Strategies for Wireless Networked Control Systems

Mercedes Chacón Vásquez

A thesis presented in fulfilment of the requirements for the degree of Doctor of Philosophy

Industrial Control Centre
Department of Electronic and Electrical Engineering
University of Strathclyde, Glasgow
September, 2017

Declaration of Authorship

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

Signed: _____

Date: _____

Abstract

Networked Control Systems (NCS) and Wireless Networked Control Systems (WNCS) are control systems where controllers, sensors and final elements of control are connected to a mutual communication network. The inclusion of the network introduces delays and dropouts, which greatly influence the stability and robustness of the controller. While there is wealth in theoretical contributions to NCS, it is still imperative to study more applications and investigate the effects of networks in a real-time operation. There are also open problems that require further study of the impact of disturbances, constraints and strong interactions in complex NCS.

This thesis is concerned with the design of control strategies for WNCS mainly focused on Model-Based Predictive Control (MBPC), Proportional Integral Derivative (PID) and decentralised schemes with the aim of creating control laws suitable for compensating time-varying delays and dropouts. These strategies rely on optimisation problems which incorporate robustness and performance restrictions to compute the optimum controller. The performance and robustness of the controllers are evaluated through extensive experiments in a network simulator. A new adaptive Internal Model Control (IMC) controller has been developed to adapt to the network requirements and compute the IMC model parameters online. A new robust PID for NCS under random delays has been created by solving a new constrained optimisation problem that included constraints of maximum sensitivity to guarantee robustness. A novel optimal immune PID is developed to improve the performance of NCS under time-varying delays and dropouts. Simulation results show that the controller offers greater flexibility and improves the performance and robustness with respect to the other methods studied. Four more controllers have been tested and extensive tests have indicated stability for a limited percentage of process model variations and dropouts.

Predictive PID controllers, with similar properties to MBPC, are developed to compensate dropouts in WNCS. A quadratic programming problem optimises a new MBPC cost function to find the optimal PID gains. The approach successfully maximises the performance by changing the controller gains at every sampling time and allowing maximum variations of system parameters and dropouts. Also, a new constrained predictive PID controller is presented to deal with input saturation. Simulation results show the superiority of the design in comparison with the control schemes studied earlier.

Furthermore, a decentralised wireless networked model predictive control design for complex industrial systems has been developed. Also, the method has been applied to wind farm control. The proposed decentralised control offers an effective and innovative solution to improve the performance of large industrial applications.

Acknowledgements

This work has been supported by the University of Costa Rica, the Ministry of Science, Technology and Telecommunications of Costa Rica (MICCIT) and National Council for Scientific and Technological Research of Costa Rica (CONICIT).

I would like to thank the guidance of my supervisor Dr Reza Katebi and his constant support. I am very grateful for this opportunity and for enlightening me in the correct direction for my research work at each stage.

Contents

List of Figures

List of Tables

Abbreviations

BACnet	**B**uilding **A**utomation and **C**ontrol **net**work
CAN	**C**ontrol **A**rea **N**etwork
CSMA/CA	**C**arrier **S**ense **M**ultiple **A**ccess **C**ollision **A**voidance
DMPC	**D**ecentralised **M**odel **P**redictive **C**ontrol
DNCS	**D**ecentralised **N**etworked **C**ontrol **S**ystems
DNMPC	**D**ecentralised **N**etworked **M**odel **P**redictive **C**ontrol
DWNCS	**D**ecentralised **W**ireless **N**etworked **C**ontrol **S**ystems
GM	**G**ain **M**argin
GPC	**G**eneralised **P**redictive **C**ontrol
IAE	**I**ntegral **A**bsolute **E**rror
IMC	**I**nternal **M**odel **C**ontrol
ITAE	**I**ntegral **T**ime **A**bsolute **E**rror
KF	**K**alman **F**ilter
LFC	**L**oad **F**requency **C**ontrol
LMI	**L**inear **M**atrix **I**nequality
LQG	**L**inear–**Q**uadratic–**G**aussian control
LTI	**L**inear **T**ime **I**nvariant
MAC	**M**edia **A**ccess **C**ontrol
MBPC	**M**odel **B**ased **P**redictive **C**ontrol
MIMO	**M**ultiple-**I**nput **M**ultiple-**O**utput
NCS	**N**etworked **C**ontrol **S**ystems
NPC	**N**etworked **P**redictive **C**ontrol

NWFC	**N**etwork **W**ind **F**arm **C**ontroller
PAC	**P**ower **A**djusting **C**ontroller
PID	**P**roportional **I**ntegral **D**erivative
PM	**P**hase **M**argin
PPID	**P**redictive **P**roportional **I**ntegral **D**erivative
PROFIBUS	**P**rocess **F**ield **Bus**
QoS	**Q**uality **o**f **S**ervice
RHPZ	**R**ight **H**alf **P**lane **Z**eros
SISO	**S**imple-**I**nput **S**imple-**O**utput
SQP	**S**equencial **Q**uadratic **P**rogramming
TWFC	**T**urbine **W**ind **F**arm **C**ontroller
WLAN	**W**ireless **L**ocal **A**rea **N**etwork
WNCS	**W**ireless **N**etworked **C**ontrol **S**ystems
ZOH	**Z**ero **O**rder **H**old

Symbols

d	Process dead time	s
d_{pca}	Dropouts from controller to actuator	
d_{psc}	Dropouts from sensor to controller	
$e(k)$	Error signal	
J	Cost function	
J_d	Regulatory performance index	
J_r	Servo performance index	
K_d	Derivative gain	
K_i	Integral gain	
K_p	Proportional gain	
N	Prediction horizon	
N_u	Control horizon	
P_{loss}	Percentage of dropouts	
$r(k)$	Reference	
S_i	Subsystem i	
T_e	Generator torque	Nm
T_s	Sampling time	s
$u(k)$	Control input	
$u_d(k)$	Control input with delays	
$y(k)$	Process output	
$y_d(k)$	Process output with delays	

Δ	Difference operator	
γ	Maximum number consecutive dropouts	
ω_g	Generator angular speed	rad/s
λ	Control weight	
τ_c	Controller computation time	s
τ_{ca}	Network delay from controller to actuator	s
τ_{sc}	Network delay from sensor to controller	s
$\xi(k)$	Zero mean white noise	

To my family.

Chapter 1

Introduction

1.1 Networked Control Systems

Networked Control Systems (NCS) and Wireless Networked Control Systems (WNCS) are control systems where controllers, sensors and final elements of control are connected to a common communication network. NCS applications are increasing as a result of stronger industrial and academic interests in the potential benefits that these systems can offer: for instance, the use of a shared communication network reduces costs. Moreover, NCS are scalable and flexible as they allow adding more devices as needed to perform the system functions. Likewise, NCS make equipment maintenance easy. Most significantly, through the network, multiple control loops can be managed; consequently, multiple goals can be achieved simultaneously. These are some of the reasons why NCS are gaining an important role in automation systems, large-scale systems and remote control applications, for example, power systems, robotics, transportation networks, space exploration, manufacturing processes and many others.

However, the main issue that arises with the inclusion of network communication in the control system is the limited capacity of the shared channel that causes degradation of NCS control performance and can lead to instability. In particular, the network may introduce large communication delays and loss of information, which greatly influences the stability and robustness of the control system. An introduction to the network constraints is covered by Bemporad and Barcelli (2010) and Heemels, Teel, Wouw and Nesic (2010) that address the nature of the problems and the effects.

WNCS offer many attractive advantages, but also their limited capacity leads to constraints, which become more significant in the stability of the closed-loop system. For instance, the limited energy in the wireless nodes leads to non-periodic measurements (Tiberi, Araujo and Johansson, 2012).

In recent years, the control engineering community has developed a full range of control schemes to cope with the network constraints. For example, Proportional Integral Derivative (PID) control, optimal control, adaptive control, robust control, fuzzy control, H_∞ control, model predictive control, event-based and event-triggered control, and so forth. Each of these methodologies shows a wide range of application because they differ in the network modelling techniques, the type of constraints that are addressed and the network that is used. Recent analytical reviews covering these aspects and providing a framework for comparison are given by Zhang et al. (2017), Guo et al. (2014), Wang et al. (2011) and Zhang, Gao and Kaynak (2013).

1.1.1 Predictive PID control

Compensating for time-varying delays and dropouts in NCS is a complex problem requiring controllers with high levels of performance and robustness to ensure the reliability of the control system. The predictive control scheme is considerably effective since it can actively compensate for consecutive packet dropouts (Sun et al., 2014a). Moreover, the method has proven to be robust to perturbations and leads to efficient controllers used in many industrial applications (Camacho and Bordons, 2007). The literature shows that PID is the most successful form of control and has received the most attention in the history of process control. Recent contributions of PID control in NCS have been discussed by Dasgupta et al. (2015), Tran et al. (2013), Okano, Ohtani and Nagashima (2008), Zhang, Shi and Mehr (2011) and Ungan (2010). Motivated by the optimality of the predictive control solution and the simplicity and flexible characteristics of the PID control, these controllers with predictive capabilities have been proposed in the literature. For instance, Miklovičová and Mrosko (2012) demonstrated the effectiveness of a control strategy to compensate dropouts based on Generalised Predictive Control (GPC) with a PID control structure.

1.1.2 Decentralised networked model predictive control

A Decentralised Networked Model Predictive Control (DNMPC) is a conventional DMPC that uses a communication network. DMPC began to gain industrial and academic importance in recent years. For example, the industrial inclusion of MPC to Distributed Control Systems (DCS) reported by Qin and Badgwell (2003) is an important advance in this field. In DNMPC the inclusion of the network implies consideration of control over non-ideal channels. Therefore, performance degradation and stability problems are added to the problems of the decentralised scheme. DNMPC can implement cooperative strategies where the controllers, also called agents, share their control actions among the other agents to improve the performance (Vaccarini, Longhi and Katebi, 2009). In some approaches, information exchange may include predictions of the control signal, states and outputs.

However, according to the nature of the systems, the DNMPC deals with delays, dropouts, disturbances, constrained systems, strong interactions, among other issues. A recent survey by Ge, Yang and Han (2017) defined the many challenges in Distributed Networked Control Systems (DNCS). For example, to guarantee the stability of large-scale NCS is still an area that requires further attention. The potential of DNMPC to deal with these issues had been highlighted (see Tuan et al. (2015) and references therein). A special interest is in the study of the reduction of information exchange in WNCS where power is a significant constraint in the design.

1.2 Motivation of the research

Despite the extensive growth and benefits of the NCS, there are enormous issues that attempt to degrade the reliability of these systems. Two important concerns arise with the use of the network in the control system including delays and loss of information due to the limited capacity of the channel. These problems greatly influence the system performance and they may lead to unstable conditions.

Many of the control approaches that have been developed for NCS require high levels of computation that result in very complex algorithms that make the implementation a hard or unsuitable task for real control systems. In some cases, the achievements of these methods are still very conservative and unable to cope with high and consecutive

occurrence of dropouts and long and random time delays. Moreover, as the complexity of the process increases the performance degrades. The research for simple and flexible algorithms that can effectively compensate NCS under long network delays and a high percentage of loss of information or dropouts is still an open problem.

Among the control approaches, the PID controller is the one that poses the widest applicability in industry. Moreover, the feasibility for systems under time-varying delays has been well demonstrated. However, only a scarce quantity of PID methods for NCS have been proposed with simple structures. Even less is the number of approaches that offer tuning rules for NCS and the stability is guaranteed for a limit value of time delay and a restricted number of process models.

Good results have been found where the benefits of PID controllers are combined with other control structures such as a Smith predictor. For instance, approximations of the varying-time and random behaviour of the delays have been made to increase the accuracy of the model. These solutions attempt to study the stability conditions for the NCS and how to increase the maximum acceptable values of dropouts and time delays.

MBPC is another method greatly accepted in industrial applications and well known for its good performance in many complex control systems with model uncertainty. This methodology can compensate time-varying delays and dropouts within a limit range of uncertainty. However, the higher requirements of performance and robustness demand the development of accurate models for the random and varying-time nature of the delays and further examinations of the effects on the stability of the system.

The formulation of predictive PID controllers promises a viable solution where the flexibility and simplicity of the PID structure and the predictor capabilities allow to examine the stability, performance and robustness of the closed-loop under severe network conditions. Moreover, it represents a good opportunity to increase the reliability in the NCS and a challenge for the trade-off between effectiveness and simplicity. Also, since the PID controller has been widely used in industrial applications, it will be economically attractive to implement the predictive approach in existing PID control loops and adapt them to network communication.

While there is a large volume of theoretical work that has been carried out in DNCS, there are still challenging problems for control theory that need attention. The distributed estimation algorithm requires further extension, where network induced constraints such as delays and dropouts are included in the analysis (Han, Peng and Fei, 2016). Moreover, a unified study of the effect of input and output constraints and disturbance in the state estimation for systems with strong interactions has not been done. This is the motivation to develop simple control structures that can handle time-varying delays, consecutive dropouts, disturbance compensation and constraints.

1.3 Aims and objectives

The aim of this work is to design simple control methods for NCS and WNCS that can compensate for constraints of the network such as time-varying delays and dropouts. For this, the following objectives are targeted:

- To review and produce a comparison of the robustness and performance of the PID methodologies for time-varying delays and dropouts in NCS. To design new PID controllers using robust and optimisation techniques, Smith predictor and Internal Model Control (IMC).
- To design Model-based predictive controllers with a PID control structure to compensate for the dropouts in WNCS. To add a Kalman filter and a dropouts detector to the control scheme as an estimation algorithm to compensate for the dropouts from sensor to controller.
- To extended the prediction control techniques to the development of constrained Predictive PID (PPID) controllers that achieve stability in WNCS, where the communication is subject to dropouts in both communication routes: sensor to control and control to actuator transmission. To include constraint handling to stop input saturation.
- To test the control system using a network simulator. To provide several scenarios to demonstrate each algorithm and its effectiveness.
- To extend the predictive approach to DNMPC and test the result in an industrial context (wind farm control). To study a cooperative exchange of information to

achieve a global performance objective in systems with interactions. To develop a decentralised estimation algorithm to maintain information of the sensor packets.

1.4 Outline of the Thesis

This section explains the structure of the thesis and the contribution of each chapter.

1.4.1 Chapter 2: Networked Control Systems: An overview

This chapter involves an extensive literature review of developing areas related to networked control. It covers the key topics in NCS and the benefits and drawbacks of the implementation. The chapter clearly defines the problems of the network that still require more research and points out some possible research directions. The main focus is to create a framework where NCS approaches are divided according to the type of control and constraints that they compensate. The structure of the chapter is as follows: Section 2.1 explores the definition and overview of NCS. Characteristics, structures and modelling of NCS are described in Section 2.2. Section 2.3 explains the network induced constraints. The specific case of WNCS is studied in Section 2.4. Finally, a review, analysis and categorisation of the proposed control strategies of NCS are presented in Section 2.5. Applications of NCS are covered in Section 2.6.

1.4.2 Chapter 3: PID design methods for NCS

This chapter presents significant developments in NCS based on PID, IMC and Smith Predictor algorithms. The main purpose of this chapter is to study the performance and robustness offered by these methods when handling systems subject to time-varying delays and dropouts. For this, the proposed methods have to achieve the design requirements, such as margins of robustness, performance criteria and stability conditions. The performance of the controllers is evaluated and extensive simulations of these methods are presented. The structure of the chapter is as follows: Section 3.2 presents the design of the Smith predictor controller. Section 3.3 reports a new adaptive IMC for NCS. Section 3.4 proposes a robust compensation scheme for uncertain time delays acting on NCS. The design of an optimal PID controller for NCS with time-varying delays is presented in Section 3.5. Section 3.6 includes the design of an optimal robust PID

controller using the maximum sensitivity. A design of a jitter-aware PID is shown in Section 3.7. A new approach is proposed in Section 3.8, where an optimal immune PID controller has been applied to an NCS subject to dropouts and time-varying delays. Finally, Section 3.9 outlines a comparison of performance among the different techniques.

1.4.3 Chapter 4: Wireless networked predictive PID control design for packet dropouts

In this chapter, five new control strategies based on MBPC with PID control structure have been studied to compensate dropouts. The chapter focuses on the receding control theory combined with a Kalman filter to achieve a control system and an estimation algorithm that successfully resolved two main problems in the WNCS: missing sensor packets and control actions. The proposed strategies are divided into the following sections: The problem formulation of MBPC and PID is proposed in Section 4.1. Section 4.2 presents MBPC strategy for dropouts in WNCS. Networked predictive control for consecutive dropouts is developed in Section 4.3. In Section 4.4, a parallel predictive PID approach is illustrated. Predictive control for dropouts with an augmented state-space model is shown in Section 4.5. In Section 4.6, the control performances of the four methodologies are compared. Section 4.7 presents a new constrained predictive PID controller to achieve stability in WNCS and shows through several experiments the performance and robustness of the controller. Finally, in Section 4.8, a summary is presented.

1.4.4 Chapter 5: Decentralised wireless networked model predictive control design for complex industrial systems

In this chapter, the inclusion of the network communication and the compensation of its constraints in DMPC are studied. Section 5.1 presents a framework of comparison of the different DMPC approaches for NCS and discusses the optimality of the solutions. It also shows the many points that need further attention. Section 5.2 presents a description of structures NCS and characteristics. In Section 5.3 a constrained decentralised networked predictive control is developed with a coordination strategy. A decentralised Kalman filter solution is added to the control scheme to compensate dropouts. The effectiveness of the proposed method under disturbances, strong interactions and the reduction of information exchange is studied in Section 5.4.

1.4.5 Chapter 6: Decentralised wireless networked model predictive control design for wind turbines

In this chapter, the design of Chapter 5 is extended to the application of wind farm control, where the DNMPC algorithm has been exploited to manipulate the power of a wind farm of several interconnected wind turbines. Firstly, the description of the wind turbine system is presented in Section 6.2. Then, a description of the wind farm control methodology and the network communication is presented. In Section 6.4, the dynamics and model of the wind turbine are discussed. Next, Section 6.5 explains the control problem and methodologies employed to control it. Decentralised and distributed control is also incorporated in the section. The multiple objectives and specifications to meet by the proposed control system are introduced in Section 6.6. The steps of the control design are explained in next section. Simulations results of the wind turbine control and the configuration of the network are detailed in Section 6.8. Finally, discussions and conclusions are presented.

1.5 Contribution of the thesis

1. The study, design and implementation of simple controllers to compensate for time-varying delays and dropouts in NCS are presented in Chapters 3 and 4. The topic is gaining an important role in academic and industrial applications and the contribution of this thesis offers a suitable, innovative and interesting solution for both fields.

2. A unified framework for comparison of the different approaches in NCS and a discussion of the contribution of them to the challenging problem of network constraints are presented in Chapter 2. The classification is based on the modelling approach and the network issues that were selected in the proposed solutions. Moreover, a dedicated literature review for the areas of MBPC and DMPC is presented in Chapter 4 and 5, respectively. The main contribution is the examination of the available strategies for network constraints in NCS and their achievement in the compensation of dropouts and time delays. It also highlighted the future research directions in this area.

3. The design and application of seven control methods for dropouts and time-varying delays in NCS are presented in Chapter 3. A conference paper has been

presented from the work herein. The systems have been analysed for different models of random and time-varying delays. Also, the solutions are suitable to compensate for different scenarios of dropouts. The analyses of the robustness, performance and stability of all methods have also been performed. The most relevant feature of all these strategies is that it adheres to a consistent rule of controller simplicity and flexibility. PID, Smith predictors and IMC approaches are the offered solutions. Since PID control provides the most extended popularity in process control, it will be very convenient to adapt the existing loops to network communication and apply the proposed controllers.

4. The key role of wireless communications in the optimisation of control systems has been stated by the research community. Despite the increasing contribution on dropouts compensation, none has reported a simple and flexible control strategy that investigates the robustness of WNCS with dropouts and control constraints. This thesis solidly contributes with the proposal of four new control methods to compensate for dropouts and fill this gap. These methods include MBPC algorithms, PID control and detection/compensation of consecutive dropouts of measurements and control actions, which have been presented in Chapter 4. The work has also been presented in a conference paper as detailed in Section 1.6. The contribution focused on the design of a robust predictive controller to the variations of the dropouts and the process parameters and disturbances. Moreover, the reported contributions guarantee the application of the control signal according to the actual limitations of the controller.

5. It can be pointed out that in the NCS approaches the loss of information is generally assumed only in one way. The constraints in the controller input are not taken into account in the control systems. In comparison, a new constrained predictive PID controller for WNCS presented in Chapter 4 has the advantage of addressing both communication ways and dealing with constrained systems. It compensates for higher occurrences of dropouts and the stability with robustness to disturbance and system gain and pole variations have been proved. A journal paper has been submitted from this work.

6. An exhaustive review of DNCS is presented in Chapter 5 and showed the many points that need more attention. In this area, the contributions of this thesis are twofold. First, an innovative DNMPC solution has been introduced to provide effective compensation for dropouts and delays in a wireless environment. A

decentralised estimation algorithm that considers dropouts in the analysis has been added to the control scheme. Second, the performance of decentralised control is improved through the implementation of a coordination strategy. The effects of disturbances, strong interactions, management of constraints and the reduction of information exchange have been investigated. An abstract from this work has been submitted for journal publication.

7. The DNMPC algorithm has been exploited to manipulate the power of the wind farm of several interconnected wind turbines. This work is presented in Chapter 6. A journal paper containing the work of this chapter has been submitted. The control strategy could meet the multiple objectives of regulating electrical power, generator speed and torque. The application provided results within the desired requirements of power fluctuations and frequency response. The proposed controller provides good performance while compensating for high percentages of dropouts and acceptable robustness.

8. Most importantly, the developed algorithms were tested in a Wireless Local Area Network (WLAN) simulator and exhaustive tests demonstrated the effectiveness of the proposed solutions.

1.6 Publications

The following publications have been presented:

- Vasquez, M. C., & Katebi, R. (2016). Comparison of PID methods for networked control systems. In 2016 UKACC International Conference on Control, UKACC Control 2016, Belfast, Northern Ireland, pp. 1-6.

- Chacón, M., & Katebi, R. (2017). New Predictive PID Controllers for packet dropouts in Wireless Networked Control Systems. In 15th International Conference on Industrial Informatics, INDIN 2017, Emden, Germany. Presented as a conference paper.

- Chacón, M., & Katebi, R. (2017). A new constrained predictive PID control for packet dropouts in Wireless Networked Control Systems. Submitted for Journal publication. IET.

- Chacón, M., & Katebi, R. (2017). Decentralised wireless networked model predictive control design for complex industrial systems. Submitted for Journal publication. IET.

- Chacón, M., & Katebi, R. (2017). Decentralised wireless networked model predictive control design for wind turbines. Abstract submitted for Journal publication. AIMS Electronic Engineering.

Chapter 2

Networked Control Systems: An overview

2.1 Overview and definition

NCS and WNCS are defined as a set of components connected to a network communication (wired or wireless) that manipulate the behaviour of a process, device or machine to achieve a desired performance.

The networks considered in control systems are Building Automation and Control Network (BACnet), Fieldbus (Field Bus), PROFIBUS (Process Field Bus), Ethernet, Control Area Network (CAN), Round Robin, Switched Ethernet, Internet among others. Some network protocols for Wireless are IEEE 802.11b/g (WLAN) and IEEE 802.15.4 (ZigBee).

NCS have been successfully introduced into simple and complex industrial control systems because of several benefits they offer, such as low cost, simplified installation and low maintenance and increase in plant supervision.

However, the inclusion of the network implies the consideration of control over non-perfect communication channels. According to the nature of the NCS, the design task deals with signal sampling, asynchronism in the communication, scheduling, single and multi-packet transmission, quantization, disorder of packet arrival, among others. These will be further discussed in Section 2.4.

In recent years, the control engineering community has developed a full range of control schemes to cope with these problems. The following sections are intended to review such control strategies for NCS to examine the improvement they offer. Finally, it will evaluate the validity of each solution.

2.2 Characteristics of Networked Control System

2.2.1 QoS

To provide good Quality of Service (QoS) to end users, problems such as congestion and network induced delays and dropouts must be controlled in network communications. The QoS must comply with the transmission of information on time and accurately.

However, control engineers generally do not consider the design for a desired QoS directly. Instead, they deal indirectly with it by solving the problem of the percentage of dropouts. They ensure stability and measure NCS performance using traditional cost criteria.

Bjorkbom (2010) provides a study of the relationship between control performance and QoS which is one of the few approaches in this area.

2.2.2 NCS structures

A typical structure of NCS is shown in Figure 2.1. The closed-loop includes the actuator and the sensor devices, which have been defined in the block diagram. The controller signal is u, y stands for process output, e is the error and r is the reference signal. $K(z)$ is the controller and the controlled plant is $G_p(z)$. An NCS whose sensor and control information is transported over a wired/wireless network is considered. This scheme captures the time delays in the control input and output. Network induced delays can be represented as sensor to controller and controller to actuator delay: τ_{sc} and τ_{ca}, respectively. Similarly, the dropouts from the sensor to controller and from the controller to actuator are represented as d_{psc} and d_{pca}, respectively.

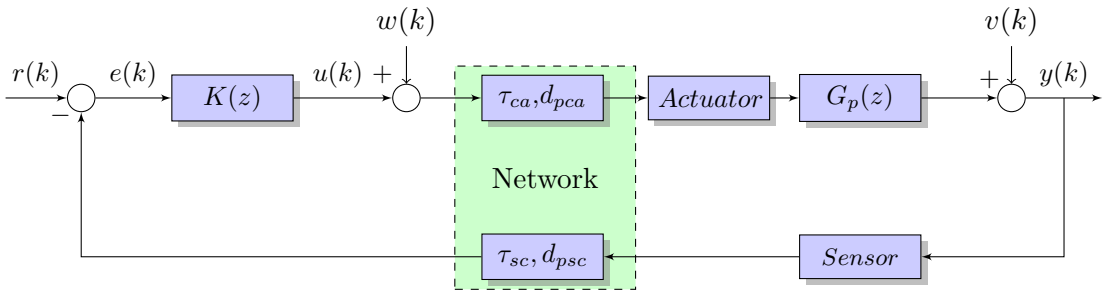

FIGURE 2.1: Typical NCS structure

2.2.3 Network control systems model

Modelling networked control systems can be done by considering the following Linear Time-Invariant (LTI) system S:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) + \mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{v}(k) \end{aligned} \tag{2.1}$$

where $x(k) \in \mathbb{R}^n$ is the state vector, $u(k) \in \mathbb{R}^m$ is the vector of control actions, $y(k) \in \mathbb{R}^{ny}$ is the measurement vector, $w(k)$, $v(k)$ are the process and measurement noise sequences, respectively. A discrete-time setting is assumed and the current time is labelled as $k \in \mathbb{N}$.

In uniform sampling, the control system is checked every sampling time Ts. A hybrid representation can be done by combining the continuous characteristic of the system with event or time driven actuators and controllers.

To analyse the communication delay, it can be transformed into an input delay system, assuming a linear control law:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) + \mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{v}(k) \\ \mathbf{u}(k) &= \mathbf{K}\mathbf{x}\big(k - \tau(k)\big) \\ k &\in [kTs + \tau_k, (k+1)Ts + \tau_{k+1}] \end{aligned} \tag{2.2}$$

where $\mathbf{K}$ is the state-feedback controller gain and $\tau(k)$ is the network delay that satisfies:

$$\tau(k) \in [\tau_{min}, \tau_{max}] \tag{2.3}$$

where

$$\tau_{min} := \min_k\{\tau(k)\}, \quad \tau_{max} := \max_k\{(k+1)Ts - kTs + \tau(k+1)\} \tag{2.4}$$

Dropouts can be viewed as a delay greater than the maximum bound. Therefore, for a maximum number of consecutive dropouts m, (2.4) is given as follows:

$$\tau_{min} := \min_k\{\tau(k)\}, \quad \tau_{max} := \max_k\{(k+1+m)Ts - kTs + \tau(k+1+m)\} \tag{2.5}$$

In the following sections, the modelling for dropouts and delays using stochastic or deterministic approaches is studied.

2.3 Network constraints

The inclusion of a communication network into the structure of feedback control systems presents a new challenge in the modelling, analysis and design of NCS. There are intrinsic and inevitable network phenomena that the control system has to cope with to improve the performance of the NCS. The most important communication problems caused by sharing the network by multiple nodes can be divided into variable transmission intervals, variable delays and dropouts (Heemels et al., 2010). The time delay occurs during transmission, that is when sending data from the sender to the receiver. Often, the network is congested and the information does not arrive on time. Therefore, communication in the NCS is ineffective. Moreover, packet dropout is a serious problem for the quality of service provided. In what follows, the consideration of these challenges is extended.

2.3.1 Limited capacity

The network is a communication channel with a limited capacity to carry information, that is, the channel can take a limited amount of data per time. Due to this limitation, the data cannot be sent and received correctly all the time. As a result, the reliability of the NCS is reduced.

A digital network is considered where the information unit is called a packet. Each packet contains the data that is required to transmit. The data are packaged in one bit or hundreds of bits according to the frame or protocol that the network is modelled. The

theoretical capacity of the channel can be found in Shannon's theory. This indicates that the amount of data that can be transmitted reliably through a channel is limited. In fact, the closed-loop stability of the system depends on the packet rate that can be sent through this network.

Industrial buses known as PROFIBUS, CAN and Fieldbus are examples of networks with limited bandwidth. For the former, 1 Mbit/s is the standard value and can be improved up to 12 Mbits/s using PROFIBUS-DP (Decentralised Peripherals) at the expense of raising the cost (Xia et al., 2005). Despite this, these are the standardised control communication networks that support the requirement of a real-time and deterministic characteristic required by the control system.

Ethernet is another well-used network for monitoring high and low levels of control in industrial applications. It has gained an important role and replaced the use of buses during the recent years (Decotignie, 2005). The main benefit of this network has been the increase in data rate, which generally allows up to 1000 Mbits/s for Gigabit Ethernet. However, the drawback of this is its nondeterministic behaviour. Some approaches in switching technology have been made to turn it into a deterministic process (Moyne and Tilbury, 2007).

In the case of wireless networks, the limited capacity can be a tighter restriction. For example, ZigBee allows until 250 Kbits/s and Wifi offers up to 600 Mbits/s (protocol 802.11n).

2.3.2 Network delay

It is known that a delay in a control loop typically affects the performance and stability of the control system. In the worst case scenarios due to the presence of time delays, all system stability is threatened. Furthermore, as quoted by Wang and Liu (2008), this situation is not tolerable for applications in which time is critical, such as a process of extinguishing fires, underwater operations and automatic driving on the road.

A communication delay consists of two main parts: the access delay and propagation or transport delay (Mazumder, 2011). The first refers to the time the data waits at the sending points to transmit. The second, responds to the time required to send the data through the medium and will depend on the distance and the speed of the

signal. The access delay, τ_{access}, and the propagation delay are generally referred to as network-induced delays.

The nature of the network-induced delay depends on the hardware and software of the network: network characteristics, such as the protocol that controls transmission, network load, topologies, the number of nodes, route schemes and scheduling policies on the network and nodes.

There are some categories for time delays. A possible division is between constant and time-varying delay. Constant delay, τ_c is less common because it can only be used with deterministic protocols with high precision programming times (Ungan, 2010). In this case, the longest delay is used for the value of the time delay $\tau(t) = \tau_c$.

The stochastic delay can be represented using random models. These are modelled by discrete probability functions using distributed variables with a designed variance. For example, Bai, Fu and Su (2011) used a Poisson process to model the delay in wireless networks.

Other examples for the random delay and the correlated delay are given by Ungan (2010) as follows:

$$\begin{aligned} \tau_1(t) &= x_1, X_1 \sim N(\mu, \sigma) \\ \tau_2(t) &= \int_0^t e^{-q\alpha} x_2(t-\alpha)\, d\alpha, X_2 \sim U(a_{min}, a_{max}) \end{aligned} \tag{2.6}$$

where x_1 is a Gaussian random variable with mean μ and variance σ^2, x_2 is a uniformly distributed random variable on $[a_{min}, a_{max}]$, and q is a filtering coefficient.

A Markov chain is presented by Qiu et al. (2015) to model the delays with the benefit of capturing the dependency within the delays. They used finite-state discrete-time homogeneous Markov chains.

The work of finding the conditions that guarantee the stability of the system subject to random delays is very complex. Although many proposals have been made in this regard, a criterion that can be applied as a standard has not yet been established.

2.3.3 Data packet dropouts

Another challenging problem in the NCS is the loss of information, commonly referred to as data packet dropouts or dropouts. A simple definition is given by Wang and Liu (2008) who pointed out that the network protocols have a limit time designed for the data to be transmitted, and in the case that this maximum interval is exceeded, the data is discarded. Therefore, packets never reach the receiver and this event becomes a loss of information that is considered as a serious situation because it degrades the reliability of the NCS.

Dropouts are a consequence of errors during transmission, noise, or congestion that lead to buffer overflows. According to the most widely used network protocols, during data transmission, routers are used to guide information over links with limited bandwidth. If the data reaches the router when it is temporarily overloaded, they are sent to a temporary storage area. However, when this storage capacity is exceeded, the link is congested and data are lost. Some other considerations are likely to be found in the literature. For example, dropouts may also occur when long transmission delays are presented. In this situation, the receiver chooses to discard packets arriving outside of the time, which is considered a dropout (Mazumder, 2011).

The occurrence of dropouts during transmission from the sensor to the controller and from the controller to the actuator results in an open-loop system that degrades the reliability of the NCS.

Dropouts can be modelled using a stochastic or deterministic approach. The first mostly uses the Bernoulli process. For example, Sinopoli et al. (2004) showed a critical value of the dropout percentage to ensure system stability. However, in these analyses, the critical value is found by using the lower and upper limits of the percentage (Li et al., 2015). Wenlong, Bae and Tomizuka (2015) represented the dropouts for both controller actuator and sensor to controller using Bernoulli variables. These variables are expressed as follows:

$$d_{pca}(k) = \begin{cases} 1, & \text{with probability of } \lambda \\ 0, & \text{with probability of } 1-\lambda \end{cases} \tag{2.7}$$

$$d_{psc}(k) = \begin{cases} 1, & \text{with probability of } \rho \\ 0, & \text{with probability of } 1-\rho \end{cases} \tag{2.8}$$

where λ and ρ are the probabilities of unsuccessful transmission in the controller to actuator and sensor to controller, respectively. Thus, a dropout from the controller to the actuator is defined as $d_{pca} = 1$ and a sensor to controller dropout as $d_{psc} = 1$.

However, this is an ideal abstraction to represent this problem since the temporal and variable nature of the events generally causes correlations between them that affect the arrival of the packets. Therefore, another modelling process is suggested to address the presence of packet correlations. For example, Ungan (2010) approximated dropouts using a Markov chain model that has two states to represent the transmission of binary data. A state illustrates the case that packets arrive successfully. The second state is in the case of failure. The probability of failure is q and the probability of recovery is p. Further examples are studied by Zhang, Gao and Kaynak (2014) and Park, Marco, Soldati, Fischione and Johansson (2009).

In the deterministic approach, the analysis is based on established bounded values to the dropouts. For example, Wang and Liu (2008) pointed to the fact that there is a limiting number of dropouts that can be tolerated to ensure system stability. Therefore, this research and many other similar approaches performed an analysis where a stability condition parameter η, represents the maximum number of dropouts. Moreover, further analysis emphasised the severity of not only dropouts but also the impact of the successive dropouts to the NCS stability. Therefore, it is required to know the maximum number of consecutive dropouts that can guarantee the system stability. Some recent examples are explained by Sun et al. (2014a) and Li et al. (2014).

2.3.4 Quantization

Despite the great interest in solving the stability problems introduced by dropouts and delays in the network, some authors argue that there are other concerns that can not be ignored. For example, Rasool, Huang and Nguang 2011 argues that the design of any controller must deal with the fact that in real communication networks packets are sent through the medium with finite precision. Also, the size of the packets is limited according to the conditions in the network.

Each control system requires the data to be converted into digital information to be sent to the network. The accuracy of the information that results after this conversion depends on the precision that is used. As a result of poor precision, the uncertainty of the system may increase. Therefore, this restriction plays an important role in performance, since the stability of the NCS can be put at risk for the effect of data quantification. An ordinary thought might be to increase the number of bits that can be sent in packets to reduce system uncertainty. However, increasing the length of the packet will cause more congestion in the network, since it means that more information must be transmitted.

Two types of quantification are described in the literature: static and dynamic. Static quantification is the simple way to overcome this problem using schemes that do not change over time. Two types of quantifiers can be found for static quantification, uniform and logarithmic. These are representations of fixed-point and floating-point numbers, respectively.

There is an increase in the interest of dynamic quantifiers. They use an optimum approximation of the signals at the control input and output according to the dynamics of the plant. Such examples are given by Lu, Xu and Tian (2012) for a discrete dynamic quantizer and Sawada and Shin (2012) for a continuous one.

2.4 Wireless networked control systems

Mobile communication technologies have shown an impressive growth since their first application over forty years ago. Following this tendency, it can be guaranteed that WNCS are going to be required to manipulate an increasing number of mobile systems. Moreover, these systems are present in all sectors of daily life, from personal, residential, commercial to industrial applications.

Some benefits of WNCS that make them desirable for many applications are cited below. Firstly, and one of the most important for process automation is the reduction of operation and wiring costs in long-distance control systems. Moreover, they offer flexibility, easy installation, maintenance and facilitate the reconstruction. Regarding safety, mobile systems are being used to protect human operators from dangerous scenarios such as those with the risk of explosions or high temperatures. For example,

mobile robots are used in the design of mines and extra-spatial explorations and wireless controllers are implemented to send them the commands over the network.

However, the application of wireless communication in the industry is almost limited to monitoring (Hassan, Ibrahim, Bingi, Chung and Saad, 2017). For instance, the use of wireless technologies in Wireless Sensor Networks (WSNs) is the most important application in this field. An attempt to use wireless measurements in control applications has been made by Blevins, Nixon and Wojsznis (2014).

As it was mentioned earlier in this chapter, wireless communications bring outstanding advantages but also their limited capacity leads to significant technical barriers that limit the application of wireless technologies in process control.

Unlike wired communications that have at their disposal all the bandwidth of the medium to transmit information, wireless communications must deal with the fact that the electromagnetic spectrum is unique and must be shared with all wireless services. Moreover, as rapidly as the mobile devices increase, the concern that radio spectrum is limited puts a special attention to this issue.

Also, the problems that appear with the inclusion of the network become more significant in the wireless communications due to its constraints, such as limited energy and computing power, channel fading, time-varying capacity, transmission delays, packet losses and out-of-sequence data (Du and Du, 2009). Moreover, those constraints are closely related, for example, non-periodic measurements are made to conserve the limited energy in the wireless nodes (Tiberi, Araujo and Johansson, 2012).

One of the first works that addressed the complexity and intricate estimation of these problems was proposed by Wong and Brockett (1999). They illustrated a closed-loop system where a channel is introduced to communicate between the process and the controller. Moreover, the finite communication bandwidth imposed a delay on the transmission of the information that leads to a more complicated estimation of the process. To overcome the bandwidth constraints several approaches have been proposed. In the following section some of the most relevant are discussed.

2.5 Approaches for NCS

In this section, the main methodologies suggested in the literature to cope with typical network-induced constraints and packet losses are discussed. The methodologies suggested in the literature are presented in different categories according to the control method that is implemented. The section ends with a table that summarises, the constraints and the most relevant control approaches to address them.

2.5.1 PID control

While numerous approaches have been reviewed such as fuzzy, predictive, H_∞, event-based and event-triggered control, the PID has received the most attention in the history of process control. However, only a small number of PID methods for NCS have been proposed with simple structures. Even less is the numbers of approaches that offer tuning rules for NCS and the stability is guaranteed for a limit value of time delay and a restricted number of process models.

Relevant work has been proposed where the PID tuning problem for time-varying delay systems has been approached using multi-objective optimisation to develop rules that maximise the jitter margin, i.e. the maximum value of any additional time-varying delay in the control system. See for instance, the approaches of Eriksson and Johansson (2007b) and Eriksson and Oksanen (2007). Constraints can also be added to guarantee desired gain and phase margins as presented by Eriksson and Koivo (2005). This approach was tested by Pohjola, Eriksson and Koivo (2006) in a network simulator and showed satisfactory performance for a simple process model.

There are some networked PID control methods to compensate for delays and dropouts. For instance, Dasgupta et al. (2015) addressed the closed-loop stability of a NCS under time-varying delays and dropouts with a discrete PID controller. Similarly, Tran et al. (2013) focus on robustness to develop a normalised PID controller. Moreover, Okano, Ohtani and Nagashima (2008) examined the performance of PID control under dropouts. Zhang, Shi and Mehr (2011) designed a digital PID controller via the robust static output feedback. These approaches offered stable solutions and the main advantage of extending NCS to practical industrial control. Ungan (2010) combined a PID controller with a time-varying Kalman filter and tested using IEEE 802.15.4 wireless equipment. Two main problems were found, firstly, the control system presented a bad performance

when consecutive dropouts occurred, since they occur in a recurrent way and the estimation algorithm has not enough time to predict. Secondly, for higher periods of loss of communication the model was not accurate enough leading to a permanent error in the output. More applications of wireless technology in monitoring are cited by Hassan, Ibrahim, Bingi, Chung and Saad (2017) and Blevins, Nixon and Wojsznis (2014). Moreover, one industrial control application of PID in WNCS is developed by Abdullah, Ibrahim, Hassan and Chung (2016). The results showed compensation of network constraints while noise was acting in the system. These approaches demonstrated the potential of less complex and easy to implement solutions.

As introduced in Chapter 1, predictive PID control can offer suitable performance in NCS. Very few studies have been focused on predictive control algorithms with PID structures that can effectively compensate dropouts. For instance, Miklovičová and Mrosko (2012) addressed the compensation of control dropouts using GPC and pole placement structure to design a PID controller. Tan et al. (2002) combined the optimal tracking control of the GPC and the PID structure. Hassan et al. (2016) presented a predictive WNCS to compensate variable delays and disturbance, where a Smith predictor is combined with PID control. A similar solution is postulated by Wu, Wu and Zhao (2016) to compensate random delays and dropouts. Friman and Nikunen (2013) offered a PID controller with a simple estimator. The results showed a permanent error as a consequence of the trial and error mechanism used to select the design parameter of the estimator.

The design of WNCS results in a trade-off between reducing the number of communications and maintaining the performance. Blevins, Nixon and Zielinski (2013) showed an industrial application using wireless communication in a closed-loop control. They developed and tested a PID implementation that reduces the power consumption by a reduction of the number of communications of wireless transmitters (WirelessHart protocol) at the same time they manage to maintain a similar performance to a wired control system. Moreover, they showed the adequate performance to cope with the loss of communication.

2.5.2 Predictive control

Among the large number of methods available, the Networked Predictive Control scheme (NPC) is considerably effective, since it can actively compensate for the transmission delays and consecutive packet dropouts (Sun, Wu, Liu and Wang, 2014b). Recent publications in this area provide theoretical developments and simulation results.

The first idea to exploit the predictive control in NCS can be referred to Bemporad (1998) and its design of a predictive control in communication channels with unlimited delays. They improved a predictive controller to overcome with the long arbitrary temporal delays.

Pin and Parisini (2011) designed a nonlinear NPC with a network delay compensation for a network User Datagram Protocol (UDP). Their results indicated a significant contribution, however, some assumptions that can not be guaranteed in a practical implementation such as synchronisation of all nodes and accessibility to all measurements. Du and Du (2009) proposed a Smith predictor combined with a Cerebellar Model Articulation Controller (CMAC) and PID control for WNCS. The proposal hid the predictor models of the network delays into real network data transmission processes. Based on IEEE 802.11b/g (WLAN), simulation results showed the validity of this control scheme. The predictor is combined with a non-linear PID control by Du, Du and Lei (2009b) and with a PI control by Du, Du and Lei (2009a).

Sun, Chen and Dou (2014) presented a predictive control scheme for time-varying delay. They obtained a stability condition using a Lyapunov switched function approach and tested on a servo control system. Finally, they demonstrated the stability of the system with some illustrative examples. Liu, Chen and Zhu (2014) proposed a robust NPC for systems with time-varying delays. Pang (2017) provided another contribution for networked non-linear systems that combined the effect of dropouts and random delays. Franzè, Tedesco and Famularo (2015) addressed the stabilisation of constrained systems under dropouts. Model predictive tracking control for dropouts and process uncertainties is provided by Lu, Xu and Zhang (2016).

Experimental results for NCS are cited by Onat et al. (2011) who implemented a predictive control scheme to guarantee the stability of the control system under random delays

and dropouts. The results using Ethernet and IEEE 802-11 wireless showed the effectiveness of this approach regarding control performance. Ulusoy, Gurbuz and Onat (2011) offered a strategy that combined the design of Media Access Control (MAC) parameters and dropouts and delays of the wireless network.

Tang and Silva (2006) presented an adaptive predictive controller which showed a reasonable performance for an experimental test using UDP and Transmission Control Protocol (TCP). However, for higher delays the performance declined and a proposed gain scheduling is proposed as a future improvement.

Other examples are given by Liu et al. (2007) who built a state observer to develop solutions that guarantee system stability for a constant and random network delay, both limited to a maximum value. Real examples have been presented using a DC motor implementation and communication UDP. The main drawback is that, it is difficult to manage changes and then controls due to the effect of large system matrix. Du et al. (2014) followed this approach to formulate a sufficient condition stability for network predictive control systems.

An important issue with NPC is the controller sensitivity to errors in the delay model. Since NCS are subject to time-varying random delays, it is recommended to include a robustness analysis where the effect of variations in the model parameters is studied.

2.5.3 State feedback controller

A typical solution found in the literature is the formulation of the NCS as a sampled-data system. For this, consider the linear state-space system representation in Equation (2.2). A feedback digital control algorithm is developed for the NCS by a Linear Matrix Inequality (LMI) approach that guarantees stability for a bounded maximum delay. When control action is not updated during the sample time, it is just a special case of longer delay than one sampling period and might consider as packet dropout. The principal benefit of this approach is that this digital system simulated the effects of the delay and dropouts in the network. (Hu, Bai, Shi and Wu, 2007).

In Montestruque and Antsaklis (2003) a stability analysis for NCS has been proposed where a state feedback controller considers time delay. Although this approach can only be used to view the systems in the sensor to controller delay case and does not consider dropouts, it has been used as a reference in some publications.

Yue, Han and Peng (2004) developed a NCS model which considers time delay and dropouts. The analysis is carried out for a continuous-time domain. In this approach, the restriction of N number as the maximum value that can be tolerated for the dropouts is assumed and stability criteria are found using Lyapunov theory and by solving a set of LMI. Some illustrative examples were performed and they showed the effectiveness of the design. The drawback of this approach is that it uses a simplistic model, which is generally poor for representing the stochastic behaviour of the NCS.

The problem of energy consumption has been considered by Zhang, Gao and Kaynak (2014) who designed a WNCS focusing on the reduction of energy consumption for an IEEE 802.15.4 network using Carrier Sense Multiple Access with Collision Avoidance algorithm (CSMA/CA). They studied the Media Access Control (MAC) parameters and the sampling time, dropouts and random delay of the network and derived a stability condition that reduces the energy consumption. They found the stability conditions and solved a constrained optimisation problem where the optimal solution changes the sampling time and tuning parameters (maximum backoff limit and maximum retry limit) according to the network conditions.

Two tables summarise the typical control approaches presented in the literature review to compensate network constraints. Tables 2.1 and 2.2 list the references addressing the constraints for SISO systems. Note that, Case 1 refers to State Feedback Control, Case 2: Fuzzy control, Case 3: PID control, Case 4: Predictive control, Case 5: Wireless and Case 6: H_∞ control.

TABLE 2.1: References with respect to network delays

Case	Time-varying delays
1	Montestruque and Antsaklis (2003)
2	Feng and Du (2010), Chen et al. (2014)
3	Eriksson and Oksanen (2007), Eriksson and Johansson (2007b), Eriksson and Koivo (2005)
4	Montestruque and Antsaklis (2003), Liu, Chen and Zhu (2014), Bemporad (1998)
5	Eriksson and Koivo (2005)

TABLE 2.2: References with respect to network delays and dropouts

Case	Time-varying delays and dropouts
1	Yue, Han and Peng (2004), Hu et al. (2007), Heemels et al. (2010), Onat et al. (2011)
2	
3	Pohjola et al. (2006), Du and Du (2009), Tiberi et al. (2012), Friman et al. (2000)
4	Sun et al. (2014), Sun, Chen and Dou (2014), Pin et al. (2011), Onat et al. (2011), Du et al. (2014), Tan et al. (2006), Liu et al. (2007)
5	Zhang et al. (2014), Onat et al. (2011), Ulusoy et al. (2011), Ungan (2010), Du and Du (2009), Friman et al. (2000)
6	Yue et al. (2005), Gao and Chen (2008)

2.6 Applications

There is a wide application area for NCS. They have been successfully applied in remote control and large-scale systems such as smart power grids, sensor networks, network traffic control, water networks, automatic and intelligent building management systems, unmanned aerial vehicles (UAVs), advanced aircraft, industrial automation, signal processing, remote surgery systems and manufacturing processes. Other applications are space exploration, land exploration, factory automation, remote diagnostics/troubleshooting, hazardous environments, experimental facilities, domestic robot navigation and automotive (Wang and Liu, 2008).

To organise the study of NCS applications, five main areas of application have been established by Murray et al. (2003), including aerospace and transportation, information and networking, robotics and intelligent machines, biology and medicine, and materials and processing.

The use of NCS in some of these areas will be described below with a few examples.

2.6.1 Teleoperation

Teleoperation was the first application of NCS to be popular and showed that the introduction of networks in the control system is a convenient configuration for its applications. According to Zhang, Gao and Kaynak (2013) remote control or teleoperations can cover the execution of tasks from long distance positions, for example, space explorations.

The teleoperation control configuration consists of a master and slave telemanipulation in which the dynamics of the controllers and a communication channel interact with human and environmental factors (Arcara and Melchiorri, 2002). The teleoperation includes the communication channel characterised by a delay time.

This application can be very useful when using a large network like Internet. Internet-based teleoperation was used in telerobotics, remote manufacturing, telesurgery and education.

The importance of this application is highlighted: “Teleoperation research began with concern for safety and convenience in hazardous environments, such as space projects

and nuclear power plant and was only possible after NCS development" (Gupta and Chow, 2010a, p.2527). The use of robots for these tasks is a common example, for instance, Tipsuwan and Chow (2004) developed an IP controller for mobile robot path-tracking teleoperation. Also, Bemporad (1998) showed this application for the control of a servomechanism.

2.6.2 Automotive local area networks

The vehicle industry is an example of the use of NCS. Many devices need to be monitored in a vehicle such as an engine, active suspensions, Anti-lock Braking System (ABS), Acceleration Slip Regulation system (ASR), cruise control, infotainment and body electronics. The maximum configuration for the vehicle contains about 40 electronic control units. The use of NCS reduces the number of cables and networks used to connect sensors, actuators and controllers. In addition, it increases the reliability and productivity of the system (Johansson et al., 2005).

Manufacturers use automotive Local Area Network (LAN) technology to implement control algorithms. Each LAN has a specific purpose and protocol. The application of NCS began in 1983 with the introduction of serial buses in the control architecture. In the 1990s the CAN protocol was internationally standardised for real-time control. Other higher layer protocols such as DeviceNet and CANopen were also created. A low speed CAN network is used to control the air conditioner, door locks and meters. For the power train, including the engine, transmission and steering, a faster and more reliable network would be required (Bemporad and Barcelli, 2010). For instance, CAN with a communication speed of 500 kbps connects the motor and brake control. Car manufacturers base their control architecture on CAN, including Mercedes-Benz, BMW, Fiat, Renault, SAAB, Volkswagen, and Volvo.

Control architectures are increasingly being implemented in distributed computing systems and require many functionalities and coordination of many subsystems. At the same time, the control specifications for quality, safety and profitability should be meeting. For example, distributed architectures in marine applications include an autopilot based on a closed feedback control-loop over the network that shares information of rudder, GPS, pitch, roll and title, engine throttle, among others. Another application is the attitude and orbit control system for spacecraft, where control objectives must be meeting despite the hazardous environment. In automobiles, dynamic stability control

is introduced to assist the driver using relevant sensor data. The automatic control intervenes by applying breaks when necessary to increase the safety.

2.6.3 Networked and distributed wind farm control

The configuration of wind turbines that are part of a large-scale wind farms has become a leading source of clean and green energy. Wind turbines have gained considerable interest and investment. Therefore, their control is a major concern in power generation.

Large-scale networked energy systems are divided into smaller subsystems, usually monitored by a distributed control framework. Control of the wind farm is challenging due to the natural variation in wind and the coupling between the turbines due to the wake effects. The fast and precise control of the power generated by the wind contributes to the efficient operation of the wind farm. Robustness is also a major concern because the stochastic nature of the wind and therefore must be evaluated in the control design.

The wind farm control uses the signals with the state of each turbine to distribute the requested power from the farm. The adjust of power is manipulated using classic integral control. Then, the power references are sent to the local controllers in the wind turbines. The control signal of the local turbine and neighbouring turbines are transmitted through the network connection.

The cooperation between the turbines adds flexibility by allowing the individual turbine to adjust its power and reduce the load. There are distributed control schemes in which wind speed measurements of upstream turbines are transmitted to their nearest wind neighbours to improve performance.

There are also network architectures with supervisory control for optimal management and operation of power systems where wind turbines are included. For instance, MBPC scheme has been successfully introduced to optimise wind power references while minimising a cost function. The control objective is to manipulate the operating points of the wind turbine to generate enough energy to satisfy the load demand; while the maximum rates of increase of the power generated are constrained (Christofides, Liu and Peña, 2011).

Distributed MPC algorithm is also used for Load Frequency Control (LFC). Distributed model of an interconnected power system composed of several power areas including wind farms have been found effective to measure the effect ot the wind farm in an integrated generation system.

2.7 Summary

NCS are gaining an important role in academic and industrial applications since they reduce costs, simplify installation and maintenance and increase the supervision and control capabilities for the whole system. However, the inclusion of non-ideal channels in the closed-loop leads to challenging problems such as time-varying delays and consecutive dropouts, which greatly influence the stability and robustness of the control system.

Network-induced delays can be represented using random models. These are modelled by discrete probability functions using distributed variables with a designed variance. For example, Poisson process and Markov chain are proposed. The feasibility of the NCS to handle dropouts in the literature primarily rely on delayed measurements, Bernoulli process or Markov chains to model the data flow in the network. This research and many other similar approaches performed an analysis where dropouts and network-induced delays are assumed to be variable but bounded. Moreover, the severity of not only dropouts but also the impact of the successive dropouts to the NCS stability are also emphasized. Therefore, it is required to know the maximum number of consecutive dropouts that can guarantee the system stability.

While there is a large number of theoretical papers that have been done in NCS, there are still challenging problems for control theory that needs attention. Moreover, a single method to compensate all the problems introduced by the network and guarantee the reliability complex NCS has not been achieved yet.

Only a small number of PID methods for NCS have been proposed with simple structures. Even less is the numbers of approaches that offer tuning rules for NCS and the stability is guaranteed for a limit value of time delay and a restricted number of process models. Good results have been found where the benefits of PID controllers are combined with other control structures such as the Smith predictor. These solutions

attempt to study the stability conditions for the NCS and how to increase the maximum acceptable values of dropouts and time delays.

The feasibility of MBPC to handle time-varying delays and dropouts has been proved. However to cope with the network constraints the complexity of the controller increases. To address the easy to implement and less complex algorithms a few studies have been focused on predictive control algorithms with PID structures that can effectively compensate time-varying delays and dropouts.

The exploitation of the significant advantages of wireless communications to decrease costs in control systems has been so far very limited due to its constraints, such as non-periodic sampling limited bandwidth and energy in the wireless nodes. Thus, it is necessary to further extend WNCS applications and investigate the effects of wireless networks on the real-world operation. In particular, there is a need to explore simpler solutions in WNCS not only for power limitation, consecutive dropouts and time-varying delays but also for disturbances, constrained systems, variations of the model parameters, among others.

Chapter 3

PID design methods for NCS

This chapter presents significant developments in NCS based on PID, IMC and Smith predictor algorithms. The formulation of PID controllers provides a good opportunity to increase the reliability in the NCS and a challenge for the trade-off between effectiveness and simplicity. Also, since the PID controller has been widely used in industrial applications, it will be economically attractive to implement the proposed approaches in existing PID control loops and adapt them to network communication.

The performance and robustness offered by these control design methods in handling the challenging control problem encountered with systems subject to time-varying delays and dropouts are presented. The design requirements such as margins of robustness, performance criteria and stability conditions are introduced as well. A comparison of performance of the different techniques is presented in Section 3.9 using Matlab/Simulink-based TrueTime network simulator.

3.1 Introduction

It is harder to design NCS to meet performance objectives than stand alone process control systems. This is down to two important reasons. Firstly, it has to consider system dynamics and disturbances. Secondly, constraints of the network such as dropouts and long delays limit the performance of the control system.

The compensation of time-varying delays in NCS is a complex problem that requires controllers with high levels of performance and robustness to ensure reliability in the

control system. The need for a simple and flexible algorithm that fulfils these requirements has given birth to different methods of controlling NCS. In this work, novel methods of designing PID controllers for NCS are presented.

The first method is a Smith predictor combined with a PI controller that includes the network delays in both communication directions. Moreover, a stability analysis of the NCS through the measurement of the gain margin, phase margin and maximum sensitivity is offered. The second method is an adaptive IMC controller that has been created to adapt to the network requirements and to compute the parameters of the IMC model online. The third method is a robust PID controller that has been implemented for NCS where the network delay has been studied as an uncertain delay in the model. The controller parameters that ensure a desired margin of stability are found using gain and phase margin restrictions. Moreover, a study of performance and robustness of the NCS is presented. The fourth method is an optimal PID controller designed for NCS where the network delays have been modelled as Gamma function. An optimisation problem is proposed using the delay model and the PID gains are found by minimising a cost function. The optimal solution is tested in the network simulator showing the effectiveness of the proposed method. The fifth method applies exiting tuning rules of a PID controller that compensates time-varying delay. Moreover, the design has been tested to analyse the applicability in a NCS subject to the network delay. An optimal robust PID controller for NCS under random delays is the sixth method, developed by solving a new constrained optimisation problem that included constraints of maximum sensitivity to guarantee robustness. The seventh method presents a modified PI control with an immune feedback algorithm. In this work, the controller has been optimised for a NCS under time-varying delays and dropouts. The tests in the simulator show the effectiveness of the proposed algorithm.

3.2 Smith predictor controller design for NCS

This section addresses the compensation of time delay acting in NCS. To achieve this, a modified Smith predictor combined with a PI controller is examined. An extended stability analysis through the measurement of the gain margin, phase margin and maximum sensitivity has been conducted to prove the validity of this approach.

3.2.1 Introduction

The Smith predictor is a recognised algorithm for time delay compensation. The most important characteristic of this controller is that the design and performance will depend on the information of the process model and the time delay (O.J.M, 1958). In this analysis, a modified Smith predictor structure proposed by Du, Du and Lei (2009a) is considered. The controller's block diagram is depicted in Figure 3.1.

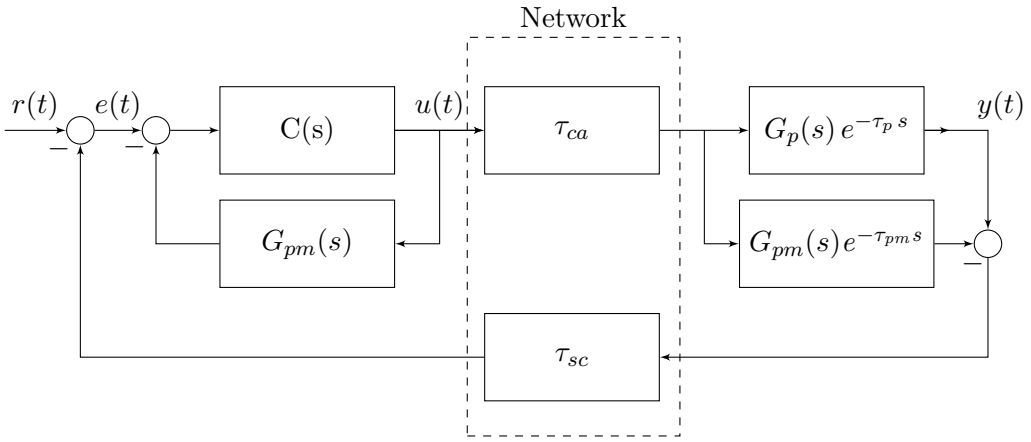

FIGURE 3.1: Smith predictor

$C(s)$ is the controller, that can be designed, as the process were delay free, $G_{pm}(s)$ is the prediction model of the controlled plant $G_p(s)$ and τ_{pm} is the prediction value of the process time delay τ_p.

The closed-loop transfer function of this system is given by:

$$\frac{y(s)}{r(s)} = \frac{C(s)\,e^{-\tau_{ca}\,s}\;G_p(s)\,e^{-\tau_p\,s}}{1+C(s)\,G_{pm}(s)+C(s)\,e^{-\tau_{ca}\,s}\,[G_p(s)\,e^{-\tau_p\,s}-G_{pm}(s)\,e^{-\tau_{pm}\,s}]\,e^{-\tau_{sc}\,s}} \quad (3.1)$$

According to (3.1) if the prediction model matches accurately the dynamics of the process model, the delays of the network can be effectively eliminated from the characteristic equation. However, for a real process subject to disturbances and variations of its parameters, the accuracy of the prediction model is not entirely guaranteed. Furthermore, an analysis of this situation and effects on the performance of the controller is presented in Section 3.2.3.

3.2.1.1 The PI controller tuning

For a good performance and robustness in the control design, a standard PI controller in the inner loop of the Smith predictor structure is tuned using the AMIGO rules from Åström and Hägglund (2004). The AMIGO rules have been derived for a first-order system with time delay and a maximum sensitivity, $Ms = 1.4$ as a measurement of robustness.

The sensitivity transfer function is implemented as follows:

$$S(s) = \frac{1}{1 + G_{ol}(s)} \tag{3.2}$$

where $G_{ol}(s) = C(s)\, G_{pm}(s) + C(s)\, e^{-\tau_{ca}\, s}\, [G_p(s)\, e^{-\tau_p\, s} - G_{pm}(s)\, e^{-\tau_{pm}\, s}]\, e^{-\tau_{sc}\, s}$.
By calculating the maximum sensitivity, Ms, it is possible to examine the robustness of the closed-loop control system to the variation of the model parameters. The maximum sensitivity is given by:

$$M_s = max_\omega |S(j\omega)| = max_\omega \left| \frac{1}{1 + G_{ol}(j\omega)} \right| \tag{3.3}$$

The robustness is guaranteed if there is at least a maximum sensitivity of $Ms = 1.4$ (Alfaro, Vilanova and Arrieta, 2010).

The standard structure of the PI controller is as follows:

$$C(s) = K_c \left(1 + \frac{1}{s\, T_i} \right) \tag{3.4}$$

where K_c is the controller gain and T_i is the integral time. The following tuning rules are used in the design:

$$K_c = \frac{1}{K_p} \left(0.2 + 0.45\, \frac{T}{L} \right), \qquad T_i = \frac{0.4\, L + 0.8\, T}{L + 0.1\, T} L \tag{3.5}$$

where K_p, T and L are the process gain, time constant and dead time, respectively.

The equivalent time-discrete controller has been found using the Euler backwards approximation and an appropriate sampling time. The time-discrete equation is given by:

$$u(k) = K_p\, e(k) + i(k-1) + K_i\, Ts\, e(k) \tag{3.6}$$

where $i(k-1)$ is the integral action computed at time $k=1$.

3.2.2 The Smith predictor control algorithm

The Smith predictor control algorithm is implemented using the following procedure.

Step 1: Initialisation

(a) Set up the model parameters of the process and substitute them in $G_{pm}(s)$.

(b) Implement the controller, sensor/actuator and network nodes from the Truetime simulator. Follow the Smith predictor configuration in Figure 3.1.

(c) Select the network configuration and the percentage of dropouts.

(d) Set an interfering node to increase the time delays.

Step 2: Off-line calculation.

(a) Tune the PI gains using the AMIGO tuning rules.

(b) Implement the PI control law using (3.6).

Step 3: On-line calculation

(a) Compute the control signal and apply to the process.

Step 4: Fine-tuning

(a) Check the closed-loop performance and fine-tune Ts *and the PID gains if necessary.*

3.2.3 Numerical example 1

Consider the first-order process described by:

$$G_p(s) = \frac{10\, e^{-5\, s}}{15\, s + 1} \tag{3.7}$$

The robustness of the control system subject to changes in the process model has been studied by varying the three parameters of the process model: time delay, the time

constant and gain. Then, an analysis of stability using the gain margin, phase margin and maximum sensitivity of the system is performed for each parameter variation.

To examine the sensitivity of the closed-loop to variation in the parameters, the sensitivity transfer function in (3.2) is implemented. To simulate the effect of the network delays, Gaussian distributed random delays are implemented with a variance $\sigma^2 = 0.1$. The mean is selected $\mu = 1$ for τ_{sc} and $\mu = 0.5$ for τ_{ca}. Figure 3.2 displays the time delays from controller to actuator and from sensor to actuator.

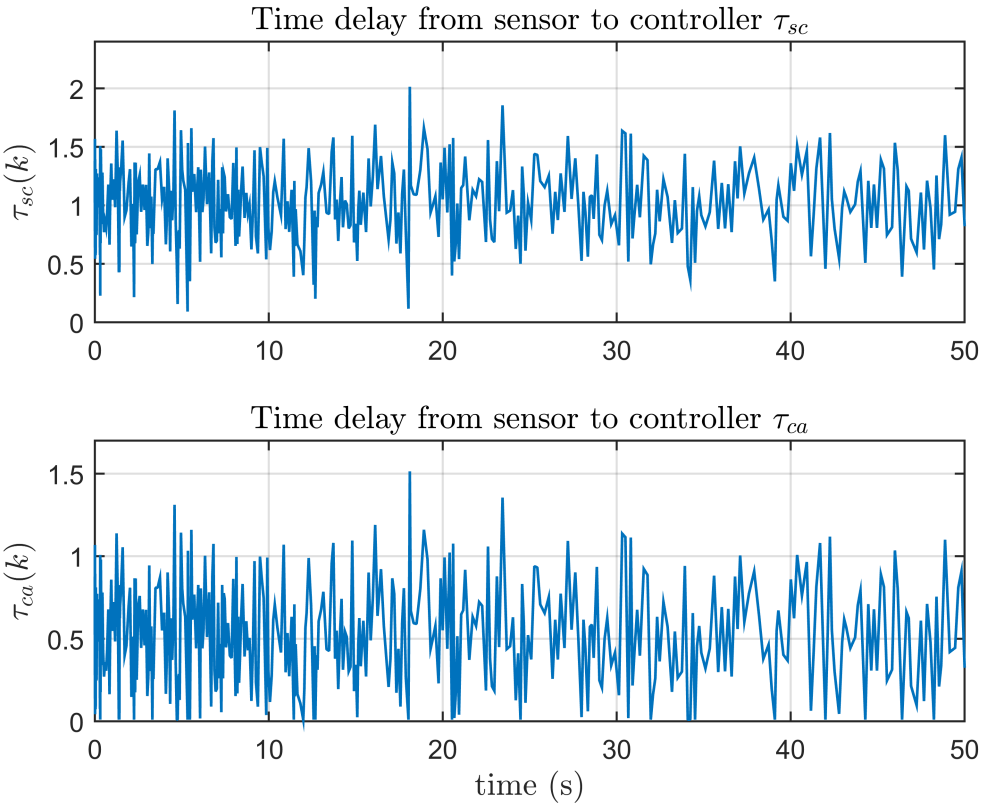

FIGURE 3.2: Network time delays

From the simulation, the maximum time delay from controller to actuator is $\tau_{ca,max} = 1.51\,s$. The minimum value is $\tau_{ca,min} = 0.01\,s$. The maximum time delay from sensor to controller is $\tau_{sc,max} = 2.01\,s$. The minimum value is $\tau_{ca,min} = 0.09\,s$.

The network delays in Figure 3.1 are implemented using Padé approximations of the maximum and minimum values found.

Based on this parameters, several experiments where the parameters of the prediction model: time delay τ_{pm}, the time constant τ_{1pm} and gain K_{pm} are varied up to 50%. The results of this experiments are shown in the next subsections.

3.2.3.1 Effect of time delay variation

A study of the variation of the prediction value τ_{pm} has been made for an interval of $2.5\,s < \tau_{pm} < 7.5\,s$ taking 16 points between them. The results of the Ms are shown in Figure 3.3. The solid line represents the results using the minimum values of the network time delays τ_{ca} and τ_{sc}. The dashed line shows the results using the maximum values.

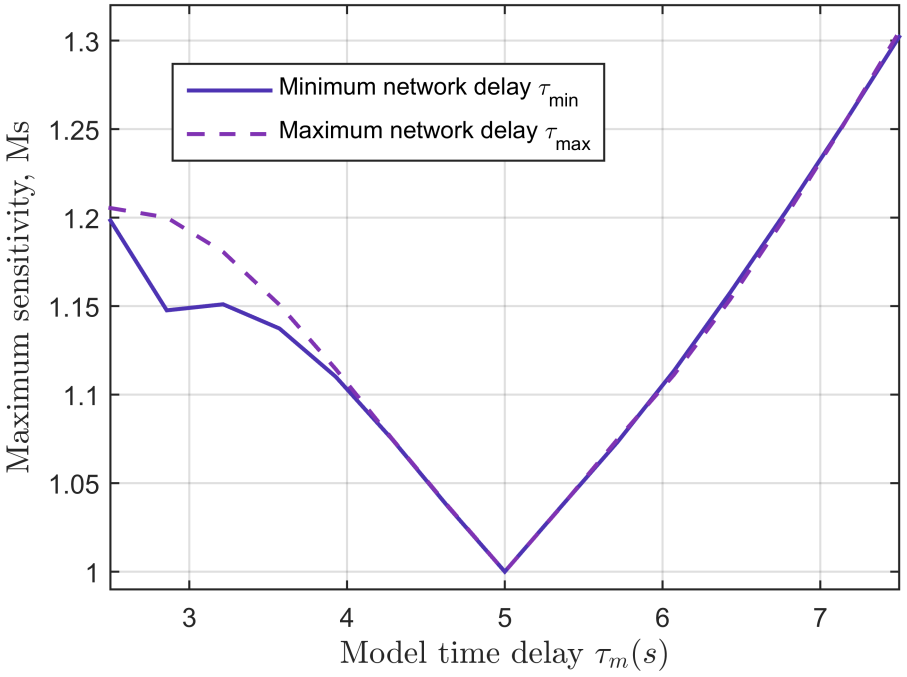

FIGURE 3.3: Maximum sensitivity for time delay variation

It can be seen that higher and lower values of the model time delay increase the maximum sensitivity. As expected, the maximum sensitivity decreases when the mismatch between the nominal process delay and the process delay prediction is small, this is near to the real value $\tau = 5\,s$. However, the maximum value is below $Ms = 1.4$ which guarantees the robustness of the system.

The phase and gain margins are calculated as a function of the network delay and the results are shown in Table 3.1. The table presents two cases depending which network delay is employed: Columns two and three display the margins using the maximum network delay and columns four and five depict the results using the minimum network delay.

TABLE 3.1: Measurement of robustness for delay variation

τ_{pm} (s)	τ_{max} GM (dB)	τ_{max} $PM(^\circ)$	τ_{min} GM (dB)	τ_{min} $PM(^\circ)$
2.5	17.07	66.53	24.95	65.57
3.93	22.66	75.32	22.63	74.77
5.00	Inf.	81.18	Inf	81.18
6.07	22.84	85.90	22.41	86.84
7.50	16.59	88.50	15.31	91.88

The results demonstrate the robustness of the method. Further simulations using minimum and maximum network delays show that the phase margin decreases rapidly when the time delay is greater than 7.5 s. Therefore, the proposed method is robust for a limited interval of time delay variations.

3.2.3.2 Effect of time constant variation

In this section, the sensitivity of the system is studied for variations in the time constant of the prediction model, τ_{1pm}. The results for the variation of Ms versus the time constant are shown in the Figure 3.4. As a result of the increase and decrease of the value of the time constant prediction, the maximum sensitivity increases and decreases proportionally. In the case of a match between the nominal process time constant and the process time constant prediction, i.e., near to the real value $\tau = 15\,s$ the $Ms = 1$. The values of Ms are smaller than the previous case. Therefore, the system is more robust to the variation of the time constant.

The phase and gain margins are calculated in function of the network delay and the results are shown in Table 3.2.

According to Table 3.2, good stability margins are achieved with both cases of maximum and minimum network time delay. In particular, the gain margins are infinity which means the system has a good relative stability.

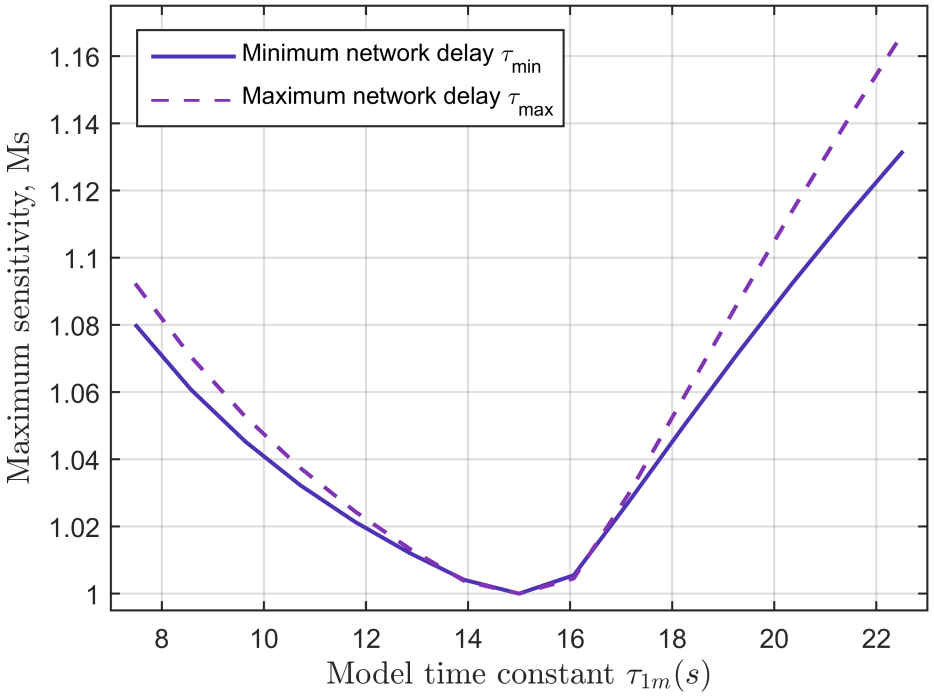

FIGURE 3.4: Maximum sensitivity for time constant variation

TABLE 3.2: Time constant variation

τ_{1m} (s)	τ_{max} GM (dB)	τ_{max} $PM(^\circ)$	τ_{min} GM (dB)	τ_{min} $PM(^\circ)$
7.50	Inf.	102.31	Inf.	96.44
11.79	Inf.	89.88	Inf.	87.68
15.00	Inf.	81.18	Inf	81.18
18.21	Inf.	74.56	Inf.	76.11
22.50	Inf.	68.15	Inf.	71.14

Further simulations show that for a value greater than 31 s, the phase margin decreases rapidly, for both minimum and maximum network delay. However, the gain margin is still good with values greater than 10 dB. Thus, the method is robust for a limited interval of time constant variations.

3.2.3.3 Effect of gain variation

A study of the variation of the prediction value K_{pm} has been performed for a interval of $5 < K_{pm} < 15$ taking 16 points between them. The results for the variation of Ms versus gain are shown in Figure 3.5. It can be seen that lower values of the gain increase

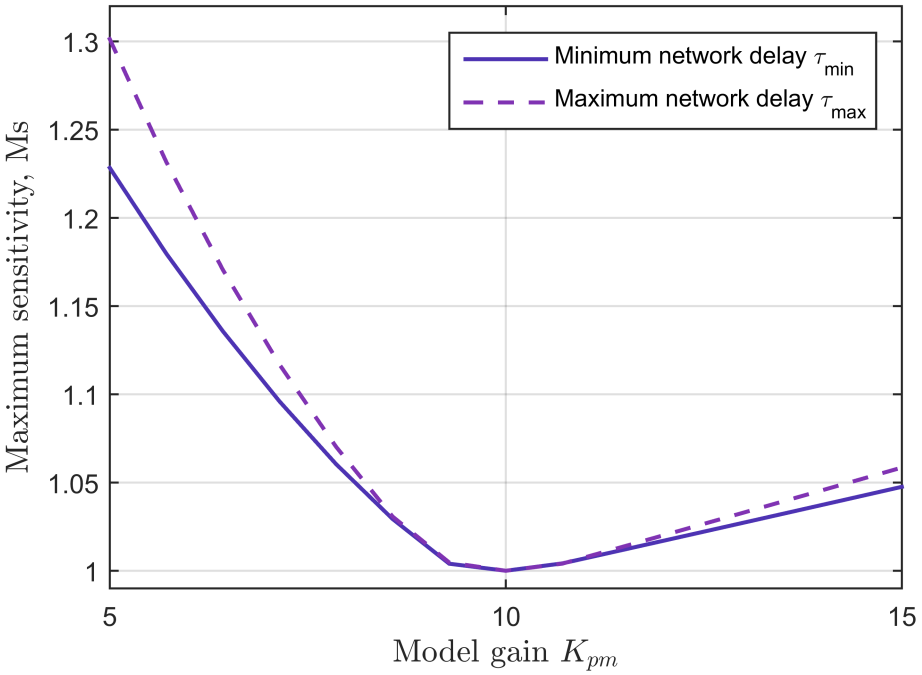

FIGURE 3.5: Maximum sensitivity for gain variation

the maximum sensitivity rapidly. In contrast with the previous studies, for higher values of the parameter, Ms presents a small increment. Indeed, the maximum sensitivity occurs for $K_{pm} = 5$ i.e., the 50 % less of the nominal value $K_p = 15$. Therefore, increasing the gain has the minor effect on the sensitivity of the system.

The phase and gain margins are calculated in function of the network time delay, and the results are shown in Table 3.3. The system presented a good relative stability for

TABLE 3.3: Gain variation

K_{pm} (s)	τ_{max} GM (dB)	τ_{max} $PM(°)$	τ_{min} GM (dB)	τ_{min} $PM(°)$
5.00	38.32	60.78	42.99	65.17
7.86	Inf.	72.59	Inf.	74.38
10.00	Inf.	81.18	Inf	81.18
11.43	Inf.	86.80	Inf.	85.68
15.00	Inf.	100.29	Inf.	96.87

variations in the prediction value of the gain. Even for greater values, for instance, $K_{pm} = 30$ this characteristic remains.

3.2.4 Numerical example 2

Consider the following first order plus dead time (FOPDT) process. A system with faster response with respect to the numerical example one has been selected to test the effectiveness of the control system using the TrueTime simulator.

$$G_p(s) = \frac{1}{2\,s+1}\,e^{-2\,s} \tag{3.8}$$

A NCS has been created using the TrueTime simulator (Cervin et al., 2003). Truetime is a simulation toolbox based on MATLAB/Simulink which allows to test the real-time control system design and provide a detailed analysis of the effects of the communication network. Simulink blocks are used to model the NCS including simple models of communication networks, sensors, controllers and actuators. The network block is event driven and executes when messages enter or leave the network. The messages contain the measurement or control signals, the length of the message and optional real-time attributes such as a priority or a deadline. In the network block, it is possible to specify the network parameters such as transmission rate, the medium access control protocol, the number of network nodes, the probability of loss, among others. More details are explained in Appendix A.

Three blocks of the Truetime library are used to implement the NCS. The blocks are described using S-Functions of Matlab. First, a network node has been configured for Ethernet protocol with a transmission rate of 80000 bit/s and a frame size of 80 bits. The dropouts have been set to zero. The second node is configured as the discrete controller. The third node is the sensor/actuator that will send and receive the process output and the controller input. The block diagram that illustrates the Truetime nodes configuration is depicted in Figure 3.6. These nodes are connected to common Simulink blocks representing the plant and the process model following the configuration in Figure 3.1.

The equivalent time-discrete controller has been found using the Euler backwards approximation and a sampling time $Ts = 0.01\,s$. The controller execution time is set 2 ms. The PI parameters are found using the AMIGO tuning rules as $K_c = 0.65$ and $T_i = 2.18\,s$. For the Smith predictor, the model parameters are chosen equal to those of the process. Finally, to test the system subject to disturbances, a step disturbance signal which amplitude is 0.4, is introduced at $t = 35\,s$. Figure 3.7 shows the closed-loop response without delays. The solid line depicts that the output can reach the desired

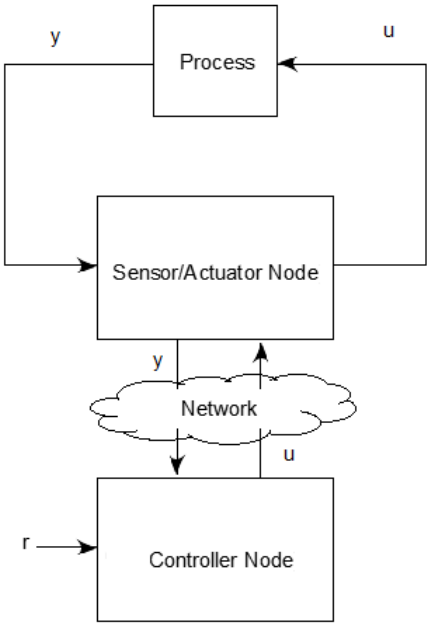

FIGURE 3.6: The TrueTime nodes

value in a few seconds. Moreover, it responds quickly to the disturbance demonstrating a good prediction of the output. The performance of the design has been measured with the Integral of Time-weighted Absolute Error (ITAE) criterion. The values are $J_r = 20.52$ for the servo control and $J_d = 82.98$ for the regulatory control.

Next, the delays and dropouts have been set in TrueTime simulator. To increase the effect of the time delay, an interfering node sending disturbing traffic over the network is implemented with an occupation of the 47% of the network bandwidth. The dropouts have been set up through a percentage of dropouts $P_{loss} = 30\%$. Figure 3.8 displays the time instants of dropouts from sensor to controller $d_{psc}(k)$ (top) and from controller to actuator $d_{pca}(k)$ (bottom). A variable $d_p(k)$ ε $[0, 1]$ indicates if the packet containing the feedback signal $y(k)$ is received $(d_{psc}(k) = 0)$ or if it is dropped $(d_{psc}(k) = 1)$. Similarly, dropouts from the controller to the actuator are represented as $(d_{pca}(k) = 1)$ and $(d_{pca}(k) = 0)$ if there is no dropouts. From the simulation, the mean values for time delay from controller to actuator and from sensor to controller are $\tau_{ca} = 0.5\,s$ $\tau_{sc} = 1\,s$, respectively.

The dashed lines in Figure 3.7 represent the scenario with delays. Although the process has a larger rise time, no overshoot and zero steady error are observed. On the other hand, it can be seen that the controller returned the system smoothly to the setpoint

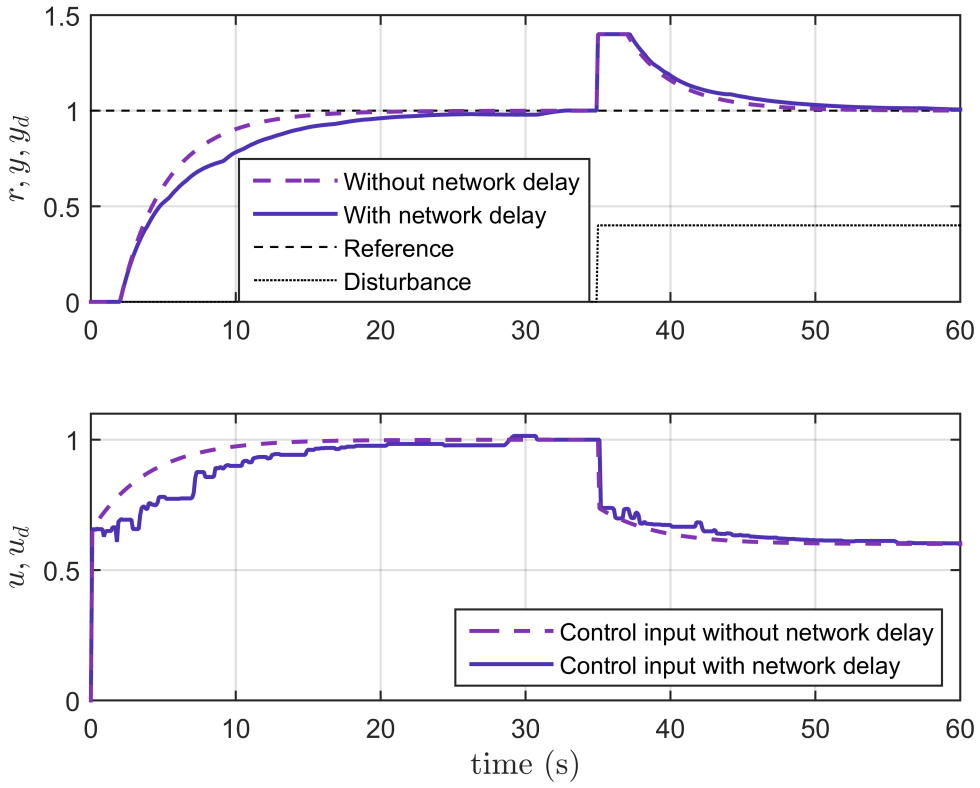

FIGURE 3.7: System outputs for Smith predictor

after the application of the disturbance. ITAE criterion returned a value of $J_r = 41.4$ for the servo control and $J_d = 104.66$ for the regulatory control. This value is bigger than the previous scenario ($J_r = 20.52$ and $J_d = 82.98$) which demonstrates the adverse effect of the delay in the NCS. More performance and robustness measurements are displayed in Table 3.6.

3.3 Adaptive IMC for NCS

In this section, an adaptive IMC algorithm is created to address the challenge of compensation of time delay and dropouts in NCS. A recursive least squares estimator is implemented to estimate the discrete process model online and adapt it during every sampling period. A filter is used for the design of the IMC controller.

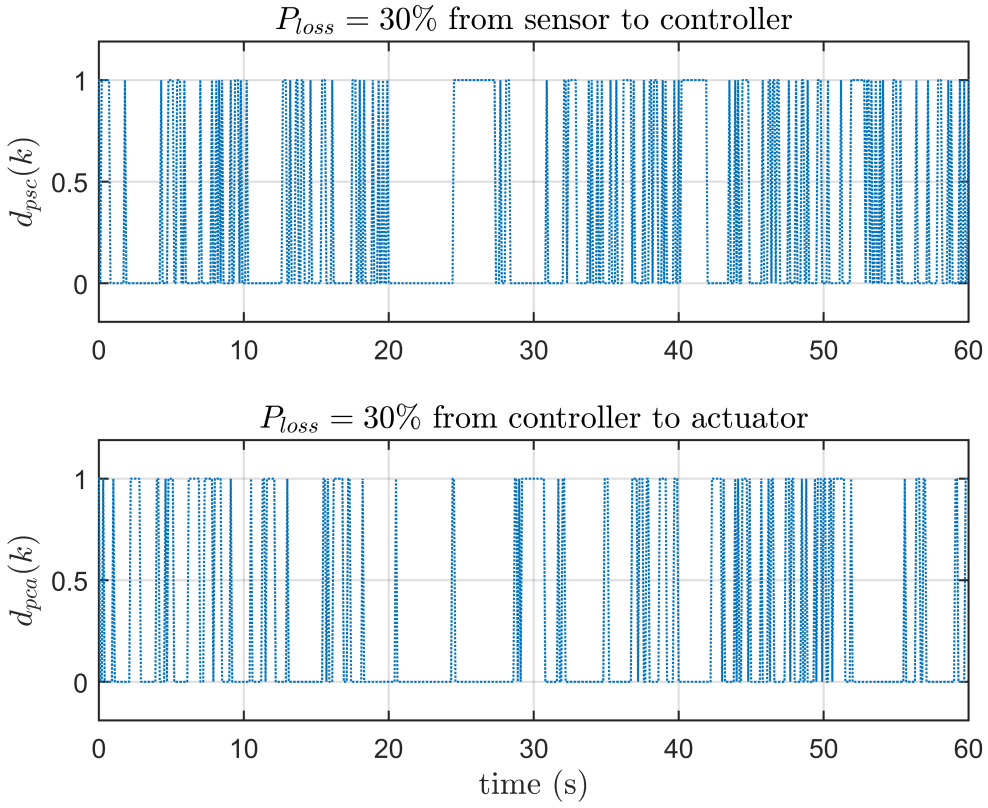

FIGURE 3.8: Time instants of data dropouts $P_{loss} = 30\%$

3.3.1 Introduction

Model-based controllers are proposed to compensate for the network constraints in the literature. IMC is a well known model-based controller and the implementation of an adaptive algorithm in the IMC structure can effectively control the process under delays and disturbances. The adaptive internal model control IMC is implemented using a Least Squares Estimation (LSE) to identity the parameters of the model and adapt to the network constraints.

In Figure 3.9 the network delays have been added to the typical IMC structure. The process transfer function is $G(s)$, $G_m(s)$ is the internal model of the process and $C(s)$ is the function of the controller. The model can be represented as $G_m(s) = G_{\mathrm{m}+}(s)\, G_{\mathrm{m}-}(s)$ where $G_{\mathrm{m}-}$ is the invertible part of the process model and $G_{\mathrm{m}+}$ is the non-invertible part. The networks delay τ_{sc} *and* τ_{ca} are chosen to be random, time-varying and uncertain. Their maximum and minimum values are selected from the experiments using

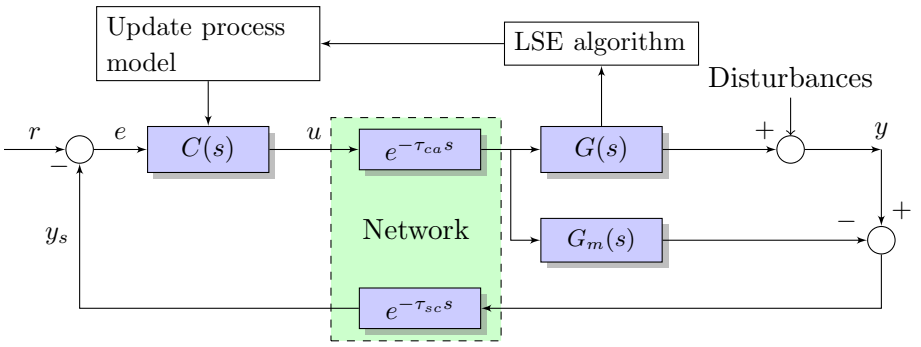

FIGURE 3.9: Networked IMC structure

the TrueTime simulator.

The scheme exhibits the closed-loop system including the LSE algorithm block which will compute online the model parameters and update them in the transfers function of the controller and the internal model of the process.

The closed-loop transfer function for this system is:

$$\frac{y(s)}{r(s)} = \frac{C(s)e^{-\tau_{ca}s}G(s)}{1 + C(s)e^{-\tau_{ca}s}[G(s) - G_m(s)]e^{-\tau_{sc}s}} \tag{3.9}$$

The design of the controller is performed by the cancellation of the invertible part of the process model and the addition of a filter. Thus, the transfer function $C(s)$ is:

$$C(s) = G_{\mathrm{m}-}(s)F(s) \tag{3.10}$$

The typical transfer function of the filter is:

$$F(s) = 1/(\lambda\, s + 1)^n \tag{3.11}$$

where λ is the time constant of the filter and it is used to tune the closed-loop response. The value of n is chosen to obtain an appropriate transfer function.

3.3.2 The design of the adaptive IMC

According to (3.9), if $G(s) = G_m(s)$ the closed-loop transfer function becomes:

$$y(s)/r(s) = F(s)e^{-\tau_{ca}s}G_{\mathrm{m}+}(s) \tag{3.12}$$

In this case, the major challenge is to design the filter to find the robust and less sensitive response. However, in a realistic case with a non-perfect model, $G(s) \neq G_m(s)$, the closed-loop transfer function becomes:

$$\frac{y(s)}{r(s)} = \frac{G_{m-}^{-1}(s)F(s)e^{-\tau_{ca}s}G(s)}{1 + G_{m-}^{-1}(s)\ F(s)e^{-\tau_{ca}s}[G(s) - G_{\mathrm{m}}(s)]e^{-\tau_{sc}s}} \tag{3.13}$$

Equation (3.13) shows that the stability of the system decreases due to variations of process parameters. Thus, the IMC structure is improved with the combination of an adaptive algorithm. The essential part of the adaptive system is the online identification of the process parameters, which allows to update the process model and the controller every sampling time.

The least squares criterion is used to find a simple model, that describes the physical process. Since the system is constantly changing and because the interest is to conduct a real-time identification, a method of estimating the parameters recursively is needed. Therefore, the least squares criterion is combined with a regression model structure to identify the linear systems.

A recursive least squares estimation for discrete systems is studied here. A second-order model is given by the equation:

$$y(k) = a_0\, u(k) + a_1\, u(k-1) - b_1\, y(k-1) - b_2\, y(k-2) \tag{3.14}$$

where a_0, a_1, b_0, b_1 are the model coefficients, $y(k)$ is the present value, $y(k-n)$ and $u(k-n)$ are the output and input values at the $k-n$ sampling instant. These represent the regression values, denote by ϕ. The unknown parameters are found by computing the vector of parameters $\theta = [a_0, a_1, b_0, b_1]$.

The implementation minimises the sum of the squares of the differences between observed and calculated values and it is weighted by multiplying it by a constant. The recursive algorithm computes the error every sampling time and updates the vector of parameters. This identification algorithm is implemented in a S-function in Matlab as explained in Appendix B.

Finally, the estimated parameters are sent to a discrete filter block that represents the model process in the IMC structure. The parameters are also sent to the controller to compute the output signal.

3.3.3 The adaptive IMC algorithm

The adaptive IMC algorithm is implemented using the following procedure.

Step 1: Initialisation

(a) Set up the order and model parameters of the process.

(b) Set the initial value of θ, Ts and forgetting factor for the LSE.

(c) Construct the filter and select λ.

(d) Implement the controller, sensor/actuator and network nodes from the Truetime simulator. Follow the IMC configuration in Figure 3.9.

(e) Follow steps c to d from the control algorithm in 3.2.2.

Step 2: Off-line calculation.

(a) Implement the controller law using the model parameters and the filter.

Step 3: On-line calculation

(a) LSE computes and updates the process model and the controller.

Step 4: Fine-tuning

(a) Check the closed-loop performance and fine-tune Ts and λ if necessary.

3.3.4 Numerical example 1

The following first-order process has been selected to test the effectiveness of the control system:

$$G_p(s) = \frac{1\, e^{-0.02\, s}}{0.01\, s + 1} \tag{3.15}$$

The model of the controlled process is identified online by using the recursive least squares algorithm. These parameters are represented in a time-discrete filter defined as follows:

$$G_{pm}(z) = \frac{a_0 z^{-1} + a_1 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} \tag{3.16}$$

A second-order model is selected to improve the accuracy of the approximation. The model is updated every sampling time. The sampling time is investigated and optimised for the open-loop system and the lowest Integral of Absolute Error (IAE) criterion is found using $Ts = 0.04\,s$. A forgetting factor of 0.998 is chosen to make the system less sensitive to the noise. A discrete filter is implemented as follows:

$$F(z) = \frac{(1-\lambda)^2 z^{-2}}{1 - 2\lambda z^{-1} + \lambda^2 z^{-2}} \tag{3.17}$$

The filter time constant is selected through several simulations and the optimal value of λ is found to be 0.39.

The TrueTime network has been configured for Ethernet protocol as explained in Section 3.2. The minimum frame size has been selected as 40 bits, to increase the delays. The percentage of dropouts is $P_{loss} = 30\%$. The input is chosen to be a square signal with frequency 0.2 Hz. A step disturbance of magnitude 0.4 is introduced at $t = 9\,s$. The results of the experiments are shown a continuation. The evolution of the model

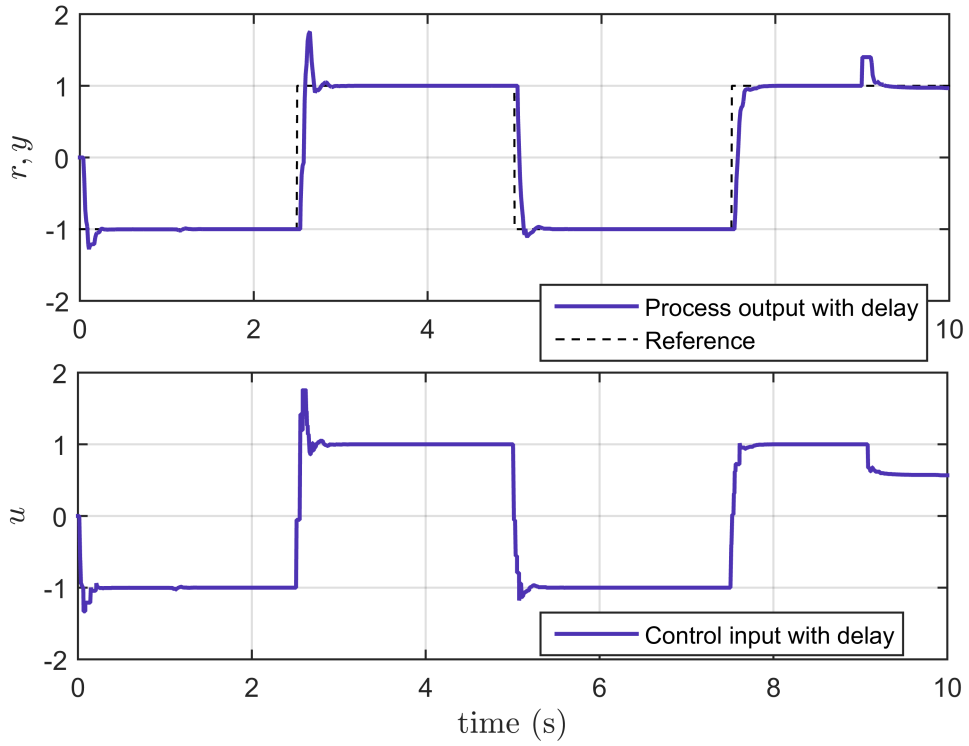

FIGURE 3.10: System outputs for loss probability = 0.3, interference 47 %

parameters is depicted in Figure 3.11. The final value at which the parameters converge at the end of the simulation is indicated in Table 3.4. It can be seen that the second parameter of the denominator b_2 tends to zero, indicating that a first order model can be used for the design.

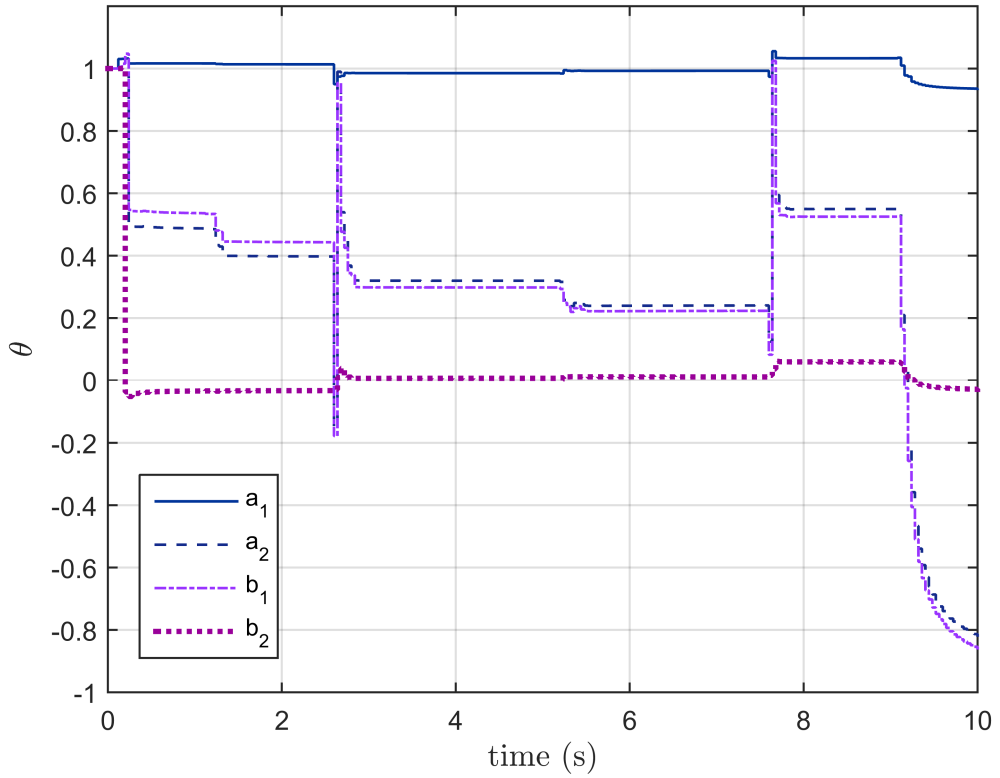

FIGURE 3.11: Evolution of the model parameters θ

The effect of this delay is significant in Figure 3.10. However, the simulation shows the controller is able to adapt the system response and compensate the random and variable delays and dropouts in the network.

It can be seen that the response is fast enough and the new adaptive IMC provides a good setpoint tracking. The responses to the positive and negative changes are almost symmetrical as expected for the linear plant. For the case of the regulatory control, the system presents a good disturbance rejection. Further experiments show that the conditions to guarantee the stability of the systems are a percentage of dropouts of 60% and interference bandwidth 60 %. The results of the optimal parameters of the model, the IAE criteria and network delays for this scenario and the previous one are listed

in Table 3.4. The performance criterion shows that the effect of the new settings is very aggressive. The maximum network delay is 1.86 s. Increasing the values of delays and the percentage of dropouts cause the delay of the controller output and the output system to mismatch the setpoint. At the end of the simulation, the process does not recover from the disturbance. The IAE cost is 5.82, approximately seven times bigger than the value found in the previous network conditions. Thus, the adaptive IMC has a

TABLE 3.4: Optimal parameters, performance and network delays

Values	Loss probability = 0.3, Interference BW 47.5 %	Loss probability = 0.6, Interference BW 60 %
θ	$[0.94, -0.82, -0.86, 0.03]$	$[1, 0.00087, 0.0016, 4.03x10^{-6}]$
IAE	0.79	5.82
Maximum delay	0.24 s	1.86 s

good performance and robustness against the variant and random delay for conservative values of dropouts and time delays. The results conclude that the systems is stable for loss of probability < 0.6 and interference bandwidth < 60 %.

3.3.4.1 Numerical example 2

The process example used in (3.8) is considered here. The forgetting factor is set as 0.98, $Ts = 0.5\ s$ and a desired closed-loop time constant as $\lambda = 0.96$. The initial point is $\theta = [1.011, -0.77, -0.773, 0.0001]$. The response of the system is depicted in Figure 3.12. There is a large rising time as a result of the large value of λ. The ITAE values are $J_r = 48.6$ and $J_d = 131.4$. The model parameters are found as $\theta = [1,\ 4x10^{-4},\ -21x10^{-4},\ -3.2x10^{-5}]$. In addition, network delays and dropouts have been configured in the same way as described in previous section and the results for the adaptive IMC controller are shown in Figure 3.12. The model parameters are found as $\theta = [1,\ 5x10^{-4},\ 2x10^{-3},\ -2.3x10^{-5}]$. The ITAE criterion returned the cost of $J_r = 70.6$ and $J_d = 162.8$.

According to the simulation, the proposed method can tolerate the percentage of dropouts without becoming unstable. However, the response is very slow and it has a poor recovery after the application of the disturbance. The sluggish in the output is expected since the time constant value of the filter is high. Although some simulations are performed using smaller values the system became unstable, therefore, the slow response is preferred.

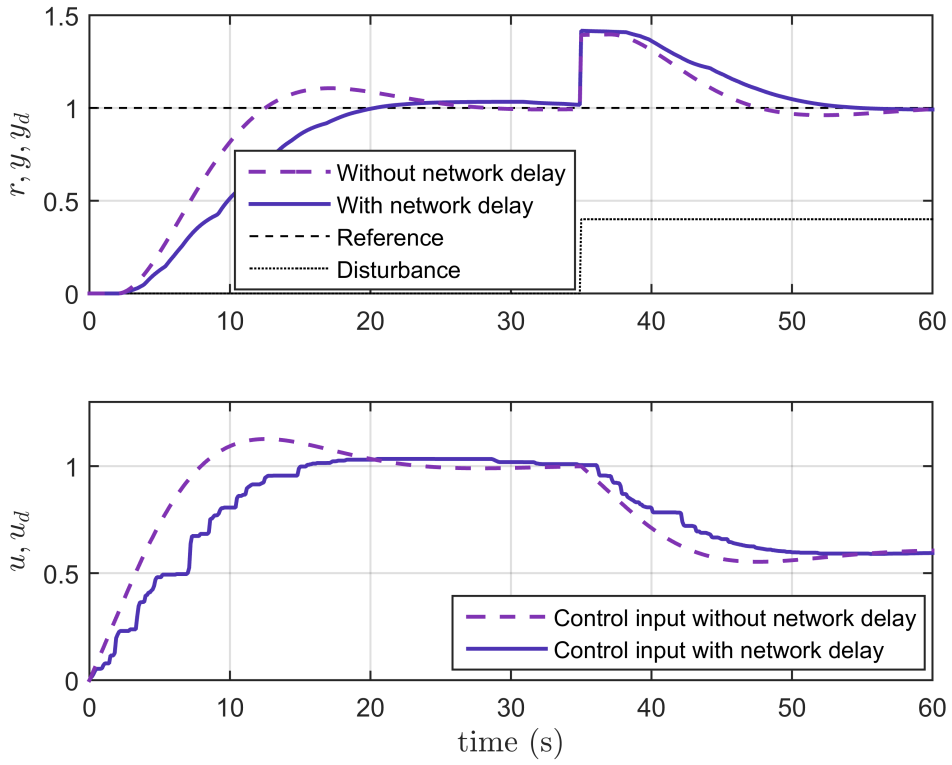

FIGURE 3.12: System outputs for adaptive IMC

In comparison with the Smith predictor, the performance of the adaptive IMC has a longer rising time and slower recovery from the disturbance. The adaptive IMC response is slower due to the computation time required by the online algorithm.

3.4 A design of robust PID controller using gain/phase margin

The time delay in the control system decreases the phase and gain margin of the closed-loop system. In this section, a new robust PID controller is presented to compensate the time delay in NCS. The delay in the NCS includes the computation delay in each component, the waiting delay and the transmission delays τ_{sc} and τ_{ca}. The controller and actuator are event-driven, which leads to the time-varying delay in NCS. Therefore, the controller has been designed to compensate for a system subject to an uncertain delay and follows specifications of gain and phase margins.

The characteristic equation of the system is modified by adding a gain-phase margin tester function. A set of stability equations is defined to find the desired gain margin and phase margin boundaries that are represented in a parameter plane. Then, the PID parameters that guarantee the stability and the desired margins are obtained from the admissible region in that plane.

3.4.1 Introduction

The robustness analysis of control systems using the phase and gain margins is a very well known design approach. Frequency domain analysis with Bode, Nichols and Nyquist are familiar and straightforward techniques to study these margins. However, they are not suitable for analysis where various parameters have to be adjusted.

The parameter plane method gives a locus representation using the controller parameters where a final enclosed area by the stability boundaries can be found. Moreover, it can be extended to find information about the gain and phase margin of the system.

A robust PID has been derived in Huang and Wang (2001) for non-minimum phase process with uncertain time delay. Pansari, Timande and Chandrakar (2012) exploited this approach for NCS. Here, the parameter plane method is implemented to find the set of PID parameters that achieve the desired compromise in NCS. A study of the uncertain delay and the compromise between performance and robustness is investigated.

The NCS control closed-loop system is shown in the Figure 3.13 where, $G_p(s)$ is the process transfer function, $C(s)$ is the controller and the transmission delays are τ_{sc} and τ_{ca}.

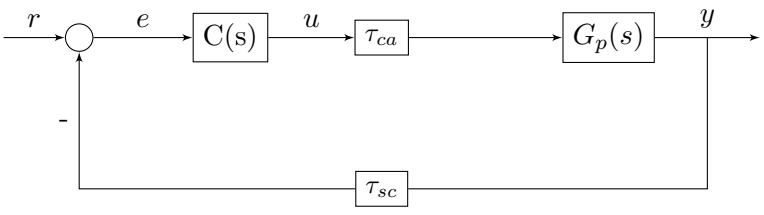

FIGURE 3.13: Closed-loop of the NCS

The controller has the PID parallel structure:

$$C(s) = K_p + \frac{K_i}{s} + K_d s \tag{3.18}$$

where $K_p,\ K_i,\ K_d$ are the proportional, integral and derivative gains, respectively. The open-loop transfer function is:

$$G(s) = C(s)\,G_p(s) = \frac{N(s)}{D(s)}e^{-Ts} \tag{3.19}$$

where $N(s),\ D(s)$ stands for the numerator and denominator, both polynomials function of s. T is the time delay of the NCS and includes the computation delay in each component, the waiting delay, the transmission delays τ_{sc} and τ_{ca} and the dead time of the process. By letting $s = j\omega$, yields:

$$G(j\omega) = \frac{N(j\omega)}{D(j\omega)}e^{-Tj\omega} \tag{3.20}$$

However, the real and imaginary parts can be found as follows:

$$G(j\omega) = Re[G(j\omega)] + jIm[G(j\omega)] \tag{3.21}$$

In terms of magnitude $|G(j\omega)|$ and phase ϕ, this is equivalent to:

$$G(j\omega) = |G(j\omega)|\,e^{j\phi} \tag{3.22}$$

where

$$|G(j\omega)| = \sqrt{Re[G(j\omega)]^2 + Im[G(j\omega)]^2} \tag{3.23}$$

$$\phi = \arctan\left\{\frac{Im[G(j\omega)]}{Re[G(j\omega)]}\right\} \tag{3.24}$$

Substituting (3.22) in (3.20) leads to:

$$D(j\omega) - \frac{1}{|G(j\omega)|\,e^{j\phi}}N(j\omega) = 0 \tag{3.25}$$

Define: $A = \frac{1}{|G(j\omega)|}$ and $\theta = \phi + T\omega + 180°$. If $A = 1$ means that θ is the phase margin of the system and $\theta = 0$ results in A becoming the gain margin. Therefore, the gain and phase margin can be determined using the characteristic equation of the system with a gain-phase margin tester. The representation is depicted in Figure 3.14.

The characteristic equation for the system of Figure 3.14 is given by:

$$F(j\omega) = D(j\omega) + Ae^{-j\theta}N(j\omega) = 0 \tag{3.26}$$

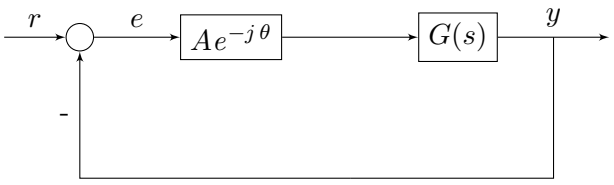

FIGURE 3.14: Control system with gain-phase margin tester

According to Figure 3.13 the open-loop transfer function is $G(j\omega)G_c(j\omega)$. By noting that $Ae^{-j\theta} = A\cos\theta - jA\sin\theta = X - jY$, the characteristic equation can be written as:

$$\begin{aligned} F(j\omega) &= D(j\omega) + (X - jY)N(j\omega) = 0 \\ F(j\omega) &= F_r(X, Y, \omega) + jF_i(X, Y, \omega) = 0 \end{aligned} \tag{3.27}$$

where

$$\begin{aligned} F_r(X, Y, \omega) &= 0 \\ F_i(X, Y, \omega) &= 0 \end{aligned} \tag{3.28}$$

The previous equations are called the stability equations. If X and Y are considered as parameters, the following can be estimated:

$$\begin{aligned} F_r(X, Y, \omega) &= XB_1 + YC_1 + D_1 = 0 \\ F_i(X, Y, \omega) &= XB_2 + YC_2 + D_2 = 0 \end{aligned} \tag{3.29}$$

where B_1, C_1, D_1, B_2, C_2 and D_2 are functions of ω. By solving (3.29) for X and Y, it can be found that:

$$\begin{aligned} X &= \frac{C_1D_2 - C_2D_1}{B_1C_2 - B_2C_1} \\ Y &= \frac{D_1B_2 - D_2B_1}{B_1C_2 - B_2C_1} \end{aligned} \tag{3.30}$$

By varying ω, a locus representing the common roots of (3.30) can be plotted in a $X - Y$ plane. The stability boundaries of the system can be found as well by defining the Jacobian as:

$$J = B_1C_2 - B_2C_1 \tag{3.31}$$

When $J < 0$, the right of the state boundary, facing against the increasing direction of ω gives the stable parameter area. Moreover, letting A be constant and $\theta = 0$, the locus

is a boundary of constant gain margin. Similarly, if $A = 1$ and $\theta = 0$ is constant, the locus represents the boundary of constant phase margin. The two stability equations are a function of the set of process and controller parameters. Since the process model is assumed to be known, by letting K_d constant, a $K_p - K_i$ plane can be plotted. The resulting area contains the feasible controller parameters that guarantee stability and the desired margins of the system.

3.4.2 The robust PID control algorithm

The robust PID control algorithm is implemented using the following procedure.

Step 1: Initialisation

(a) Set up the model parameters of the process.

(b) Implement the controller, sensor/actuator and network nodes from the Truetime simulator. Follow the NCS configuration in Figure 3.13.

(c) Follow steps c to d from the control algorithm in 3.2.2.

(d) Find from the simulation the controller computation time and the mean values of the network delays τ_{sc} and τ_{ca}.

Step 2: Off-line calculation.

(a) Compute the stability equations and construct the parameter plane.

(b) Select the PID gains that guarantee the desired stability margins and implement the PID control law.

Step 3: On-line calculation

(a) Compute the control input and apply to the process.

Step 4: Fine-tuning

(a) Check the closed-loop performance and robustness and fine-tune the gains if necessary.

3.4.3 Numerical example 1

Consider the process in (3.8):

$$G_p(s) = \frac{1\, e^{-T\, s}}{2\, s + 1} \tag{3.32}$$

where $T = 2+\tau$, τ stands for the uncertain delay of the network and 2 s is the dead time of the process. The network delay is consider as $\tau = \tau_{sc} + \tau_c + \tau_{ca}$. τ_c is the controller computation time that has been set as 2 ms. The mean values of the transmission delays are found from the simulation in Truetime as $\tau_{ca} = 0.5\, s$ and $\tau_{sc} = 1\, s$. The computation time of the sensor and actuator are almost negligible compared to the transmission delays and therefore omitted here. The waiting delay is also excluded since it is usually compensated by the appropriate network protocols.

Substituting the previous equation and (3.18) in (3.27) the resulting characteristic equation is:

$$F(s) = 1 + Ae^{-j\theta} \left(K_p + \frac{K_i}{s} + K_d s\right) \left(\frac{1\, e^{-Ts}}{2\, s + 1}\right) = 0 \tag{3.33}$$

Letting $s = j\omega$:

$$F(j\omega) = 1 + Ae^{-j\theta} \left(K_p + \frac{K_i}{j\omega} + K_d j\omega\right) \left(\frac{1\, e^{-Lj\omega}}{2\, j\omega + 1}\right) = 0 \tag{3.34}$$

Note that $Ae^{-j\theta} = A[cos(\theta) - jsin(\theta)]$. By working with the trigonometric identities, two stability equations are found:

$$\begin{aligned} F_r(j\omega) &= K_p B_1 + K_i C_1 + D_1 = 0 \\ F_i(j\omega) &= K_p B_2 + K_i C_2 + D_2 = 0 \end{aligned} \tag{3.35}$$

where:

$$B_1 = -Acos(\theta_1)\omega \tag{3.36}$$

$$C_1 = Asin(\theta_1) \tag{3.37}$$

$$D_1 = -w - Asin(\theta_1) K_d \omega^2 \tag{3.38}$$

$$B_2 = Asin(\theta_1)\omega \tag{3.39}$$

$$C_2 = Acos(\theta_1) \tag{3.40}$$

$$D_2 = -2\,\omega^2 - Acos(\theta_1)K_d\omega^2 \tag{3.41}$$

By resorting 3.30 and letting K_d to be constant the controller parameters are defined by:

$$\begin{aligned} K_p &= \frac{C_1D_2 - C_2D_1}{B_1C_2 - B_2C_1} \\ K_i &= \frac{D_1B_2 - D_2B_1}{B_1C_2 - B_2C_1} \end{aligned} \tag{3.42}$$

Using (3.42) and performing the following procedure the locus is plotted and shown in Figure 3.15. Initially, T is set 2 s and K_d is fixed as 0.1. The first boundary $J < 0$ is found by setting $A = 1$ and $\theta = 0°$. The stability region has been marked in the figure. The boundaries for constant margins are also plotted. As depicted in the figure, the point $K_p = 0.59, K_i = 0.30$ is selected to guarantee a phase margin at least of 30° and a gain margin at least of 6 dB. The closed-loop system response to a step input can be found in Figure 3.16. It can be seen that the system has a good performance and good rejection to the disturbance. The margins are $PM = 62.5°$ and $GM = 12\ dB$. Therefore, the relative stability of the system fulfils the specified criteria.

Furthermore, the value of T have been changed to $T = 2.5\ s$, $T = 3\ s$ and $T = 3.5\ s$. This variation will cover the mean values of the network delay. The resultant intersection area is displayed in Figure 3.17. The shaded region represents the admissible parameters for $K_p - K_i$ that will guarantee at least a phase margin of 30° and a gain margin of 6 dB. Based on this plot, a point of $K_p = 0.45$ and $K_i = 0.24$ is selected for the design of the robust PID controller. The closed-loop performance and robustness for each time delay have been assessed in Table 3.5. The results show that the system

TABLE 3.5: ITAE values for variations of time delay

Time delay	J_r	J_d	GM (dB)	PM (°)
T = 2.0 s	14.25	48.46	12	62.5
T = 2.5 s	20.68	53.87	9.62	55.9
T = 3 s	32.56	61.09	7.74	49.2
T = 3.5 s	50.94	68.25	6.17	42.5

specifications have been fulfilled. Nevertheless, experiments using higher time delays that are not presented in the table, such as 4 s, show that the minimum requirement of gain margin could not be achieved.

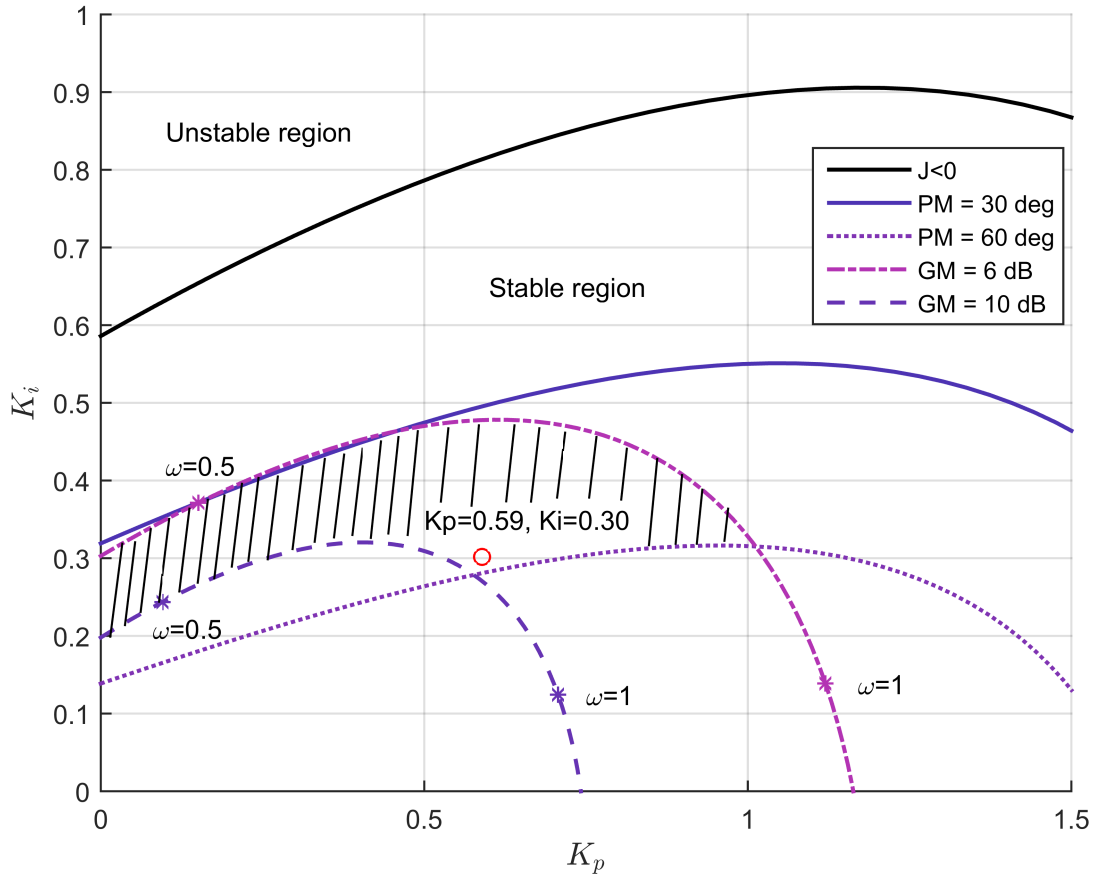

FIGURE 3.15: K_p – K_i plane

Finally, the control system is tested using the TrueTime simulator following the configuration described in Section 3.2.4. The percentage of dropouts is set to 30%, the interference bandwidth is 47 % and $Ts = 0.01\,s$. Results are shown in Figure 3.16. The system presented a sluggish response compared with the output without delays. The measurements of the ITAE criterion are $J_r = 21.74$ and $J_d = 95.36$ showing an increase respect the scenario without the network ($J_r = 14.14$ and $J_d = 69.68$). In general terms, there is a good disturbance rejection and good setpoint tracking despite the presence of time delays and dropouts. Moreover, the performance of the robust PID is superior to the previous model-based controllers.

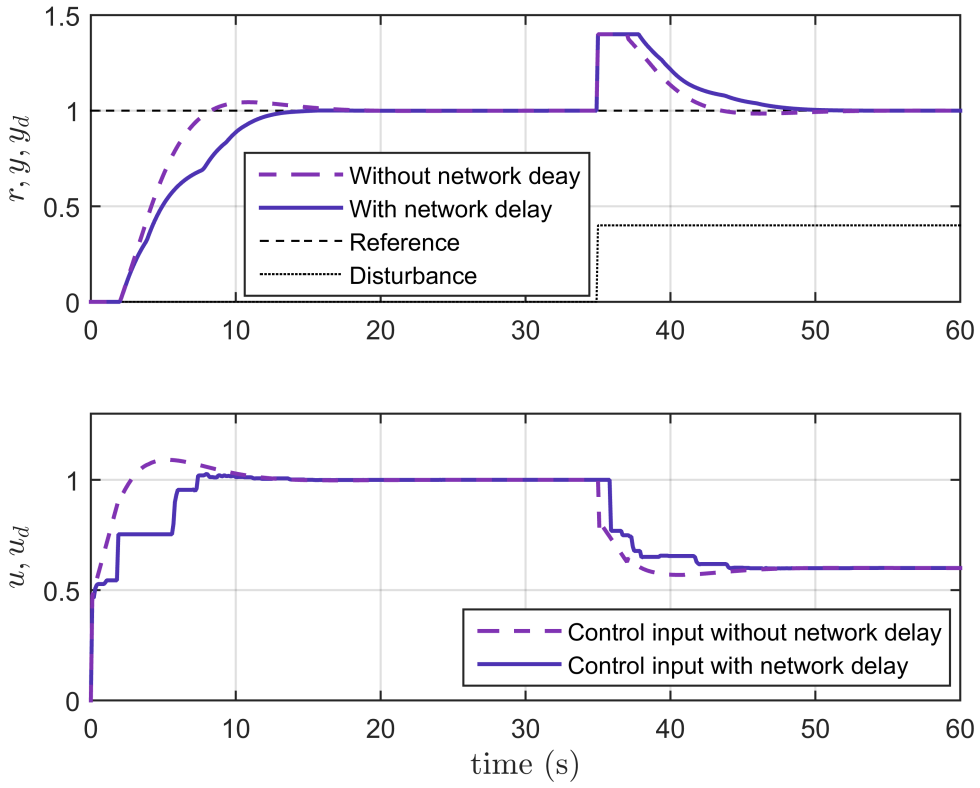

FIGURE 3.16: System outputs for robust PID

3.5 A design of an optimal PID controller for NCS with time-varying delays

The formulation of optimisation problems is a powerful technique in control design. The effectiveness of its application for the tuning of PID controllers is well known.

In this section, an unconstrained optimisation problem is proposed and solved to find the parameters of a new PID controller that minimises a performance index for a NCS subject to random and time-varying delays.

The NCS for different scenarios has been simulated to demonstrate that under the network conditions the optimal controller is robust and has a good performance.

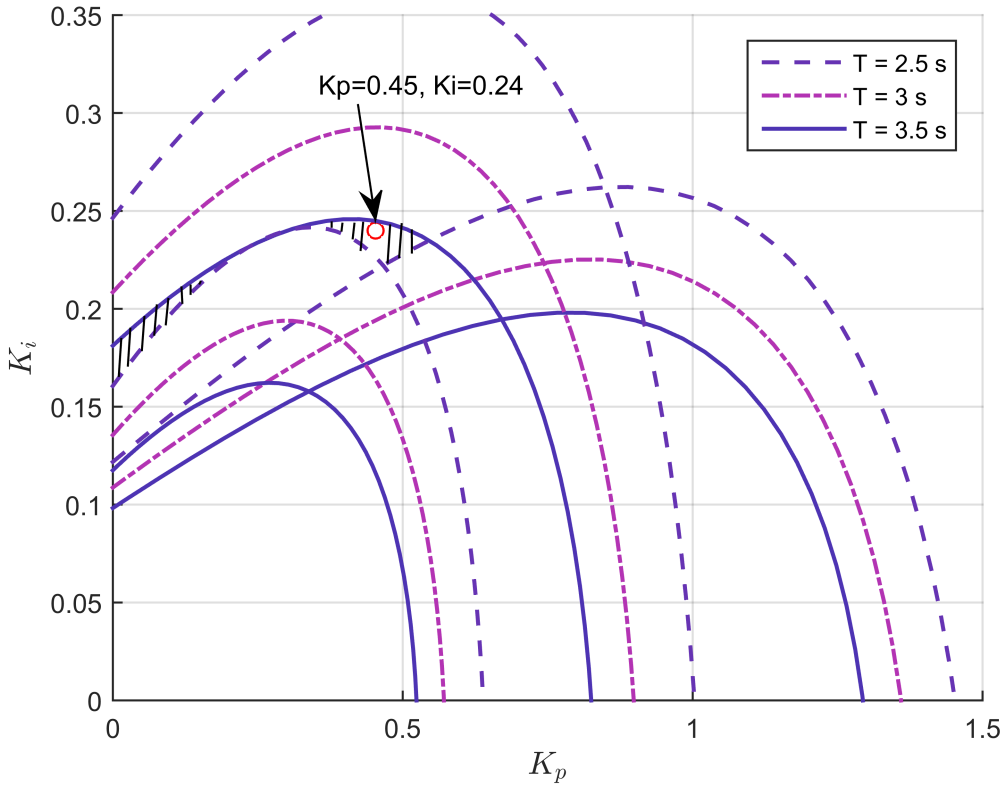

FIGURE 3.17: K_p – K_i plane for different values of time delay

3.5.1 The PID controller

Consider the PID controller with parallel structure:

$$u(t) = K_p\, e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de(t)}{dt} \tag{3.43}$$

where K_p, K_i, K_d are the proportional, integral and derivative gains, respectively. $u(t)$ is the control signal and $e(t)$ is the error signal. The equation is approximated to a discrete-time PID by using a backwards approximation, a sampling time Ts and a filter for the derivative part. The algorithm is given by:

$$\begin{gathered} u(k) = K_p\, e(k) + i(k-1) + K_i\, Ts\, e(k) + \frac{K_d}{K_d + K_p N Ts} d(k-1) + \\ \frac{K_p K_d N}{K_d + K_p N Ts}[y(k-1) - y(k)] \end{gathered} \tag{3.44}$$

where $y(k)$ is the process signal, $i(k)$ is the integral action, $d(k)$ is the derivative action and $(k-1)$ denotes the past value of the signal. N is the filtering constant and it is

selected to be a fraction of the derivative time constant T_d.

3.5.2 The time-varying delay

To simulate the effect of the time-varying delay, the delay distribution of the network has been approximated by a Gamma distribution. The Gamma function Γ is defined as follows:

$$\Gamma(k) = \int_0^\infty x^{k-1} e^{-x} dx, \; k \in (0, \infty) \tag{3.45}$$

where k is the shape parameter. The Gamma probability distribution function with shape parameter k and scale parameter b is given by:

$$f(x) = b^{-k}/\Gamma(k) \;\; x^{k-1} e^{-x/b}, \; x \in (0, \infty) \tag{3.46}$$

The parameters of the Gamma distribution are identified with properties of the network by Pohjola, Eriksson and Koivo (2006). k is the number of hops between the first and last node. The rate parameter is defined as $1/b = k/T$, where T is the mean delay. A S-function of Matlab has been created to generate the random delay with Gamma probability distribution every sampling time.

3.5.3 The optimisation problem

A NCS with time delay from controller to actuator is implemented in Simulink. The delay is represented using a variable time delay block. The S-function generates a number with Gamma distribution and updates the value in the variable time delay block each sampling time.

The tuning of the PID controller is obtained by solving an optimisation problem that minimises a cost function J. The cost criterion is chosen to be the ITAE and is given by:

$$J_{ITAE} = \int_0^\infty t \, |e(t)| \, dt = \int_0^\infty t |r(t) - y(t - \tau(t))| dt \tag{3.47}$$

where $e(t)$ is the signal error, $r(t)$ is the reference signal, $y(t)$ is the system output and $\tau(t)$ represents the random delay.

The Optimisation Toolbox is used to assess the cost criterion and find the minimum value. The function *fminsearch* is selected. The following line of code describes its implementation:

```
x0 = [1.001,1.15,0.1];
options = optimset('MaxIter',5000,'MaxFunEvals',5000);
[xopt,fval,exitflag,output]=fminsearch(@funpidwdp2,x0,options);
```

LISTING 3.1: fminsearch code function

The function called *funcpidwdp2* returns the ITAE index. It defines the PID gains as global variables and calls to the Simulink block where the NCS and the ITAE cost function have been implemented according to (3.47).

$x0$ stands for the initial point given to the optimisation problem. Since a single iteration requires more than one function evaluation, the maximum values have been stated in the options of the *fminsearch* function. The limitation of the optimal tuning is that it might take several iterations to find the local minimum. The maximum number of iterations is limited to 5000. The maximum number of function evaluations per iteration is 5000. The function and optimality tolerances are selected as 1×10^{-4}. If the measure between the current point and a possible minimum is less than optimality tolerance, the solver iterations end.

3.5.4 The optimal PID control algorithm

The optimal PID control algorithm is implemented using the following procedure.

Step 1: Initialisation

(a) Set up the model parameters of the process.

(b) Implement the controller, sensor/actuator and network nodes from the Truetime simulator.

(c) Follow steps c to d from the control algorithm in 3.2.2.

(d) Find from the simulation the mean value of the network delay τ_{sc}.

Step 2: Off-line calculation.

(a) Solve the optimisation problem using the network delay model from Section 3.5.2 and find the optimal PID gains.

(b) Update the PID gains in the simulator.

Step 3: On-line calculation

(a) Compute the control input and apply to the process.

Step 4: Fine-tuning

(a) Check the closed-loop performance and fine-tune the gains if necessary.

3.5.5 Numerical example

Consider the example given in (3.8). The optimisation problem is solved for $k = 3$ and $T = 0.135\ s$ over a simulation of 35 s. The initial point is $x_0 = [1.001, 1.15, 0.1]$. The number of iterations is 1889. The optimal results for the process are: $K_p = 0.4237$, $T_i = 1.5503\,s$ and $T_d = 0.1106\,s$. After some tests the value of $N = 10$ is selected for a good response.

The NCS is implemented using the TrueTime simulator and the network has been configured as in Section 3.2.4. The percentage of dropouts is set to 30%, the interference bandwidth is 47 % and a sampling time of $Ts = 0.008\,s$. The simulation results are depicted in Figure 3.18. The ITAE costs without delay are $J_r = 14.23$ and $J_d = 72.3$. Another test is performed for the system subject to random delay with a Gamma distribution. It is found that the performance is good for small variations of the time delay. The network delay effect using the TrueTime simulator is much more severe than the simulation with variable time with Gamma probability distribution. The ITAE costs for the Gamma distributed delay are $J_r = 17.7$ and $J_d = 50.96$ and they increased to $J_r = 33.4$ and $J_d = 104$ in the TrueTime simulation.

The tracking performance of the proposed design is better than the robust PID controller. However, the robustness is not improved with the optimisation method. To solve this, a constrained optimisation problem that includes robustness requirements is offered in the section.

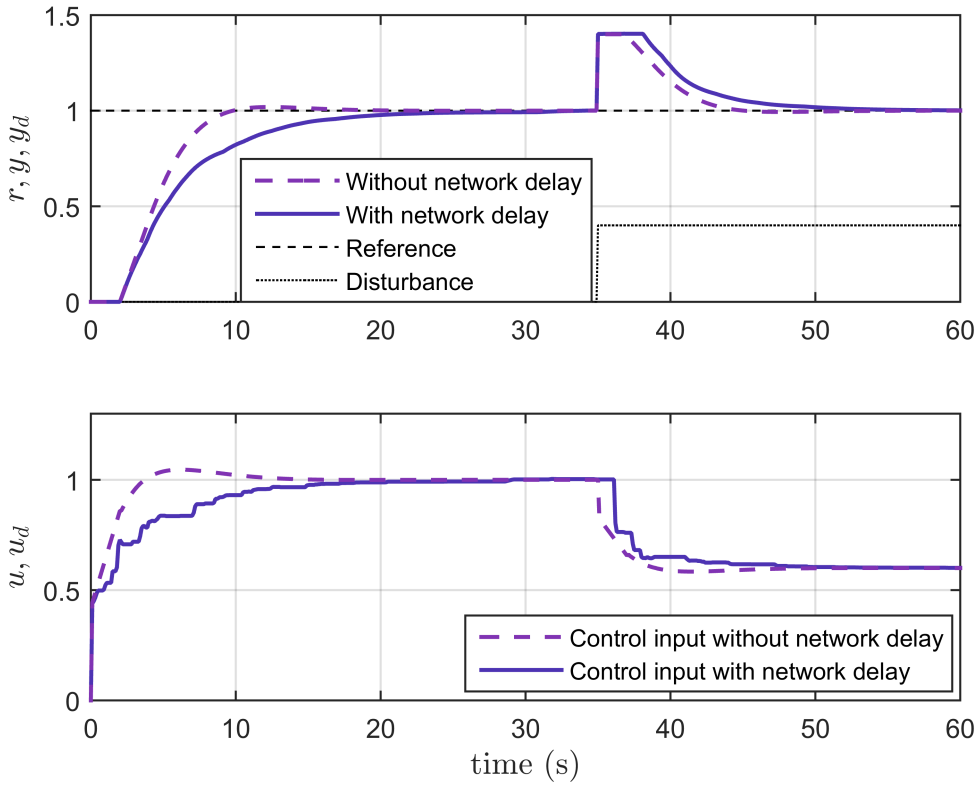

FIGURE 3.18: Systems outputs for optimal PID

3.6 A design of an optimal robust PID controller using the maximum sensitivity

A new constrained optimisation problem is proposed and solved to find an optimal robust PID controller that guarantees the robustness of the system subject to time-varying delays. The robustness is studied using the maximum sensitivity of the system.

3.6.1 Constrained optimisation

The work of Eriksson and Koivo (2005) proposes a tuning method for discrete-time PID controllers where the gains are found by solving an optimisation problem that minimises the cost criterion ITAE while the desired gain and phase margin are set as constraints of the problem. A similar method is followed in this study, but the maximum sensitivity value is used since the complexity of the computation is reduced significantly.

Equation (3.2) defines the sensitivity of the system S and by limiting its maximum value Ms, a good robustness of the system can be achieved. A reasonable range is $Ms = [1.4 - 2.0]$. The lower the value of Ms, the better the robustness. The PID controller is implemented using (3.44). The constrained optimisation problem is formulated as:

$$\begin{aligned} &\min\ f(x) = \int_0^\infty t|r(t) - y(t - \tau(t), x)|dt \\ &\text{s.}t. \qquad h(x) = \begin{cases} -x + \varepsilon \leq 0 \\ M_s - 1.4 - \varepsilon \leq 0 \end{cases} \\ &x = [K_p \quad K_i \quad K_d]^T \in \mathbb{R}^n \end{aligned} \tag{3.48}$$

where $\tau(t)$ represents the random delays. The minimisation cost criterion is chosen to be the ITAE. Considering constraints, firstly, the PID controller parameters have to be positive. Secondly, the robustness is guaranteed if there is at least a maximum sensitivity of $Ms = 1.4$. Defining a small positive value ε the inequality constraints are arranged in the general formulation form.

A block diagram of the closed-loop system is implemented in Simulink following the NCS structure in Figure 2.1. To simulate the effect of the network delays a Gaussian distributed random delay with mean $\mu = 1$ and variance $\sigma^2 = 0.1$ is implemented at the output of the controlled process. Every sampling time, the output of the system subject to the random delay is measured and the optimisation algorithm evaluates and minimises the cost function to find the parameters of the controller.

A Sequential Quadratic Programming (SQP) method is selected to solve the problem and find the controller parameters. The Optimisation Toolbox of Matlab is used. In particular, the function *fmincon* with the *active-set* algorithm is used to find the minimum of the cost function subject to the system constraints.

The optimality tolerance is 1×10^{-6}. If the measure between the current point and the possible minimum is less than optimality tolerance, the solver iterations end. The constraint tolerance is selected as 1×10^{-6}. If the magnitude of any constraint is more than the constraint tolerance the solver reports that the constraints are violated. The maximum number of function evaluations and iterations is set to 200 and 400, respectively.

Once the optimal PID gains are found, the design is tested using the TrueTime simulator.

3.6.2 The optimal robust PID control algorithm

The optimal robust PID control algorithm is implemented using the following procedure.

Step 1: Initialisation

(a) Set up the model parameters of the process.

(b) Implement the controller, sensor/actuator and network nodes from the Truetime simulator.

(c) Follow steps c to d from the control algorithm in 3.2.2.

(d) Select the Maximum sensibility Ms.

Step 2: Off-line calculation.

(a) Solve the constrained optimisation problem using the Gaussian distributed network random delay and find the optimal PID gains.

(b) Update the PID gains in the simulator.

Step 3: On-line calculation

(a) Compute the control input and apply to the process.

Step 4: Fine-tuning

(a) Check the closed-loop performance and robustness and fine-tune the gains if necessary.

3.6.3 Numerical example

For the process shown in (3.8), an optimal PI controller is studied for a random delay. N is selected with a constant value 10. The optimisation problem is done for a simulation

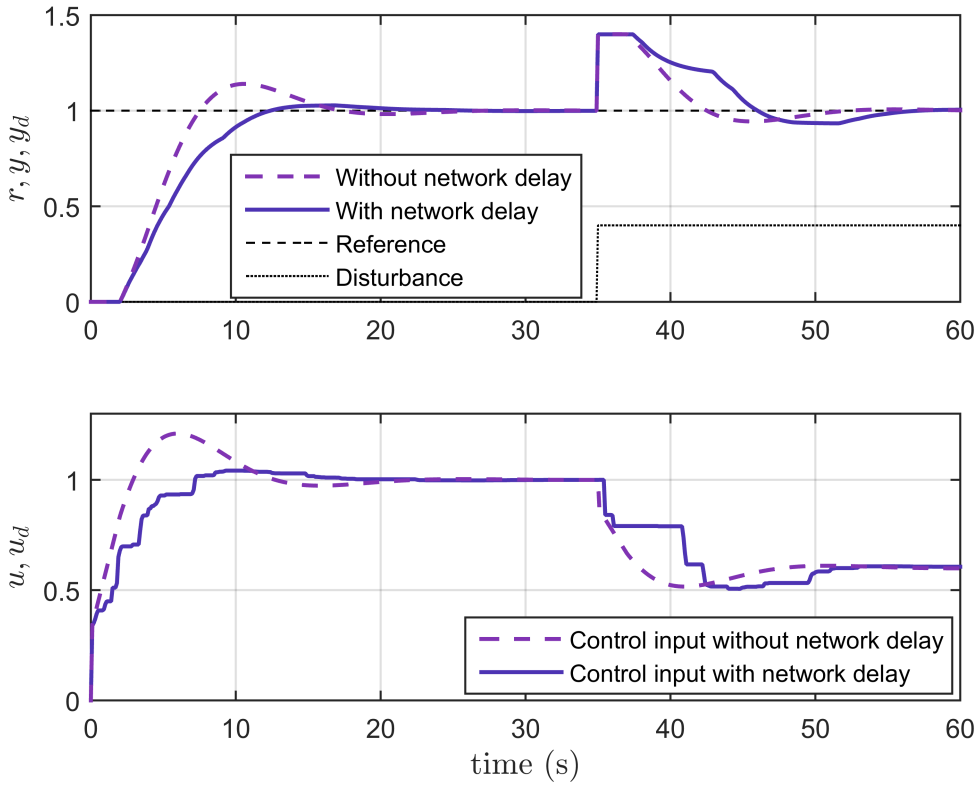

FIGURE 3.19: System outputs for optimal robust PID

of 35 s, the initial point is $x_0 = [0.1\ 0.5]$ and the number of iterations is 26. The optimal results for a $Ts = 0.03\ s$ are: $T_i = 3.72$ and $K_p = 0.32$. Using these values the TrueTime simulator tests are performed following the same configuration of the network than previous examples. Figure 3.19 shows that the responses have an overshoot.

The ITAE costs are: $J_r = 23.12$ and $J_d = 85.44$ for the systems without delays and $J_r = 23.63$ and $J_d = 129.3$ with delays. The performances are affected for the inclusion of the varying delays and loss of data. However, the optimal PI controller shows robustness to the random delay and dropouts. Moreover, the robustness of the optimal robust PID is superior to the robust PID controller designed in Section 3.4 using specifications of phase and gain margins.

3.7 A design of a jitter-aware PID for NCS with time-varying delays

To address the adverse effect of time-varying delays in NCS, a robust PID controller has been implemented for a first-order system. A method by Eriksson and Johansson (2007a) has been used in this section where the AMIGO tuning rules are combined with the maximum time delay that the system can tolerate. This last concept is known as the jitter margin. A set of tuning equations gives the PID parameters that guarantee the robustness of the system. Performance and robustness have been studied with several experiments in the Truetime simulator to validate the effectiveness of the control system.

3.7.1 Introduction

A robust and jitter-aware PID controller for a time-varying delay problem is studied as follows. Consider the following PID controller structure:

$$u(t) = K_p[b\,r(t) - y_f(t)] \; + K_i \int_0^t [r(\tau) - y_f(\tau)]\, d\tau + K_d \left(c\,\frac{dr(t)}{dt} - \frac{dy_f(t)}{dt} \right) \tag{3.49}$$

where b and c are the setpoint weighting factors. $y_f(t)$ is the output after the measurement filter, that is given by:

$$y_f(s) = G_f(s)Y(s) = \frac{1}{(1+sT_f)^n} Y(s) \tag{3.50}$$

where $G_f(s)$ is the filter transfer function, T_f is the filtering constant. The value of n is chosen to obtain an appropriate transfer function.

3.7.2 Jitter margin

The variance of the time delay is studied under the concept of jitter margin. It is defined as the maximum time-varying delay that can be increased in the system without causing instability. Consider a linear time invariant system with process $P(s)$ and controller $C(s)$. The control system is perturbed by an uncertain time-varying delay $\Delta(\upsilon)$ in the

feedback loop. The system is stable for any time-varying delay defined by:

$$\Delta(\upsilon) = \upsilon(t - \delta(t)), \quad 0 \leq \delta(t) \leq \delta_{\max} \quad if \tag{3.51}$$

$$\left| \frac{P(j\omega)C(j\omega)}{1 + P(j\omega)C(j\omega)} \right| < \frac{1}{\delta_{\max}\,\omega}, \quad \forall\, \omega \in [0, \infty[\tag{3.52}$$

where $\delta_{\max}\,\omega$ is the maximum jitter margin.

3.7.3 Tuning rules

The PID parameters for the jitter-aware controller are presented by Eriksson and Johansson (2007a). The authors solved an optimisation problem where the robustness using the maximum sensitivity and the jitter margin are maximised. The resultant tuning rules for a FOPDT process were proposed as follows:

$$\begin{aligned} K_p &= \frac{1}{K}\left(\frac{0.4T - 0.04}{L} + 0.16\right) \\ K_i &= \frac{1}{100K}\left(\frac{-0.11T^3 + 1.5T^2 - 1.5}{L^2} + \frac{0.35T^2 + 4T + 50}{L}\right) \\ K_d &= \frac{1}{100K}(0.4T^2 + 11T) \end{aligned} \tag{3.53}$$

where L is the dead time, K is the gain and T is the time constant of the process.

The controller gains will be set using these rules, and for T_f, b, c will use the AMIGO rules.

The function to calculate the lowest jitter margin value in terms of the time delay τ is:

$$f(\tau) = -12.3\tau^4 + 17.1\tau^3 - 5.5\tau^2 + 0.72 \tag{3.54}$$

And the jitter margin is given by:

$$\delta_{\max} = 0.71L[f(\tau) + 1] \tag{3.55}$$

The tuned controller is tested in a closed-loop system implemented in Simulink following the NCS structure in Figure 2.1 with the modification that, the network delay has been

modelled as a time delay at the output of the controlled process. To simulate the effect of the network delays a distributed random signal with mean one and variance 0.1 have been implemented. Then, the maximum jitter margin is found by computing (3.54) and (3.55).

Once the PID tuning is done, the performance and robustness is studied with several experiments in the Truetime simulator to validate the effectiveness of the design.

3.7.4 The jitter-aware PID control algorithm

The jitter-aware PID control algorithm is implemented using the following procedure.

Step 1: Initialisation

(a) Set up the model parameters of the process.

(b) Implement the controller, sensor/actuator and network nodes from the Truetime simulator.

(c) Follow steps c to d from the control algorithm in 3.2.2.

Step 2: Off-line calculation.

(a) Select the PID gains and time filter constant using the tuning rules.

(b) Approximate the network delay and find the jitter margin.

(c) Implement the NCS and the PID control law.

Step 3: On-line calculation

(a) Compute the control input and apply to the process.

Step 4: Fine-tuning

(a) Check the jitter margin, closed-loop performance and robustness and fine-tune the gains if necessary.

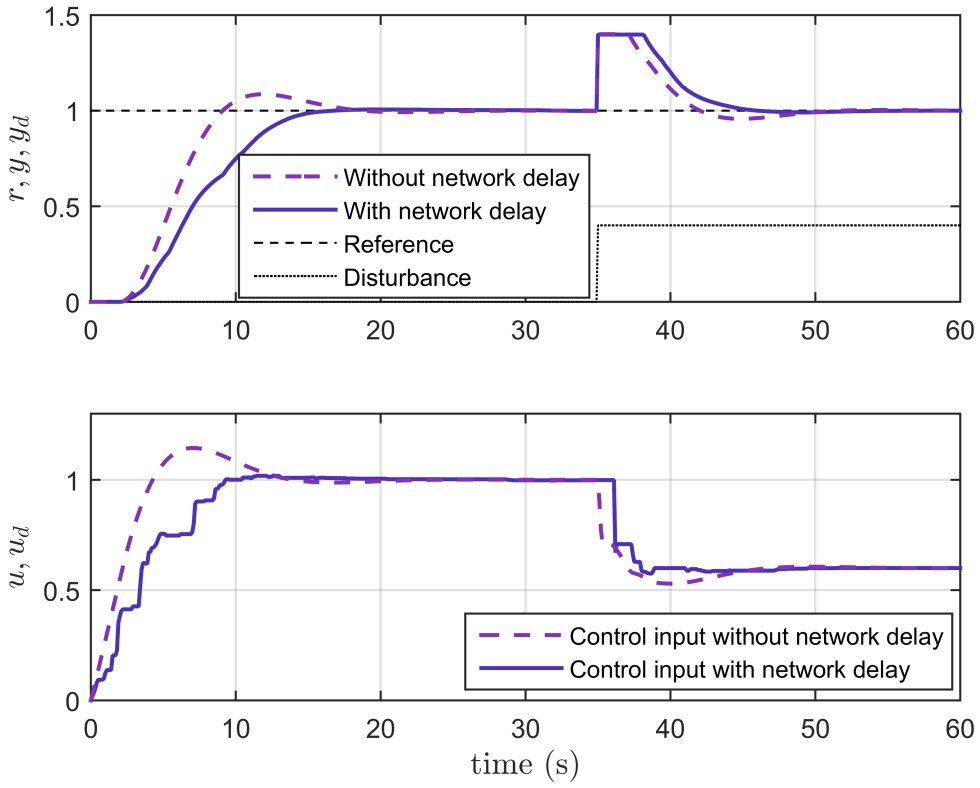

FIGURE 3.20: System outputs for jitter-aware PID

3.7.5 Numerical example

Consider the first-order system given in (3.8). A time constant $T_f = 0.2\,s$ is used for the filter, $b = 0$ and $c = 0$. Applying (3.53), the controller parameters are $K_p = 0.54$, $K_i = 0.31$ and $K_d = 0.24$.

The effect of the network random delay modelled with a distributed random signal with mean one and variance 0.1 has been studied. The response presented a small oscillation in the output and an increase of the overshoot. Using (3.54) and (3.55), the maximum jitter margin is 2.43 s and the maximum delay that the system experienced is 2.21 s. The ITAE values are $J_r = 63.01$ and $J_d = 122.29$.

Following, the design is tested in the simulator. The implementation of the system using the TrueTime simulator follows the same configuration than previous tests. The first experiment has been made for the process without time-varying delays. Figure 3.20 shows the good performance of the system with a small overshoot.

The response for the time-varying delays shows that the controller output is lost during the first seconds of the simulation and again at the 35th second when the disturbance is present. That results in a slower and damped response. The ITAE criteria for the first case are $J_r = 24.13$ and $J_d = 73.5$ and for the second one are $J_r = 35.93$ and $J_d = 84.04$, which gives a better scenario compared to the results without the simulator.

The maximum sensibility value for the system with the additional varying delay is $Ms = 1.96$, which is bigger than the goal of $M = 1.4$ given by the tuning rules, but less than the limit $Ms = 2$. The reason for this is that for lag dominated processes the AMIGO rules could give poor robustness in comparison with the delay-dominant processes. Further tests where the time constant is $T = 0.1\,s$ showed that the maximum sensitivity is $M = 1.37$.

3.8 A design of an optimal immune PID controller for NCS

In this section, a new optimal immune PID controller is applied to a NCS subject to dropouts and time-varying delays. The immune feedback law proposed by Yang, Zhao, Zhou and Liu (2007) is used in this work. A constrained optimisation problem is proposed to find the parameters of the immune PID controller and the effectiveness and robustness of the design are demonstrated through several simulations.

3.8.1 Introduction

Immune systems have been applied in recent years to modern processes that present complex characteristics such as long and time-varying delays (Peng et al., 2011). This methodology shows a good robustness and adaptability when is combined with fuzzy theory in applications of temperature control and electronic throttles as showed by Xue and Yan (2010) and Chen and Chen (2009), respectively.

The immune control system is a physiological action that produces antibodies to combat antigens. The primary components of this system are the recognition cells and the killing cells. When the antigens arrive, recognition cells begin to multiply themselves at the same time they activate the helper T cells (TH). Then, the helper T cells

activate B cells, which secrete the antibodies. The Antigen-Presenting Cell (APC) can also activate the suppressor T cells (TS), which can suppress the secretion of the helper T cells and the B cells.

It can be generalised that the immune feedback algorithm is mainly based on the feedback regulating principle of the T cell. The principle is defined by the number of antigens or antigen concentration, $\varepsilon(k)$, at the kth generation:

$$\varepsilon(k) = \gamma\varepsilon(k-1) - u_{kill}(k-d) \tag{3.56}$$

where γ is the proliferative factor of external substance, u_{kill} is the concentration of the B cells and d is the postmortem interval or the delay time of immune response. The concentration of the B cells can be expressed as:

$$u_{kill}(k) = TH(k) - TS(k) \tag{3.57}$$

where $TS(k)$ is the effect of TS cells on the B cells and $TH(k)$ is the output from TH stimulated by the antigens:

$$TH(k) = K_1\varepsilon(k) \tag{3.58}$$

where K_1 is the stimulating factor of TH. The action from restraining B cell using T cell is given as:

$$TS(k) = K_2 f[\Delta u_{kill}(k)])\varepsilon(k) \tag{3.59}$$

where K_2 is a suppression factor of TS cell and $\Delta u_{kill}(k) = u_{kill}(k-d) - u_{kill}(k-d-1)$. This is the concentration change of the B cells. Finally, $f(\cdot)$ is a non-linear function. Then, a mathematical representation of the concentration of B cells is expressed as:

$$u_{kill} = K_1\varepsilon(k) - K_2\{f[\Delta u_{kill}(k)]\}\varepsilon(k) \tag{3.60}$$

By selecting the amount of the antigens $\varepsilon(k)$ as the control error $e(k)$ and the total stimulation received by B cells, u_{kill} as the control input $u(k)$, the immune feedback law can be described as:

$$u(k) = k\{(1 - \eta f[\Delta u(k)])\}e(k) \tag{3.61}$$

where $k = K_1$, $\eta = K_2/K_1$. Therefore, the immune PID control algorithm can be described by (3.44), where the proportional gain is: $K_{pl} = K(1 - \eta f[\Delta u(k)])\, K_p$. The parameter K is used to control the response speed. The parameter η is used to control

the stabilisation effect. It will make the system to have better stability and smaller overshoot or even no overshoot. This value is limited according to:

$$\eta \in \left\{0, \frac{1}{max(f[\Delta u(k)])}\right\} \tag{3.62}$$

The function $f(\cdot)$ is the regulating effect related to the antibodies concentration. In this work several functions have been tested for the NCS and for the best performance it is selected as:

$$f[\Delta u(k)] = 1 - 2/(e^{-a\,\Delta u(k)} + e^{a\,\Delta u(k)}), \qquad a > 0 \tag{3.63}$$

where $f[\Delta u(k)] \in [0, 1]$ and a is the factor of antibodies concentration. This parameter changes the form of the function for instance, choosing larger values for a, the function can be less smother.

To determine the values of η, a, K and the PID parameters, a constrained optimisation problem has been proposed and solved using *fmincon*. The *active-set* algorithm is selected to find the minimum value for the function J. The maximum number of function evaluations and iterations is set to 3000 and 1000, respectively.

The cost function J has been selected as the ITAE criterion. Considering constraints, firstly, the parameters have to be positive. Secondly, the value of η is restricted to the interval $[0, 10]$ according to (3.62) and (3.63).

The optimisation problem is solved for a system under time-varying delay. The delay has been approximated by a Gaussian distributed random signal, with mean one and variance 0.1. $N = 10$ and sampling time $Ts = 0.015\,s$. Once the optimal values are found, the NCS is implemented and tested in the Truetime simulator.

3.8.2 The optimal immune PID control algorithm

The optimal immune PID control algorithm is implemented using the following procedure.

Step 1: Initialisation

(a) Set up the model parameters of the process.

(b) Implement the controller, sensor/actuator and network nodes from the Truetime simulator.

(c) Follow steps c to d from the control algorithm in 3.2.2.

Step 2: Off-line calculation.

(a) Approximate the network delay and solve the optimisation problem.

(b) Implement the NCS and the immune PID control law using the optimal values.

Step 3: On-line calculation

(a) Compute the control input and apply to the process.

Step 4: Fine-tuning

(a) Check the closed-loop performance and fine-tune the gains if necessary.

3.8.3 Numerical example

Consider the first-order system with time delay process in (3.8). The network environment has been configured as explained in Section 3.2.4. The optimisation problem is solved for a simulation of 35 s, an initial point $x_0 = [0.1; 0.1; 0.1; 0.1; 0.1; 0.1]$ and the number of iterations is 491. The optimality and constraint tolerances are selected as 1×10^{-6}.

The optimal results are: $K_p = 0.08$, $T_i = 0.41\,s$, $T_d = 0.97\,s$, $a = 0.057$, $\eta = 3.18$ and $K = 6.75$. A test is performed where the time-varying delay has been approximated by the Gaussian distributed random signal. The results show a quick response with a small oscillation. Moreover, the recovery from the disturbance signal is quick as well. The ITAE costs are $J_r = 16.44$ and $J_d = 55.27$.

Now, the system is tested using the TrueTime simulator. Figure 3.21 displays the closed-loop response for the system when no time delays are presented. The simulation shows that the control action of the optimal immune PID brings the system smoothly to the setpoint with no overshoot. It rejects the disturbance applied at time $t = 35\ s$.

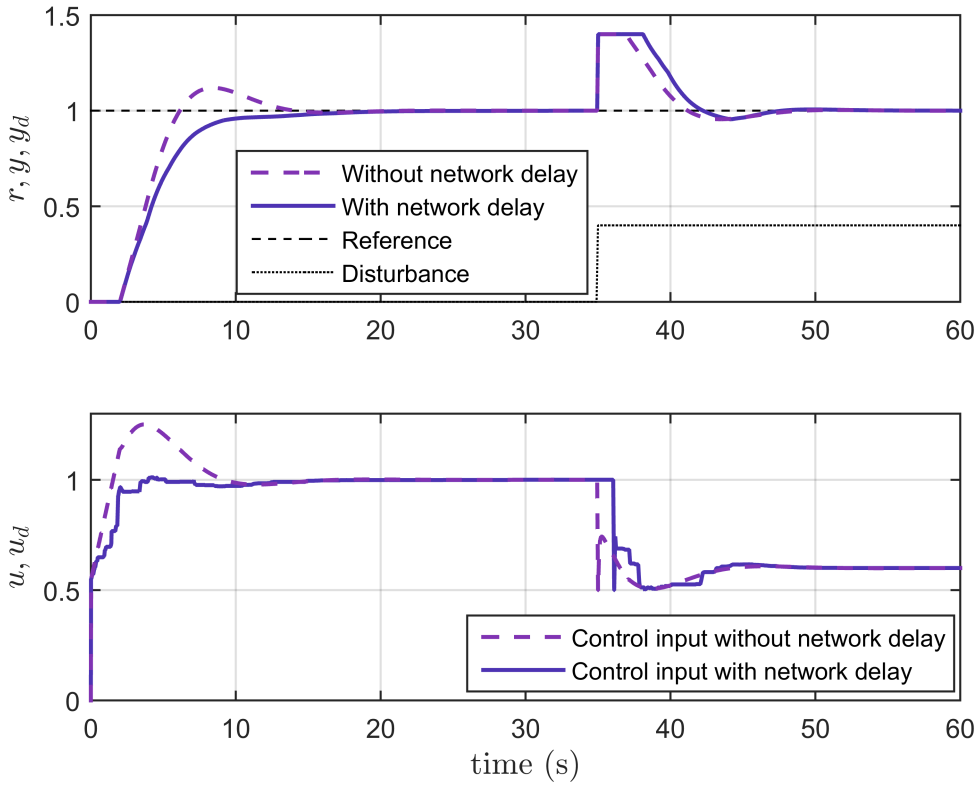

FIGURE 3.21: System outputs for optimal immune PID

The results of the experiment with time delays show a slower response compared with the output without time delays. This is because the controller has to compensate the lack of information. In general terms, there is a good disturbance rejection and good enough setpoint tracking despite the presence of time delays and dropouts. Moreover, the performance of the optimal immune PID is superior to all the previous controllers.

A comparison of the tests results with and without the network simulation shows that the implementation of the delay with a Gaussian distributed random signal represents a good approximation of the time-varying delays scenario.

3.9 Comparison of existing methods

Seven control methods for high percentages of dropouts and time-varying delays are studied in this work. Figure 3.22 shows the comparison of these methods. Performance

and robustness indices are summarised in Table 3.6. It shows performance evaluation based on ITAE cost function values for servo (J_r) and regulatory control J_d. Rise time (RT) and maximum overshoot (OS) are also given. The steady state errors are zero. Considering robustness, the following indices are presented: the maximum sensitivity Ms, gain and phase margin.

TABLE 3.6: Performance evaluation

Control Algorithm	ITAE Servo J_r	ITAE Regulatory J_d	Rise Time RT (s)	Overshoot OS (%)	Maximum Sensitivity Ms	Gain Margin GM (dB)	Phase Margin PM (°)
1. Jitter-aware PID	35.93	84.04	7.1	0.6	5	3	20
2. Optimal robust PI	23.63	129.3	5.4	2.7	1	11	77
3. Optimal PID	33.4	104	5.1	0.1	4	3	22
4. Optimal immune PID	16.1	80.41	4.2	0	3	4	29
5. Smith predictor (PI)	41.4	104.66	5	0.1	1	25	62
6. Adaptive IMC	70.6	162.8	9.9	1.7	1	18	Inf.
7. Robust PID	21.74	95.36	5	0.1	3	4	33

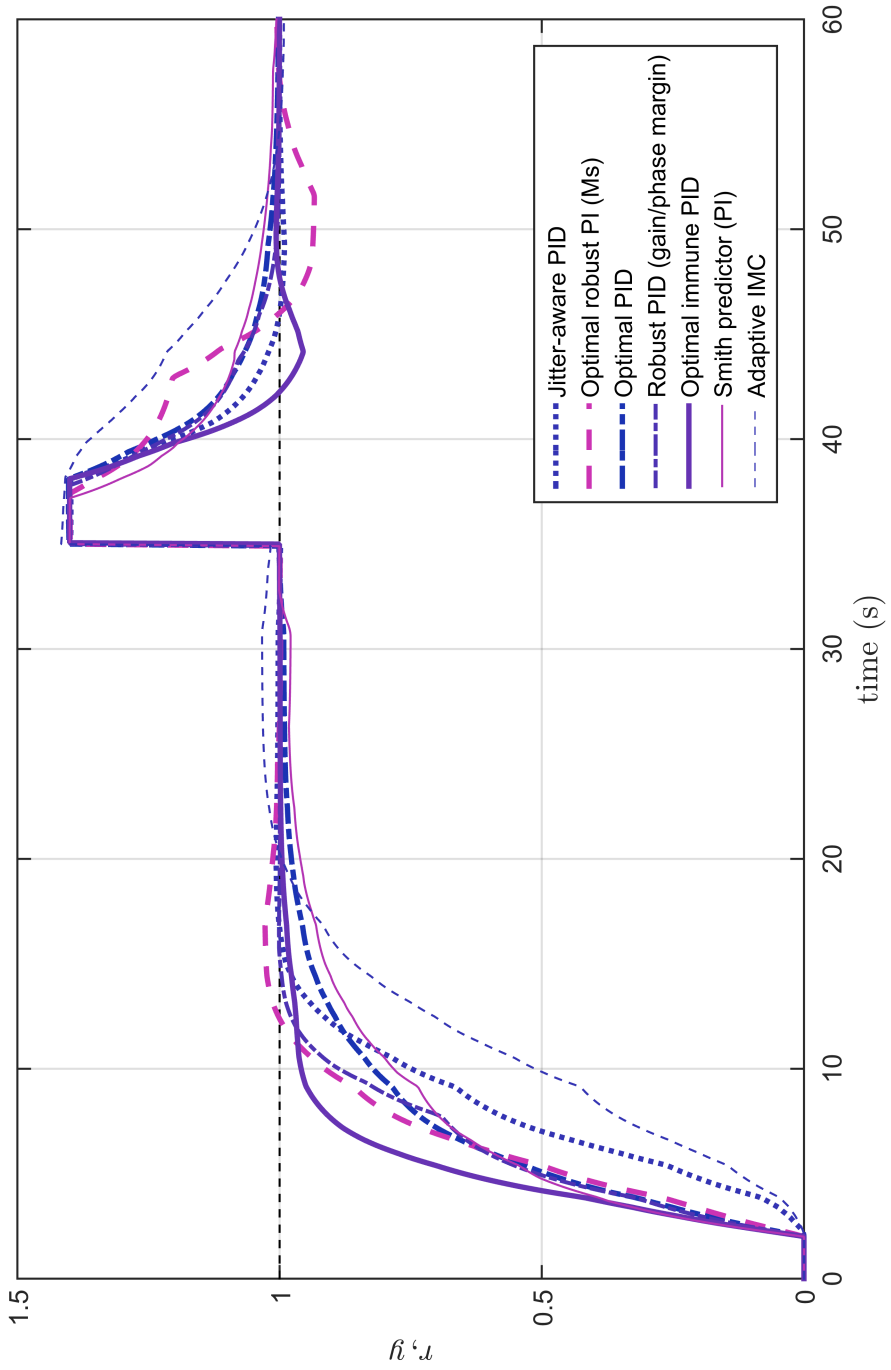

FIGURE 3.22: Process outputs of all methods

The methodology of the optimal immune PID offers the best performance and also a good robustness. Moreover, the difficulty of choosing the adequate values has been overcome with an optimisation method.

The jitter-aware PID and the robust PID shows less robustness than the other robust methods studied here. The restriction of a process with small dead time is a limitation of the jitter-aware tuning rules. The limitation of the robust PID is that robustness of the proposed tuning method is not sustained for long time-varying delays.

From the optimal controllers, the optimal PID has one of the best performances. It is also robust even though it is optimally tuned only in the performance sense. The optimal robust PI presented good margins of robustness and guaranteed $Ms = 1.4$.

From the model based controllers, the Smith predictor gave the best performance with a faster rising time and a lower overshoot. It presents a good robustness too. Therefore, there is a good prediction for slow changes in the network. Conversely, the adaptive IMC presented the poorest performance. This is the result of a sluggish control signal which does not yield good control action under the adverse network conditions. Online computation of IMC parameters revealed that response could be too slow and the optimisation methods can take several seconds to compute the required solution.

3.10 Summary

NCS are gaining importance in the enhancement of the control of simple and complex industrial systems. However, the performance degradation with the introduction of the network is still a concern to address before the systems can be reliable for process control. The need for flexibility and stability in systems under time delays and dropouts support the interest of PID control, Smith predictor and IMC control to operate in NCS. These simple control algorithms offer the opportunity to deploy NCS in industrial applications fully.

These strategies rely on optimisation problems that incorporate robustness and performance restrictions to compute the optimum control signal every sampling time. Approximations of Gamma and Gaussian distributed functions defined with the network parameters have been used to model the random and time-varying delay. The simulations indicate that the model is adequate for the controller design.

Several tests using the Matlab/Simulink Trutime simulator showed that the assumption of time-varying delays with maximum and minimum limits is accurate. Similarly, assuming the percentage of dropouts is varying but bounded the critical value of the percentage of dropouts that can guarantee the stability of the system can be found. It requires incorporating an accurate model of the process and small delays and percentages dropouts to execute the task timely. The long delays and consecutive dropouts affect the stability of the control process adversely and cannot be overcome with fine-tuning of the controller parameters.

Since the proposed designs are based on a simplified model of the plant, the closed-loop system indicated stability for a limited percentage of variations of the model process dynamics and percentage of dropouts. Simulations of stability, performance and robustness analysis demonstrated the applicability for variations of the model parameters of up to 50% and high percentages of dropouts of up to $P_{loss} = 30\%$.

Chapter 4

Wireless networked predictive PID control design for packet dropouts

This chapter presents predictive PID control algorithms to compensate dropouts in WNCS. The dropouts are an important issue in WNCS as mentioned in Chapter 2. Five control strategies based on MBPC with PID control structure have been studied. The problem of the occurrence of dropouts from sensor to controller is compensated by combining the controllers with a Kalman Filter (KF). The measured output y is switched to the Kalman estimation $\hat{y}$ allowing the controller to have always information of the process even in the presence of dropouts. In the last design the constraint handling is presented to stop input saturation.

Notably, the proposed strategies are also applicable to compensate for dropouts from controller to actuator. The method is applied to a second-order and a non-minimum phase system with delays. The control system is implemented using the Matlab/Simulink-based TrueTime network simulator. Performance and robustness analysis is investigated and the results show the design is stable to higher occurrences of dropouts in comparison with the control schemes of Chapter 3.

4.1 Introduction

The compensation of dropouts to ensure reliability on the WNCS represents a challenging problem to the traditional control methodologies. Among these methods, the predictive control scheme is considerably effective since it can actively compensate for consecutive packet dropouts (Sun et al., 2014a). Moreover, the method has been proved to be robust to perturbations and leads to efficient controllers used in many industrial applications (Camacho and Bordons, 2007). However, PID control has received the most attention in the history of process control. The research of a simple and flexible algorithm that can effectively compensate for dropouts in WNCS, is the motivation to create new predictive control algorithms with PID structures. Moreover, it will be economically beneficial to apply the predictive approaches to PID control loops and adapt them to network communication.

For instance, Miklovičová and Mrosko (2012) addressed the compensation of dropouts using GPC and pole placement structure to design a PID controller. The approach of Tan et al. (2002) combined the optimal tracking control of the GPC and the PID structure. Hassan et al. (2016) combined a Smith predictor with PI controller to compensate variable delays and disturbance in WNCS. A similar solution is postulated by Wu, Wu and Zhao (2016) to compensate random delays and dropouts.

Note that in the WNCS approaches loss of information is generally assumed only in one direction. In comparison, the new predictive controllers have the advantage of compensating both communication directions and dealing with a high incidence of dropouts, which has not been addressed in the literature. Also, the new predictive PID controllers provide a reliable solution to WNCS where stability with robustness to disturbances is demonstrated.

The proposed control scheme is presented by the block diagram depicted in Figure 4.1.

The control signal is u , y stands for process output, e is the error and r is the reference signal. $C(z)$ is the controller and the controlled plant is $G_p(z)$. A WNCS whose sensor and control information is transported over a wireless network is considered. The dropouts from sensor to controller and from controller to actuator are represented as d_{psc} and d_{pca}, respectively. A dropout detector has been configured in the Truetime simulator to identify the occurrence of dropouts. The controller reads the information from sensor every sampling time. In case the sensed output y_s is not available, a

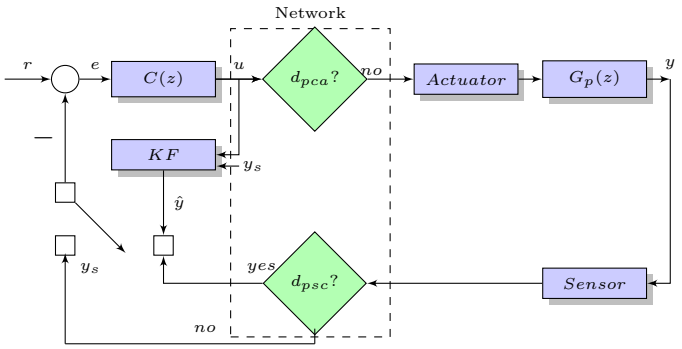

FIGURE 4.1: Diagram of predictive controllers structure

switch function selects the Kalman estimation $\hat{y}$ allowing the controller to always have information of the process even in the presence of dropouts. The simulations results show that the prediction is very aggressive to maintain the control performance. The Kalman filter corrects its estimation and tracks the process output very quickly. This is especially important during a switch between the estimation and the current process to avoid abrupt changes.

4.1.1 Preliminaries: Network modelling

4.1.2 Network constraints

A Wireless Local Area Network (WLAN) is selected in this study. Due to collisions or congestion in the channel, the system has to tolerate dropouts. As mentioned before in Chapter 2, wireless communications bring outstanding advantages but also its limited capacity leads to significant problems. Loss of information is one of the most critical ones and the one is addressed in this chapter.

Dropouts can be modelled using a stochastic or a deterministic approach. The first one, mostly employs independently and identically distributed Bernoulli process with probability of dropout $P_{loss} \in [0, 1]$. For instance, Sun, Wu, Liu and Wang (2014b) transformed the system with random delays and packet dropouts into a stochastic process. Sinopoli, Schenato, Franceschetti, Poolla, Jordan and Sastry (2004) proposed a Kalman estimation where the network was modelled using a stochastic process and showed the existence of a critical value for the percentage of dropouts. However, Li, Shi, Wang and Agarwal (2015) claimed that in these approaches, the critical value is found using its lower and upper bounds. Therefore, in this thesis an optimal deterministic

control methodology is proposed. Furthermore, to get deterministic stability results, the total percentage of dropouts and delays are assumed to be bounded. Also, the maximum number of dropouts that occur consecutively has been investigated γ_{max}. The approach has been presented involving different scenarios of the percentage of dropouts. Simulation experiments show that these assumptions are practical. In NCS, most transmissions are subject to random delay nonetheless, this problem has been covered before in Chapter 3.

4.2 Model based predictive control with PID structure

4.2.1 MBPC formulation

This section defines the process model and the predictive algorithm for the MBPC design. In the first part, the MBPC method is reviewed to interpret the modelling and procedure to obtain the control law. If the reader wishes to learn more, they are referred to text books, for example Camacho and Bordons (2007).

Consider the following MBPC quadratic cost function which seeks a set of optimal control signals that minimises J:

$$J = \sum_{j=N_1}^{N} [\hat{y}(k+j) - r(k+j)]^2 + \sum_{j=N_1}^{N_u} \lambda[\,\Delta u(k+j-1)]^2 \tag{4.1}$$

where N_1 and N are positive scalars indicating the initial and final predictive horizons. λ is a constant weight used to penalise the control effort. N_u is the control horizon. The future reference trajectory is $r(k+j)$ and has been assumed to be known. The control objective is to minimise the cost function to compute the future control signals that guaranties that the future process output $y(k+j)$ follows the future reference $r(k+j)$. Meantime it assures that the control signal is penalized as well.

The control law in MBPC is computed using a linear quadratic cost function. The advantage of combining this with the linear process is that an optimisation problem can be proposed and its solution is an optimal minimum. This guarantees a robust design.

It is considered that after an interval (horizon), the control signal does not change any more until the next sampling instant:

$$\Delta u(k+j-1) = 0 \quad for \quad N_u \leq j \leq N \tag{4.2}$$

To optimise the performance, appropriate horizons and an accurate model are required. To find the prediction of process output $\hat{y}(k+j)$, a linear SISO plant is described using the Controlled Auto Regressive and Integrated Moving–Average (CARIMA) model (Clarke, Mohtadi and Tuffs, 1987):

$$A(q^{-1})y(k) = q^{-d}B(q^{-1})u(k-1) + C(q^{-1})\xi(k)/\Delta \tag{4.3}$$

where $y(k)$ and $u(k)$ are the process output and the control input, respectively. The process delay is d. A, B and C are polynomials function of the backwards shift operator q^{-1} such that:

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + a_2 q^{-2} + \cdots + a_{n_a} q^{-n_a} \\ B(q^{-1}) &= b_0 + b_1 q^{-1} + b_2 q^{-2} + \cdots + b_{n_b} q^{-n_b} \\ C(q^{-1}) &= c_0 + c_1 q^{-1} + c_2 q^{-2} + \cdots + c_{n_c} q^{-n_c} \end{aligned} \tag{4.4}$$

The model represents the uncertainty of random disturbances in the process. $\xi(k)$ is a zero mean white noise, and $\Delta = 1 - q^{-1}$ is a difference operator, indicating the difference between the current time point and the previous time point. The proposed model is more appropriate in industrial applications where disturbances are non-stationary. For simplicity, C is chosen as one in the following analysis. Next, a Diophantine equation is used to find the output predictions:

$$1 = E_j(q^{-1})\Delta A(q^{-1}) + q^{-j}F_j(q^{-1}) \tag{4.5}$$

where E_j and F_j are polynomials. Multiplying (4.3) by $\Delta E_j(q^{-1})q^j$ gives:

$$\Delta A(q^{-1})E_j(q^{-1})\hat{y}(k+j) = E_j(q^{-1})B(q^{-1})\Delta u(k+j-d-1) + E_j(q^{-1})\xi(k+j) \tag{4.6}$$

The best estimation of the future disturbance is achieved by selecting $\xi(t+k) = 0$. Substituting $A(q^{-1})E_j(q^{-1})$ from (4.5) in (4.6), it results:

$$\left[1 - q^{-j}F_j(q^{-1})\right]\hat{y}(k+j) = E_j(q^{-1})B(q^{-1})\Delta u(k+j-d-1) \tag{4.7}$$

Simplifying results in:

$$\hat{y}(k+j) = F_j(q^{-1})y(k) + E_j(q^{-1})B(q^{-1})\Delta u(k+j-d-1) \tag{4.8}$$

where

$$E_j(q^{-1}) = e_{d+j,0} + e_{d+j,1}\, q^{-1} + \cdots + e_{d+j,j-1}\, q^{-(d+j-1)}$$
$$F_j(q^{-1}) = f_{d+j,0} + f_{d+j,1}\, q^{-1} + \cdots + f_{d+j,n_a}\, q^{-n_a}$$

Define $G_j = E_j(q^{-1})B(q^{-1})$. If the set of j ahead predictions is considered, (4.8) can be written as:

$$\mathbf{y} = \boldsymbol{F}(q^{-1})y(k) + \boldsymbol{G}\boldsymbol{u} + \boldsymbol{G'}(q^{-1})\Delta u(k-1) \tag{4.9}$$

where

$$\begin{aligned} \mathbf{y} &= [\hat{y}(k+d+1) \quad \hat{y}(k+d+2) \quad \cdots \quad \hat{y}(k+d+N)]^T \\ \mathbf{u} &= [\Delta u(k) \quad \Delta u(k+1) \quad \cdots \quad \Delta u(k+N_u-1)]^T \end{aligned} \tag{4.10}$$

$$\begin{aligned} \boldsymbol{F}(q^{-1}) &= \begin{bmatrix} F_{d+1}(q^{-1}) \\ F_{d+2}(q^{-1}) \\ \vdots \\ F_{d+N}(q^{-1}) \end{bmatrix}, \; \boldsymbol{G} = \begin{bmatrix} g_0 & 0 & \cdots & 0 \\ g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ g_{N-1} & g_{N-2} & \cdots & g_0 \end{bmatrix} \\ \boldsymbol{G'}(q^{-1}) &= \begin{bmatrix} (G_{d+1}(q^{-1}) - g_0)q \\ (G_{d+2}(q^{-1}) - g_0 - g_1 q^{-1})q^2 \\ \vdots \\ (G_{d+N}(q^{-1}) - g_0 - \cdots - g_{N-1}\, q^{-(N-1)})q^N \end{bmatrix} \end{aligned} \tag{4.11}$$

Notice that, matrix $\boldsymbol{G}$ contemplate future control predictions and $\boldsymbol{G'}$ is the sub-matrix extracted from $\boldsymbol{G}$ relate to the past control signals. Hence, the cost function J is modified by replacing (4.9) in (4.1):

$$\begin{aligned} J = &[\boldsymbol{F}(q^{-1})y(k) + \boldsymbol{G'}(q^{-1})\Delta u(k-1) + \boldsymbol{G}\boldsymbol{u} \text{ - } \boldsymbol{r}]^T \\ &[\boldsymbol{F}(q^{-1})y(k) + \boldsymbol{G'}(q^{-1})\Delta u(k-1) + \boldsymbol{G}\boldsymbol{u} \text{ - } \boldsymbol{r}] + \lambda \boldsymbol{u}^T \boldsymbol{u} \end{aligned} \tag{4.12}$$

where $\boldsymbol{r} = [r(k+d+1) \quad r(k+d+2) \quad \cdots \quad r(k+d+N)]^T$. The minimum J is obtained by computing the gradient $\partial J/\partial \Delta u(k) = 0$. The future control sequence is given by:

$$\Delta u(k) = \boldsymbol{K}(\mathbf{r} - \boldsymbol{f}) \tag{4.13}$$

where $\boldsymbol{K} = (\boldsymbol{G}^T\boldsymbol{G} + \lambda \boldsymbol{I})^{-1}\boldsymbol{G}^T$ and $\boldsymbol{f} = \boldsymbol{G'}(q^{-1})\Delta u(k-1) - \boldsymbol{F}(q^{-1})y(k)$. Note that $\boldsymbol{f}$ contains the signals that depend on the present and past. According to the MBPC

algorithm, only the first element of the set of future control signals is applied to the system and computed again in the next sampling instant. Thus, one can rewrite (4.13) as:

$$\Delta u(k) = \boldsymbol{K}_{MBPC}\, \boldsymbol{e}(k) \tag{4.14}$$

where $\boldsymbol{K}_{MBPC} = [1 \quad 0 \quad \cdots \quad 0]\boldsymbol{K}, \quad \boldsymbol{e}(k) = \mathbf{r} - \boldsymbol{f}$.

4.2.2 Predictive PID structure

The velocity form of the PID controller with sampling time Ts is considered:

$$\Delta u(k) = k_p[e(k) - e(k-1)] + k_i\, Ts\, e(k) + \frac{k_d}{Ts}[e(k) - 2e(k-1) + e(k-2)] \tag{4.15}$$

The matrix representation is:

$$\Delta u(k) = \boldsymbol{K}_{PID}[\boldsymbol{r}(k) - \boldsymbol{y}(k)] = \boldsymbol{K}_{PID} e(k) \tag{4.16}$$

where

$$\begin{aligned} \boldsymbol{K}_{PID} &= [k_p\ k_i\ k_d] \begin{bmatrix} 0 & -1 & 1 \\ 0 & 0 & 1 \\ 1 & -2 & 1 \end{bmatrix} \\ \boldsymbol{y}(k) &= [y(k-2)\ y(k-1)\ y(k)]^T \\ \boldsymbol{e}(k) &= [e(k-2)\ e(k-1)\ e(k)]^T \\ \boldsymbol{r}(k) &= [r(k-2)\ r(k-1)\ r(k)]^T \end{aligned} \tag{4.17}$$

Since MBPC is optimal robust, the challenge is to find the PID gains such that the control signal (4.16) is as close as possible to the MBPC signal (4.14). Thus, an optimisation problem is proposed as:

$$\begin{aligned} &\min\ f(x) = \sum_{k=0}^{k_f} |e_{diff}(k,x)| \\ &\text{s.}t. \quad h(x) = \begin{cases} -k_p + \varepsilon \le 0 \\ -k_i - \varepsilon \le 0 \\ -k_d - \varepsilon \le 0 \end{cases} \\ &\boldsymbol{x} = [k_p \quad k_i \quad k_d]^T \in \mathbb{R}^n \end{aligned} \tag{4.18}$$

where $e_{diff}(k) = u_{MBPC}(k) - u_{PID}(k)$ is the error between the control signals, k_f is the final time of the simulation and ε is a small positive scalar. The minimisation cost criterion is chosen to be the IAE. The cost function has been defined to compute the optimal PID gains in such way that the error between the MBPC control signal and the predictive PID is almost zero. The function *simulannealbnd* from the Optimization Toolbox is selected to assess the cost function and find the minimum value. The simulation time is $k_f = 5\ s$.

The function *simulannealbnd* is selected since it solves quickly the optimisation problem subject to the bounds of the PID gains. The following line of code describes its implementation:

```
x0 = [-4.3819  \;  0.4911   \; 4.2522];
lb = [0;0;0];
ub = [10;10;10];
% options = saoptimset('TolFun',1e-6);
[xopt, fval] =simulannealbnd(@(x) fun(x),x0,lb,ub);
```

LISTING 4.1: simulannealbnd code function

$x0$ stands for the initial point given to the optimisation problem. The upper and lower bounds have been stated in the options of the function. fun computes the MBPC and PID control laws and calculates the cost criterion J. The maximum number of iterations is 200. The optimality tolerance is 1×10^{-6}. If the measure between the current point and the possible minimum is less than optimality tolerance, the solver iterations end.

Once the optimal gains are found, the PID control signal can be implemented using (4.16).

4.2.3 Dropouts from sensor to controller compensation

4.2.3.1 Estimation algorithm

In this section, the design of the Kalman filter is presented. It will have an important role to improve the performance of the whole control system. The occurrence of dropouts during the transmission from the sensor to the controller results in an open-loop

system which degrades the reliability of the WNCS. To solve this problem a Kalman filter is proposed to estimate $\hat{y}(k)$. An estimator algorithm is used according to:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + v(k) \\ y(k) &= Cx(k) + w(k) \end{aligned} \tag{4.19}$$

where the process noise $v(k)$ and measurement noise $w(k)$ are independent Gaussian white sequences with covariance Q_f and R_f, respectively. The covariance is described as follows:

$$\mathbb{E}\left\{ \begin{bmatrix} \omega(k) \\ \upsilon(k) \end{bmatrix} \begin{bmatrix} \omega(k) & \upsilon(k) \end{bmatrix} \right\} = \begin{bmatrix} Q_f(k) & S(k) \\ S(k)^T & R_f(k) \end{bmatrix} \delta(k) \tag{4.20}$$

where $\delta(k)$ is the Kronecker delta function. The cross covariance $S(k)$ is assumed to be zero for independent processes. $\mathbb{E}$ stands for the estimation. The matrices A, B and C are calculated using the discrete transformation from process transfer function to state-space representation. A Zero Order Hold (ZOH) is used. The sampling time is selected according to the process. The control input $u(k)$ is not used to compute the output signal $y(k)$ therefore, $D = 0$. The KF gives an estimation as follows:

$$\begin{aligned} \hat{x}(k+1) &= A\hat{x}(k) + Bu(k) + K_f[y(k) - \hat{y}(k)] \\ \hat{y}(k) &= C\hat{x}(k) \end{aligned} \tag{4.21}$$

where $\hat{y}(k)$ is the estimation of $y(k)$ at time k. K_f represents the steady-state filter gain that is calculated as:

$$K_f = PC^T R_f^{-1} \tag{4.22}$$

where P is the covariance of estimation error that satisfies the steady state Riccati Difference Equation ARE:

$$PA^T + AP - PC^T R_f^{-1} CP + Q_f = 0 \tag{4.23}$$

By choosing appropriate values for Q_f and R_f the filter gain is computed. In case of dropouts, the predictive PID works with the estimation $\hat{y}(k)$ allowing the control system to have always the process information. The observer is implemented on the controller side using the estimate space-state model of the process.

The estimation error covariance converges to a unique solution based upon the assumptions that the pair of matrices (A, C) is observable and $(A, Q^{1/2})$ is stabilizable. For

stability of the predictive control, the model is assumed to be stable and the closed-loop system is assumed to be controllable and observable.

4.2.4 Predictive PID control algorithm

The predictive PID control algorithm can be implemented using the following procedure.

Step 1: Initialisation

(a) Set up the discrete model of the process.

(b) Select the prediction horizons N, N_u and the future reference vector.

(c) Implement the controller, sensor/actuator and wireless network nodes from the Truetime simulator.

(d) Add the dropout detector to the controller code.

(e) Select the network configuration and the percentage of dropouts.

(f) Set an interfering node to increase the time delays.

Step 2: Off-line calculation

(a) Compute the prediction matrices $\boldsymbol{F}(q^{-1}), \boldsymbol{G}$ and $\boldsymbol{G}'(q^{-1})$.

(b) Calculate the MBPC gain, $\boldsymbol{K}$, using (4.14).

Step 3: On-line calculation

(a) Read the actual information from the process. In case of dropouts, compute the process estimation using the KF.

(b) Compute the MBPC control law and solve the optimisation problem in (4.18) *to find the optimal PID gains.*

(c) Apply the predictive PID control signal.

Step 4: Fine-tuning

(a) Check the closed-loop performance and fine-tune λ if necessary.

4.2.5 Numerical example 1

To illustrate the results consider the following linear control system, set as the nominal example in this chapter:

$$G_p(z) = (-0.0033z^{-1} + 0.017z^{-2})/(1 - 1.07z^{-1} + 0.095z^{-2}) \tag{4.24}$$

The sampling time is $T_s = 0.01\ s$. The proposed algorithm has been implemented and tested using the TrueTime network simulator configured for wireless protocol 802.11b (WLAN), with a data rate of 800000 bits/s. The minimum frame size has been selected as 272 bits.

Next, the dropouts have been set up through a loss of probability of approximately $P_{loss} = 30\%$ in the TrueTime network block parameters. Although different values of the penalty in the control action can be selected, for a faster response of the closed-loop $\lambda = 0.1$ is chosen. The prediction horizon has been chosen $N = 30$ and the control horizon $N_u = 1$. The MBPC gain is $K_{MBPC} = [-26.25\ 25.78\ -2.26\ -0.29\ -0.41]$.

The optimisation problem in (4.18) has been solved using the function *simulannealbnd*. The function parameters have been defined as follows. The initial point is selected as $x_0 = [0.1\ 0.1\ 0.1]$, the lower limits as $lb = [0\ 0\ 0]$ and the upper limits as $ub = [20\ 20\ 20]$. The function tolerance is 1×10^{-6}. The number of iterations is 2572 and the number of function evaluation per iteration is 2627. The optimal PID gains are: $k_p = 17.68$, $k_i = 19.99$ and $k_d = 4.44$. For the KF design the noise covariances are chosen as: $Q_f = R_f = 1 \times 10^{-8}$. The vector of Kalman gains converges in steady state to the value: $K_f = [0.31\ 0.32]^T$.

Figure 4.2 shows that the controller acts very fast and the process output reaches the reference. It can be seen that the controller has not been constrained and there is an aggressive behaviour to cope with the dropouts. By relaxing the weighting factor, slower responses can be found nonetheless, the performance is deteriorated as a consequence. The IAE criterion returned a value of $J = 0.096$ for both responses. Figure 4.2 shows that the KF provides an accurate estimate of the process output. Moreover, the percentage and occurrence of dropouts during the simulation are depicted in Figure 4.3. The control system is stable for the entire drop of sensor and controller packets. Further simulations where higher percentages of dropouts are applied show that the stability is severely compromised. The percentages of dropouts could be increased up to $P_{loss} = 40\%$ which is the threshold where the closed-loop appears to be stable. A

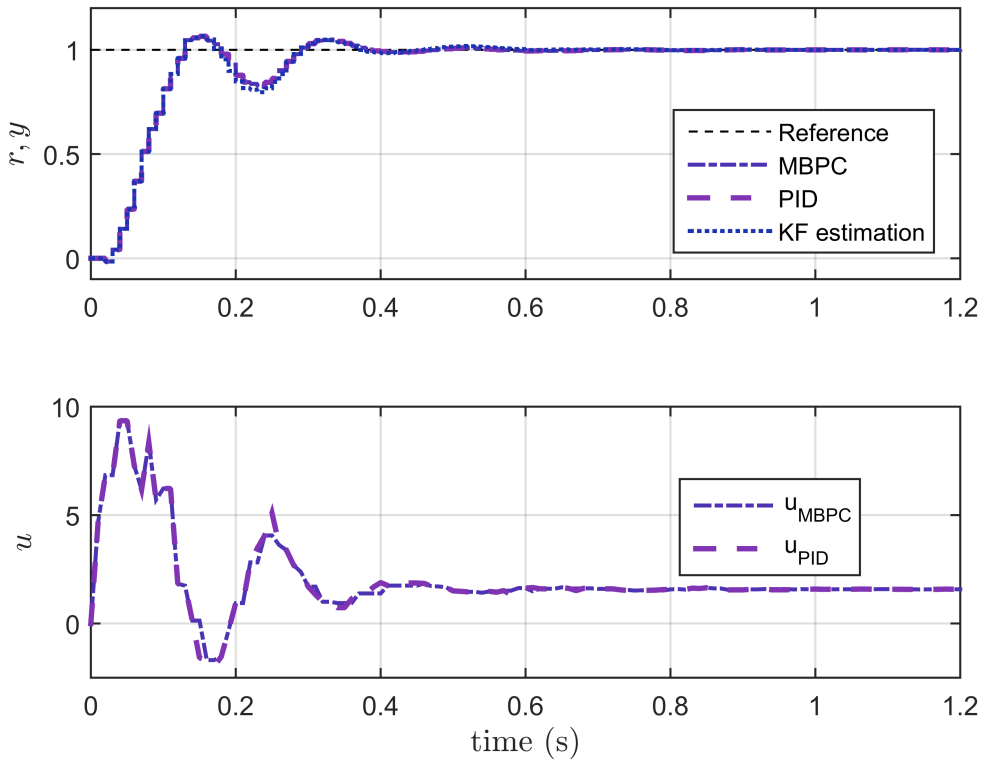

FIGURE 4.2: System outputs for MBPC and predictive PID, $\lambda = 0.1$

new approach is introduced in Section 4.3 to offer a less conservative percentage of dropouts.

4.3 Networked predictive control for consecutive dropouts

The present section addresses the occurrence of consecutive dropouts by combining the receding control algorithm with PID control. In the previous implementation, only the first element of the equation (4.14) is used at every sampling time. However, $N_u - 1$ predictive control signals can be calculated as stated in (4.10). It has been assumed that $N_u = N$.

Taking advantage of this feature, all the control signals are calculated and saved at each sampling instant. If packets from the sensor are dropped at present k, the control

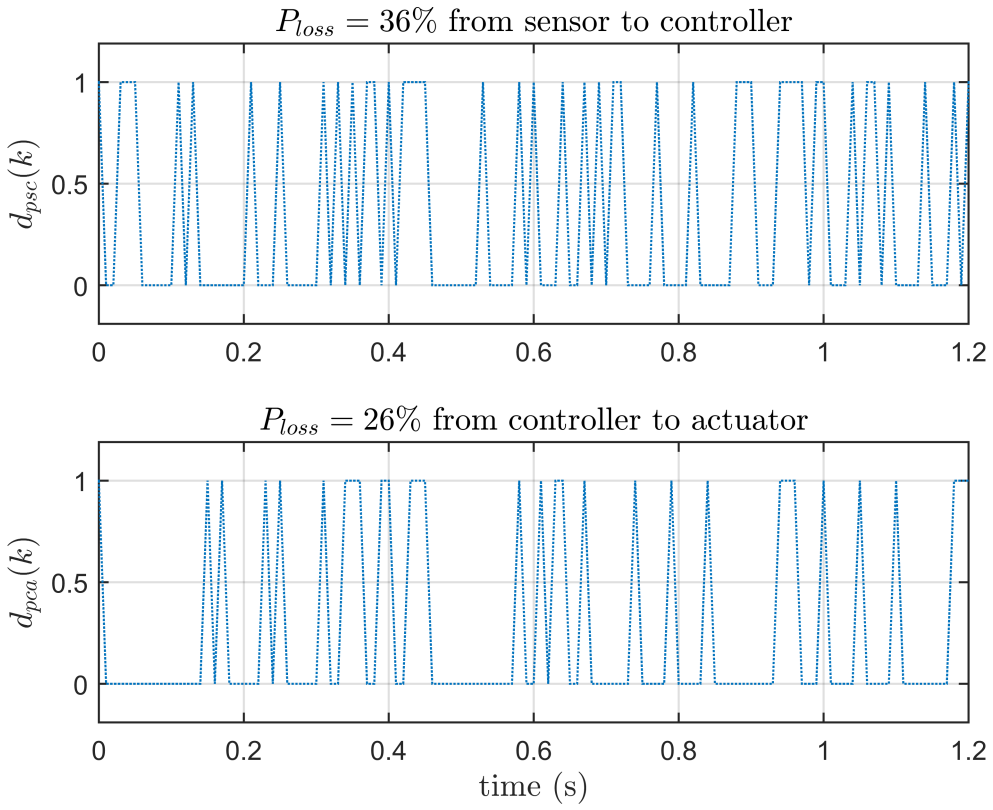

FIGURE 4.3: Time instants of data dropouts for predictive PID

system sends the estimation $u(k+1)|k-1$ stored from the previous sampling time $k-1$. To achieve the proposed algorithm the control law in (4.13) is reformulated as:

$$\Delta u(k+j) = K_{(j+1)}(\mathbf{r} - \boldsymbol{f}) \tag{4.25}$$

where $N_1 \leq j \leq N_u - 1$. Therefore, the j prediction of the control signal $\Delta u(k+j)$ is computed using the coefficients of $j+1-th$ row of the matrix $\boldsymbol{K}$. In the case of consecutive dropouts, the maximum number of consecutive dropouts γ_{max} is selected to match N. Thus, the controller can determine the occurrence of consecutive dropouts and apply the past predictions until either the condition is over or γ_{max} has been reached.

To obtain γ_{max} the WNCS is implemented in the simulator and the number of consecutive dropouts is measured for variations of the percentages of dropouts from 25% to 80%. The table 4.1 summarises the number of consecutive dropouts for percentages of

dropouts from sensor to controller d_{psc} and from controller to actuator d_{pca}. A network load disturbance of 0.28, $T_s = 0.01$ and simulation time of 5 s is selected.

TABLE 4.1: Consecutive dropouts length

P_{loss} %	d_{psc}	d_{pca}
25	11	16
30	5	8
40	6	11
50	9	26
60	11	33
70	14	28
73	14	35
75	13	61
80	20	62

The behaviour is found to be mostly random. A maximum value of $\gamma_{max} = 30$ is selected since it covered most of the number of consecutive dropouts.

Using (4.25) the MBPC control law is computed and then compared to the PID control law using (4.18). The optimisation problem minimises the error between the two control signals to find the optimal PID gains.

The function *simulannealbnd* from the Optimization Toolbox is selected to assess the cost function and find the minimum value. The simulation time is $k_f = 5\ s$. The parameters of the optimisation function *simulannealbnd* have been defined the same as in the Section 4.2.5. Once the optimal gains are found, the PID control signal can be implemented using (4.16).

4.3.1 Networked control algorithm

The networked control can be implemented using the following procedure.

Step 1: Initialisation

(a-f) Follow steps a to f from the predictive PID algorithm in Section 4.2.4.

(g) Set the index $m = 0$, as the number of consecutive dropouts.

Step 2: Off-line calculation

(a) Calculate the MBPC gain, $\boldsymbol{K}$, using (4.25).

(b) Follow step b from the predictive PID algorithm in Section 4.2.4.

Step 3: On-line calculation

(a) Read the actual information from the process. In case of dropouts, apply the previous control prediction and increment the index $m = m + 1$.

(b) Compute the MBPC control law and save all control predictions.

(c) Solve the optimisation problem in (4.18) *to find the optimal PID gains.*

(d) Apply the predictive PID control signal.

Step 4: Fine-tuning

(a) Check the closed-loop performance and fine-tune λ *if necessary.*

4.3.2 Simulation results

Consider the nominal example to test the effectiveness of the new predictive algorithm. The same network conditions of the first experiment are applied. As stated before $N = N_u = 30$ and $\lambda = 10$. The optimisation problem is solved after 1535 iterations and 1569 function evaluations per iteration. The optimal steady state PID parameters are: $k_p = 5$, $k_i = 15$ and $k_d = 0.1$.

The output and control signal are shown in Figure 4.4 and the percentage and occurrence of dropouts are depicted in Figure 4.5. A stable response is found. The IAE criterion is $J = 0.48$ for both responses. From the bottom of Figure 4.4, one can see the saved control input values are applied effectively to compensate the occurrence of the consecutive dropouts. Moreover, although there is a switching control input, the control signal is smooth.

Further validations report that the closed-loop response is stable for a maximum percentage of dropouts of 68%. This percentage is the threshold where the closed-loop appears to be stable. This shows the robustness of the controller to compensate a high incidence of dropouts.

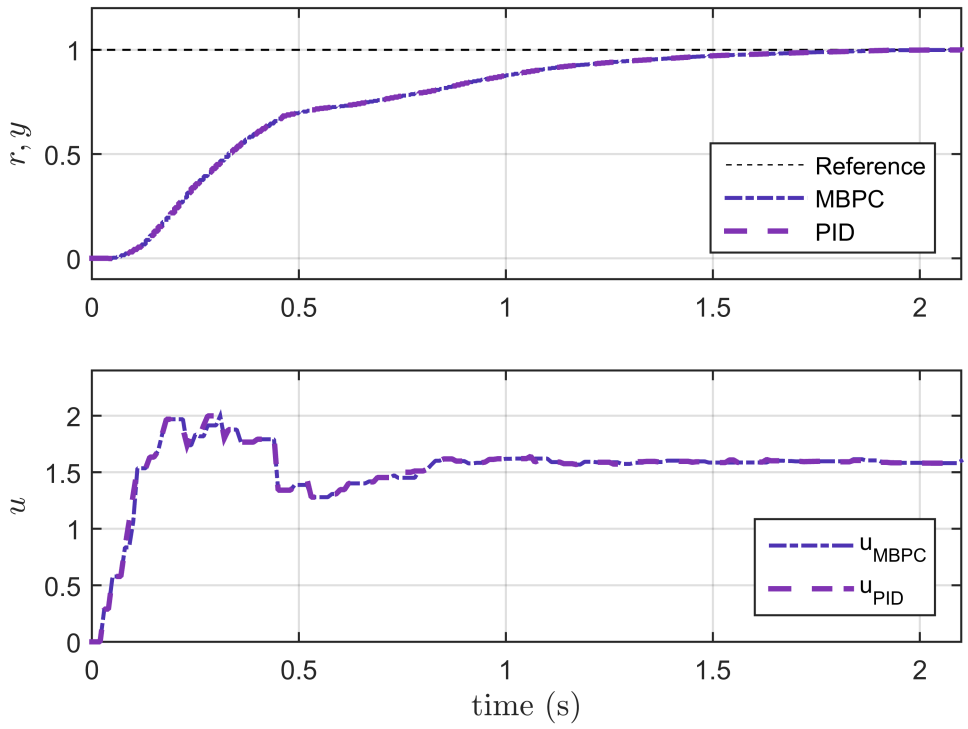

FIGURE 4.4: System outputs of networked control, $\lambda = 10$

Moreover, in comparison to the previous design, the networked controller for consecutive dropouts can compensate higher percentages of dropouts and a higher consecutive occurrence. This is because the controller can apply saved predictive control actions in response to dropouts. However, in case the percentage of dropouts is small, the MBPC with PID structure controller is faster than the networked controller and therefore it will be preferable to apply the first one in that scenario.

4.4 Parallel predictive PID approach

In this section, the predictive PID controller from Katebi (2001) is combined with the KF to compensate the dropouts in the WNCS. A parallel predictive PID controller computes M control signal predictions using M parallel PID controllers (Johnson and Moradi, 2005), each one acting on its respectively time $k + i$. Consider the discrete parallel form of the PID controller given by (4.16) and $\boldsymbol{K} = \boldsymbol{K}_{PID}$. The control signal

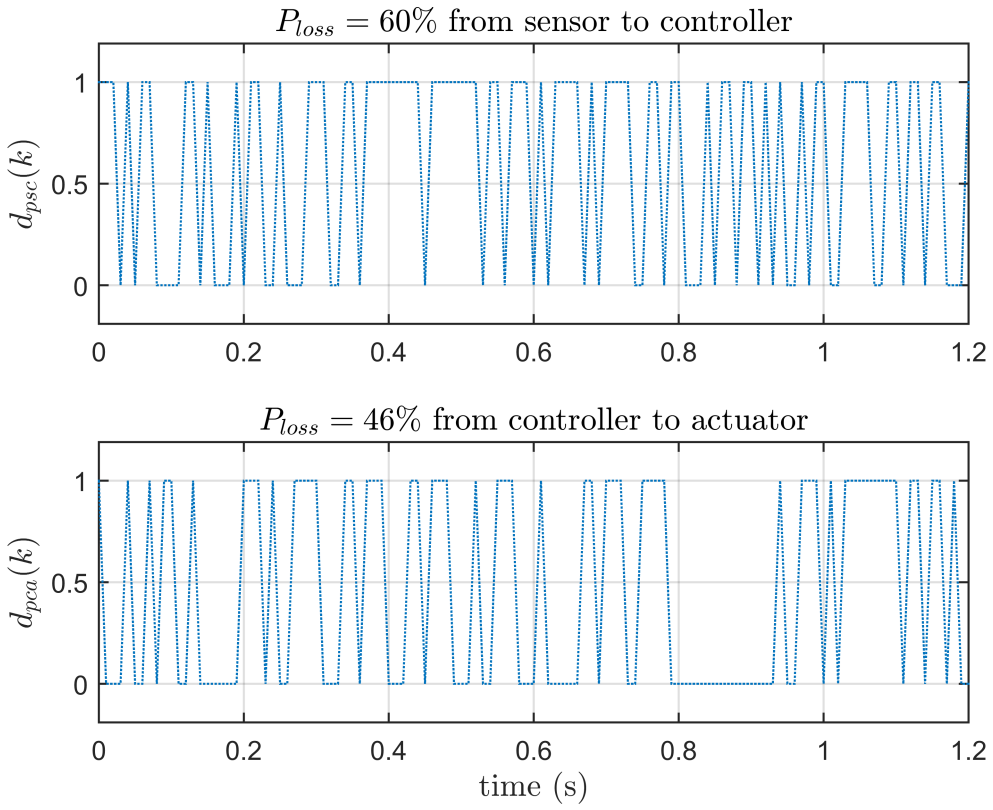

FIGURE 4.5: Time instants of data dropouts for networked control

is the sum of the M PID control signals as follows:

$$\Delta \boldsymbol{u}(k) = \boldsymbol{K} \sum_{i=0}^{M} [\boldsymbol{r}(k+i) - \boldsymbol{y}(k+i)] = \boldsymbol{K} \sum_{i=0}^{M} \boldsymbol{e}(k+i) \tag{4.26}$$

where

$$\begin{aligned} \boldsymbol{e}(k+i) &= [e(k+i-2) \quad e(k+i-1) \quad e(k+i)]^T \\ \boldsymbol{r}(k+i) &= [r(k+i-2) \quad r(k+i-1) \quad r(k+i)]^T = \boldsymbol{r}, \\ \boldsymbol{y}(k+i) &= [y(k+i-2) \quad y(k+i-1) \quad y(k+i)]^T \end{aligned}$$

By following the procedure in Section 4.2.1 and assuming the setpoints and the future control inputs are constant, one can find the error predictions for the ith PID controller as:

$$\boldsymbol{e}(k+i) = \boldsymbol{F}_i \boldsymbol{e}_0 + \boldsymbol{G}_i \Delta \boldsymbol{u}(k) + \boldsymbol{G}_i' \Delta \boldsymbol{u}_0 \tag{4.27}$$

where

$$
\begin{aligned}
&\boldsymbol{e}_0(k) = [e(k)\ e(k-1)\dots\ e(k-n_a)]^T \\
&\Delta\boldsymbol{u}(k) = [\Delta u(k-1)\ \Delta u(k)\ \dots\ \Delta u(k+i-1)]^T \\
&\Delta\boldsymbol{u}_0 = [\Delta u(k)\ \Delta u(k-1)\ \dots\ \Delta u(k-n_b)]^T \\
&\boldsymbol{F}_i = \begin{bmatrix} f_{i1} & f_{i2} & \cdots & f_{i(n_a+1)} \\ f_{(i-1)1} & f_{(i-1)2} & \cdots & f_{(i-1)(n_a+1)} \\ f_{(i-2)1} & f_{(i-2)2} & \cdots & f_{(i-2)(n_a+1)} \end{bmatrix} \\
&\boldsymbol{G}_i = \begin{bmatrix} g_{i-3} & \cdots & g_0 & 0 & 0 \\ g_{i-2} & g_{i-3} & \cdots & g_0 & 0 \\ g_{i-1} & g_{i-2} & \cdots & g_1 & g_0 \end{bmatrix} \\
&\boldsymbol{G}'_i = \begin{bmatrix} g'_{i1} & g'_{i2} & \cdots & g'_{in_b} \\ g'_{(i-1)1} & g'_{(i-1)2} & \cdots & g'_{(i-1)n_b} \\ g'_{(i-2)1} & g'_{(i-2)2} & \cdots & g'_{(i-2)n_b} \end{bmatrix}
\end{aligned}
\tag{4.28}
$$

Substituting the error predictions in (4.26), leads to the following predictive PID control law:

$$
\Delta\boldsymbol{u}(k) = (1 - \boldsymbol{K}\,\boldsymbol{\alpha})^{-1}\boldsymbol{K}\,[\boldsymbol{\beta}_M\ \boldsymbol{e}_0(k) + \boldsymbol{\gamma}_M\ \Delta\boldsymbol{u}_0(k)] \tag{4.29}
$$

where

$$
\boldsymbol{\alpha} = [g_0\ g_0\ g_0], \qquad \boldsymbol{\beta}_M = \sum_{i=0}^{M} \boldsymbol{F}_i, \qquad \boldsymbol{\gamma}_M = \sum_{i=0}^{M} \boldsymbol{G}'_i \tag{4.30}
$$

The vector of PID gains $\boldsymbol{K} = \boldsymbol{K}_{PID}$ is found by solving the optimisation problem in (4.18). The parallel PID control signal (4.29) is compared to the MBPC in (4.14). The minimisation cost criterion is chosen to be the IAE. Thus, the optimal PID parameters are calculated in such way that the error between the MBPC control signal and the predictive PID is almost zero. Once the optimal gains are found, the parallel predictive PID control signal can be implemented using (4.29). Note that, the gains depend on the value of M. This value can be tuned to improve the performance of the controller.

4.4.1 Parallel PID control algorithm

The parallel PID can be implemented using the following procedure.

Step 1: Initialisation

(a-f) Follow steps a to f from the predictive PID algorithm in Section 4.2.4.

Step 2: Off-line calculation

(a-b) Follow steps a to b from the predictive PID algorithm in Section 4.2.4.

(c) Compute the prediction matrices $\boldsymbol{\alpha}$, $\boldsymbol{\beta}_M$ *and* $\boldsymbol{\gamma}_M$.

Step 3: On-line calculation

(a) Read the actual information from the process. In case of dropouts, compute the process estimation using the KF.

(b) Implement the MBPC and parallel predictive PID control laws.

(c) Solve the optimisation problem in (4.18) *to find the optimal PID gains.*

(d) Apply the predictive PID control signal.

Step 4: Fine-tuning

(a) Check the closed-loop performance and fine-tune M *and* λ *if necessary.*

4.4.2 Simulation results

Consider the nominal example given in (4.24). The MBPC prediction horizon is selected as $N = 30$. The closed-loop response is tuned by selecting $\lambda = 10$. Further simulations show that smaller values of λ lead to a larger overshoot and some small variations in the response, which causes a higher cost. The vector of MBPC gains is: $K_{MBPC} = [5.04\ {-5.21}\ 0.46\ 0.06\ 0.06]$. The optimisation problem is solved for $M = 2$ and $P_{loss} = 0$. The initial point is selected as $x_0 = [4\ -4\ 0.4]$, the lower limits as $lb = [0\ 0\ 0]$ and the upper limits as $ub = [10\ 10\ 10]$. The function tolerance is 1×10^{-8}. The number of iterations and function evaluations are 2031 and 2077, respectively. The optimal vector of PID gains is $K = [1.21\ -1.19\ 0.07]$.

Note that, a greater value of M can be used to improve the results. However, increasing the value of M increases the PID gains and that can be a concern for the stability.

Now the system is tested using a $P_{loss} = 30\%$. For the KF design the noise covariances are chosen as: $Q_f = R_f = 1 \times 10^{-8}$. The vector of Kalman gains converges in steady state to the value: $K_f = [0.307\ 0.315]^T$. The system output, the KF estimation and

control signal are shown in Figure 4.6. The plot shows that the filter estimation is very close to the real output of the parallel predictive PID controller, allowing the control system to have a good estimation of the process in case of dropouts. The percentage and occurrence of dropouts are depicted in Figure 4.7.

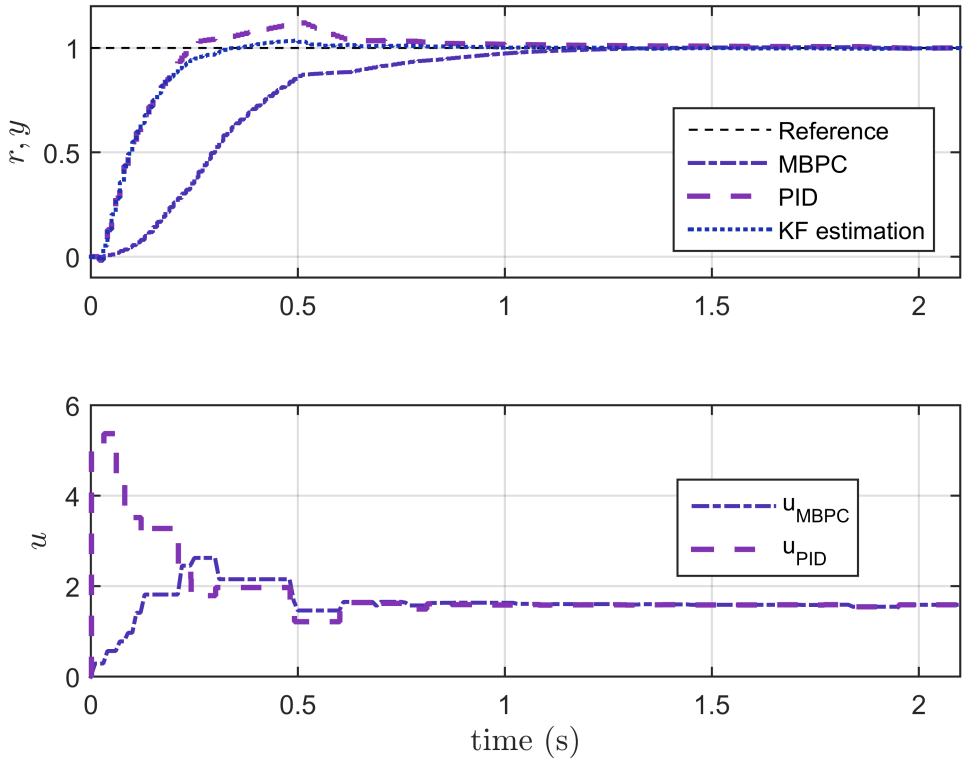

FIGURE 4.6: System outputs of MBPC and parallel predictive PID, $\lambda = 10$

Although the process has a small overshoot, a fast rise time and zero steady error are accomplished. The IAE cost are $J = 0.15$ and $J = 0.35$ for the PID and MBPC, respectively. The formulation of the parallel predictive PID improves the performance of the MBPC. The control system appears stable for a rate of dropouts of up to $P_{loss} = 78\%$. The robustness of the control design for a fast closed-loop response and under the condition of a high percentage of dropouts is superior to other methods studied before.

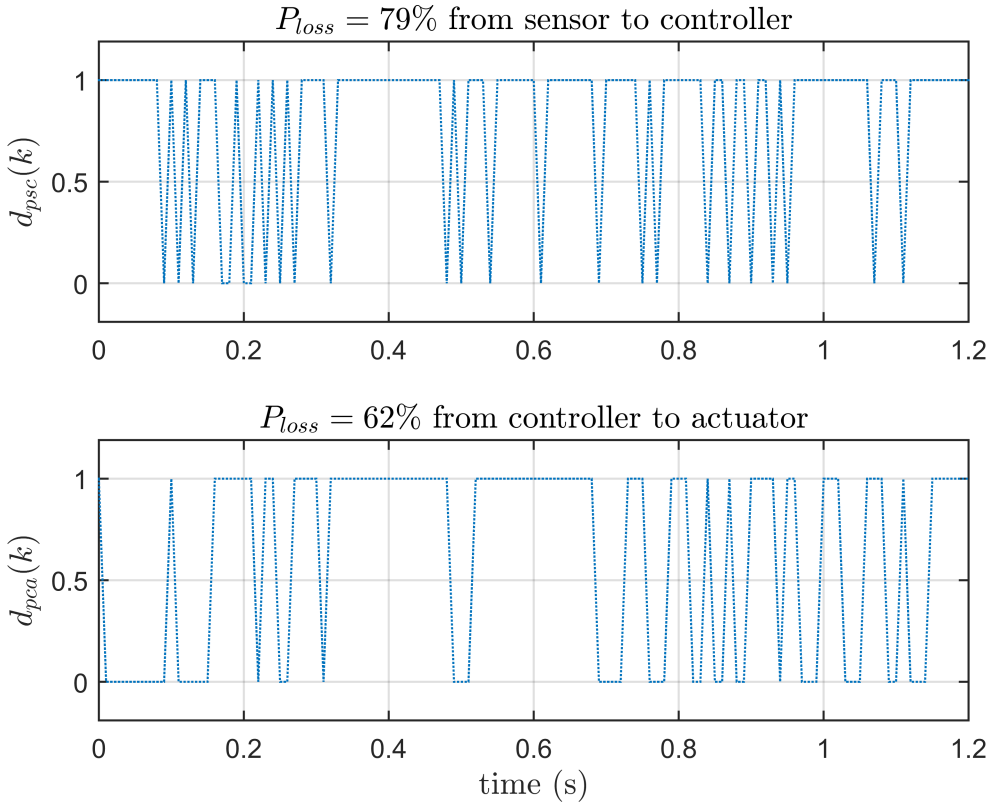

FIGURE 4.7: Time instants of data dropouts

4.5 Predictive control for dropouts with augmented state-space model

A new MBPC with an embedded integrator and PID structure is presented. The KF is added to the control scheme to compensate dropouts. An augmented state-space model is proposed to achieve tracking of the step reference (Wang, 2009). Consider the following state-space system:

$$\begin{aligned} \Delta \boldsymbol{x}(k+1) &= \boldsymbol{A}\Delta \boldsymbol{x}(k) + \boldsymbol{B}\Delta u(k) \\ y(k+1) - y(k) &= \boldsymbol{CA}\Delta \boldsymbol{x}(k) + \boldsymbol{CB}\Delta u(k) \end{aligned} \tag{4.31}$$

where $\Delta \boldsymbol{x}(k+1) = \boldsymbol{x}(k+1) - \boldsymbol{x}(k)$, $\Delta \boldsymbol{x}(k) = \boldsymbol{x(k)} - \boldsymbol{x}(k-1)$ and $\Delta u(k) = u(k) - u(k-1)$. The augmented model is built by connecting the output to the difference of the state variables, which is necessary to obtain the integrating effect in the design. Thus, the

new state vector is given as:

$$\boldsymbol{x}(k) = [\Delta \boldsymbol{x}(k)^T \quad y(k)]^T \tag{4.32}$$

Now, the augmented state-space system is defined as:

$$\begin{aligned} \begin{bmatrix} \Delta \boldsymbol{x}(k+1) \\ y(k+1) \end{bmatrix} &= \begin{bmatrix} \boldsymbol{A} & \boldsymbol{O}^T \\ \boldsymbol{CA} & 1 \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{x}(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{CB} \end{bmatrix} \Delta u(k) \\ y(k) &= \begin{bmatrix} \boldsymbol{O} & 1 \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{x(k)} \\ y(k) \end{bmatrix} \end{aligned} \tag{4.33}$$

where $\boldsymbol{O} = [0 \quad 0]$.

4.5.1 Recursive matrices for augmented state-space model

The prediction matrices are calculated using recursion (Rossiter, 2003, pp. 31-33). Replacing one step-ahead prediction recursively, the N ahead predictions are given by:

$$\begin{aligned} \boldsymbol{x}(k+N) &= \boldsymbol{A}^N \boldsymbol{x}(k) + \boldsymbol{A}^{N-N_u} \boldsymbol{B} \Delta u(k+N_u-1) \\ y(k+N) &= \boldsymbol{CA}^N \boldsymbol{x}(k) + \cdots + \boldsymbol{CA}^{N-N_u} \boldsymbol{B} \Delta u(k+N_u-1) \end{aligned} \tag{4.34}$$

Then, the notation can be reduced as:

$$\boldsymbol{y} = \boldsymbol{P}\boldsymbol{x}(k) + \boldsymbol{H}\boldsymbol{u} \tag{4.35}$$

where $\boldsymbol{y}$ and $\boldsymbol{u}$ have been defined in (4.10) for $d = 0$, $\boldsymbol{P}$ is the matrix that depends on past values of $\boldsymbol{x(k)}$ and $\boldsymbol{H}$ depends on the predictions of the input $\Delta u(k)$.

4.5.2 Predictive PID structure

A new cost function J can be rewritten by replacing (4.35) in (4.1). The minimisation of J results in the following optimal control law:

$$\Delta u(k) = \boldsymbol{K}[\boldsymbol{r} - \boldsymbol{P}\boldsymbol{x}(k)] \tag{4.36}$$

where

$$\boldsymbol{K} = (\boldsymbol{H}^T \boldsymbol{H} + \lambda \boldsymbol{I})^{-1} \boldsymbol{H}^T \tag{4.37}$$

Moreover, $\Delta u(k)$ can be reformulated as:

$$\Delta u(k) = \boldsymbol{K}\, \boldsymbol{e}(k) \tag{4.38}$$

where $\boldsymbol{e}(k) = \boldsymbol{r}$ - $\boldsymbol{P}\boldsymbol{x}(k)$.

The PID gains can be found by solving the optimisation problem (4.18). The error between the MBPC control law given by (4.38) and the PID one (4.16) is minimised. The cost function has been defined to compute the optimal PID parameters in such way that the error between the MBPC control signal and the predictive PID is almost zero. The function *simulannealbnd* is selected since it solves quickly the optimisation problem subject to the bounds of the PID gains. Once the optimal gains are found, the PID control signal can be implemented using (4.16).

4.5.3 Predictive PID control with augmented state-space algorithm

The predictive PID control with augmented state-space can be implemented using the following procedure.

Step 1: Initialisation

(a) Set up the discrete state-space model of the process and compute the augmented state-space model.

(b-f) Follow steps b to f from the predictive PID algorithm in Section 4.2.4.

Step 2: Off-line calculation

(a) Compute the prediction matrices $\boldsymbol{P}$ *and* $\boldsymbol{H}$*.*

(b) Calculate the MBPC gain, $\boldsymbol{K}$*, using* (4.37)*.*

Step 3: On-line calculation

(a) Read the actual information from the process. In case of dropouts, compute the process estimation using the KF.

(b) Implement the MBPC and PID control laws.

(c) Solve the optimisation problem in (4.18) *to find the optimal PID gains.*

(d) Apply the predictive PID control signal.

Step 4: Fine-tuning

(a) Check the closed-loop performance and fine-tune λ *if necessary.*

4.5.4 Simulation results

The augmented state-space model of the nominal example in (4.24) is calculated using (4.33). The prediction horizon $N = 20$, control horizon $N_u = 3$ and $\lambda = 0.1$. For the KF design the noise covariances are chosen as: $Q_f = R_f = 1 \times 10^{-8}$. The vector of Kalman gains converges in steady state to the value: $K_f = [0.3071\ 0.315]^T$. The optimal PID gains are computed using (4.18) and the parameters of the optimisation function *simulannealbnd* have been defined the same as in the Section 4.2.5. The number of iterations is 1801 and the number of function evaluations is 1841. The optimal PID gains are: $k_p = 15.21$, $k_i = 19.79$ and $k_d = 16.80$.

The predictive responses are shown in Figure 4.8 and the percentage and occurrence of dropouts are depicted in Figure 4.9. The estimation of the real process is achieved with the implementation of the KF. The IAE is $J = 0.12$ for the MBPC and the predictive PID responses. Note the influence of the weighting value λ that has been selected as 0.1. The new controller gives lower tracking error with a smaller weighting factor. Some tests varying the control horizon indicate that too long prediction could deteriorate the performance and bigger overshoots are found. This is reasonable since the errors in the prediction are bigger for long prediction horizon. The percentage of dropouts from the sensor to controller and from the controller to actuator is 65%. Therefore, using lower values of λ the method can effectively control the loss of information when the rate of dropouts is very high, while a good performance is accomplished.

4.6 Comparison

Four new predictive control algorithms have been presented to compensate dropouts in WNCS. Step responses of the four approaches are compared in Figure 4.10. The performance has been assessed using the IAE criterion. The results are summarised

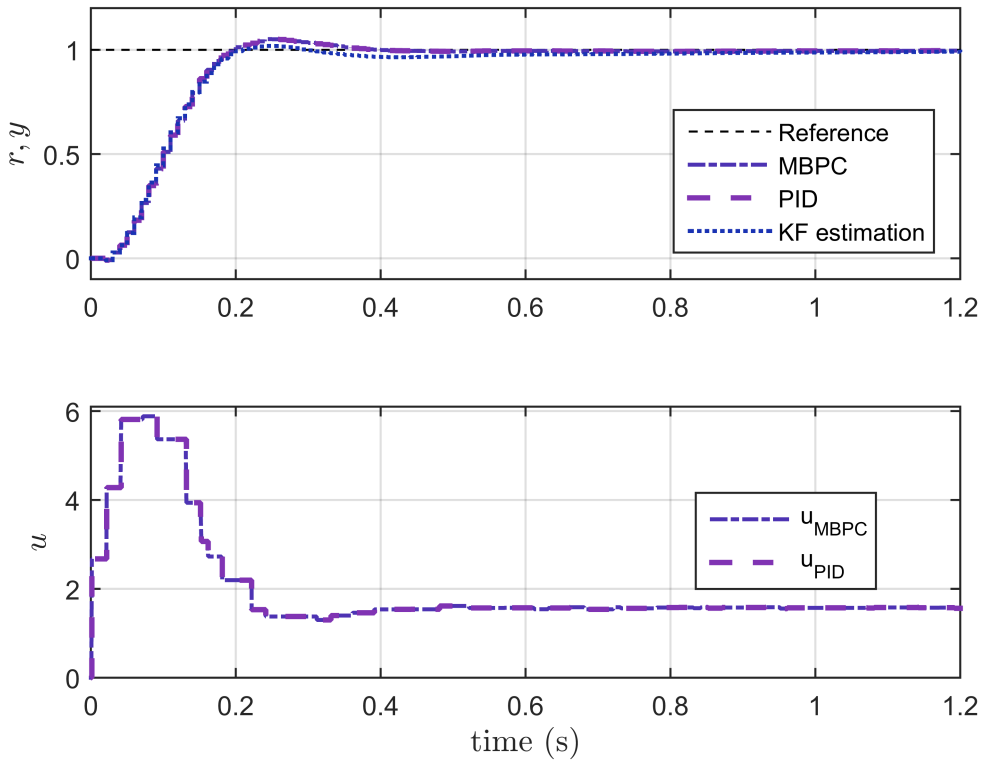

FIGURE 4.8: System outputs of augmented model for MBPC and PID, $\lambda = 0.1$

in the Table 4.2. A percentage of dropouts $P_{loss} = 30\%$ and $\lambda = 10$ is selected. The time instant of the dropouts is presented in Figure 4.3. According to the results, when

TABLE 4.2: Summary of controllers performance

Controller	IAE
MBPC PID	0.23526
MBPC PID consecutive dropouts	0.31681
MBPC Parallel PID	0.15486
MBPC Augmented PID	0.10028

the dropout percentage is low, the predictive PID based on MBPC (dash-dotted line) performs better than the networked controller for consecutive dropouts (dotted line). However, further experiments demonstrate that the latter deals with the higher dropouts and a higher consecutive occurrence much better. This is because the controller can apply saved predictive control actions in response to dropouts. In addition, the time response can be manipulated using the MBPC tuning parameter λ.

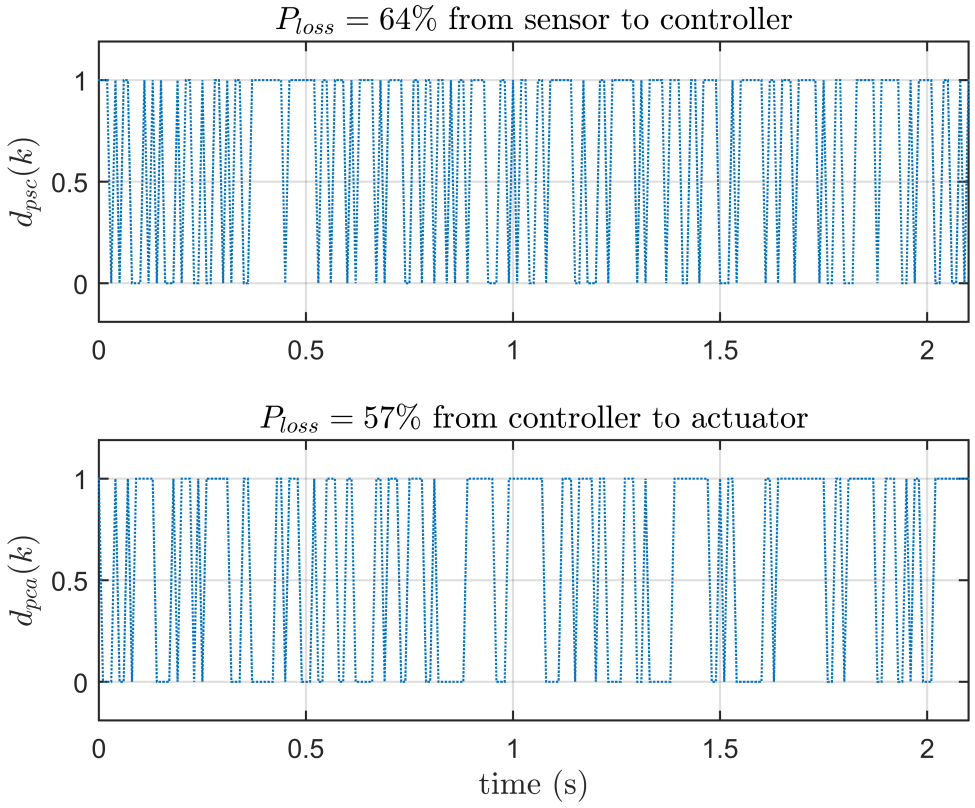

FIGURE 4.9: Time instants of data dropouts

Parallel predictive PID control (dashed line) gives better performance than MBPC with PID structure for low and high percentages of dropouts. As explained before, the predictive PID controller is designed in such a way that the control performance is close to MBPC. In addition, the M value can be used to tune the response. For $M = 2$ and $\lambda = 10$ the WNCS is stable up to $P_{loss} = 78\%$, which is superior to the previous strategies.

The predictive PID control with an augmented state-space model produces a very efficient controller with a great response, being the one that has the best performance. The integral action has been included in the model and allows to have the minimum performance index under the presence of the dropouts. The superiority of the proposed approach has been indicated in comparison with the other methods using a very lossy wireless network.

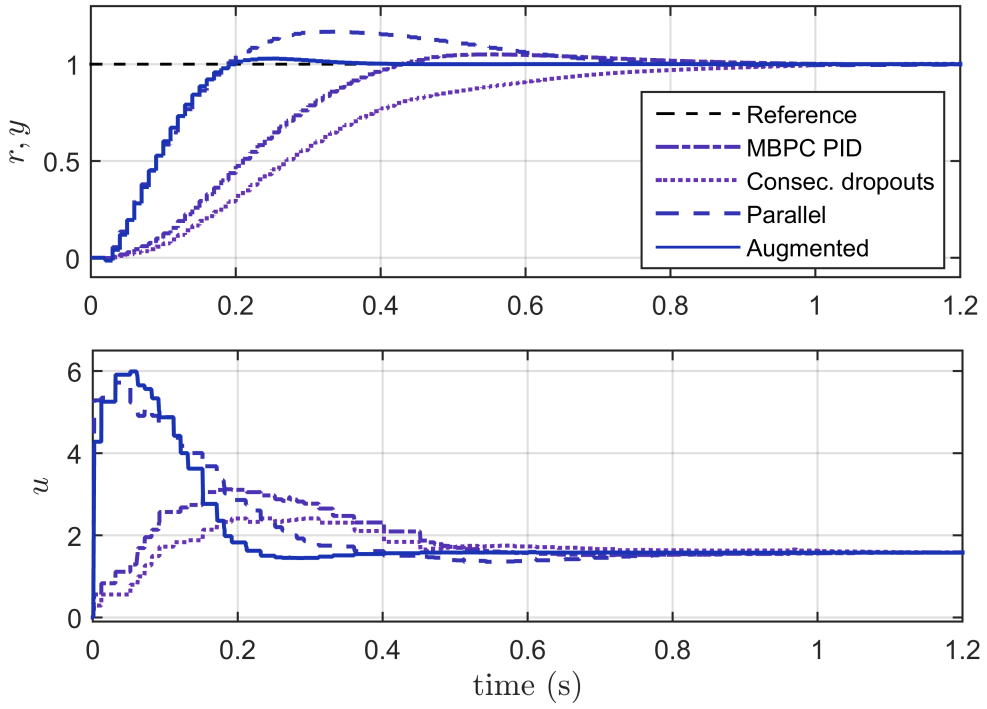

FIGURE 4.10: Step responses comparison, $\lambda = 10$

4.7 Constrained Predictive PID control for packet dropouts in WNCS

In this section, a new constrained predictive PID controller with similar properties to MBPC is developed to compensate dropouts in WNCS. A quadratic programming problem optimises a MBPC cost function to find the optimal PID gains at every sampling time. The constraint handling is presented to stop input saturation. The problem of the occurrence of dropouts from sensor to controller is compensated by combining the controller with a Kalman filter. The measured output $y(k)$ is replaced to the Kalman estimation $\hat{y}(k)$ allowing the controller to have always information on the process even in the presence of dropouts. To compensate consecutive dropouts from controller to actuator, predictions of the control signal are calculated and saved in the actuator for the next sampling instant. The method is applied to typical second order and non-minimum phase systems with delays.

4.7.1 Constrained predictive PID implementation

The proposed control scheme is presented by the block diagram depicted in Figure 4.11. The control signal is u, y stands for process output, e is the error and r is the reference signal. The controlled plant is $G_p(z)$. The proposed framework is for a class of linear, discrete-time, constrained process. A WNCS whose sensor and control information is transported over a wireless network is considered. The dropouts from sensor to controller and from controller to actuator are represented as d_{psc} and d_{pca}, respectively. A quadratic programming problem optimises a MBPC criterion to find the optimal PID gains at every sampling time. The constraint handling is presented to stop input saturation.

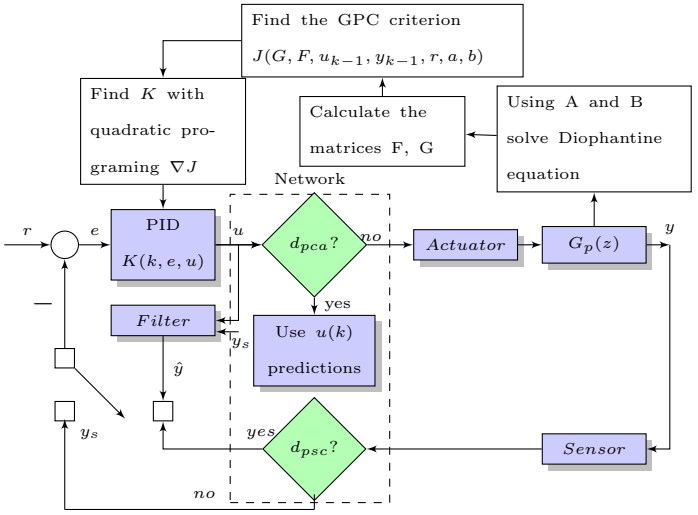

FIGURE 4.11: Diagram of constrained predictive PID controller structure

The measured output y_s is switched to the KF estimation $\hat{y}$ allowing the controller to have always information of the process even in the presence of dropouts. Predictions of the control signal are calculated and applied accordingly to compensate consecutive dropouts from controller to actuator.

4.7.2 The MBPC reduced criterion

Consider the MBPC cost function described by (4.1). By selecting the control horizon $N_u = 1$, a GPC reduced criterion is presented as follows:

$$J = \sum_{j=N_1}^{N_{ph}} [\hat{y}(k+j) - r(k+j)]^2 + \lambda[\Delta u(k)]^2 \tag{4.39}$$

The design of the predictive PID controller requires that the control horizon is selected as one since the PID control law only computes $\Delta u(k)$. Now, the procedure stated in section 4.2.1 is used to find the prediction matrices. The output predictions found in (4.9) can be rewritten as:

$$\hat{y}(k+j) = F_j(q^{-1})y(k) + G_j(q^{-1})\Delta u(k+j-d-1) \tag{4.40}$$

Applying the last result to (4.39) the cost function J is formulated as the following quadratic problem:

$$J = (\mathbf{Gu} + \mathbf{Fy} - \mathbf{r})^{\mathbf{T}}(\mathbf{Gu} + \mathbf{Fy} - \mathbf{r}) + \lambda \boldsymbol{u}^{\mathbf{T}}\boldsymbol{u} \tag{4.41}$$

where

$$\begin{aligned}
\boldsymbol{r} &= [r(k+1) \quad r(k+2) \quad \cdots \quad r(k+N)]^T \\
\mathbf{y} &= [\hat{y}(k+1) \quad \hat{y}(k+2) \quad \cdots \quad \hat{y}(k+N)]^T \\
\mathbf{u} &= [\Delta u(k) \quad \Delta u(k+1) \quad \cdots \quad \Delta u(k+N_u-1)]^T \\
\boldsymbol{F}(q^{-1}) &= \begin{bmatrix} F_{d+1}(q^{-1}) \\ F_{d+2}(q^{-1}) \\ \vdots \\ F_{d+N}(q^{-1}) \end{bmatrix}, \quad \boldsymbol{G} = \begin{bmatrix} g_0 & 0 & \cdots & 0 \\ g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ g_{N-1} & g_{N-2} & \cdots & g_0 \end{bmatrix}
\end{aligned}$$

For simplicity of notation, it is assumed that $d = 0$ in the equations above. The quadratic cost function is minimised by solving $\nabla J = 0$. Note that the optimal input solution $\Delta u(k+j-1) = 0$ for $j > 1$. The control horizon has been selected as one because the PID law only computes $\Delta u(k)$.

4.7.3 The design of the Predictive PID controller

4.7.4 The PID

The velocity form of the PID controller in (4.16) is considered and the vector of gains $\boldsymbol{K}$ is defined as:

$$\boldsymbol{K} = \begin{bmatrix} k_p + k_i\,Ts + \dfrac{k_d}{Ts} & -k_p - 2\dfrac{k_d}{Ts} & \dfrac{k_d}{Ts} \end{bmatrix}^T = [k_1 \; k_2 \; k_3]^T \tag{4.42}$$

Then, the controller law is defined as follows:

$$\Delta u(k) = \mathbf{K}^T \boldsymbol{e}(k) \tag{4.43}$$

where $\boldsymbol{e}(k)$ is the vector of control errors:

$$\boldsymbol{e}(k) = [e(k) \quad e(k-1) \quad e(k-2)]^T \tag{4.44}$$

The PID controller gains must be positive scalars: $k_p > 0,\ k_i > 0,\ k_d > 0$. Therefore, it is easy to see that the vector of gains $\boldsymbol{K}$ must fulfil the linear inequality constraints:

$$k_1 + k_2 + k_3 > 0, \quad k_2 + 2k_3 < 0, \quad k_3 \geq 0 \tag{4.45}$$

4.7.5 The predictive PID controller

The PID predictive controller is obtained by combining the MBPC and PID control laws. The purpose of the design is to compute the PID gains in such a way that the control signal is as close as possible to the MBPC signal. First, by simplifying (4.41) yields:

$$J(K) = \boldsymbol{u}^T(\boldsymbol{G}^T\boldsymbol{G} + \lambda\boldsymbol{I})\boldsymbol{u} + \boldsymbol{u}^T 2\boldsymbol{G}^T(\boldsymbol{F}\ \boldsymbol{y}\ \text{-}\ \boldsymbol{r}\ ) \tag{4.46}$$

Replacing Δu from (4.43) leads to:

$$J(K) = [\mathbf{e(k)}^\mathbf{T}\boldsymbol{K}]^\mathbf{T}(\boldsymbol{G}^\mathbf{T}\mathbf{G} + \lambda\boldsymbol{I})\mathbf{e(k)}^\mathbf{T}\boldsymbol{K} + [\mathbf{e(k)}^\mathbf{T}\boldsymbol{K}]^\mathbf{T}\mathbf{2}\boldsymbol{G}^\mathbf{T}(\boldsymbol{F}\ \boldsymbol{y}\ \text{-}\ \boldsymbol{r}\ ) \tag{4.47}$$

This is equivalent to:

$$J(K) = \boldsymbol{K}^\mathbf{T}\ \boldsymbol{e}(\mathbf{k})\,(\boldsymbol{G}^\mathbf{T}\boldsymbol{G} + \lambda\boldsymbol{I})\boldsymbol{e}(\mathbf{k})^\mathbf{T}\ \boldsymbol{K} + \boldsymbol{K}^\mathbf{T}\ \mathbf{2e(k)}\ \boldsymbol{G}^\mathbf{T}(\boldsymbol{F}\ \boldsymbol{y}\ \text{-}\ \boldsymbol{r}\ ) \tag{4.48}$$

The new algorithm will be carried out by minimising the cost function respect to the PID controller gains $\boldsymbol{K}$:

$$\Delta\boldsymbol{u}_{cons} = \min_{\boldsymbol{K}} J(\boldsymbol{K}, \boldsymbol{y}) \tag{4.49}$$

where $\Delta \boldsymbol{u}_{cons}$ is the constrained optimal input at time instant k. It follows directly from (4.48) that the PID gains can be found by solving the following quadratic program:

$$\begin{aligned} \min_{\boldsymbol{K}} \quad & \frac{1}{2}\, \boldsymbol{K}^{\mathrm{T}}\, \boldsymbol{H}\, \boldsymbol{K} + \boldsymbol{f}^{\mathrm{T}}\, \boldsymbol{K} \\ s.t. \quad & \boldsymbol{a}(k)\, \boldsymbol{K} \leq \boldsymbol{b}(k) \end{aligned} \tag{4.50}$$

where

$$\begin{aligned} \boldsymbol{H} &= 2(\boldsymbol{G}^T \boldsymbol{G} + \lambda \boldsymbol{I}) \boldsymbol{e}(k) \boldsymbol{e}(k)^T \\ \boldsymbol{f} &= 2 \boldsymbol{G}^T (\boldsymbol{F}\ \boldsymbol{y}\ \boldsymbol{-}\ \boldsymbol{r}\ ) \boldsymbol{e}(k) \end{aligned} \tag{4.51}$$

The constraints of (4.50) will guarantee the contributions of control input and rate input are applied according to the controller limitations. $\boldsymbol{a}(k)$ depends on the past values of the error and $\boldsymbol{b}(k)$ on the upper and lower limits on the control input and rate input. The design is extended in the next section. The control law can be rewritten from (4.43) as:

$$\Delta \boldsymbol{u}(k) = \boldsymbol{K}(k) \boldsymbol{e}(k) \tag{4.52}$$

The optimisation problem has been set using the command $quadprog(H, f, a, b)$, where $\boldsymbol{H, f}$ have been stated in Equation (4.51) and $\boldsymbol{a, b}$ are the constraint matrices of the linear inequality. The *interior-point-convex* algorithm is used. The solver tries to find the optimal point based on the Karush-Kuhn-Tucker (KKT) conditions, where the gradient must be zero at the minimum and take constraints into account.

It is important to stress that the vector of PID gains will change at every time instant k. As a consequence, the proposed predictive PID is time-varying and it will be optimised for a bounded percentage of dropouts.

The advantage of the predictive PID is that it improves the traditional PID performance by equating its control signal with the (MBPC) control signal. Since the gains are varying every sampling time, the performance of the PID controller is as good as the MBPC controller. Moreover, the control signal along the input steps changes smoothly, as demonstrated in Section 4.7.10.

4.7.6 Constraints for the control input and control input increment

Here the constraints to select the appropriate predictive PID gains to prevent input saturation are formulated. To introduce the constraint handling, the predictive PID

control subject to linear constraints for the input and input increment is solved:

$$\begin{aligned} -\Delta u_{min} \leq\ & \Delta u(k) \leq \Delta u_{max} \\ -u_{min} \leq\ & u(k) \leq u_{max} \end{aligned} \tag{4.53}$$

Using (4.42) the predictive PID control law in equation (4.52) can be defined as:

$$\Delta u(k) = k_1\, e(k) + k_2\, e(k-1) + k_3\, e(k-2) \tag{4.54}$$

By combining the previous result and $\Delta u(k) = u(k) - u(k-1)$ in equation (4.54) the constraints can be written as:

$$\begin{aligned} &-\Delta u_{min} \leq\ k_1\, e(k) + k_2\, e(k-1) + k_3\, e(k-2) \leq \Delta u_{max} \\ &-u_{min} - u(k-1) \leq\ k_1\, e(k) + k_2\, e(k-1) + k_3\, e(k-2) \\ &\leq u_{max} - u(k-1) \end{aligned} \tag{4.55}$$

The inequalities in (4.55) can be separated and a matrix arrangement is obtained. Moreover, by combining equation (4.45) with the previous result the final constraint matrix is found:

$$\begin{bmatrix} 0 & 1 & 2 \\ 0 & 0 & -1 \\ -1 & -1 & -1 \\ e(k) & e(k-1) & e(k-2) \\ -e(k) & -e(k-1) & -e(k-2) \\ e(k) & e(k-1) & e(k-2) \\ -e(k) & -e(k-1) & -e(k-2) \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} \leq \begin{bmatrix} -\epsilon \\ 0 \\ -\epsilon \\ \Delta u_{max} \\ -\Delta u_{min} \\ u_{max} - u(k-1) \\ -u_{min} + u(k-1) \end{bmatrix} \tag{4.56}$$

Note that, the constraints should be fulfilled for every $\Delta u(k)_j,\ j = 1, \ldots, N-1$. The final constraint matrix in equation (4.56) has the form $\boldsymbol{a}(k)\boldsymbol{K} \leq \boldsymbol{b}(k)$ previously defined in the optimisation problem proposed in (4.50).

4.7.7 Dropouts from controller to actuator compensation

To compensate for dropouts the from controller to the actuator, the predictions of the control signal are calculated. First, from (4.51) the matrix G_1 is computed instead of

G:

$$\mathbf{G1} = \mathbf{G}(\mathbf{1} : \mathbf{N}, \mathbf{j}) \tag{4.57}$$

where j stands for columns of matrix $\boldsymbol{G}$ and $N_1 \leq j \leq N$. Therefore, the quadratic program computes N predictions of the control signal $\Delta u(k)$ using the coefficients of $j-th$ column of matrix $\boldsymbol{G}$. During a successful transmission from controller to actuator, the controller inputs are saved in the actuator for the next sampling instant. At time k, a dropouts detector at the actuator location indicates if the new control signal is not received and applies the next prediction $u_j(k)$ to compensate the dropout. If there are not saved predictions, the actuator arbitrarily applies the initial condition $u_0 = 0$. Note, that it has been assumed that some computational and buffering resources are available at the actuator.

Notably, in the proposed algorithm the controller has no knowledge of the control input that the actuator applies. However, this is not a limitation since it has been demonstrated that acknowledgements from the actuator to the controller do not improve the stability of the networked predictive control (see Gupta and Martins (2010) and the references therein).

In the case of consecutive dropouts, the maximum number of consecutive dropouts γ_{max} is selected to match the prediction horizon N. Thus, the 'smart" actuator can determine the occurrence of consecutive dropouts and apply the past predictions until either the condition is over or γ_{max} has been reached. For this end, a consecutive dropouts detector has been created. It consists of an index, m, that counts the number of consecutive dropouts and it is reset every time the information is available. To obtain γ_{max} the WNCS is implemented in the simulator and the number of consecutive dropouts is measured for variations of the percentages of dropouts from 25% to 80%. A maximum value of $\gamma_{max} = 30$ is selected since it covered most of the number of consecutive dropouts.

4.7.8 Dropouts from sensor to controller compensation

The Kalman filter estimation proposed in 4.2.3 is extended to handle the delay in the system using a smoothed estimation. As mentioned in Section 4.2, the delay has been included in the predictive algorithm. Therefore, the estimation $\hat{y}(k)$ should include the delay as well. An estimation of $x(k)$ can be obtained by smoothed estimates for lags

up to d samples, that is:

$$\hat{x}^d(k) = \mathbb{E}\left\{x(k-d)|y(0),\cdots,y(k-1)\right\} \tag{4.58}$$

This is called a smoothed estimate. The accuracy increases when more measurements are used to estimate the state. However, the greater the delay, the greater the complexity of the estimator. A fixed-lag smoothing problem with a fixed delay d has been implemented to address the delay in the system. It can be derived by augmenting the state vector by the delayed versions of the state as follows:

$$\begin{bmatrix} x(k+1) \\ x^1(k+1) \\ \vdots \\ x^d(k+1) \end{bmatrix} = \begin{bmatrix} A & 0 & \cdots & 0 \\ I & 0 & \cdots & \vdots \\ 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & I & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ x^1(k) \\ \vdots \\ \vdots \\ x^d(k) \end{bmatrix} + \begin{bmatrix} I \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix} \upsilon(k) \tag{4.59}$$

$$y(k) = \begin{bmatrix} C & 0 & \ldots & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ x^1(k) \\ \vdots \\ \vdots \\ x^d(k) \end{bmatrix} + \omega(k) \tag{4.60}$$

The filter for the composite state has the following form:

$$\begin{bmatrix} \hat{x}(k+1) \\ \hat{x}^1(k+1) \\ \vdots \\ \hat{x}^d(k+1) \end{bmatrix} = \begin{bmatrix} A & 0 & \cdots & 0 \\ I & 0 & \cdots & \vdots \\ 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & I & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ x^1(k) \\ \vdots \\ \vdots \\ x^d(k) \end{bmatrix} + \begin{bmatrix} K(k) \\ K^1(k) \\ \vdots \\ \vdots \\ K^d(k) \end{bmatrix} [y(k) - C\hat{x}(k)] \tag{4.61}$$

where the gains $[K(k),\cdots,K^d(k)]$ are obtained from the standard filter for the composite state.

The examples in Section 4.7.10 help to illustrate the implementation of the proposed scheme that joins the Kalman estimator with the previous predictive PID control.

4.7.9 Constrained predictive PID control algorithm

The constrained predictive PID can be implemented using the following procedure.

Step 1: Initialisation

(a) Set up the model parameters of the process.

(b) Select the prediction horizon N and the future reference vector.

(c) Compute the prediction matrices $\boldsymbol{F}$, $\boldsymbol{G}$.

(d-g) Follow steps c to f from the predictive PID algorithm in Section 4.2.4.

(h) Set the index $m = 0$, as the number of consecutive dropouts.

Step 2: Off-line calculation.

(a) Implement the predictive PID control law and the KF.

(b) Compute the time-invariant part of the constraint matrices.

Step 3: On-line calculation

(a) Read the actual information from the process. In case of dropouts, compute the process estimation using the KF.

(b) Compute the predictive PID control law and send all control predictions to the actuator.

(c) The actuator applies the present value of the control signal to the process. In case of dropouts, it applies the previous control prediction and increments the index $m = m + 1$.

Step 4: Fine-tuning

(a) Check the closed-loop performance and fine-tune Q_f, R_f and λ if necessary.

4.7.10 Simulation studies

4.7.11 Numerical example 1: Second order process

The performance of the predictive PID control is investigated using simulation studies. The proposed algorithm has been implemented and tested using the TrueTime network simulator configured for wireless protocol 802.11b (WLAN) as explained in Section 4.2.5. The results have been compared with the solutions obtained by the classical MBPC with constraints.

Consider the following system with a sampling time $T_s = 0.01\ s$:

$$G_p(z) = \frac{0.003319(z + 0.5215)}{(z - 0.9755)(z - 0.09748)} \tag{4.62}$$

Although different values of the penalty in the control action can be selected, $\lambda = 0.5$ is chosen for a faster response of the closed-loop. As explained before, the prediction horizon is $N = 30$ and the control horizon $N_u = 1$. Control input constraints have been assumed as $u_{max} = 3$, $u_{min} = -3$ and the rate input $\Delta u_{max} = 10$. For the KF design the noise covariances are chosen as: $Q_f = R_f = 1 \times 10^{-8}$. The vector of Kalman gains converges in steady state to the value: $K_f = [0.12\ 0.13]^T$.

The optimisation problem has been set using the command *quadprog*. Lower and upper bounds of zero and ten, respectively have been selected to find positive and finite PID gains. Note, that these values can be adjusted according to the application. The maximum number of iterations is 200. The optimality tolerance is 1×10^{-8}. Similarly, the tolerance on the constraint violation has been set as 1×10^{-8}. The maximum number of iterations is six.

The optimal vector of PID parameters $\boldsymbol{K}$ is found and compared to equation (4.42) to find the optimal gains of the predictive PID every sampling time.

Figure 4.12 shows the system outputs and constrained controller inputs of the predictive PID and MBPC for a sin wave reference signal (black dashed line). The reference tracking is achieved (top) and the control signal (bottom) satisfies the constraints (red dashed line). Moreover, the percentage and occurrence of dropouts for the simulation are $P_{loss} = 2\%$ from sensor to controller and $P_{loss} = 4\%$ from controller to actuator. A low percentage of dropouts is chosen to test the tracking response.

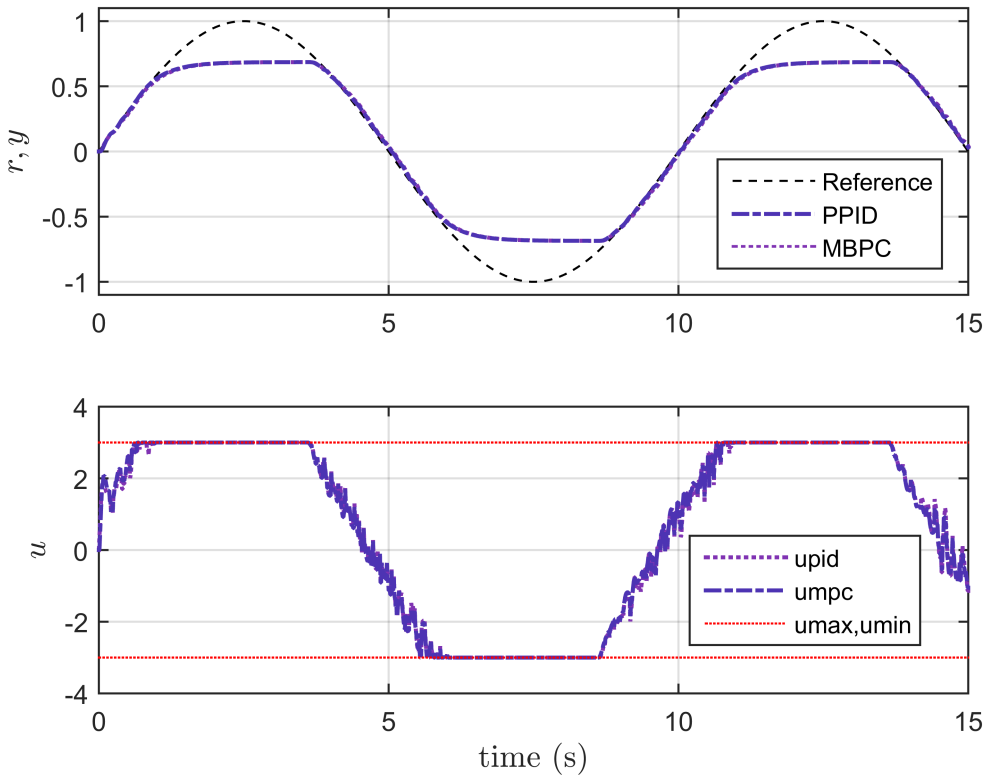

FIGURE 4.12: System outputs and control inputs for constrained predictive PID

The predictive PID (dash-dotted line) shows almost the same behaviour as the MBPC (dotted line). The performance of the controllers has been assessed using the Integral of Absolute Error (IAE). The criterion returned a value of $J = 1.8636$ for the predictive PID response which is better than the MBPC: $J = 1.9293$. Although minor oscillations are found in the signals, the controller works within the requirements even with the presence of dropouts.

Figure 4.13 compares the real output with the Kalman filter estimation. The plot shows that the filter output has the same behaviour than the real process. The error is almost negligible as a result of the small covariance noises applied in the design.

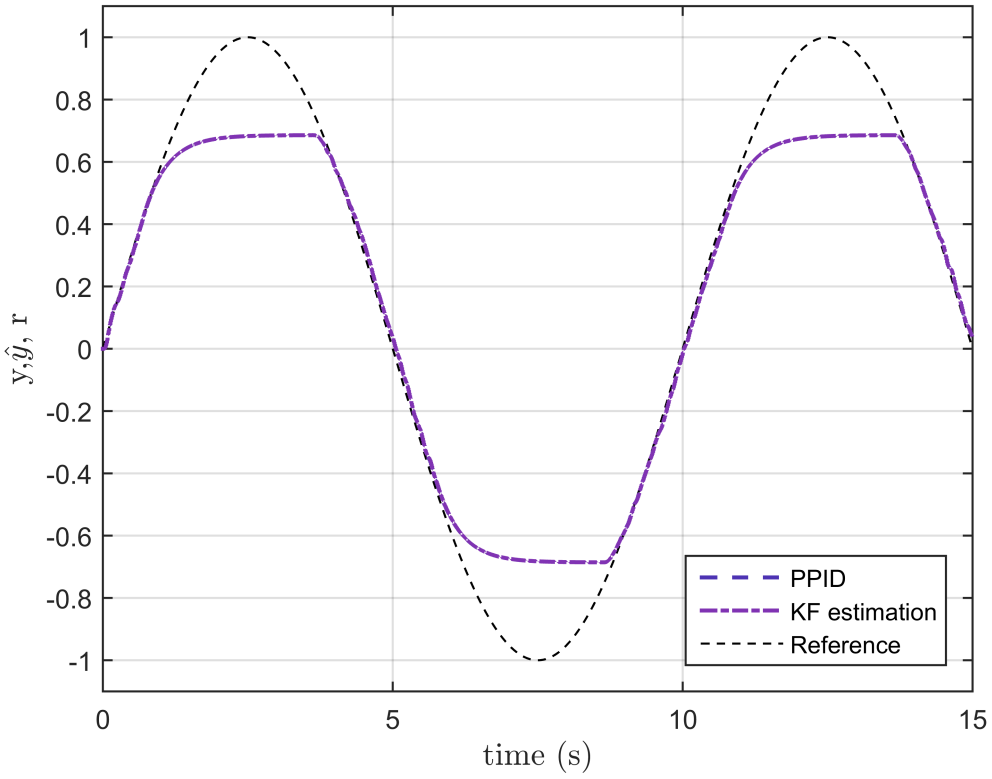

FIGURE 4.13: Comparison between real process and estimation for constrained predictive PID

4.7.12 Numerical example 2: Non-minimum phase process

Consider the following non-minimum phase process with dead time and sampling time $T_s = 1\ s$:

$$G_p(z) = \frac{-0.26785\,(z - 1.292)\,z^{-3}}{(z - 0.6065)(z - 0.006738)} \tag{4.63}$$

The network parameters and the optimisation Toolbox are set as in Example 1. As explained before, the prediction horizon is $N = 30$ and the control horizon $N_u = 1$. The closed-loop stability is achieved by selecting $\lambda = 25$. Control input constraints have been assumed as $u_{max} = 10$, $u_{min} = -5$ and the rate input $\Delta u_{max} = 10$. A step disturbance of magnitude 1.1 is introduced at time $t = 450\,s$ to test the robustness of the design. The maximum number of iterations is three. The results have been compared with the solutions obtained by the classical MBPC with constraints.

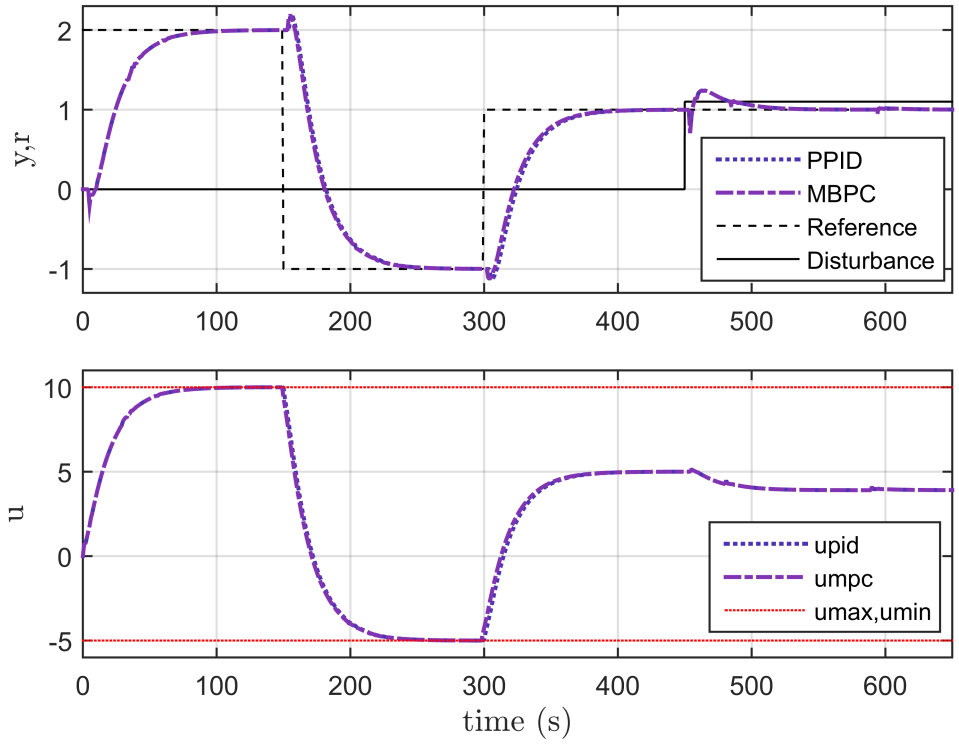

FIGURE 4.14: System outputs for predictive PID and MBPC

Figure 4.14 shows the system outputs and constrained controller inputs of predictive PID and MBPC for step changes in reference signal (dashed line). The percentage and occurrence of dropouts for the simulation are depicted in Figure 4.15. The predictive PID shows almost the same behaviour than MBPC as it is expected. The reference tracking and the disturbance rejection are achieved. Note that the input constraints (dotted line) are satisfied. However, further tests shown this leads to a slower rising time compared to the case without constraints.

Figure 4.16 shows that the KF estimation is close to the process output. Therefore, when a dropout from sensor to controller occurs, the KF provides an accurate estimate of the process output. The control system is stable and works within the requirements for the entire drop of sensor and controller packets. Further tests show that the percentages of dropouts could be increased up to $P_{loss} = 84\%$ which is the threshold where the closed-loop appears to be stable.

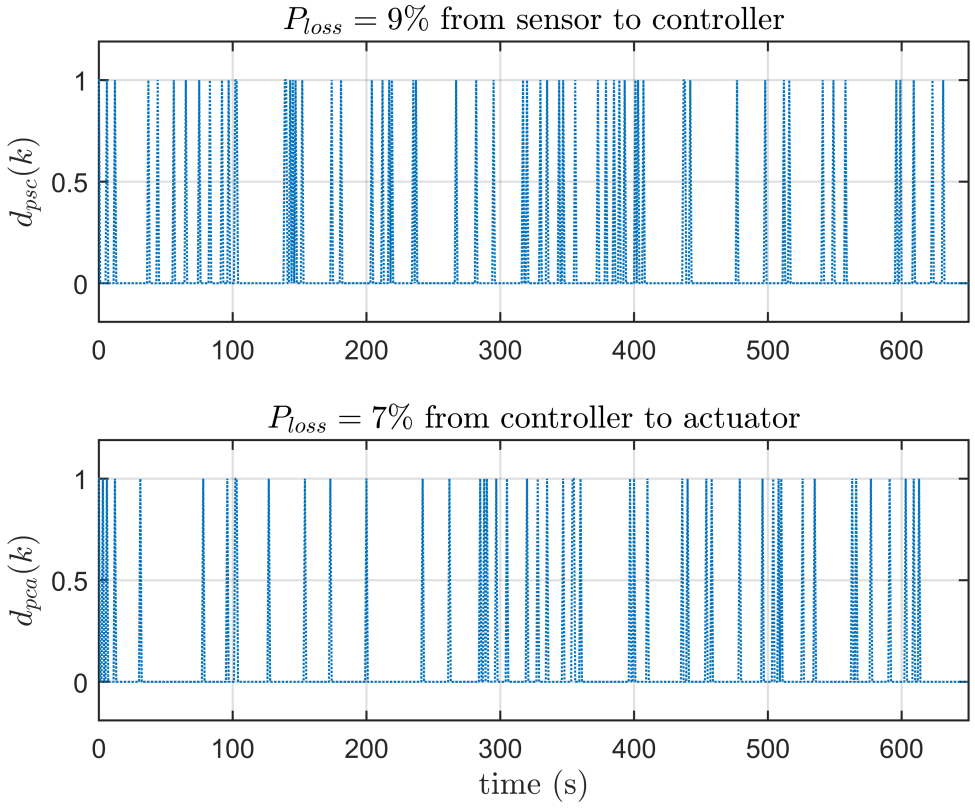

FIGURE 4.15: Time instant of data dropouts

The performance of predictive PID and MBPC responses for servo and regulatory responses has been assessed using the IAE criterion. The results are summarised in the Table 4.3.

TABLE 4.3: IAE values for step responses

Controller	J_r	J_d
Predictive PID	211.8	7.988
MBPC	205	7.989

The indexes values demonstrate that the predictive PID method performs as good as the MBPC scheme.

4.7.13 Robustness results

Since the PID predictive controller is model based, the effects of model uncertainties and dropouts on NCS's stability require further examination. Therefore, the stability

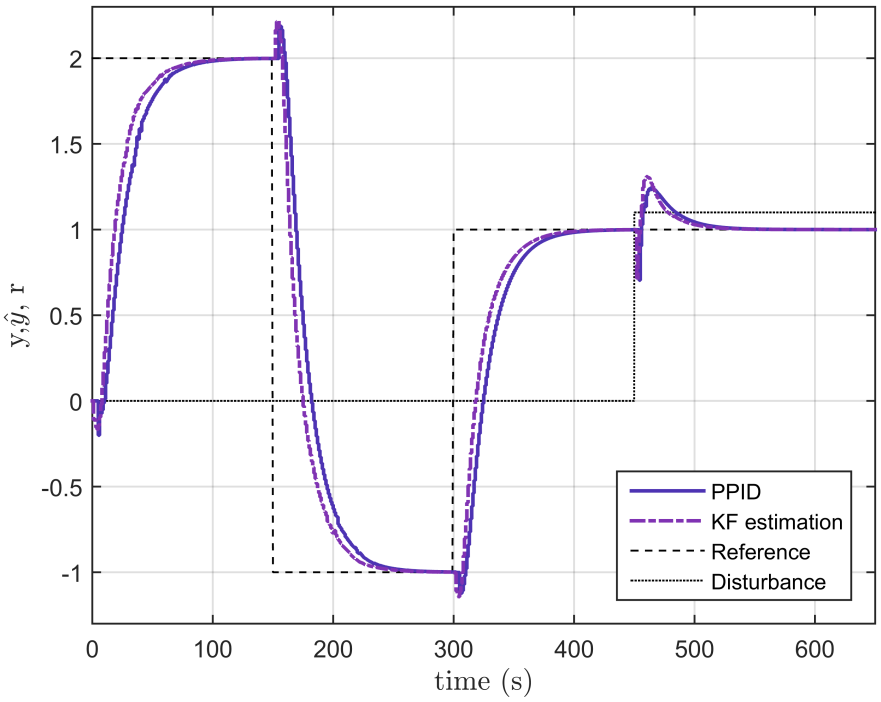

FIGURE 4.16: KF estimation for constrained predictive PID

of the method is investigated here studying the closed-loop responses for variations of the process model parameters and the percentage of dropouts. The following second order process is selected for this analysis:

$$G_p(z) = \frac{0.06347z^{-1} + 0.04807z^{-2}}{1 - 1.323z^{-1} + 0.4346z^{-2}} \tag{4.64}$$

The controller settings are the same than the previous example. A step of magnitude one is selected. Control input constraints have been chosen as $u_{max} = 1$, $u_{min} = 0$ and the rate input as $\Delta u_{max} = 10$.

4.7.14 Study of stability for variations of percentage of dropouts

The percentages of dropouts from sensor to controller and from controller to actuator are varied to demonstrate the robustness of the design. The step responses for different scenarios are shown in Figure 4.17. The dashed lines show that when the probability of loss is increased from 0% to 65%, the responses are similar. Nevertheless, after 20 s, the control input for a 65% of packet loss presents small oscillations. If the probability

keeps increasing, the oscillations continue to grow until the output is unstable. The dotted line shows the response to a higher percentage of dropouts and the performance of control system has decreased considerably.

Further validations reveal that the closed-loop response is stable for a maximum probability of dropouts from sensor to controller of 84%, that means the system is still stable when only 16% of process measurements are transmitted from sensor to controller. Moreover, the control system can compensate at the same time for a percentage of dropouts from controller to actuator of 13% which means that only 87% of the control inputs are received by the process.

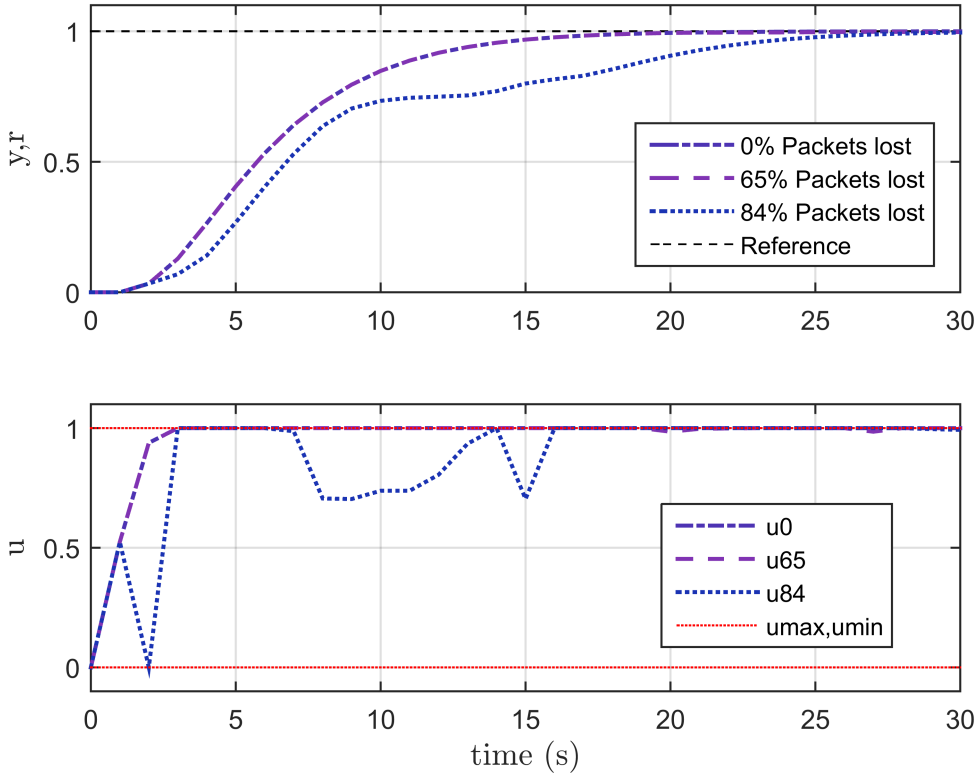

FIGURE 4.17: Comparison of step responses with dropouts variations

4.7.15 Study of stability for variations of the gain

Figure 4.18 shows that even with the constraints, the closed-loop system is stable if the gain is increased and reduced to ±35% of the model process gain. Although the

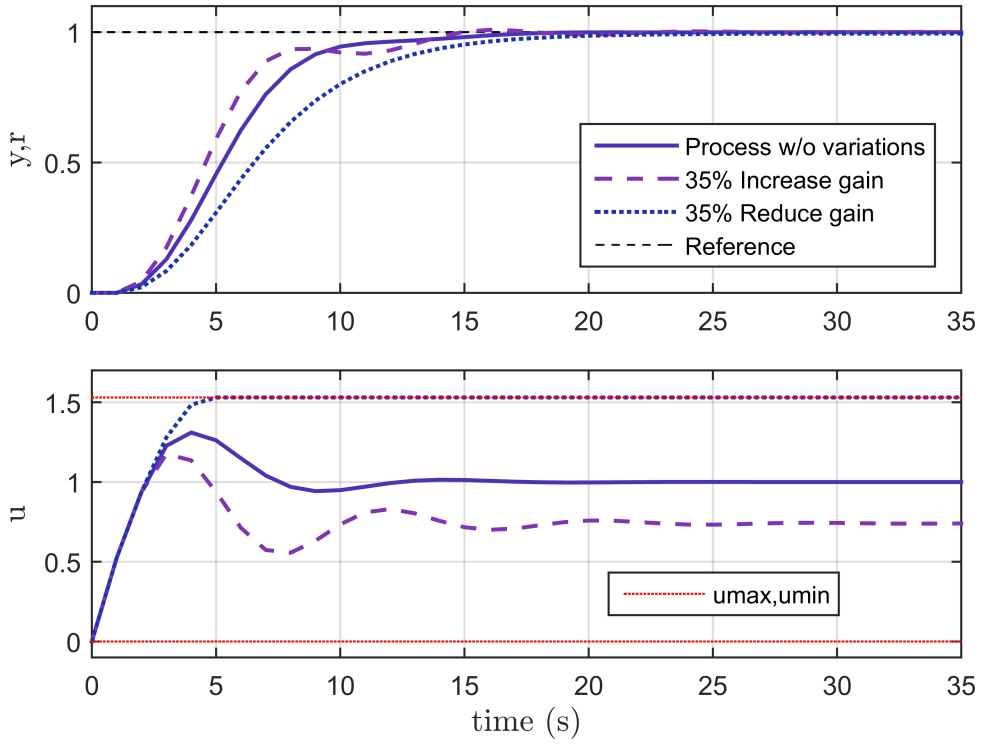

FIGURE 4.18: Comparison of step responses with gain process model variations

process presented a small oscillation and slower rising time, zero steady error and a good tracking performance are accomplished when the process gain changes within the given percentages.

4.7.16 Study of stability for variations of the poles

Figure 4.19 shows the closed-loop responses for variations in the open-loop non-dominant pole called p_1. Note that, the effect of varying the p_2 is similar since the poles are closer to each other. It is evident from the plot, that pole variations of $\pm 35\%$ are permitted without making the closed-loop system unstable.

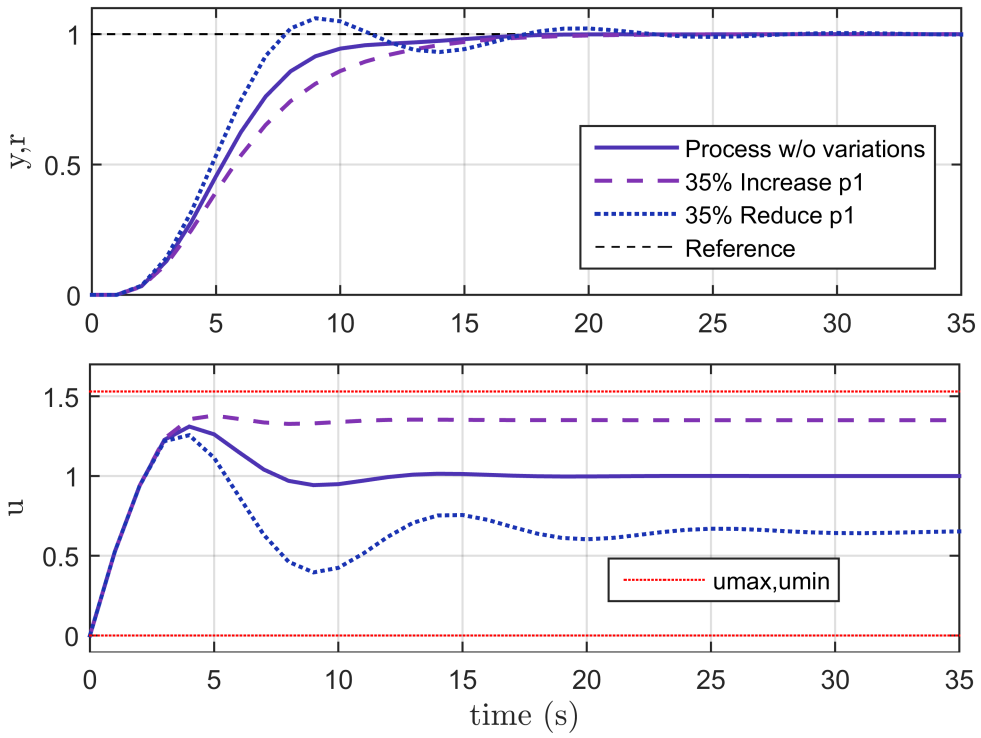

FIGURE 4.19: Comparison of step responses with pole 1 process model variations

4.7.17 Discussion

The performance stability of the control scheme is satisfactory for the selected prediction horizon. Further tests indicate that a larger N degrades the performance because the errors in the prediction are bigger for long prediction horizon. The predictive PID controller and MBPC show similar performances; in some cases, the predictive PID controller performs better than the MBPC for higher percentages of dropouts.

The constraint handling produces a reduction of the performance, but satisfactory results are still found. In most cases, a faster weight λ can improve the sluggish response of the control signal. However, there are scenarios where the control strategy cannot stabilise faster responses with high percentages of dropouts.

The sampling time of the WNCS should be carefully chosen. The selected values for the previous validations can guarantee the stability of the implementation. A special attention has to be considered for the interaction between sensor, actuator and controller

nodes. The sampling times of the first ones must be smaller than the later one.

For the given example, the control performance of the PID controller degrades gracefully when the percentage of dropouts is increased and it is found that $P_{loss} = 84\%$ is the threshold where the closed-loop appears to be stable.

The new predictive PID controller offers a good performance to variations in the model parameters. Also, this methodology can compensate systems subject to dropouts within a large range of variations. The fact that the closed-loop system is robust to process and dropouts variations obeys to the optimisation tool used to obtain the predictive PID controller gains and the accurate estimation of the KF. The approach successfully minimises the error by changing the controller gains at every sampling time and allowing a maximum system parameters variation and percentage of dropouts.

Simulation results demonstrated that the switching action between the KF estimation and the output of the process could have affected the performance, but the KF parameters were adjusted to provide an accurate estimation of the actual output of the process. The Kalman filter corrects the error and tracks the output of the process very quickly. In the next chapter, a KF with time-varying gain has been used to provide the controller with the estimate of the output at each time instant.

4.8 Summary

In this chapter, predictive PID controllers were presented to compensate dropouts in WNCS. In summary, the results demonstrated the effectiveness of the proposed predictive controllers with PID structure for WNCS. The control system was implemented using the TrueTime network simulator. The results showed that the five new approaches successfully solved two major problems in the WNCS: missing sensor measurements and controller actions.

The receding control theory has been combined with a Kalman filter to achieve a control system and an estimation algorithm that compensate for dropouts from sensor to controller. The predictions of the control signal were calculated to compensate consecutive dropouts from controller to actuator. The numerical studies demonstrated that the proposed methods successfully deal with longer dropouts and higher consecutive occurrence.

The designs have been compared with MBPC using a non-minimum phase system with delay. The performance analysis showed that the predictive PID method performs as good as the MBPC scheme. Also, an analysis of the robustness of the system has been studied for a very lossy wireless network and variations of the process model parameters. The simulations scenarios illustrated that the control system meets the stability requirements in the presence of disturbance, process uncertainties, constraints and dropouts. Flexibility and stability supported the interest in predictive PID control.

In the next chapter, the predictive approach will be extended to the complex WNCS with MIMO systems and decentralised control whose results can be tested in an industrial application.

Chapter 5

Decentralised wireless networked model predictive control design for complex industrial systems

5.1 Introduction

The Decentralised Wireless Networked Model Predictive Control system (DWNMPC) is a decentralised model predictive control that involves the exchange of information across a wireless network. Decentralised structures are attractive and widely used solutions for controlling large systems, where the centralised system decomposes into n subsystems each with its local controller; reducing computation time and communication costs. There are decentralised applications in traffic control, load frequency control, unmanned aerial vehicles, energy households control, water and sewer networks, wind farms, wireless sensor/actuator networks and robotics.

The introduction of the network offers the possibility of decreased costs, simplified installation and maintenance and increases the supervision and control capabilities for the whole system. However, the network may introduce large communication delays and loss of information, which greatly influences the stability and robustness of the decentralised control system. A recent survey by Ge, Yang and Han (2017) defined the many points that require more attention in distributed NCS. Han, Peng and Fei (2016) claimed that it is necessary to explore further distributed NCS applications and

investigate the effects of networks in a real-time operation. Also, there are challenging problems that require further studies, such as power limitation, disturbances, constrained systems, strong interactions, among others. These issues are the motivation to develop control structures with flexible and robust characteristics that can handle the wide range of these limitations.

The Decentralised Model Predictive Control (DMPC) began to gain industrial and academic importance in recent years. For example, the industrial inclusion of MPC to Distributed Control Systems (DCS) reported by Qin and Badgwell (2003) is an important advance in this field. In the literature, few contributions can be found and yet the potential of DMPC in control performance has been highlighted (see Tuan et al. (2015) and references therein).

The effect of interactions between subsystems can be considered to improve the performance of the DMPC. Cooperative strategies can be implemented where controllers, also called agents, share their control actions among other agents. For example, Vaccarini, Longhi and Katebi (2009) proposed an unconstrained networked DMPC with interaction predictions for systems with strong interactions. The agents coordinate with each other through the exchange of predictions of control input and state predictions.

It is important to note that DMPC has been addressed mainly for ideal communication channels. However, the introduction of the network communication leads to long delays and dropouts which affect the system performance. The reliability of DMPC for NCS under network constraints is still under investigation. There is a small contribution in this area that includes the work of Freirich and Fridman (2016) and Heemels et al. (2013). They addressed the problem of network delays for large NCS with multiple local networks operating independently and asynchronously. A model of the network that incorporates varying transmission intervals and varying delays is presented. Borgers and Heemels (2014) and Liu, Wang and Liu (2016) extended the analysis to multiple subsystems and a maximum transmission interval is given for each network using the small-gain theorem. Another effort has been made by Zhang, Bao and Xu (2013), who presented an iterative algorithm for unconstrained networked MPC with one-step delay communication using neighbourhood optimisation. Note that, the problem of dropouts has not addressed yet.

The decentralised control of multiple cooperative vehicles under random delays and dropouts was investigated by Izadi, Gordon and Zhang (2011). The concept of the

tube MPC is also employed to limit the exchange of neighbour trajectories when large delays occur. The work of Bemporad and Barcelli (2010) showed that dropouts could be modelled using a Markov chain with m states. The experiments showed the detriment of the performance as a result of the decentralised solution and the effect of the dropouts. The stabilisation condition of a decentralised control system under dropouts using the eigenvalues of the system has been considered by Liu and Gupta (2012) and Liu and Gupta (2017). Also, Alessio, Barcelli and Bemporad (2011) provided the stability criteria of linear systems subject to input constraints and dropouts of maximum one sampling interval. They considered an intermittent lack of communication of measurement data between agents. Note that, the inclusion of the dropouts in the solution gives a complex computation problem and yet the input and output interaction between the subsystems has not taken into account.

In this chapter, the insertion of the network communication and the compensation of its constraints in DMPC are studied. The chapter is structured as follows. Section 5.1 presents a framework of comparison of the different DMPC approaches for NCS and discusses the optimality of the solutions. It also shows the many points that need more attention. Section 5.2 offers a description of structures NCS and characteristics. Then, a new constrained decentralised networked predictive control is developed with a coordination strategy and presented in Section 5.3. The approach is novel to include the effect of the interactions in the subsystems to increase the control performance while using a reduced exchange rate of information. A decentralised KF solution is added to the control scheme to compensate dropouts when the control loop is closed with a wireless network. The effectiveness of the proposed method under disturbances, strong interactions and the reduction of information exchange is demonstrated with several simulation studies in Section 5.4.

5.1.1 Centralised, decentralised and distributed systems

In the literature, there are three main control structures of NCS: centralised, decentralised and distributed. Figures 5.1, 5.2, and 5.3 illustrate the three control schemes.

In the first, a single controller manages all the control actions in the system, which may be unsuitable for large scale systems due to the spatial distribution of the components and the heavy computational load involved. The second scheme divides the system into

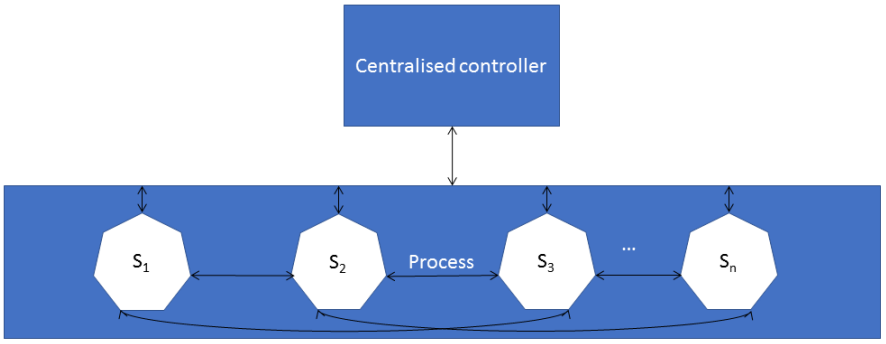

FIGURE 5.1: Centralised control scheme

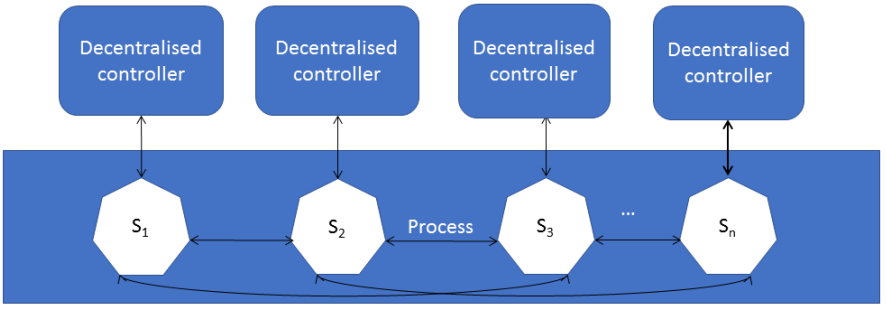

FIGURE 5.2: Decentralised control scheme

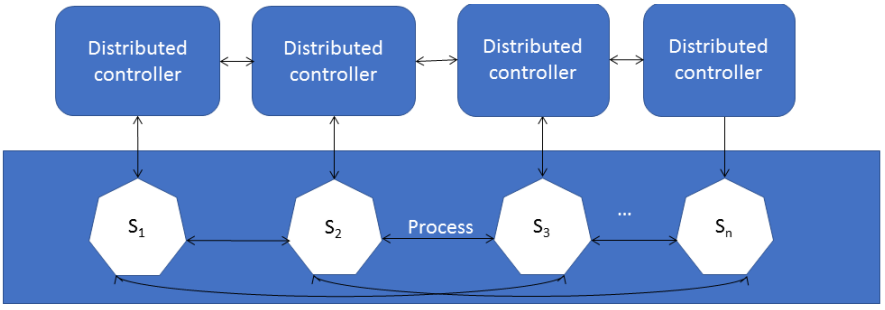

FIGURE 5.3: Distributed control scheme

n subsystems each with its local controller. The lack of communication between agents can lead to suboptimal performance compared to the centralised solution.

Decentralised or fully decentralised systems are classified according to the intensity of interaction between subsystems. In the first option, the interaction is modelled and combined with the subsystem model to construct a composite model. For the latter option, the interaction is considered weak and omitted in the controller design. However, it can be treated as disturbances.

The third structure is similar to the decentralised one, with the main difference being that in the distributed solution a process of negotiation between agents is carried out via the exchange of possible control inputs and the acceptance of a solution where the optimum performance is achieved.

5.2 Decentralised networked model-based predictive control

In the proposed DWNMPC solution, the information that includes predictions of the control input and state is transferred through multiple sensors/actuators connected via a wireless network, which is shared by other control systems. In the presence of dropouts, an estimator for each subsystem is necessary to supply the information.

A decentralised estimation algorithm based on KF was introduced to estimate the states locally in a two-level coordination strategy for GPC by Katebi and Johnson (1997). Also, Roshany-Yamchi et al. (2013) applied a KF based distributed predictive control to large scale multi-rate systems in power networks to compensate for the loss of information.

A decentralised estimation using a decentralised KF is presented to provide an estimate of the states that allows to always have information of the process even in the presence of dropouts. A KF, based on the state-space model, is available for each subsystem. Consider the state-space model S:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\Delta u(k) + \mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{v}(k) \end{aligned} \tag{5.1}$$

where $x(k) \in \mathbb{R}^{nx}$ is the state vector, $\Delta u(k) \in \mathbb{R}^{nu}$ is the vector of control increments, $y(k) \in \mathbb{R}^{ny}$ is the measurement vector, $w(k)$, $v(k)$ are the process and measurement noise sequences, respectively; with covariances Q_f and R_f, respectively. Matrix D is assumed to be zero. Without loss of generality, suppose that the whole system S is composed of n linear, discrete-time subsystems $S_i = 1, \cdots, n$. The subsystems can be divided based on the dynamical interactions existing in the model of S. Each subsystem interacts with others by inputs, outputs and states. Then, the state-space

representation of S_i can be expressed as:

$$\begin{aligned}\mathbf{x}_i(k+1) &= \mathbf{A}_{ii}\mathbf{x}_i(k) + \mathbf{B}_{ii}\Delta\mathbf{u}_i(k) + \mathbf{w}_i(k) + \sum_{j=1;j\neq i}^{n} \mathbf{A}_{ij}\mathbf{x}_j(k) + \sum_{j=1;j\neq i}^{n} \mathbf{B}_{ij}\Delta\mathbf{u}_j(k) \\ \mathbf{y}_i(k) &= \mathbf{C}_{ii}\mathbf{x}_i(k) + \mathbf{v}_i(k) + \sum_{j=1;j\neq i}^{n} \mathbf{C}_{ij}\mathbf{x}_j(k)\end{aligned} \tag{5.2}$$

where $x_i(k) \in \mathbb{R}^{nx_i}$ is the state vector, $\Delta u_i(t) \in \mathbb{R}^{nu_i}$ is the vector of control actions and $y_i(k) \in \mathbb{R}^{ny_i}$ is the measurement vector. Matrices $\mathbf{A_{ii}}$, $\mathbf{B_{ii}}$ and $\mathbf{C}_{ii}$ are sub-matrices of the original $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ matrices, respectively, describing a possible approximation of the evolution of the states of the subsystem i. The process and measurement noise sequences vectors are $w_i(k)$, $v_i(k)$ of dimension $nx_i \times 1$ and $ny_i \times 1$ respectively, which have been assumed Gaussian, uncorrelated, white, with mean $(\bar{x}_{0i}; 0; 0)$ and covariance $(P_{0i}; \mathrm{Q}_{fi}; \mathrm{R}_{fi})$, respectively. It is assumed that the pair $(A_{ii}; C_i)$ is observable, $(A_{ii}; {\mathrm{Q}_{fi}}^{1/2})$ is stabilizable and $\mathrm{R}_{fi} > 0$.

Note that (5.2) is equivalent to:

$$\begin{aligned}\mathbf{x}_i(k+1) &= \mathbf{A}_{ii}\mathbf{x}_i(k) + \mathbf{B}_{ii}\Delta\mathbf{u}_i(k) + \mathbf{w}_i(k) + S_{xi} \\ \mathbf{y}_i(k) &= \mathbf{C}_{ii}\mathbf{x}_i(k) + \mathbf{v}_i(k) + S_{yi}\end{aligned} \tag{5.3}$$

where state and output interaction vectors are given by:

$$\begin{aligned}S_{xi} &= \sum_{j=1;j\neq i}^{n} \mathbf{A}_{ij}\mathbf{x}_j(k) + \sum_{j=1;j\neq i}^{n} \mathbf{B}_{ij}\Delta\mathbf{u}_j(k), \\ S_{yi} &= \sum_{j=1;j\neq i}^{n} \mathbf{C}_{ij}\mathbf{x}_j(k)\end{aligned} \tag{5.4}$$

By the principle of separation, the estimator and the control algorithm are designed separately. The proposed estimator is a decentralised KF, which will provide the estimation of the state vectors $\hat{x}_i(k)$ and $\hat{x}_j(k)$.

5.2.1 Decentralised estimation for dropouts

Consider the decentralised KF for subsystem i denote by:

$$\begin{aligned}\hat{\mathbf{x}}_i(k+1|k) &= \hat{\mathbf{x}}_i(k|k-1) + \gamma_i(k)\,\mathbf{K}_{fi}(k)\,[y_i(k) - C_{ii}\hat{\mathbf{x}}_i(k|k-1)] \\ \hat{\boldsymbol{y}}_i(k) &= C_{ii}\hat{\mathbf{x}}_i(k|k-1)\end{aligned} \tag{5.5}$$

where the index $(k|k-1)$ refers to the information at sampling time k given observations up to and including time $k-1$. The KF gain for each local observer is denoted as K_{fi} and the variable $\gamma_i(k)$ is defined as follows:

$$\gamma_i(k) = \begin{cases} 1 & \text{if } y_i(k) \text{ is received at time } k \\ 0 & \text{if } y_i(k) \text{ is not received} \end{cases} \tag{5.6}$$

During the presence of dropouts, only the equations for estimation (prediction) of the KF are computed. Once the information $y_i(k)$ is available the measurement (update) equations are calculated.

5.2.1.1 Prediction

In the proposed decentralised KF, each local filter should estimate $x_i(k)$ such that the state estimate error for subsystem i, i.e. $\mathbf{e}_i(k) = \mathbf{x}_i(k) - \hat{\mathbf{x}}_i(k)$ is minimised. The one step-ahead prediction is expressed as follows:

$$\hat{\mathbf{x}}_i(k+1|k) = \mathbf{A}_{ii}\hat{\mathbf{x}}_i(k|k-1) + \mathbf{B}_{ii}\Delta\mathbf{u}_i(k) \tag{5.7}$$

Define $P_i(k|k-1)$ as the covariance of the state estimation error:

$$\mathbf{P}_i(k|k-1) = \mathbb{E}\left[\mathbf{e}_i(k|k-1)\mathbf{e}_i^{\mathrm{T}}(k|k-1)\right] \tag{5.8}$$

Then, the noise covariance matrices are: $\mathbb{E}\left[\mathbf{w}_i(k)\mathbf{w}_i^{\mathrm{T}}(k)\right] = \mathrm{Q}_{fi}(k)$, $\mathbb{E}\left[\mathbf{v}_i(k)\mathbf{v}_i^{\mathrm{T}}(k)\right] = \mathrm{R}_{fi}(k)$ respectively. $\mathbb{E}[\cdot]$ denotes the expectation of the argument. Thus:

$$\mathbf{P}_i(k|k-1) = \mathbf{A}_{ii}\mathbf{P}_i(k-1|k-1)\mathbf{A}_{ii}^{\mathrm{T}} + \mathrm{Q}_{fi}(k)^{\mathrm{T}} \tag{5.9}$$

The equations (6.10) and (6.12) are forward projections of state and covariance for a priori estimation. During the dropouts, there is not update of the information.

5.2.1.2 Measurement update

Once the information is available, the prediction is combined with the current observation information. Consider the decentralised KF for subsystem i in (6.8). The modified innovation covariance is:

$$\mathbf{\Omega}_i(k) = \mathbf{C}_{ii}\mathbf{P}_i(k|k-1)\mathbf{C}_{ii}^T + \mathrm{R}_{fi}(k) \tag{5.10}$$

Define the innovation or measurement residual for each subsystem i as:

$$\Lambda_i(k) = \mathbf{y}_i(k) - \mathbf{C}_{ii}\hat{\mathbf{x}}_i(k|k-1) \tag{5.11}$$

Now, updating the estimation error covariance as:

$$\mathbf{e}_i(k+1|k) = \mathbf{x}_i(k+1|k) - \hat{\mathbf{x}}_i(k+1|k) \tag{5.12}$$

Results in:

$$P(k+1|k) = \mathbb{E}\left[\mathbf{e}_i(k+1|k)\mathbf{e}_i^{\mathrm{T}}(k+1|k)\right] \tag{5.13}$$

Substituting the state equation in (5.1) and the estimate state equation in (6.8) and solving the estimation yields that:

$$\begin{aligned}\mathbf{P}_i(k+1|k) = {} & \mathbf{A}_{ii}\mathbf{P}_i(k|k-1)\mathbf{A}_{ii}^{\mathrm{T}} - 2\mathbf{K}_{fi}(k)\mathbf{C}_{ii}\mathbf{P}_i(k|k-1)\mathbf{A}_{ii}^{\mathrm{T}} \\ & + \mathbf{K}_{fi}(k)\mathbf{\Omega}_i(k)\mathbf{K}_{fi}^{\mathrm{T}}(k) + \mathrm{Q}_{fi}(k)^{\mathrm{T}}\end{aligned} \tag{5.14}$$

The minimum error covariance is obtained by calculating $\partial\mathbf{P}_i(k+1|k)/\partial\mathbf{K}_{fi}(k) = 0$. One can find that the following equation should be satisfied:

$$-2\,\mathbf{A}_{ii}\mathbf{P}_i(k|k-1)\mathbf{C}_{ii}^T + 2\mathbf{K}_{fi}(k)\mathbf{\Omega}_i(k) = 0 \tag{5.15}$$

Thus, the KF gain is:

$$\mathbf{K}_{fi}(k) = \mathbf{P}_i(k|k-1)\mathbf{C}_{ii}^T\mathbf{\Omega}_i^{-1}(k) \tag{5.16}$$

The algorithm is a function of $\gamma_i(k)$, which is a binary random number of 0 or 1. As a consequence, the proposed decentralised KF has a time-varying gain where the optimal gain and the error covariance values are calculated at each time sampling. If necessary, the KF parameters $\mathrm{R}_{fi}(k), \mathrm{Q}_{fi}(k)$ can be tuned to minimise the innovation error and improve the estimation algorithm.

5.2.2 Decentralised MBPC

The DWNMPC design is combined with the decentralised KF to compensate dropouts in WNCS. Each subsystem is controlled by an agent that has a local MBPC and a local KF. The MBPC algorithm computes the optimal control input to accomplish a local performance objective. Meanwhile, the KF provides the estimation of the present state, $\mathbf{x}_i(k)$ to the MBPC. In Figure 5.4 a diagram indicates the main functional blocks and the interactions of the DWNMPC control system.

To enhance the global performance of the decentralised solution in comparison to the centralised one, interactions between the subsystems are taken into account in the formulation. The interactions defined in (5.4) show that each subsystem can interact with others by input, states and output. Moreover, predictions of the interactions can be calculated using an estimated state trajectory and a control input trajectory of the subsystems.

The estimated states are computed over the prediction horizon $\hat{\mathbf{x}}_i(k+l)$ for $l = 0, 1, \ldots, N-1$ and stacked in a vector called $\hat{\mathbf{X}}_i(k)$. Similarly, input trajectories are computed over a control horizon and denoted $\hat{\mathbf{U}}_i$.

Then, a coordination strategy is implemented where agents share the estimated trajectories between them. With the available information, each agent calculates its interaction predictions and includes them in the optimisation problem to improve the performance of the decentralised solution.

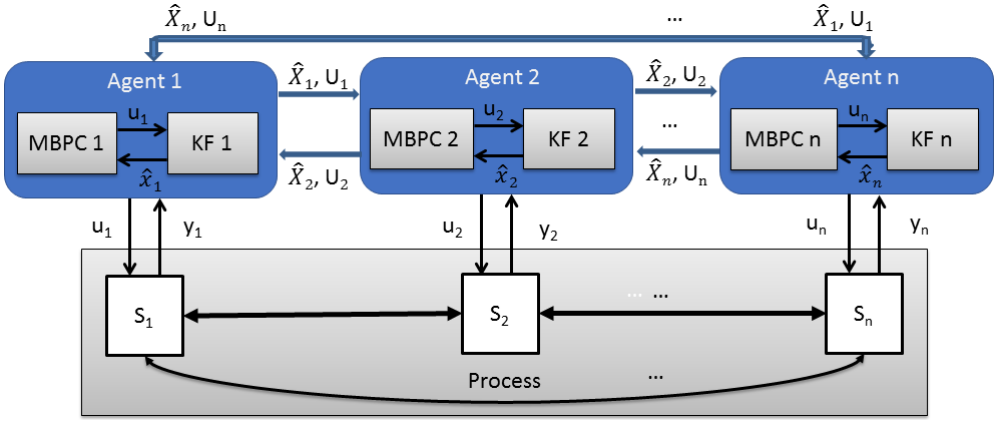

FIGURE 5.4: Structure of DWNMPC control system

The states trajectories $\hat{\mathbf{X}}_i(k)$ and the input trajectories $\hat{\mathbf{U}}_i$ are exchanged between controllers during a selected time interval. For this an exchange loop that runs every β seconds has been implemented. A batch of data containing the trajectories is stored during β seconds and sent to the other agents in the next cycle. Therefore, the predictions of the interactions at time k are computed using the estimated input and state trajectories at time $k-\beta$. Each agent waits the same time before taking the next sampling batch. The reduced rate of exchange of information has a significant impact in the resources utilisation, such as reducing the traffic in the wireless network and thus, reducing the delays and dropouts and reducing the power consumption of the wireless nodes.

Consider the state-space representation in (5.2). For each subsystem S_i the current states $\mathbf{x}_i(k)$ and $\mathbf{x}_j(k)$ are estimated using the proposed decentralised KF and represented as $\hat{\mathbf{x}}_i(k)$ and $\hat{\mathbf{x}}_j(k)$. Then, the estimated state trajectory is computed by replacing one-step-ahead prediction in (5.2) recursively. The obtained output prediction can be written in a compact form as:

$$\begin{aligned}\hat{\mathbf{Y}}_i(k) = \mathbf{F}_i\hat{\mathbf{x}}_i(k) + \phi_{ii}(k)\Delta U_i(k) + \sum_{\substack{j=1\\ j\neq i}}^{n}\hat{\mathbf{X}}_j(k)\phi_{ij}(k) + \\ \sum_{\substack{j=1\\ j\neq i}}^{n}\boldsymbol{\Delta}\mathbf{U}_j(k)\varphi_{ij}(k) + \sum_{\substack{j=1\\ j\neq i}}^{n}\hat{\mathbf{X}}_j(k)\theta_{ij}(k)\end{aligned} \tag{5.17}$$

where

$$\begin{aligned}\hat{\mathbf{Y}}_i(k) &= \left[\mathbf{y}_i^{\mathrm{T}}(k+1)\ \mathbf{y}_i^{\mathrm{T}}(k+2)\ \ldots\ \mathbf{y}_i^{\mathrm{T}}(k+N)\right]^{\mathrm{T}}\\ \hat{\mathbf{X}}_j(k) &= \left[\hat{\mathbf{x}}_j^{\mathrm{T}}(k)\ \hat{\mathbf{x}}_j^{\mathrm{T}}(k+1)\ \ldots\ \hat{\mathbf{x}}_j^{\mathrm{T}}(k+N-1)\right]^{\mathrm{T}}\\ \Delta\mathbf{U}_i(k) &= \left[\Delta u_i^{\mathrm{T}}(k)\ \Delta u_i^{\mathrm{T}}(k+1)\ \ldots\ \Delta u_i^{\mathrm{T}}(k+N_{\mathrm{u}}-1)\right]^{\mathrm{T}}\\ \Delta\mathbf{U}_j(k) &= \left[\Delta u_j^{\mathrm{T}}(k)\ \Delta u_j^{\mathrm{T}}(k+1)\ \ldots\ \Delta u_j^{\mathrm{T}}(k+N_{\mathrm{u}}-1)\right]^{\mathrm{T}}\\ \mathbf{F}_i &= \left[(\mathbf{C}_{ii}\mathbf{A}_{ii})^{\mathrm{T}}\left(\mathbf{C}_{ii}\mathbf{A}_{ii}^2\right)^{\mathrm{T}}\ldots\left(\mathbf{C}_{ii}\mathbf{A}_{ii}^N\right)^{\mathrm{T}}\right]^{\mathrm{T}}\\ \phi_{ii}(k) &= \begin{bmatrix}\mathbf{C}_{ii}\mathbf{B}_{ii} & 0 & \ldots & 0\\ \mathbf{C}_{ii}\mathbf{A}_{ii}\mathbf{B}_{ii} & \vdots & \ddots & \vdots\\ \vdots & \vdots & \vdots & 0\\ \mathbf{C}_{ii}\mathbf{A}_{ii}^{N-1}\mathbf{B}_{ii} & \ldots & \ldots & \mathbf{C}_{ii}\mathbf{A}_{ii}^{N-N_{\mathrm{u}}}\mathbf{B}_{ii}\end{bmatrix}\end{aligned} \tag{5.18}$$

$$\phi_{ij}(k) = \begin{bmatrix} \mathbf{C}_{ii}\mathbf{A}_{ij} & 0 & \dots & 0 \\ \mathbf{C}_{ii}\mathbf{A}_{ii}\mathbf{A}_{ij} & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & 0 \\ \mathbf{C}_{ii}\mathbf{A}_{ii}^{N-1}\mathbf{A}_{ij} & \dots & \dots & \mathbf{C}_{ii}\mathbf{A}_{ii}^{N-N_{\mathrm{u}}}\mathbf{A}_{ij} \end{bmatrix}$$

$$\varphi_{ij}(k) = \begin{bmatrix} \mathbf{C}_{ii}\mathbf{B}_{ij} & 0 & \dots & 0 \\ \mathbf{C}_{ii}\mathbf{A}_{ii}\mathbf{B}_{ij} & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & 0 \\ \mathbf{C}_{ii}\mathbf{A}_{ii}^{N-1}\mathbf{B}_{ij} & \dots & \dots & \mathbf{C}_{ii}\mathbf{A}_{ii}^{N-N_{\mathrm{u}}}\mathbf{B}_{ij} \end{bmatrix}$$

$$\theta_{ij}(k) = \begin{bmatrix} \mathbf{C}_{ij} & \mathbf{C}_{ij} & \cdots & \mathbf{C}_{ij} \end{bmatrix}^T$$

It is assumed that the best estimation of the future process noise and measurement noise is zero, i.e. $w_i(k+j) = 0$ and $v_i(k+j) = 0$, respectively.

Note that, $\hat{X}_j(k)$ is the vector of estimated states over the prediction horizon of the subsystem j and is provided by the state estimation of the agent j. This information and the estimated input trajectory, $\Delta U_j(k)$, are exchanged between the agents during a selected time interval.

5.2.3 Quadratic programming (QP) problem

The proposed DWNMPC is based on the following finite-time constrained local optimisation problem that seeks a set of optimal control signals that minimises the following quadratic cost function:

$$\begin{aligned} &\min_{\Delta u_i(k)} J_i = \sum_{j=1}^{N} \|\mathbf{y}_i(k+j|k) - \mathbf{r}_i(k+j|k)\|^2_{\mathbf{Q}_i} + \sum_{j=0}^{N_{\mathrm{u}}-1} \|\Delta u_i(k+j|k)\|^2_{\mathbf{R}_i} \\ &\text{subject to} \\ &u_{i,\min} \le u_i(k+j) \le u_{i,\max} \\ &\Delta u_{i,\min} \le \Delta u_i(k+j) \le \Delta u_{i,\max} \quad \text{for } j = 0, 1, 2, \dots, N_u \\ &\mathbf{C}_{yi}\mathbf{y}_i(k+j) \le \mathbf{d}_{yi} \quad \text{for } j = 1, 2, \dots, N \end{aligned} \tag{5.19}$$

where $R_i \ge 0, Q_i \ge 0$ are the weighting matrices and the reference is denoted by $\mathbf{r}_i$.

The linear discrete-time systems have control inputs constraints, which satisfy at each sampling time desired maximum and minimum bounds. The input constraints will guarantee the contributions of control input and rate input are applied according to the controller limitations. The lower and upper bounds for the input $u_i(k)$ are denoted by $u_{i,\min}$ and $u_{i,max}$, respectively. Similarly, the lower and upper bounds for the rate input $\Delta u_i(k)$ are denoted by $\Delta u_{i,\min}$ and $\Delta u_{i,max}$, respectively. The output constraints matrices $\mathbf{C}_{yi}, \mathbf{d}_{yi}$ are defined in Section 5.2.4.

The problem in (5.19) is equivalent to a quadratic programming problem, which can be solved using Matlab *quadprog* function. Consider the notation $\|y\|_Q$ defines the weighted Euclidean norm, i.e., $\|y\|_Q^2 = y^T Q y$. Then, expression J can be written as:

$$J_i = [\hat{\mathbf{Y}}_i(k) - \mathbf{R}_i^0(k)]^T \bar{\mathbf{Q}}_i [\hat{\mathbf{Y}}_i(k) - \mathbf{R}_i^0(k)] + \Delta \mathbf{U}_i^T(k) \bar{\mathbf{R}}_i \Delta \mathbf{U}_i(k) \tag{5.20}$$

where $\mathbf{R}_i^0(k) = [r_i^T(k+1)\ r_i^T(k+2), \ldots, r_i^T(k+N)]^T$ is the future reference and $\bar{\mathbf{Q}}_i = diag[Q_i, \ldots, Q_i]$ and $\bar{\mathbf{R}}_i = diag[R_i, \ldots, R_i]$ are the block-diagonal output and input weight matrices, respectively.

The quadratic function can be expressed as:

$$J_i = \quad \frac{1}{2}\, \mathbf{x}_i^{\mathrm{T}}\, \mathbf{H}_i\, \mathbf{x}_i + \mathbf{G}_i^T\, \mathbf{x}_i \tag{5.21}$$

where

$$\begin{aligned} \mathbf{x}_i &= \Delta \mathbf{U}_i(k) \\ \mathbf{H}_i &= 2[\phi_{ii}^{\mathrm{T}}(k) \bar{\mathbf{Q}}_i \phi_{ii}(k) + \bar{\mathbf{R}}_i] \\ \mathbf{G}_i &= 2\phi_{ii}^T(k) \bar{\mathbf{Q}}_i [\hat{Z}_i(k) - \mathbf{R}_i^0(k)] \end{aligned} \tag{5.22}$$

where

$$\begin{aligned} &\hat{\mathbf{Z}}_i(k) = \mathbf{F}_i \hat{\mathbf{x}}_i(k) + \mathbf{\Gamma}_{ii}(k)\, \mathbf{U}_i(k-1) + \sum_{\substack{j=1 \\ j \neq i}}^{n} \hat{\mathbf{X}}_j(k) \phi_{ij}(k) + \sum_{\substack{j=1 \\ j \neq i}}^{n} \Delta \mathbf{U}_j(k) \varphi_{ij}(k) + \\ &\mathbf{\Gamma}_{ij}(k)\, \mathbf{U}_j(k-1) + \sum_{\substack{j=1 \\ j \neq i}}^{n} \hat{\mathbf{X}}_j(k) \theta_{ij}(k) \end{aligned} \tag{5.23}$$

with

$$\mathbf{\Gamma}_{ii}(k) = \begin{bmatrix} \mathbf{C}_{ii}\mathbf{B}_{ii} \\ \mathbf{C}_{ii}\mathbf{A}_{ii}\mathbf{B}_{ii} \\ \cdots \\ \mathbf{C}_{ii}\mathbf{A}_{ii}^{N-1}\mathbf{B}_{ii} \end{bmatrix}, \; \mathbf{\Gamma}_{ij}(k) = \begin{bmatrix} \mathbf{C}_{ii}\mathbf{B}_{ij} \\ \mathbf{C}_{ii}\mathbf{A}_{ii}\mathbf{B}_{ij} \\ \vdots \\ \mathbf{C}_{ii}\mathbf{A}_{ii}^{N-1}\mathbf{B}_{ij} \end{bmatrix},$$

$$\mathbf{U}_i(k-1) = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T \mathbf{u}_i(k-1)$$

Note that the past value of the control signals u_i and u_j are included to provide integral action in the design. Therefore, the quadratic programming (QP) is:

$$\begin{aligned} & J_i = \Delta\mathbf{U}_i^{\mathrm{T}}(k)\mathbf{H}_i\Delta\mathbf{U}_i(k) \; + \mathbf{G}_i\Delta\mathbf{U}_i(k) \\ & \text{subject to } \boldsymbol{C}_{ci}\Delta\boldsymbol{U}_i \leq \boldsymbol{d}_{ci} \end{aligned} \tag{5.24}$$

The problem is subject to output and input constraints as stated in (5.19). The design of $\boldsymbol{C}_{ci}$ and $\boldsymbol{d}_{ci}$ is extended in the next section.

5.2.4 Constraints handling

For $i \in \{1, n\}$, consider the inequities for the input rate in (5.19):

$$-\Delta u_{i,min} \leq \Delta u_i(k+j) \leq \Delta u_{i,max}, \quad \text{for } j = 0, 1, 2, \ldots, N_u \tag{5.25}$$

Then, the inequality can be arranged as a matrix which has the compact form $\boldsymbol{C}_{\Delta ui}\Delta\mathbf{U}_i \leq \mathbf{d}_{\Delta i}$. Similarly, the inequities for the input are:

$$-u_{i,min} \leq u_i(k+j) \leq u_{i,max} \quad \text{for } j = 0, 1, 2, \ldots, N_u \tag{5.26}$$

If $\mathbf{U}_i = \begin{bmatrix} u_i^{\mathrm{T}}(k) \; u_i^{\mathrm{T}}(k+1) \; \ldots \; u_i^{\mathrm{T}}(k+N_{\mathrm{u}}-1) \end{bmatrix}^{\mathrm{T}}$ then, the inequality can be arranged as a matrix which has the compact form $\boldsymbol{C}_{ui}\boldsymbol{U}_i \leq \boldsymbol{d}_{ui}$.

Consider $\Delta u_i(k) = u_i(k) - u_i(k-1)$. Now, the future input increments are:

$$u_i(k+i) = u_i(k-1) + \Delta u_i(k) + \Delta u_i(k+1) + \ldots + \Delta u_i(k+i) \tag{5.27}$$

If $\mathbf{L}_i = [1 \; 1 \; \ldots \; 1]^{\mathrm{T}}$ and $\mathbf{E}_i$ is a lower triangular matrix of ones, the equations can be expressed in compact form as: $\mathbf{U}_i = \mathbf{E}_i\Delta\mathbf{U}_i + \mathbf{L}_i \; u_i(k-1)$. The final constraint matrix

is found by combining the last three compact forms:

$$\begin{bmatrix} \boldsymbol{C}_{\mathbf{\Delta ui}} \\ \boldsymbol{C}_{ui}\boldsymbol{E}_i \end{bmatrix} \Delta \mathbf{U_i} + \begin{bmatrix} 0 \\ \boldsymbol{C}_{ui}\boldsymbol{L}_i \end{bmatrix} \mathbf{u_i}(\mathbf{k-1}) \leq \begin{bmatrix} \mathbf{d}_{\Delta ui} \\ \boldsymbol{d}_{ui} \end{bmatrix} \tag{5.28}$$

Now, consider the constraints for the output as: $y_{i,\min} \leq y_i(k+j) \leq y_{i,\max}$ for $j = 1, 2, \ldots, N$. One can find the compact as $\mathbf{C}_{yi}\mathbf{y}_i(k+j) \leq \boldsymbol{d}_{yi}$. Next, replacing the output prediction in (5.17) results in:

$$\begin{aligned} &\mathbf{C}_{yi}\mathbf{F}_i\hat{\mathbf{x}}_i(k) + \mathbf{C}_{yi}\phi_{ii}(k)\Delta\mathbf{U}_i(k) + \mathbf{C}_{yi}\sum_{\substack{j=1\\j\neq i}}^{n}\hat{\mathbf{X}}_j(k)\phi_{ij}(k) + \mathbf{C}_{yi}\sum_{\substack{j=1\\j\neq i}}^{n}\Delta\mathbf{U}_j(k)\varphi_{ij}(k) + \\ &\mathbf{C}_{yi}\sum_{\substack{j=1\\j\neq i}}^{n}\hat{\mathbf{X}}_j(k)\theta_{ij}(k) \leq \boldsymbol{d}_{yi} \end{aligned} \tag{5.29}$$

This is equivalent to:

$$\mathbf{C}_{yi}\mathbf{T}_i(k) + \mathbf{C}_{yi}\phi_{ii}(k)\Delta\mathbf{U}_i(k) \leq \boldsymbol{d}_{yi} \tag{5.30}$$

where

$$\mathbf{T}_i(k) = \mathbf{C}_{yi}\mathbf{F}_i\hat{\mathbf{x}}_i(k) + \mathbf{C}_{yi}\sum_{\substack{j=1\\j\neq i}}^{n}\hat{\mathbf{X}}_j(k)\phi_{ij}(k) + \mathbf{C}_{yi}\sum_{\substack{j=1\\j\neq i}}^{n}\Delta\mathbf{U}_j(k)\varphi_{ij}(k) + \mathbf{C}_{yi}\sum_{\substack{j=1\\j\neq i}}^{n}\hat{\mathbf{X}}_j(k)\theta_{ij}(k) \tag{5.31}$$

In terms of $\Delta\mathbf{U_i}$ the constraint can be reformulated as:

$$\mathbf{C}_{yi}\phi_{ii}(k)\Delta\mathbf{U}_i(k) \leq \boldsymbol{d}_{yi} - \mathbf{C}_{yi}\mathbf{T}_i(k) \tag{5.32}$$

Finally, input and output constraints can be combined in: $\boldsymbol{C}_{ci}\,\Delta\mathbf{U_i} \leq \boldsymbol{d}_{\mathbf{ci}}$ with $\boldsymbol{C}_{ci}$ and $\boldsymbol{d}_{ci}$ corresponding to the appropriate matrices. It can be seen that, this is the form defined in the optimisation problem (5.24).

In case the system is not constrained, the solution is found as follows: $\Delta\boldsymbol{U}_i(k) = \boldsymbol{K}_i[\mathbf{R}_i^0(k) - \hat{\mathbf{Z}}_i]$ where $\boldsymbol{K}_i = \boldsymbol{H}_i^{-1}\phi_{ii}(k)^T\bar{\mathbf{Q}}_i$.

5.2.5 DWNMPC algorithm

The DWNMPC for each agent can be implemented using the following procedure.

Step 1: Initialisation

(a) Find the system model S and a possible representation of the subsystems $S_i = 1, \ldots, n$ based on the interactions.

(b) Select the prediction horizons N, N_u and the future reference vector.

Step 2: Off-line calculation

(a) Compute the time-invariant part of the constraint matrices.

Step 3: Control loop kTs

(a) Read the actual information from the process.

(b) Compute the present state vector using the decentralised KF.

(c) Read the future state trajectories and control inputs from other agents.

(d) Combine the local trajectories with those of the other agents and calculate the predictions of the interactions.

(e) Solve the constrained optimisation problem and apply the first element as the control input to the subsystem S_i.

(f) Check the closed-loop performance and innovation errors and fine-tune the weights and KF parameters if necessary.

Step 4: Data exchange loop βTs

(a) Send the future state trajectories and the optimal control sequence to other agents.

5.3 Simulation results

The effectiveness of the DWNMPC solution has been tested using the following process:

$$\begin{bmatrix} y_1(k) \\ y_2(k) \\ y_3(k) \end{bmatrix} \begin{bmatrix} \frac{2(z^{-1}-0.492z^{-2})}{(1-1.5z^{-1}+0.7z^{-2})} & \frac{0.0018\,\alpha(z^{-1}+0.935z^{-2})}{(1-0.25z^{-1})(1+0.3z^{-1})} & \frac{0.0047\,\alpha z^{-1}}{(1-0.119z^{-1})} \\ \frac{0.0126\,\alpha z^{-1}}{(1-0.368z^{-1})} & \frac{3(z^{-1}-0.668z^{-2})}{(1-1.572z^{-1}+0.670z^{-2})} & \frac{0.0126\,\alpha z^{-1}}{(1-0.1z^{-1})} \\ \frac{0.0126\,\alpha z^{-1}}{(1-0.368z^{-1})} & \frac{0.0147\,\alpha z^{-1}}{(1-0.1z^{-1})} & \frac{0.99z^{-1}}{(1-0.4z^{-1})} \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \\ u_3(k) \end{bmatrix} \tag{5.33}$$

The process has been transformed into a state-space representation using a sampling time $Ts = 0.7\,s$. The final state-space matrices can be found in Appendix C.

The parameter $\alpha = 0.1$ has been selected to simulate the interaction. $N = N_u = 3$. The weights have the following values: $R_i = 1$ and $Q_i = 100$ for $i = 1, 2, 3$. The constraints are set as $u_{i,max} = 0.8$, $u_{i,min} = -1.2$ and $\Delta u_i = 50$ for $i = 1, 2, 3$. The noise covariances are chosen to ensure a satisfactory closed-loop performance and the values are: $Q_{f1} = 0.005$, $R_{f1} = 2.3$, $Q_{f2} = 1 \times 10^{-8}$, $R_{f2} = 0.1$ and $Q_{f3} = 0.015$, $R_{f3} = 2$.

The time interval to exchange the state trajectory and input predictions between agents is given by $\beta = 10\,s$. Note, that exchanged information only affects the interaction predictions S_{xi}, S_{yi}.

The system responses and the KF estimations are shown in Figure 5.5. The control inputs are depicted in Figure 5.6. The step responses of y_1 and y_2 present smaller oscillations and overshoots meanwhile y_3 shows the best performance. According to the process in (5.33) the dominant eigenvalues are provided by the subsystems S_1 and S_2 which cause their slower convergence in the simulation.

The occurrence of consecutive dropouts for the three responses are depicted in Figures 5.7, 5.8 and 5.9. The innovation errors for each response are depicted as well. During intervals that the dropouts from sensor to controller occurred, the variance goes to zero. A better accuracy can be found with lower noise covariances. The results assured that the decentralised solution with dropouts, noise and interactions is effective. Also, the input constraints are followed.

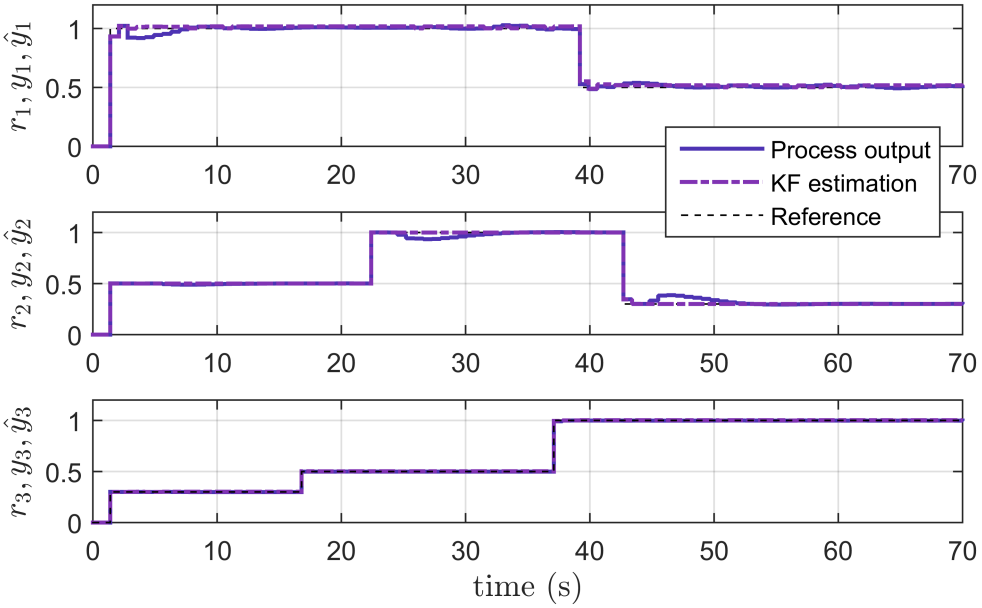

FIGURE 5.5: Process outputs

5.3.1 Study of stability and robustness

More simulations are performed for different scenarios of the percentage of dropouts. The results demonstrate that increasing the percentages from 1% to 90% slightly increases the IAE value and the system appears to be stable. The percentages of dropouts could be increased up to $P_{loss} = 90\%$ which is the threshold where the closed-loop appears to be stable. Table 5.1 lists the results of the IAE criteria for different scenarios of dropouts.

TABLE 5.1: Performance indexes for different scenarios of percentage of dropouts

P_{loss}	IAE Values
10 %	1.88
60 %	18.77
90 %	31.02

A study of the stability for variations of the gain of the model process used by the KF has been performed. The gains of the three transfer functions in the main diagonal of the model process in (5.33) are varied. The tests show that the closed-loop system is stable if the gains are increased and reduced to ±30% of the real process gain. For $P_{loss} = 10\%$, the results of an increment and decrement of the gain return performance indexes of $J = 1.81$ and $J = 1.86$.

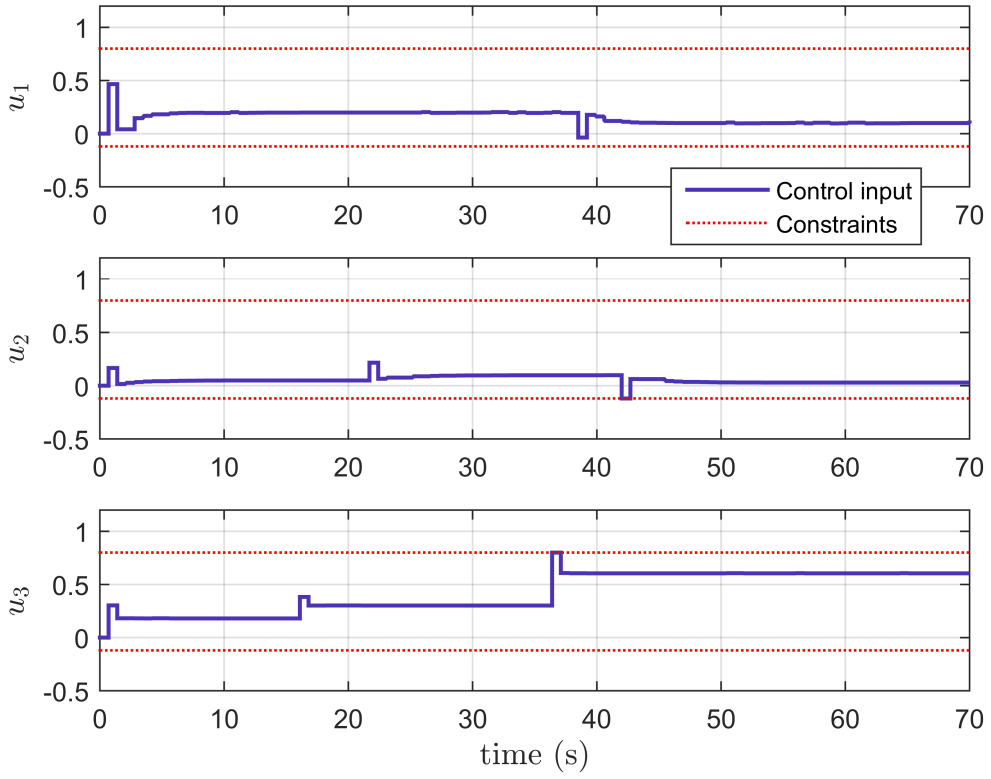

FIGURE 5.6: Control inputs

Moreover, further tests are performed where the stability for variations of the poles is examined. The results exhibit that pole variations of ±5% are permitted in the transfer functions of the main diagonal and the closed-loop system appears to be stable with good tracking performance. For $P_{loss} = 10\%$, the results of an increment and decrement of the poles return performance indexes of $J = 6$ and $J = 5.62$, respectively. The effect of variation in the poles of the model is more severe than the gain variation for the control systems stability.

The stability test is extended to examine the effect of parameter variations on the DWNMPC system. In this case, a small variation of the gain and poles of the model is allowed to maintain stability with good tracking response. The gain of the model can be varied up to ±2%. The performance indexes for an increment and decrement of the gain are $J = 3.56$ and $J = 3.83$, respectively. Poles variations up to ±2% are the limit that can guarantee the system stability with good tracking performance. The performance indexes for an increment and decrement in the poles are $J = 4.35$ and $J = 3.65$,

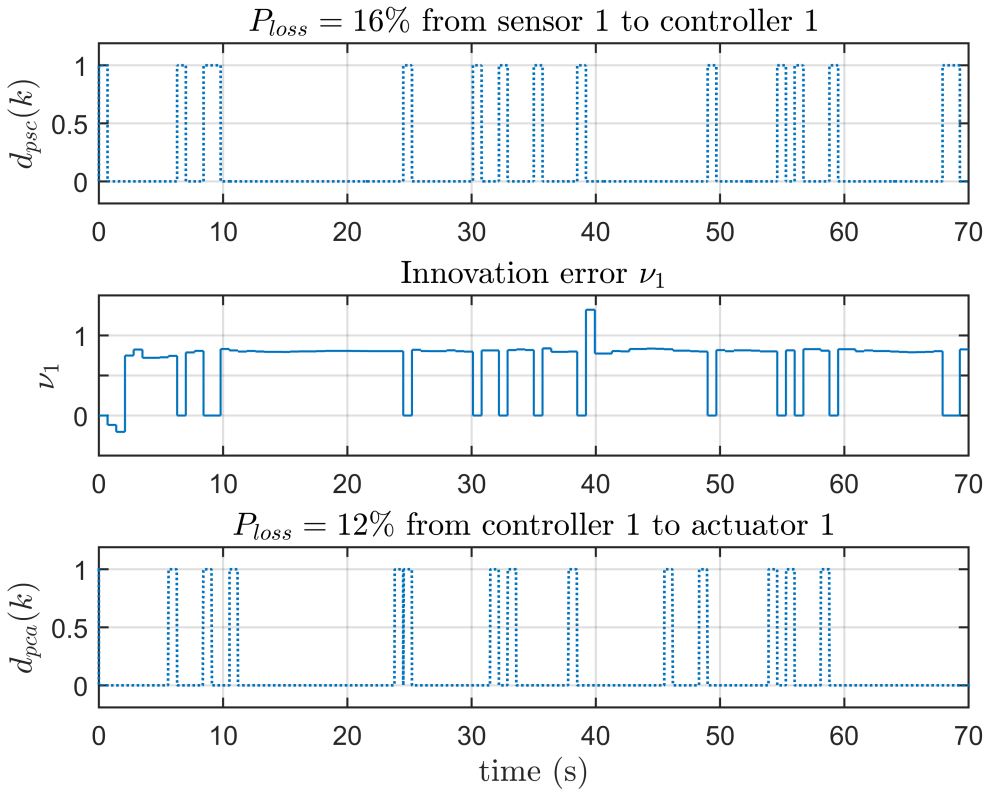

FIGURE 5.7: Time instant of dropouts and innovation error for y_1

respectively. The decentralised MBPC design is more susceptible to variations in the model parameters. The effect can be reduced by including an offset term in the output predictions that can correct deviations between the output of the process model and the actual process. This modification is examined in the next chapter.

5.3.2 Discussion

Several simulations are done where the performance is measured for different values of prediction horizon N and the parameter β. Figure 5.10 shows that the correlation between prediction horizon and the performance is almost negligible. It also demonstrates that similar IAE values are found for any value of β. The reason is that the exchange loop that runs every β seconds affects the prediction interactions, which will be combined with the local estimation to compute the process predictions using (5.17). Moreover, the DWNMPC uses the KF to obtain the best estimation of the process

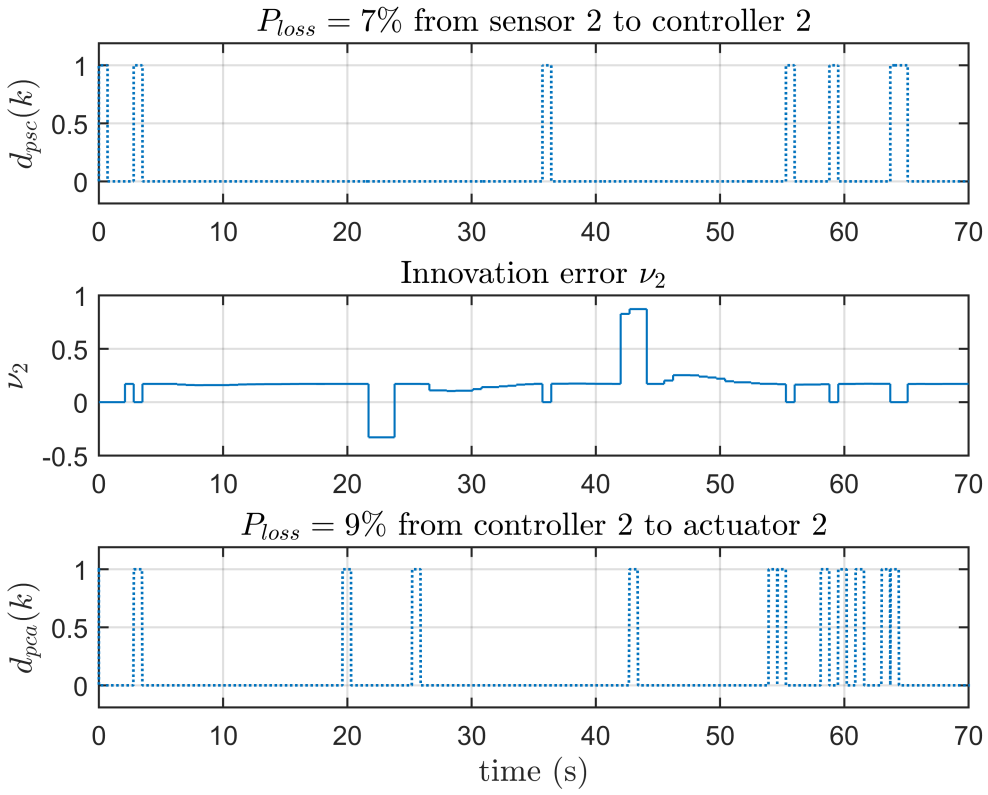

FIGURE 5.8: Time instant of dropouts and innovation error for y_2

which allows to maintain an accurate estimate of the real process during the sampling times that the systems do not exchange information.

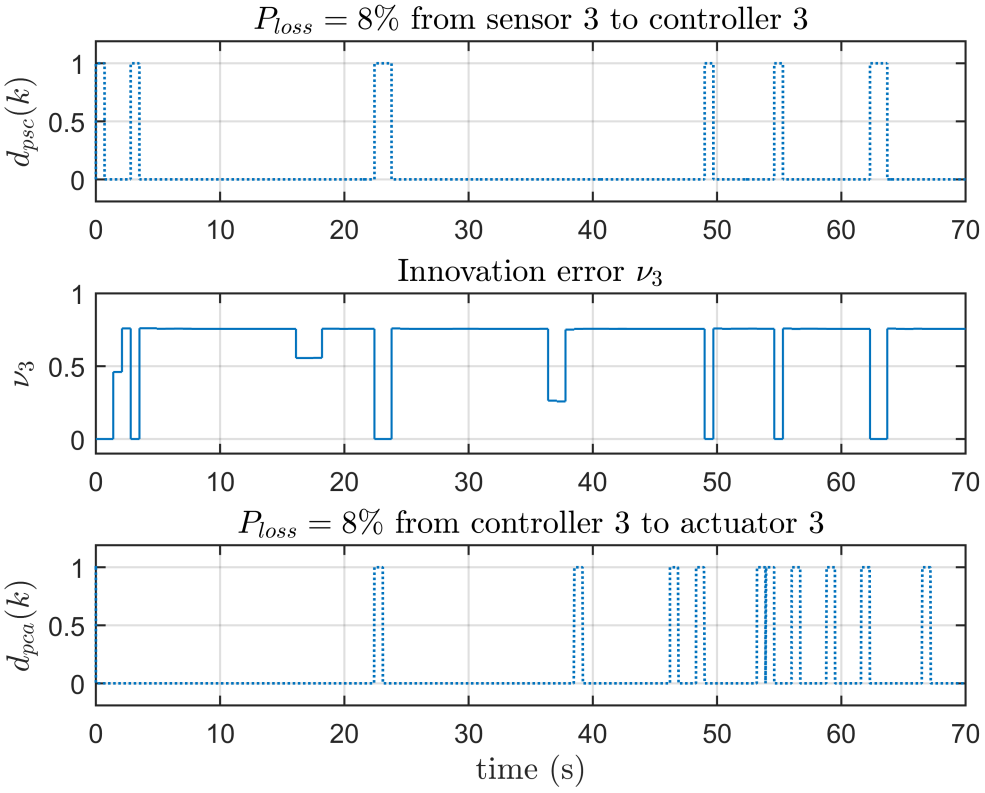

FIGURE 5.9: Time instant of dropouts and innovation error for y_3

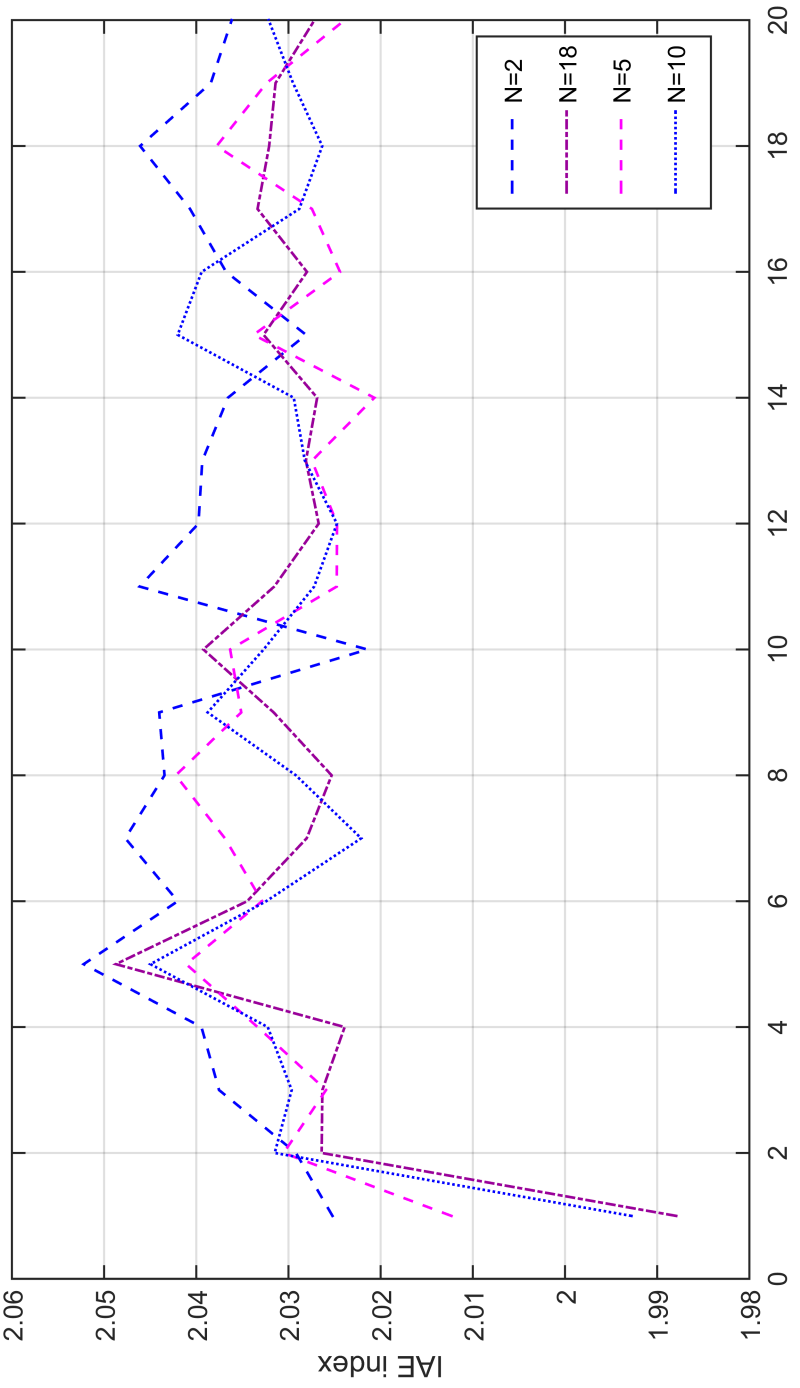

FIGURE 5.10: IAE index versus β for different values of N

Further tests where the IAE criterion is measured for different values of Q_i and R_i under various scenarios of percentages of dropouts show that the best performance is found with a larger output weight relative to the input weight. Therefore, it is more important to penalise the deviation from the reference than the control input.

The noise in the process and measurement noise have been added to each control system according to (5.2) and the decentralised KF has been designed to handle these terms. Moreover, the KF has to be tuned to compensate the dropouts. The simulations results show that the prediction is very aggressive to maintain the system stability. This is especially important during a switch between the prediction stage and the update stage of the KF since abrupt changes can occur from the decentralised estimation and the current process. Another tests where the IAE criterion is measured show that the best performance is obtained with the smallest relation between Q_{fi}/R_{fi}. Therefore, a larger value of measurement noise was introduced to find the best estimation. The innovation errors depicted in Figures 5.7, 5.8 and 5.9 show that the measurement noise has more effect in the innovation error, reducing it will cause the error to tend to zero.

The state-space model has been constructed assuming there is an interaction between the subsystems outputs. For this, output interaction vectors have been added and the variable α has been changed from one to ten to simulate the effect of strong interactions. The simulations show that the increment of α causes the detriment of the performance which is measured using the IAE criterion (for $\alpha = 0.1$: IAE = 54.17, $\alpha = 1$: IAE = 54.56 and $\alpha = 10$: IAE = 61.94). However, the response can be fine-tuned to reduce the error using the weight parameters of the MBPC and the KF.

The decentralised KF in (6.8) can include the interactions in the states, input and the output as defined in (5.4). In this work, the interactions are handle successfully using the decentralised MBPC formulation.

5.4 Summary

An extensive literature review showed that only a few contributions in DMPC consider the effect of the dropouts introduced by the network and their analysis does not take into account system dynamics such as interactions, disturbance and noise. In this chapter, an innovative DNMPC with a decentralised KF was presented to compensate dropouts in WNCS. A unified study of the effect of input and output constraints and disturbance

in the decentralised state estimation for subsystems with strong interactions was done. A decentralised constrained optimisation problem was introduced and it provided the optimum set of control inputs that met the constraints. Moreover, the problem included output interactions between the subsystems. The effect of these was simulated by manipulating the parameter α. The results demonstrated that increasing α allowed to simulated strong interactions. As a result of strong interactions a permanent error in the less dominant subsystem was found. However, this was eliminated by a fine-tuning of the MPC weights.

A decentralised estimation algorithm that considers dropouts in the analysis was added to the control scheme. A decentralised KF provided the DMPC quadratic problem with the estimate of the present states in case of dropouts. An aggressive set of tuning parameters Q_{fi}, R_{fi} was selected to reduce the deviation between the estimation and the real process. This led to a good performance of the system even in the presence of dropouts.

The decentralised control performance was improved through the implementation of a coordination strategy. The approach required an exchange of the state trajectory and the input between each agent to accomplish a global performance objective. A reduced exchange rate was proposed to decrease the traffic in the network and thus, the occurrence of long delays, dropouts and the problem of power limitations in the wireless nodes. It can be concluded that the decentralised strategy reduces the required amount of communication while maintaining control performance.

Chapter 6

Decentralised wireless networked model predictive control design for wind farm

6.1 Introduction

Wind turbines have attracted a significant interest and investment in the search for clean and green energy sources. This alternative energy contributes to the sustainable objectives that most countries will have compromised to meet in the next years due to environmental concerns and depletion of fossil fuels. In 2016, more than 54 GW of renewable and clean wind energy was installed, in more than 90 countries, including nine with more than 10,000 MW installed and 29 having exceeded the 1,000 MW mark. The five-year forecast by the Global Wind Energy Council foresees almost 60 GW of new wind power facilities by 2017, reaching an annual market of around 75 GW by 2021 (Council, 2016).

However, the total cost of the turbine life cycle has to be reduced to make wind power more competitive compared to other energy sources. The challenge associated with wind energy harvesting is the high cost of operation and maintenance, especially in offshore wind turbines due to logistical difficulties in accessing production sites. By employing appropriate control strategies, the cost of producing wind power can be further reduced (Njiri and Söffker, 2016).

Control of wind turbines is a major concern in energy production to ensure that the maximum exploitation of the resource is made. Fast and accurate control of the power generated by the wind contributes to the efficient operation of the wind turbine.

Following the same argument, the potential of decentralised control is applied to wind farm control due to the interest of exploiting the maximum potential of the resources. New decentralised and autonomous architectures have become the future and a reality for some industry and manufacturing processes. The advent of the decentralised solution affects power plants, vehicle industry, cyber physical systems, smart cities, logistical systems, transportation, shipping and more others.

The main drawback in decentralised solutions is the difficulty of determining where the control action is taking place. The fundamental solution to make the decentralised control reliable is to show transparency of all actions in the process. To this end, in this chapter a decentralised structure of the control system and an algorithm of exchange of information and cooperation between all wind turbine controllers are developed. The wind farm control proposed is a benchmark that can be used to investigate the effect of network constraints on the operation of the wind farm.

6.2 Wind turbine

From the first vertical axis windmills found at the Persian-Afghan borders around 200 BC (Kaldellis and Zafirakis, 2011) to the first wind turbine built in 1887 by the professor James Blyth from Anderson College, now University of Strathclyde in Glasgow, Scotland (Poushpas, 2016); wind turbines have been brought into the front line of the global power generation scene.

Wind turbines extract the aerodynamic power from the wind and convert it into electricity. Modern wind turbines for megawatts scale have three blades with horizontal-axis configuration. Also, they are variable pitch blades and variable speed, which allows improving the power take-off efficiency. They have several subsystems, such as aerodynamics, drive train, tower, generator, pitch actuator, and the wind turbine controller. As shown in Figure 6.1 the rotor is on top of the tower where the wind has more energy and is less turbulent. The gearbox and the generator are inside a nacelle.

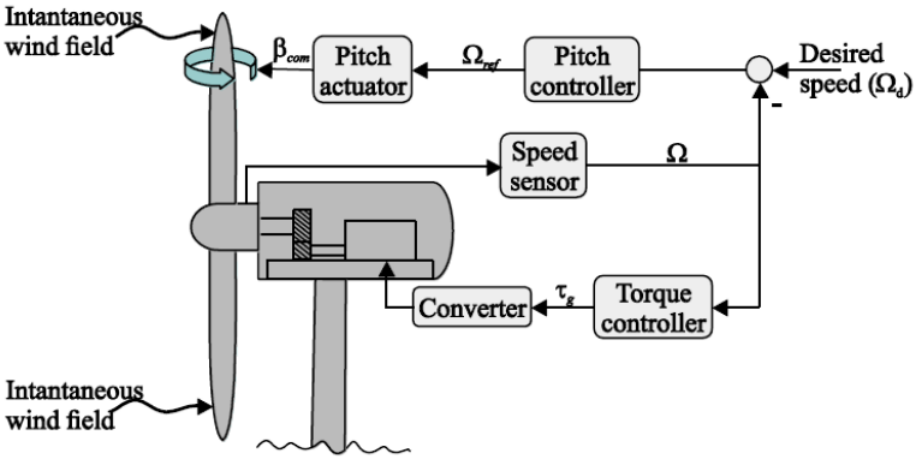

FIGURE 6.1: Wind turbine system (Njiri and Söffker, 2016)

Modern wind turbines are controlled by the electromagnetic torque of the generator, the pitch of the rotor blades and by the yaw of the nacelle. For research purposes, the last two will be omitted from this chapter.

Wind turbines are known to be inherently difficult to control due to the stochastic nature of the wind. Depending on the wind speed, the controller needs to adjust its mode of operation to an ideal wind turbine strategy. A variable-speed variable-pitch wind turbine has two operating regions. The partial load regime includes all wind speeds between the cut in wind speed and the rated wind speed (wind speed at which the system rated power is achieved). When the wind speed is above the rated wind speed and below the cut out wind speed the turbine is operating in the full load regime Soliman, Malik and Westwick, 2011.

Within these operation regions, four operating points are defined: Low wind speed (mode one), intermediate (mode two), high but below rate (mode three) and above rate (mode four). The first two operating points are linear and the last two are nonlinear. In modes one and two, torque control is used to regulate the power output. Meanwhile, in modes three and four pitch control is employed. Linear controllers can be designed for mode three and four and switched according to the operating point resulting in a full envelope nonlinear controller.

6.3 Wind farm control

With wind turbines becoming bigger and more expensive, wind farm control plays a key role in optimising energy capture. The control of wind farms optimises the functionality of a series of wind turbines in the wind farm. The wind field is a nonlinear system subject to perturbations due to the random nature of the wind (Spudic et al., 2010).

The centralised solution becomes very complex with the increasing number of wind turbines operating. These control strategies provide a hierarchical control configuration in which a central wind farm controller sends the reference signals to each local wind turbine controller (Hansen, Sørensen, Iov and Blaabjerg, 2006).

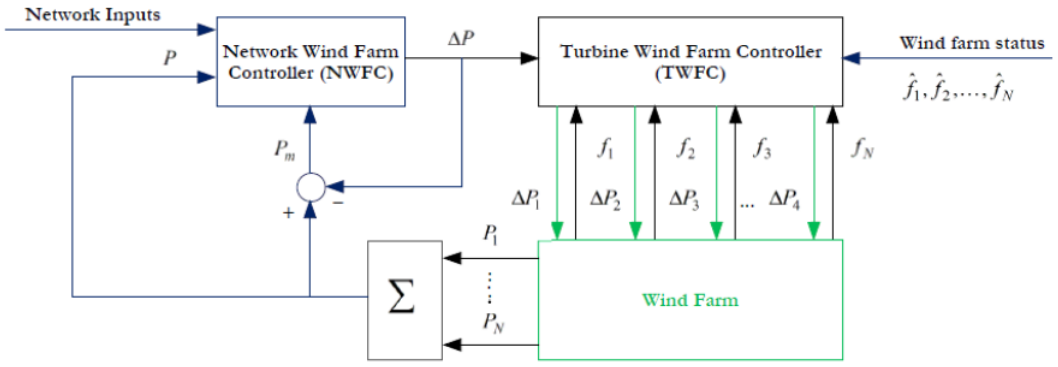

FIGURE 6.2: Control diagram of the wind farm (Hur and Leithead, 2016)

The wind farm controller is composed by the Network Wind Farm Controller (NWFC) and the Turbine Wind Farm Controller (TWFC) as depicted in Figure 6.2 (Hur and Leithead, 2016). The NWFC receives the required wind farm power from the network and the adjusted power output from the wind farm P and computes the required adjusted power ΔP relative to the unadjusted power P_m. The TWFC reads the state of the wind farm and the turbines therein to allocate adjustments of power ΔP_i for $i = 1, \ldots, N_t$ where N_t is the number of turbines. The flags f_i and the wind farm status $\hat{f}_i$ are values representing the operation mode of the wind turbines. Thus, TWFC calculates ΔP_i using the following equation:

$$\Delta P_i = \frac{\Delta P \, min(f_i, \hat{f}_i)}{\sum_{j=1}^{N_t} min(f_j, \hat{f}_j)} \tag{6.1}$$

The flags are set to values of 0, 1 or 3 depending the operating mode of the turbine, e.g. if a turbine is operating in below rate $f_i = 3$, if it is higher but below rate, $f_i = 1$ and

if it is operating in above rate, $f_i = 0$, which means there is no power adjustment. The wind farm status can be determined by the health, age and location of the turbines, however, it has been assumed $\hat{f}_i = 3$ for all i.

The generated power is computed as follows:

$$P_g = T_e * \omega_g \tag{6.2}$$

where P_g is the output power measured in watts (W), T_e is generator torque measured in Newton meters (Nm) and ω_g generator angular speed measured in radians per second (rad/s). The appropriate efficiency E_{ff} can be defined according to the specification. For the 5 MW exemplar wind turbine used in this work the value is 94.4%.

A Power Adjusting Controller (PAC) in the wind turbine adjusts the operation according to the reference signal sent by the wind farm controller. The power output of the wind turbines is adjusted using the torque demand and pitch demand signals.

Classical wind turbine control strategies use a set of two PI controllers to regulate the generator speed. In partial load operation, the generator torque is manipulated by the first PI controller so that the generator speed follows the reference. In the full load regime, the generator torque is fixed at its rated value, while the pitch angle is manipulated by the second PI controller to regulate the generator speed at its rated value. Gain scheduling is used to switch the operation of the PI controllers according to the variations in the aerodynamics. One of the main challenges in full load regime is the presence of severe fluctuations in the turbine power caused by random changes in the wind speed (Soliman, Malik and Westwick, 2011).

The network communication for monitoring and control system in wind farms is via Ethernet or serial communication. An optional redundant structure is also possible through a TCP/IP connection and redundant fibre optic. However, to the best knowledge of the author, network constraints such as loss of information have not been explored in this context. An effort to take into account the effect of the communication delay between the turbine wind farm controller and the wind turbines is reported by Hur and Leithead (2016). The authors demonstrated that a large delay caused considerable fluctuations in the wind farm power output. Delays and loss of information in a control scenario of a wind farm were measured for different network technologies

WLAN in Madsen et al., 2015. The effect of communication delay in hybrid power systems is highlighted in Singh, Samuel and Kishor, 2015.

A coordination based decentralised solution is offered in this work where each wind turbine controller shares its state to the other controllers and uses this information and the required adjusted power from the network to allocate its local adjustments of power.

6.4 Dynamics and model description

The model dynamics of variable speed wind turbines include the aerodynamics, the drive train dynamics and the power generation unit dynamics (Leithead and Connor, 2000).

The aerodynamic torque T_f, depends nonlinearly on the effective wind speed (the average over the rotor speed) U_w, the rotor speed, Ω, the hub speed V and the pitch angle of the rotor blades β. Linearising around a specific equilibrium point the aerodynamic model is given by:

$$\Delta T_f = K_v \Delta\Omega + K_\beta \Delta\beta + K_u \Delta U_w \tag{6.3}$$

where $K_v = \partial T_f/\partial V$, $K_\beta = \partial T_f/\partial \beta$, $K_u = \partial T_f/\partial U_w$ and Δ represents small variations in the variables respect to the nominal values.

The drive train dynamics include the rotor, high-speed shaft, gearbox, the load-speed shaft and mechanical load dynamics. The losses are represented as viscous damping of the shaft. The model is as follows:

$$\begin{bmatrix} V \\ \omega_g \end{bmatrix} = \begin{bmatrix} A(s) & B(s) \\ C(s) & D(s) \end{bmatrix} \begin{bmatrix} T_f \\ T_e \end{bmatrix} \tag{6.4}$$

where $A(s), B(s), C(s)$ and $D(s)$ are functions of the low and high speed shaft stiffness, the low-speed shaft external damping coefficient, the gearbox ratio and the rotor and generator inertia. The power generation unit dynamics contain the generator and the associated power electronics including the rectifier and the inverter required to connect to the grid. The model is depicted in Figure 6.3. The rectifier angle is α and ω_0, T_0 and α_0 are constants. The appropriate transfer functions $G_1(s)$ and $G_2(s)$ are defined by Leithead and Connor (2000) according to the electrical aspects of the power generation.

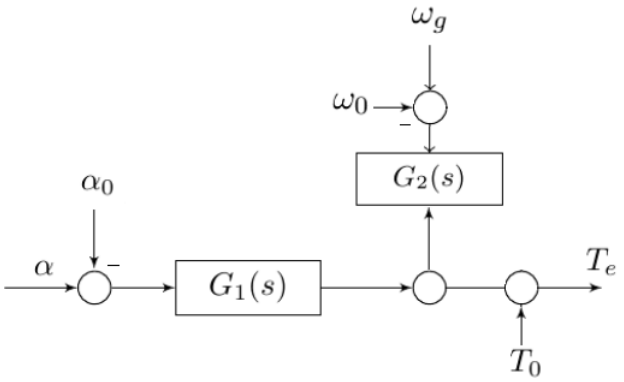

FIGURE 6.3: Model of power unit dynamics

The coupling between the tower and the drive train introduces a pair of complex Right Half Plane Zeros (RHPZ) into the model. The RHPZ impose constraints on the control of the generator speed using pitch control in above rated speed (Poushpas, 2016). When the mode of the rotor is coupled with the drive train mode and the tower mode, peaks in the frequency response are generated. The frequency of the tower modes depends on the turbine size. For the 5 MW model used in this work, the tower frequency is 1.76 rad/s.

Hur and Leithead (2017) combined the linearised aerodynamics, the drive-train dynamics and the power generation unit dynamics to obtain the model for the variable speed wind turbine and implemented in Matlab/Simulink. The parameters were chosen to correspond to the physical parameters of a Supergen (Sustainable Power Generation and Supply) 5 MW variable-speed pitch-regulated horizontal-axis wind turbine model, whose rated wind speed is approximately 11.5 m/s.

Consider the model of the turbine in state-space form as follows:

$$\begin{aligned} \Delta \boldsymbol{x}(k+1) &= \boldsymbol{A}\Delta \boldsymbol{x}(k) + \boldsymbol{B}_u \Delta \boldsymbol{u}(k) + \boldsymbol{B}_d\, d \\ \Delta \boldsymbol{y}(k) &= \boldsymbol{C}\Delta \boldsymbol{x}(k) \end{aligned} \tag{6.5}$$

where

$$\begin{aligned} \Delta \boldsymbol{y}(k) &= \boldsymbol{y}(k) - \boldsymbol{y}_{op}(k) \\ \Delta \boldsymbol{u}(k) &= \boldsymbol{u}(k) - \boldsymbol{u}_{op}(k) \\ \Delta \boldsymbol{x}(k+1) &= \boldsymbol{x}(k) - \boldsymbol{x}_{op}(k) \end{aligned} \tag{6.6}$$

$\boldsymbol{u}_{op}(k)$, $\boldsymbol{y}_{op}(k)$ and $\boldsymbol{x}_{op}(k)$ are the output, input and state operating points. The manipulated variable y is the generator speed. The generator torque controller is designed for the model above by considering the wind velocity between the inputs, $\boldsymbol{u}(k)$, as a disturbance d, and zero pitch angle. For the pitch controller, the input $\boldsymbol{u}(k)$ is the pitch angle, the wind speed will be treated as a disturbance and the generator torque will be set to a constant value (Hur and Leithead, 2017). Figure 6.4 illustrates the control diagram using this model. Note that $\boldsymbol{D}_d$ is assumed to be zero.

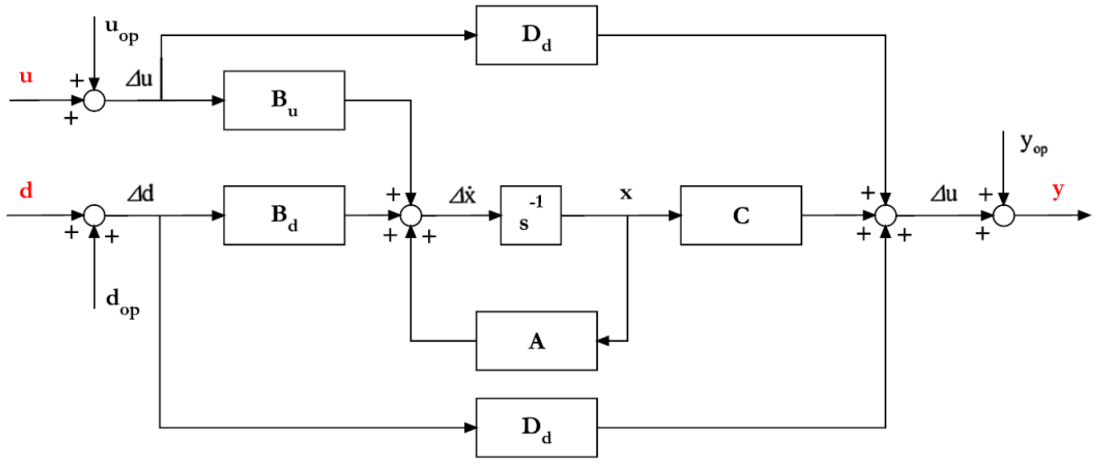

FIGURE 6.4: Control diagram using the wind turbine model (Hur and Leithead, 2017)

For simplicity in the notation, the variables $\Delta\boldsymbol{x}(k)$, $\Delta\boldsymbol{u}(k)$ and $\Delta\boldsymbol{y}(k)$ are called as $\boldsymbol{x}(k)$, $\boldsymbol{u}(k)$ and $\boldsymbol{y}(k)$ respectively, in the following analysis.

6.5 Control problem

The control of wind turbines is a complex problem that involves the collaboration of multiple disciplines including mechanical, aerodynamic and electrical engineering. The size of the structures of the wind turbines is increasing and becoming more expensive. To anticipate the cost increase, lighter components that can be more flexible are exploited. However, these elements may be more susceptible to fatigue failures and extreme loads arising from the chaotic nature of the wind. Therefore, optimisation of the control system is necessary to avoid excessive loads and to reduce fatigue load cycle (Aho et al., 2012).

Various control schemes have appeared during the past decades. For instance, PID control in turbine control and wind farm control is found, Linear–Quadratic–Gaussian control (LQG), modern optimal control for LFC, fuzzy control, adaptive and variable structure methods, robust approach, distributed /decentralised control approach and optimisation control strategy to decrease wear off for unbalanced load (Poushpas, 2016).

6.5.1 Torque control

Torque control has been widely used for tracking the maximum energy intake in the low wind speed operation. The generator torque is used as an estimation of the aerodynamic torque. A large torque unbalance is necessary to accelerate or decelerate the turbine because of its high inertia. This scheme relies on the intrinsically stable behaviour of the turbine in the low wind speed region as shown in Figure 6.5.

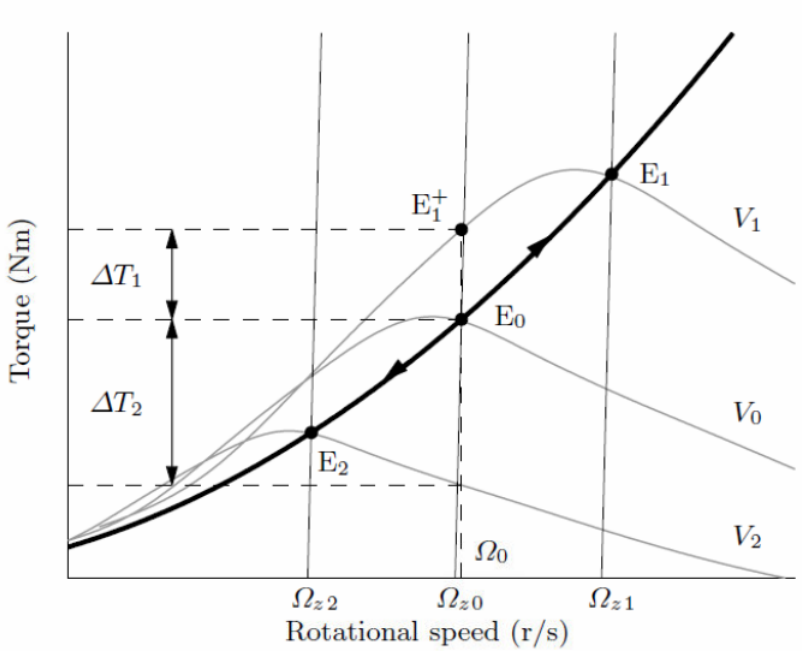

FIGURE 6.5: Torque versus velocity (Bianchi, Battista and Mantz, 2007)

If the wind speed is initially V_0 and the zero torque speed is Ω_{z0}, the turbine is operating at point E_0 on the locus of maximum conversion efficiency, also called maximum power or C_{pmax} locus. Now, suppose that wind speed increases suddenly from V_0 to V_1. The rotational speed cannot change abruptly because of the inertia of the drive train. Therefore, the generator keeps operating at the point E_0, whereas the aerodynamic

torque jumps to E_{+1} immediately after the wind speed step. As a consequence of this positive torque unbalance, the rotor and generator accelerate. The generator variables move from E_0 to E_1 along the C_{pmax} locus whereas the rotor variables evolve from E_{+1} to E_1 along the aerodynamic torque characteristic. Steady-state operation is recovered when both generator and rotor coincide at point E1. Thus, the new operating point of the turbine is E_1, which effectively belongs to the C_{pmax} locus.

If the wind speed falls from V_0 to V_2 leads to a negative net torque that decelerates the turbine until the new operating point E_2 is reached (Bianchi, Battista and Mantz, 2007).

6.5.2 DMPC

MPC has recently received significant attention for wind turbine control. The model predictive control is chosen because it can incorporate in the optimisation problem input and output constraints, time-varying process and predicted disturbance inputs (Yang, Li and Seem, 2015).

Decentralised solution and analysis are the future of control of large systems. In Chapter 5 the advances and benefits of applying DMPC to a large and complex system were highlighted. It was demonstrated that decentralised control with a coordination strategy improves the performance and stability while handling constraints and provides transparency of the information of the subsystems and the control actions taken by the controllers.

Large scale networked power systems are divided into smaller interacting generator subsystems. With the size and capacity of wind farms increasing in recent years, traditional centralised MPC solutions are not optimal because of the heavy computational load and limitations in exchanging information with large scale, geographically extensive control areas.

Distributed control is particularly suitable for controlling these systems because decentralised computation and constraints handling can be achieved. For instance, cooperation-based MPC method is used in distributed LFC. Recently, Distributed MPC for the LFC of a multiarea interconnected power system in the presence of variable speed wind turbine had been studied by Yan, Zhang and Liu (2015). Liu, Zhang and Lee (2017) proposed a Distributed MPC that shared measurements and prediction data between

controllers that were employed to achieve their local objective. Moreover, generation rate constraints, wind speed, pitch angle, and load input constraints had been considered. Also, Madjidian, Mårtensson and Rantzer (2011) proposed wind farm control for load mitigation using CPC. A distributed feedforward control was considered where the wind speed measurements from upwind turbines were shared between closest downwind neighbours.

In this work, the DWNMPC from Chapter 5 is applied to a wind farm of two turbines in which the study of the effect of dropouts in the network communication is performed. Two subsystems representing two wind turbines have been created with a decentralised cooperative strategy that allocates power according to the wind experienced by the turbines while maintaining the desired power production.

In a control system that responds quickly to the turbulent nature of the wind, the cycle life of the actuators needs to be guaranteed, for this, a constrained optimisation problem is proposed to keep the operation of the wind turbine within the input and rate constraints.

In wind farm operation, the performance and loads of downstream turbines are heavily influenced by the wake of the upstream turbines. This work offers a simplistic wind field model and the consideration of the awakening effect of the wind turbines is not take into account.

The potential benefits of the proposed solution for the networked power system are:

- The possibility to share measurements among other turbines allowing a decentralised solution to have transparency of the information of control actions and present states.
- Allow each turbine to adjust their power according to the wind farm status and its status.
- Introduce wireless communication with compensated effect of network constraints and improve the performance of the networked power system.

6.6 Control objective

The wind turbine operates over a wide wind speed range and the control objectives change over that range. Wind turbines have two operation requirements: at low speeds, is to capture the maximum wind energy and in high wind speeds to maintain the rated output power. Thus, the control problem is multiobjective. The objectives are:

- The principal objective is to manipulate the generator torque to guide the generator speed ω_g from an initial state to the desired steady state and keep it there regulated while rejecting the disturbance and respecting control and input constraints. Generator speed is controlled by varying the generator torque demand in low speeds.
- The second objective is to limit the effect of process model variations on the control system performance. For this, the cost function of the DMPC in (5.20) has been modified to include deviations between the process model and the real process.
- The third objective is to improve the performance of the control system by compensating the dropouts of the network communication. A decentralised KF has been added to the control scheme to allow the system to always have information of the sensor packets.
- The four objective is to fulfil the controller frequency response requirements. High-frequency roll-off is required to reduce actuator activity, and a crossover frequency around 0.6 to 2 rad/s is needed to minimise the standard deviation of the power output. Also, a high gain margin is desirable.

In this work, the DWNMPC is proposed to meet the first two control objectives. Moreover, the third objective has been achieved by enhancing the control systems with an estimation algorithm. The results in the next section shows that the last objective is partially reached since the results are within limits, but the robustness is not the optimal. However, this can be improved in a next step design by adding a tower and drive train damping to the control scheme.

6.7 Control design

The wind farm controller uses a coordinated decentralised solution, where the controller in each wind turbine exchanges their operation mode (flags) through the network. Using this information and the adjusted power ΔP required from the wind farm, the controller computes using (6.1) the adjusted power ΔP_i that the turbine can allocate. According to the resulting ΔP_i, the local MPC controller adjusts the torque to control the speed and meet the requirement of power.

The flags are exchanged during a selected time interval. For this an exchange loop that runs every ρ seconds has been implemented.

The design of the torque control is based on the DWNMPC explained in 5.2.2. Two modifications to the DWNMPC design have been necessary to control the wind turbine. Firstly, the cost function of the DMPC in (5.20) has been modified to include deviations between the process model and the real process, which are caused by the disturbances and model uncertainty. Secondly, to avoid saturation of the controller the control system works using the absolute value, i.e., $u_i(k)$.

The decentralised strategy considers that the whole system S represents the wind farm which is composed of n wind turbines represented by linear, discrete-time subsystems $S_i = 1, \cdots, n$. The state-space model of the turbine provided in (6.5) for a wind speed of 10 m/s is used to model each subsystem S_i and to design the local controller. The operating points are chosen as $\boldsymbol{u}_{op}(k) = 2.97 \times 10^4$ Nm and $\boldsymbol{y}_{op}(k) = 117$ rad/s.

Then, the state-space representation of S_i can be expressed as:

$$\begin{aligned} \boldsymbol{x}_i(k+1) &= \boldsymbol{A}_{ii}\boldsymbol{x}_i(k) + \boldsymbol{B}_{ii}\boldsymbol{u}_i(k) \\ \boldsymbol{y}_i(k) &= \boldsymbol{C}_{ii}\boldsymbol{x}_i(k) \end{aligned} \tag{6.7}$$

where the matrices $\boldsymbol{A}_{ii}, \boldsymbol{B}_{ii}$ and $\boldsymbol{C}_{ii}$ can be found in the Appendix D.

For each subsystem S_i the estimation of the current states $\hat{\mathbf{x}}_i(k)$ is computed using a decentralised KF. The KF design is defined in the next section.

6.7.1 Decentralised estimation for dropouts

Consider the decentralised KF for subsystem i denote by:

$$\begin{aligned}\hat{\mathbf{x}}_i(k+1|k) &= \hat{\mathbf{x}}_i(k|k-1) + \gamma_i(k)\,\mathbf{K}_{fi}(k)\,[y_i(k) - C_{ii}\hat{\mathbf{x}}_i(k|k-1)] \\ \hat{\boldsymbol{y}}_i(k) &= C_{ii}\hat{\mathbf{x}}_i(k|k-1)\end{aligned} \tag{6.8}$$

where the index $(k|k-1)$ refers to the information at sampling time k given observations up to and including time $k-1$. The KF gain for each local observer is denoted as K_{fi} and the variable $\gamma_i(k)$ is defined as follows:

$$\gamma_i(k) = \begin{cases} 1 & \text{if } y_i(k) \text{ is received at time } k \\ 0 & \text{if } y_i(k) \text{ is not received} \end{cases} \tag{6.9}$$

During the presence of dropouts, only the equations for estimation (prediction) of the KF are computed. Once the information $y_i(k)$ is available the measurement (update) equations are calculated.

6.7.1.1 Prediction

In the proposed decentralised KF, each local filter should estimate $x_i(k)$ such that the state estimate error for subsystem i, i.e. $\mathbf{e}_i(k) = \mathbf{x}_i(k) - \hat{\mathbf{x}}_i(k)$ is minimised. The one step-ahead prediction is expressed as follows:

$$\hat{\mathbf{x}}_i(k+1|k) = \mathbf{A}_{ii}\hat{\mathbf{x}}_i(k|k-1) + \mathbf{B}_{ii}\Delta\mathbf{u}_i(k) \tag{6.10}$$

Define $P_i(k|k-1)$ as the covariance of the state estimate error:

$$\mathbf{P}_i(k|k-1) = \mathbb{E}\left[\mathbf{e}_i(k|k-1)\mathbf{e}_i^{\mathrm{T}}(k|k-1)\right] \tag{6.11}$$

Then, the noise covariance matrices are: $\mathbb{E}\left[\mathbf{w}_i(k)\mathbf{w}_i^{\mathrm{T}}(k)\right] = \mathrm{Q}_{fi}(k)$, $\mathbb{E}\left[\mathbf{v}_i(k)\mathbf{v}_i^{\mathrm{T}}(k)\right] = \mathrm{R}_{fi}(k)$ respectively. $\mathbb{E}[\cdot]$ denotes the expectation of the argument. Thus:

$$\mathbf{P}_i(k|k-1) = \mathbf{A}_{ii}\mathbf{P}_i(k-1|k-1)\mathbf{A}_{ii}^{\mathrm{T}} + \mathrm{Q}_{fi}(k)^{\mathrm{T}} \tag{6.12}$$

The equations (6.10) and (6.12) are forward projections of state and covariance for a priori estimation. During the dropouts, there is not update of the information.

6.7.1.2 Measurement update

Once the information is available, the prediction is combined with the current observation information. Consider the decentralised KF for subsystem i in (6.8). The modified innovation covariance is:

$$\mathbf{\Omega}_i(k) = \mathbf{C}_{ii}\mathbf{P}_i(k|k-1)\mathbf{C}_{ii}^T + \mathrm{R}_{fi}(k) \tag{6.13}$$

Define the innovation or measurement residual for each subsystem i as:

$$\Lambda_i(k) = \mathbf{y}_i(k) - \mathbf{C}_{ii}\hat{\mathbf{x}}_i(k|k-1) \tag{6.14}$$

Now, updating the estimate error covariance as:

$$\mathbf{e}_i(k+1|k) = \mathbf{x}_i(k+1|k) - \hat{\mathbf{x}}_i(k+1|k) \tag{6.15}$$

Results in:

$$P(k+1|k) = \mathbb{E}\left[\mathbf{e}_i(k+1|k)\mathbf{e}_i^{\mathrm{T}}(k+1|k)\right] \tag{6.16}$$

Substituting the state equation in (5.1) and the estimate state equation in (6.8) and solving the estimation yields that:

$$\begin{aligned}\mathbf{P}_i(k+1|k) = {} & \mathbf{A}_{ii}\mathbf{P}_i(k|k-1)\mathbf{A}_{ii}^{\mathrm{T}} - 2\mathbf{K}_{fi}(k)\mathbf{C}_{ii}\mathbf{P}_i(k|k-1)\mathbf{A}_{ii}^{\mathrm{T}} \\ & + \mathbf{K}_{fi}(k)\mathbf{\Omega}_i(k)\mathbf{K}_{fi}^{\mathrm{T}}(k) + \mathrm{Q}_{fi}(k)^{\mathrm{T}}\end{aligned} \tag{6.17}$$

The minimum error covariance is obtained by calculating $\partial\mathbf{P}_i(k+1|k)/\partial\mathbf{K}_{fi}(k) = 0$. One can find that the following equation should be satisfied:

$$-2\,\mathbf{A}_{ii}\mathbf{P}_i(k|k-1)\mathbf{C}_{ii}^T + 2\mathbf{K}_{fi}(k)\mathbf{\Omega}_i(k) = 0 \tag{6.18}$$

Thus, the KF gain is:

$$\mathbf{K}_{fi}(k) = \mathbf{P}_i(k|k-1)\mathbf{C}_{ii}^T\mathbf{\Omega}_i^{-1}(k) \tag{6.19}$$

The algorithm is a function of $\gamma_i(k)$, which is a binary random number of 0 or 1. As a consequence, the proposed decentralised KF has a time-varying gain where the optimal gain and the error covariance values are calculated at each time sampling. If necessary, the KF parameters $\mathrm{R}_{fi}(k), \mathrm{Q}_{fi}(k)$ can be tuned to minimise the innovation error and improve the estimation algorithm.

6.7.2 Prediction equations

The predictions of the output are estimated over the prediction horizon by replacing one-step-ahead prediction in (6.7) recursively. To limit the effect of process model variations on the system performance, an additional term d_{si} has been included in the output predictions. The offset $d_{si} = y_i - \hat{y}_i$ corrects deviations between the process model and the real process. Thus, the prediction equations are given by:

$$\hat{\mathbf{Y}}_i(k) = \mathbf{F}_i \hat{\mathbf{x}}_i(k) + \phi_{ii}(k)\mathbf{U}_i(k) + \boldsymbol{L} d_{si} \tag{6.20}$$

where

$$\begin{aligned}
\hat{\mathbf{Y}}_i(k) &= \left[\mathbf{y}_i^{\mathrm{T}}(k+1)\ \mathbf{y}_i^{\mathrm{T}}(k+2)\ \ldots\ \mathbf{y}_i^{\mathrm{T}}(k+N)\right]^{\mathrm{T}} \\
\mathbf{U}_i(k) &= \left[u_i^{\mathrm{T}}(k)\ u_i^{\mathrm{T}}(k+1)\ \ldots\ u_i^{\mathrm{T}}(k+N_{\mathrm{u}}-1)\right]^{\mathrm{T}} \\
\mathbf{F}_i &= \left[(\mathbf{C}_{ii}\mathbf{A}_{ii})^{\mathrm{T}}\left(\mathbf{C}_{ii}\mathbf{A}_{ii}^2\right)^{\mathrm{T}} \ldots \left(\mathbf{C}_{ii}\mathbf{A}_{ii}^N\right)^{\mathrm{T}}\right]^{\mathrm{T}} \\
\boldsymbol{L} &= \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T_{N\times 1}
\end{aligned} \tag{6.21}$$

$$\phi_{ii}(k) = \begin{bmatrix} \mathbf{C}_{ii}\mathbf{B}_{ii} & 0 & \ldots & 0 \\ \mathbf{C}_{ii}\mathbf{A}_{ii}\mathbf{B}_{ii} & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & 0 \\ \mathbf{C}_{ii}\mathbf{A}_{ii}^{N-1}\mathbf{B}_{ii} & \ldots & \ldots & \mathbf{C}_{ii}\mathbf{A}_{ii}^{N-N_{\mathrm{u}}}\mathbf{B}_{ii} \end{bmatrix}$$

6.7.3 Optimisation problem

The quadratic cost function in (5.19) has been modified to use the absolute value of the input instead of the incremental value, resulting in the following objective function:

$$\begin{aligned}
&\min_{u_i(k)} J_i = \sum_{j=1}^{N} \|\mathbf{y}_i(k+j|k) - \mathbf{r}_i(k+j|k)\|^2_{\mathbf{Q}_i} + \sum_{j=0}^{N_{\mathrm{u}}-1} \|u_i(k+j|k)\|^2_{\mathbf{R}_i} \\
&\text{subject to} \\
&u_{i,\min} \le u_i(k+j) \le u_{i,\max} \\
&\Delta u_{i,\min} \le \Delta u_i(k+j) \le \Delta u_{i,\max} \quad \text{for } j = 0, 1, 2, \ldots, N_u
\end{aligned} \tag{6.22}$$

The input constraints will guarantee the contributions of control input and rate input are applied according to the turbine limitations.

Following the procedure in Section 5.2.3, the control solution can be found by minimising the following objective function:

$$J_i = [\hat{\mathbf{Y}}_i(k) - \mathbf{R}_i^0(k)]^T \bar{\mathbf{Q}}_i [\hat{\mathbf{Y}}_i(k) - \mathbf{R}_i^0(k)] + \mathbf{U}_i^T(k) \bar{\mathbf{R}}_i \mathbf{U}_i(k) \tag{6.23}$$

Note that $\mathbf{R}_i^0, \bar{\mathbf{Q}}_i, \bar{\mathbf{R}}_i$ were previously defined in Section 5.2.3.

The quadratic function can be expressed as:

$$J_i = \quad \frac{1}{2}\, \mathbf{x}_i^{\mathrm{T}}\, \mathbf{H}_i\, \mathbf{x}_i + \mathbf{G}_i^T\, \mathbf{x}_i \tag{6.24}$$

where

$$\begin{aligned} \mathbf{x}_i &= \mathbf{U}_i(k) \\ \mathbf{H}_i &= 2[\phi_{ii}^{\mathrm{T}}(k) \bar{\mathbf{Q}}_i \phi_{ii}(k) + \bar{\mathbf{R}}_i] \\ \mathbf{G}_i &= 2\phi_{ii}^T(k) \bar{\mathbf{Q}}_i [\hat{\mathbf{Z}}_\mathbf{i}(k) - \mathbf{R}_i^0(k)] \end{aligned} \tag{6.25}$$

with

$$\hat{\mathbf{Z}}_i(k) = \mathbf{F}_i \hat{\mathbf{x}}_i(k) + \boldsymbol{L} d_{si} \tag{6.26}$$

Thus, the quadratic programming is given by:

$$\begin{aligned} &J_i = \mathbf{U}_i^{\mathrm{T}}(k) \mathbf{H}_i \mathbf{U}_i(k) \; + \mathbf{G}_i \mathbf{U}_i(k) \\ &\text{subject to } \boldsymbol{C}_{ci}\, \boldsymbol{U}_i \leq \boldsymbol{d}_{ci} \end{aligned} \tag{6.27}$$

The problem is subject to input and rate input constraints that can be added as explained in Section 5.2.4.

6.8 Wind farm control algorithm

The wind farm control algorithm is implemented using the following steps.

Step 1: Initialisation

(a) Configure the network parameters and percentage of dropouts.

(b) Allocate the number of wind turbines as the number of subsystems

$S_i = 1, \ldots, n.$

(c) Select the prediction horizons $N = N_u$.

(d) Add the wind speed model as a disturbance.

Step 2: Off-line calculation

(a) Compute the prediction matrices using the wind turbine state-space model (6.7).

(b) Add the constraints according to the turbine specifications.

(c) Compute the time-invariant part of the constraint matrices.

Step 3: Control loop kTs

(a) Read the generator velocity from the wind turbine.

(b) Compute the present state vector using the decentralised KF.

(c) Read the wind farm reference power from the network and wind turbine's status from other agents.

(d) Combine the local status with those of the other agents and calculate the new local power reference using (6.1).

(e) Adjust the operation according to the reference signal.

(f) Compute the control signal by solving the constrained optimisation problem in (6.22) *and apply the generator torque to the wind turbine* S_i.

(g) Check performance and robustness responses and fine-tune the weight R *and KF parameters if necessary.*

Step 4: Data exchange loop ρTs

(a) Send the wind turbine status to other agents.

6.9 Simulation results

To demonstrate the effectiveness of the DWNMPC, a study of the effect of dropouts in the network communication is proposed for a wind farm of two turbines. The stochastic wind speed model proposed by Hur and Leithead (2016) has been used here. The wind speed is depicted in Figure 6.6.

Two subsystems representing two wind turbines have been created in the Truetime simulator using a sampling time $Ts = 0.02\,s$. The linearised model of the wind turbine is used to predict the future outputs of each subsystem.

The predictions horizon are $N = N_u = 30$. The weights have the following values: $R_i = 1 \times 10^{-8}$ and $Q_i = 1$ for $i = 1, 2$. The constraints are set as $u_{max,i} = 1 \times 10^{13}$, $u_{min,i} = 0$ and $\Delta u_i = 60$ for $i = 1, 2$. The noise covariances are chosen as: $Q_{fi} = 2.3$, $R_{fi} = 1 \times 10^{-8}$ for $i = 1, 2$. The percentage of dropouts from sensors to controllers is $P_{loss} = 48\%$ and from controllers to sensors is $P_{loss} = 2\%$. The reference is set as 117 rad/s. The time interval to exchange the state trajectory and input predictions between agents is given by $\rho = 0.02\,s$. The generator speed, the KF estimation and torque of the two turbines are shown in Figure 6.7 and Figure 6.8. When the turbines are experiencing lower wind speed, for much of the time, generate less power. The results are satisfactory as the fluctuations remains below 12%, which is often within the controller design specification. It can be seen that the input constraints are also satisfied.

The top subplot, which is the generator velocity of the first turbine, is rescaled at around 20 s in Figure 6.9. The KF estimation is close to the process. Therefore, when a dropout from sensor to controller occurs, the KF provides an accurate estimate of the process output. The accuracy of the KF estimation requires precise modelling of the wind turbine and the wind acting as a disturbance. Further tests show that the generator velocity reference can be decreased to 80 rad/s and the control system effectively moves the manipulated variable to the new point. However, in this case, the minimum constraint input is activated in several periods.

When the operating points are 80 rad/s and 110 rad/s, a new test where the power output of the turbine is increased 5% is studied. The torque is used to adjust the velocity and the generated power according to (6.2).

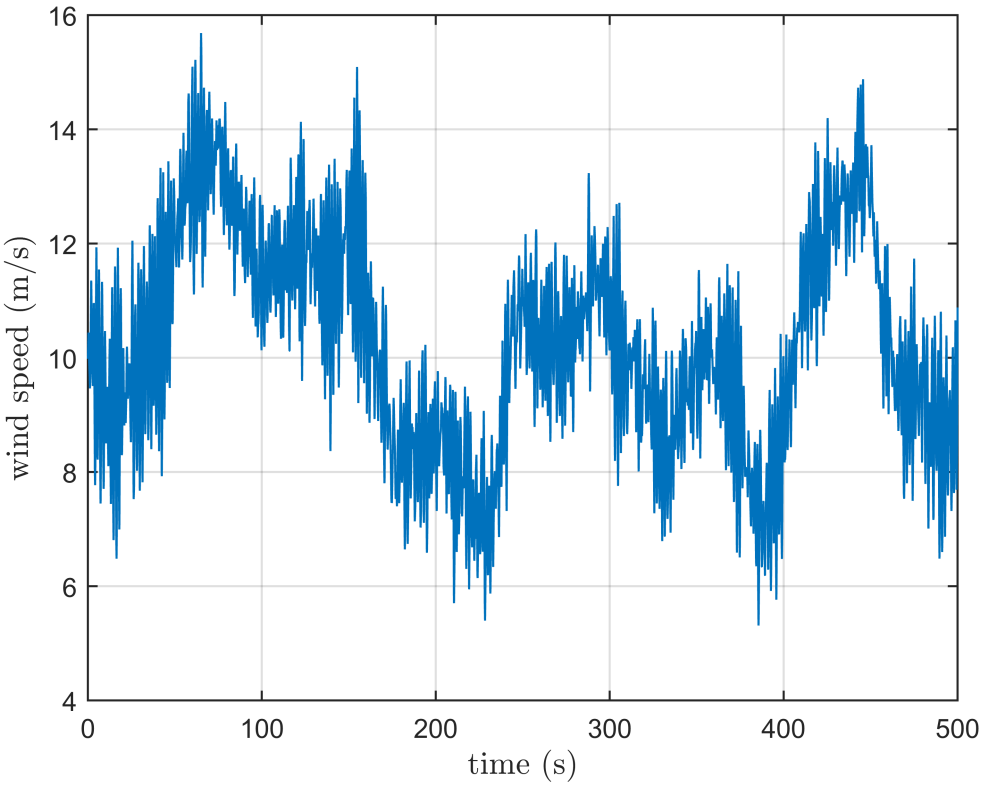

FIGURE 6.6: Effective wind speed at a mean wind speed of 10 m/s

To decide on the allocation of the adjusted power, each controller communicates its flag status to the other controllers. Considering that information and its local flag status, the controller computes the new reference using (6.1). In this case, the adjusted power of the turbine one is $\Delta P_1 = 3/4\,\Delta P$, which is three times larger than the one of turbine two. The reason is that the flag status of the first turbine is $f_1 = 3$, which indicates it is working below rate and therefore a bigger allocation of power can be done. Meanwhile, the second turbine status is $f_1 = 1$, which indicates it is operating closer to the limit rate and therefore, less allocation of adjusted power can be made, $\Delta P_2 = 1/4\,\Delta P$. Simulation results for both wind turbines are shown in Figures 6.10 and 6.11. The RMS generator speeds are 79 rad/s and 107 rad/s for turbine one and two respectively. Fluctuations are under 12% and the RMS power values are 1.6 MW and 2.1 MW.

It is evident from the figures that the power is varying at the below rate speed. Due to the random behaviour of the wind the system trajectory is stochastic and the control

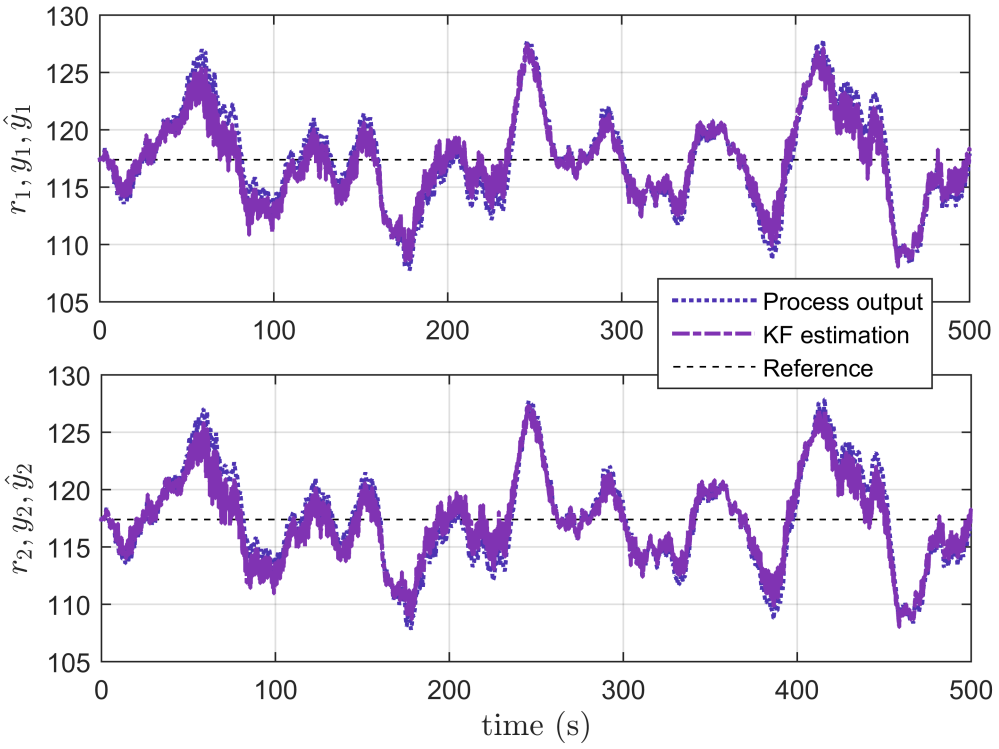

FIGURE 6.7: Generator speeds y_1, y_2 (rad/s)

becomes very challenging. Power reference tracking is possible in above rated operation, where the available power is greater than the reference power. As explained earlier, pitch control is applied to control the generator speed and therefore the power in above rated speeds.

6.10 Performance analysis

Additional tests are performed where the IAE criterion is measured for various percentages of dropouts. Table 6.1 lists the results of the IAE criteria and the innovation errors for different dropouts scenarios. The results assured that the decentralised solution with dropouts and noise is effective.

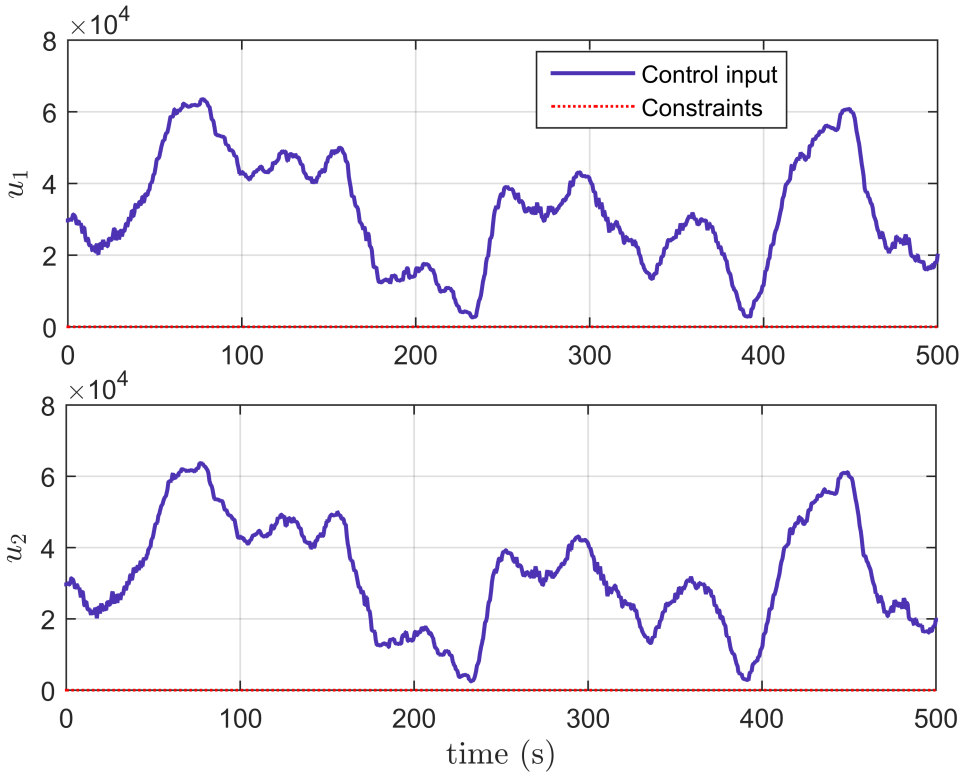

FIGURE 6.8: Generator torques u_1, u_2 (Nm)

TABLE 6.1: Performance indexes for different scenarios of percentage of dropouts

Values	$P_{loss} = 45\,\%$	$P_{loss} = 80\,\%$
IAE	8.59×10^4	1.14×10^5
Innovation error	2.43×10^4	1.65×10^5

6.11 Robustness analysis

The frequency responses of the open-loop, the sensibility and the complementary sensitivity are investigated to analyse the stability and robustness of the design. The wind controller requires a high-frequency roll-off to reduce actuator activity, and the gain crossover frequency should be between 0.6 and 2 rad/s to minimise the standard deviation of the power output (DJ and WE, 1996).

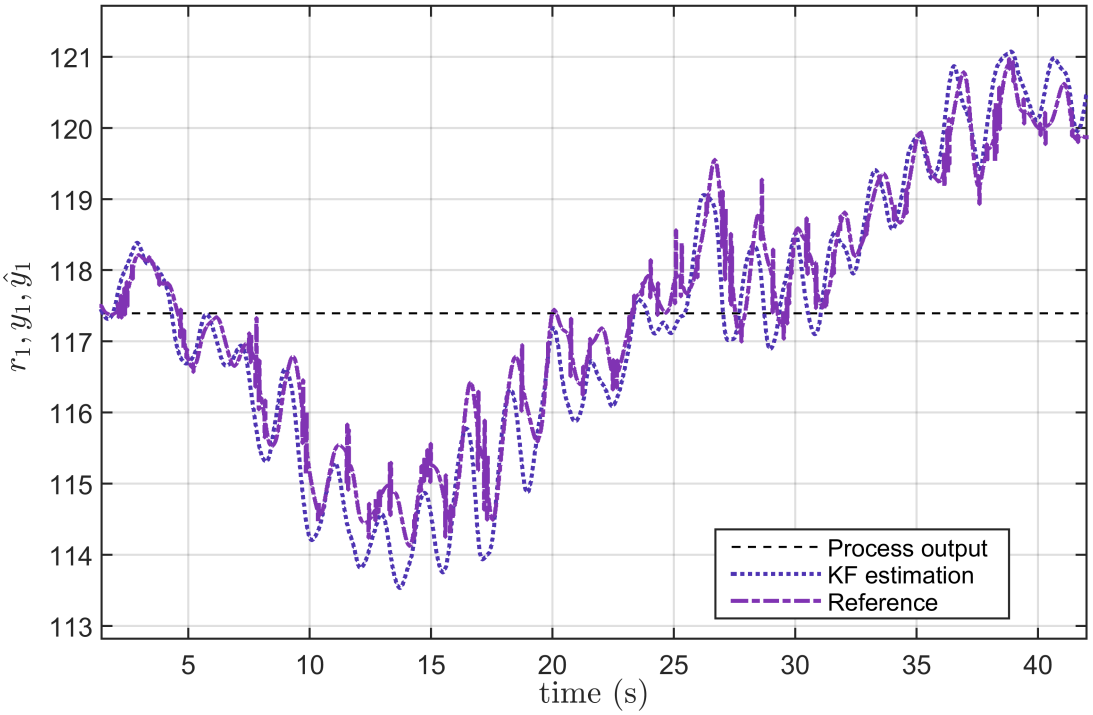

FIGURE 6.9: Rescaled plot of Figure 6.7

The open-loop frequency response is computed using the standard unity feedback system. The open-loop transfer function is given by:

$$G_{ol}(s) = C(s)G_p(s) \tag{6.28}$$

where $C(s)$ and $G_p(s)$ are the transfer functions of the controller and the turbine model, respectively. The input and output are the torque and generator speed, respectively. The sensitivity function has been defined in (3.2). The complementary sensitivity function is defined as:

$$T(s) = \frac{C(s)G_p(s)}{1 + C(s)G_p(s)} \tag{6.29}$$

where $T(s) + S(s) = 1$.

Figure 6.12 shows the frequency responses. From the dash-dotted line, the open-loop is stable. Phase margin is around 77.2 deg and gain margin is 18.2 dB. The gain crossover frequency is 0.64 rad/s within the specification range for an optimal control action. The closed-loop frequency response has been investigated using the complementary sensitivity response (solid line). The desired unitary gain at low frequency is

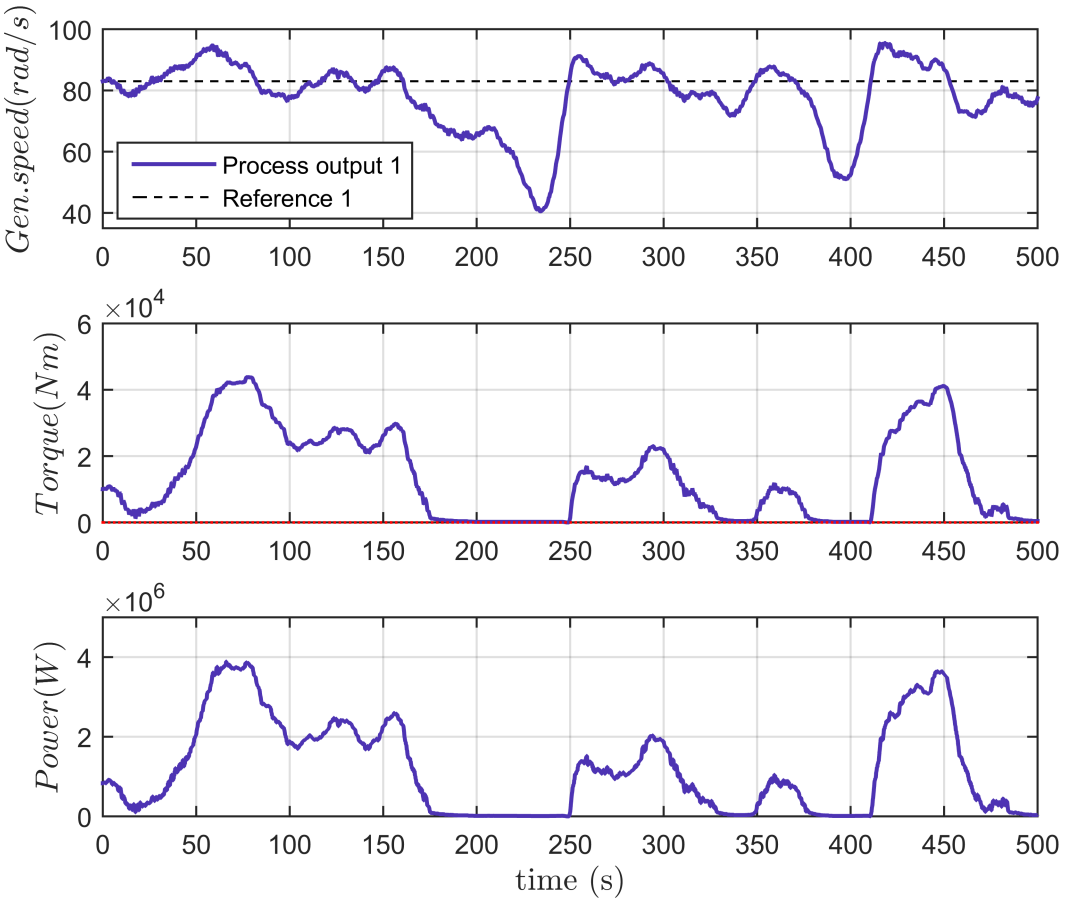

FIGURE 6.10: Wind turbine 1 response for adjusted power

achieved. However, a higher degree of roll-off is desirable to remove the noise or high-frequency disturbances. The sensitivity plot (dashed line) shows that the low-frequency disturbances will be attenuated by the system and there is no amplification of signals.

The peak above 0 dB in the open-loop response indicates the sensitivity to noise. The KF has been tuned to decrease the peak as much as possible. Low Qf/Rf attenuates high frequency and increases phase lag. Increasing the MPC weight R reduces the DC gain of the system. However, it does not affect reducing the peak magnitude. The prediction horizon N can also be used to manipulate the DC gain. The peak corresponds to the drive train dynamics, and it is typically removed by applying a drive train damper. However, the design of the required filter is the next step in the wind control strategy that is not discussed in this work.

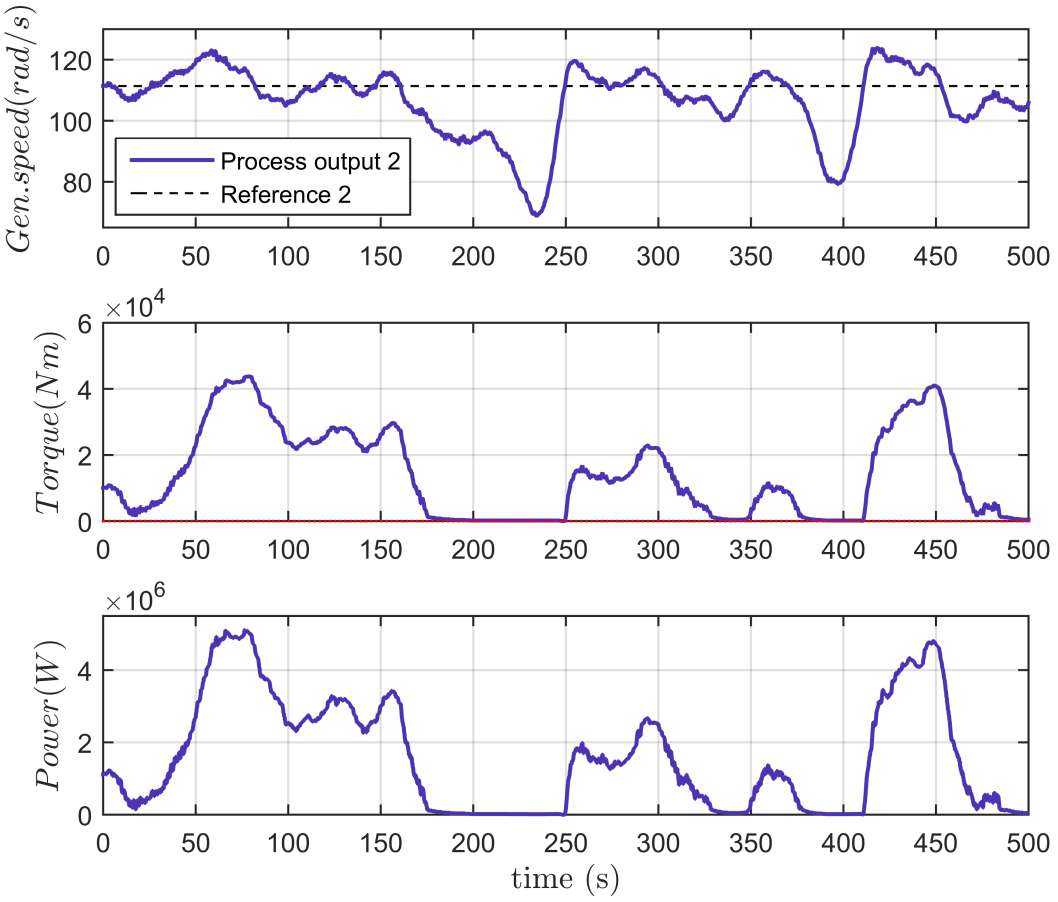

FIGURE 6.11: Wind turbine 2 response for adjusted power

The lower figure 6.12 shows that there is a quick change in the phase of the system at a frequency of approximately 1.73 rad/s, which corresponds to the tower mode (Stock, 2015). The next change at a frequency of 3 rad/s obeys to the drive train operation, which introduces oscillations in the response. However as discussed before, the system is stable, and the robustness can be improved adding filters to the control scheme that compensate the tower and drive train effects.

6.12 Discussion

Figures 6.10 and 6.11 illustrated that using the proposed solution the rate constraints are not activated during the wind farm operation for a reference of 117 rad/s. Therefore,

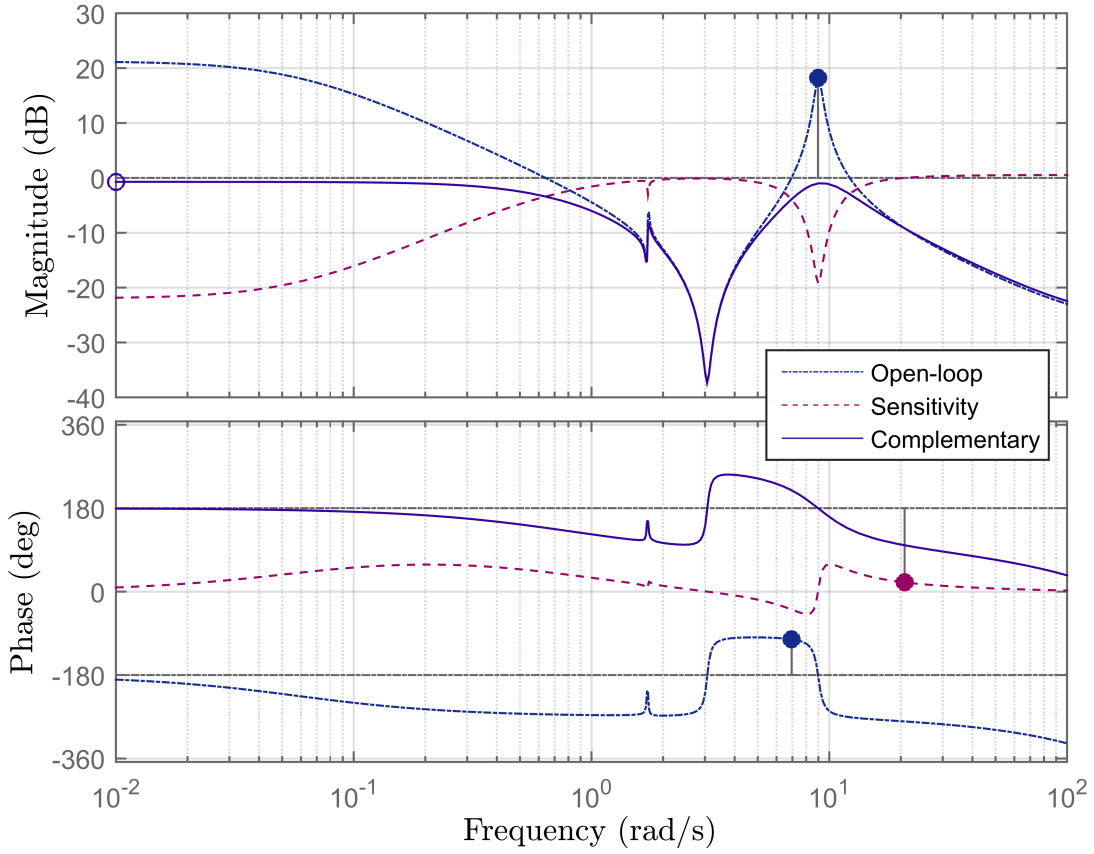

FIGURE 6.12: Open-loop, closed-loop and sensitivity frequency responses

it is possible to keep the system variables within the safe operating limits and protect the actuators from fatigue.

Since there is always a trade-off between performance and robustness, the controller has a good performance but does not achieve the best robustness possible. The dynamics of the mechanical aspect of the power generator introduce changes in the magnitude and phase of the open-loop response at approximately 3 rad/s. Additionally, the peaks above 0 dB can amplify noise in the system.

Simulation results have been considered for a configuration of two wind turbines due to the high computational demand. However, the number of turbines can be increased. It has been demonstrated that the WNCS compensates up to 80% of dropouts. Therefore, additional dropouts cause for increasing the number of nodes sharing the network will not be as significant and will be compensated by the proposed estimation algorithm.

6.13 Summary

Simulations results demonstrated that the control strategy could meet the multiple objectives of regulating generator speed and power using torque control. The application of the DNMPC to a wind farm provided results within the desired requirements of fluctuations less than 12% and frequency response. The proposed controller provided good performance while compensating for high percentages of dropouts. Up to 80% of information loss can be afforded and still ensure the stability of the control system.

The cost function of the DMPC has been modified to include the offset between the output of the process model and the real process, which is caused by the disturbances and model uncertainty. In addition, the design works using the absolute value to avoid saturation of the control input.

KF parameters have been fine-tuned to improve the performance and robustness of the proposed solution. However, due to oscillations of the mechanical dynamics of the power generator unit, the robustness of the control systems was not optimal. The robustness of traditional wind turbine control systems is enhanced by adding filters to perform tower and drive train and damper compensation to the control scheme. To improve the robustness of the proposed solution it was required a robust control formulation of the MPC optimisation problem.

A torque control for low speeds has been investigated in this work. Another linear control can be designed to operate at higher speeds, using active pitching to control the generator speed and electrical power. In a next step of the control strategy, a switching control can be developed to give a smooth transition from one controller to another according to the wind speed resulting in a full envelope controller.

Chapter 7

Conclusions and future work

7.1 Conclusions

This thesis was concerned with the design of control strategies for WNCS with the main focus on MBPC and PID control with the aim of creating control laws suitable for compensating delays and dropouts. The main objective was to analyse the performance that these algorithms can offer to the challenging problem of dropouts and time-varying delays. This thesis includes two main parts. The first part is a comparative study of existing control methodologies presented in Chapter 2. It defines the problems of the network that still requires further research and points out some possible research directions. The second part is the control design using mainly predictive structures for SISO and MIMO systems presented from Chapter 3-6. To this end, suitable models of the network behaviour which included the dropouts and delays were presented. Several simulations using the approximations under different scenarios of network constraints were performed and compared with the results from a network simulator in Chapter 3. It was found that the models are suitable for the controller design. PID, IMC and Smith Predictor algorithms were developed to meet design requirements, such as margins of robustness, performance criteria and stability conditions. Predictive PID controllers with similar properties to MBPC methods were created in Chapters 4. Complex industrial systems were addressed in Chapter 5 and 6. The control systems were designed to adapt to the requirements of the network, the decentralised solution, reduced data communication, constrained systems and to meet requirements of robustness, performance and efficiency. All tests included disturbance and noise which were rejected

by including the incremental variables in the process model of the MBPC. Also, by performing state augmentation, the tracking performance could be improved. The estimation algorithms based on KF dealt with the process and measurements noises. This chapter summarises the main conclusions of this thesis and suggestions for future work.

NCS are gaining importance in improving the control of simple and complex industrial systems. However, the performance degradation with the introduction of the network is still a concern to address before the systems can be reliable for process control. The need for flexibility and stability supports the interest of PID control, Smith predictor and IMC control to operate on NCS. These simple control algorithms offer the opportunity to fully deploy NCS in industrial applications.

These strategies rely on optimisation problems that incorporate robustness and performance constraints to compute the optimum control signal at every sampling time. The simulations using approximations of Gamma and Gaussian distributed functions to model the random and time-varying delay showed satisfactory results for the design of the controller. The comparison between the experiments using the approximations and the ones using the Matlab/Simulink Trutime simulator showed that the assumption of time-varying delays with maximum and minimum limits was accurate. Similarly, assuming that the percentage of dropouts was variable, but limited the critical value of the dropout percentage to ensure system stability can be found. However, it requires incorporating an accurate model of the process and small delays and percentages dropouts to execute the task timely. Long delays and consecutive dropouts adversely affect the stability of the control process and can be not overcome by fine-tuning of the controller parameters. Also, only limited variations of the process model parameters are allowed. The seven proposed methods showed adequate compensation of the network constraints with good performance and robustness.

Despite the outstanding advantages of the investment, the monitoring and security aspects of the installation and maintenance of WNCS, the introduction and exploration of wireless communication have not had a profound impact on process control. The main contribution of the thesis was to prove that simple and flexible control strategies using a simple model of the network behaviour were robust against dropouts and time-varying delays in WNCS. The characteristics of wireless communication using limited bandwidth networks based on standards such as 802.11b have been tested and demonstrated the effectiveness of the reported strategies. It is believed that these results

enable the possibility of further extension of wireless technologies in industrial applications.

The limitations of the simple PID controllers that were proposed in Chapter 3 have been overcome with the inclusion of predictive capabilities to the PID and estimation algorithms. The lack of information from the sensor to the controller can be solved using KF. The receding horizon approach was exploited to compensate consecutive dropouts from sensor to controller and from controller to actuator. $N_u - 1$ predictive control signals were calculated and saved at each sampling instant. If sensor or controller packets were dropped at present k, the control system uses the estimate of the control signal stored from the previous sampling time $k - 1$. The methodology requires a selection of the prediction horizon that coincides with the maximum number of consecutive dropouts. The strategies using MBPC showed that the algorithm could be successfully adapted to the networked environment and estimate high percentages of the data loss. From the four reported predictive methodologies, the approach using augmented state-space model produces a very efficient controller to ensure tracking response for limited percentages.

The inclusion of input constraints showed that MBPC was powerful enough to find a solution when time delays and dropouts occurred, if reliable process information was available. This last part was covered by the implementation of a KF estimation that is based on an accurate process model. The constraint handling produces a reduction of the performance, but satisfactory results were still found. In most cases, a faster weight λ can improve the sluggish response of the control signal. However, there are some cases where the control strategy can not stabilise faster responses with high percentages of dropouts. In DMPC the constrained problem with long prediction horizon increases the computation time and the number of iterations to find the minimum due to the complexity of the cost function, which will affect systems with faster dynamics.

Since the proposed design was based on a model of the plant, the closed-loop system showed stability for a limited percentage of variations of the model process dynamics and percentage of dropouts. Simulations of stability, performance and robustness analysis demonstrated the applicability of model uncertainty up to 35% and percentages of dropouts up to $P_{loss} = 80\%$.

DWNMPC design considered the division of the controlled process into n subsystems each of which has appropriately integrated the network characteristics. The effect of the

interactions presented a challenging problem when strong interactions were considered. This has been successfully overcome by the computation of predictions of the interactions and the inclusion of a coordination strategy that exchanges the state trajectory and control input trajectory between the subsystems. The state and input trajectories were used to compute the interactions predictions. A good closed-loop performance was obtained through optimisation of the decentralised KF parameters and the fine-tuning of the MBPC weights. Once this was done, the sampling time of the internal loop for the exchange of information can be extended long enough and the prediction horizon can be chosen small to reduce computation time. An analytical study demonstrated performance was strictly related to the accuracy of the decentralised KF estimation.

The control strategy allowed to increase the performance, reduce the effect of interactions and the fact that the exchange of data could be carried out by a greater sampling time decreased the congestion in the network and the possibility of long delays and dropouts. Moreover, wireless nodes energy can be saved through reduced data exchange.

A case of study of a wind farm was also considered. Two interconnected subsystems representing two wind turbines have been created with a decentralised cooperative strategy that allocates power according to the wind experienced by the turbines while maintaining the desired power production. Each controller communicates its status to the neighbouring controllers to assign the adjusted power required by the wind farm. The cost function of the DMPC was modified to meet the requirements of low fluctuations of the controlled process. The deviations between the process model and the real process, which are caused by the disturbances and model uncertainty were included. In this case, the accuracy of the KF estimation requires precise modelling of the wind turbines and the wind acting as a disturbance. The proposed controller provides good performance. However, due to oscillations of the mechanical dynamics of the power generator unit, the robustness of the control systems was not optimal.

7.2 Future work

1. It has been demonstrated that the critical percentage of dropouts to ensure the closed-loop stability can be found for linear NCS. However, an analysis of stability criteria of the control systems for the basic model of the network communication

could increase the impact of the reported methods. Moreover, stability results can be employed to adjust the controller and tuning rules can be found.

2. It is necessary to explore further DWNCS applications and investigate the effects of networks in a real-time operation. Energy is an important bedrock in the application of wireless technology in process control. Thus, power constraint in DMPC for WNCS needs further study. The problem in DMPC has been addressed using distributed H_∞ filter, investigation of inclusion of the dropouts and time-varying delays is worthy of further study.

3. DWNMPC showed that good performance could be achieved by using a coordination strategy and the appropriate network communication. However, the optimisation problem has been proposed to consider only the performance of the local controller. To pursue global performance, a neighbourhood optimisation where a cost function that takes into account the performance of the local and the agent neighbouring can be developed.

4. An extension of the DWNMPC design to consider PID control structures will find a good use in practice and special interest in the industrial area. Local controllers can be replaced with Predictive PID controllers operating in each subsystem and constraints handling can be provided as defined in Chapter 4.

5. Also, the DWNMPC in this work focuses on complex systems which work in parallel. However, the algorithm can be redefined to consider large-scale systems with serial structures. The method can be modified to produce the estimation algorithm and control system in a hierarchical sequence that starts from the top level to bottom level of the control structure.

6. The experiments in TrueTime simulator showed that all the control methods were successful in compensating time delays and dropouts. It is believed that the design can be extended to compensate other network constraints such as asynchronism and time-varying transmission/sampling intervals. These phenomena are also simulated by the TrueTime. In the decentralised control case, the KF estimation algorithm can be modified for multi-rate systems using an approach similar to the dropout compensation where multi-rate input and output signals are considered.

7. A case of study of torque control for low speeds in wind turbines has been investigated in this work. In future work, pitch control can be investigated to control

the power at higher speeds. Also, in a next step to design a full envelope controller, a switching control can be developed to give a smooth transition from one controller to another in accordance with the wind speed.

Bibliography

Åström, K. J. and T. Hägglund (2004). 'Revisiting the Ziegler-Nichols step response method for PID control'. In: *Journal of Process Control* 14 (6), pp. 635–650.

Abdullah, H., R. Ibrahim, S. M. Hassan and T. D. Chung (2016). 'Filtered feedback PID control for WirelessHART networked plant'. In: *2016 6th International Conference on Intelligent and Advanced Syst. (ICIAS)*. Malaysia, pp. 1–5.

Aho, J., A. Buckspan, J. Laks, P. Fleming, Y. Jeong et al. (2012). 'A tutorial of wind turbine control for supporting grid frequency through active power control'. In: *2012 American Control Conference (ACC)*, pp. 3120–3131.

Alessio, A., D. Barcelli and A. Bemporad (2011). 'Decentralized model predictive control of dynamically coupled linear systems'. In: *Journal of Process Control* 21 (5). Special Issue on Hierarchical and Distributed Model Predictive Control, pp. 705 – 714.

Alfaro, V. M., R. Vilanova and O. Arrieta (2010). 'Maximum Sensitivity Based Robust Tuning for Two-Degree-of-Freedom Proportional-Integral Controllers'. In: *Industrial and Engineering Chemistry Research* 49 (11), pp. 5415–5423.

Arcara, P. and C. Melchiorri (2002). 'Control schemes for teleoperation with time delay: A comparative study'. In: *Robotics and Autonomous Systems* 38 (1), pp. 49–64.

Bai, J., M. Fu and H. Su (2011). 'Delay modeling and estimation of a wireless based network control system'. In: *2011 8th Asian Control Conference (ASCC)*, pp. 187–192.

Bemporad, A. (1998). 'Predictive Control of Teleoperated Constrained Systef with Unbounded Communication Delays'. In: *Proceedings of the 37th IEEE Conference on Decision and Control Tampa, Florida USA December 1998* 55 (8), 2133–2138.

Bemporad, A. and D. Barcelli (2010). 'Decentralized Model Predictive Control'. In: *Networked Control Systems*. Ed. by A. Bemporad, M. Heemels and M. Johansson. London: Springer London, pp. 149–178.

Bianchi, F. D., H. De Battista and R. J. Mantz (2007). *Wind Turbine Control Systems Principles, Modelling and Gain Scheduling Design*. London, U.K.: Springer-Verlag London Limited, pp. 95–97.

Bjorkbom, M. (2010). 'Wireless Control System Simulation and Network Adaptive Control'. PhD thesis. Aalto University, School of Science and Technology.

Blevins, T., M. Nixon and W. Wojsznis (2014). 'PID control using wireless measurements'. In: *2014 American Control Conference*, pp. 790–795.

Blevins, T., M. Nixon and M. Zielinski (2013). 'Using Wireless measurements in control applications'. In: *International Society of Automation (ISA) Automation Week conference, Nasville, USA*.

Borgers, D. P. and M. Heemels (2014). 'Stability Analysis of Large-scale Networked Control Systems with Local Networks: A Hybrid Small-gain Approach'. In: *Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control*. HSCC '14. Berlin, Germany: ACM, pp. 103–112.

Camacho, E. and A. Bordons (2007). *Model Predictive Control*. Springer-Verlag, London, U.K.

Cervin, A., D. Henriksson, B. Lincoln, J. E. and K. Erickzon (2003). 'How Does Control Timing Affect Performance: Analysis and Simulation of Timing Using Jitterbug and TrueTime'. In: *IEEE Control Systems Magazine* 23 (3), pp. 16–30.

Chen, H., Y. Qu, X. Zhou and B. Liu (2014). 'The Improved Fuzzy-Smith Compensator based on Modeling Error for Network Control System'. In: *The 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems* 82 (7), pp. 106–110.

Chen, L. and R. Chen (2009). 'A Fuzzy Immune PID Controller for Electronic Throttle'. In: *Computational Intelligence and Design, 2009. ISCID '09. Second International Symposium on*. Vol. 1, pp. 72–75.

Christofides, P. D., J. Liu and D. Muñoz de la Peña (2011). *Networked and Distributed Predictive Control*. Springer-Verlag London.

Clarke, D.W., C. Mohtadi and P.S. Tuffs (1987). 'Generalized predictive control—Part I. The basic algorithm'. In: *Automatica* 23 (2), pp. 137 –148.

Council, Global Wind Energy (2016). *Global Wind Report on Annual Market Update*. Tech. rep. Belgium.

Dasgupta, S., K. Halder, S. Banerjee and A. Gupta (2015). 'Stability of Networked Control System (NCS) with discrete time-driven PID controllers'. In: *Control Engineering Practice* 42 (2015), pp. 41 –49.

Decotignie, J. D. (2005). 'Ethernet-based real-time and industrial communications'. In: *Proceedings of the IEEE* 93 (6), pp. 1102–1117.

DJ, Leith and Leithead WE (1996). 'Appropriate realization of gain-scheduled controllers with application to wind turbine regulation'. In: *International Journal of Control* 65 (2), pp. 223–248.

Du, F., W. Du and Z. Lei (2009a). 'A Novel Smith Predictor for Wireless Networked Control Systems'. In: *Control, Automation and Systems Engineering, 2009. CASE 2009. IITA International Conference on*, pp. 667–670.

– (2009b). 'New Smith Predictor and Nonlinear Control for Networked Control Systems'. In: *Engineers and Computer Scientists, 2009. IMECS 2009. International MultiConference on*.

Du, S.L., X. M. Sun, W. Wang and C. Q. Zhong (2014). 'A new stability criterion for networked predictive control system with varying controller gains scheme'. In: *Control and Decision Conference (2014 CCDC), The 26th Chinese* 95 (1), pp. 893 –897.

Du, W. and F. Du (2009). 'New Smith Predictor and CMAC- PID Control for Wireless Networked Control Systems'. In: *10th ACIS*. College of Information Sciences and Technology, Hainan University Haikou, Hainan, China: ACIS, pp. 1184–1188.

Eriksson, L. and H. N. Koivo (2005). 'Tuning of Discrete-Time PID Controllers in Sensor Network based Control Systems'. In: *in Proc. 2005 IEEE Int. Symp. on Computational Intelligence in Robotics and Automation.* , Finland: IEEE, p. 6.

Eriksson, L. and T. Oksanen (2007). 'PID Controller Tuning for Integrating Processes: Analysis and New Design Approach'. In: *Fourth International Symposium on Mechatronics and its Applications (ISMA07)*. LA, USA, pp. 26–29.

Eriksson, L.M and M. Johansson (2007a). 'PID Controller Tuning Rules for Varying Time-Delay Systems'. In: *American Control Conference, 2007. ACC '07*, pp. 619–625.

Eriksson, L.M. and M. Johansson (2007b). 'Simple PID Tuning Rules for Varying Time-Delay Systems'. In: *The 46th IEEE Conference on Decision and Control CDC 2007*. LA, USA: International Institute of Electrical and Electronics Engineers (IEEE), pp. 12–14.

Feng, S. and F. Du (2010). 'Research of networked control systems based on Neuro-Fuzzy control'. In: *Proceedings - 2010 3rd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2010* 6, pp. 297–300.

Franzè, G., F. Tedesco and D. Famularo (2015). 'Model predictive control for constrained networked systems subject to data losses'. In: *Automatica* 54 (2015), pp. 272 –278.

Freirich, D. and E. Fridman (2016). 'Decentralized networked control of systems with local networks: A time-delay approach'. In: *Automatica* 69 (2016), pp. 201 –209.

Friman, M. and J. Nikunen (2013). 'A Practical and Functional Approach to Wireless PID Control'. In: *2013 21st Mediterranean Conf. on Control and Automation (MED) Platanias-Chania, Greece, June*, pp. 25–28.

Gao, H. and T. Chen (2008). 'Network-based H∞ output tracking control'. In: *IEEE Trans. Autom. Control* 53 (3), 655–667.

Ge, X., F. Yang and Q. L. Han (2017). 'Distributed networked control systems: A brief overview'. In: *Information Sciences* 380 (Supplement C), pp. 117 –131.

Guo, C., P. Chen, J. Zhang and D. Peng (2014). 'A survey on networked control systems subject to limited network resources'. In: *Control and Decision Conference (2014 CCDC), The 26th Chinese* 82 (7), pp. 4958–4965.

Gupta, V. and N. C. Martins (2010). 'On Stability in the Presence of Analog Erasure Channel Between the Controller and the Actuator'. In: *IEEE Transactions on Automatic Control* 55 (1), pp. 175–179.

Han, Q. L., C. Peng and M. R. Fei (2016). 'Special issue on Recent Developments in Distributed Networked Control Systems'. In: *Information Sciences* 370–371, pp. 551 –553.

Hansen, A. D., P. Sørensen, F. Iov and F. Blaabjerg (2006). 'Centralised power control of wind farm with doubly fed induction generators'. In: *Renewable Energy* 31 (7), pp. 935 –951.

Hassan, S., R. Ibrahim, K. Bingi, T. D. Chung and N. Saad (2017). 'Application of Wireless Technology for Control: A WirelessHART Perspective'. In: *Procedia Computer Science* 105, pp. 240 –247.

Hassan, S. M., R. Ibrahim, N. Saad, V. S. Asirvadam and T. D. Chung (2016). 'Predictive PI controller for wireless control system with variable network delay and disturbance'. In: *2016 2nd IEEE Int. Symp. on Robotics and Manufacturing Automation (ROMA)*. Malaysia, pp. 1–6.

Heemels, M., D. P. Borgers, N. van de Wouw, D. Nešić and A. R. Teel (2013). 'Stability analysis of nonlinear networked control systems with asynchronous communication: A small-gain approach'. In: *52nd IEEE Conference on Decision and Control*, pp. 4631–4637.

Heemels, W. P. H., A. R. Teel, N. van de Wouw and D. Nesic (2010). 'Networked control systems with communication constraints: Tradeoffs between transmission intervals, delays and performance'. In: *IEEE Transactions on Automatic Control* 55 (8), 1781–1796.

Hu, L.S., T. Bai, P Shi and Z. Wu (2007). 'Sampled-data control of networked linear control systems'. In: *Automatica* (2007), 903–911.

Huang, Y. J. and Y. J. Wang (2001). 'Robust PID controller design for non-minimum phase time delay systems'. In: *ISA Transactions* 40 (2001), pp. 31–39.

Hur, S. and W.E. Leithead (2017). 'Model predictive and linear quadratic Gaussian control of a wind turbine'. In: *Optimal Control Applications and Methods* 38 (1), pp. 88–111.

Hur, S.H. and W. E. Leithead (2016). 'Adjustment of wind farm power output through flexible turbine operation using wind farm control'. In: *Wind Energy* 19 (9), pp. 1667–1686.

Izadi, H. A., B. W. Gordon and Y. Zhang (2011). 'Decentralized Model Predictive Control for Cooperative Multiple Vehicles Subject to Communication Loss'. In: *International Journal of Aerospace Engineering* (2011), pp. 1–13.

Johansson, K., M. Törngren, L. Nielsen, I. D. Hristu-Varsakelis and W. S. Levine (2005). *Vehicle applications of controller area network, in Handbook of Networked and Embedded Control Syst.* Boston: Birkhaüse.

Johnson, M. A. and M. H. Moradi (2005). *PID Control: New Identification and Design Methods.* London, U.K.: Springer-Verlag London Limited, pp. 500–515.

Kaldellis, J. K. and D. Zafirakis (2011). 'The wind energy (r)evolution: A short review of a long history'. In: *Renewable Energy* 36 (7), pp. 1887 –1901.

Katebi, M. R. (2001). 'Predictive PID controllers'. In: *International Journal of Systems Applications, Engineering and Development* 148 (6), pp. 478–487.

Katebi, M.R. and M.A. Johnson (1997). 'Predictive control design for large-scale systems'. In: *Automatica* 33 (3), pp. 421 –425.

Leithead, W. E. and B. Connor (2000). 'Control of variable speed wind turbines: Design task'. In: *International Journal of Control* 73 (13), pp. 1189–1212.

Li, F., P. Shi, X. Wang and R. Agarwal (2015). 'Fault detection for networked control systems with quantization and Markovian packet dropouts'. In: *Signal Processing* 111 (Supplement C), pp. 106 –112.

Li, H., H. Yang, F. Sun and Yu. Xia (2014). 'Sliding-Mode Predictive Control of Networked Control Systems Under a Multiple-Packet Transmission Policy'. In: *IEEE Transactions on Industrial Electronics* 61 (11), pp. 6234–6243.

Liu, G. P., Y. Xia, D. Rees and W. Hu (2007). 'Design and Stability Criteria of Networked Predictive Control Systems With Random Network Delay in the Feedback Channel'. In: *IEEE Transactions on Systems, MAN, and Cybernetics* 37, pp. 173–184.

Liu, J. and V. Gupta (2012). 'Decentralized control over analog erasure links'. In: *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pp. 6926–6931.

– (2017). 'On Stabilization of Decentralized Systems Across Analog Erasure Links'. In: *IEEE Transactions on Automatic Control* 62 (3), pp. 1411–1416.

Liu, K. Z., R. Wang and G. P. Liu (2016). 'Tradeoffs Between Transmission Intervals and Delays for Decentralized Networked Control Systems Based on a Gain Assignment Approach'. In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 63 (5), pp. 498–502.

Liu, X., Y. Zhang and K. Y. Lee (2017). 'Coordinated Distributed MPC for Load Frequency Control of Power System With Wind Farms'. In: *IEEE Transactions on Industrial Electronics* 64 (6), pp. 5140–5150.

Liu, Y., Q. Chen and H. Zhu (2014). 'Robust model predictive control for a class of networked control systems with timevarying delays'. In: *2014 IEEE Workshop on Electronics, Computer and Applications*, pp. 70–73.

Ljung, L (1999). *System Identification: Theory for the User*. Upper Saddle River, NJ: Prentice-Hall PTR, pp. 363–369.

Lu, R., Y. Xu and R. Zhang (2016). 'A New Design of Model Predictive Tracking Control for Networked Control System Under Random Packet Loss and Uncertainties'. In: *IEEE Transactions on Industrial Electronics* 63 (11), pp. 6999–7007.

Lu, Z., F. Xu and Q. Tian (2012). 'Research on quantization errors of stability for model-based networked control system'. In: *2012 Proceedings of International Conference on Modelling, Identification and Control*, pp. 867–872.

Madjidian, D., K. Mårtensson and A. Rantzer (2011). 'A distributed power coordination scheme for fatigue load reduction in wind farms'. In: *Proceedings of the 2011 American Control Conference*, pp. 5219–5224.

Madsen, J. T., M. Findrik, D. Drenjanac and H. P. Schwefel (2015). 'Investigating Wind Farm Control Over Different Communication Network Technologies'. In: *Energy Informatics: 4th D-A-CH Conference, EI 2015, Karlsruhe, Germany, November 12-13,*

2015, Proceedings. Ed. by Sebastian Gottwalt, Lukas König and Hartmut Schmeck. Cham: Springer International Publishing, pp. 129–140.

Mazumder, S. K. (2011). *Wireless Networking Based Control*. New York: Springer-Verlag New York.

Miklovičová, E. and M. Mrosko (2012). 'PID control strategy for sensor random packet dropouts in networked control system'. In: *Int. Journal of Systems Applications, Engineering and Development* 6 (1), pp. 154–162.

Montestruque, L. A. and P. J. Antsaklis (2003). 'On the model-based control of networked systems'. In: *Automatica* 39 (4), 1837–1843.

Moyne, James R. and D.M. Tilbury (2007). 'The Emergence of Industrial Control Networks for Manufacturing Control, Diagnostics, and Safety Data'. In: *Proceedings of the IEEE* 95 (1), pp. 29–47.

Murray, R. M., K. J. Åström, S.P. Boyd, R. W. Brockett and G. Stein (2003). 'Future directions in control in an information-rich world'. In: *IEEE Control Systems* 23 (2), pp. 20–33.

Njiri, J. G. and D. Söffker (2016). 'State-of-the-art in wind turbine control: Trends and challenges'. In: *Renewable and Sustainable Energy Reviews* 60.2016, pp. 377 –393.

O.J.M, Smith (1958). *Feedback Control Systems*. New York: McGraw-Hill.

Okano, R., T. Ohtani and A. Nagashima (2008). 'Networked control systems by PID controller Improvement of performance degradation caused by packet loss'. In: *2008 6th IEEE Int.Conf. on Industrial Informatics*, pp. 1126–1132.

Onat, A., T. Naskali, E. Parlakay and O. Mutluer (2011). 'Control Over Imperfect Networks: Model-Based Predictive Networked Control Systems'. In: *IEEE Transactions on Industrial Electronics* 58 (3), pp. 905–913.

Pang, Z. H. et al. (2017). 'Input Design-Based Compensation Control for Networked Nonlinear Systems With Random Delays and Packet Dropouts'. In: *Circuits Syst. II, Exp. Briefs* 64, pp. 299–303.

Pansari, D., B. Timande and D. Chandrakar (2012). 'A design of robust PID controller for non-minimum Network Control System'. In: *International Journal of Advances in Engineering and Technology* 5 (1), pp. 486–493.

Park, P., P. Di Marco, P. Soldati, C. Fischione and K. H. Johansson (2009). 'A Generalized Markov Chain Model for Effective Analysis of Slotted IEEE 802.15.4'. In: *2009 IEEE 6th International Conference on Mobile Adhoc and Sensor Systems (Mass 2009)* 56 (1), pp. 285–294.

Peng, Daogang, Hao Zhang, Conghua Huang, Jiajun Lin and Hui Li (2011). 'Study of Immune PID Networked Control System Based on TrueTime'. In: *Journal of Networks* 6 (6), pp. 912–915. ISSN: 1796-2056.

Pin, G. and T. Parisini (2011). 'Networked predictive control of uncertain constrained nonlinear systems recursive feasibility and input-to-state stability analysis'. In: *IEEE Trans. Autom. Control* 56 (1), pp. 72–87.

Pohjola, M., L. Eriksson and H. Koivo (2006). 'Tuning of PID Controllers for Networked Control Systems'. In: *in Proc. The 32nd Annual Conf. of the IEEE Industrial Electronics Society (IECON'060)*. Paris, France: IEEE, p. 6.

Poushpas, S. (2016). 'Wind Farm Simulation Modelling and Control'. PhD thesis. University of Strathclyde.

Qin, S.J. and T. A. Badgwell (2003). 'A survey of industrial model predictive control technology'. In: *Control Engineering Practice* 11 (7), pp. 733 –764.

Qiu, L., Y. Shi, F. Yao, G. Xu and B. Xu (2015). 'Network-Based Robust H_2/H_∞ Control for Linear Systems With Two-Channel Random Packet Dropouts and Time Delays'. In: *IEEE Transactions on Cybernetics* 45.8, pp. 1450–1462.

Rasool, F., D. Huang and S .K. Nguang (2011). 'Robust H_∞ output feedback control of discrete-time networked systems with adaptive quantizers'. In: *IEEE Conference on Decision and Control and European Control Conference*, pp. 2381–2386.

Roshany-Yamchi, S., M. Cychowski, R. R. Negenborn, B. De Schutter, K. Delaney and J. Connell (2013). 'Kalman Filter-Based Distributed Predictive Control of Large-Scale Multi-Rate Systems: Application to Power Networks'. In: *IEEE Transactions on Control Systems Technology* 21 (1), pp. 27–39.

Rossiter, J.A. (2003). *Model-Based Predictive Control: A Practical Approach*. CRC Press, pp. 31–33.

Sawada, K. and S. Shin (2012). 'On numerical optimization design of continuous-time feedback type quantizer for networked control systems'. In: *IEEE International Conference on Automation Science and Engineering* 3, pp. 1144–1149.

Singh, V. P., P. Samuel and N. Kishor (2015). 'Effect Of communication delay on load frequency control application in autonomous hybrid power system'. In: *2015 IEEE Innovative Smart Grid Technologies - Asia (ISGT ASIA)*, pp. 1–6.

Sinopoli, B., L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan and S. S. Sastry (2004). 'Kalman filtering with intermittent observations'. In: *IEEE Transactions on Automatic Control* 49 (9), pp. 1453–1464.

Soliman, M., O. P. Malik and D. T. Westwick (2011). 'Multiple Model Predictive Control for Wind Turbines With Doubly Fed Induction Generators'. In: *IEEE Transactions on Sustainable Energy* 2 (3), pp. 215–225.

Spudic, V., M. Jelavi, M. Baoti and N. Peri (2010). 'Hierarchical Wind Farm Control for Power-Load Optimization'. In: *The science of making torque from wind (Torque2010)*.

Stock, A. (2015). 'Augmented Control for Flexible Operation of Wind Turbines'. PhD thesis. University of Strathclyde.

Sun, J., J. Chen and L. Dou (2014). 'Networked predictive control for linear systems with unknown communication delay'. In: *2014 UKACC International Conference on Control 9th - 11th July 2014, Loughborough, U.K* 61 (6), pp. 668 –672.

Sun, X. M., D. Wu, C. Wen and W. Wang (2014a). 'A Novel Stability Analysis for Networked Predictive Control Systems'. In: *IEEE Trans. Circuits and Systems II: Express Briefs* 61 (6), pp. 453–457.

Sun, X. M., D. Wu, G. P. Liu and W. Wang (2014b). 'Input-to-State Stability for Networked Predictive Control With Random Delays in Both Feedback and Forward Channels'. In: *IEEE Transactions on Industrial Electronics* 61 (7), pp. 3519–3526.

Tan, K. K., T. H. Lee, S. N. Huang and F. M. Leu (2002). 'PID Control Design Based on a GPC Approach'. In: *Industrial and Engineering Chemistry Research* 41 (8), pp. 2013–2022.

Tang, P. and C. de Silva (2006). 'Compensation for transmission delays in an Ethernet-based control network using variable-horizon predictive control'. In: *IEEE Trans. Control Syst. Technol* 14 (4), pp. 707–718.

Tiberi, O., J. Araujo and K. H. Johansson (2012). 'On event-based PI control of first-order processes'. In: *IFAC Conference on Advances in PID Control PID'12 Brescia (Italy)*. Stockholm, Sweden: ACCESS Linnaeus Center, KTH Royal Institute of Technology, pp. 28–30.

Tipsuwan, Y. and Mo-Yuen Chow (2004). 'Gain scheduler middleware: a methodology to enable existing controllers for networked control and teleoperation-part II: teleoperation'. In: *Industrial Electronics, IEEE Transactions on* 51 (6), pp. 1228–1237.

Tran, H.D., Z.H. Guan, X.K. Dang, X.M Cheng and F.S. Yuan (2013). 'A normalized PID controller in networked control systems with varying time delays'. In: *ISA Transactions* 52 (5), pp. 592 –599.

Tuan, H.D., A. Savkin, T.N. Nguyen and H.T. Nguyen (2015). 'Decentralised model predictive control with stability constraints and its application in process control'. In: *Journal of Process Control* 26 (2015), pp. 73 –89.

Ulusoy, A., O. Gurbuz and A. Onat (2011). 'Wireless Model-Based Predictive Networked Control System Over Cooperative Wireless Network'. In: *IEEE Transactions on Industrial Informatics* 7 (1), pp. 41–51.

Ungan, V. (2010). 'Networked PID Controllers for Wireless Systems'. Master's Degree Project. Stockholm, Sweden: KTH ELectrical Engineering.

Vaccarini, M., S. Longhi and M. R. Katebi (2009). 'Unconstrained networked decentralized model predictive control'. In: *Journal of Process Control* 19 (2), pp. 328 – 339.

Wang, F. Y. and D. Liu (2008). *Networked control systems: theory and applications*. London: Springer.

Wang, L. (2009). *Model Predictive Control System Design and Implementation using MATLAB*. 1st. Springer Publishing Company, Incorporated, pp. 1–7.

Wang, Y. L., W. T. Liu, X. L. Zhu and Z. Du (2011). 'A survey of networked control systems with delay and packet dropout'. In: *2011 Chinese Control and Decision Conference (CCDC)*, pp. 2342–2346.

Wenlong, Z., J. Bae and M. Tomizuka (2015). 'Modified Preview Control for a Wireless Tracking Control System with Packet Loss'. In: 20 (1), pp. 299–307.

Wong, W. S. and R. W. Brockett (1999). 'Systems with finite comunication bandwidth constraints. II: Stabilization with limited information feedback'. In: *IEEE Trans. Autom. Control* 44 (5), 1049–1053.

Wu, Y., Y. Wu and Y. Zhao (2016). 'An Enhanced Predictive Control Structure for Networked Control System with Random Time Delays and Packet Dropouts'. In: *2016 3rd Int. Conf. on Inform. Science and Control Eng. (ICISCE)*. China, pp. 834–838.

Xia, F., X. Dai, Z. Wang and Y. Sun (2005). 'Feedback based network scheduling of networked control systems'. In: *Control and Automation, 2005. ICCA '05. International Conference on*. Vol. 2, 1231–1236 Vol. 2.

Xue, Yang and Z. J. Yan (2010). 'Research on RBF tuning PID and fuzzy immune control system of superheat temperature'. In: *Natural Computation (ICNC), 2010 Sixth International Conference on*. Vol. 7, pp. 3528–3531.

Yan, Y., Y. Zhang and X. Liu (2015). 'Distributed MPC strategy with application to AGC in the presence of variable speed wind turbine'. In: *2015 34th Chinese Control Conference (CCC)*, pp. 4151–4155.

Yang, P., H. Zhao, Y. Zhou and Z. Liu (2007). 'Study of Immune PID-PI Controller for FG-3000 Temperature Control System'. In: *2007 IEEE International Conference on Integration Technology*, pp. 343–347.

Yang, Z., Y. Li and J. E Seem (2015). 'Multi-model predictive control for wind turbine operation under meandering wake of upstream turbines'. In: *Control Engineering Practice* 45 (2015), pp. 37–45.

Yue, D., Q. Han and C. Peng (2004). 'State feedback controller design of networked control systems'. In: *IEEE Trans. Circuits Syst. II, Exp. Briefs* 51 (11), 640–644.

Zhang, D., P. Shi, Q. G. Wang and L.Yu (2017). 'Analysis and synthesis of networked control systems: A survey of recent advances and challenges'. In: *ISA Transactions* 66 (2017), pp. 376 –392.

Zhang, H., Y. Shi and A. S. Mehr (2011). 'Robust Static Output Feedback Control and Remote PID Design for Networked Motor Systems'. In: *IEEE Trans. on Industrial Electronics* 58 (12), pp. 5396–5405.

Zhang, L., H. Gao and O. Kaynak (2013). 'Network-Induced Constraints in Networked Control Systems-A Survey'. In: *IEEE Transactions on Industrial Informatics* 9 (1), pp. 403–416.

– (2014). 'An adaptive tuning algorithm for IEEE 802.15.4-based network control system'. In: *Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference* 9 (1), pp. 1–6.

Zhang, Y., C. Bao and C. Xu (2013). 'Neighbourhood Optimization Enhanced Networked MPC with One-step Delay Communication for Large-scale Systems'. In: *IFAC Proceedings Volumes* 46 (13), pp. 166 –171.

Appendix A

TrueTime simulator

A.1 Description

Truetime is a networked control system simulation toolbox based on MATLAB/Simulink which allows to test the control system real-time design and provide a detailed analysis of the effects of the communication network. TrueTime (version 2.0-beta-10.2) has been developed by Cervin et al. (2003).

At the core of TrueTime is a kernel that enables controller tasks execution. Simulink blocks are used to model the processes controlled by those tasks, including simple models of communication networks, sensors, controllers and actuators. These blocks are run by MATLAB S-functions written in C++. The execution of the task that these blocks have to perform is defined by code functions that the user can write in C++ or Matlab m-files. In particular, these tasks are periodic or not periodic activities such as reading and writing controller tasks, communication and the triggering of events for the controller functionality.

In the core, the task has an execution instruction that allows to simulate input/output delays. These can be constant, random or time-dependent.

Four kinds of task scheduling algorithms are provided, such as fixed priority (prioFP), rate monotonic (prioRM), deadline monotonic (prioDM) and earliest deadline first (prioEDF).

The TrueTime library consists of six blocks: TrueTime Kernel, TrueTime Network, TrueTime Send, TrueTime Receive, TrueTime Battery, TrueTime Wireless Network and TrueTime Ultrasound Network. A real-time networked control system can be built by connecting these blocks with common blocks in Simulink.

The network block is event driven and executes when messages enter or leave the network. The messages contain the measurement or control signals, the length of the message and optional real-time attributes such as a priority or a deadline which can be defined by the user.

In the network block, it is possible to specify the network parameters such as transmission rate, the medium access control protocol, the number of network nodes, the probability of loss, among others.

A suitable number of medium access protocols can be supported by this simulator, i.e. CSMA/CD (e.g. Ethernet), CSMA/ AMP (e.g. CAN), Round Robin (e.g. Token Bus), FDMA, TDMA (e.g. TTP), and Switched Ethernet (Cervin et al., 2003). Two network protocols are supported for Wireless: IEEE 802.11b/g (WLAN) and IEEE 802.15.4 (ZigBee).

A.2 Example

To describe the implementation of a NCS, a simulation using the Truetime DC servo motor process is described here. An Ethernet network protocol has been configured for this example. The transfer function of the process is:

$$H(s) = \frac{1000}{s(s+1)} \tag{A.1}$$

The block diagram that illustrates the implementation of the NCS is depicted in Figure A.1. The sensor/actuator node are represented in a single block. It will send and receive the process output and the controller input. An interfering node sending disturbing traffic over the network is implemented. The percentage of occupation of the network bandwidth can be selected to increase the time delays.

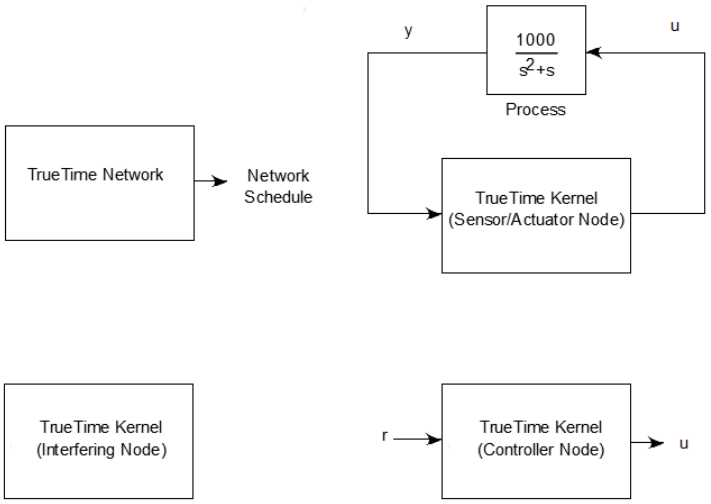

FIGURE A.1: The TrueTime Network nodes

The controller node is configured as follows. A simple discrete PD controller is implemented in a Matlab S-function by using the equations:

$$u(k) = P(k) + D(k) \tag{A.2}$$

where $P(k) = K[r(k) - y(k)]$
$D(k) = aD(k-1) + b[(y(k-1) - y(k)]$
The proportional gain is K, the reference is $r(k)$, the output is $y(k)$, a and b are tuning parameters. The TrueTime kernel calls to the S-function controller_code.m which is the code function where the user can change the execution time to simulate the effect of a delay. The code presented below shows the instruction for an execution time of 0.5 ms.

```
function [exectime, data] = controller_code(seg, data)

switch seg
case 1
y = ttGetMsg;          % Obtain sensor value
if isempty(y)
disp('Error in controller: no message received!');
y = 0.0;
end
r = ttAnalogIn(1);     % Read reference value
P = data.K*(r-y);
```

```
D = data.a*data.Dold + data.b*(data.yold-y);
data.u = P + D;
data.Dold = D;
data.yold = y;
exectime = 0.0005;

case 2
ttSendMsg(2, data.u, 80);     % Send 80 bits to node 2 (actuator)
exectime = -1; % finished
end
```

LISTING A.1: Truetime controller code function

For this example, the TrueTime Network node is configured for CSMA/CD (Ethernet), with a data rate of 80000 bits/s. The minimum frame size has been selected as 80 bits. The time delays can be set through the executive time of the sensor, actuator and controller tasks. Also by adding the interfering node or by reducing the frame size or data rate. The dropouts have been set up zero.

The results are illustrated in Figure A.2. The simulation shows that the controller behaved as expected. Some simulations with higher times, which are not presented here, exhibit that the system became unstable as the delay was increased.

The schedule plot in Figure A.3 demonstrates the states of the network three nodes. The states are represented by the following level: high=sending, medium=waiting, low=idle. The schedule plot can be modified in the *ttnetwork.cpp* function to display the time instant of the dropouts. In case of using the wireless network the function is *ttwnetwork.cpp*.

The nodes must wait several milliseconds until they gain access to the network, leading to various delays that affect the performance of the NCS. For instance, during the first sampling time, the controller node is the last one to send its message and it waits approximately 5 ms.

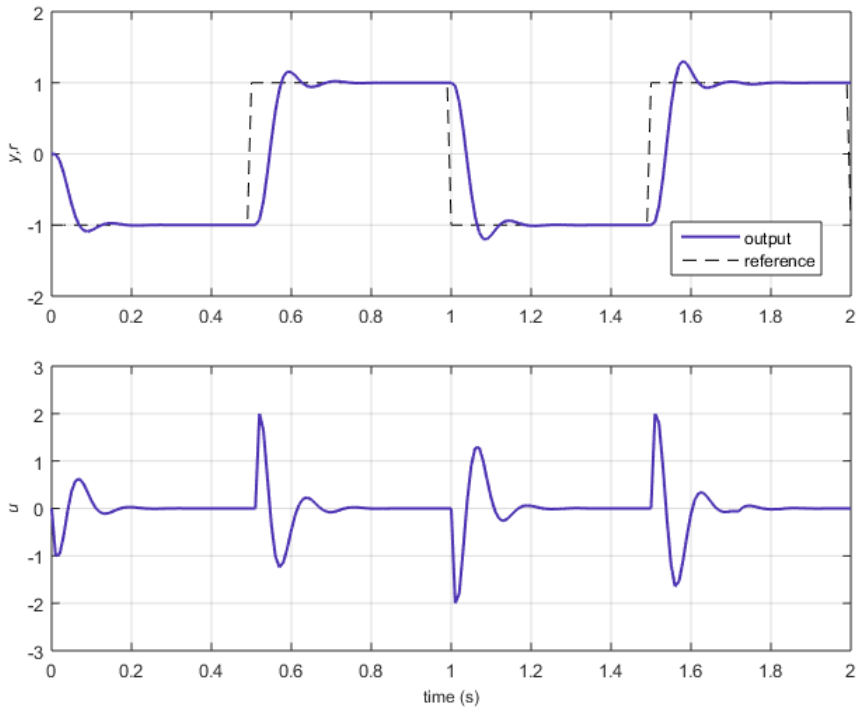

FIGURE A.2: Systems outputs

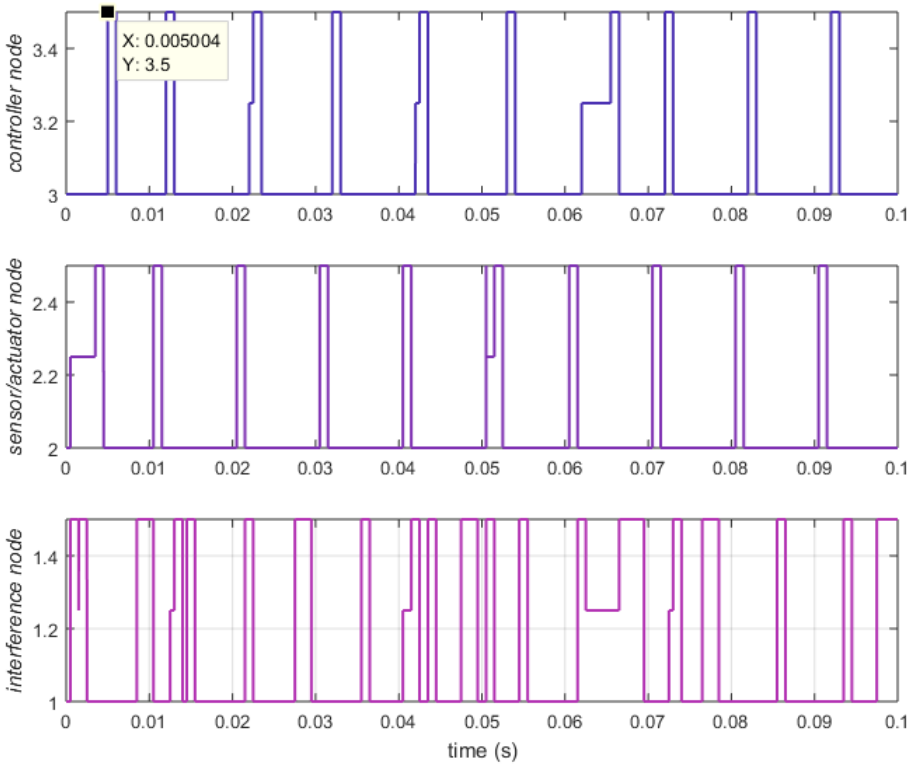

FIGURE A.3: The network schedule

Appendix B

Recursive least squares algorithm

A second-order model is given by the equation (Ljung, 1999, pp. 363-369):

$$y(k) = a_0\, u(k) + a_1\, u(k-1) - b_1\, y(k-1) - b_2\, y(k-2) \tag{B.1}$$

where a_0, a_1, b_0, b_1 are the model coefficients, $y(k)$ is the present value, $y(k-n)$ and $u(k-n)$ are the output and input values at the $k-n$ sampling instant. These represent the regression values, denote by ϕ. The unknown parameters are found by computing the vector of parameters θ.

The implementation minimises the sum of the squares of the differences between observed and calculated values and it is weighted by multiplying it by a constant. This can be defined as:

$$\theta = min \sum_{k=1}^{t} \beta(t,k)\, [y(k) - \phi^T(k)\, \theta]^2 \tag{B.2}$$

where

$$\beta(t,k) = \prod_{i=k+1}^{t} \lambda(i) \tag{B.3}$$

where λ is the forgetting factor, a value between zero and one.

The recursive algorithm is implemented using the following new parameters estimation equation:

$$\theta = \theta(t-1)\ + \frac{P(t-1)\, \phi(t)}{\lambda + \phi^T(t)\, P(t-1)\, \phi(t)} [e(t)] \tag{B.4}$$

where $e(t)$ is the estimation error given by:

$$e(t) = y(t) - \phi^T(t)\,\theta(t-1) \tag{B.5}$$

And the covariance matrix $P(t)$ is computed as:

$$P(t) = \frac{1}{\lambda}\left[P(t-1)\ - \frac{P(t-1)\,\phi(t)\phi^T(t)\,P(t-1)}{\lambda + \phi^T(t)\,P(t-1)\,\phi(t)}\right] \tag{B.6}$$

Appendix C

State-space matrices for DWNMPC method

The effectiveness of the DWNMPC solution has been tested using the MIMO process illustrated in (5.33). The model has been transformed into a state-space representation using a sampling time $Ts = 0.7\,s$. The final matrices are given as follows:

$$
A = \begin{bmatrix}
1.5 & -0.7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.37 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.37 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -0.05 & 0.3 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1.57 & -0.67 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0.12 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4
\end{bmatrix}
$$

$$
B = \begin{bmatrix}
2 & 0 & 0 \\
0 & 0 & 0 \\
0.03 & 0 & 0 \\
0.03 & 0 & 0 \\
0 & 0.03 & 0 \\
0 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 0 \\
0 & 0.03 & 0 \\
0 & 0 & 0.02 \\
0 & 0 & 0.03 \\
0 & 0 & 1
\end{bmatrix}
\tag{C.1}
$$

$$
C = \begin{bmatrix}
1 & -0.49 & 0 & 0 & 0.006 & 0.02 & 0 & 0 & 0 & 0.03 & 0 & 0 \\
0 & 0 & 0.04 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.04 & 0 \\
0 & 0 & 0 & 0.04 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.99
\end{bmatrix}
$$

Appendix D

State-space matrices for wind turbine

The matrices for the state-space representation of the wind turbine model studied in Chapter 6 are:

$$
A = \begin{bmatrix}
0.6228 & -0.4237 & -0.0004 & 0.0000 & -0.0000 & 0.0003 & 0.0351 & -0.4525 & -0.0215 & -0.0006 & 0 \\
-0.1668 & -0.3394 & -0.6438 & -0.0000 & 0.0000 & -0.0001 & -0.0143 & 0.1839 & 0.0087 & 0.0003 & 0 \\
-0.1705 & -0.5500 & 0.6940 & -0.0150 & 0.0000 & -0.0002 & -0.0245 & 0.3162 & 0.0150 & 0.0005 & 0 \\
-0.0038 & -0.0048 & 0.0017 & 0.9041 & 0.0623 & -0.0000 & -0.0005 & 0.0061 & 0.0003 & 0 & 0 \\
0.0012 & 0.0004 & 0.0031 & -0.0887 & 1.0208 & 0.0899 & 0.0001 & -0.0008 & -0.0000 & 0 & 0 \\
0.0001 & 0.0007 & -0.0026 & 0.0704 & -0.0455 & 0.9748 & -0.0074 & 0.0002 & 0.0000 & 0 & 0 \\
0.0568 & 0.0006 & -0.0029 & 0.1020 & 0.0358 & 0.1687 & 0.9986 & 0.1045 & 0.0013 & 0 & 0 \\
-0.7535 & -0.0002 & 0.0003 & 0.0050 & 0.0012 & 0.0071 & 0.1027 & -0.3244 & -0.0630 & -0.0005 & 0 \\
-0.0291 & 0.0141 & 0.0037 & 0.0612 & 0.0304 & 0.1406 & 0.0164 & -0.0656 & 0.9936 & -0.0302 & 0 \\
-0.0007 & -0.0006 & 0.0002 & 0.0367 & 0.0000 & 0.0163 & -0.0286 & -0.0049 & 0.0142 & 0.9975 & 0.0194 \\
-0.0057 & -0.0116 & -0.0089 & 0.1073 & 0.0274 & 0.1520 & 0.0113 & 0.0039 & -0.0016 & -0.0384 & 1
\end{bmatrix}
$$

$$
B = 1 \times 10^6 \times \begin{bmatrix}
-0.1352 \\
0.2344 \\
-0.4634 \\
-0.0121 \\
0.0176 \\
-0.0006 \\
-0.0318 \\
0.0169 \\
0.1613 \\
0.0055 \\
-0.1256
\end{bmatrix}
$$

$$
C = \begin{bmatrix}
10.2047 & -28.2781 & 51.6790 & 1.3672 & -1.9456 & 0.0669 & 3.2881 & 1.1196 & -17.6512 & -0.6068 & 13.8506
\end{bmatrix}
$$

(D.1)