$ nodes, see Fig. 6.1. Control values cannot be calculated by the sensor nor the actuator nodes, they are just used to measure the plant state and apply the control signal, respectively. Therefore, according to Fig. 6.1, the network has $M \times M$ nodes that could act as the controller. Transmissions are done in sequential manner as shown in Fig. 6.2. More precisely, the packet $s^{(ij)}(k)$ is transmitted from node (ij) to its closest neighbors, these are $(i+1)(j+1)$, $i(j+1)$ and $(i-1)(j+1)$, at times $kT + i\tau$, where T is the sampling period and $\tau \ll T/(M+1)$ refers to the times between transmissions of packets. The plant input $u(k)$ is applied at time $kT + (M+1)\tau$. It is assumed that in-network processing is much faster than the plant dynamics (6.1) and, as in, e.g., [C. L. Robinson and P. R. Kumar, 2008], neglect delays introduced by the network.

A distinguishing characteristic of the situation at hand is that (due to channel fading) the network introduces stochastic packet dropouts. To study the situation, an analog erasure channel model is adopted and the binary success random processes is introduced

$$\gamma_{(ij)}^{(i(j-1))}(k) \in \{0,1\}, \quad k \in \mathbb{N}_0,\ i \in \{0,1,2,\ldots,M+1\}, j \in \{0,1,2,\ldots,M+1\}$$

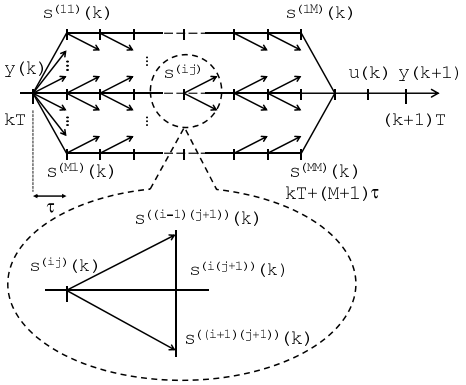

Figure 6.2: Transmission Schedule; $t \in \mathbb{R}_{\geq 0}$ is actual time.

where $\gamma_{(ij)}^{(i(j-1))}(k) = 1$ indicates that transmission of the packet $s^{(i(j-1))}(k)$ from node $(i(j-1))$ to node (ij) at time $kT + (j+1)\tau$, is successful, i.e., error-free; $\gamma_{(ij)}^{(i(j-1))}(k) = 0$ refers to a packet-dropout. Throughout this work it is assumed that the sensor node (00) has direct access to plant output measurements. For notational convenience, $\gamma^{(00)}(k) = 1$, for all $k \in \mathbb{N}_0$.

To save energy, in this formulation the wireless nodes (ij) $\forall i \in \{0,1,2,\ldots,M\}$ and $\forall j \in \{0,1,2,\ldots,M\}$ do not provide acknowledgments of receipt of the packets. However, the actuator node, $((M+1)(M+1))$, will in general have less stringent energy constraints, so at time $kT + (M+1)\tau$ the control signal is received, at $kT + (M+2)\tau$ this control value is applied and at time $kT + (M+3)\tau$, the actuator broadcasts the control value applied, namely $u(k) = [u_{(ij)}^A(k)^T, u_{(i^*j)}^A(k)^T]^T$, back to the wireless nodes (ij) $\forall i \in \{0,1,2,\ldots,M\}$ and $\forall j \in \{0,1,2,\ldots,M\}$, and with $i^* \in \{(i-1),(i+1)\}$, see Fig. 6.1. This acknowledgment-like signal is unreliable and affected by dropouts with associated success processes

$$\delta^{(ij)}(k) \in \{0,1\}, \quad k \in \mathbb{N}_0,\ i \in \{0,1,2,\ldots,M\}, j \in \{0,1,2,\ldots,M\}.$$

More precisely, if $u(k)$ is successfully received at node (ij), then $\delta^{(ij)}(k) = 1$; see also [E. Garone, B. Sinopoli and A. Casavola, 2010] and [O. C. Imer and S. Yüksel and T.

Başar, 2006] for studies on the importance of acknowledgments in closed loop control. It is assumed that the actuator node has perfect knowledge of plant inputs, and thus, writes $\delta^{((M+1)(M+1))}(k) = 1, \forall k \in \mathbb{N}_0$.

Due to packet dropouts, plant state measurements are not always available at the actuator node. On the other hand, the sensor node will, in general, not have perfect information of previous plant inputs. This makes the implementation of (6.4) via estimated state feedback a challenging task. The main purpose of the present work is to investigate which nodes of the network (with the exception of the sensor and actuator nodes) should use their local state estimates to implement the control law (6.4). This approach will lead to a dynamic assignment of the role played by the individual network nodes. Which tasks are carried out by each node at each time instant, will depend upon transmission outcomes, i. e., on $\gamma_{(ij)}^{(i(j-1))}(k)$ and $\delta^{(ij)}(k)$.

A flexible NCS architecture

The packets transmitted by each node (ij) have three fields, namely, state measurements, tentative plant inputs (if available) and the value of the objective function under consideration:

$$s^{(ij)}(k) = \left(x(k), u_{(\alpha\beta)}^{(\alpha^*\beta)}(k), J(k)\right), \alpha \in \{1,\ldots,i\}, \alpha^* \in \{(\alpha-1),(\alpha+1)\}, \beta \in \{1,\ldots,j\}. \quad (6.5)$$

The plant states $x(k)$ includes the two components corresponding to the cooperation nodes, that is $x(k) = [x_{(ij)}(k)^T, x_{(i^*j)}(k)^T]$, with $i^* \in \{(i-1),(i+1)\}$.

The control signal $u(k)_{(\alpha\beta)}^{(\alpha^*\beta)}$ in (6.5) with the structure shown in (6.3), is the plant input which is applied at the plant provided the packet $s^{(ij)}(k)$ is delivered at the actuator node. If $s^{(ij)}(k)$ is lost, then by following Algorithm 2, which will be described in Section 6.1.4, the plant input will be provided by one of the nodes in subsequent columns, see Fig. 6.1, which thereby takes on the controller role at time k. For further reference, the node which calculates the plant input at time k will be denoted as

$$c(k) \in \{(11),(12),\ldots,(1M),(21),\ldots,(2M),(M1),\ldots,(MM)\}.$$

Example

Consider the network in Fig. 6.3. Some nodes in the network will have one part of the plant model and the other nodes will have the other part. Therefore the information will be distributed like this: (11), (31), (12), (32), (13) and (33) will have the same information about the plant, that is

$$A_{(11)} = A_{(31)} = A_{(12)} = A_{(33)} = A_{(13)} = A_{(33)}, \quad B_{(11)} = B_{(31)} = B_{(12)} = B_{(33)} = B_{(13)} = B_{(33)}$$

and

$$W_{(11)(21)} = W_{(31)(21)} = W_{(12)(22)} = W_{(32)(22)} = W_{(13)(23)} = W_{(33)(23)}.$$

On the other hand, the rest of the nodes will have the other part of the information about the plant:

$$A_{(21)} = A_{(22)} = A_{(23)}, \quad B_{(21)} = B_{(22)} = B_{(23)}$$

and

$$W_{(21)(11)} = W_{(21)(31)} = W_{(22)(12)} = W_{(22)(32)} = W_{(23)(13)} = W_{(23)(33)}.$$

Moreover, the cooperating couples are:

$$(11) \longleftrightarrow (21), \quad (12) \longleftrightarrow (22),$$
$$(13) \longleftrightarrow (23), \quad (31) \longleftrightarrow (21),$$
$$(32) \longleftrightarrow (22), \quad \text{and} \quad (33) \longleftrightarrow (23).$$

So, it is easy to see that:

$$A_{(11)(11)} = A_{(12)(12)} = A_{(13)(13)} = A_{(31)(31)} = A_{(32)(32)} = A_{(33)(33)},$$
$$B_{(11)(11)} = B_{(12)(12)} = B_{(13)(13)} = B_{(31)(31)} = B_{(32)(32)} = B_{(33)(33)},$$
$$A_{(21)(21)} = A_{(22)(22)} = A_{(23)(23)}, \quad B_{(21)(21)} = B_{(22)(22)} = B_{(23)(23)},$$

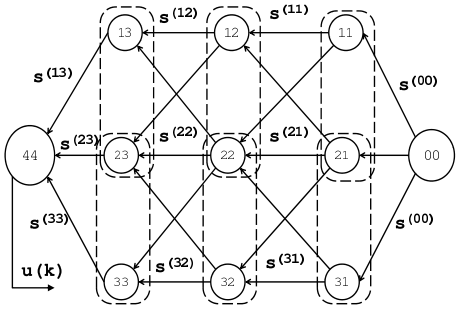

Figure 6.3: Graph with $3 \times 3 + 2$ nodes.

$$A_{(11)(21)} = A_{(31)(21)} = A_{(12)(22)} = A_{(32)(22)} = A_{(13)(23)} = A_{(33)(23)},$$
$$B_{(11)(21)} = B_{(31)(21)} = B_{(12)(22)} = B_{(32)(22)} = B_{(13)(23)} = B_{(33)(23)},$$
$$A_{(21)(11)} = A_{(21)(31)} = A_{(22)(12)} = A_{(22)(32)} = A_{(23)(13)} = A_{(23)(33)}$$

and

$$B_{(21)(11)} = B_{(21)(31)} = B_{(22)(12)} = B_{(22)(32)} = B_{(23)(13)} = B_{(23)(33)}.$$

In this example the numerical values are:

$$A_{(11)} = \begin{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & \begin{bmatrix} 0.1 & 0 \\ 0 & 0.2 \end{bmatrix} \end{bmatrix}, \quad A_{(21)} = \begin{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & 0.1 \end{bmatrix} \end{bmatrix},$$

$$B_{(11)} = \begin{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{bmatrix}, \quad B_{(21)} = \begin{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{bmatrix},$$

$$W_{(11)(21)} = \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0.4 \end{bmatrix} \end{bmatrix}, \quad W_{(21)(11)} = \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{bmatrix}.$$

So the decentralized models are:

$$A_{(11)(11)} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad A_{(21)(21)} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},$$

$$B_{(11)(11)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad B_{(21)(21)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

whereas the interacting models are given by:

$$A_{(11)(21)} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.2 \end{bmatrix}, \quad A_{(21)(11)} = \begin{bmatrix} 0 & 0 \\ 0 & 0.1 \end{bmatrix},$$

$$B_{(11)(21)} = \begin{bmatrix} 0 \\ 0.4 \end{bmatrix}, \quad B_{(21)(11)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

6.1.2 Control Implementation

To implement the control law (6.4) over the network using packets of the form (6.5), a cooperative MPC will be used.

In this work it is assumed that pairs of neighbor nodes in the same column can exchange information. In Algorithm 1 the cooperation between a pair of nodes is shown. Then, with the CM in (6.2) and the Algorithm 1, it is possible to calculate a *Feasible Cooperation-Based MPC (FC-MPC)*, that is well explained in [A. N. Venkat, J. B. Rawlings and S. J. Wright, 2007].

The calculation of the suboptimal control input, $u^{*p}_{(ij)}$, for each iteration p, is performed by solving the FC-MPC problem. So, the objective function will be chosen as a linear combination of the individual nodes' objectives, i.e.,

$$J_{(ij)} = J_{(i^*j)} = \varpi_{(ij)} V_{(ij)} + \varpi_{(i^*j)} V_{(i^*j)}, \quad \varpi_{(ij)}, \varpi_{(i^*j)} > 0, \varpi_{(ij)} + \varpi_{(i^*j)} = 1, i^* \in \{(i-1),(i+1)\}$$

The local objective for each cooperative node is depending on the value of $\gamma^{((i-1)(j-1))}_{(ij)}(k)$, $\gamma^{((i)(j-1))}_{(ij)}(k)$ and $\gamma^{((i+1)(j-1))}_{(ij)}(k)$. If any of them is equal to 1, the cost function will be:

$$J_{(ij)} = V_{(ij)}(x'^{p}_{(ij)}(k), u'^{p}_{(ij)}(k); x_{(ij)}(k)) = \sum_{t=k}^{\infty} x^{p}_{(ij)}(t|k)^T Q x^{p}_{(ij)}(t|k) + u^{p}_{(ij)}(t|k)^T R u^{p}_{(ij)}(t|k), \quad (6.6)$$

where

$$x'^{p}_{(ij)}(k) = [x^{p}_{(ij)}(k+1|k)^T, x^{p}_{(ij)}(k+2|k)^T, ...]^T,$$

$$u'^{p}_{(ij)}(k) = [u^{p}_{(ij)}(k|k)^T, u^{p}_{(ij)}(k+1|k)^T, ...]^T,$$

and $Q > 0$ and $R > 0$ are weighting matrices. To calculate these predictions the CM for the node (ij) (6.2) has been used.

The notation p indicates the iteration number. During each MPC optimization, the state and input trajectories ($x'_{(i^*j)}(k)$, $u'_{(i^*j)}(k)$) of the interacting node MPC are not updated, so they remain at ($x'^{p-1}_{(i^*j)}(k)$, $u'^{p-1}_{(i^*j)}(k)$).

On the other hand, if $\gamma^{((i-1)(j-1))}_{(ij)}(k) = \gamma^{((i)(j-1))}_{(ij)}(k) = \gamma^{((i+1)(j-1))}_{(ij)}(k) = 0$, no information about the state has arrived at node (ij), so an estimation is used. Therefore, the cost

function will be an expected value ([K. J. Åström, 2006]):

$$J_{(ij)} = \sum_{t=k}^{\infty} \hat{x}^{p}_{(ij)}(t|k)^{T} Q \hat{x}^{p}_{(ij)}(t|k) + \mathbf{tr}\big(QP_{(ij)}(k)\big) + u^{p}_{(ij)}(t|k)^{T} R u^{p}_{(ij)}(t|k), \tag{6.7}$$

where $P_{(ij)}(k)$ approximates the covariance of $x_{(ij)}(k)$ and is calculated as follows:

$$P_{(ij)}(k+1) = A_{(ij)}\big(\mathbf{I} - \Gamma^{(ij)}(k)\mathbf{I}\big)P^{(ij)}(k)A^{T}_{(ij)} + D, \tag{6.8}$$

where

$$\Gamma^{(ij)}(k) = \alpha \in \{0,1,\ldots,i-1\}\beta \in \{0,1,\ldots,i-1\}$$

$\gamma^{(ig^{*}\,jg^{*})}_{(\alpha\beta)}(k)$ is equal to 1 if and only if $x_{(ij)}(k)$ is available at node (ij) at time $kT+(j-1)\tau$, and $(ig^{*}\,jg^{*})$ will be one of the preceding nodes of (ij) and will be the one that provides the best (the smallest or unique) value of $J_{(ij)}(k)$.

Remark. The objective function $J_{(ij)}$ is an approximation of:

$$J_{(ij)} \approx \mathbf{E}\{V_{(ij)}(\hat{x}^{'p}_{(ij)}(k), u^{'p}_{(ij)}(k); \hat{x}_{(ij)}(k))\},$$

since $P^{(ij)}(k)$ is not the covariance of $x_{(ij)}(k)$, but just an approximate value. Therefore the term $\mathbf{tr}\big(QP_{(ij)}(k)\big)$ ([K. J. Åström, 2006]) is not exact.

The following notation is used for simplicity: $x_{(ij)} \equiv x_{(ij)}(k)$ and $u_{(ij)} \equiv u_{(ij)}(k)$.

In the Algorithm 1, the state sequence generated by the input sequence $u_{(ij)}$ and initial state $x_{(ij)}$ has been represented by $x^{(u_{(ij)};x_{(ij)})}_{(ij)}$. Also, the notation $\hat{x}$ is representing $[\hat{x}^{T}_{(ij)}, \hat{x}^{T}_{(i^{*}j)}]^{T}$.

Algorithm 1 Cooperative MPC algorithm

Given $(\bar{u}^0_{(\iota j)}, x_{(\iota j)}(k))$, $Q>0$, $R>0$, $\iota \in \{i, i^*\}$, $i^* \in \{(i-1),(i+1)\}$, $p_{max}(k) \geq 0$, $\varepsilon > 0$, $p \leftarrow 1$, $e_{(\iota j)} \leftarrow \Phi$ and $\Phi >> 1$. {being ε a bound for $e_{(\iota j)}$}

while $e_{(\iota j)} > \varepsilon$ for some $\iota \in \{i, i^*\}$ and $p \leq p_{max}(k)$ **do** $\forall \iota \in \{i, i^*\}$

$u^{*p}_{(\iota j)} \in \arg\min_{(\iota j)} \quad FC-MPC_{(\iota j)}$

$u^p_{(\iota j)} = \varpi_{(\iota j)} u^{*p}_{(\iota j)} + (1-\varpi_{(\iota j)}) u^{p-1}_{(\iota j)}$

$e_{(\iota j)} = \left\| u^p_{(\iota j)} - u^{p-1}_{(\iota j)} \right\|$

for each $\iota \in \{i, i^*\}$ **do**

The node (ιj) transmits $u^p_{(\iota j)}$ to its neighbor

end for

if $\gamma^{((i-1)(j-1))}_{(ij)}(k) = 1 \vee \gamma^{(i(j-1))}_{(ij)}(k) = 1 \vee \gamma^{((i+1)(j-1))}_{(ij)}(k) = 1$ **then**

$x^p_{(\iota j)} \leftarrow x^{(\bar{u}^p_{(ij)}, \bar{u}^p_{(i^* j)}; x_{(\iota j)})}_{(\iota j)}$, $\forall \iota \in \{i, i^*\}$

else

$\hat{x}^p_{(\iota j)} \leftarrow \hat{x}^{(\bar{u}^p_{(ij)}, \bar{u}^p_{(i^* j)}; \hat{x}_{(\iota j)})}_{(\iota j)}$, $\forall \iota \in \{i, i^*\}$

end if

$p \leftarrow p+1$

end while

Due to the communications constraints, the maximum number of iterations p_{max} is limited. It is also possible to loose information during the cooperation. For these reasons only a suboptimal control input $\mathbf{u}^{*p}_{(ij)}$ will be available.

Notice that $i^* \in \{(i-1),(i+1)\}$, that means that the node (ij) can communicate with the nodes $((i-1)j)$ and $((i+1)j)$, see Fig 6.4. Therefore, (ij) will solve two cooperative MPC and will have two control values. The control value that the node (ij) will transmit will be the one that provides the lowest cost.

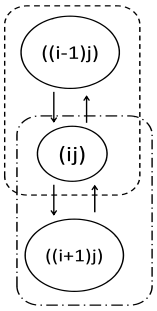

Figure 6.4: Cooperative nodes for node (ij).

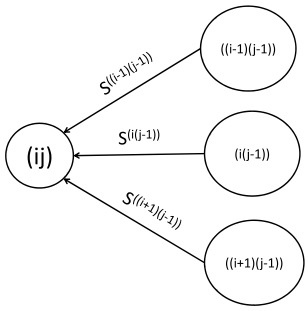

Figure 6.5: Packets received by node (ij).

6.1.3 State Estimation

While only the node $c(k)$, will provide the plant input at instant k, in the present formulation all nodes compute local state estimates, $\hat{x}_{(ij)}(k)$, by using the data received from one of the preceding nodes, $(ig^* jg^*)$. This serves as safeguard for instances when the loop is broken due to dropouts.

Since the nodes do not have full information about the plant, they are only able to calculate a part of the state. That means that $\hat{x}_{(ij)}(k)$ is not an estimate of the global state of the plant.

In the following, it will be assumed that the acknowledgments of plant inputs are *quite reliable*. Thus, the state estimates are simply calculated as:

$$\hat{x}_{(ij)}(k) = A_{(ij)}\hat{x}_{(ij)}(k-1) + B_{(ij)}u_{(ij)}(k-1) + W_{(ij)(i^*j)}u_{(i^*j)}(k-1) + \tag{6.9}$$

$$K^{(ij)}(k)\big(x_{(ij)}(k) - (A_{(ij)}\hat{x}_{(ij)}(k-1) + B_{(ij)}u_{(ij)}(k-1) + W_{(ij)(i^*j)}u_{(i^*j)}(k-1))\big), \tag{6.10}$$

where $K^{(ij)}(k) = \Gamma^{(ij)}(k)\mathbf{I}$.

In (6.9), $u_{(ij)}(k-1)$ and $u_{(i^*j)}(k-1)$ are local plant input estimates. In particular, if $\delta^{(ij)}(k-1) = 1$, then $[u^A_{(ij)}(k-1)^T, u^A_{(i^*j)}(k-1)^T]^T = u(k-1)$, and $u_{(ij)}(k-1) = u^A_{(ij)}(k-1)$ and $u_{(i^*j)}(k-1) = u^A_{(i^*j)}(k-1)$. On the other hand, at instances where $\delta^{(ij)}(k-1) = 0$, node (ij) uses the tentative plant input value transmitted in the second field of the previous packet $s^{(ij)}(k-1)$ (if non-empty), or otherwise sets $u_{(ij)}(k-1)$ and $u_{(i^*j)}(k-1)$ as per (6.4), see Algorithm 2.

Intuitively, good control performance will be achieved if the state estimation is accurate. Clearly, nodes which are closer to the sensor will have access to more output measurements, see Fig. 6.1. On the other hand, one can expect that nodes which are physically located closer to the actuator node will on average receive more plant input acknowledgments, thus, have better knowledge of plant inputs.

6.1.4 Algorithm for Dynamic Controller Placement

Algorithm 2 is run at every node (ij). Since it is assumed that acknowledgment from the actuator node is, in general, available, but transmissions of packets $s^{(ij)}(k)$ are less reliable, nodes closer to the sensor nodes can be expected to have better state estimates than nodes located further down the network. Therefore, preference is given to forward incoming tentative plant input values.

The sensor node $(ij) = (00)$ uses as input: $s^{(00)}(k) = (x(k), \emptyset, \emptyset), \quad \gamma_{(00)}(k) = 1$, where $\emptyset$ means that the field is empty.

This node just passes the information to all the nodes in the first column. The node (00), as the node $(M+1, M+1)$, cannot calculate control values.

The rest of the nodes in the network can only send information to their three closest neighbors in the following column, except for the lower and uppermost nodes who can only send to two neighbors, see Fig. 6.2. Therefore, the generic node (ij) can receive zero, one, two or three (if not border node) packets. In the case that it receives more than one packets (as shown in Fig. 6.5), it will have to choose between them. Then, the chosen one will be the one with the minimun value of the cost function J.

The first column of nodes, the nodes (ij) with $j = 1$, calculate control values cooperating between them per pairs, as explained in Section 6.1.2. And each node of that column transmits:

$$s^{(ij)}(k) = \left([x_{(ij)}(k)^T, x_{(i^* j)}(k)^T]^T, [u_{(ij)}(k)^T, u_{(i^* j)}(k)^T]^T, J_{(ij)}(k)\right)$$

to its three closest neighbors in the next column of nodes. Subsequent nodes then relay the

Algorithm 2 Dynamic Controller Placement

$k \leftarrow 0$, $\hat{x}_{(ij)}(0) \leftarrow 0$, $P_0^{(ij)} \leftarrow P_0$, $m \leftarrow 0$, $i^* = (i-1)$ or $i^* = (i+1)$, the cooperative nodes for (ij).

while $t \geq 0$ **do** {$t \in \mathbb{R}_{\geq 0}$ is actual time}

while $t \leq kT + m\tau$ **do** {wait-loop}

$m \leftarrow m+1$

end while

$P^{(ij)}(k+1) \leftarrow A_{(ij)}P^{(ij)}(k)A_{(ij)}^T + D$

if $\gamma_{(ij)}^{((i-1)(j-1))}(k) = 0 \wedge \gamma_{(ij)}^{(i(j-1))}(k) = 0 \wedge \gamma_{(ij)}^{((i+1)(j-1))}(k) = 0$ **then**

if $\delta^{(ij)}(k-1) = 1$ **then**

$u_{(ij)}(k), u_{(i^*j)}(k), J_{(ij)}(k) \leftarrow Algorithm \quad 1$

$s^{(ij)}(k) \leftarrow \left([x_{(ij)}(k)^T, x_{(i^*j)}(k)^T]^T, [u_{(ij)}(k)^T, u_{(i^*j)}(k)^T]^T, J_{(ij)}(k)\right)$ {a tentative input}

else

$s^{(ij)}(k) \leftarrow (\emptyset, \emptyset, \emptyset)$

end if

end if

if $\gamma_{(ij)}^{((i-1)(j-1))}(k) = 1 \vee \gamma_{(ij)}^{(i(j-1))}(k) = 1 \vee \gamma_{(ij)}^{((i+1)(j-1))}(k) = 1$ **then**

$S \leftarrow s^{((i-1)(j-1))}(k)$ and/or $s^{(i(j-1))}(k)$ and/or $s^{((i+1)(j-1))}(k)$ {S is a set containing all the packets arraived. If all the packets arrive, S will contain $s^{((i-1)(j-1))}(k)$, $s^{(i(j-1))}(k)$ and $s^{((i+1)(j-1))}(k)$}

$(x^S, u^S, J^S) \leftarrow \arg\min_l \quad J_l \in S$

if $x^S \neq \emptyset$ **then** {$x_{(ij)}(k)$ is available}

$\hat{x}_{(ij)}(k) \leftarrow x_{(ij)}^S$

$P^{(ij)}(k+1) \leftarrow D$

end if

if $u^S \neq \emptyset$ **then**

$u_{(ij)}(k) = u_{(ij)}^S$

$u_{(i^*j)}(k) = u_{(i^*j)}^S$

else
$u_{(ij)}(k), u_{(i^*j)}(k), J_{(ij)}(k) \leftarrow Algorithm \quad 1$
end if
if $u^S = \emptyset \wedge \delta^{(ij)}(k-1) = 1$ **then**
$s^{(ij)}(k) \leftarrow \left(x^S, [u_{(ij)}(k)^T, u_{(i^*j)}(k)^T]^T, J_{(ij)}(k)\right)$ {a tentative input}
else
$s^{(ij)}(k) \leftarrow (x^S, u^S, J^S)$
end if
end if
while $t < kT + (j+1)\tau$ **do** {wait-loop}
$m \leftarrow m+1$
end while
transmit $\mathbf{s}^{(ij)}(k)$
while $t \leq kT + (M+3)\tau$ **do** {wait-loop}
$m \leftarrow m+1$
end while
if $\delta^{(ij)}(k) = 1$ **then**
$\hat{x}_{(ij)}(k+1) \leftarrow A_{(ij)}\hat{x}_{(ij)}(k) + B_{(ij)}u^A_{(ij)}(k) + W_{(ij)(i^*j)}u^A_{(i^*j)}(k)$
else
$\hat{x}_{(ij)}(k+1) \leftarrow A_{(ij)}\hat{x}_{(ij)}(k) + B_{(ij)}u_{(ij)}(k) + W_{(ij)(i^*j)}u_{(i^*j)}(k)$
end if
$k \leftarrow k+1$
end while

arrived packets to the actuator node, choosing the ones with minimum $J_{(ij)}(k)$.

A new tentative control value has to be calculated only in the following cases:

- No packet has arrived from the previous column,

$$\gamma_{(ij)}^{((i-1)(j-1))}(k) = \gamma_{(ij)}^{(i(j-1))}(k) = \gamma_{(ij)}^{((i+1)(j-1))}(k) = 0,$$

  but the acknowledgment from the actuator has arrived, $\delta^{(ij)}(k-1) = 1$.

- At least one packet has arrived but all of them have the following structure:

$$s = (x, \emptyset, \emptyset),$$

that means that the state is available but there is no information about the control.

Then $u_{(ij)}(k)$ is calculated and the following packet is transmitted to the next column: $s^{(ij)}(k) = \left([\hat{x}_{(ij)}(k)^T, \hat{x}_{(i^*j)}(k)^T]^T, [u_{(ij)}(k)^T, u_{(i^*j)}(k)^T]^T, J_{(ij)}(k)\right)$.

The estimated state $\hat{x}_{(ij)}(k)$ is sent to take into account the cases in where the nodes that receive the packet (they do not calculate control value because they have a packet with $u_{(ij)}(k) \neq 0$) have some neighbor that is required to calculated a control value, but it cannot estimate the whole state by itself.

6.1.5 Simulation example

A system with decentralized models is considered

$$\mathbf{A}_{(1,1)(1,1)} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{A}_{(2,1)(2,1)} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},$$

$$\mathbf{B}_{(1,1)(1,1)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{B}_{(2,1)(2,1)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

and interacting models given by:

$$\mathbf{A}_{(1,1)(2,1)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{A}_{(2,1)(1,1)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},$$

$$\mathbf{B}_{(1,1)(2,1)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{B}_{(2,1)(1,1)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

where no noise is considered and $\bar{x}(0) = 1$.

The network is as depicted in Fig. 6.3, with i.i.d. transmission processes and success probabilities $\mathbf{Prob}\{\gamma_{(i,j))}^{(i,j-1)}(k) = 1\} = 0.4$ and $\mathbf{Prob}\{\delta^{(i,j)}(1) = 1\} = 1$.

Fig. 6.6 shows the empirical distribution of the controller node $c(k)$ obtained by running the algorithm for 100 steps. It is possible to see how 43% of the times, the controller node is located in the last column of nodes, the one closest to the actuator.

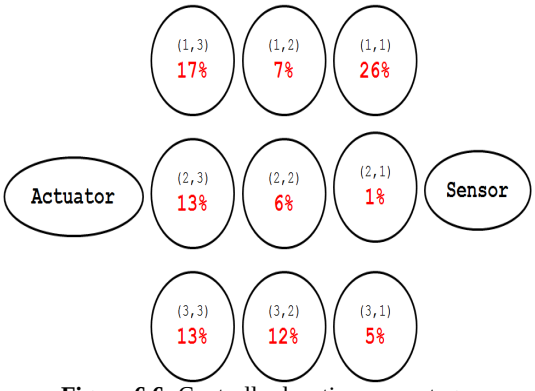

Figure 6.6: Controller location percentage.

Fig. 6.7 and 6.8 compare the plant state trajectory when the algorithm proposed is used with the case in which the controller is located at the actuator node. The results suggest that the proposed algorithm yields a stable system, while when the controller is at the actuator, the system becomes unstable.

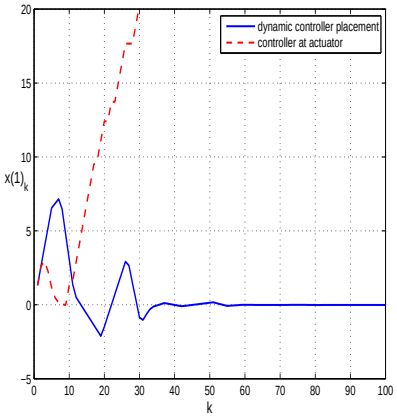

Figure 6.7: $x(1)$ trajectory.

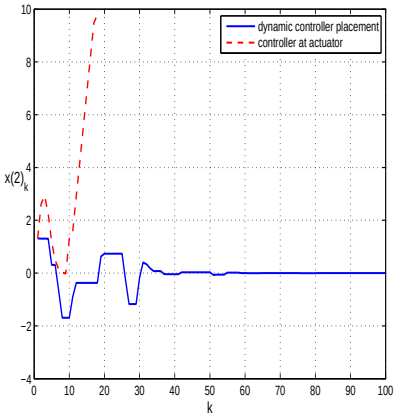

Figure 6.8: $x(2)$ trajectory.

6.2 Estimator Placement for a Single-Loop Networked Control System

In this section, the network is composed by multiple nodes connected in series. A new algorithm will be proposed, in which the nodes will send their state estimation, and the last node (the actuator node) will calculate the control value.

6.2.1 Wireless Sensor-Actuator Network Setup

It will be considered MIMO LTI plant models of the form

$$x(k+1) = Ax(k)+Bu(k)+w(k) \tag{6.11}$$

$$y(k) = Cx(k)+v(k), \quad k \in \mathbb{N}_0 \tag{6.12}$$

where $x(0) \sim \mathcal{N}(\bar{x}(0), P_0)$, $P_0 > 0$. In (6.11), $u(k) \in \mathbb{R}^{m_1}$ is the plant input, $x(k) \in \mathbb{R}^n$ is the state, $y(k) \in \mathbb{R}^p$ is the output, and $w(k) \sim \mathcal{N}(0,Q)$, $Q > 0$ and $v(k) \sim \mathcal{N}(0,R)$, $R > 0$, are noise and measurement noise, respectively.[1] As explained in the introduction, it is considered that a suitable feedback and estimator gains $L \in \mathbb{R}^{m_1 \times n}$ and $K \in \mathbb{R}^{n \times p}$ have been pre-designed. Consequently, it is assumed that if the control inputs

$$u(k) = L\hat{x}_{nom}(k), \quad k \in \mathbb{N}_0, \tag{6.13}$$

with

$$\hat{x}_{nom}(k) = (A+BL)\hat{x}_{nom}(k-1)K\big(y(k)-C\big((A+BL)\big)\hat{x}_{nom}(k-1)\big) \tag{6.14}$$

where $\hat{x}_{nom}(k)$ denotes an estimate of the state $x(k)$, that provides satisfactory performance when it is implemented at the plant. The main theme of the present section is to investigate how to implement the above nominal controller, when using a wireless sensor-actuator network.

[1] $v(k)$ can also represent quantization errors, modelled as Gaussian; see, e.g., [Quevedo et al., 2010].

Sensor and actuator nodes are connected through a wireless network, characterised via a (directed) line-graph having M nodes, as it is represented in Fig. 6.9. Transmissions are in sequential Round-Robin fashion $\{1,2,\dots,M,1,2,\dots\}$ as depicted in Fig. 6.10. More precisely, the packet $s(k)^{(i)}$ is transmitted from node i to node $i+1$ at times $kT+i\tau$, where T is the sampling period of (6.11) and $\tau \ll T/(M+1)$ refers to the times between transmissions of packets $s(k)k^{(i)}$. The plant input $u(k)$ is applied at time $kT+(M+1)\tau$. Thus, it is assumed that the transmissions in the network are much faster than the plant dynamics (6.11) and, as in, e.g., [C. L. Robinson and P. R. Kumar, 2008], therefore delays introduced by the network are not considered.

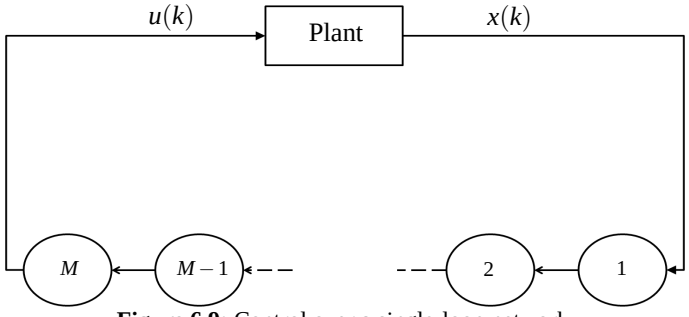

Figure 6.9: Control over a single-loop network.

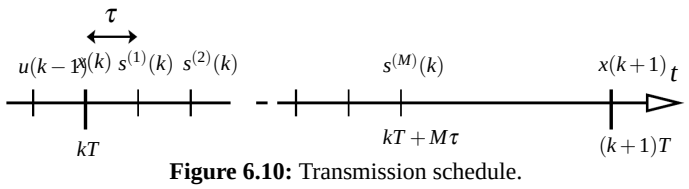

Figure 6.10: Transmission schedule.

Although the network is not introducing delays in the communication, it does introduce stochastic packet dropouts. To study the situation, an erasure channel model is adopted, that introduces the binary success processes

$$\gamma^{(i)}(k) \in \{0,1\}, \quad k \in \mathbb{N}_0,\ i \in \{1,2,\dots,M-1\},$$

where $\gamma^{(i)}(k) = 1$ indicates that transmission of the packet $s(k)^{(i)}$ from node i to node $i+1$ at time $kT + i\tau$, is successful, i.e., no error has occurred; $\gamma^{(i)}(k) = 0$ means there is a packet dropout at k. Throughout this work, it is assumed that transmission outcomes are known at the corresponding receiver sides ([Ma et al., 2011]). Furthermore, it is also assumed that the sensor node $i = 1$ has direct access to plant output measurements. The notation $\gamma^{(0)}(k) = 1$ is used, for all $k \in \mathbb{N}_0$. To save energy, the wireless nodes $i \in \{1, 2, \ldots, M-1\}$ do not provide acknowledgments of receipt of the packets.

Whilst packet acknowledgments are not provided, in the present scheme the actuator node M provides a feedback mechanism: At time $(k+1)T - \tau$, it broadcasts the control value $u(k)$ to nodes $i \in \{1, \ldots, M-1\}$, see Fig. 6.9. Due to channel fading, the *feedback links* between actuator and sensors are also affected by dropouts. The following notation is used to denote the associated success processes:

$$\delta^{(i)}(k) \in \{0,1\}, \quad k \in \mathbb{N}_0,\ i \in \{1,2,\ldots,M-1\}.$$

More precisely, if $\hat{x}(k)$ is successfully received at node i, then $\delta(k)^{(i)} = 1$; see also [E. Garone, B. Sinopoli and A. Casavola, 2010, O. C. Imer and S. Yüksel and T. Başar, 2006] for studies on the importance of acknowledgments in closed loop control. It is assumed that the actuator node has perfect knowledge of plant inputs, therefore $\delta(k)^{(M)} = 1$, $\forall k \in \mathbb{N}_0$. Since the actuator node will, in general, have less stringent energy constraints than the other nodes, it is more important to focus on situations where the feedback links are more reliable than the forward links moving data from the sensor to the actuator.

Due to packet dropouts, plant output measurements are not always available at the actuator node. On the other hand, the sensor node will, in general, not have perfect information of previous plant inputs. This makes the implementation of the estimate of the state (6.13) a challenging task. The main purpose of the present section is to present an adaptive state estimator placement algorithm, where the computations of the state estimation are distributed across the network. This approach will lead to a dynamic assignment of the role played by

the individual network nodes. The tasks carried out by individual nodes at each time instant will depend on transmission outcomes.

6.2.2 Flexible Estimator Placement

To keep communications and thereby energy use low, the packets transmitted by each node i have only three fields, namely, output measurements and tentative plant state estimate and the acknoledgement success process:

$$s^{(i)}(k) = \big(y(k), \hat{x}^{(i)}(k), \delta^{(c^{(i)}(k))}(k-1)\big). \tag{6.15}$$

Plant outputs are transmitted in order to pass on information on the plant state to the nodes $\{i+1, i+2, \ldots, M\}$, see Fig. 6.9. On the other hand, $\hat{x}^{(i)}(k)$ in (6.15) is the plant state estimate which is used at the actuator to calculate the control input provided the packet $s^{(i)}(k)$ is delivered at the actuator node. If $s^{(i)}(k)$ is lost, then following the algorithm described in Section 6.2.2, the state estimate will be provided by one of the later nodes $\ell > i$, which takes the role of the estimator at time k.

State Estimation and Control Calculations

In the sequel, the notation $c(k) \in \{1, 2, \ldots, M\}$ will be used to denote the node that calculates the state estimation that will be used to compute the plant input at time k. Thus, the plant input is given by

$$u(k) = L\hat{x}^{(c(k))}(k), \quad k \in \mathbb{N}_0, \tag{6.16}$$

where $\hat{x}^{(c(k))}(k)$ is the local plant state estimate computed at node $c(k)$, and the control input $u(k)$ is always calculated at the actuator node. Intuitively, good control performance will be achieved if the estimate used in (6.16) is accurate. Clearly, due to the multi-hop nature of the network, nodes which are closer to the sensor will have access to more output measurements, see Fig. 6.9. On the other hand, nodes which are physically located closer

to the actuator node will on average receive more plant input acknowledgments, thus, have better knowledge of plant inputs.

In the present formulation not all nodes compute local state estimates, $\hat{x}^{(i)}(k)$, by using the data received from the preceding node. Situations where the feedback links from the actuator to the intermediate sensors are *quite reliable* are considered in this section. State estimators are of the form (6.14) when the plant output is available at node i; at instants when the state estimation used to compute the plant input is not available, an open loop estimate is used, that is:

$$\hat{x}^{(i)}(k) = (A+BL)\hat{x}^{(i)}(k-1) + K^{(i)}(k)\big(y(k) - C(A+BL)\hat{x}^{(i)}(k-1))\big), \tag{6.17}$$

where

$$K^{(i)}(k) \triangleq \Gamma^{(i)}(k)K, \tag{6.18}$$

and

$$\Gamma^{(i)}(k) \triangleq \prod_{j\in\{0,1,\ldots,i-1\}} \gamma^{(j)}(k) \tag{6.19}$$

is equal to 1 if and only if $y(k)$ is available at node i at time $kT + (i-1)\tau$. In (6.17), $\hat{x}^{(i)}(k-1)$ is a local plant state estimate. In particular, if $\delta^{(i)}(k-1) = 1$, then $L\hat{x}^{(i)}(k-1) = u(k-1)$. On the other hand, at instants when $\delta^{(i)}(k-1) = 0$, node i uses $\hat{x}^{(i)}(k-1)$. More details on the estimator are given in Section 6.2.4.

Remark 6.2.1. Of course, the above transmission and control strategy will in general not be optimal. In particular, nodes do not transmit local state estimates and the control law does not depend upon network parameters, e.g., dropout probabilities ([Chiuso and Schenato, 2011]). The aim of the present section is to develop a simple and practical method, which uses an existing control and estimation policy for implementation over an unreliable network and only requires little communication. □

Remark 6.2.2. In the recent work [D. E. Quevedo, K. H. Johansson, A. Ahlén and I. Jurado, 2012], instead of using (6.18), the gains $K^{(i)}(k)$ were taken as the Kalman filter

gains for a system with intermittent observations:

$$K^{(i)}(k) = \Gamma^{(i)}(k)P^{(i)}(k)C^T(CP^{(i)}(k)C^T+R)^{-1} \quad (6.20)$$

$$P^{(i)}(k+1) = A\big(I_n - K^{(i)}(k)C\big)P^{(i)}(k)A^T + Q, \quad (6.21)$$

see, e.g., [Sinopoli et al., 2004, Huang and Dey, 2007, Quevedo et al., 2012]. The subsequent analysis (up to Equation (6.35)) can be applied to this structure as well. However, the jump-linear model derived in Section 6.2.4 requires a jump-linear estimation model, such as (6.18). □

Algorithm for Dynamic Estimator Placement

Algorithm 3 is run at every node $i \in \{1,2,\ldots,M\}$ and describes the adaptive estimator allocation method explained in the preceding section. As can be appreciated, which calculations are carried out at each node, depends on transmission outcomes involving the current node and also transmission outcomes at previous nodes. In particular, node i only calculates a tentative plant state estimate when no tentative plant state estimate is received from node $i-1$ and node i has successfully received $\hat{x}(k-1)$ while $c^{(i)}(k)$ has not. Therefore, preference is given to simply relay incoming state estimation values rather than to replace valid tentative ones.

The reason for adopting this decision procedure lies in the assumption that data sent from the actuator node to intermediate nodes is often available, whereas transmissions of packets $s^{(i)}(k)$ are less reliable. This suggests that nodes closer to the sensor node can be expected to have better state estimates than nodes located closer to the actuator node.

In particular, the sensor node $i = 1$ uses as input

$$s^{(0)}(k) = (y(k), \emptyset, \delta^{(1)}(k-1)), \quad \gamma^{(0)}(k) = 1. \quad (6.22)$$

Then the sensor node calculates a tentative control value and transmits

$$s^{(1)}(k) = (y(k), \hat{x}^{(1)}(k), \delta^{(1)}(k-1))$$

Algorithm 3 Adaptive Estimator Placement

$k \leftarrow 0$, $\hat{x}^{(i)}(0) \leftarrow 0$, $j \leftarrow 0$
while $t \geq 0$ **do** {$t \in \mathbb{R}_{\geq 0}$ is actual time}
  while $t \leq kT + (i-1)\tau$ **do** {wait-loop}
    $j \leftarrow j+1$
  end while
  if $\gamma^{(i-1)}(k) = 0$ **then** {$s^{(i-1)}(k)$ is dropped}
    if $\delta^{(c^{(i)}(k-1))}(k-1) = 1$ **then**
      $\hat{x}^{(i)}(k) = \big(A+BL\big)\hat{x}^{(c^{(i)}(k-1))}(k-1)$
    else
      $\hat{x}^{(i)}(k) = \big(A+BL\big)\hat{x}^{(i)}(k-1)$
    end if
    $s^{(i)}(k) \leftarrow \big(\emptyset, \hat{x}^{(i)}(k), \delta^{(i)}(k-1)\big)$
    $c^{(i)}(k) = i$
  end if
  if $\gamma^{(i-1)}(k) = 1$ **then** {$s^{(i-1)}(k)$ is received}
    if $\delta^{(c^{(i-1)}(k))}(k-1) = 1 \vee \big(\delta^{(i)}(k-1) = 0 \wedge \delta^{(c^{(i-1)}(k))}(k-1) = 0\big)$ **then**
      $\hat{x}^{(i)}(k) \leftarrow \hat{x}^{(i-1)}(k)$
    else
      $\hat{x}^{(i)}(k) \leftarrow \big(A+BL\big)\hat{x}^{(i)}(k-1) + K^{(i)}(k)\big(y(k) - C(A+BL)\hat{x}^{(i)}(k-1)\big)$
      $c^{(i)}(k) = i$
    end if
    $s^{(i)}(k) \leftarrow \big(y(k), \hat{x}^{(i)}(k), max\{\delta^{(c^{(i-1)}(k))}(k-1), \delta^{(i)}(k-1)\}\big)$
  end if

while $t < kT + i\tau$ **do** {wait-loop}
  $j \leftarrow j+1$
end while
transmit $s^{(i)}(k)$
while $t \leq (k+1)T - \tau$ **do** {wait-loop}
  $j \leftarrow j+1$
end while
if $i = M$ **then** { i is the actuator node}
  $u(k) = L\hat{x}^{(c^{(i)}(k))}(k)$
end if
$k \leftarrow k+1$
end while

to node $i = 2$. Subsequent nodes then relay this packet to the actuator node. If the packet is dropped along the way, or $\delta^{(i)}(k-1) = 1$ while $\delta^{(c^{(i)}(k))}(k-1) = 0$, then the next node i, calculates a tentative state estimation $\hat{x}^{(i)}(k)$ and transmits

$$s^{(i)}(k) = (\emptyset, \hat{x}^{(i)}(k), \delta^{(i)}(k-1))$$

to node $i+1$, etc. Control calculations are then carried out at the actuator node M with the last state estimate calculated $\hat{x}^{(c^{(i)}(k))}(k)$ and $\hat{x}^{(c^{(i)}(k))}(k)$ is relayed towards all the previous nodes. The actuator node implements $u(k) = L\hat{x}^{(c^{(i)}(k))}(k)$, using the value contained in the second field of $s^{(M)}(k)$.

Remark 6.2.3. An advantage of allowing the state estimate calculations to be located arbitrarily and in a time-varying fashion, is that it makes more difficult for someone to attack the NCS. The latter problem has been studied, for example, in [Gupta et al., 2010, Smith, 2011]. □

6.2.3 Dynamic Estimator Location

With Algorithm 3, which of the nodes calculates the plant state estimates $\hat{x}(k)$, depends upon the transmission outcomes. For further reference, the set of nodes which calculate a

tentative state estimates will be denoted as

$$\mathscr{C}(k) \subset \{1,2,\ldots,M\}.$$

Preliminary Analysis

To analyze the situation, it is convenient to introduce the process $\left\{c^{(i)}(k)\right\}$, where $k \in \mathbb{N}_0$, $i \in \{0,1,\ldots,M\}$ and with

$$c^{(i)}(k) \quad \triangleq \max(\mathscr{C}(k) \cap \{1,2,\ldots,i\}). \tag{6.23}$$

If $c^{(i)}(k) > 0$, then $c^{(i)}(k)$ denotes the node where the second field of $s^{(i)}(k)$ was calculated.

It is easy to see that, with the algorithm proposed and since the packets $s^{(i)}(k)$ are communicated sequentially, see Fig. 6.10, $c^{(1)}(k) = \delta^{(1)}(k-1)$, for all $k \in \mathbb{N}_0$, whereas

$$c^{(i)}(k) = \begin{cases} i\delta^{(i)}(k-1) & \text{if } c^{(i-1)}(k) = 0 \vee \gamma^{(i-1)}(k) = 0, \\ c^{(i-1)}(k) & \text{if } c^{(i-1)}(k) > 0 \wedge \gamma^{(i-1)}(k) = 1, \end{cases} \tag{6.24}$$

for $i \in \{2,\ldots,M\}$, $k \in \mathbb{N}_0$. The estimator node at time k is given by

$$c(k) \triangleq c^{(M)}(k) = \max(\mathscr{C}(k)), \quad \forall k \in \mathbb{N}_0, \tag{6.25}$$

see (6.16). To derive the results, the aggregated transmission outcome process $\{\beta(k)\}$, $k \in \mathbb{N}_0$, is introduced, where

$$\beta(k) \triangleq \sum_{i=1}^{M-1} \left(2^{M-1}\gamma^{(i)}(k+1) + \delta^{(i)}(k)\right)2^{i-1}, \quad k \in \mathbb{N}_0. \tag{6.26}$$

Note that $\beta(k-1) \in \mathbb{I} \triangleq \{0,1,\ldots,2^{2M-2}-1\}$ collects the outcomes of all transmissions which occur during the time-interval $[kT-\tau, kT+M\tau]$, see Fig. 6.10. Thus, $\beta(k-1)$ determines $\mathscr{C}(k)$ and $c(k)$.

Results

As seen in the preceding analysis, with Algorithm 3, the estimator location $c(k)$ will dynamically adapt to the network conditions, as quantified in the aggregated transmission

outcomes $\beta(k)$. To further elucidate the situation, in the sequel it will be adopted a stochastic framework and thereby regard $\{\beta(k)\}$, $k \in \mathbb{N}_0$ as a stochastic process. It will be assumed that the transmission and acknowledgment processes are Bernoulli distributed. In Section 6.2.5 it will be adopted a more realistic model, wherein transmission processes are correlated in time and among each other.

Assumption 6.2.4. *The link transmission processes are independent and identically distributed (i.i.d.) with a common success probability $p \in [0,1]$:*

$$\mathbf{Pr}\big\{\gamma^{(i)}(k) = 1\big\} = p, \quad \forall i \in \{1,2,\ldots,M-1\}. \tag{6.27}$$

The feedback link success processes are i.i.d., with

$$\mathbf{Pr}\big\{\delta^{(i)}(k) = 1\big\} = q_i, \quad \forall i \in \{1,2,\ldots,M-1\}, \tag{6.28}$$

for given success probabilities $q_1, q_2, \ldots, q_{M-1} \in [0,1]$. □

Note that while the above assumption imposes that transmission processes are i.i.d., it does take into account the fact that radio connectivity from the actuator node to the other nodes will be distance dependent; see, e.g., [Goldsmith, 2005]. It also does not impose that the processes $\big\{\mu^{(i)}(k)\big\}$, $k \in \mathbb{N}_0$ for different nodes i are independent. However, the assumption made does imply stationarity, as apparent from Proposition 6.2.5 given below.

Proposition 6.2.5. *Suppose that Assumption 6.2.4 holds. Then*

$$\mathbf{Pr}\{i \in \mathscr{C}(k)\} = \begin{cases} 1 & \text{if } i = 1 \\ 1 - p\big[1 - q_i(1 - q_{c^{(i-1)}(k)})\big] & \text{if } i \in \{2,\ldots,M\} \end{cases} \tag{6.29}$$

$$\mathbf{Pr}\{c(k) = i\} = \mathbf{Pr}\{i \in \mathscr{C}(k)\} p^{M-i} \times \tag{6.30}$$

$$\big[1 - (1 - q_i)[1 - (1 - q_{i+1})(1 - q_{i+2}) \ldots (1 - q_{M-1})]\big],$$

for all $k \in \mathbb{N}_0$ and $i \in \{1,2,\ldots,M\}$, and where $q_M = 1$.

Proof Clearly, if Assumption 6.2.4 holds, then

$$\mathbf{Pr}\{\mu^{(1)}(k) = 1\} = \mathbf{Pr}\{\delta^{(1)}(k-1) = 1\} = q_1.$$

It is easy to see from Algorithm 3 that

$$i \in \mathscr{C}(k) \Longleftrightarrow \gamma^{(i-1)}(k) = 0 \tag{6.31}$$

$$\vee\big(\gamma^{(i-1)}(k) = 1 \wedge \big(\delta^{(i)}(k-1) = 1 \wedge \delta^{(c^{(i-1)}(k-1))}(k-1) = 0\big)\big)$$

$$\Longleftrightarrow \gamma^{(i-1)}(k) = 0 \vee \big(\delta^{(i)}(k-1) = 1 \wedge \delta^{(c^{(i-1)}(k-1))}(k-1) = 0\big)$$

for all $i \in \{1,2,\ldots,M\}$. Expression (6.31) gives

$$\mathbf{Pr}\{i \in \mathscr{C}(k)\}$$

$$= 1 - \mathbf{Pr}\big\{\gamma^{(i-1)}(k) = 1 \wedge \big(\delta^{(i)}(k-1) = 0 \vee \delta^{(c^{(i-1)}(k-1))}(k-1) = 1\big)\big\}$$

$$= 1 - p \times \mathbf{Pr}\big\{\delta^{(i)}(k-1) = 0 \vee \delta^{(c^{(i-1)}(k-1))}(k-1) = 1\big\}$$

$$= 1 - p\big(1 - \mathbf{Pr}\big\{\delta^{(i)}(k-1) = 1 \wedge \delta^{(c^{(i-1)}(k-1))}(k-1) = 0\big\}\big)$$

$$= 1 - p\big[1 - q_i(1 - q_{c^{(i-1)}(k-1)})\big],$$

thus establishing (6.29). By (6.25) the distribution of $c(k)$ can be determined from

$$\mathbf{Pr}\{c(k) = i\} = \mathbf{Pr}\{\max(\mathscr{C}(k)) = i\}$$

$$= \mathbf{Pr}\{i \in \mathscr{C}(k) \wedge \big[\big(\gamma^{i}(k) = \gamma^{i+1}(k) = \cdots = \gamma^{M-1} = 1\big)$$

$$\wedge\big(\delta^{(i)}(k-1) \vee (\delta^{(i+1)}(k-1) = \delta^{(i+2)}(k-1) = \cdots = \delta^{(M-1)}(k-1) = 0)\big)\big]\}$$

$$= p^{M-1}\mathbf{Pr}\{i \in \mathscr{C}(k)\} \times \big[1 - (1 - q_i)$$

$$\times \mathbf{Pr}\{\delta^{(i+1)}(k-1) = 1 \vee \delta^{(i+2)}(k-1) = 1 \cdots \vee \delta^{(M-1)}(k-1) = 1\}\big]$$

$$= p^{M-1}\mathbf{Pr}\{i \in \mathscr{C}(k)\} \times \big[1 - (1 - q_i)$$

$$\times \mathbf{Pr}\{\delta^{(i+1)}(k-1) = 0 \wedge \delta^{(i+2)}(k-1) = 0 \cdots \wedge \delta^{(M-1)}(k-1) = 0\}\big]$$

for $i \in \{1,\ldots,M-1\}$, whereas for the actuator node: $\mathbf{Pr}\{c(k) = M\} = \mathbf{Pr}\{M \in \mathscr{C}(k)\}$. This proves (6.30). □

The above result characterizes the distributions of $\mu^{(i)}(k)$, of $\mathscr{C}(k)$, and of the controller node location $c(k)$. These distributions depend upon the communication success probabilities; i.e., the distribution of $\beta(k)$, here modeled as i.i.d.

Examples

Before showing the closed loop performance with the algorithm proposed, two simple examples which illustrate how the control location distribution depends upon the dropout probabilities are included.

Example 6.2.1. Suppose that Assumption 6.2.4 holds and that the feedback links are always available, that is, $q_i = 1$, for all $i \in \{1,\dots,M\}$. Proposition 6.2.5 gives that

$$\mathbf{Pr}\big\{i \in \mathscr{C}(k)\big\} = \begin{cases} 1 & \text{if } i = 1 \\ 1-p & \text{if } i \in \{2,\dots,M\} \end{cases}$$

and the controller location sequence has the following geometric-like distribution

$$\mathbf{Pr}\{c(k) = i\} = \begin{cases} p^{M-1} & \text{if } i = 1 \\ (1-p)p^{M-i} & \text{if } i \in \{2,3,\dots,M\}. \end{cases}$$

□

Example 6.2.2. Consider an NCS as in Fig. 6.9 with $M = 3$ nodes and suppose that Assumption 6.2.4 holds. In this case, Proposition 6.2.5 establishes that

$$\mathbf{Pr}\big\{i \in \mathscr{C}(k)\big\} = \begin{cases} 1 & \text{if } i = 1 \\ 1-p\big[1-q_2(1-q_1)\big] & \text{if } i = 2 \\ \begin{cases} 1-p\big[1-q_3(1-q_1)\big] & \text{if } c^{(i-1)}(k) = 1 \\ 1-p\big[1-q_3(1-q_2)\big] & \text{if } c^{(i-1)}(k) = 2 \end{cases} & \text{if } i = 3 \end{cases}$$

and the controller location distribution

$$\mathbf{Pr}\{c(k) = i\} = \begin{cases} p^3\big[1-q_2(1-q_1)\big] & \text{if } i = 1, \\ p\big(1-p\big[1-q_2(1-q_1)\big]\big)\big[1-(1-q_2)^2\big] & \text{if } i = 2 \\ \begin{cases} q_2\big(1-p\big[1-q_3(1-q_1)\big]\big) & \text{if } c^{(i-1)}(k) = 1 \\ q_2\big(1-p\big[1-q_3(1-q_2)\big]\big) & \text{if } c^{(i-1)}(k) = 2 \end{cases} & \text{if } i = 3 \end{cases}$$

6.2.4 Closed Loop Model

The algorithm proposed in the present work embodies a network driven distributed state estimation and control architecture. Closed loop dynamics depend upon transmission outcomes, the plant model (6.11) and nominal controller/estimator dynamics, see (6.13)–(6.14).

To derive a compact model for the wireless sensor-actuator control system of interest, it is convenient to introduce the aggregated state estimation vector

$$\hat{x}(k) \triangleq \begin{bmatrix} \hat{x}(k)^{(1)} \\ \hat{x}(k)^{(2)} \\ \vdots \\ \hat{x}^{(M)}(k) \end{bmatrix} \in \mathbb{R}^{Mn}.$$

The *backup value* for $u(k)$ used at node i is denoted as

$$v^{(i)}(k) = \begin{cases} \hat{x}^{(i)}(k) & \text{if } \mu^{(i)}(k) = 0, \\ \hat{x}^{(j)}(k),\ j = c^{(i)}(k) & \text{if } \mu^{(i)}(k) = 1, \end{cases}$$

see (6.23) and note that $v^{(1)}(k) = L\hat{x}^{(1)}(k)$, for all $k \in \mathbb{N}_0$. In view of (6.24),

$$v^{(i)}(k) = b^{(i)}(k)\hat{x}(k), \tag{6.32}$$

where $b^{(1)}(k) \triangleq e_1 \in \mathbb{R}^{Mn}$ for all $k \in \mathbb{N}_0$, whereas for $i \geq 2$,

$$b^{(i)}(k) \triangleq e_\ell \otimes L \in \mathbb{R}^{m \times Mn} \tag{6.33}$$

$$\ell = \begin{cases} i \longrightarrow \text{if } c^{(i-1)}(k) = 0 \vee \gamma^{(i-1)}(k) = 0 \\ \vee(\gamma^{(i-1)}(k) = 1 \wedge (\delta^{(i)}(k-1) = 1 \wedge \delta^{(c^{(i)}(k))}(k-1) = 0)), \\ c^{(i-1)}(k) \longrightarrow \text{if } c^{(i-1)}(k) > 0 \wedge \gamma^{(i-1)}(k) = 1 \\ \wedge \delta^{(c^{(i)}(k))}(k-1) = 1 \vee (\delta^{(i)}(k-1) = 0 \wedge \delta^{(c^{(i)}(k))}(k-1) = 0), \end{cases}$$

depends on the realization of $\beta(k-1)$, see (6.26).

Since the algorithm gives $u(k) = Lv^{(M)}(k)$, the plant input estimates used in the state estimators satisfy:

$$\hat{u}^{(i)}(k) = \begin{cases} Lv^{(M)}(k) & \text{if } \delta^{(i)}(k) = 1, \\ Lv^{(i)}(k) & \text{if } \delta^{(i)}(k) = 0 \end{cases} \tag{6.34}$$

$$= \big(\delta^{(i)}(k)(e_M \otimes I_m) + \big(1 - \delta^{(i)}(k)\big)(e_i \otimes I_m)\big)Lv(k),$$

where

$$v(k) \triangleq \begin{bmatrix} v^{(1)}(k) \\ v^{(2)}(k) \\ \vdots \\ v^{(M)}(k) \end{bmatrix} \in \mathbb{R}^{Mm}$$

forms part of the internal variables used by the M state estimators.

Now, the plant model can be written as

$$x(k+1) = Ax(k) + BLv^{(M)}(k) + w(k). \tag{6.35}$$

Expressions (6.35), (6.17) and (6.34) then give that the state estimator at node

$$i \in \{1,2,\ldots,M\},$$

obeys the recursion:

$$\begin{aligned}\hat{x}^{(i)}(k+1) &= \big(I_n - K^{(i)}(k+1)C\big)\big(A\hat{x}^{(i)}(k) + B\hat{u}^{(i)}(k)\big) \\ &\quad + K^{(i)}(k+1)\big(CAx(k) + CBLv^{(M)}(k) + Cw(k) + v(k+1)\big) \\ &= K^{(i)}(k+1)CAx(k) + \big(I_n - K^{(i)}(k+1)C\big)(e_i \otimes A)\hat{x}(k) \\ &\quad + d^{(i)}(k)Lv(k) + K^{(i)}(k+1)\big(Cw(k) + v(k+1)\big),\end{aligned} \tag{6.36}$$

where

$$\begin{aligned}d^{(i)}(k) &\triangleq \big(1 - \delta^{(i)}(k)\big)\big(I_n - K^{(i)}(k+1)C\big)(e_i \otimes B) \\ &\quad + \big(\big(1 - \delta^{(i)}(k)\big)K^{(i)}(k+1)C + \delta^{(i)}(k)I_n\big)(e_M \otimes B).\end{aligned}$$

Now, introducing the following

$$\Theta(k) \triangleq \begin{bmatrix} x(k) \\ \hat{x}(k) \\ v(k) \end{bmatrix}, \quad n(k) \triangleq \begin{bmatrix} w(k) \\ v(k+1) \end{bmatrix}, \tag{6.37}$$

and use (6.18), then (6.36) becomes

$$\hat{x}^{(i)}(k+1) = \mathscr{D}^{(i)}(\beta(k))\Theta(k) + \mathscr{E}^{(i)}(\beta(k))n(k),$$

with

$$\mathscr{D}^{(i)}(\beta(k)) \triangleq \Big[\Gamma^{(i)}(k+1)KCA \quad \big(I_n - \Gamma^{(i)}(k+1)KC\big)(e_i \otimes A) \quad d^{(i)}(k)\Big]$$

$$\mathscr{E}^{(i)}(\beta(k)) \triangleq \Gamma^{(i)}(k+1)K\begin{bmatrix} C & I_p \end{bmatrix}.$$

State estimators, thus, follow the dynamic relation

$$\hat{x}(k+1) = \mathscr{D}(\beta(k))\Theta(k) + \mathscr{E}(\beta(k))n(k) \tag{6.38}$$

where

$$\mathscr{D}(\beta(k)) \triangleq \begin{bmatrix} \mathscr{D}^{(1)}(\beta(k)) \\ \vdots \\ \mathscr{D}^{(M)}(\beta(k)) \end{bmatrix}, \quad \mathscr{E}(\beta(k)) \triangleq \begin{bmatrix} \mathscr{E}^{(1)}(\beta(k)) \\ \vdots \\ \mathscr{E}^{(M)}(\beta(k)) \end{bmatrix}.$$

On the other hand, (6.32) provides the following dynamic relationship:

$$\nu(k+1) = \mathscr{F}(\beta(k))\Theta(k) + \mathscr{G}(\beta(k))n(k), \tag{6.39}$$

where

$$\mathscr{F}(\beta(k)) \triangleq \begin{bmatrix} b^{(1)}(k+1)\mathscr{D}(\beta(k)) \\ \vdots \\ b^{(M)}(k+1)\mathscr{D}(\beta(k)) \end{bmatrix}, \quad \mathscr{G}(\beta(k)) \triangleq \begin{bmatrix} b^{(1)}(k+1)\mathscr{E}(\beta(k)) \\ \vdots \\ b^{(M)}(k+1)\mathscr{E}(\beta(k)) \end{bmatrix}.$$

Expressions (6.35), (6.38) and (6.39) lead to the jump-linear model

$$\Theta(k+1) = \mathscr{A}(\beta(k))\Theta(k) + \mathscr{B}(\beta(k))n(k), \tag{6.40}$$

where

$$\mathscr{A}(\beta(k)) \triangleq \begin{bmatrix} \begin{bmatrix} A & 0_{Mn} & e_M \otimes BL \end{bmatrix} \\ \mathscr{D}(\beta(k)) \\ \mathscr{F}(\beta(k)) \end{bmatrix}, \mathscr{B}(\beta(k)) \triangleq \begin{bmatrix} \begin{bmatrix} I_n & 0 \end{bmatrix} \\ \mathscr{E}(\beta(k)) \\ \mathscr{G}(\beta(k)) \end{bmatrix}$$

and where the jump variable $\{\beta(k)\}$, $k \in \mathbb{N}_0$ is given by the aggregated transmission outcome process defined in (6.26).

Example 6.2.3. Consider a simple NCS with only two nodes, $M = 2$, in which case $\beta(k) = 2\gamma^{(1)}(k+1) + \delta^{(1)}(k)$ and $\mathbb{I} = \{0,1,2,3\}$, see (6.26). Since $c^{(1)}(k) = \delta^{(1)}(k-1)$, $\delta^{(2)}(k-1) = 1$ and $b^{(1)}(k+1) = e_1 \otimes L$ for all $k \in \mathbb{N}_0$, (6.33) yields:

$$b^{(2)}(k+1) = \begin{cases} e_2 \otimes L & \text{if } \beta(k) < 3, \\ e_1 \otimes L & \text{if } \beta(k) = 3. \end{cases}$$

Direct calculations give that, in this case, the matrices in (6.38) are given by

$$\mathscr{D}(\beta(k)) = \begin{bmatrix} KCA & \big(I_n - KC\big)(e_1 \otimes A) & d^{(1)}(k) \\ \gamma^{(1)}(k+1)KCA & \big(I_n - \gamma^{(1)}(k+1)KC\big)(e_2 \otimes A) & e_2 \otimes B \end{bmatrix}$$

$$\mathscr{E}(\beta(k)) = \begin{bmatrix} KC & K \\ \gamma^{(1)}(k+1)KC & \gamma^{(1)}(k+1)K \end{bmatrix},$$

where

$$d^{(1)}(k) = \begin{cases} \begin{bmatrix} \big(I_n - KC\big)B & KCB \end{bmatrix} & \text{if } \beta(k) \in \{0,2\}, \\ e_2 \otimes B & \text{if } \beta(k) \in \{1,3\} \end{cases}$$

thus, characterizing the model (6.40). □

6.2.5 Performance Analysis

The jump-linear system model derived in the previous section can be used to carry out performance analysis of the flexible NCS architecture of interest, provided the aggregated transmission outcome process $\{\beta(k)\}$ is suitably described. In the remainder of this section, the stochastic modeling framework recently introduced in [Quevedo et al., 2011, D. E. Quevedo, A. Ahlén and K. H. Johansson, n.d.] will be adopted.

The underlying idea of the network model in [Quevedo et al., 2011, D. E. Quevedo, A. Ahlén and K. H. Johansson, n.d.] is that transmission outcome distributions depend upon the fading radio environment. To allow for temporal and spatial correlations of the radio environment (and possibly also for power and bit-rate control), in [Quevedo et al., 2011] a Markovian *network state*, $\{\Xi(k)\}$, $k \in \mathbb{N}_0$, which takes values in a finite set, say $\mathbb{B}$ is used. Each element of $\mathbb{B}$ corresponds to a possible configuration of the physical environment, e.g., position of mobile objects. Dropout probabilities of individual channels, when conditioned on the network state, are considered independent and fixed. In the particular instance where $\mathbb{B}$ has only one element, the model describes a situation with independent i.i.d. Bernoulli channels, as considered in previous works such as [Chiuso and Schenato, 2011]. Further details of the model can be found in [Quevedo et al., 2011, D. E. Quevedo, A.

Ahlén and K. H. Johansson, n.d.]. For the present purposes, the model can be summarized as follows:

Assumption 6.2.6. *The network state process* $\{\Xi(k)\}$, $k \in \mathbb{N}_0$ *is an aperiodic homogeneous Markov Chain with transition probabilities*

$$p_{ij} = \mathbf{Pr}\{\Xi(k+1) = j \,|\, \Xi(k) = i\}, \quad i, j \in \mathbb{B}.$$

and stationary distribution

$$\pi_i = \lim_{k\to\infty} \mathbf{Pr}\{\Xi(k) = i\}.$$

The aggregated transmission outcome process $\{\beta(k)\}$, $k \in \mathbb{N}_0$ *in (6.26) is conditionally independent given the network state* $\{\Xi(k)\}$, $k \in \mathbb{N}_0$,

$$\phi_{ij} \triangleq \mathbf{Pr}\{\beta(k) = i \,|\, \Xi(k) = j\},$$

for all $(i, j) \in \mathbb{I} \times \mathbb{B}$. □

It is worth noting that, with the above model, the process $\beta(k)$ is correlated, but not necessarily Markovian. However, the augmented jump process $(\beta(k), \Xi(k))$, $k \in \mathbb{N}_0$ forms a finite Markov Chain. Thus, under Assumption 6.2.6, (6.40) belongs to the class of Markov jump-linear systems, as studied for example in [O. L. V. Costa, M. D. Fragoso, and R. P. Marques, 2005, Lee and Dullerud, 2007]. In particular, Theorems 3.9 and 3.33 of [O. L. V. Costa, M. D. Fragoso, and R. P. Marques, 2005] establish necessary and sufficient conditions for mean-square stability (MSS) which can be stated in terms of feasibility of a linear-matrix inequality. The following result characterizes closed loop performance of the flexible networked control systems architecture of interest in the present work. It is tailored directly to the model in Assumption 6.2.6 without needing to resort to the augmented jump process $(\beta(k), \Xi(k))$.

Theorem 6.2.7. *Suppose that Assumption 6.2.6 holds, that the system (6.40) is MSS and define*

$$\overline{\mathscr{A}}_j \triangleq \mathbf{E}\big\{\mathscr{A}(\beta(k)) \,\big|\, \Xi(k) = j\big\} = \sum_{i\in\mathbb{I}} \phi_{ij}\mathscr{A}(i), \quad j \in \mathbb{B},$$

$$\overline{\mathscr{B}}_j \triangleq \mathbf{E}\big\{\mathscr{B}(\beta(k)) \,\big|\, \Xi(k) = j\big\} = \sum_{i\in\mathbb{I}} \phi_{ij}\mathscr{B}(i), \quad j \in \mathbb{B}.$$

Then

$$\lim_{k\to\infty}\mathbf{E}\{\Theta(k)\Theta(k)^T\}=\sum_{i\in\mathbb{B}}H_i, \tag{6.41}$$

where

$$H_i=\sum_{j\in\mathbb{B}}p_{ji}\overline{\mathscr{A}}_iH_j(\overline{\mathscr{A}}_i)^T+\pi_i\overline{\mathscr{B}}_iW(\overline{\mathscr{B}}_i)^T \tag{6.42}$$

with $W\triangleq\mathrm{diag}(Q,R)$. □

Proof. By the law of total expectation:

$$\mathbf{E}\{\Theta(k+1)\Theta(k+1)^T\}=\sum_{i\in\mathbb{B}}H_{k+1,i}, \tag{6.43}$$

where

$$H_{k+1,i}\triangleq\mathbf{E}\{\Theta(k+1)\Theta(k+1)^T\,|\,\Xi(k)=i\}\mathbf{Pr}\{\Xi(k)=i\}. \tag{6.44}$$

Now, the system equation (6.40) together with the network fading model in Assumption 6.2.6 allow one to write

$$\begin{aligned}\mathbf{E}\{\Theta(k+1)\Theta(k+1)^T\,|\,\Xi(k)=i\}&=\mathbf{E}\big\{\big(\mathscr{A}(\beta(k))\Theta(k)+\mathscr{B}(\beta(k))n(k)\big)\\&\quad\times\big(\mathscr{A}(\beta(k))\Theta(k)+\mathscr{B}(\beta(k))n(k)\big)^T\,\big|\,\Xi(k)=i\big\}\\&=\mathbf{E}\big\{\mathscr{A}(\beta(k))\Theta(k)\Theta(k)^T\mathscr{A}(\beta(k))^T\,\big|\,\Xi(k)=i\big\}\\&\quad+\mathbf{E}\big\{\mathscr{B}(\beta(k))n(k)n(k)^T\mathscr{B}(\beta(k))^T\,\big|\,\Xi(k)=i\big\}\\&=\mathbf{E}\big\{\mathscr{A}(\beta(k))\Theta(k)\Theta(k)^T\mathscr{A}(\beta(k))^T\,\big|\,\Xi(k)=i\big\}+\overline{\mathscr{B}}_iW(\overline{\mathscr{B}}_i)^T\end{aligned} \tag{6.45}$$

since $\{n(k)\}$ is zero-mean i.i.d.

Now, the rule of total expectation, Bayes' rule and the Markovian property of the model (6.40) give that

$$\mathbf{E}\big\{\mathscr{A}(\beta(k))\Theta(k)\Theta(k)^T\mathscr{A}(\beta(k))^T\,\big|\,\Xi(k)=i\big\} \tag{6.46}$$

$$= \sum_{j\in\mathbb{B}} \mathbf{E}\big\{\mathscr{A}(\beta(k))\Theta(k)\Theta(k)^T\mathscr{A}(\beta(k))^T \,\big|\, \Xi(k)=i, \Xi(k-1)=j\big\}$$

$$\times \mathbf{Pr}\{\Xi(k-1)=j \,|\, \Xi(k)=i\}$$

$$= \sum_{j\in\mathbb{B}} \mathbf{E}\big\{\mathscr{A}(\beta(k))\Theta(k)\Theta(k)^T\mathscr{A}(\beta(k))^T \,\big|\, \Xi(k)=i, \Xi(k-1)=j\big\}$$

$$\times \mathbf{Pr}\{\Xi(k)=i \,|\, \Xi(k-1)=j\}\mathbf{Pr}\{\Xi(k-1)=j\}/\mathbf{Pr}\{\Xi(k)=i\}$$

$$= \sum_{j\in\mathbb{B}} p_{ji}\overline{\mathscr{A}}_i \mathbf{E}\big\{\Theta(k)\Theta(k)^T \,\big|\, \Xi(k-1)=j\big\}(\overline{\mathscr{A}}_i)^T \frac{\mathbf{Pr}\{\Xi(k-1)=j\}}{\mathbf{Pr}\{\Xi(k)=i\}}.$$

Substitution of (6.46) into (6.45) and then into (6.44) provides the recursion

$$H_{k+1,i} = \sum_{j\in\mathbb{B}} p_{ji}\overline{\mathscr{A}}_i \mathbf{E}\big\{\Theta(k)\Theta(k)^T \,\big|\, \Xi(k-1)=j\big\}(\overline{\mathscr{A}}_i)^T \tag{6.47}$$

$$\times \mathbf{Pr}\{\Xi(k-1)=j\} + \overline{\mathscr{B}}_i W(\overline{\mathscr{B}}_i)^T \mathbf{Pr}\{\Xi(k)=i\}$$

$$= \sum_{j\in\mathbb{B}} p_{ji}\overline{\mathscr{A}}_i H_{k,j}(\overline{\mathscr{A}}_i)^T + \overline{\mathscr{B}}_i W(\overline{\mathscr{B}}_i)^T \mathbf{Pr}\{\Xi(k)=i\}$$

Since by assumption the system is MSS, it is also asymptotically wide-sense stationary [O. L. V. Costa, M. D. Fragoso, and R. P. Marques, 2005, Thm. 3.33]. Defining

$$H_i \triangleq \lim_{k\to\infty} H_{k,i}, \quad i \in \mathbb{B},$$

and recall that $\{\Xi(k)\}$, $k \in \mathbb{N}$ is aperiodic, then, (6.47) becomes (6.42), and (6.43) establishes (6.41). The above result quantifies the stationary covariance of the system state $\Theta(k)$. In view of (6.37) and the fact that $u(k) = v(k)^{(M)}$, Equation (6.41) can be directly used to evaluate the plant state and input covariances.

Remark 6.2.8. By using results in [Lancaster, 1970, Sec.5], the matrices H_i in (6.42) can be expressed in terms of the solution to a system of linear equations. To be more specific, if h_i is defined as the vectorized version of H_i and b_i as the vectorized version of the terms $\pi_i\overline{\mathscr{B}}_i W(\overline{\mathscr{B}}_i)^T$, $i \in \mathbb{B}$, then (6.42) leads to

$$h_i = b_i + \big(\overline{\mathscr{A}}_i \otimes \overline{\mathscr{A}}_i\big)\sum_{j\in\mathbb{B}} p_{ji}h_j, \quad i,j \in \mathbb{B},$$

from where h_i and thus H_i can be readily obtained. □

6.2.6 Simulation Study

In this example, the network under consideration will have $M = 10$ nodes. This simulation study is comparing the performance obtained by the implementations of the controller via Algorithm 3 with other two NCS architectures. In the first one, the controller and estimator are fixed at the actuator node. This system is described via:

$$\begin{aligned} x(k+1) &= Ax(k) + BL\hat{x}^a(k) + w(k), \\ \hat{x}^a(k) &= A\hat{x}^a(k-1) + Bu(k-1) \\ &\quad + \Gamma^{(10)}(k)K\big(y(k) - C(A\hat{x}^a(k-1) + Bu(k-1))\big), \end{aligned} \tag{6.48}$$

where $\Gamma^{(10)}(k)$ is as in (6.19). In the second configuration, the estimator is implemented at the sensor node, the controller is still at the actuator node. If the controller output is lost, then the previous plant input is held ([**?**]):

$$\begin{aligned} x(k+1) &= Ax(k) + \Gamma^{(10)}(k)BL\hat{x}^s(k) + (1 - \Gamma^{(10)}(k))Bu(k-1) + w(k), \\ \hat{x}^s(k) &= A\hat{x}^s(k-1) + B\hat{u}^s(k-1) \\ &\quad + K\big(y(k) - C(A\hat{x}^s(k-1) + B\hat{u}^s(k-1))\big), \end{aligned} \tag{6.49}$$

where $\Gamma^{(10)}(k)$ is as in (6.19) and

$$\hat{u}^s(k-1) = \begin{cases} u(k-1) & \text{if } \delta^{(1)}(k-1) = 1, L\hat{x}^s(k-1) \\ \text{if } \delta^{(1)}(k-1) = 0. \end{cases}$$

6.2.7 Independent and identically distributed dropouts

First of all, it is considered i.i.d. transmission processes as per Assumption 6.2.4. Figures 6.11 to 6.13 illustrate histograms of $c(k)$, obtained by running the algorithm for 1000 steps with dropout probabilities as indicated. Note the different scales used on the y-axes, and recall Proposition 6.2.5. Fig. 6.11 shows that, with the algorithm proposed, for smaller link transmission success probabilities p, control calculations are at most times, carried out

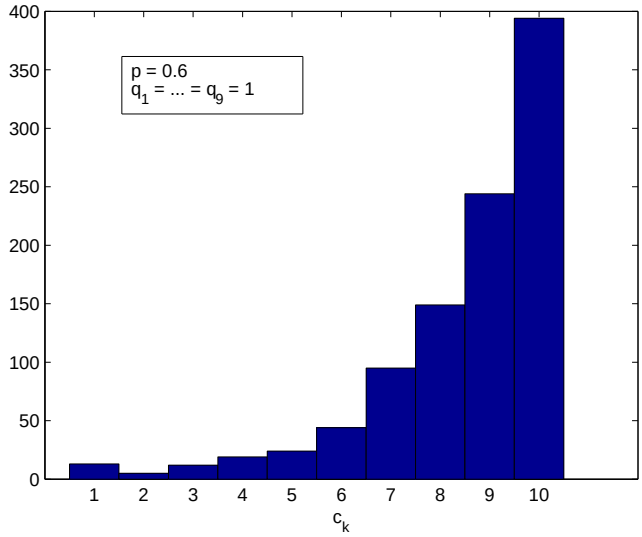

Figure 6.11: Histogram of the controller location $c(k)$ for an i.i.d. network with success probabilities $p = 0.6$ and $q_1 = \cdots = q_9 = 1$.

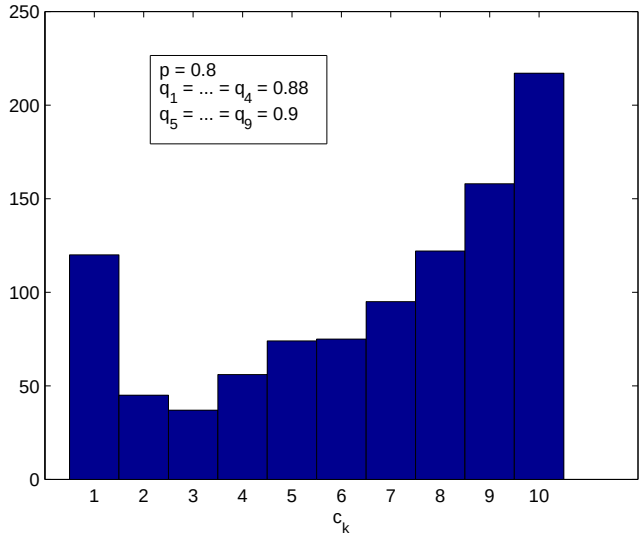

Figure 6.12: Histogram of the controller location $c(k)$ for an i.i.d. network with success probabilities $p = 0.8$, $q_1 = \cdots = q_4 = 0.88$, and $q_5 = \cdots = q_9 = 0.9$.

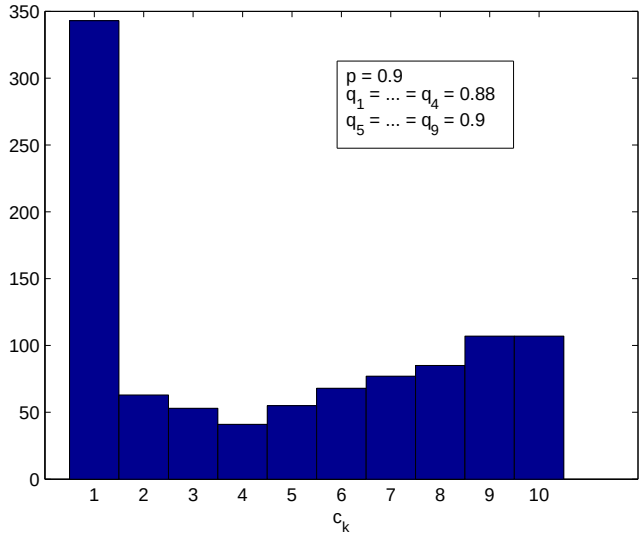

Figure 6.13: Histogram of the controller location $c(k)$ for an i.i.d. network with success probabilities $p = 0.9$, $q_1 = \cdots = q_4 = 0.88$, and $q_5 = \cdots = q_9 = 0.9$.

at the actuator node. On the other hand, if links are more reliable, then the controller will be placed at the sensor node at most time steps, see Fig. 6.13. In intermediate cases, $c(k)$ is more uniformly distributed, see Fig. 6.12.

To illustrate performance of the algorithms, it is considered first a plant model with an integrator, described by (6.11), where

$$A = \begin{bmatrix} 1.8 & -0.8 \\ 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, C = \begin{bmatrix} 0.047 & 0.044 \end{bmatrix}, \tag{6.50}$$

The noise covariances are given by $Q = 0.01 \times I_2$ and $R = 0.01$, and with Gaussian initial state having mean $\bar{x}(0) = [5 \quad 5]^T$ and covariance $P_0 = 0.1 \times I_2$. Controller and estimator gains L and K correspond to the steady state LQG/LQR controller with stage cost $\|x(k)\|^2 + \|u(k)\|^2/10$; see, [Bertsekas, 2005, Ch.5.2]. All nodes use as initial state estimates, $\hat{x}^{(i)}(0) = [0 \quad 0]^T$. The network has i.i.d. dropouts as per Assumption 6.2.4 with success probabilities $p = 0.9$ and $q_1 = \cdots = q_9 = 1$.

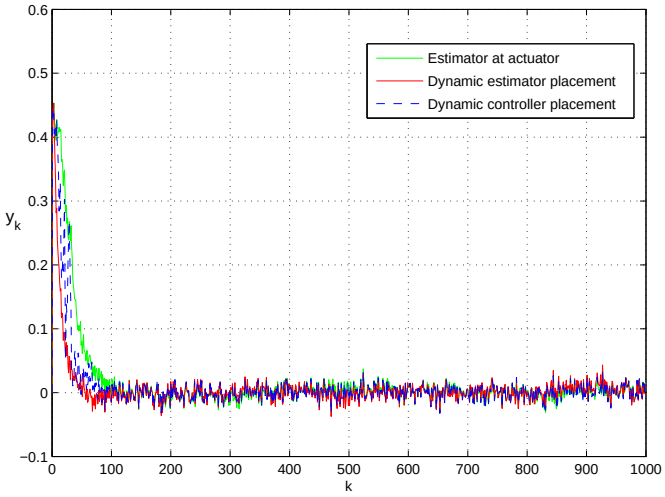

Figure 6.14: Output trajectory of the plant model (6.50) for an i.i.d. network with success probabilities $p = 0.9$, $q_1 = \cdots = q_9 = 1$ and the system in (6.50).

The baseline NCS with estimator placed at the sensor as in (6.49) failed to stabilize the system in the present case. Fig. 6.14 compares the plant output trajectory obtained by using the proposed algorithm with that provided by the baseline NCS configuration (6.48), and also with the algorithm presented in the previous work [D. Quevedo and Jurado, n.d.]. As can be appreciated in that figure, the adaptive estimator allocation algorithm presented reacts more quickly to plant outputs. It thereby recovers more quickly from the very bad local initial state estimates and provides control actions leading to faster convergence to the origin. If the empirical performance measure is adopted

$$J \triangleq \sum_{k=1}^{1000} y(k)^2, \tag{6.51}$$

then, with the dynamic architecture, $J \approx 1.4$, whereas for the baseline NCS described by (6.48), $J \approx 3.3$, for the baseline NCS in (6.49), $J \approx 1.2 \times 10^{114}$, and for the dynamic controller placement presented in [D. Quevedo and Jurado, n.d.], $J \approx 2.6$.

Table 6.1 illustrates how the performance gained by using the proposed method de-

pends upon the network reliability. For the situation examined, larger performance gains are obtained with smaller p. For larger p, the performance gains become less relevant. This finding is intuitive, since for $p \approx 1$ the network becomes transparent and overall performance is dominated by the nominal design (6.11)–(6.14).

Table 6.1: Performance indices J when controlling the system (6.50) over an i.i.d. network with $q_1 = \cdots = q_4 = 0.99$, $q_5 = \cdots = q_9 = 0.995$.

p	NCS (6.16)–(6.17)	NCS (6.48)	NCS (6.49)	Algorithm in [D. Quevedo and Jurado, n.d.]
0.8	6.81	16.15	(unstable)	10.21
0.85	1.51	4.15	(unstable)	2.65
0.9	1.23	2.99	(unstable)	1.86

6.2.8 Network with moving obstacle

This section is focused on a sensor-actuator network where there is an obstacle (e.g., a robot or crane) moving between four different positions, see Fig. 6.15. This situation is modeled in terms of the network model used in Section 6.2.5, using the network state process $\Xi(k) \in \mathbb{B} = \{1,2,3,4\}$. The transition probabilities for $\Xi(k)$ are given by:

$$[p_{ij}] = \begin{bmatrix} 0.99 & 0.01 & 0 & 0 \\ 0.003 & 0.99 & 0.007 & 0 \\ 0 & 0.003 & 0.99 & 0.007 \\ 0.007 & 0 & 0.003 & 0.99 \end{bmatrix}.$$

The individual link reliabilities depend on the position of the obstacle in the network. Nodes which are not blocked benefit from high success probabilities, namely 99%. However, due to the obstacle, some of the success probabilities will, at times, be lowered to

60%:

$$\mathbf{Pr}\{\gamma^{(i)}(k)=1\,|\,\Xi(k)=j\}=\begin{cases}0.6 & \text{if } i\in\{2j-1,2j,2j+1\}\\ 0.99 & \text{in all other cases}\end{cases}$$

$$\mathbf{Pr}\{\delta^{(i)}(k-1)=1\,|\,\Xi(k)=j\}=\begin{cases}0.6 & \text{if } i\in\{2j,2j+1\}\\ 0.99 & \text{in all other cases.}\end{cases}$$

Figs. 6.16 and 6.17 illustrate how using Algorithm 3 the controller location depends upon the network state $\Xi(k)$. It turns out that, in the present case, the plant input is always provided by one of the nodes located between the sensor node and the node immediately following the blocked ones. This behaviour can be explained by noting that, in absence of the obstacle, the network is very reliable. In fact, if none of the nodes were blocked, then the algorithm would (almost) always locate the controller at the sensor node. Fig. 6.18 documents the associated histogram.

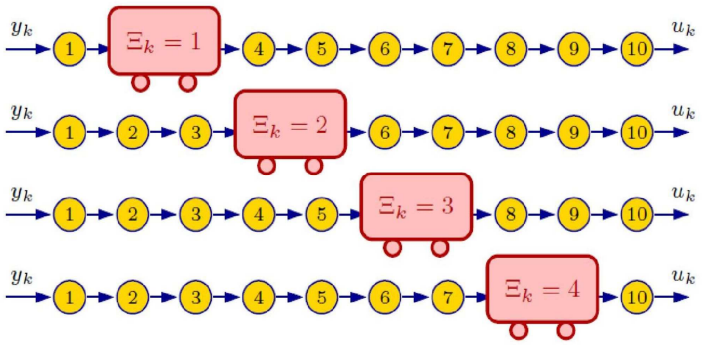

Figure 6.15: Sensor-actuator network with moving obstacle.

For the plant model (6.50), using the dynamic architecture proposed in the present work, gave a performance index of $J = 1.8$, see (6.51). In contrast, with the controller

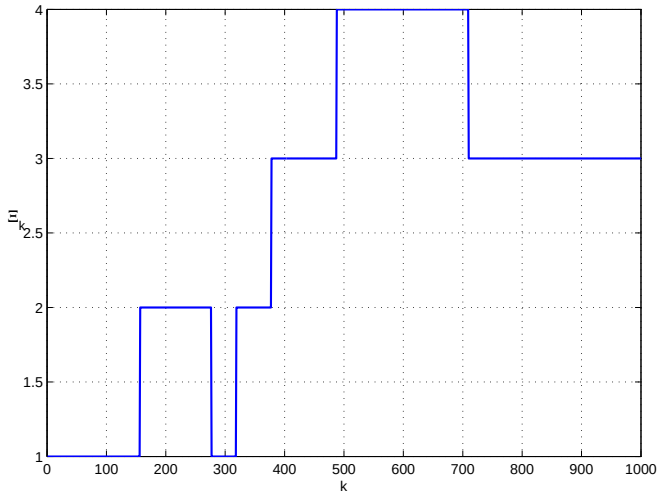

Figure 6.16: Network State trajectory, $\Xi(k)$.

at the actuator node, see (6.48), $J = 4.7$ is obtained, and with the architecture presented in [D. Quevedo and Jurado, n.d.], $J = 2.9$. In the situations examined, positioning the controller at the sensor node, see (6.49), failed to stabilize the plant model.

6.3 Conclusions

In the first section, a flexible cooperative MPC formulation for NCSs subject to data dropouts has been presented. Also, an algorithm that decides which nodes are in charge of the calculation of the the control input, and which ones just relay the received information, has been provided. This decision depends on tranmission outcomes. Once the controller node has been chosen, it interacts with its neighbors solving a cooperative MPC, which is also subject to data dropouts. Future works may include some stability analysis of the proposed architecture, as well as an extension to more than two cooperative nodes.

The second section has presented a flexible architecture for the implementation of an estimated state feedback control law over a wireless sensor-actuator network with analog

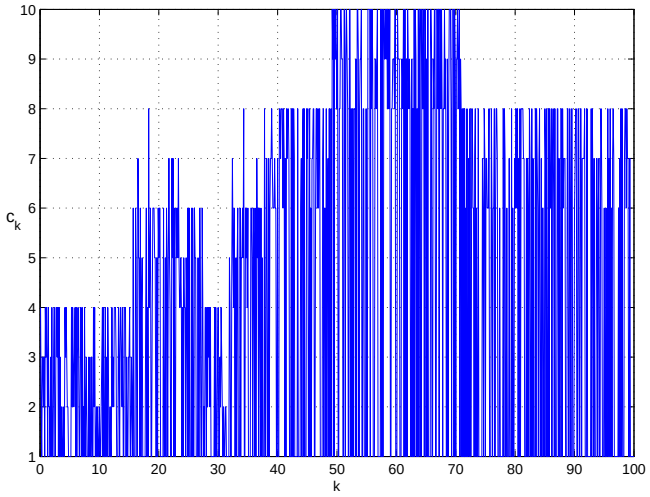

Figure 6.17: Controller location $c(k)$ for the network in Fig. 6.15.

erasure channels without acknowledgments. As with the previous one, with the algorithm provided, the role played by individual nodes depends on transmission outcomes. In particular, the estimator location depends upon the availability of past plant input values and transmission outcomes. By deriving a Markovian jump-linear system model, a closed form expression for the stationary covariance of the system state in the presence of correlated dropout processes has been established. Future work may include extending the ideas presented to the control of multiple-loops, to general network topologies, and to controller design.

6.4 Related publications

- Isabel Jurado, Daniel E. Quevedo, Karl H. Johansson and Anders Ahlén. *Cooperative Dynamic MPC for Networked Control Systems*.

  Book: *Distributed MPC Made Easy*

  Editors: Dr. José M. Maestre and Dr. Rudy R. Negenborn.

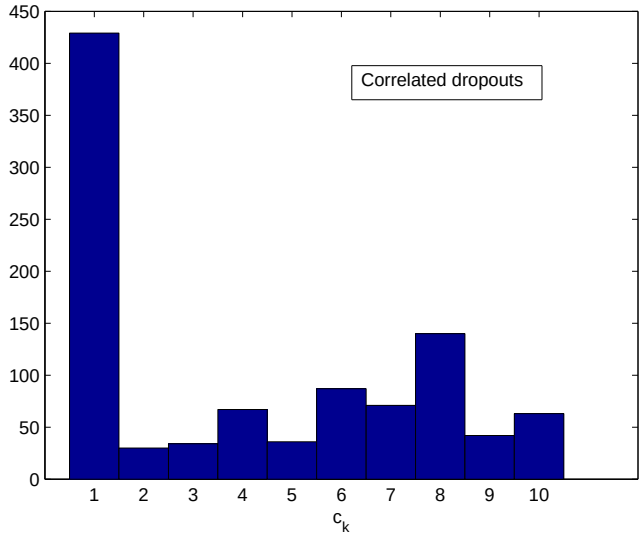

Figure 6.18: Histogram of $c(k)$ for the network in Fig. 6.15.

- Daniel Quevedo, Karl H. Johansson, Anders Ahlén and Isabel Jurado. *An Adaptive Architecture for Control using Erasure Channels*. Submitted to Automatica.

Chapter 7

Distributed estimation in networked systems under periodic and event-based communication policies

This chapter's aim is to present a novel design technique for distributed estimation in networked systems. The problem assumes a network of interconnected agents each one having partial access to measurements from a linear plant and broadcasting their estimations to their neighbors. The objective is to reach a reliable estimation of the plant state from every agent location. The observers structure implemented in each agent is based on local Luenberger-like observers in combination with consensus strategies. The chapter focuses on the following network related issues: delays, packet dropouts, and communication policy (time and event-driven). The design problem is solved via linear matrix inequalities and stability proofs are provided. The problem is of application for sensor networks and large scale systems where centralized estimation schemes are not advisable and energy-aware implementations are of interest.

The design of the observers contemplates the possibility of sharing only a part of the estimated state between neighbors, instead of communicating the whole estimated vector state. This economy in the use of network resources is, by its own nature, further improved with the event-driven communication approach.

Simulation examples are provided to show the performance of the proposed methodologies.

7.1 Problem description and motivation

Consider a sensor network intended to estimate the state of a linear plant in a distributed way, where the sensors measure some variables (outputs), compute a local estimation of the overall state of the system, and broadcast to a set of neighbors some information related with their own estimations. As Figure 7.1 illustrates, the set of nodes are connected by means of a communication network, which may introduce delays and packet dropouts. When the local information received for each of the different observers is not sufficient to estimate the complete state of the plant, then the proposed type of distributed observation makes sense.

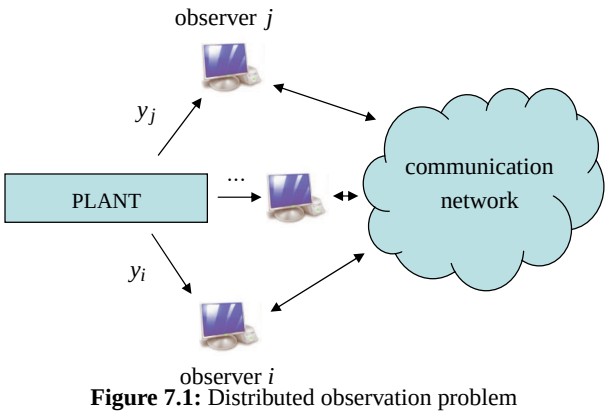

Figure 7.1: Distributed observation problem

The concepts of *local observability* and *collective observability* refer, respectively, to a situation in which the measurement performed by any sensor is sufficient to guarantee observability of the process state; and to a situation in which all the sensors, if put together, guarantee this property. See [Olfati-Saber, 2007] for a complete explanation. In this chapter it is assumed that all the sensors must estimate the overall state of the system even when

local observability does not hold.

To motivate the problem, consider a possible application where the state of a plant is monitored from different geographically distributed locations, provided that only some local information of the plant can be directly measured from each location. This scenario might consist of a number of observers having access to some, generally different, plant outputs. The plant is not necessarily fully observable from any of the observers. The different observers are able to communicate themselves by sharing information with a set of neighbors in order to estimate the complete state of the plant. The communication among the different observers is assumed to be implemented through a communication network, in which time-delays and possible packet dropouts have to be taken into consideration. A different situation in which the framework considered in this work might be of application could be that in which the observers are connected using a shared medium, managing traffic information in a urban environment.

Taking into account the aforementioned ideas, the present chapter focuses on network-related issues, specifically communication efficiency and robustness against problems induced by the network. The chapter provides an observer design method to operate with time-driven communication between the agents, being the objective to reach a common reliable estimate of the system state, despite of the presence of delays and dropouts. It is also proposed to include an event-based communication strategy between agents, aiming at reducing the traffic over the network and the energy consumption.

In the latter case, the estimation error will eventually enter into an arbitrary small region around the equilibrium point. The size of that region depends on a free parameter that sets the threshold triggering the communication events, which allows to trade off between communication savings and estimation performance.

7.1.1 Network topology

The communication network is represented using a directed graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, with $\mathscr{V} = \{1, 2, ..., p\}$, being the set of nodes (observers) of the graph (network), and $\mathscr{E} \subset \mathscr{V} \times \mathscr{V}$, being the set of links. Assuming the cardinality of $\mathscr{E}$ equal l, and defining $\mathscr{L} = \{1, 2, ..., l\}$, it is obvious that a bijective function $g : \mathscr{E} \to \mathscr{L}$ can be built so that a given link can be either referenced by the pair of nodes that connects $(i, j) \in \mathscr{E}$ or the link index $r \in \mathscr{L}$, so that $r = g(i, j)$. The set of nodes connected to node i is termed the neighborhood of i, and denoted as $\mathscr{N}_i \triangleq \{j \in \mathscr{V} | (i, j) \in \mathscr{E}\}$. Directed communications are considered, so that link (i, j) implies that node i receives information from node j.

7.1.2 System description

In this work, the system to be observed is assumed to be an autonomous linear time-invariant plant given by the following equations:

$$x(k+1) = Ax(k), \tag{7.1}$$

$$y_i(k) = C_i x(k), \quad \forall i \in \mathscr{V}, \tag{7.2}$$

where $x(k) \in \mathbb{R}^n$ is the state of the plant and $y_i(k) \in \mathbb{R}^{m_i}$ are the system outputs. In general, each of the different p observers has access to a distinct output of the plant. Collective observability is assumed, i.e. the pair (A, C) is observable, where C is a matrix stacking the output matrices C_i of all the agents.

Furthermore, the observers can communicate with each other by means of a communication network that can be represented by the graph $\mathscr{G}$. More precisely, each neighbor j of the observer i communicate some estimated outputs $\hat{y}_{ij} = C_{ij}\hat{x}_j$. It is assumed that node i knows the matrix C_{ij} corresponding to the output $\hat{y}_{ij}$. Exchanging estimates instead of the measurements from the plant provides some freedom and flexibility to choose the information sent through the network. That way, taking into account the plant dynamics and the output measured by a particular node, it is possible to collect only the information from its

neighbors that allows estimation, leading to a policy in which only relevant information for each agent is transmitted.

Let us define $\bar{C}_i$ as a matrix stacking the matrix C_i and matrices C_{ij} for all $j \in \mathcal{N}_i$. It is assumed that each pair $(A, \bar{C}_i)$ is observable. This is a necessary condition that impose some restrictions on the network topology and the information that is sent via each connection.

7.2 Periodic time-driven communication between agents

This section is devoted to the observer design method under periodic communication between agents. Next, a description of the node dynamics is explained in detail.

7.2.1 Node dynamics

The communication between agents may be affected by delays and packet dropouts. Therefore, it is convenient that the observer to be proposed takes both effects under consideration. Figure 7.2 illustrates a possible time scheduling in which both effects appears. Let $\tau_{ij}(k) \in \mathbb{N}$ represent the time difference between the current time instant k and the instant in which the last packet received by node i was sent from its neighbor j. This constant includes the effect of delays and packet dropouts. Note that packet dropouts have the effect of enlarging $\tau_{ij}(k)$.

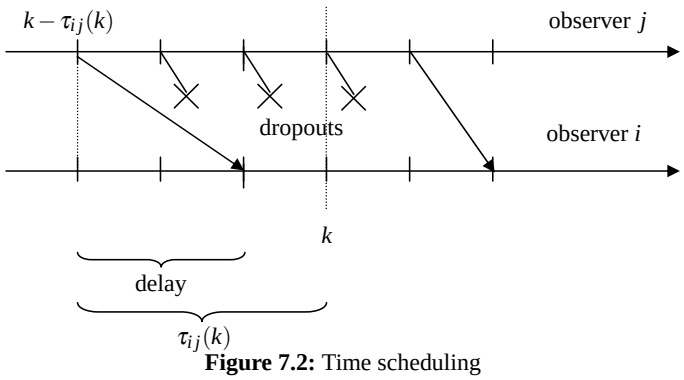

Figure 7.2: Time scheduling

Assuming that the number of consecutive data dropouts is bounded by n_p and that the effective network-induced delays are also bounded by $d_{\min}$ and $d_{\max}$, it is obvious that $\tau_{ij}(k)$ belongs to the interval $[\tau_m, \tau_M]$, where

$$\tau_m = d_{\min}, \tag{7.3}$$

$$\tau_M = n_p + d_{\max}. \tag{7.4}$$

Figure 7.3 illustrates a characteristic shape for $\tau_{ij}(k)$. For the sake of simplicity in the technical developments to come, it is assumed that the minimum delay bound is exactly zero, $\tau_m = 0$, and the maximum delay will be denoted by τ_M. In other works as [Millán et al., 2012] or [Orihuela et al., 2011], this assumption is relaxed, leading to a more cumbersome, but still solvable, design problems.

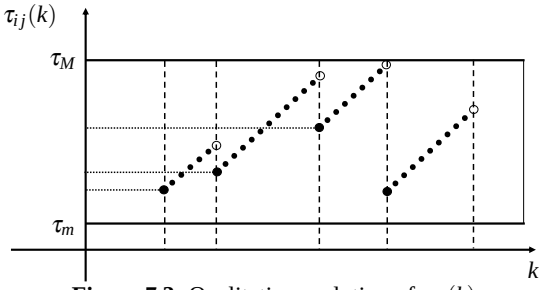

Figure 7.3: Qualitative evolution of $\tau_{ij}(k)$

Also note that each τ_{ij} is directly associated to a link $(i, j) \in \mathscr{E}$. Then, it is possible to establish a relation between each connection and the corresponding τ_{ij}. Numbering the links from 1 to l the following equivalent notation can be used:

$$\tau_r(k) = \tau_{ij}(k),\ r = 1, ..., l, \tag{7.5}$$

where $r = g(i, j)$. That is, τ can refer either to a pair of nodes (τ_{ij}) or to a link (τ_r).

Once the considerations about delays and packet dropouts have been done, structure for the observers given by the following equation is proposed:

$$\hat{x}_i(k+1) = A\hat{x}_i(k) + M_i(C_i\hat{x}_i(k) - y_i(k)) + \tag{7.6}$$

$$\sum_{j\in\mathcal{N}_i} N_{ij}(C_{ij}\hat{x}_j(k-\tau_{ij}(k)) - C_{ij}\hat{x}_i(k-\tau_{ij}(k))),$$

for $i \in \mathcal{V}$. The structure of the observers comprises two main parts, namely

- A local Luenberger-like observer, weighted with matrices M_i, that corrects the estimated state of the plant based on the measured output $y_i(k)$ accessible for each observer i.

- A consensus-based observer, weighted with matrices N_{ij}, which takes into account the information received from neighboring observers.

The name consensus comes from the fact that all the nodes will eventually achieve the same value of the estimated state. Note that the node i must known the exact value of the actual artificial delay $\tau_{ij}(k)$, as it needs to compare the received information $C_{ij}\hat{x}_j(k-\tau_{ij}(k))$ with past values of its own estimated state $C_{ij}\hat{x}_i(k-\tau_{ij}(k))$. To do that, the nodes must be synchronized. Assuming that some kind of synchronization algorithm is running, the delay $\tau_{ij}(k)$ can be known by adding a timestamp to every data packet. Furthermore, each node must buffer all its past estimates until instant $k-\tau_M$.

Let us consider now the observation error of a generic observer i defined as $e_i(k) = \hat{x}_i(k) - x(k)$, i.e. the difference between the estimation of node i and the state of the plant. Taking into account equations (7.1) and (7.6), the dynamics of the observation error can be written as:

$$e_i(k+1) = (A+M_iC_i)e_i(k) + \sum_{j\in\mathcal{N}_i} N_{ij}C_{ij}(e_j(k-\tau_{ij}(k)) - e_i(k-\tau_{ij}(k))) \tag{7.7}$$

Considering that the number of observers is given by p, the dynamic equations of the observation errors can be written in a compact form defining a stacked error vector as $e^T(k) = [e_1^T(k) \quad e_2^T(k) \quad \dots \quad e_p^T(k)]$:

$$e(k+1) = \Phi(\mathscr{M})e(k) + \Lambda(\mathscr{N})d(k) \tag{7.8}$$

where $d(k)$ is a delayed version of the stacked error vector taking into account the delays of the different links (see equation (7.5)) $d^T(k) = [e^T(k-\tau_1(k)) \ldots e(k-\tau_l(k))^T]$, or equivalently the delays of the communications between neighbors i and j. The matrices $\Phi(\mathscr{M})$ and $\Lambda(\mathscr{N})$ depend on the sets of observers to be designed: $\mathscr{M} = \{M_i, i \in \mathscr{V}\}$ and $\mathscr{N} = \{N_{ij}, i \in \mathscr{V}, j \in \mathscr{N}_i\}$. It is not difficult to see that the structure of such matrices are given by:

$$\Phi(\mathscr{M}) = \begin{bmatrix} A+M_1C_1 & 0 & \cdots & 0 \\ 0 & A+M_2C_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A+M_pC_p \end{bmatrix} \tag{7.9}$$

$$\Lambda(\mathscr{N}) = [\Lambda_1 \, \Lambda_2 \, \cdots \, \Lambda_l] \tag{7.10}$$

where Λ_r, $r = g(i,j) \in \{1,...,l\}$, are block matrices in correspondence with each of the links r communicating the observer i with j, in which the only blocks different from zero are $-N_{ij}C_{ij}$ and $N_{ij}C_{ij}$ in the (i,i) and (i,j) positions respectively:

$$\Lambda_r = \begin{array}{c} \begin{matrix} \textit{column} & & i & & j & & \end{matrix} \\ \begin{bmatrix} 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \cdots & -N_{ij}C_{ij} & \cdots & N_{ij}C_{ij} & \cdots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \end{bmatrix} \end{array} \begin{array}{l} \\ \\ \textit{row } i \\ \\ \\ \end{array}$$

7.2.2 Observers design

In the following the main result of this section is introduced. For periodic communication between agents, next theorem states a sufficient condition for the asymptotic convergence of the estimates of each observer to the plant state.

Theorem 7.2.1. *Given τ_M according to equation* (7.4)*, if the nonlinear matrix inequality (7.11) has a feasible solution for positive definite matrices Z_1, Z_2, P_i, $i \in \mathscr{V}$, and observers matrices M_i, N_{ij}, $i \in \mathscr{V}$, $j \in \mathscr{N}_i$, then the estimations of all the observers asymptotically converge to the plant state.*

$$\begin{bmatrix} \Xi & \Theta & 0 & \Phi^T(\mathscr{M})P & (\Phi^T(\mathscr{M})-I)P\tau_M \\ * & \Psi & \Theta^T & \Lambda^T(\mathscr{N})P & \Lambda^T(\mathscr{N})P\tau_M \\ * & * & \Omega & 0 & 0 \\ * & * & * & -P & 0 \\ * & * & * & * & -\frac{1}{l}PZ_2^{-1}P \end{bmatrix} < 0, \tag{7.11}$$

where:

$$P = diag(P_1, P_2, ..., P_p),$$
$$\Xi = -P + Z_1 - lZ_2,$$
$$\Theta = \overbrace{[Z_2 \quad Z_2 \quad \cdots \quad Z_2]}^{l \text{ times}},$$
$$\Psi = diag\overbrace{(-2Z_2, \cdots, -2Z_2)}^{l \text{ times}},$$
$$\Omega = -Z_1 - lZ_2.$$

Proof: Choose the following Lyapunov-Krasovskii functional:

$$V(k) = e^T(k)Pe(k) + \sum_{i=k-\tau_M}^{k-1} e^T(i)Z_1e(i) + l \times \tau_M \sum_{j=-\tau_M+1}^{0} \sum_{i=k+j-1}^{k-1} \Delta e^T(i)Z_2\Delta e(i), \tag{7.12}$$

where $\Delta e(k) = e(k+1) - e(k)$. Note that the third term is included l times, one for each communication link. The forward difference can be calculated as

$$\begin{aligned} \Delta V(k) &= e^T(k+1)Pe(k+1) - e^T(k)Pe(k) + e^T(k)Z_1e(k) - e^T(k-\tau_M)Z_1e(k-\tau_M) \\ &\quad + l \times \tau_M^2 \Delta e^T(k)Z_2\Delta e(k) - l \times \tau_M \sum_{j=k-\tau_M}^{k-1} \Delta e^T(j)Z_2\Delta e(j) \\ &= \begin{bmatrix} e^T(k) & d^T(k) \end{bmatrix} \begin{bmatrix} \Phi^T(\mathscr{M}) \\ \Lambda^T(\mathscr{N}) \end{bmatrix} P \begin{bmatrix} \Phi(\mathscr{M}) & \Lambda(\mathscr{N}) \end{bmatrix} \begin{bmatrix} e(k) \\ d(k) \end{bmatrix} \\ &\quad + \begin{bmatrix} e^T(k) & e^T(k-\tau_M) \end{bmatrix} \begin{bmatrix} Z_1 - P & 0 \\ 0 & -Z_1 \end{bmatrix} \begin{bmatrix} e(k) \\ e(k-\tau_M) \end{bmatrix} \\ &\quad + l \times \tau_M^2 \Delta e^T(k)Z_2\Delta e(k) - l \times \tau_M \sum_{j=k-\tau_M}^{k-1} \Delta e^T(j)Z_2\Delta e(j). \end{aligned}$$

Defining the augmented state vector

$$\xi(k) = \begin{bmatrix} e(k) \\ e(k-\tau_1(k)) \\ e(k-\tau_2(k)) \\ \vdots \\ e(k-\tau_l(k)) \\ e(k-\tau_M) \end{bmatrix} = \begin{bmatrix} e(k) \\ d(k) \\ e(k-\tau_M) \end{bmatrix},$$

the forward difference of the Lyapunov-Krasovskii functional can be written using the following quadratic form:

$$\Delta V(k) = \xi^T(k) \left(\begin{bmatrix} Z_1 - P & 0 & 0 \\ * & 0 & 0 \\ * & * & -Z_1 \end{bmatrix} + \begin{bmatrix} \Phi^T(\mathscr{M}) \\ \Lambda^T(\mathscr{N}) \\ 0 \end{bmatrix} P \begin{bmatrix} \Phi(\mathscr{M}) & \Lambda(\mathscr{N}) & 0 \end{bmatrix} \right.$$

$$\left. + l \times \tau_M^2 \begin{bmatrix} \Phi^T(\mathscr{M}) - I \\ \Lambda^T(\mathscr{M}) \\ 0 \end{bmatrix} Z_2 \begin{bmatrix} (\Phi(\mathscr{M}) - I) & \Lambda(\mathscr{M}) & 0 \end{bmatrix} \right) \xi(k)$$

$$- l \times \tau_M \sum_{j=k-\tau_M}^{k-1} \Delta e^T(j) Z_2 \Delta e(j).$$

In order to take into account the delay of each different communication link ($\tau_r(k), \forall r = 1,...,l$), the last term in the above equation (which appears l times) in 2 terms is split, considering the delay in each specific link:

$$-\tau_M \sum_{j=k-\tau_M}^{k-1} \Delta e^T(j) Z_2 \Delta e(j) = -\tau_M \sum_{j=k-\tau_M}^{k-\tau_r(k)-1} \Delta e^T(j) Z_2 \Delta e(j) - \tau_M \sum_{j=k-\tau_r(k)}^{k-1} \Delta e^T(j) Z_2 \Delta e(j).$$

The resulting terms can be bounded using the Jensen inequality:

$$-\tau_M \sum_{j=k-\tau_M}^{k-\tau_r(k)-1} \Delta e^T(j) Z_2 \Delta e(j) \leq -\left[\sum_{j=k-\tau_M}^{k-\tau_r(k)-1} \Delta e(j) \right]^T Z_2 \left[\sum_{j=k-\tau_M}^{k-\tau_r(k)-1} \Delta e(j) \right],$$

$$-\tau_M \sum_{j=k-\tau_r(k)}^{k-1} \Delta e^T(j) Z_2 \Delta e(j) \leq -\left[\sum_{j=k-\tau_r(k)}^{k-1} \Delta e(j) \right]^T Z_2 \left[\sum_{j=k-\tau_r(k)}^{k-1} \Delta e(j) \right].$$

The terms in brackets can be cancelled in pairs, except the first and the last one in the summatory, yielding:

$$-\tau_M \sum_{j=k-\tau_M}^{k-\tau_r(k)-1} \Delta e^T(j) Z_2 \Delta e(j) \leq$$

$$-\left[e(k-\tau_r(k)) - e(k-\tau_M)\right]^T Z_2 \left[e(k-\tau_r(k)) - e(k-\tau_M)\right],$$

$$-\tau_M \sum_{j=k-\tau_r(k)}^{k-1} \Delta e^T(j) Z_2 \Delta e(j) \leq -\left[e(k) - e(k-\tau_r(k))\right]^T Z_2 \left[e(k) - e(k-\tau_r(k))\right].$$

The above terms are also written in the same quadratic manner:

$$-\tau_M \sum_{j=k-\tau_M}^{k-\tau_r(k)-1} \Delta e^T(j) Z_2 \Delta e(j) \leq$$

$$\begin{bmatrix} e^T(k-\tau_r(k)) & e^T(k-\tau_M) \end{bmatrix} \begin{bmatrix} -Z_2 & Z_2 \\ * & -Z_2 \end{bmatrix} \begin{bmatrix} e(k-\tau_r(k)) \\ e(k-\tau_M) \end{bmatrix},$$

$$-\tau_M \sum_{j=k-\tau_r(k)}^{k-1} \Delta e^T(j) Z_2 \Delta e(j) \leq$$

$$\begin{bmatrix} e^T(k) & e^T(k-\tau_r(k)) \end{bmatrix} \begin{bmatrix} -Z_2 & Z_2 \\ * & -Z_2 \end{bmatrix} \begin{bmatrix} e(k) \\ e(k-\tau_r(k)) \end{bmatrix}.$$

Including all the terms, the forward difference is:

$$\Delta V(k) = \xi^T(k) L_1 \xi(k)$$

where

$$L_1 = \xi^T(k) \left(\begin{bmatrix} \Xi & \Theta & 0 \\ * & \Psi & \Theta^T \\ * & * & \Omega \end{bmatrix} + \begin{bmatrix} \Phi^T(\mathscr{M}) \\ \Lambda^T(\mathscr{N}) \\ 0 \end{bmatrix} P \begin{bmatrix} \Phi(\mathscr{M}) & \Lambda(\mathscr{N}) & 0 \end{bmatrix} \right. \tag{7.13}$$

$$\left. + l \times \tau_M^2 \begin{bmatrix} \Phi^T(\mathscr{M}) - I \\ \Lambda^T(\mathscr{M}) \\ 0 \end{bmatrix} Z_2 \begin{bmatrix} (\Phi(\mathscr{M}) - I) & \Lambda(\mathscr{M}) & 0 \end{bmatrix} \right) \xi(k)$$

In order to show the error convergence to zero, it will be demonstrated that $\Delta V(k) < 0$ for all $\xi(k) \neq 0$ through the negative definiteness of L_1. Applying Schur complements, one

can obtain that previous matrix is negative definite if and only if the following holds:

$$\begin{bmatrix} \Xi & \Theta & 0 & \Phi^T(\mathscr{M}) & (\Phi^T(\mathscr{M})-I)\tau_M \\ * & \Psi & \Theta^T & \Lambda^T(\mathscr{N}) & \Lambda^T(\mathscr{N})\tau_M \\ * & * & \Omega & 0 & 0 \\ * & * & * & -P^{-1} & 0 \\ * & * & * & * & -\frac{1}{l}Z_2^{-1} \end{bmatrix} < 0$$

And the negative definiteness of this matrix is equivalent to that of the matrix in Theorem 7.2.1, which can be obtained by pre- and post- multiplying previous matrix by $diag(I,I,I,P,P)$ and its transpose. □

As it is clearly seen from (7.11), the matrix inequality to be solved in order to design the observers is not linear because of the presence of the terms $\Phi^T(\mathscr{M})P$, $\Lambda^T(\mathscr{N})P$ and $PZ_2^{-1}P$.

The two first nonlinearities, related to $\Phi^T(\mathscr{M})P$, $\Lambda^T(\mathscr{N})P$, can be trivially settled by defining $M_iP_i = W_i$ and $N_{ij}P_i = X_{ij}$. That way, those terms are now a function of the new matrices in the change of variables, i.e, $\Phi^T(\mathscr{M})P \to \Phi(\mathscr{W})$ and $\Lambda^T(\mathscr{N})P \to \Lambda(\mathscr{X})$, where the sets are defined as $\mathscr{W} = \{W_i, i \in \mathscr{V}\}$ and $\mathscr{X} = \{X_{ij}, i \in \mathscr{V}, j \in \mathscr{N}_i\}$.

In the following subsections, two solutions in order to deal with the nonlinearity $PZ_2^{-1}P$ are presented. The first one introduces an additional constraint which let us address the problem by means of a set of linear matrix inequalities. The second solution employs the cone complementary algorithm to transform the nonlinear inequality into an iterative optimization problem with linear constraints. Comparing both solutions, the former could be more conservative, but it is computationally more efficient, as the number of constraints and variables is lower.

Constraint on $PZ_2^{-1}P$

A method to deal with this nonlinearity consists in introducing the following additional constraint:

$$-PZ_2^{-1}P < -\frac{1}{\mu}P,$$

being μ a positive design scalar. Note that previous condition is equivalent to $Z_2 < \mu P$. Then, the nonlinear constraint in Theorem 7.2.1 can be replaced by:

$$\begin{cases} \Upsilon < 0, \\ Z_2 < \mu P \end{cases} \tag{7.14}$$

where Υ is the matrix required to be negative definite in (7.11), but substituting the terms $\Phi^T(\mathscr{M})P$, $\Lambda^T(\mathscr{N})P$ and $PZ_2^{-1}P$ by $\Phi^T(\mathscr{W})$, $\Lambda^T(\mathscr{X})$ and $\frac{1}{\mu}P$ respectively.

It is worth comparing the proposed method with the one introduced in [D. Yue, Q. H. and J. Lam, 2005] and used in other works to handle the same nonlinearity. While in [D. Yue, Q. H. and J. Lam, 2005] it is directly imposed Z_2 to be P times a given scalar, our method just restricts $Z_2 < \mu P$, which covers a much wider range of possible solutions in the space of positive definite matrices.

Cone complementary algorithm

As it has been done in other works, an extended procedure (see [El Ghaoui et al., 1997]) can be also adapted, which let us address the nonlinearity $PZ_2^{-1}P$ by introducing some new matrix variables and constraints. Following the same steps than [Moon et al., 2001], the original inequality (7.11) can be replaced by the following nonlinear minimization problem involving LMI conditions:

$$\text{Minimize}\,\mathbf{Tr}\left(\hat{P}P + \hat{Z}_2 Z_2 + \hat{T}_2 T_2\right) \tag{7.15}$$

subject to

$$\begin{cases} \Upsilon < 0, \\ \begin{bmatrix} -\hat{T}_2 & \hat{P} \\ * & -\hat{Z}_2 \end{bmatrix} \leq 0, \begin{bmatrix} P & I \\ * & \hat{P} \end{bmatrix} \geq 0, \begin{bmatrix} \hat{Z}_2 & I \\ * & Z_2 \end{bmatrix} \geq 0, \begin{bmatrix} T_2 & I \\ * & \hat{T}_2 \end{bmatrix} \geq 0, \end{cases} \tag{7.16}$$

where Υ is as before the matrix required to be definite negative in (7.11), but substituting $PZ_2^{-1}P$ by T_2. The set of additional linear constraints and variables needs to be introduced to ensure the convergence of the solution.

In order to solve the aforementioned minimization problem (7.15) the algorithm introduced in [El Ghaoui et al., 1997] can be implemented. It has been omitted here.

Remark. The computational burden required to solve the LMIs (7.14) and (7.16) directly depends on the dimension of the system, the number of agents, and the number of links between them. Although the implementation of the distributed estimation scheme is completely distributed, the design procedure is centralized, as all the weighting matrices are designed together. With respect to centralized schemes, the number of links is now the bottleneck of all the proposed solutions in the literature. It would be of undeniable interest to distribute the mathematical calculus, in such a way that each node does not need information of the rest of the nodes, but only of its neighbours, to design its observer. This is matter of future research.

7.3 Event-based communication between agents

This section analyzes an asynchronous event-based communication policy between agents to reduce the energy consumption and make an efficient use of the network resources. The event-based communication is a means to reduce the information exchange rates between the components in the network by triggering the communication only after an event has indicated that a certain relevant variable exceeds a tolerable pre-defined threshold. Next, the event-based implementation of the observer designed previously is studied in detail.

From a modeling point of view, the main difference between the time-driven scheme in Section 7.2 and the event-driven paradigm described here is the non-uniform pattern of transmission of information.

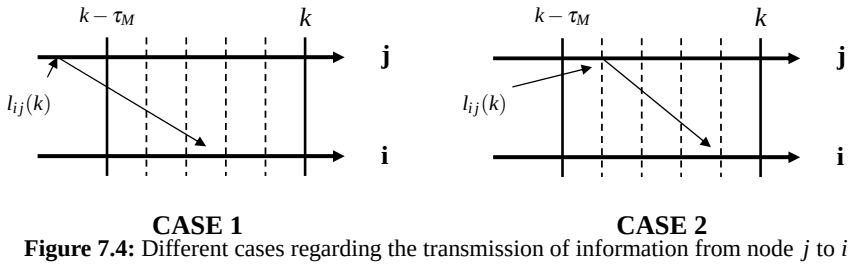

Figure 7.4: Different cases regarding the transmission of information from node j to i

Transmissions are now assumed to be triggered at specific time instants, when a triggering condition is satisfied. Let $l_{ij}(k)$ denote the time instant when node j sent the more recent packet available for the node i at current time instant k. Then, $\{l_{ij}(k)\} \subset \mathbb{N}$, as node j only sends packets when an event is triggered.

Triggering condition: Given a threshold δ, at instant k node j broadcasts its estimates to every agent i such that $j \in \mathcal{N}_i$ if

$$\|\hat{x}_j(l_{ij}(k)) - \hat{x}_j(k)\|_\infty \geq \delta, \quad \text{for } k > l_{ij}(k). \tag{7.17}$$

In this section, packet dropouts are not considered, so only delays affect the communication between agents.

7.3.1 Remodelling of node dynamics

This section introduces the modifications needed to remodel the system dynamic equations according to the approach introduced above. Consider a generic agent i at time instant k. As Figure 7.4 suggests, there are two possible situations with respect to the information received from each of its neighbours $j \in \mathcal{N}_i$:

- *Case 1:* The last packet received in node i was sent before $k-\tau_M$. It is obvious that $l_{ij}(k) < k-\tau_M$. In this case, node i does not have in memory[1] the estimate $\hat{x}_i(l_{ij}(k))$.

[1] Recall that each node stores only a finite amount of past estimates, as was explained in Section 7.2.

Then, it compares its older buffered state, that is $\hat{x}_i(k-\tau_M)$, with the available state of its neighbour $\hat{x}_j(l_{ij}(k))$ to correct its estimation.

- *Case 2:* The last packet received in node i was sent after $k-\tau_M$, so $l_{ij}(k) \geq k-\tau_M$. Given that the estimate $\hat{x}_i(l_{ij}(k))$ is still in the buffer of the node i, it compares it with the received information $\hat{x}_j(l_{ij}(k))$.

Taking into account these cases, the dynamics of a generic node i can be rewritten as

$$\hat{x}_i(k+1) = A\hat{x}_i(k) + M_i(\hat{y}_i(k) - y_i(k)) \tag{7.18}$$

$$+ \sum_{j\in\mathcal{N}_i} N_{ij}C_{ij}\left(\hat{x}_j(l_{ij}(k)) - \hat{x}_i(k-\mu_{ij}(k))\right),$$

where

$$\mu_{ij}(k) = \begin{cases} \tau_M, & l_{ij}(k) < k-\tau_M, \\ k-l_{ij}(k), & l_{ij}(k) \geq k-\tau_M. \end{cases}$$

Considering the addition of the term $\pm\hat{x}_j(k-\mu_{ij}(k))$ and defining

$$w_{ij}(k) = \hat{x}_j(l_{ij}(k)) - \hat{x}_j(k-\mu_{ij}(k)), \tag{7.19}$$

equation (7.18) can be rewritten as:

$$\hat{x}_i(k+1) = A\hat{x}_i(k) + M_i(\hat{y}_i(k) - y_i(k))$$

$$+ \sum_{j\in\mathcal{N}_i} N_{ij}C_{ij}\left(\hat{x}_j(k-\mu_{ij}(k)) - \hat{x}_i(k-\mu_{ij}(k))\right) + \sum_{j\in\mathcal{N}_i} N_{ij}C_{ij}w_{ij}(k).$$

This way, the observer have a dynamics equivalent to that of the periodic communication case, difference being in the terms that depends on $w_{ij}(k)$, which can be interpreted as an external perturbation due to the discontinuous flow of information between neighbors that is reset to zero at every transmission time. In this case $\mu_{ij}(k)$ plays the role of $\tau_{ij}(k)$ in equation (7.7). It is straightforward to check that $0 \leq \mu_{ij}(k) \leq \tau_M$ for cases 1 and 2.

Moreover, it is easy to see that

$$\|w_{ij}(k)\|_\infty < \delta,$$

in both cases. In the case 2, $w_{ij}(k) = 0$. In the case 1, it holds $w_{ij}(k) = \hat{x}_j(l_{ij}(k)) - \hat{x}_j(k - \tau_M)$, with $l_{ij}(k) < k - \tau_M$. Since no packet has been transmitted between $l_{ij}(k)$ and $k - \tau_M$ (because, otherwise, this packet had been available in node i at current instant k), it implies that $\|\hat{x}_j(l_{ij}(k)) - \hat{x}_j(k - \tau_M)\|_\infty < \delta$ (see the triggering condition defined above). Therefore, $\|w_{ij}(k)\|_\infty < \delta$ holds for both cases.

The dynamics of the augmented observation vector $e(k)$ is similar to that of the time-driven case, but including an additional term related with these disturbances:

$$e(k+1) = \Phi(\mathscr{M})e(k) + \Lambda(\mathscr{N})d(k) + \Gamma(\mathscr{N})w(k), \tag{7.20}$$

where $w^T(k) = [w_1^T(k), \cdots, w_r^T(k), \cdots, w_l^T(k)]$, with $r = g(i,j)$. As before, it is not difficult to see that matrix $\Gamma(\mathscr{N})$ has the following structure:

$$\Gamma(\mathscr{N}) = \left[\begin{array}{cccccc} \Gamma_1(\mathscr{N}) & \cdots & \Gamma_r(\mathscr{N}) & \cdots & \Gamma_l(\mathscr{N}) \end{array} \right],$$

where $\Gamma_r(\mathscr{N})$, $r = g(i,j) \in \{1, ..., l\}$, are vectors of matrices, in which the only block different from zero is $N_{ij}C_{ij}$ in the i row:

$$\Gamma_r(\mathscr{N}) = \left[\begin{array}{c} 0 \\ \vdots \\ N_{ij}C_{ij} \\ \vdots \\ 0 \end{array} \right] \quad row\ i$$

As it has been mentioned, in this section it is considered that the observers are designed according to Theorem 7.2.1, so in the following, notation Φ, Λ, Γ will be used instead of $\Phi(\mathscr{M}), \Lambda(\mathscr{N}), \Gamma(\mathscr{N})$.

7.3.2 Practical stability for delayed asynchronous systems

Next, the main result of this section is introduced. Given the distributed observer synthesized by Theorem 7.2.1, the following result proves that, by implementing the event-based sampling policy described above, the observation error $e(k)$ can be ultimately bounded into an arbitrary small region that depends on the triggering threshold δ. The proof of the theorem makes use of the Lyapunov-Krasovskii functional (7.12) and the fact that it can be written in the following quadratic way

$$V(k) = \zeta^T(k) L_4 \zeta(k), \tag{7.21}$$

where

$$\zeta(k) = \begin{bmatrix} e(k) \\ e(k-1) \\ e(k-2) \\ \vdots \\ e(k-\tau_M) \end{bmatrix},$$

and L_4 is a positive definite matrix that can easily found, for all the terms in the functional are quadratic.

Theorem 7.3.1. *Consider a set of distributed observers designed by Theorem 7.2.1. Then, using an event-based communication with triggering condition* (7.17)*, the estimation error $e(k) \in \mathbb{R}^n$ whose dynamics is given in equation (7.20) is ultimately bounded by*

$$\|e(k)\|_\infty \leq \sqrt{\frac{\lambda_{\max}^{L_4}}{\lambda_{\min}^{P}}} \left[(\|\Phi\|_\infty + \|\Lambda\|_\infty) D_1 + \|\Gamma\|_\infty \delta \right],$$

where matrices P, Φ, Λ and Γ are given in Theorem 7.2.1 and

$$D_1 = \frac{\|L_2\|_\infty + \sqrt{\|L_2\|_\infty^2 + \lambda_{\min}^{Q} \|L_3\|_\infty}}{\lambda_{\min}^{Q}} \delta, L_2 = \Gamma^T P \begin{bmatrix} \Phi & \Lambda & 0 \end{bmatrix} + \Gamma^T Z_2 \begin{bmatrix} (\Phi - I) & \Lambda & 0 \end{bmatrix},$$

$$L_3 = \Gamma^T P \Gamma + \Gamma^T Z_2 \Gamma,$$

being Q any positive definite matrix such that $-Q > L_1$, with L_1 given in equation (7.13)*.*

Proof: Consider the Lyapunov-Krasovskii functional (7.12). Including the disturbances due to the asynchronous flow of information, the forward difference takes the form:

$$\Delta V(k) \leq \xi^T(k) L_1 \xi(k) + 2 w^T(k) L_2 \xi(k) + w^T(k) L_3 w(k).$$

From Theorem 7.2.1 the matrix L_1 is negative definite, so there exists a positive matrix Q such that $L_1 < -Q$. Taking norms

$$\Delta V(k) \leq -\lambda_{\min}^{Q} \|\xi(k)\|_\infty^2 + 2\|L_2\|_\infty \|w(k)\|_\infty \|\xi(k)\|_\infty + \|L_3\|_\infty \|w(k)\|_\infty^2,$$

so taking into account the triggering condition (7.17), it yields

$$\Delta V(k) \leq -\lambda_{\min}^{Q} \|\xi(k)\|_\infty^2 + 2\|L_2\|_\infty \delta \|\xi(k)\|_\infty + \|L_3\|_\infty \delta^2.$$

Therefore, solving the second order equation it can be ensured that $\Delta V(k) \leq 0$ for $\|\xi(k)\|_\infty > D_1$, with $D_1 = \frac{\|L_2\|_\infty + \sqrt{\|L_2\|_\infty^2 + \lambda_{\min}^{Q} \|L_3\|_\infty}}{\lambda_{\min}^{Q}} \delta$.

For a generic vector x and a positive scalar D, let B_D^x denote the region of the space defined by $\{x : \|x\|_\infty \leq D\}$. Given that $V(k)$ is positive and decreasing for $\xi(k) \notin B_{D_1}^{\xi}$, there exists a time instant k^* when $\xi(k^*)$ enters into the region $B_{D_1}^{\xi}$. Since it is considered infinite norms and $e(k)$ is included in the augmented vector $\xi(k)$, it turns out that $e(k^*) \in B_{D_1}^{e}$.

As $\xi(k^*) \in B_{D_1}^{\xi}$ for any realization of $\mu_r(k) \in [0, \tau_M]$, $r \in \mathscr{L}$, it also holds that $\zeta(k^*) \in B_{D_1}^{\zeta}$.

Once $\xi(k^*)$ belongs to this region, the Lyapunov function is not necessarily decreasing and the augmented state may jump outside, that is, $\xi(k^*+1) \notin B_{\xi}$. Using the dynamics of the observation error given in equation (7.20), it is possible to bound the error at instant k^*+1 by

$$\begin{aligned} \|e(k^*+1)\|_\infty &< \|\Phi\|_\infty \|e(k^*)\|_\infty + \|\Lambda\|_\infty \|d(k^*)\|_\infty + \|\Gamma\|_\infty \|w(k^*)\|_\infty \\ &< (\|\Phi\|_\infty + \|\Lambda\|_\infty) D_1 + \|\Gamma\|_\infty \delta. \end{aligned}$$

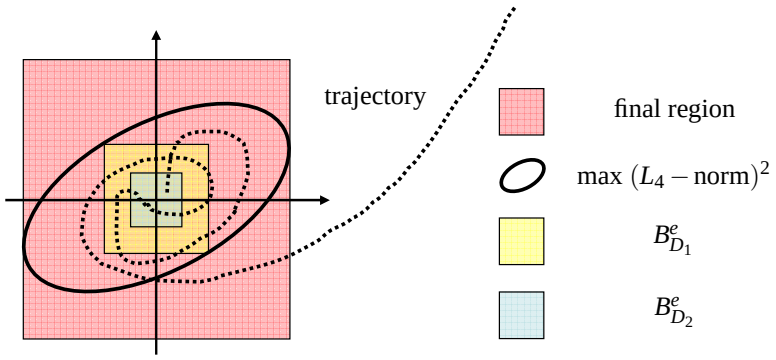

Figure 7.5: Trajectory of the error in a two-dimensional space

Then $e(k^*+1) \in B^e_{D_2}$, where $D_2 = (\|\Phi\|_\infty + \|\Lambda\|_\infty)D_1 + \|\Gamma\|_\infty \delta$. Figure 7.5 illustrates a possible evolution of the observation error and the different regions.

This way, $\xi(k^*+1)$ and $\zeta(k^*+1)$ may leave the regions $B^\xi_{D_1}$ and $B^\zeta_{D_1}$, respectively. Then, the Lyapunov-Krasovskii functional must be decreasing again, implying that

$$\forall k > k^*+1, \quad V(k) < \max\{V(k^*+1)\} = \max\{\zeta^T(k^*+1)L_4\zeta(k^*+1)\}$$

$$< \lambda^{L_4}_{\max} \max\{\|\zeta(k^*+1)\|^2_\infty\}$$

$$< \lambda^{L_4}_{\max} D_2^2.$$

Finally, to get the final bound on $e(k)$ for $k > k^*+1$, note that all the terms of the Lyapunov functional involve positive definite matrices, so

$$e(k)^T P e(k) < V(k), \forall k,$$

$$V(k) < \lambda^{L_4}_{\max} D_2^2, \forall k > k^*+1.$$

And using fairy extended properties, it yields

$$\lambda^P_{\min} \|e(k)\|^2_\infty < e(k)^T P e(k) < \lambda^{L_4}_{\max} D_2^2$$

$$\Rightarrow \|e(k)\|_\infty < \sqrt{\frac{\lambda^{L_4}_{\max}}{\lambda^P_{\min}}} D_2.$$

This ends the proof. □

Note that the final bound of $e(k)$ depends on the threshold δ that triggers the sampling. With $\delta = 0$, the event-based sampling becomes a periodic one and the asymptotic convergence will be accomplished. Furthermore, it is possible to use the theorem to find the suitable δ in order to achieve a prescribed final bound on the estimation error and a trade-off between estimation performance and traffic reduction.

7.4 Numerical example

Consider a plant whose dynamics is given by:

$$x(k+1) = \begin{bmatrix} 0.99 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.005 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.9945 & -0.08757 & 0 & 0 \\ 0 & 0 & 0.1248 & 0.9945 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.9 & 0.09 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} x(k).$$

Observer 1 measures the output $y_1 = x_1$ while observers 2, 3, and 4 receives $y_2 = x_2$, $y_3 = x_4$, and $y_4 = x_6$ respectively. The observers are connected according to an incomplete communication graph, depicted in Figure 7.6. The outputs measured from every node and the received estimates are summarized in Table 7.1.

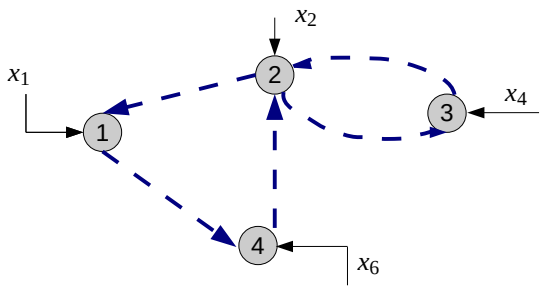

Figure 7.6: Graph representing the network connectivity

	Measurements	Received outputs
Node 1	$y_1 = C_1 x = [1\,0\,0\,0\,0\,0]x$	$C_{12} = [C_2; C_3; C_4]$
Node 2	$y_2 = C_2 x = [0\,1\,0\,0\,0\,0]x$	$C_{23} = C_3$, $C_{24} = [C_1; C_4]$
Node 3	$y_3 = C_3 x = [0\,0\,0\,1\,0\,0]x$	$C_{32} = [C_1; C_2; C_4]$
Node 4	$y_3 = C_3 x = [0\,0\,0\,0\,0\,1]x$	$C_{41} = [C_1; C_4]$

Table 7.1: Outputs and information shared with neighbours

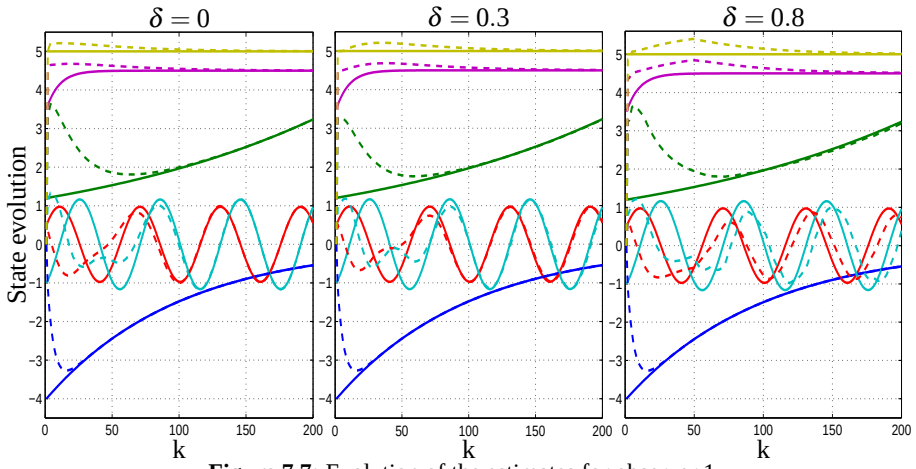

Figure 7.7: Evolution of the estimates for observer 1

Note that local observability is not achieved from any of the observers. The design of the observation matrices and the simulation have been performed for a maximum delay of $\tau_M = 3$ in all links.

Figures 7.7 and 7.8 represent the evolution of the plant states (continuous lines) and the estimated states (dashed lines) for observers 1 and 4 respectively. The initial states for all the observers are set to zero. The plant's initial state is

$$x(0) = \begin{bmatrix} -4 & 1.2 & 0.5 & -1 & 3.5 & 5 \end{bmatrix}^T.$$

It is worth pointing out that all the observers converge faster to the states that can be locally estimated, given that they are not affected by communication effects (delays and asynchronicity).

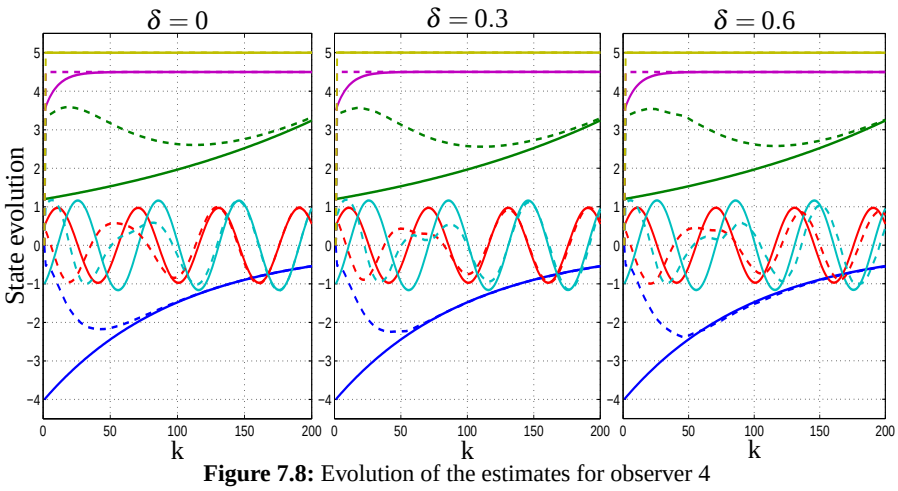

Figure 7.8: Evolution of the estimates for observer 4

For both observers, the simulations on the left correspond to periodic communication policy ($\delta = 0$). As expected, when the threshold to communicate is enlarged, the observers broadcast less information to their neighbors and the estimation performance progressively diminishes. Nonetheless, it is possible to find an adequate trade-off between estimation performance and communication savings, achieving remarkable reduction on the network traffic load while maintaining an estimation performance close to the periodic communication case.

Finally, Figure 7.9 shows the percentage of packets transmitted with respect to periodic communication policy for observer 1 and different communication thresholds.

7.5 Conclusions

In this chapter, the problem of distributed estimation considering network-induced delays and dropouts is solved. Two schemes are analyzed, namely, periodic time-driven and event-based approaches, the latter being specially beneficial in terms of economy of use of network resources. For both scenarios, the observers employ a local Luenberger-like structure and consensus matrices to weight the information received from neighbors. The informa-

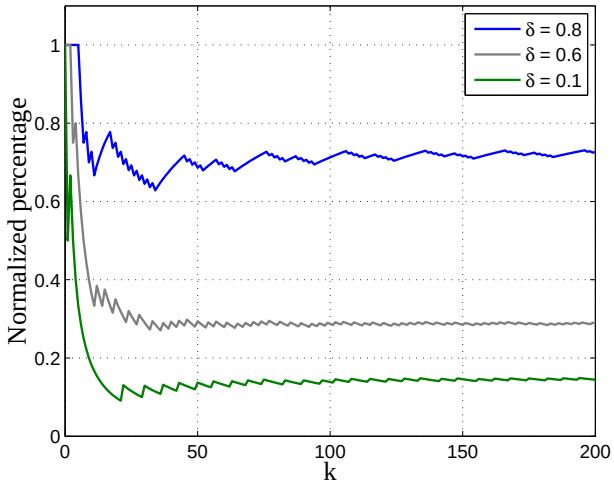

Figure 7.9: Normalized percentage of transmitted packets for different thresholds

tion shared between neighbors does not need to be necessarily the complete estimated state, but can be selected to reduce communication requirements. Stability proofs are provided and performance of the design methods is showed by a simulation example.

Future research may include the consideration of different delay bounds for each link as well as the robustification of the method to deal with parametric uncertainties in the system model and exogenous perturbations or noise. Other important line of research consists in reducing the computational requirements of the proposed algorithms, by distributing the design algorithms.

7.6 Related publications

- Pablo Millán, Luis Orihuela, Isabel Jurado Carlos Vivas and Francisco R. Rubio. *Distributed Estimation in Networked Systems Under Periodic and Event-Based Communication Policies*. International Journal of System Science.

Chapter 8

Conclusions and future work

In this chapter a summary of the main contributions included in this thesis is presented. It also provides some possible directions to follow in order to continue with the research lines that have been started in this thesis.

8.1 Conclusions

This thesis has been focused in the field of Networked Control Systems (NCSs), in particular in those that the network introduces packet losses. It have been also considered other problems such as delays and uncertainties in the model of the plant. Different techniques have been studied in order to deal with these problems.

The main contributions can be included in three different areas. The first three chapters (Chapters 2, 3 and 4), have been focused on H_∞ techniques, dealing with plant with uncertainties in their models. Chapter 5 has presented contributions in the area of Model Predictive Control. Chapter 6 has also used MPC strategies in order to solve a distributed control problem. That chapter has presented contributions also in the last area under consideration in this thesis, the distributed controlled systems. This area is exclusively treated in Chapter 7, which has been centered in distributed estimation.

- Firstly, Chapter 2 has focused on control loops for SISO LTI plants, where the feed-

back path comprises a communication channel affected by Bernoulli data losses. These systems have been studied as equivalent ones wherein the unreliable channel has been replaced by an additive i.i.d. noise channel, plus a gain. The objective of the chapter has been the synthesis of controllers that compensate model uncertainties and failed transmissions. To perform this task, an H_∞ control problem has been proposed.

- In Chapter 3 NCSs subject to data dropouts constraints have been considered. The unreliable channel has been replaced by an additive i.i.d. noise channel, plus a gain.

  A controller that avoid the model uncertainties has been synthesized. Also, a lower bound of the success probability in the transmission has been found. To perform this task, a mixed H_2/H_∞ control problem has been proposed. Moreover, the minimal successful transmission probability that guarantees MSS and robustness properties for the closed-loop system has been obtained.

  Also, some numerical results that illustrate the closed-loop system performance have been presented. These simulation results showed that robust performance is achieved if the successful probability transmission is higher than the minimum computed, while the differents systems performances get worse, until the robust stability is lost, as the successful probability transmission decreases.

  Furthermore, an application of this technique to the problem of the glucose control for diabetic patients subject to sensor errors constraints has been presented. Different patient characteristics have been considered, representing the uncertainties to take into account for the synthesis of the controller.

- Chapter 4 has focused also on control loops for SISO LTI plants, where the feedback path comprises a communication channel that produces data losses.

  Firstly, it is considered that only one consecutive packet can be lost.

One of the objective of this chapter has been the synthesis of a controller and a filter that avoid the model uncertainties and compensate failed transmissions. When a data dropout occurs, the control system uses an estimated output given by the filter to do the feedback. To perform this task, a H_∞ control problem has been proposed in order to calculate the controller. To obtain a robust controller, some functions have been chosen to weight some sensitivity functions. The filter is calculated with a technique based on the location of the unstable poles of the model of the plant.

An example has been exposed to obtain some numerical results that illustrates the closed-loop system performance. These simulation results corroborated that robust performance is achieved.

The other objective of the chapter has been the synthesis of a controller and a filter that avoid the model uncertainties and compensate failed transmissions considering that the maximum number of consecutive dropouts is known.

The NCS has been modelled as a Markov Jump Linear System, with modes depending on the network situation.

When a data dropout occurs, the system uses an estimated output given by the filter as feedback. To perform this task, a H_∞ control problem has been proposed in order to calculated the controller. The filter is calculated with an H_∞ technique together with the controller.

Finally, some simulations have been showed to illustrate the closed-loop system performance. These results corroborated that robust performance is achieved.

- The first part of Chapter 5 has presented a methodology to compensate data dropouts and delays in networked control systems, using model predictive control. The methodology takes advantage of the intrinsic computation of future control signals in predictive control, to cope with eventual data dropouts. A key aspect is the inclusion of

a buffering strategy together with a model-based plant estimator that, under certain conditions, ensure stability of the controlled system.

Simulation results show that remarkable data dropout rates up to 40% can be achieved without significant performance degradation, as well as traffic load alleviation up to 85% with respect to conventional buffered predictive control systems.

Secondly, Chapter 5 has shown how statistical information on packet delays and dropouts can be used in the design of a networked control system.

Also, an experimental plant has been chosen to obtain some numerical results.

- Chapter 6 has presented a cooperative MPC formulation for NCSs subject to data dropouts. An algorithm that decides which nodes are in charge of the calculation of the the control input and which ones just relay the received information is provided. This decision depends on transmission outcomes. Once the controller node has been chosen, it interacts with its neighbors over unreliable links solving a cooperative MPC.

  Also, Chapter 6 has presented a flexible architecture for the implementation of an estimated state feedback control law over a wireless sensor-actuator network with analog unreliable channels without acknowledgments. With the algorithm provided, the role played by individual nodes depends on transmission outcomes. In particular, the state estimator location depends upon the availability of past plant input values and transmission outcomes, while the controller is always located at actuator node. By deriving a Markovian jump-linear system model, it is established a closed form expression for the stationary covariance of the system state in the presence of correlated dropout processes.

- Finally, in this Chapter 7 the problem of distributed estimation considering network-induced delays and dropouts is solved. Two schemes are analyzed, namely, periodic

time-driven and event-based approaches, the latter being specially beneficial in terms of economy of use of network resources. For both scenarios, the observers employ a local Luenberger-like structure and consensus matrices to weight the information received from neighbors. The information shared between neighbors does not need to be necessarily the complete estimated state, but can be selected to reduce communication requirements. Stability proofs are provided and performance of the design methods is showed by a simulation example.

8.2 Future work

Future works that could be considered for further study are the following:

- Different structures for the parameter $Q(z)$ in Chapter 2, non-linear systems and also to include delays in the communication channel.
- Studying closed loop stability and performance issues for the contributions on Chapter 5 and Chapter 6.
- Study a practical application of the method proposed in Chapter 6 for systems controlled over unreliable networks with time-varying reliability, for example, if there are moving obstacles blocking the nodes. Also, future works may include extending the ideas presented to the control of multiple-loops, to general network topologies, and to controller design.
- In Chapter 7, future researches may include the consideration of different delay bounds for each link as well as the robustification of the method to deal with parametric uncertainties in the system model and exogenous perturbations or noise. Other important line of research consists in reducing the computational requirements of the proposed algorithms, by distributing the design algorithms.

Capítulo A

Resumen en Castellano

Los sistemas de control a través de redes se han convertido en un área importante dentro de la comunidad de control, lo cual es debido a su bajo coste y a la flexibilidad de sus aplicaciones. Los sistemas de control a través de redes (NCSs) se componen de sensores, actuadores y controladores; las operaciones entre ellos se coordinan a través de una red de comunicación. Típicamente, estos sistemas están espacialmente distribuidos, y pueden funcionar de manera asíncrona, pero sus operaciones han de estar coordinadas para conseguir los objetivos deseados.

En este resumen se presenta una perspectiva general de los NCSs, y en particular, los casos específicos en los que se ha basado esta tesis, abordando los temas principales relacionados con NCS, con todos los problemas y ventajas asociados, se describen en este resumen. Por último, se presenta un índice de la tesis con sus contribuciones más relevantes.

A.1 Introducción a los Sistemas de Control a través de Red

Los Sistemas de Control a través de Red (NCSs) son sistemas espacialmente distribuidos donde la comunicación entre plantas, sensores, actuadores y controladores se realiza a través de una red de comunicación. Este tipo de sistemas y sus características son descritos

ampliamente en [J. P. Hespanha, P. Naghshtabrizi and Y. Xu, 2007], [W. Zhang, M. S. Branicky and S. M. Phillips, 2001], [R. A. Gupta, 2010] and [J. Chen, K. H. Johansson, S. Olariu, I. Ch. Paschalidis and I. Stojmenovic, 2011].

La complejidad en el diseño y la realización, el coste del cableado, la instalación y el mantenimiento pueden ser reducidos drásticamente incluyendo una red de comunicación. Sin embargo, las redes de comunicación en los sistemas también traen algunos incovenientes como los retrasos y la pérdida de datos, los errores de codificación, etc. Estos incovenientes pueden ser la causa de la de la degradación del comportamiento del sistema e incluso causar su desestabilización.

Hoy en día, hay un gran número de situaciones prácticas en las que el uso de redes de comunicación para el control son necesarias para aplicaciones o procesos de control en ingeniería. Algunos ejemplos son:

- Situaciones en las que el espacio y el peso están limitados.
- Situaciones en las que las distancias a considerar son grandes.
- Aplicaciones de control donde el cableado no es posible.

El uso de redes de comunicación digitales proporciona también algunas ventajas:

1. La complejidad en el cableado en conexiones punto a punto se reduce mucho, así como el coste. Además, los costes de instalación pueden reducirse también drásticamente.
2. La reducción en la complejidad del cableado hace mucho más fácil el diagnóstico y el mantenimiento del sistema, dando lugar a un ahorro en el coste debido a que la instalación y el funcionamiento tienen una eficiencia mayor.
3. Los NCSs son flexibles y reconfigurables.
4. Fiabilidad, redundancia y robustez ante los fallos.

5. Los NCSs proporcionan modularidad, control descentralizado y diagnósticos integrados.

Todas estas ventajas sugieren que los NCSs jugarán un papel principal en un futuro cercano, siendo un área de investigación muy prometedora.

A.2 Objetivos de la tesis

La idea general de esta tesis es proponer algunas soluciones novedosas a diferentes problemas relacionados con NCSs. Todos los problemas considerados son típicos dentro del marco del control a través de redes, considerándose principalmente el de las pérdidas de paquetes en la transmisión de datos.

Dentro del contexto de sistemas con pérdida de paquetes, se han estudiado diferentes problemas. Para obtener soluciones diferentes para este tipo de sistemas, se han considerado los siguientes objetivos:

- Diseño de controladores.

  Controladores H_∞, que consigan la robustificación de sistemas con incertidumbres.

  Controladores *MPC*, combinados con estrategias de buffer.

- Diseño de filtros.

  Filtros H_∞ para sistemas con incertidumbres, usando técnicas frecuenciales y cadenas de Markov.

- Diseño de algoritmos.

  Localización dinámica de un control distribuido en una red formada por una estructura matricial de nodos.

Localización dinámica del estimador de la salida del sistema, en una red formada por una estructura lineal de nodos.

- Estimación distribuida cooperativa.

  Basada en observadores locales de Luenberger.

Uno de los objetivos de esta tesis ha sido el análisis de la estabilidad y comportamiento de sistemas bajo control. En algunos casos, el diseño se ha realizado imponiendo restricciones en cuanto a la estabilidad.

La robustificación de sistemas, en particular la de aquellos con incertidumbres, ha sido también tenida en cuenta. Las técnicas de control H_∞ se han usado en los casos de análisis y diseño de sistemas de control.

Otro objetivo importante de esta tesis ha sido el diseño de algoritmos para una red dinámica, la cual está compuesta por cierta estructura de nodos. El algoritmo es capaz de decidir qué nodo será el controlador o el estimador de la salida del sistema en la red. La estabilidad y el comportamiento del sistema de control ha sido analizado.

También se ha abordado el diseño de estimación y esquemas distribuidos. Se han considerado redes que introducen retrasos temporales, junto con pérdidas aleatorias. La reducción en el consumo de energía ha sido un objetivo importante en esta parte de la tesis. En este caso, se ha examinado una política de comunicación entre agentes basada en eventos, la cual da lugar a un compromiso entre el comportamiento del sistema y los ahorros en la comunicación.

A.3 Contribuciones de la tesis

En esta sección, se presenta un breve resumen con las contribuciones de cada capítulo.

En todos los capítulos, se considera un NCS en el que un canal de comunicación introduce pérdida de datos.

En el Capítulo 2, se considera que el modelo de la planta tiene incertidumbres estructurales. Así pues, el principal objetivo de este capítulo es encontrar un controlador robusto para la planta con incertidumbres y con pérdida de datos en la transmisión; a la vez que se minimiza la varianza de la señal de error. La *Estabilidad Media Cuadrática (MSS)* y propiedades de robustez tienen que estar garantizadas también. Se propone una técnica de control H_∞ de manera se puedan tolerar las incertidumbres estructurales y las pérdidas al mismo tiempo, a la vez que se optimiza el comportamiento del sistema. Para tratar con las incertidumbres de la planta, se ha calculado un controlador central LTI. Este controlador central se combina con una función de transferencia, $Q(z)$, que es la que está a cargo de la minimización de la varianza del error. Con el fin de encontrar $Q(z)$, se propone un algoritmo que proporciona una solución que satisface las restricciones. La unión de estas dos funciones de trasferencia proporciona el controlador propuesto.

En el Capítulo 3 se ha usado la misma estructura para el NCS que en el Capítulo 2. La diferencia aquí ha sido el uso de la técnica de control H_2/H_∞ (mientras que el Capítulo 2 sólo se ha usado la técnica H_∞). La parte H_2 se ha diseñado de manera que estabilice el NCS, teniendo en cuenta la probabilidad de pérdida de datos, mientras que la parte H_∞ se ha usado para hacer el sistema en lazo cerrado suficientemente robusto frente a las incertidumbres estructurales del sistema.

Además, se ha presentado una aplicación de esta técnica para el control de la glucosa en pacientes diabéticos, teniendo en cuenta que los posibles errores producidos por los sensores. Se han considerado diferentes características de pacientes, lo cual representa las incertidumbres a tener en cuenta en la síntesis del controlador.

En el Capítulo 4, también se consideran incertidumbres estructurales en el modelo de la planta. Uno de los objetivos de este capítulo es encontrar un controlador robusto para la planta con incertidumbres, lo cual se ha realizado mediante la técnica de control H_∞. Otro objetivo importante es el diseño de un filtro que calcule una estimación de la salida

de la planta. Esta estimación se usa cuando se produce se pierda un paquete, de manera que sustituya a la realimentación cuando ésta no esté disponible. La *Estabilidad Cuadrática Media (MSS)* y las propiedades de robustez tienen que estar garantizadas también. El diseño del filtro ha sido realizado mediante una técnica basada en la localización de los polos inestables del modelo de la planta. Se puede encontrar más información sobre este tema en [J. E. Normey-Rico and E. F. Camacho, 2009].

En este mismo capítulo, se ha modelado el sistema como un *Markov Jump Linear System (MJLS)* y se ha derivado un LMI para encontrar un filtro y un controlador robustos mediante técnicas H_∞ (ver [S. Skogestad, and I. Postlethwaite, 2005]). El filtro diseñado calcula una estimación del estado de la planta. Al igual que antes, esta estimación se usa cuando haya una pérdida y se anule la realimentación.

El Capítulo 5 propone un esquema con control predictivo que se basa en una política de intercambio de información del sensor/actuador vs. el controlador. El problema aquí es el diseño de una estrategia para sistemas lineales con red y con ruido, con pérdida de datos grande, de manera que se conserve un buen comportamiento. Adicionalmente, también ha sido un aspecto de interés el limitar la cantidad de información transmitida en el sistema de control a través de red. En este capítulo, se ha explorado también el efecto de reducir el intercambio de paquetes entre controlador y actuador, mientras que se conserva un umbral para el error de las señales de control del actuador. Este umbral permite limitar la cantidad de información a través de la red, de manera que se transmite sólo cuando sea necesaria información relevante para el control.

El modelo de la red considera pérdida de paquetes tanto en la conexión controlador-actuador como en la de sensor-actuador. Esto motiva que se incluya un buffer para la detección y compensación de paquetes perdidos y un estimador del estado, respectivamente. Para mostrar el comportamiento de la estrategia propuesta de compensación, se han incluido resultados de simulación para el problema del control de nivel de agua en un sistema

de tres tanques.

Se ha propuesto también el envío desde el controlador de una secuencia de señales de control que, tratadas apropiadamente en el buffer y en el actuador, actúan como salvaguarda en el caso de retrasos o pérdidas de paquetes eventuales. Este concepto encaja de forma natural en el modelo de control predictivo (MPC).

También en el Capítulo 5, se supone que se pueden medir con suficiente precisión las propiedades estadísticas de los retrasos y las pérdidas. Esto se puede explotar para el diseño de un MPC estocástico basado en paquetes para mejorar el comportamiento del control.

El Capítulo 6 estudia NCSs en los que la red está compuesta por una cierto número de nodos que forman una estructura matricial. Estos nodos siguen un algoritmo que decide cuál será el que calcule la señal de control. Este nodo resuelve un MPC cooperativo comunicándose con sus vecinos. Este nodo conoce parte del modelo del sistema y comparte su información con un grupo de nodos vecinos, de manera que cooperan para intercambiar su información sobre el sistema. En cada instante de muestreo se elige un grupo diferente de nodos para calcular la señal de control. Este grupo de nodos es elegido dependiendo del particular estado de la red en ese tiempo de muestreo.

El Capítulo 6 extiende también el reciente trabajo [D. E. Quevedo, K. H. Johansson, A. Ahlén and I. Jurado, 2012] para incluir NCSs con enlaces paralelos entre nodos y el uso de MPC cooperativo. La idea está motivada por el hecho de que el estado de las conexiones pueden cambiar en cada instante de muestreo, de manera que un nodo en particular no será siempre la mejor opción para el cálculo de la señal de control.

En el Capítulo 6 también se estudia un NCS con un lazo simple que usa conexiones en serie a través de un canal con pérdidas. Así pues, las transmisiones están afectadas por pérdidas aleatorias de paquetes. En este capítulo, se tratan situaciones en las que los nodos tienen limitados la energía y la potencia de procesado. El único nodo que proprociona realimentación es el actuador, que transmite el valor de la entrada aplicada a la planta a través

de enlaces paralelos no fiables a los nodos intermedios. En vez de afrontar formulaciones de control óptimo (que dependen de los parámetros de la red y pueden ser difíciles de implementar en la práctica), el controlador ha sido prediseñado. Más específicamente, se asume que la política de control consiste en una ganancia prediseñada en la realimentación del estado combinada con un observador del mismo, el cual, en ausencia de los efectos de la red, daría lugar al comportamiento deseado. Dentro de este contexto, se ha presentado una arquitectura flexible para NCSs donde el papel que desempeña cada nodo en particular depende del resultado de las transmisiones. Con el algoritmo propuesto, los resultados de las transmisiones determinan, en cada instante, si la estimación del estado se calcula en el nodo actuador, en el sensor o en algunos de los intermedios.

En el Capítulo 7 se ha discutido el caso de estimación distribuida cooperativa, basada en observadores locales de Luenberger, en combinación con estrategias de consenso. Se considera que la red induce retrasos y pérdidas de paquetes.La eficiencia en el uso de los recursos de la red recibe mucha importancia, tanto en el caso de comunicación periódica entre agentes como en el de comunicación basada en eventos. Se trata de reducir la cantidad de información que se transmite a través de la red recurriendo a dos ideas diferentes: por un lado, sólo nodos vecinos pueden comunicarse entre sí, reduciendo las transmisiones con respecto a esquemas en los que todos comunican con todos. Por otro lado, el diseño de los observadores contempla la posibilidad de compartir parte del estado estimado entre vecinos, en vez de comunicar el vector de estado estimado completo. Esta economía en el uso de los recursos de la red es, por su propia naturaleza, notablemente mejorada con la estrategia de comunicación basada en eventos.

Bibliography

R. Olfati-Saber, J. A. Fax and R. M. Murray [2007], 'Consensus and cooperation in networked multi-agent systems', *Proceedings of the IEEE* **95**(1), 215–233.

A. N. Venkat, J. B. Rawlings and S. J. Wright [2007], Distributed model predictive control of large-scale systems, *in* 'In Assessment and Future Directions of Nonlinear Model Predictive Control', Springer-Verlag Berlin Heidelberg, pp. 591–605.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. and Cayirci, E. [2002], 'Wireless sensor networks: a survey', *Computer networks* **38**(4), 393–422.

Alriksson, P. and Rantzer, A. [2006], Distributed Kalman filtering using weighted averaging, *in* '17th International Symposium on Mathematical Theory of Networks and Systems', Kyoto, Japan.

Antsaklis, P. and Baillieul, J. [2004], 'Guest editorial special issue on networked control systems', *IEEE Trans. Automat. Contr.* **49**(9), 1421–1423.

Antsaklis, P. and Baillieul, J. [2007], 'Special issue on technology of networked control systems', *Proc. IEEE* **95**(1), 5–8.

B. Kovatchev, S. Anderson, L. H. and Clarke, W. [2008], 'Comparison of numerical and clinical accuracy of four glucose monitors', *Diabetes Care* **31**, 1160–1164.

Bertsekas, D. P. [2005], *Dynamic Programming and Optimal Control, Vol.1*, Athena Scientific, Belmont, MA.

Briñón Arranz, L., Seuret, A. and Canudas de Wit, C. [2009], Translation control of a fleet circular formation of AUVs under finite communication range, *in* '48th IEEE Conference on Decision and Control and 28th Chinese Control Conference', Shangai, China, pp. 8345 – 8350.

C. L. Robinson and P. R. Kumar [2008], 'Optimizing controller location in networked control systems with packet drops', *IEEE Journal on Selected Areas in Communications* **26**(4), 661–671.

Cardoso de Castro, N., Canudas-de-Wit, C. and Garin, F. [2012], Energy-aware wireless networked control using radio-mode management, *in* 'Proc. Amer. Contr. Conf.'.

Chiuso, A. and Schenato, L. [2011], 'Information fusion strategies and performance bounds in packet-drop networks', *Automatica* **47**, 1304–1316.

Cortes, J., Martinez, S., Karatas, T. and Bullo, F. [2004], 'Coverage control form mobile sensing networks', *IEEE Transactions on Robotic and Automation* **20**(2), 243–255.

D. E. Quevedo, A. Ahlén and K. H. Johansson [n.d.], 'State estimation over sensor networks with correlated wireless fading channels', *IEEE Transactions on Automatic Control* **Accepted for publication**.

D. E. Quevedo, K. H. Johansson, A. Ahlén and I. Jurado [2012], Dynamic controller allocation for control over erasure channels, *in* 'Proc. 3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems', Santa Barbara, CA.

D. Limon, T. Alamo, E.F. Camacho [2002], Stability analysis of systems with bounded additive uncertainties based on invariant sets: Stability and feasibility of mpc, *in* 'In Proc. Amer. Contr. Conf.', Anchorage, Alaska, USA.

D. Muñoz and P. D. Christofides [2008], 'Lyapunov-based model predictive control of nonlinear systems subject to data losses', *IEEE Transactions on Automatic Control* **53**(9), 2076–2089.

D. Muñoz, C. Panagiotis D. [2007], 'Estimation-Based Networked Predictive Control of Nonlinear Systems', *Dynamics of Continuous Discrete and Impulsive Systems* .

D. Quevedo and D. Nešic [2011], 'Input-to-state stability of packetized predictive control over unreliable networks affected by packet-dropouts', *IEEE Transactions on Automatic Control* **56**(2), 370–375.

D. Quevedo, J. Østergaard and D. Nešić [2011], 'Packetized predictive control of stochastic systems over bit-rate limited channels with packet loss', *IEEE Transactions on Automatic Control* **56**(12), 2854—-2868.

D. Quevedo, K.H. Johansson, A. A. and Jurado, I. [n.d.], 'An adaptive architecture for control using erasure channels', *Submitted to Automatica* .

D. Yue, Q. H. and J. Lam [2005], 'Network-based robust h_∞ control of systems with uncertainty', *Automatica* **41**(6), 999–1007.

Dong, H., Wang, Z. and Gao, H. [2012], 'Distributed filtering for a class of time-varying systems over sensor networks with quantization errors and successive packet dropouts', *IEEE Transactions on Signal Processing* **60**(6), 3164–3173.

Dormido, S., Sánchez, J. and Kofman, E. [2008], 'Muestreo, control y comunicación basado en eventos', *Revista Iberoamericana de Automática e Informática Industrial* **5**(1), 5–26.

D.Q. Mayne, J.B. Rawlings, C.V. Rao [2000], 'Constrained model predictive control: Stability and optimality', *Automatica* .

E. Garone, B. Sinopoli and A. Casavola [2010], 'LQG control over lossy TCP-like networks with probabilistic packet acknowledgements', *International Journal of Systems Control and Communications* **2**(1,2,3), 55–81.

E. I. Silva and S. A. Pulgar [2011], 'Control of lti plants over erasure channels', *Automatica* **47**(8), 1729–1736.

El Ghaoui, L., Oustry, F. and AitRami, M. [1997], 'A cone complementary linearization algorithm for static output-feedback and related problems', *IEEE Transactions on Automatic Control* **42**(8), 1171–1176.

Estrin, D., Govindan, R., Heidemann, J. and Kumar, S. [1999], Next century challenges: scalable coordination in sensor networks, *in* '5th annual ACM/IEEE international conference on Mobile computing and networking', Seattle, WA, USA, pp. 263–270.

F. Xiao and L. Wang [2008], 'Consensus protocols for discrete-time multi-agent systems with time-varying delays', *Automatica* **44**(10), 2577—-2582.

Farina, M., Ferrari-Trecate, G. and Scattolini, R. [2009], Distributed moving horizon estimation for sensor networks, *in* '1st IFAC Workshop on Estimation and Control of Networked Systems', Venice, Italy, pp. 126–131.

FeedBack, I. [2012], *Data Sheet: 33-041 Coupled Tank System for Matlab*.

Franceschetti, M., Javidi, T., Kumar, P. R., Mitter, S. and Teneketzis, D. [2008], 'Guest editorial control and communications', *IEEE J. Select. Areas Commun.* **26**(4), 577–579.

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

Goodwin, G. C., Quevedo, D. E. and Silva, E. I. [2008], 'Architectures and coder design for networked control systems', *Automatica* **44**(1), 248–257.

G.P. Liu, J.X. Mu, D.Rees, S.C. Chai [2006], 'Design and stability analysis of networked control systems with random communication time delay using the modified MPC', *International Journal of Control* .

Gupta, A., Langbort, C. and Başar, T. [2010], Optimal control in the presence of an intelligent jammer with limited actions, *in* 'Proc. IEEE Conf. Decis. Contr.', Atlanta, GA, pp. 1096–1101.

Gupta, V., Dana, A. F., Hespanha, J. P., Murray, R. M. and Hassibi, B. [2009], 'Data transmission over networks for estimation and control', *IEEE Trans. Automat. Contr.* **54**(8), 1807–1819.

H. Ishii [2008], 'h_∞ control with limited communication and message losses', *Systems & Control Letters* **57**(4), 322—-331.

Huang, M. and Dey, S. [2007], 'Stability of Kalman filtering with Markovian packet losses', *Automatica* **43**(4), 598–607.

J. A. Giraldo and N. Quijano [2011], Current results and research trends in networked control systems, *in* 'Robotics Symposium, IEEE IX Latin American and IEEE Colombian Conference on Automatic Control and Industry Applications (LARC)', pp. 1–6.

J. C. Doyle, K. Zhou, K. Glover and B. Bodenheimer [1994], 'Mixed h_2 and h_∞ performance objectives ii: optimal control', *IEEE Transactions on Automatic Control* **39**(8), 1575–1587.

J. Chen, K. H. Johansson, S. Olariu, I. Ch. Paschalidis and I. Stojmenovic [2011], 'Guest editorial special issue on wireless sensor and actuator networks', *IEEE Transactions on Automatic Control* **56**(10), 2244–2246.

J. Daafouza and J. Bernussoub [2001], 'Parameter dependent lyapunov functions for discrete time systems with time varying parametric uncertainties', *Systems & Control Letters* **43**(5), 355—-359.

J. E. Normey-Rico and E. F. Camacho [2009], 'Unified approach for robust dead-time compensator design', *Journal of Process Control* **19**, 38–47.

J. P. Hespanha, P. Naghshtabrizi and Y. Xu [2007], 'A survey of recent results in networked control systems', *Proceedings of the IEEE* **95**(1), 138–162.

Johansson, K. [2000], 'The quadruple-tank process: A multivariable laboratory process with an adjustable zero', *IEEE Transactions on Control Systems Technology* **8**(3), 456–465.

K. J. Åström [2006], *Introduction to Stochastic Control Theory*, Dover Publications, Inc.

K. Zhou, J. C. Doyle, and K. Glover [1996], *Robust and optimal control*, Prentice Hall.

Kovacs, L. and Kulcsár, B. [2007], Lpv modeling of type 1 diabetes mellitus, *in* '8^{th} International Symposium of Hungarian Researchers on Computational Intelligence and Informatics'.

L. Xie [1996], 'Output feedback h_∞ control of systems with parameter uncertainty', *International Journal of Control* **63**(4), 741–750.

Lancaster, P. [1970], 'Explicit solutions of linear matrix equations', *SIAM Review* **12**(4), 544–566.

Lee, J.-W. and Dullerud, G. E. [2007], 'A stability and contractiveness analysis of discrete-time Markovian jump linear systems', *Automatica* **43**(1), 168–173.

Liang, J., Shen, B., Dong, H. and Lam, J. [2011], 'Robust distributed state estimation for sensor networks with multiple stochastic communication delays', *International Journal of Systems Science* **42**(9), 1459–1471.

Liang, J., Wang, Z. and Liu, X. [2012], 'Distributed state estimation for uncertain Markov-type sensor networks with mode-dependent distributed delays', *International Journal of Robust and Nonlinear Control* **22**(3), 331–346.

Lu, B., Oyekan, J., Gu, D., Hu, H. and Nia, H. F. G. [2011], 'Mobile sensor networks for modelling environmental pollutant distribution', *International Journal of Systems Science* **42**(9), 1491–1505.

Lunze, J. and Lehmann, D. [2010], 'A state-feedback approach to event-based control', *Automatica* **46**, 211–215.

Lynch, S. M. and Bequette, B. W. [2001], Estimation-based model predictive control of blood glucose in type i diabetics: A simulation study, *in* 'Proceedings of the IEEE 27^{th} Annual Northeast Bioengineering Conference'.

Lynch, S. M. and Bequette, B. W. [2002], Model predictive control of blood glucose in type i diabetics using subcutaneous glucose measurements, *in* 'Proc. Amer. Contr. Conf.'.

M. G. Ortega and F. R. Rubio [2004], 'Systematic design of weighting matrices for the h_∞ mixed sensitivity problem', *Journal of Process Control* **14**(1), 89–98.

M. G. Ortega, M. Vargas, L. F. Castaño and F. R. Rubio [2006], 'Improved design of the weighting matrices for the s/ks/t mixed sensitivity problem - application to a multivariable thermodynamic system', *IEEE Transactions on Control Systems Technology* **14**(1), 82–90.

Ma, X., Djouadi, S. M. and Charalambous, C. D. [2011], 'Optimal filtering over uncertain wireless communication channels', *IEEE Signal Processing Lett.* **18**(6), 359–362.

Maestre, J. M., Giselsson, P. and Rantzer, A. [2010], Distributed receding horizon Kalman filter, *in* '49th IEEE Conference on Decision and Control', Atlanta, GA, USA, pp. 5068 – 5073.

Millán, P., Orihuela, L., Vivas, C. and Rubio, F. [2012], 'Control óptimo-L_2 basado en red mediante funcionales de Lyapunov-Krasovskii', *Revista Iberoamericana de Automática e Informática Industrial* **9**(2).

Moon, Y. S., Park, P., Kwon, W. H. and Lee, Y. S. [2001], 'Delay-dependent robust stabilization of uncertain state-delayed systems', *International Journal of Control* **74**(14), 1447–1455.

N. Elia [2005], 'Remote stabilization over fading channels', *Systems & Control Letters* **54**(3), 237—249.

O. C. Imer and S. Yüksel and T. Başar [2006], 'Optimal control of LTI systems over unreliable communication links', *Automatica* **42**(9), 1429–1439.

O. L. V. Costa, M. D. Fragoso, and R. P. Marques [2005], *Discrete-time Markovian jump linear systems*, Springer-Verlag.

Olfati-Saber, R. [2005], Distributed Kalman filter with embedded consensus filters, *in* '44th IEEE Conference on Decision and Control and the European Control Conference', Seville, Spain, pp. 8179–8184.

Olfati-Saber, R. [2007], Distributed Kalman filtering for sensor networks, *in* '46th IEEE Conference on Decision and Control', New Orleans, LA, USA, pp. 5492–5498.

Olfati-Saber, R. and Shamma, J. [2005], Consensus filters for sensor networks and distributed sensor fusion, *in* '44th IEEE Conference on Decision and Control and the European Control Conference', Seville, Spain, pp. 6698 – 6703.

Orihuela, L., Millán, P., Vivas, C. and Rubio, F. R. [2011], 'Robust stability of nonlinear time-delay systems with interval time-varying delay', *International Journal of Robust and Nonlinear Control* **21**(7), 709–724.

P. Dua, F. J. D. I. and Pistikopoulos, E. N. [2005], Multi-objective parametric control of blood glucose concentration for type 1 diabetes, *in* 'Proceedings of the 44^{th} IEEE Conference on Decision and Control and the European Control Conference'.

P. Dua, F. J. D. I. and Pistikopoulos, E. N. [2006], 'Model-based blood glucose control for type 1 diabetes via parametric programming', *IEEE Transtactions on Biomedical Engineering* **53**(8), 1478–1491.

P. Millan, I. Jurado, C. Vivas and F. R. Rubio [n.d.], Networked predictive control of systems with large data dropouts, *in* 'Proc. 47th IEEE Conference on Decision and Control (CDC)'.

P. Millan, L. Orihuela, C. Vivas and F. R. Rubio [n.d.], An optimal control l2-gain disturbance rejection design for networked control systems, *in* 'American Control Conference (ACC)'.

P. Seiler and R. Sengupta [2005], 'An h_∞ approach to networked control', *IEEE Transactions on Automatic Control* **50**(3), 356—364.

Pajic, M., Sundaram, S., Pappas, G. J. and Mangharam, R. [2011], 'The wireless control network: A new approach for control over networks', *IEEE Trans. Automat. Contr.* **56**(10), 2305–2318.

Pantazis, N. A. and Vergados, D. D. [2007], 'A survey on power control issues in wireless sensor networks', *IEEE Commun. Surv. Tutorials* **9**(4), 86–107.

Park, P. G., Fischione, C. and Johansson, K. H. [2008], Experimental evaluation of power control algorithms for wireless sensor networks, *in* 'Proc. IFAC World Congr.', Seoul, Korea.

Q. Ling and M. Lemmon [2004], 'Power spectral analysis of networked control systems with data dropouts', *IEEE Transactions on Automatic Control* **49**(6), 955–960.

Quevedo, D. E., Ahlén, A. and Johansson, K. H. [2011], Stability of state estimation over sensor networks with Markovian fading channels, *in* 'Proc. IFAC World Congr.'.

Quevedo, D. E., Ahlén, A., Leong, A. S. and Dey, S. [2012], 'On Kalman filtering over fading wireless channels with controlled transmission powers', *Automatica* **48**(7), 1306–1316.

Quevedo, D. E., Ahlén, A. and Østergaard, J. [2010], 'Energy efficient state estimation with wireless sensors through the use of predictive power control and coding', *IEEE Trans. Signal Processing* **58**(9), 4811–4823.

R. A. Gupta [2010], 'Networked control systems: overview and research trends', *IEEE Transactions on Industrial Electronics* **57**(7), 2527–2535.

R. N. Bergman, D. T. F. and Ader, M. [1985], 'Assessment of insulin sensitivity in vivo', *Endocrine Reviews* **6**(1), 45–86.

R. N. Bergman, L. S. P. and Cobelli, C. [1981], 'Physiological evaluation of factors controlling glucose tolerance in man', *The Journal of Clinical Investigation* **68**, 1456–1467.

R. S. Parker, F. J. Doyle III, J. H. W. and Peppas, N. A. [2000], 'Robust h_∞ glucose control in diabetes using a physiological model', *AIChE Journal* **46**(12), 2537–2549.

S. Kamath, V. I. George, S. V. [2006], 'A review of control algorithms for non invasive monitoring', *ACSE Journal* **6**(4).

S. Skogestad, and I. Postlethwaite [2005], *Multivariable feedback control*, Wiley.

Shen, B., Wang, Z. and Hung, Y. S. [2010], 'Distributed H_∞-consensus filtering in sensor networks with multiple missing measurements: The finite-horizon case', *Automatica* **46**(10), 1682–1688.

Shen, B., Wang, Z., Hung, Y. S. and Chesi, G. [2011], 'Distributed h_∞ filtering for polinomial nonlinear stochastic systems in sensor networks', *IEEE Transactions on Industrial Electronics* **58**(5), 1971–1979.

Shen, B., Wang, Z. and Xiaohui, L. [2011], 'A stochastic sampled-data approach to distributed h_∞ filtering in sensor networks', *IEEE Transactions on Circuits and Systems I: Regular Papers* **58**(9), 2237–2246.

Sinopoli, B., Schenato, L., Franceschetti, M., Poolla, K., Jordan, M. I. and Sastry, S. S. [2004], 'Kalman filtering with intermittent observations', *IEEE Trans. Automat. Contr.* **49**(9), 1453–1464.

Smith, R. [2011], A decoupled feedback structure for covertly appropriating networked control systems, *in* 'Proc. IFAC World Congr.', Milan, Italy, pp. 90–95.

Tabuada, P. [2007], 'Event-triggered real-time scheduling of stabilizing', *IEEE Transactions on Automatic Control* **52**(9), 1680–1685.

U. Münz, A. Papachristodoulou and Frank Allgöwer [2008], Delay-dependent rendezvous and flocking of large scale multi-agent systems with communication delays, *in* 'Proceedings of the 47th IEEE Conference on Decision and Control', pp. 2038–2043.

W. Zhang, M. S. Branicky and S. M. Phillips [2001], 'Stability of networked control systems', *IEEE Control Systems Magazine* **21**(1), 84–99.

Wenshan, H., Liu, G. and Rees, D. [2007], 'Networked predictive control system with data compression', *IEEE International Conference on Networking, Sensing and Control, ICNSC'07* pp. 52 – 57.

Xiao, L., Boyd, S. and Lall, S. [2005], A scheme for asynchronous distributed sensor fusion based on average consensus, *in* 'Fourth International Symposium on Information Processing in Sensor Networks', pp. 4209–4214.

Yang, C., Zhu, S.-A., Kong, W.-Z. and Lu, L.-M. [2006], 'Application of generalized predictive control in networked control system', *Journal of Zhejiang University: Science* **7**(2), 225 – 233.

Zilic, Z. and Radecka, K. [2011], Fault tolerant glucose sensor readout and recalibration, *in* 'Proceedings of Wireless Health'.