UNIVERSIDAD NACIONAL DEL LITORAL
FACULTAD DE INGENIERÍA QUÍMICA

Tesis presentada como parte de los requisitos de la Universidad Nacional del Litoral para la obtención del Grado Académico de

Doctor en Tecnología Química

En el campo de: **Control Óptimo de Procesos**

Título de la tesis:

Fault-tolerant Model-based Predictive Control Applied to Industrial Processes

Institución donde se realizó:
Departamento de Ingeniería de Procesos
Facultad de Ingeniería Química, UNL

Autor: Emanuel Bernardi

Director de Tesis:
Dr. Eduardo Adam

Miembros del jurado:
Dra. Marta Susana BASUALDO
Dr. Antonio FERRAMOSCA
Dr. Jorge Rubén VEGA

Año de presentación: 2021

"I have dreams, and I have nightmares. I overcame the nightmares because of my dreams."

Jonas E. Salk

Acknowledgements

First of all, I would like to express my sincere thanks to my supervisor, Prof. Eduardo J. Adam, for his support, encouragement and advisement during these years. Without his guidance and constant feedback, this achievement would not have been possible.

I would also like to thank my lab colleagues for the work done together, sharing their experience and knowledge with me.

Thanks also to Prof. Julio E. Normey-Rico, for receiving me in his research group, at UFSC, Florianópolis, in Brazil. In particular, a special thanks to Marcelo Menezes Morato, for the work done together and his great help during my research stay.

I would really like to thank my parents, grandparents and brothers, for being always on my side supporting me. As well as, to my closest friends for their encouragement and support.

Finally, I would like to dedicate this thesis to my beloved wife Virginia, for her patience, support and unconditional love. Especially, for blessing me with Sara, my sweet daughter.

Contents

List of Figures

List of Tables

Acronyms

AFTCS Active Fault-Tolerant Control System
CHP Combined Heat and Power
CIESOL Centro de Investigación en Energía Solar
CSTR Continuous-Stirred Tank Reactor
DNO Distribution Network Operator
DOS Dedicated Observer Scheme
EKF Extended Kalman Filter
EMS Energy Management System
FDD Fault Detection and Diagnosis
FT-MPC Fault-Tolerant Model Predictive Control
FTC Fault-Tolerant Controller
FTCS Fault-Tolerant Control System
GOS Generalised Observer Scheme
HE Heat Exchanger
IAE Integral Absolute Error
IFAC International Federation of Automatic Control
ISE Integral of Squared Error
ISS Input-to-State Stability
LPV Linear Parameter Varying
LTI Linear Time Invariant
LMI Linear Matrix Inequalities
LPV-RUIO Linear Parameter Varying Reduced-order Unknown Input Observer
LPV-UIOO Linear Parameter Varying Unknown Input Output Observer
MHE Moving Horizon Estimation
MHE-MPC Moving Horizon Estimation - Model Predictive Control
MPC Model Predictive Control
NMPC Non-linear Model Predictive Control
OCE On-line Computational Effort
PFTCS Passive Fault-Tolerant Control System
PI Proportional-Integral
PID Proportional-Integral-Derivative
PJL Parameterised Jacobian Linearisation
qLPV quasi-Linear Parameter Varying
qLPV-MPC quasi-Linear Parameter Varying Model Predictive Control
QP Quadratic Programming
RMS Root Mean Square

sMPC standard Model Predictive Control
ST Solar-Thermal
TV Total Variance
UIO Unknown Input Observer

Abstract

Modern plants rely on sophisticated control systems to meet performance and stability requirements. In particular, a conventional feedback control design for a complex system may result in unsatisfactory performance, or even instability, in the event of malfunctions in actuators, sensors or other system components. Hence, in concordance with Gene Kranz's[1] epic quote "*Failure is not an option*", specially in safety-critical systems. For that reason, in order to circumvent such weaknesses, control systems must be designed to mitigate component malfunctions while maintaining the required levels of stability and performance.

Accordingly, fault-tolerant control systems are control schemes that possess the ability to accommodate component faults automatically. They are capable of maintaining overall system stability and acceptable performance in case such faults occur. Thus, to design these kinds of systems, several aspects should be taken into account. Being these the fault type considered, its classification, the use or not of a mathematical model of the plant, the selection of a fault detection and diagnosis method, the adoption of a controller reconfiguration strategy, among others.

In view of these aspects, this thesis addresses the design, development and evaluation of fault-tolerant controllers for typical industrial processes, which ensure the compliance of operational constraints despite the presence of faults. To begin with, the current state-of-art and the main specific concepts are introduced. Then, two model-based strategies are presented. On the one side, the design of a novel observer-based fault detection and diagnosis scheme and the development of an adaptive predictive controller are combined to deploy a non-linear active fault-tolerant control system, on the basis of the linear parameter varying system representation. The controller stability conditions and the observers design are established on terms of linear matrix inequalities problems. This proposed scheme is evaluated on typical non-linear chemical industrial processes. On the other hand, an optimisation-based fault-tolerant predictive controller was proposed to develop a tertiary-level energy management system, based on a sugarcane distillery power plant. This strategy guarantees the uninterrupted and efficient energy generation on an industrial microgrid, choosing between different energy sources to overcome the fault effects.

Lastly, it is important to remark that for each proposed scheme a realistic simulation scenario was presented. Enabling vast discussions about its performance and effectiveness, via graphical observations and metric indices.

[1] Chief Flight Director of Gemini, Apollo and Space Shuttle missions at NASA

Resumen

Las plantas modernas dependen de sofisticados sistemas de control para cumplir con los requisitos de rendimiento y estabilidad. En particular, el diseño de un controlador por retroalimentación convencional para un sistema complejo, puede resultar en un desempeño insatisfactorio, o incluso en inestabilidad, frente al mal funcionamiento de actuadores, sensores u otros componentes del sistema. Por lo tanto, en concordancia con la cita épica de Gene Kranz[2], "*Failure is not an option*", especialmente en sistemas críticos en seguridad. Por esa razón, para evitar tales debilidades, los sistemas de control deben diseñarse para mitigar el mal funcionamiento de los componentes mientras se mantienen los niveles requeridos de estabilidad y desempeño.

En consecuencia, los sistemas de control tolerantes a fallas son esquemas de control que poseen la habilidad de soportar fallas en sus componentes. Son capaces de mantener la estabilidad general del sistema y un rendimiento aceptable en caso de que ocurran tales fallas. Así, para diseñar este tipo de sistemas se deben tener en cuenta varios aspectos. Siendo estos el tipo de falla considerada, su clasificación, el uso o no de un modelo matemático de la planta, la elección de un método de detección y diagnóstico de fallas, la adopción de una estrategia de reconfiguración del controlador, entre otros.

En vista de estos aspectos, esta tesis aborda el diseño, desarrollo y evaluación de controladores tolerantes a fallas para procesos industriales típicos, que aseguren el cumplimiento de las limitaciones operativas a pesar de la presencia de fallas. Primero, se introduce el estado de arte y los principales conceptos específicos. Luego, se presentan dos estrategias basadas en modelos. Por un lado, el diseño de un nuevo esquema de detección y diagnóstico fallas basado en observadores y el desarrollo de un controlador predictivo adaptativo se combinan para implementar un sistema de control tolerante a fallas activo no lineal, sobre la base de la representación de sistemas lineales de parámetros variables. Las condiciones de estabilidad del controlador y el diseño de los observadores se establecen en términos de problemas de desigualdades lineales matriciales. Este esquema propuesto se evalúa en típicos procesos industriales químicos no lineales. Por otro lado, se propuso un controlador predictivo tolerante a fallas basado en optimización para desarrollar un sistema de gestión de energía de nivel terciario, basado en una planta de energía de una destilería de caña de azúcar. Esta estrategia garantiza la generación de energía ininterrumpida y eficiente en una microrred industrial, eligiendo entre diferentes fuentes de energía para superar los efectos de las fallas.

[2] Director de vuelo de las misiones Gemini, Apollo y Space Shuttle en la NASA

Por último, es importante señalar que para cada esquema propuesto se presentó un escenario de simulación realista. Habilitando amplias discusiones sobre su desempeño y efectividad, a través de observaciones gráficas e índices métricos.

Chapter 1

Introduction

This chapter presents the origin and motivation, as well as the research area state-of-the-art and objectives of this thesis, which will be separately detailed in different sections. Additionally, a brief outline is also provided, which introduces the contents and contributions of each chapter.

1.1 Motivation

Nowadays in process industry, the use of computer algorithms enables operators to perform advanced control system strategies, offering a wide variety of possibilities. This improvement allows the achievement of the desired aims, in terms of efficiency and reliability (e.g. optimising costs and control efforts). Thus, from the growing demand for efficiency, the strict safety standards and the hardening of the environmental requirements, modern control systems and their algorithms have become more complex. Unfortunately, all those advantages and features are, to a greater or lesser degree, subject to the proper behaviour of all components involved in the system. However, in existing industrial systems, faults will inevitably occur. In other words, the controller design for complex systems, in case of malfunctions on actuators, sensors or other components, can result in unsatisfactory performance, or even lead to system instability.

Therefore, new approaches are addressed in the controllers design to mitigate components malfunction effects, while maintaining the desired stability and performance properties of the system. This is particularly important for safety-critical systems, such as airplanes, nuclear power plants, and chemical industries that process hazardous materials. In these systems, the consequence of a minor component malfunction can be catastrophic and therefore the demands for reliability, safety and fault tolerance are generally high.

Consequently, it is necessary to design control systems which are capable of tolerating potential faults in its components, maintaining the desired behaviour. If this is successful, the system operates even after the faults income, possibly with a short time of degraded performance until the control algorithm adapts to faulty system. In the literature, this control approach is often known as *Fault-Tolerant Control System* (FTCS). More precisely, the FTCSs have the ability to accommodate component malfunctions automatically. They are capable of maintaining an overall system stability and an acceptable performance in the event of such faults. In other words, a closed-loop control system that can tolerate component malfunctions, while maintaining the desired performance and stability characteristics, is an FTCS [1, 2].

FTCS is a strategy consisting of a collection of techniques that have been developed with the aim of increasing robustness, reliability, and as a consequence, reducing the risks generated by dangerous behaviours. In general, FTCSs are classified into two types: *Passive Fault-Tolerant Control System* (PFTCS) and *Active Fault-Tolerant Control System* (AFTCS).

The PFTCSs are fixed control structures, which are designed to be robust against certain classes of faults previously considered, which are treated essentially as disturbances. The benefit of this approach is that it does not require fault diagnosis or on-line controller reconfiguration, but as a counterpart, its capabilities for fault-tolerance are limited [3].

In contrast, AFTCSs respond to the system component malfunctions in an active way by reconfiguration, so that the stability and an acceptable performance of the entire system can be maintained. Specifically, a system that includes the AFTCS characteristics is designed to adapt faults at an early stage of its development, so that a minor fault in a subsystem does not evolve into general failures in the system. That is, the AFTCS actively estimate fault magnitudes and uses them to compensate the fault effects through the closed-loop control system. Consequently, AFTCS can be commanded to either perform the complete shutdown of the plant safely, or to retain some portion of its control integrity in the event of some specific faults. Thus, the biggest advantage of AFTCS over the other control strategies to tolerate faults, is the fact that AFTCS makes “intelligent” use of the available information about states and redundancies included in the system, in order to increase its availability.

Accordingly, AFTCS have been used in various applications, such as: safety-critical systems (nuclear reactors, aircraft, guided missile systems), cost-critical systems (space structures, space vehicles, automated submarine vehicles) and volume-critical systems (automotive assembly processes, mobile communication networks, automated highways) [1, 2, 4, 5, 6, 7, 8, among others].

Furthermore, in order to understand the different study areas covered by AFTCS, a brief description of its sub-systems is provided. Typically, as shown in Figure 1.1, an AFTCS could be divided into two sub-systems:

(1) *Fault Detection and Diagnosis* (FDD),

(2) *Fault-Tolerant Controller* (FTC).

Usually, the way to address the AFTCS design is to synthesise the FDD and FTC modules separately. This idea is achieved based on the satisfaction of the separation principle and despising the bidirectional interactions between them, which results from disturbances and uncertainties [9].

Thus, into the FDD module, any fault affecting the system is detected and isolated as quickly as possible, and then the fault magnitude is estimated. Then, based on the information about the post-fault system (fault location and magni-

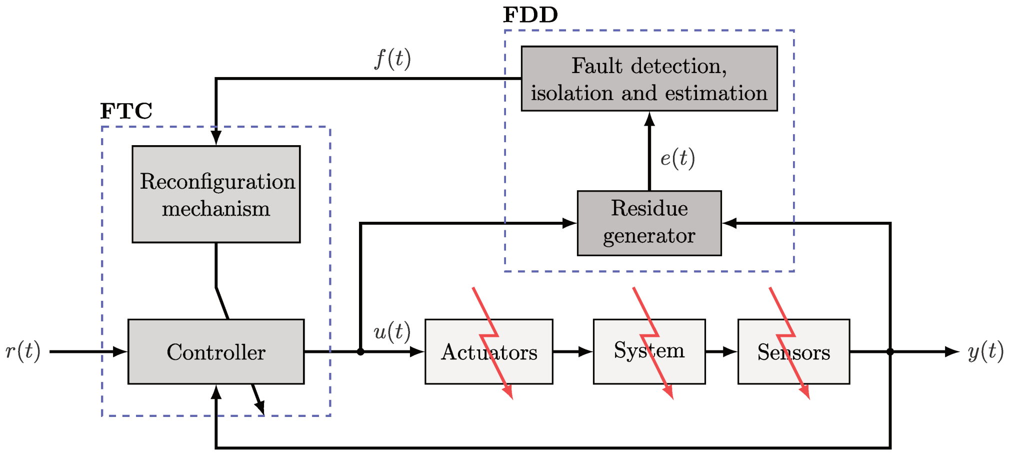


Figure 1.1: AFTCS's overall structure.

tude), the reconfiguration mechanism performs the controller redesign, in order to maintain system stability, the desired dynamic characteristic and an acceptable steady state performance.

In view of this, it is possible to distinguish at least two approaches of FTC. The first one covers the adaptation techniques of classical controllers (i.e. *Model Reference Adaptive Control* (MRAC), *Gain Scheduling Control* (GSC), *Self-Tuning Control* (STC), among others) [10, 11, 12], and the second one regards on modern mathematical tools that allow, not only an adaptation of the resulting controller dynamics, but also the constraints handling imposed by the income of faults in the system (i.e. *Dynamic Matrix Control* (DMC), *Generalised Predictive Control* (GPC), *Model Predictive Control* (MPC), among others) [8, 13, 14, 15, 16, 17].

Bearing these previous concepts in mind, the key points to consider in the AFTCS design strategy are:

i. an FDD scheme with high sensitivity to faults and robustness against model uncertainties, variations in operating conditions and external disturbances,
ii. a reconfiguration mechanism that leads the system, as much as possible, to recover the pre-fault behaviour,
iii. a suitable controller that can be easily reconfigurable considering constraints.

As a consequence, the critical point in the AFTCS design is the limited time window available for the execution of the FDD and FTC modules. Therefore, in the event of a fault income, the efficient use and management of redundancy (hardware, software and/or communication networks), the guarantee of system stability, and the transient and steady state performance, are some of the most important points to be consider.

Taking all these aspects into consideration, this thesis work addresses the fault-tolerant controllers design applied to typical systems of industrial processes, through the use of mathematical tools, ensuring the compliance with operational constraints despite the presence of uncertainties.

1.2 Brief history of FTCS

As might be expected, it is not an easy task to describe the historical development of fault monitoring and supervision methods, since the original contributions are scattered in technical literature. Perhaps the most elementary method, limit checking (or trend), is as old as the instrumentation itself, which dates from the end of the nineteenth century. Besides, monitoring through the use of ink-recorders, and later point-recorders, became a standard equipment since the middle of the twentieth century. Then, at the end of the twentieth century, in concordance with the advent of computers, the situation changed substantially, beginning the introduction of more complex supervisory and control algorithms [18].

From a practical application viewpoint, a significant amount of developments on FTCS was motivated by the design of control systems in aviation [19]. The primary goal was to provide a "self-repair" capacity in order to guarantee a safe landing, in the event of aircraft component malfunctions. These efforts have been promoted, in some degree, by two commercial aircraft events in the late of 1970s. The first one is the case of the Delta Air Lines flight 1080 (April 12, 1977) [20], where the left elevator jammed in "up" position, and the pilot had no indication for this malfunction. Fortunately, the pilot, based on his experience and the redundant performance of the Lockheed L-1011 aircraft, successfully reconfigured the remaining control elements and landed the aircraft safely. Moreover, in the American Airlines Flight 191 accident, where the McDonnell Douglas DC-10 aircraft crashed in Chicago (May 25, 1979), the pilot had only 15 seconds to react before the plane hit the ground. Subsequent investigations concluded that the accident could have been avoided [21]. Besides, an interesting article [22] provides further evidence of the need for AFTCS, showing that the fatal crash of the El Al flight 1862 involving a Boeing 747-200 cargo plane (October 4, 1992) could have also been avoided. These are three aircraft failures examples that show the need to implement fault-tolerant flight control systems.

Furthermore, in safety-critical industries such as chemical or nuclear power plants, the interest in the FTCSs design was intensified after the accident at the Tennessee Eastman chemical plant (Tennessee, United States, October 4, 1960), the incident in Three Mile Island (Pennsylvania, United States, March 18, 1979), the disaster at the Union Carbide chemical plant (Bhopal, India, December 3, 1984) and the tragedy at the Chernobyl nuclear power plant (Pripyat, Ukraine, April 26, 1986), just to mention the most important cases [1].

Process safety is a major concern in the researchers community, both in the past and today. Consequently, a wide variety of FTCS methods is available in literature, and some of them are collected in papers, such as: a deep revision of process fault detection and diagnosis is presented in the series of works [23, 24, 25]; a bibliographical review on reconfigurable fault-tolerant control is conducted in [1]; two brief surveys on fault detection methods are tackled in [26] and [27]; a recent survey on fault diagnosis and fault-tolerant techniques is exposed in [28, 29]; lastly, an even more recent review on fault detection and process diagnostics over industrial processes is exhibited in [30].

Based on these revisions, it is clear that the appropriate behaviour of an

AFTCS highly depends on a solid FDD strategy, in order to provide the accurate fault information before reconfiguration must be undertaken. Thus, the FDD module is perhaps the most difficult aspect involved in the AFTCS design. As a consequence, the development of an FDD module has a highly important role, because an early detection can avoid system shutdown, breakdowns, and even worse, the occurrence of potential catastrophes (especially for safety-critical systems). In that sense, the works [2, 31] show that the modular approach (FDD and FTC designed separately) presents its benefits, being more flexible for practical applications and, therefore, easier to test and implement.

In this way, several contributions and theoretical approaches about FDD modules in *Linear Time Invariant* (LTI) systems can be found in [11, 18, 31, 32, 33], and the references therein. Besides, there is a considerable amount of articles that involve model-based developments applied to non-linear systems. For instance, Kalman filters, fuzzy techniques, adaptive approaches or sliding-mode observer-based technologies have been recently reported [33, 34, 35, 36, 37, 38], where in some of them the stability conditions are guaranteed through the resolution of *Linear Matrix Inequalities* (LMI) problems. Moreover, even some newer papers have applied the *Moving Horizon Estimation* (MHE) framework to develop FDD schemes [39, 40, 41]. Lastly, another source of extensive publications is the database of the SAFEPROCESS technical committee of the *International Federation of Automatic Control* (IFAC) [42].

Nevertheless, in regards to the development of an FTC module, recent works are considering the use of the well-known/established MPC technique [13]. This is because MPC has inherent fault-tolerance properties [22], such as: dealing with constraints, updating internal model on-line and optimising the cost function at each iteration. Indeed, due that MPC optimises the cost function at each sampling step, it offers a high flexibility to deal with system constraints in both fault-free and post-fault conditions. This property is especially suitable to adjust control efforts depending on a timely fault information, as detailed in [8, 16, 39]. However, there are few works in literature, about AFTCS to accommodate multiple actuators and sensors faults upon non-linear systems, which motivates this current investigation. In short, the AFTCS problem is still a challenge due to the troubles of dealing with non-linearities and post-fault scenarios.

1.3 Contribution of thesis

As detailed in the prequel, the topic of AFTCS has received an increased attention in a wide range of academic and industrial communities, as well as the demands for safety and reliability have also increased. Although there are many works in literature that have developed AFTCS techniques, most of the applications in model-based fault-tolerant control systems have been used for mechanical, electrical and aerospace systems, while the data-based techniques have been dominant in petrochemical and chemical processes. This can be explained in the low availability or complexity of the model and the inherent natural process non-linearities. For this reason, the main aim of this thesis is the development, application and evaluation of novel model-based AFTCS strategies for practical purposes in typical industrial processes.

To achieve this, two different model-based approaches were addressed throughout this thesis: on the one hand, owing that the majority of the industrial plants are inherently non-linear and the faults may often amplify the non-linearities by driving plants from a relatively linear operating point into a more non-linear operating region, the design of novel FDD and FTC modules to develop a non-linear AFTCS are presented. There, the importance and practicality of *Linear Parameter Varying* (LPV) system representation provides the foundation of this work.

On the other hand, an optimisation-based fault-tolerant MPC strategy is proposed to develop a tertiary-level *Energy Management System* (EMS), which guarantees the uninterrupted and efficient energy generation on an industrial microgrid, based on a sugarcane power plant. Where the availability of several renewable energy sources highlights the capacity of MPC to manage them, seeking maximal profit and increasing sustainability, despite eventual faults.

In conclusion, the main contributions of this work are epitomised as follow:

- A novel observer-based FDD strategy for non-linear systems is presented. The observers' design is based on terms of LMI.
- An MHE procedure coupled to an MPC formulation is addressed to develop a non-linear predictive controller. Terminal ingredients to guarantee MPC stability are based on the resolution of off-line LMI problems.
- A model-based AFTCS for non-linear industrial processes is developed. An autonomous model-update and controller reconfiguration is formulated.
- A fault-tolerant MPC is proposed as an EMS. An MHE method is proposed to estimate both system states and incipient faults on renewable microgrids.
- Realistic numerical simulation scenarios are presented, showing the effectiveness of each method. Examples of typical chemical industrial processes are given, as well as a sugarcane industrial microgrid.
- Performance indices are used to demonstrate and quantify the effectiveness of the proposed methods.

1.4 Objectives

The overall objective of this thesis is to propose, discuss and explain in detail the design, testing, and validation of fault-tolerant predictive control strategies using model-based methods. In order to reach it, this overall objective is divided into several stage objectives.

- Design and develop FDD modules based on LPV observers.
- Evaluate and validate the proposed FDD approach.
- Design an appropriate FTC module to implement an AFTCS.
- Implement and evaluate the proposed model-based AFTCS strategy.
- Investigate and propose an AFTCS scheme to develop a fault-tolerant EMS of an industrial energy microgrid, based on a sugarcane processing plant.
- Simulate and evaluate the suggested tertiary-level EMS scheme.

1.5 Outline of the thesis

The remaining of this thesis is organised as detailed below.

In Chapter 2 some preliminary concepts involved in the AFTCS design are revisited. Chapter 3 presents the development of an FDD scheme based on a set of LPV observers. Chapter 4 investigates a polytopic MPC formulation for nonlinear processes. Moreover, Chapter 5 focuses on the application of the FDD and MPC methodologies described in previous chapters to conceive an AFTCS strategy for chemical industrial processes. Finally, Chapter 6 presents a fault-tolerant EMS of an industrial energy microgrid, particularly the case of a sugarcane distillery power plant. Concluding remarks and suggestions for future research are discussed in Chapter 7.

Chapter 2

Preliminary concepts

In this chapter, a brief overview of some fundamental concepts and general aspects concerning the design of AFTCS are recalled and presented.

2.1 Definitions

Before designing an AFTCS strategy it is necessary to know in detail the meaning of some involved basic concepts. These have been compiled from the IFAC SAFE-PROCESS technical committee [42], and from the existing literature [18, 43, 44]. They are listed below.

- **Fault:** An unpermitted deviation of at least one characteristic property or parameter of the system from its acceptable/usual/standard condition.
- **Failure:** A permanent interruption of a system's ability to perform a required function under specified operating conditions.
- **Disturbance:** An unknown (and uncontrolled) input acting on a system.
- **Perturbation:** An input acting on a system, which results in a temporary departure from the current state.
- **Symptom:** A change of an observable quantity from normal behaviour.
- **Error:** A deviation between a measured or computed value (of an output variable) and the true, specified or theoretically correct value.
- **Residual:** A fault indicator, based on a deviation between measurements and model-equation based computations.
- **Monitoring:** A continuous real-time task of determining the conditions of a physical system, by recording information, recognising and indicating anomalies in the behaviour.
- **Supervision:** Monitoring a physical system and taking appropriate actions to maintain the operation in the case of faults.
- **Protection:** Means by which a potentially dangerous behaviour of the system is suppressed if possible, or means by which the consequences of a dangerous behaviour are avoided.
- **Fault detection:** Determination of the faults present in a system and the time of detection.
- **Fault isolation:** Determination of the kind, location and time of detection

of a fault. Follows fault detection.

- **Fault identification:** Determination of the size and time-variant behaviour of a fault. Follows fault isolation.
- **Fault diagnosis:** Determination of the kind, size, location and time of detection of a fault. Follows fault detection. Includes fault isolation and identification.

2.2 Faults classification

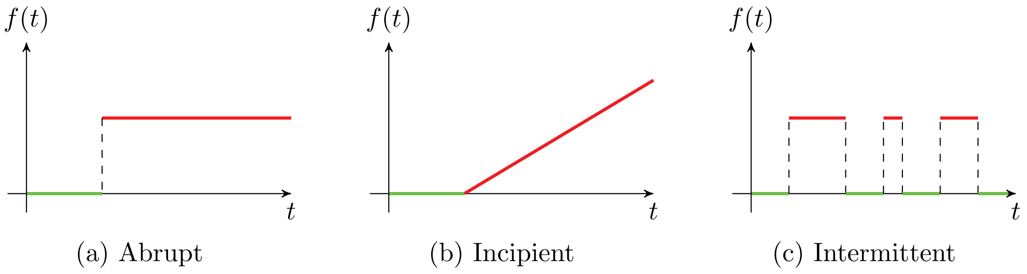
According to the above definitions a fault is an unpermitted deviation of at least one characteristic property or parameter of a system from its standard condition. It tends to degrade the system's performance but may not result in complete loss of system functionality [45]. Faults take place in different parts of a system and are classified according to where they occur in the system [46].

These faults and their causes are,

- **Actuator faults:** these are actuator malfunctions that trigger discrepancies between the computed and real control actions. These include, inter alia, gear damages, valve clogs and bearing aging.
- **Sensor faults:** these are discrepancies between the measured and true values of system output variables. Such faults describe sensor biases, thermal drifts or accumulation of tartar in sensor elements, among others.
- **Component faults:** these occur in the plant components themselves and often result in a change in the dynamical behaviour of the controlled system. These are variations in process parameters, such as heat exchanger coefficient, mass flow leaks or jammed conveyors.

Time-behaviour

On the whole, no particular time-behaviour will be assumed or employed to design residual generators[1]. In the analysis, however, it will be distinguished some typical time-functions, such as abrupt, incipient and intermittent. Accordingly, in Figure 2.1 their typical time-behaviour is sketched.



(a) Abrupt (b) Incipient (c) Intermittent

Figure 2.1: Classification of faults according to their time-behaviour.

[1] A more detailed explanation about residual generators is found in Section 2.4

Faults representation

Considering an LTI system with *Multiple-Input Multiple-Output* (MIMO) without presence of faults, disturbances and model uncertainties, whose input-output relationship is

$$y(t) = \Sigma u(t)$$

where Σ means the state space system representation

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t)\\ y(t) &= Cx(t) + Du(t)\end{aligned}$$

being $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}^p$ the state, input and output vector, respectively. In addition, A, B, C and D are constant matrices of appropriate dimensions [47].

Therefore, the faults that could affect the system are:

- actuator faults (f_u), at inputs;
- system component faults (f_Σ);
- sensor faults (f_y), on outputs.

These types of faults are suitably introduced in Figure 2.2.

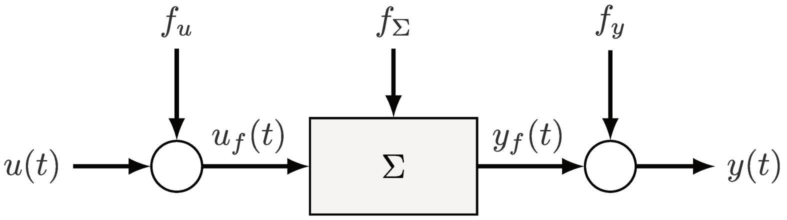


Figure 2.2: Input (f_u), sensor (f_y) and component (f_Σ) faults.

In particular, through this thesis, only the actuator and sensor faults are considered. Furthermore, the faults belong to one of the following categories [45]:

Additive faults These are considered as additional input signals, which affect process behaviour. Therefore, the system model with actuator and sensor additive faults is expressed as,

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) + F_u f_u(t)\\ y(t) &= Cx(t) + Du(t) + F_y f_y(t)\end{aligned}$$

where $f_u \in \mathbb{R}^q$ and $f_y \in \mathbb{R}^l$ are the input and output fault vectors, respectively. In addition, $F_u \in \mathbb{R}^{n \times q}$ and $F_y \in \mathbb{R}^{p \times l}$ are the fault matrices.

Multiplicative faults These occur as changes in system matrices, usually represented as a loss of effectiveness. Consequently, the system model affected by multiplicative faults is expressed as,

$$\begin{aligned}\dot{x}(t) &= A_f x(t) + B_f u(t)\\ y(t) &= C_f x(t) + D_f u(t)\end{aligned}$$

where

$$A_f = A + \Delta A; \; B_f = B + \Delta B; \; C_f = C + \Delta C; \; D_f = D + \Delta D$$

being ΔA, ΔB, ΔC and ΔD the system parameters deviation, from their nominal values.

In this context, it is important to note that disturbances are also considered as unknown extra inputs acting on the system. Thus there is no physical difference between disturbances and additive faults. This distinction is, indeed, subjective. As a general rule, the disturbances wished to be detected are considered faults, while the rest are simply disturbances [45].

Moreover, note that disturbances and model uncertainties have similar effects on the system. Disturbances are usually represented by unknown extra input signals that have to be added up to the system output. Besides, model uncertainties change the model parameters in a similar way as multiplicative faults. However, an important distinction between disturbances, model uncertainties and faults can be seen in the fact that disturbances and model uncertainties are always present, while faults may be present or not [2].

Finally, in this thesis the noise signals are also considered as unknown extra inputs to the system, but they are assumed to exhibit random behaviour, $\mathcal{N}(0, \sigma_n^2)$. Thus, any signal who exhibits a statistical distribution with arithmetic mean different from zero, is considered as a disturbance.

2.3 Redundancy

The use of redundancy in its many forms is a common strategy used to make more reliable systems. In engineering, redundancy corresponds to the duplication (or even more) of system components, or critical functions, generally as a backup. In particular, in control systems where safety requirements are the greatest priority (airplanes, nuclear power plants or chemical industries) all or some of their components are tripled, or even higher redundancy orders are implemented. In short, the faults are detected and patched by redundant elements. As a consequence, due to the fact that these components rarely fail and faults are expected to occur independently, the probability that all redundant components will fail is extraordinarily small; often only outweighed by other factors, such as human error. Thus, redundancy is the key ingredient in any fault-tolerant system.

However, it is important to note that the use of redundancy may exhibit some potential counterproductive effects, decreasing rather than increasing system reliability. Specifically, three serious problems are analysed in [48]: (1) the catastrophic common-mode error problem; (2) the social shirking problem; and (3) the overcompensation problem.

Thus, one of the biggest problems is that adding extra components can inadvertently create a catastrophic common-mode error, which is generated by a fault that causes all the components to fail. That is, in complex systems, theoretical (or design) independence is not necessarily a fact. So there is a possibility of unplanned interactions between redundant components, encouraging the presence of common-mode errors.

Another way in which redundancy can backfire is when diffusion of responsibilities leads to “social shirking”. This is a common phenomenon in which individuals, or groups, reduce their reliability in the belief that others will take up the slack. It is rarely examined in the technical literature on safety and reliability.

Finally, the last way in which redundancy can be counterproductive is when the addition of extra components encourages individuals to increase production in dangerous ways. Leading individuals to engage in inherently risky behaviours, such as: driving faster, flying higher, producing more nuclear energy, etc.

In control system engineering, the redundant units are usually mechanisms, or algorithms, so these two last effects are only perceived when people are in charge of supervision systems.

In literature, two types of redundancies are distinguished (both of them are outlined in Figure 2.3, for the sensing redundancy case), usually called: *physical* and *analytical* redundancy.

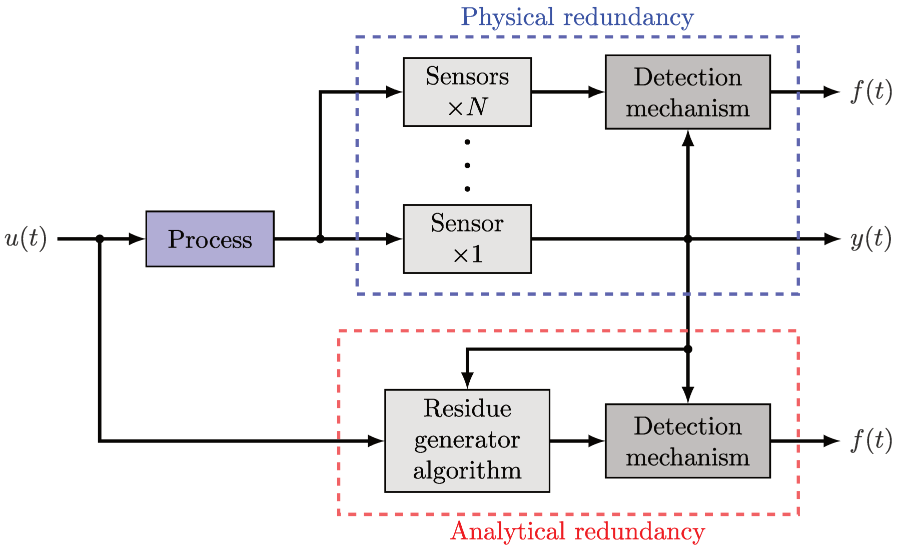


Figure 2.3: Physical and analytical redundancy.

Physical redundancy A traditional approach used in industry to guarantee the correct behaviour of control systems against faults, is the physical redundancy method. This strategy uses redundant sensors, actuators, computers and algorithms to measure and/or control a particular variable. Typically, a voting scheme is applied to the redundant systems to decide if and where a fault has occurred. The main advantage of this scheme is its high reliability and the direct fault isolation/compensation. However, the use of redundant hardware results in more expensive and complex systems. Therefore its application is commonly restricted to safety-critical components.

Analytical redundancy The intuitive idea of the analytical redundancy technique is to replace the physical redundancy with a process model which is implemented in an algorithm. A process model is a mathematical description of the monitored process behaviour, which can be obtained using the well-established process modelling techniques. The analytical redundancy introduces a new view point of redundancy, where reliability is achieved through software rather than strictly hardware components.

2.4 Residuals

Residuals are time-varying signals that are used to detect and isolate faults, and therefore this auxiliary signal constitutes an essential component in FDD modules. Typically, the residuals are designed to be zero under fault-free conditions (or close to zero when the process is affected by disturbances and modelling uncertainties), and distinctly different from zero when faults occur.

Consequently, information about faults are extracted by comparing the residual signals against suitable thresholds to avoid false alarms arising from measurement noises, disturbances or model uncertainties. The threshold values may be determined by statistical considerations, on the basis of the known or assumed noise statistics. Alternatively, they may be obtained experimentally, covering not only the noise effects but also the modelling errors. Therefore, the residual vector is the carrier of the fault information.

Note that, whilst a single residual signal is sufficient to detect faults, a set of residuals (or a vector of residuals) is required for fault isolation. That is to facilitate the fault isolation procedure, where the residual set needs to have distinctive properties. Thus, there are at least two fundamental residual enhancement approaches: *directional* and *structured* residuals.

Fixed direction residual The main idea behind this approach is to design a fault detection scheme, in which the residual vector receives specific directions, depending on the fault that is acting upon the system. In other words, designing a directional residual vector which lies in a fixed and fault-specified direction in the residual space, in response to a particular fault. Directional residuals support the isolation of simultaneous faults if the response directions are independent [49].

Structured residual Another method of fault isolation is to design a set of structured residuals. The main idea behind this approach is to design each residual to be sensitive to a subset of faults, whilst remaining insensitive to the rest. The design procedure consists of two steps, the first step is to specify the sensitivity and insensitivity relationships between residuals and faults according to the assigned isolation scheme, and the second is to design a set of residual generators according to the desired sensitivity and insensitivity relationships. The advantage of the structured residual set is that the diagnostic analysis is simplified to determine which of the residuals are non-zero [45].

Besides, the faults in which the residue must be sensitive are called *monitored* and those in which the residue must be insensitive are called *non-monitored.* The non-monitored faults are decoupled from residuals; as a result, the residual generation problem is essentially a decoupling problem. It is a fact that, when more faults are decoupled in each residue, greater is the possibility of isolating multiple faults. The cost for this improvement is in the need of more complex and model dependent residual generators.

The structured residual approach can be designed in two different ways: *dedicated* and *general* residual scheme.

- In a dedicated residual scheme, one measurement is fed into each residual generator; these generators are designed to be sensitive only to single faults. In observer-based literature, this is also known as a *Dedicated Observer Scheme* (DOS).
- An alternative approach is the general residual scheme, also known as the *Generalised Observer Scheme* (GOS). In this approach, each residual is designed to be sensitive to all but one fault.

Each technique has pros and cons, in DOS each observer is driven by only one output measurement and, as a consequence, the system states should be completely observable, which is not always the case in practical applications. On the other hand, in GOS, a set of residuals should be fired from different observers at the same time to decide which sensor is faulty, making it more reliable than DOS.

Residual generator The dynamic system used to generate residuals is called as *residual generator.* Its purpose is to generate fault indicating signals, using the available input and output information from the monitored system.

A residual generator looks for inconsistencies between the actual system variables and the mathematical model describing their relationship. That is, measurements are compared to analytically obtained values of the respective variables. Note that this analysis significantly depends on the available mathematical model representation.

Ideally, the residual should only be affected by faults. However, the presence of disturbances, noise and modelling errors also affect the residuals and thus interferes with the detection of faults. Therefore the residual generator needs to be designed robust against noise disturbances and model uncertainties. As a consequence, robustness is perhaps the most important requirement in residual generation.

In addition, the residual vector $r(t) \in \mathbb{R}$, is designed to become zero for the fault-free case

$$r(t) = 0 \Longleftrightarrow u(t) \neq 0 \wedge f(t) = 0$$

and non-zero for faulty cases,

$$r(t) \neq 0 \Longleftrightarrow f(t) \neq 0 \tag{2.1}$$

where $f(t)$ corresponds to any fault signal. Thus, the Equation (2.1) is the elementary rule involved in residual-based fault detection.

No matter what type of method is used, a residual generator is just an algorithm whose input consists of both input and output of the monitored system.

2.5 FDD methods classification

Usually, FDD methods are classified into two large groups, those that do not require a mathematical model of the monitored process, and those that do. In this thesis work, the model-based approach is used. However, to outline the general context, both are briefly introduced.

- **Model-free**. The FDD methods which do not use the mathematical model of the monitored process are called model-free methods. They range from physical redundancy [50], special sensors [51], through limit-checking [18], spectral and statistic analysis [26] up to expert systems [52], among others.
- **Model-based**. On the other hand, the FDD model-based methods are those which use an explicit mathematical model representation of the monitored process, and in general they rely on the analytical redundancy idea. As stated in Section 2.3 contrary to physical redundancy, analytical redundancy assesses the difference between the measured variables of the monitored process and their analytical computation. To do this, present and/or previous measurements data and the mathematical model of the monitored process are used [45]. Thus, the FDD model-based methods range from *parity relations* [18, 43], *parameter estimation* [45, 53], through *diagnostic observers* [54, 55], up to *Kalman filters* [56, 57].

It is important to note that this chapter is not intended to explain in detail all presented approaches, but the interested reader is invited to check the references appointed there in.

2.6 Linear parameter varying system representation

Historically, control system analysis and synthesis techniques have been developed using LTI models subject to bounded process and measurement disturbances [47, 58, 59]. However, the design of non-linear controllers demands a different strategy further than just obtaining a linearised model at an operating point, and then apply a robust technique to deal with model uncertainties. Hence, because of the difficulty to design a non-linear control system, many authors prefer to represent these systems by an LPV approach [60, 61, 62], for the observers design [34, 63, 64, among others]. The basic idea of this is to represent the system as an interpolation of i-th affine local models, which depend on time-varying scheduling parameters. These parameter values depend on endogenous variables, such as the states and inputs (typically available in real-time). That is, this technique represents an attractive solution to schedule a set of linear models by a convex weighting function. Thus, the LPV modelling framework is powerful since it allows the application of well-known linear-like control and estimation design tools to a wide range of non-linear systems. Furthermore, a key feature of the LPV paradigm resides in the ability to formulate control problems as convex optimisation problems involving LMI, which are efficiently solved using interior-point methods [65, 66]. A brief description of the procedure for the LPV system representation of a non-linear process model is presented here below.

Being the dynamic of a real (non-linear) industrial process represented by the following general state-space form:

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)), && (2.2a) \\ y(t) &= g(x(t), u(t)) && (2.2b) \\ x(0) &= x_0 \end{aligned}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}^p$ are the state, input and output vectors, respectively. Additionally, x_0 is the initial value of the state vector. Specifically, the Equation (2.2a) is called *state equation* and this determines the system dynamics. On the other hand, the Equation (2.2b) is termed as *output equation.*

Assumption 1. *Considering that the state and output functions $f(x(t), u(t))$ and $g(x(t), u(t))$ are continuously differentiable, and using the Parameterised Jacobian Linearisation (PJL) technique (see Appendix B), which is based on the first-order Taylor-series analysis, it is possible to approximate, or even represent, the dynamic behaviour of the system from the Equation* (2.2)*, in different operating points through a convex set of M affine models that depend on a time-varying parameter $\zeta(t)$ [61, 67].*

That is, the polytopic representation of affine LPV system is,

$$\begin{aligned} \dot{x}(t) &= \sum_{i=1}^{M} \mu_i\left(\zeta(t)\right)\left\{A_i x(t) + B_i u(t) + \Delta x_i\right\} \\ y(t) &= \sum_{i=1}^{M} \mu_i\left(\zeta(t)\right)\left\{C_i x(t) + D_i u(t) + \Delta y_i\right\} \end{aligned} \tag{2.3}$$

where A_i, B_i, C_i, D_i, Δx_i and Δy_i are constant matrices of appropriate dimensions, which represent the state, input, output and feedthrough matrices, respectively. Moreover, the weighting functions $\mu_i(\cdot)$ depend on a scheduling parameter $\zeta(t)$ assumed to depend on an endogenous variable (inputs or outputs). In addition, the polytopic term comes from the fact that the vector $\mu_i(\zeta(t))$ evolves over the unit M-simplex defined by

$$\mathcal{P} := \left\{\mu_i(\zeta(t)) : \sum_{i=1}^{M} \mu_i(\zeta(t)) = 1,\ \mu_i\left(\zeta(t)\right) \geq 0\right\}, \tag{2.4}$$

being $\mathcal{P} \in \Omega \subset \mathbb{R}^M$.

2.7 Unknown input observer

Definition 2.1 (Unknown Input Observer). *Consider an LTI system described as follows:,*

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + Ff_u(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{2.5}$$

where, $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$ and $f_u \in \mathbb{R}^q$ are the state, input, output and unknown input vector of the system, respectively. Moreover, A, B, C, D and F are known matrices with appropriate dimensions [47].

An observer is defined as an Unknown Input Observer (UIO) for the system described by Equation (2.5)*, if its state estimation error vector $e(t)$ approaches zero asymptotically, regardless of the presence of the unknown input (disturbance) in the system [68].*

2.8 Model-based predictive control

MPC (aka, *Receding Horizon Control* (RHC)) covers an extensive range of advanced control methods, which have been developed considerably over the last three decades, both within the research control community and industry. MPC is the most natural approach to the optimal control of processes subject to constraints. As a consequence, it has become a world-wide well-known/established control methodology, commonly used in the process industry.

Roughly speaking, as shown in Figure 2.4, the basic concept of MPC method is to use a dynamic model to forecast system behaviour, and optimise this forecast to produce the best control policy at the current time. Thus, models are therefore central to every form of MPC [13]. Specifically, MPC finds an optimal control policy by minimising a cost function over a finite receding horizon, including state, input and output constraints [69]. This cost function regards performance goals, such as reference tracking and disturbance rejection. In short, one of the main advantages of MPC over other control strategies is that it inherently considers and satisfies system constraints [70].

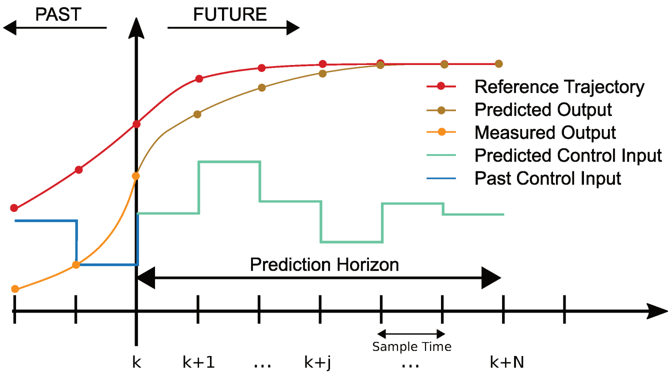


Figure 2.4: Model predictive control concept.

Consequently, the feedback control law is achieved via receding horizon technique. In other words, after obtaining a control sequence $\mathbf{u}$, by means of the optimisation problem, only the first control element is applied to the system. That is,

$$\kappa(x) = \mathbf{u}(0).$$

Afterwards, the prediction horizon is shifted by one sample, and the optimisation problem is performed again after the problem has been updated with new information from the measurements. Thus, the MPC considers a finite prediction horizon, at each sample time k.

Furthermore, there are many different MPC formulations in the literature [14], each of them are defined by an optimisation problem with different objective function and/or set of constraints. In general, the optimisation problem is posed as a minimisation problem in which the objective function penalises the distance between the reference and the predicted system evolution over the prediction horizon. However, in this section the focus is to introduce the preliminary aspects involved in MPC, where the solution is found by solving a constrained *Quadratic Programming* (QP) program.

Usually, the MPC formulations share three common elements, where different options can be chosen for each element giving rise to different algorithms. Being these itemised as follow:

- **Prediction model.** As was previously introduced, the main idea of MPC is to use a dynamic model of the plant to forecast system behaviour, and based on this prediction take the best control policy. In that sense, the model is the cornerstone of MPC. This should be complete enough to fully capture the process dynamics and allow the predictions to be calculated, and at the same time to be intuitive and permit theoretic analysis.
  For simplicity's sake, a discrete-time non-linear model form is considered:

$$\begin{aligned} x(j+1) &= f(x(j), u(j)), \\ x(0) &= x_0, \end{aligned} \tag{2.6}$$

  where $f(\cdot)$ is a continuous function of their arguments, $x \in \mathbb{X} \subset \mathbb{R}^n$ is the system state at sampling time k, x_0 is the initial state and $u \in \mathbb{U} \subset \mathbb{R}^m$ is the control input.
- **Objective function.** The MPC algorithms propose different cost functions for obtaining the control action. The general aim is that the future output should follow a determined reference target at the considered horizon. Consequently, a proper objective function $V_N(\cdot)$ is defined for the MPC optimisation problem by:

$$V_N(x; \mathbf{u}) = \sum_{j=0}^{N-1} \ell(x(j), u(j)) + V_f(x(N)) \tag{2.7}$$

  where N is the prediction horizon length, $u(j)$ is the future control action computed at the current sampling time k and $x(j)$ is the predicted state at sampling time k, starting from the initial condition $x = x(k)$. Moreover, $\ell(x(j), u(j))$ and $V_f(x(N))$ are the stage and the terminal cost functions, respectively.
- **System constraints.** Bearing in mind the previous statements, the sequence of control actions $\mathbf{u} := \begin{bmatrix} u(0) & u(1) & \dots & u(N-1) \end{bmatrix}$ is computed by minimising the objective function while satisfying the system constraints.

Thus, the control problem $P_N(x)$ is defined as follow:

$$\begin{aligned} \min_{\mathbf{u}} \quad & V_N(x, \mathbf{u}) \\ \text{s.t.} \quad & x(0) = x, \\ & x(j+1) = f(x(j), u(j)), \\ & u(j) \in \mathbb{U}, \quad x(j) \in \mathbb{X}, \quad \forall j \in \mathbb{Z}_{0:N-1} \\ & x(N) \in \mathbb{X}_f \end{aligned} \tag{2.8}$$

where $\mathbb{U} \subset \mathbb{R}^m$, $\mathbb{X} \subset \mathbb{R}^n$ and $\mathbb{X}_f \subset \mathbb{R}^n$ are the input, state and terminal constraint sets, respectively. Moreover $\mathbb{U}$ is a compact set, and $\mathbb{X}$, $\mathbb{X}_f$ are closed sets.

Most works on MPC ensure stability by the addition of a suitable terminal cost and terminal constraint to the optimal control problem solved on-line; the stabilising conditions are essential conditions that the terminal cost and constraint must satisfy to ensure stability and recursive feasibility [71].

In a resume manner, there exist different stabilising formulations of MPC [72]:

- terminal equality constraint: the stability condition is achieved by adding a new constraint over the state at the end of horizon $x(N)$,

$$x(N) = x_s$$

  where x_s is the desired steady state.
- terminal cost: the stability condition is achieved by adding an extra term to the cost function that penalises the state at the end of the horizon. This term is called terminal cost function $V_f(x(N))$.
- terminal inequality constraint: the terminal equality constraint is replaced by a set $\mathbb{X}_f$ that has to fulfil certain conditions,

$$x(N) \in \mathbb{X}_f.$$

  This approach provides a larger domain of attraction and less numerical problems than the terminal equality constraint.
- terminal cost and inequality constraint: this approach is the result of the union of the last two techniques, adding a terminal cost to the cost function and using an inequality terminal constraint.

In [13, 71, 72, 73, among others] all these formulations are analysed and it is established that adding a terminal cost $V_f(x(N))$, together with a suitable terminal state domain $\mathbb{X}_f$, has resulted to be essential to the stabilising design. These conditions, also known as *terminal ingredients*, can be written down as follows:

Let $\mathbb{X}_f$ be a set in $\mathbb{R}^n$, let $V_f(x(N))$ be a positive definite function, continuous at the origin and let $\kappa(x)$ be a control law such that,

- $0 \in \mathbb{X}_f$;
- $f(x, \kappa(x)) \in \mathbb{X}_f$, $\forall x \in \mathbb{X}_f \subseteq \mathbb{X}$;
- $V_f(f(x, \kappa(x)) - V_f(x(N)) \leq -\ell(x, \kappa(x))$, $\forall x \in \mathbb{X}_f$;
- $\kappa(x) \in \mathbb{U}$, $\forall x \in \mathbb{X}_f$.

Based on this, the optimal cost can be considered as a Lyapunov function [72]. Then, assuming feasibility of the initial state, an MPC solving the optimisation problem from Equation (2.8) guarantees asymptotic stability of the origin. That is, the invariant condition on terminal set $\mathbb{X}_f$ ensures feasibility of the closed-loop system, while the condition on terminal function $V_f(x(N))$ guarantees convergence.

At this point, in short, the MPC implementation could be represented in the following algorithm:

Algorithm 1: MPC method

```
Measure $x(k)$,
Obtain $\mathbf{u}$ by solving the problem from Equation (2.8),
Apply the control law, $\kappa(x) = \mathbf{u}(0)$,
Time update, $k := k + 1$,
Repeat from step 1.
```

2.9 Moving horizon estimation

While MPC method had its origins in industry [74, 75], MHE was developed primarily by researchers in academia [13, 76, 73], and was built on the success of the receding horizon paradigm. Roughly speaking, MHE has been suggested as a practical strategy to incorporate inequality constraints in states and parameters estimation. The basic idea of MHE is to formulate the estimation problem as a QP using a backward moving horizon of the available measurements.

Compared to full information estimation techniques, MHE reduces the computational burden by considering a finite horizon, however, it is non-trivial to summarise the effect of the discarded data on the current states. Thus, the measurements information which are not included in the estimation window is assimilated into the objective function through an extra term called *arrival cost*. In short, the arrival cost can be considered as the analogous to the *terminal cost* in MPC. Accordingly, arrival cost is fundamental to guarantee stability in MHE.

The best choice of the arrival cost remains an open issue in the domain of MHE research [13, 77, 78]. One reasonable solution is to approximate the arrival cost for the constrained problem with the arrival cost for the unconstrained problem. That is, for linear unconstrained systems, the standard *Kalman Filter* (KF) covariance updated formula can be used to express the arrival cost explicitly [76]. However, for non-linear or constrained systems, a general analytical expression for the arrival cost is rarely available, and it is often estimated using linearisation approximation as in the *Extended Kalman Filter* (EKF) formulation [79]. This method to approximate the arrival cost does not consider the bounds on state estimates in a systematic manner, but it allows to reduce the size of the optimisation problem, while retaining a good performance and robustness [13].

By and large, MHE is a multi-variable estimation algorithm that uses:

- an internal dynamic model of the process,
- a history of past input and output measurements, and

- an optimisation cost function $V_T(\cdot)$ over the estimation horizon

to compute the optimum states and parameters estimation.

Specifically, the cost function for the MHE optimisation is given by:

$$V_T(x) = \ell_{ac}(\hat{x}(k-T)) + \sum_{i=k-T}^{k-1} \left(\ell(\omega(i), \nu(i))\right) ,$$

where $\ell_{ac}(\cdot)$ is the arrival cost and $\ell(\cdot)$ denotes the main optimisation cost along the backward horizon. Moreover, ω represents process disturbance and ν the output measurement noise. Therefore, the main optimisation function is given as follows[2]:

$$\ell(\omega(i), \nu(i)) = \|\omega(i)\|_Q^2 + \|\nu(i)\|_R^2,$$

where Q and R are adequate weighting matrices.

On the other hand, the arrival term ℓ_{ac} included in the cost function is used to penalise the distance between the state estimated at the start of the backward horizon $(k-T)$ and a previous state estimation x. In other words, the arrival cost weights the influence of the past information outside the estimation horizon, i.e. from $t = 0$ to $t = k-T$, in a single concatenated term. This offset is given by:

$$\ell_{ac}(\hat{x}(k-T)) = \|\hat{x}(k-T) - x\|_P^2$$

where P is a positive definite penalty matrix and as previously stated, the estimate $\hat{x}(k-T)$ is often given by an EKF algorithm. More details on why this arrival term is necessary are given in [80, 81].

Finally, at each sampling instant k, the optimisation problem defined in Equation (2.9) below is solved, and a current state estimation becomes available for control purposes.

$$\begin{aligned} \min_{x} \quad & V_T(x) \\ \text{s.t.} \quad & x(i+1) = f(x(i), u(i)) + \omega(i), \\ & y(i) = g(x(i)) + \nu(i), \\ & x \in \mathbb{X}, \qquad\qquad i \in \mathbb{Z}_{k-T:k}. \end{aligned} \tag{2.9}$$

At the next sampling instant, the following arrival term $\ell_{ac}(\cdot)$ is updated by an EKF guess and the horizon window T slides forward by one unit.

At this point, in a resume manner, the MHE implementation could be represented in the following algorithm:

Algorithm 2: MHE method

```
1 Collect the data set based on the available information from the last $T$ steps,
2 Compute the arrival cost $\ell_{ac}(\cdot)$, through an EKF algorithm,
3 Estimate the states, solving the constrained optimisation problem from Equation (2.9),
4 Time update, $k := k + 1$,
5 Repeat from step 1.
```

[2] The weighted norm $||z||_M^2$ denotes the weighted norm of z, $z^T M z$

Chapter 3

Observer-based fault detection and diagnosis

This chapter presents the development of a novel FDD scheme based on a set of observers. Specifically, the design of two types of observers applied to LPV systems is addressed, being these a *Linear Parameter Varying Reduced-order Unknown Input Observer* (LPV-RUIO) and a full-order *Linear Parameter Varying Unknown Input Output Observer* (LPV-UIOO). Then, based on them, two banks of generalised observers are constructed to generate residuals that allow the detection, isolation and diagnosis of actuator and sensor malfunctions over non-linear systems, which accept LPV representation. Furthermore, to evaluate the behaviour and capabilities of the proposed FDD scheme, it was simulated on two typical chemical industrial processes, firstly in a *Heat Exchanger* (HE) and secondly in a *Continuous-Stirred Tank Reactor* (CSTR).

Finally, it should be noted that the main results of this chapter can be found in works [55, 82, 83].

3.1 Introduction

As stated previously, the appropriate behaviour of an AFTCS highly depends on a solid FDD module. This is specially critical in industrial processes, where the problem is still a challenge due to the troubles of dealing with non-linearities and model mismatches. In light of the foregoing, this chapter presents the design of a novel FDD scheme, composed of a bank of two types of observers, applied to LPV systems.

The main motivation for the use of two types of observers, to deal with actuators and sensors faults, is the possibility to exploit each of its specific goodness. That is, the use of LPV-RUIOs allows an easy way to compute the actuator fault estimation (without the use of extended observer, or the computation of the bad conditioned pseudo-inverse matrices), enabling a design with more degree of freedom, relieving the designer work to select the desired actuator input as unknown input without resulting unfeasible. On the other hand, the LPV-UIOOs are the simplest way to deal with sensors faults, with a better trade-off about performance and off-line design complexity, which is ideal to improve the efficiency of the proposed strategy. In such a way, the observers design is mainly doing off-line, and only a slight use of computational resources is needed at each sample time to perform on-line the FDD tasks, enabling its application as a part of an AFTCS.

Notation With the purpose of simplifying the chapter's structure, the duality between continuous-time and discrete-time is assumed equivalent, except in cases indicated explicitly. For that, the differential operator ρ is used, which denotes the time derivative for continuous-time models and the step forward shift operator for discrete-time models. That is, $\rho x(\iota)$ represents $\dot{x}(t)$ and $x(k+1)$ for continuous-time and discrete-time, respectively. In addition, ι represents the time variable ($t \in \mathbb{R}$ and $k \in \mathbb{Z}$) for continuous-time and discrete-time cases. A similar notation was used in [84].

3.2 LPV system with additive faults

Based on a non-linear system represented by an LPV model (see Section 2.6), that consider the income of additive input and output faults. That is, the polytopic representation of an affine LPV system is,

$$\begin{aligned} \rho x(\iota) &= \sum_{i=1}^{M} \mu_i\left(\zeta(\iota)\right)\{A_i x(\iota) + B_i u(\iota) + F_i f_u(\iota) + \Delta x_i\} \\ y(\iota) &= \sum_{i=1}^{M} \mu_i\left(\zeta(\iota)\right)\{C_i x(\iota) + D_i u(\iota) + H_i f_y(\iota) + \Delta y_i\} \end{aligned} \tag{3.1}$$

where $x(\iota) \in \mathbb{R}^n$, $u(\iota) \in \mathbb{R}^m$, $f_u(\iota) \in \mathbb{R}^q$, $y(\iota) \in \mathbb{R}^p$, $f_y(\iota) \in \mathbb{R}^o$ are the state vector, the input vector, the fault input vector, the output vector and the fault output vector, respectively. Additionally, A_i, B_i, C_i, D_i, F_i, H_i, Δx_i and Δy_i are constant matrices of appropriate dimensions.

Moreover, throughout this chapter, the following assumptions are made.

Assumption 2. *In general, for strictly proper systems, the matrices D_i are null and, as a consequence, the output equation in Equation* (3.1) *is reduced to a simple weighting combination of the states and faults variables. On the other hand, the matrices C_i, for this study purpose, are composed by identity matrices of appropriate dimensions $\mathbb{I}_{p\times n}$. Thus, the final output equation of Equation* (3.1) *results,*

$$y(\iota) = Cx(\iota) + Hf_y(\iota). \tag{3.2}$$

Assumption 3. *The actuators and sensors faults vectors satisfy the following euclidean-norm bounded constraint,*

$$\|f_u(\iota)\| < \beta_u, \qquad \|f_y(\iota)\| < \beta_y,$$

where $\beta_u > 0$ and $\beta_y > 0$ are known constants ($\beta_u \wedge \beta_y \in \mathbb{R}^+$).

Assumption 4. *Faults can occur in all system elements (actuators and sensors), but they do not occur at the same time.*

Consequently, taking all the above into account, Section 3.3 introduces the design procedure for an LPV-RUIO. Moreover, Section 3.4 presents the design procedure for an LPV-UIOO. Lastly, the FDD strategy built using these observers is presented in Section 3.5.

3.3 Design of LPV-RUIO

Based on the observer development for linear systems with unknown input [85], recently reformulated as a reduced-order observer applied to an LPV system with unknown input [82]. If the term $Hf_y(\iota) = 0$ is considered (no sensors faults occur in the system, simultaneously), and under the assumption that the rank of $F_i = q$ (being F_i a specific column of B_i, that corresponds with the faulty input to diagnose), then a set of non-singular matrices is selected as,

$$T_i = \begin{bmatrix} N_i & F_i \end{bmatrix}, \qquad N_i \in \mathbb{R}^{n\times(n-q)}.$$

Thereby, the system from Equation (3.1) is equivalent to,

$$\rho\bar{x}(\iota) = \sum_{i=1}^{M} \mu_i(\zeta(\iota)) \left\{ \bar{A}_i \bar{x}(\iota) + \bar{B}_i u(\iota) + \bar{\Delta} x_i + \bar{F}_i f_u(\iota) \right\} \tag{3.3a}$$

$$y(\iota) = \sum_{i=1}^{M} \mu_i(\zeta(\iota)) \left\{ \bar{C}_i \bar{x}(\iota) \right\} \tag{3.3b}$$

where

$$x(\iota) = T_i \bar{x}(\iota) = T_i \begin{bmatrix} \bar{x}_1(\iota) \\ \bar{x}_2(\iota) \end{bmatrix}; \ \bar{\Delta} x_i = T_i^{-1} \Delta x_i = \begin{bmatrix} \bar{\Delta} x_{i_1} \\ \bar{\Delta} x_{i_2} \end{bmatrix}$$

$$\bar{A}_i = T_i^{-1} \bar{A}_i T_i = \begin{bmatrix} \bar{A}_{i_{11}} & \bar{A}_{i_{12}} \\ \bar{A}_{i_{21}} & \bar{A}_{i_{22}} \end{bmatrix}; \ \bar{B}_i = T_i^{-1} B_i = \begin{bmatrix} \bar{B}_{i_1} \\ \bar{B}_{i_2} \end{bmatrix} \tag{3.4}$$

$$\bar{C}_i = CT_i = [CN_i \ CF_i]; \ \bar{F}_i = T_i^{-1} F_i = \begin{bmatrix} 0 \\ I_q \end{bmatrix}$$

with $\bar{x}_1(\iota) \in \mathbb{R}^{n-q}$ and $\bar{x}_2(\iota) \in \mathbb{R}^{q}$.

Because of Equation (3.4) it is observed that Equation (3.3a) involves directly the unknown input in the state $\bar{x}_2(\iota)$. As a consequence, it is possible to drop this state and rewrite the system from Equation (3.3) without the unknown input as,

$$\begin{bmatrix} I_{n-q} & 0 \end{bmatrix} \rho\bar{x}(\iota) = \sum_{i=1}^{M} \mu_i(\zeta(\iota)) \Big\{ [\bar{A}_{i_{11}} \ \bar{A}_{i_{12}}] \bar{x}(\iota) + \bar{B}_{i_1} u(\iota) + \bar{\Delta} x_{i_1} \Big\} \tag{3.5a}$$

$$y(\iota) = \sum_{i=1}^{M} \mu_i(\zeta(\iota)) \left\{ [CN_i \ CF_i] \bar{x}(\iota) \right\}. \tag{3.5b}$$

Assuming that $\bar{x}_2(\iota)$ is obtained from $y(\iota)$, and if the matrices $\begin{bmatrix} CF_i \end{bmatrix}$ have full column rank, then there exists non-singular matrices

$$U_i = \begin{bmatrix} CF_i & Q_i \end{bmatrix}, \qquad Q_i \in \mathbb{R}^{n\times(n-q)}$$

being

$$U_i^{-1} = \begin{bmatrix} U_{i_1} \\ U_{i_2} \end{bmatrix}, \qquad U_{i_1} \in \mathbb{R}^{q \times n},\ U_{i_2} \in \mathbb{R}^{(n-q)\times n}$$

multiplying both sides of Equation (3.5b) by U_i^{-1} and isolating,

$$\bar{x}_2(\iota) = \sum_{i=1}^{M} \mu_i(\zeta(\iota)) \left\{ U_{i_1} y(\iota) - U_{i_1} C N_i \bar{x}_1(\iota) \right\} \tag{3.6a}$$

$$y(\iota) = \sum_{i=1}^{M} \mu_i(\zeta(\iota)) \left\{ C N_i \bar{x}_1(\iota) \right\} \tag{3.6b}$$

afterwards, substituting Equation (3.6a) into Equation (3.5a) and combining it with Equation (3.6b) derive in Equation (3.1) transform to,

$$\rho \bar{x}_1(\iota) = \sum_{i=1}^{M} \mu_i(\zeta(\iota)) \Big\{ \tilde{A}_{i_1} \bar{x}_1(\iota) + E_{i_1} y(\iota) + \bar{B}_{i_1} u(\iota) + \bar{\Delta} x_{i_1} \Big\} \tag{3.7a}$$

$$y(\iota) = \sum_{i=1}^{M} \mu_i(\zeta(\iota)) \left\{ \tilde{C}_{i_1} \bar{x}_1(\iota) \right\} \tag{3.7b}$$

with $\tilde{C}_{i_1} = CN_i$, $\tilde{A}_{i_1} = \bar{A}_{i_{11}} - \bar{A}_{i_{12}} U_{i_1} C N_i$ and $E_{i_1} = \bar{A}_{i_{12}} U_{i_1}$.

At this point, if the pair $(\tilde{A}_{i_1},\ \tilde{C}_{i_1})$ is observable, following the conventional Luenberger observer design procedure [86] (see Appendix A), it is possible to design a reduced-order observer for an LPV system free of unknown inputs from Equation (3.7) as,

$$\rho \Phi(\iota) = \sum_{i=1}^{M} \mu_i(\zeta(\iota)) \left\{ K_i \Phi(\iota) + \bar{B}_{i_1} u(\iota) + L_i^* y(\iota) + \bar{\Delta} x_{i_1} \right\} \tag{3.8}$$

where $\Phi(\iota) \in \mathbb{R}^{(n-q)}$ is the observer state vector, $L_i^* = L_i + E_{i_1}$ and $K_i = \tilde{A}_{i_1} - L_i \tilde{C}_{i_1}$. Moreover, $L_i \in \mathbb{R}^{(n-q)\times(p-q)}$ are the gains of the observer to be designed.

For that, the state estimation error is defined as,

$$e(\iota) = \bar{x}_1(\iota) - \Phi(\iota) \tag{3.9}$$

therefore, the estimation error dynamic is

$$\begin{aligned} \rho e(\iota) &= \rho \bar{x}_1(\iota) - \rho \Phi(\iota) \\ &= \sum_{i=1}^{M} \mu_i(\zeta(\iota)) \left\{ \tilde{A}_{i_1} \bar{x}_1(\iota) - K_i \Phi(\iota) - L_i y(\iota) \right\} \\ &= \sum_{i=1}^{M} \mu_i(\zeta(\iota)) \left\{ K_i (\bar{x}_1(\iota) - \Phi(\iota)) \right\}. \end{aligned}$$

At last, in a resume manner, the necessary conditions for the observer existence are:

I. $\tilde{A}_{i_1}$ is asymptotically stable.
II. $(\tilde{A}_{i_1},\ \tilde{C}_{i_1})$ is observable.
III. C and F_i are full row and column rank, respectively.
IV. $\text{rank}(CF_i) = \text{rank}(F_i)$.

Continuous-time case

The estimation error dynamic for the continuous-time case results

$$\dot{e}(t) = \sum_{i=1}^{M} \mu_i(\zeta(t)) K_i e(t) \tag{3.10}$$

in this way, if K_i is quadratically Hurwitz, $\dot{e}(t) \to 0$ asymptotically.

Through the following theorem the sufficient LMI conditions are presented for the LPV-RUIO synthesis.

Theorem 3.1. *If there exists a symmetric matrix $X \succ 0$ and W_i, such that the following conditions hold $\forall i \in \mathbb{Z}_{1:M}$:*

$$(X\tilde{A}_{i_1} - W_i\tilde{C}_{i_1})^T + (X\tilde{A}_{i_1} - W_i\tilde{C}_{i_1}) + 2\alpha X \prec 0 \tag{3.11}$$

then the observer from Equation (3.8) *is an LPV-RUIO. That is, $e(t)$ towards zero asymptotically for any initial state $e(0)$.*

Proof. From Equation (3.9), and choosing a Lyapunov function with a symmetric matrix $X \succ 0$.

$$V(t) = e(t)^T X e(t) \tag{3.12}$$

thus, the exponential convergence of the estimation error is guaranteed if,

$$\dot{V}(t) + 2\alpha V(t) < 0 \tag{3.13}$$

where α is the decay rate constant [66]. Then, using Equation (3.10) and Equation (3.12) it derives in,

$$\begin{aligned} \dot{V}(t) &= \dot{e}^T(t) X e(t) + e(t) X \dot{e}^T(t) \\ &= \sum_{i=1}^{M} \mu_i(\zeta(t)) e^T(t) K_i^T X e(t) + \sum_{i=1}^{M} \mu_i(\zeta(t)) e^T(t) X K_i e(t) \\ &= \sum_{i=1}^{M} \mu_i(\zeta(t)) \left\{ e^T(t) \left(K_i^T X + X K_i \right) e(t) \right\} \end{aligned} \tag{3.14}$$

hence, replacing Equation (3.12) and Equation (3.14) in Equation (3.13) it results,

$$\sum_{i=1}^{M} \mu_i(\zeta(t)) \left\{ e^T(t) \left(K_i^T X + X K_i + 2\alpha X \right) e(t) \right\} < 0.$$

Indeed, notice that $(K_i^T X + X K_i + 2\alpha X) \prec 0, \ \forall i \in \mathbb{Z}_{1:M}$, implies that $e(t)$ towards zero asymptotically for any initial state $e(0)$.

Since $K_i = \tilde{A}_{i_1} - L_i \tilde{C}_{i_1}$, and replacing in the previous inequality,

$$\begin{gathered} (\tilde{A}_{i_1} - L_i\tilde{C}_{i_1})^T X + X(\tilde{A}_{i_1} - L_i\tilde{C}_{i_1}) + 2\alpha X \prec 0 \\ \tilde{A}_{i_1}^T X - \tilde{C}_{i_1}^T L_i^T X + X\tilde{A}_{i_1} - X L_i \tilde{C}_{i_1} + 2\alpha X \prec 0. \end{gathered}$$

Moreover, to eliminate the existing non-linearities, $W_i = X L_i$ is defined. Therefore,

$$(X\tilde{A}_{i_1} - W_i\tilde{C}_{i_1})^T + (X\tilde{A}_{i_1} - W_i\tilde{C}_{i_1}) + 2\alpha X \prec 0.$$

□

Remark 3.1. *The Theorem 3.1 shows that the LPV-RUIO design for continuous-time case is solved through the LMI from Equation* (3.11)*. For that, the LMI Lab [65] package of Matlab software is used.*

Thereby, from Equation (3.8), with $\Phi(t) \to \hat{\bar{x}}_1(t)$ pursuant to $t \to \infty$, getting

$$\begin{aligned} \hat{x}(t) &= \sum_{i=1}^{M} \mu_i(\zeta(t)) T_i \hat{\bar{x}}(t) \\ &= \sum_{i=1}^{M} \mu_i(\zeta(t)) T_i \begin{bmatrix} \Phi(t) \\ U_{i_1} y(t) - U_{i_1} C N_i \Phi(t) \end{bmatrix} \end{aligned} \tag{3.15}$$

with $\hat{x}(t) \to x(t)$ when $t \to \infty$.

Discrete-time case

The estimation error dynamic for the discrete-time case results

$$e(k+1) = \sum_{i=1}^{M} \mu_i(\zeta(k)) K_i e(k) \tag{3.16}$$

in this way, if K_i is quadratically Schur, $e(k+1) \to 0$ asymptotically.

Through the following theorem the sufficient LMI conditions are presented for the LPV-RUIO synthesis.

Theorem 3.2. *If there exists a symmetric matrix $X > 0$ and W_i, such that the following hold $\forall i \in \mathbb{Z}_{1:M}$:*

$$\begin{bmatrix} 2\alpha X & \tilde{A}_{i_1}^T X - \tilde{C}_{i_1}^T W_i^T \\ X\tilde{A}_{i_1} - W_i \tilde{C}_{i_1} & 2\alpha X \end{bmatrix} < 0 \tag{3.17}$$

then the observer from Equation (3.8) *is an LPV-RUIO. That is, $e(k)$ converges asymptotically to zero from any initial state $e(0)$.*

Proof. From Equation (3.9) and choosing a Lyapunov function with a symmetric matrix $X > 0$.

$$V(k) = e(k)^T X e(k) \tag{3.18}$$

thus, the exponential convergence of the estimation error is guaranteed if,

$$V(k+1) + 2\alpha V(k) < 0 \tag{3.19}$$

where α is the decay rate constant [66]. Then, using Equation (3.16) and Equation (3.18) it obtains,

$$\begin{aligned} V(k+1) &= e^T(k+1) X e(k+1) \\ &= \sum_{i=1}^{M} \mu_i(\zeta(k)) \left\{ e^T(k) \left(K_i^T X K_i \right) e(k) \right\} \end{aligned} \tag{3.20}$$

thus, replacing Equation (3.18) and Equation (3.20) in Equation (3.19) it results,

$$\sum_{i=1}^{M} \mu_i(\zeta(k)) \left\{ e^T(k) \left(K_i^T X K_i + 2\alpha X \right) e(k) \right\} < 0$$

note that $(K_i^T X K_i + 2\alpha X) < 0,\ \forall i \in \mathbb{Z}_{1:M}$, implies that $e(k)$ tends asymptotically to zero from any initial state $e(0)$.

Since $K_i = \tilde{A}_{i_1} - L_i \tilde{C}_{i_1}$, and replacing the previous inequality,

$$(\tilde{A}_{i_1} - L_i \tilde{C}_{i_1})^T X (\tilde{A}_{i_1} - L_i \tilde{C}_{i_1}) + 2\alpha X < 0.$$

Moreover, to eliminate the existing non-linearities, $W_i = X L_i$ is defined, and then applying the Schur complement results

$$\begin{bmatrix} 2\alpha X & \tilde{A}_{i_1}^T X - \tilde{C}_{i_1}^T W_i^T \\ X \tilde{A}_{i_1} - W_i \tilde{C}_{i_1} & 2\alpha X \end{bmatrix} < 0.$$

□

Remark 3.2. *The Theorem 3.2 shows that the LPV-RUIO design for discrete-time case is solved through the LMI from Equation* (3.17)*. For that, the LMI Lab [65] package of Matlab software is used.*

Thereby, from Equation (3.8) with $\Phi(k) \to \hat{\bar{x}}_1(k)$ pursuant to $k \to \infty$. That is,

$$\begin{aligned} \hat{x}(k) &= \sum_{i=1}^{M} \mu_i(\zeta(k)) T_i \hat{\bar{x}}(k) \\ &= \sum_{i=1}^{M} \mu_i(\zeta(k)) T_i \begin{bmatrix} \Phi(k) \\ U_{i_1} y(k) - U_{i_1} C N_i \Phi(k) \end{bmatrix} \end{aligned}$$

with $\hat{x}(k) \to x(k)$, with $k \to \infty$.

The following algorithm resumes the design procedure for the LPV-RUIO.

Algorithm 3: LPV-RUIO design:

```
Choose a scheduling parameter and compose the weighting functions,
Obtain the LPV model (Equation (3.1)), and discretise if necessary,
Check the existence conditions from 3.3.I to 3.3.IV,
Determine the $X$ and $W_i$ matrices using the Theorem 3.1 for
  continuous-time case, and the Theorem 3.2 for discrete-time case,
Compute the observer gain matrices, $L_i = X^{-1} W_i$, $L_i^* = L_i + E_{i_1}$ and
  $K_i = \tilde{A}_{i_1} - L_i \tilde{C}_{i_1}$.
```

3.4 Design of LPV-UIOO

Considering that the matrix H corresponds with a row of C, which is non-monitored output (faulty sensor), and choosing a transformation matrix T_2, such that $J = T_2 C$, where $J \in \mathbb{R}^{p-o}$ only contains the C rows corresponding with the monitored outputs (non-faulty sensors). Therefore, based on the development of the UIO for linear systems [87], the output equation of the system model from Equation (3.2) is transformed to,

$$\tilde{y}(\iota) = Jx(\iota).$$

Thereby, an UIO with the purpose of estimating the state variables $x(\iota)$, even with unknown input presence, is formulated as,

$$\begin{aligned} \rho z(\iota) &= \sum_{i=1}^{M} \mu_i(\zeta(\iota))\Big\{N_i z(\iota) + G_i u(\iota) + L_i \tilde{y}(\iota) + \Delta z_i\Big\} \\ \hat{x}(\iota) &= z(\iota) - E\tilde{y}(\iota) \\ \hat{y}(\iota) &= C\hat{x}(\iota) \end{aligned} \tag{3.21}$$

where $z(\iota) \in \mathbb{R}^n$ is the full-order observer state vector, and the matrices $N_i \in \mathbb{R}^{nxn}$, $G_i \in \mathbb{R}^{nxp}$, $L_i \in \mathbb{R}^{nxp}$ and $E \in \mathbb{R}^{nxp}$ are the observer gains to be determined. In addition, the weighting functions $\mu_i(.)$ are the same as used in Equation (3.1).

For that, defining the state estimation error

$$\begin{aligned} e(\iota) &= \hat{x}(\iota) - x(\iota) \\ &= z(\iota) - E\tilde{y}(\iota) - x(\iota) \\ &= z(\iota) - EJx(\iota) - x(\iota) \\ &= z(\iota) - (\mathbb{I} + EJ)x(\iota) \end{aligned} \tag{3.22}$$

being $T_1 = (\mathbb{I} + EJ)$, the estimation error results $e(\iota) = z(\iota) - T_1 x(\iota)$ and the estimation dynamic is

$$\begin{aligned} \rho e(\iota) &= \rho\hat{x}(\iota) - \rho x(\iota) \\ &= \sum_{i=1}^{M} \mu_i(\zeta(\iota))\Big\{N_i x(\iota) + G_i u(\iota) + L_i \tilde{y}(\iota) + \Delta z_i\Big\} - T_1 \rho x(\iota) \\ &= \sum_{i=1}^{M} \mu_i(\zeta(\iota))\Big\{N_i e(\iota) + (N_i T_1 + L_i J - T_1 A_i)x(\iota) + (G_i - T_1 B_i)u(\iota) \\ &\quad - T_1 F_i f_u(\iota) + \Delta z_i - T_1 \Delta x_i\Big\}. \end{aligned}$$

Accordingly,

$$\rho e(\iota) = \sum_{i=1}^{M} \mu_i(\zeta(\iota)) N_i e(\iota)$$

if the following constraints hold

$$T_1 F_i = 0 \tag{3.23a}$$

$$G_i - T_1 B_i = 0 \tag{3.23b}$$

$$N_i T_1 + L_i J - T_1 A_i = 0 \tag{3.23c}$$

$$\Delta z_i - T_1 \Delta x_i = 0. \tag{3.23d}$$

It is easy to see that Equation (3.23a) implies that $EJF_i = -F_i$, so to assure the existence of the matrix E it must be fulfilled that

$$rank(JF_i) = rank(F_i). \tag{3.24}$$

As a consequence the necessary existence conditions are:

I. A_i is asymptotically stable.
II. (A_i, J) is observable.
III. J and F_i are full row and column rank, respectively.
IV. $\text{rank}(JF_i) = \text{rank}(F_i)$.

It is important to note that, the observer from Equation (3.21) becomes in a traditional Luenberger observer if $T_1 = \mathbb{I}$ (that is $E = 0$), and without disturbance presence in the system ($f_u = 0$).

Continuous-time case

The estimation error dynamic for the continuous-time case results

$$\dot{e}(t) = \sum_{i=1}^{M} \mu_i(\zeta(t)) N_i e(t) \tag{3.25}$$

in this way, if N_i is quadratically Hurwitz, $\dot{e}(t) \to 0$ asymptotically.

Through the following theorem the sufficient LMI conditions are presented for the LPV-UIOO synthesis.

Theorem 3.3. *If there exist the matrices S, W_i, $\forall i \in \mathbb{Z}_{1:M}$ and a symmetric positive defined matrix $X > 0$ such that*

$$((X + SJ)A_i - W_iJ)_1^T + ((X + SJ)A_i - W_iJ) + 2\alpha X < 0 \tag{3.26}$$

subject to the decouple unknown input restriction from Equation (3.23a)

$$(X + SJ)F_i = 0 \tag{3.27}$$

then, the observer from Equation (3.21) *is an LPV-UIOO. That is, $e(t)$ converges asymptotically to zero from any initial state $e(0)$.*

Proof. From Equation (3.22), and choosing a Lyapunov function with a positive defined symmetric matrix $X = X^T > 0$.

$$V(t) = e(t)^T X e(t) \tag{3.28}$$

as a consequence, the exponential convergences of the estimation error is guaranteed if,

$$\dot{V}(t) + 2\alpha V(t) < 0 \tag{3.29}$$

where α is the decay rate constant [66]. Then, using Equation (3.25) and Equation (3.29) it obtains,

$$\begin{aligned}\dot{V}(t) =&\dot{e}^T(t)Xe(t) + e(t)X\dot{e}^T(t)\\ =&\sum_{i=1}^{M}\mu_i(\zeta(t))e^T(t)N_i^TXe(t) + \sum_{i=1}^{M}\mu_i(\zeta(t))e^T(t)XN_ie(t)\\ =&\sum_{i=1}^{M}\mu_i(\zeta(t))e^T(t)\Big(N_i^TX + XN_i\Big)e(t)\end{aligned} \tag{3.30}$$

thus, replacing Equation (3.30) and Equation (3.28) in Equation (3.29) it results,

$$\sum_{i=1}^{M}\mu_i(\zeta(t))\left\{e^T(t)\left(N_i^TX + XN_i + 2\alpha X\right)e(t)\right\} < 0.$$

Note that $(N_i^TX + XN_i + 2\alpha X) < 0,\ \forall i \in \mathbb{Z}_{1:M}$, implies that $e(t)$ tends asymptotically to zero for any initial state $e(0)$.

From Equation (3.23c) defining $K_i = N_iE + L_i$, it is obtained $N_i = T_1A_i - K_iJ$. Replacing in the previous inequality

$$\begin{aligned}(T_1A_i - K_iJ)^TX + X(T_1A_i - K_iJ) + 2\alpha X < 0\\ A_i^TX + A_i^TJ_1^TE^TX - J^TK_i^TX + XA_i + XEJA_i - XK_iJ + 2\alpha X < 0\end{aligned}$$

then, to eliminate the existing non-linearities, new variables are introduced $S = XE$, $W_i = XK_i$

$$((X + SJ)A_i - W_iJ)^T + (X + SJ)A_i - W_iJ) + 2\alpha X < 0$$

furthermore, based on Equation (3.23a) it results

$$(X + SJ)F_i = 0.$$

□

Additionally, the observer matrices result

$$\begin{aligned}N_i &= (\mathbb{I} + X^{-1}SJ)A_i - X^{-1}W_iJ\\ G_i &= (\mathbb{I} + X^{-1}SJ)B_i\\ L_i &= X^{-1}W_i - N_iE\\ \Delta z_i &= T_1\Delta x_i.\end{aligned} \tag{3.31}$$

Remark 3.3. *The Theorem 3.3 shows that the LPV-UIOO design for continuous-time case is solved through the LMI from Equation* (3.26) *subject to the restriction of Equation* (3.27). *For that, the Yalmip [88] package in Matlab software is used.*

Discrete-time case

The estimation error dynamic for the discrete-time case results

$$e(k+1) = \sum_{i=1}^{M} \mu_i(\zeta(k)) N_i e(k) \tag{3.32}$$

in this way, if N_i is quadratically Schur, $e(k+1) \to 0$ asymptotically.

Through the following theorem the sufficient LMI conditions are presented for the LPV-UIOO synthesis.

Theorem 3.4. *if there exist the matrices S, W_i, $\forall i \in \mathbb{Z}_{1:M}$ and a positive defined symmetric matrix $X > 0$ such that*

$$\begin{bmatrix} 2\alpha X & A_i^T X + A_i^T J^T S^T - J^T W_i^T \\ XA_i + SJA_i - W_i J & 2\alpha X \end{bmatrix} < 0 \tag{3.33}$$

subject to the decouple unknown input restriction of Equation (3.23a)

$$(X + SJ)F_i = 0 \tag{3.34}$$

then, the observer from Equation (3.21) *is an LPV-UIOO. That is, $e(k)$ converges asymptotically to zero from any initial state $e(0)$.*

Proof. From Equation (3.32), and choosing a Lyapunov function with a positive defined symmetric matrix $X > 0$

$$V(k) = e(k)^T X e(k) \tag{3.35}$$

as a consequence, the exponential convergence of the estimation error is guaranteed if,

$$V(k+1) + 2\alpha V(k) < 0 \tag{3.36}$$

where α is the decay rate constant [66]. Then, using Equation (3.32) and Equation (3.36) it obtains,

$$\begin{aligned} V(k+1) &= e^T(k+1) X e(k+1) \\ &= \sum_{i=1}^{M} \mu_i(\zeta(k)) e^T(k) \Big(N_i^T X N_i \Big) e(k) \end{aligned} \tag{3.37}$$

thus, replacing Equation (3.36) and Equation (3.35) in Equation (3.37) it results,

$$\sum_{i=1}^{M} \mu_i(\zeta(k)) \left\{ e^T(k) \left(N_i^T X N_i + 2\alpha X \right) e(k) \right\} < 0$$

Note that $(N_i^T X N_i + 2\alpha X) < 0$, $\forall i \in \mathbb{Z}_{1:M}$, implies that $e(k)$ tends asymptotically to zero for any initial state $e(0)$.

From Equation (3.23c) defining $K_i = N_iE + L_i$, it is obtained $N_i = T_1A_i - K_iJ$. Replacing that in the previous inequality

$$(T_1A_i - K_iJ)^T X(TA_i - K_iJ) + 2\alpha X < 0$$

since $T_1 = (\mathbb{I} + EJ)$,

$$(A_i + EJA_i - K_iJ)^T X(A_i + EJA_i - K_iJ) + 2\alpha X < 0$$

then, to eliminate the existing non-linearities, new variables are introduced $S = XE$, $W_i = XK_i$, and using the Schur complement

$$\begin{bmatrix} 2\alpha X & A_i^T X + A_i^T J^T S^T - J^T W_i^T \\ XA_i + SJA_i - W_iJ & 2\alpha X \end{bmatrix} < 0$$

furthermore, based on Equation (3.23a) it results,

$$(X + SJ)F_i = 0.$$

□

Additionally, the observer gain matrices are:

$$\begin{aligned} N_i &= (\mathbb{I} + X^{-1}SJ)A_i - X^{-1}W_iJ \\ G_i &= (\mathbb{I} + X^{-1}SJ)B_i \\ L_i &= X^{-1}W_i - N_iE \\ \Delta z_i &= T\Delta x_i. \end{aligned} \tag{3.38}$$

Remark 3.4. *The Theorem 3.4 shows that the LPV-UIOO design for discrete-time case is solved through the LMI from Equation* (3.33) *subject to the restriction from Equation* (3.34). *For that, the Yalmip [88] package in Matlab software is used.*

The following algorithm resumes the design procedure for the LPV-UIOO.

Algorithm 4: LPV-UIOO design:

```
1 Choose a scheduling parameter and compose the weighting functions,
2 Obtain the LPV model (Equation (3.1)), and discretise if necessary,
3 Check the existence conditions from 3.4.I to 3.4.IV,
4 Determinate the $X$, $S$ and $W_i$ matrices using the Theorem 3.3 for
    continuous-time case, and the Theorem 3.4 for discrete-time case,
5 Compute the observer gain matrices $E = X^{-1}S$, $K_i = X^{-1}W_i$,
    $T_1 = (\mathbb{I} + EJ)$, $N_i = T_1A_i - K_iJ$, $G_i = T_1B_i$, $L_i = K_i - N_iE$, $\Delta z_i = T_1\Delta x_i$.
```

3.5 Fault detection and diagnosis

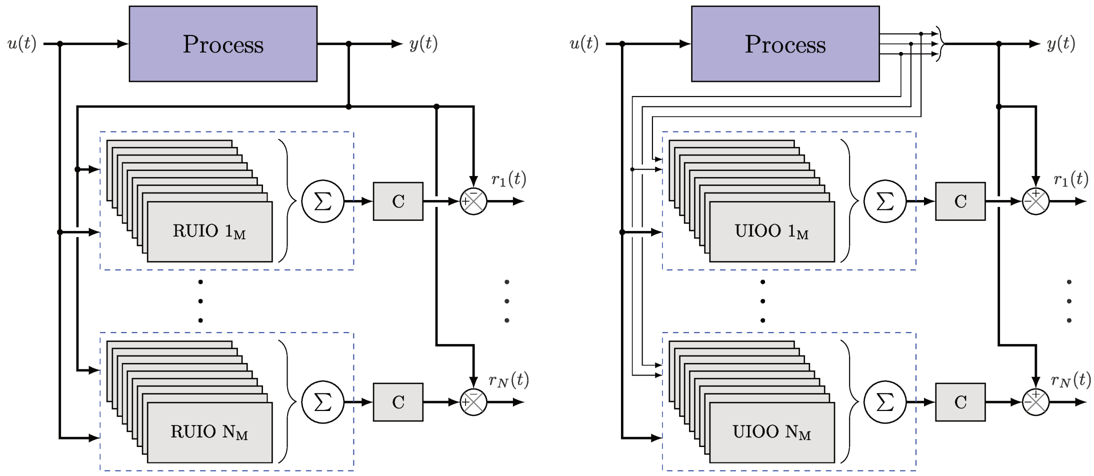
As commonly used in literature [31, 18, 11], the residual pattern should be customised to follow a certain structure (*generalised residuals*), enabling an extra degree of freedom in the observer design. Therefore, the structured residuals are characterised by selective fault responses. That is, any residual is design to be sensitive to one group of faults and insensitive to others.

Remark 3.5. *The proposed FDD scheme was designed to isolate a single fault in either a sensor or an actuator, at the time. This is based on the fact that the probability for two or more faults to occur at the same time is very small in a real situation, and in many cases it may be assumed that any fault gets repaired before another one appears. Besides, if simultaneous faults need to be isolated, the fault isolation scheme should be modified based on a regrouping of faults. That is, each residual will be designed to be sensitive to one group of faults and insensitive to others.*

In this way, the following sections explain the strategy adopted to develop the fault detection, isolation and estimation tasks, based on the use of observers banks.

3.5.1 Observer-based residue generators

Accordingly, the Figure 3.1 depicts the proposed observers banks to construct a set of residuals, that enables the FDD task of multiple actuators and sensors. In detail, the Figure 3.1a, presents the actuator fault residue generator, based on a bank of LPV-RUIOs which use all input-output data. On the other hand, the Figure 3.1b introduces the sensor fault residue generator approach, through a bank of LPV-UIOOs which use all input and all but one output signals.



(a) LPV-RUIO bank

(b) LPV-UIOO bank

Figure 3.1: Observers banks.

LPV-RUIO bank

As was briefly explained previously, the Figure 3.1a shows the actuator residue generator structure, where each observer is sensitive to all but one fault (unknown input). Thus, if the LPV-RUIO, from Equation (3.8), can be designed, and defining $|r(\iota)| = |\hat{y}(\iota) - y(\iota)| = |Ce(\iota)|$, then $|r(\iota)|$ tends to zero, when $\iota \to \infty$, and the system is fault free, or in presence of an unknown input (insensitive actuator fault). Based on this observation, each residual is compared against a threshold, and depending on that, it reveals if the fault occurs, or not:

$$\text{Fault} = \begin{cases} \text{False} & \text{if } |r(\iota)| \leq r_{th}(\iota) \\ \text{True} & \text{if } |r(\iota)| > r_{th}(\iota) \end{cases}. \tag{3.39}$$

In particular, with the aim of reduce the false positives detection, during the initial convergence, an exponential threshold is introduced. Given by,

$$r_{th}(\iota) = r_{th_i} e^{-\frac{\iota}{\tau}} + r_{th}; \tag{3.40}$$

where r_{th_i}, τ and r_{th} are the threshold initial value, the convergence period and the nominal threshold value (obtained based on modelling uncertainties, due to the LPV representation), respectively.

LPV-UIOO bank

In the same way, the Figure 3.1b shows the sensor residue generator structure. In this case, the scheme uses an observers bank that is driven by all inputs and all but one outputs of the system, so each output residual is

$$\begin{aligned} r(\iota) &= \hat{y}(\iota) - y(\iota) \\ &= Cz(\iota) - CE\tilde{y}(\iota) - Cx(\iota) - Hf_y(\iota) \\ &= Ce(\iota) + CT_1x(\iota) - CEJx(\iota) - Cx(\iota) - Hf_y(\iota) \\ &= Ce(\iota) - Hf_y(\iota) \end{aligned}$$

in such a way, if the system is without faults, each residual vector $r(\iota)$ tends to zero asymptotically for any initial state $r(0)$.

Remark 3.6. *It is important to note that the residual vector is composed from monitored and non-monitored components, which are corresponding with the monitored and non-monitored outputs.*

Therefore, only the non-monitored components of the residual vector are affected by a fault $f_y(\iota)$, and all the others should be zero. That is, when a monitored output fault income ($Hf_y = 0$) the residual vector differed from zero in their monitored components

$$r(\iota) = Ce(\iota).$$

Conversely, if a non-monitored output fault income ($Ce(\iota) = 0$) the residual differed from zero in their non-monitored components

$$r(\iota) = H_i f_y(\iota). \tag{3.41}$$

This means, using the same method as introduced in Equation (3.39), it is setting an exponential threshold for each monitored residue components, and thereby depending on the result of its comparison, it reveals if the fault occurs, or not.

3.5.2 Fault detection and isolation

Bearing this in mind, the threshold test may be performed separately for each residual, yielding to a decision incidence matrix, and then the isolation task is fulfilled using this matrix. As an example, the fault isolation based on an incidence matrix and four residual components is presented,

$$\begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix}$$

being the binary variables True = 1 and False = 0.

3.5.3 Fault estimation

When the fault detection and isolation task are successfully achieved, the actuator or sensor fault estimation is computed.

Actuators

Hence, the actuator estimation task is performed using the corresponding observer to the associated fault (selected as unknown input signal).

Continuous-time case For that, from Equation (3.3a), and taking into account that $\hat{\bar{x}}(t) \to \bar{x}(t)$ pursuant to $t \to \infty$, it is possible to estimate the unknown input

$$\hat{f}_u(t) = \sum_{i=1}^{M} \mu_i(\zeta(t)) \Big\{ U_{i_1}\dot{y}(t) - \bar{A}_{i_{22}}U_{i_1}y(t) - G_1\dot{\Phi}(t) - G_2\Phi(t) - \bar{B}_{i_2}u(t) + \bar{\Delta}x_{i_2} \Big\}$$

where $G_1 = U_{i_1}\tilde{C}_{i_1}$ and $G_2 = \bar{A}_{i_{21}} - \bar{A}_{i_{22}}U_{i_1}\tilde{C}_{i_1}$.

Discrete-time case In the same way,

$$\begin{aligned} \hat{f}_u(k-1) = \sum_{i=1}^{M} \mu_i(\zeta(k)) \Big\{ & U_{i_1}y(k) - G_1y(k-1) - G_2\Phi(k) - G_3\Phi(k-1) \\ & - \bar{B}_{i_2}u(k-1) + \bar{\Delta}x_{i_2} \Big\} \end{aligned}$$

where $G_1 = \bar{A}_{i_{22}}U_{i_1}$, $G_2 = U_{i_1}\tilde{C}_{i_1}$ and $G_3 = \bar{A}_{i_{21}} - \bar{A}_{i_{22}}U_{i_1}\tilde{C}_{i_1}$.

Sensors

Besides, the sensor estimation task is performed using the k-th non-monitored residue component from the corresponding observer to the associated fault (selected as non-monitored signal). From Equation (3.41), and taking into account the Assumption 2, that is H_i is a unitary gain selector vector. So, $f_y(\iota)$ results:

$$f_y(\iota) = r_k(\iota).$$

3.6 Illustrative example I (HE)

An HE is an industrial device designed to efficiently transfer or "exchange" heat from one matter to another, for cooling or heating. There exist in many different forms or types.

In particular, consider the HE presented in Figure 3.2.

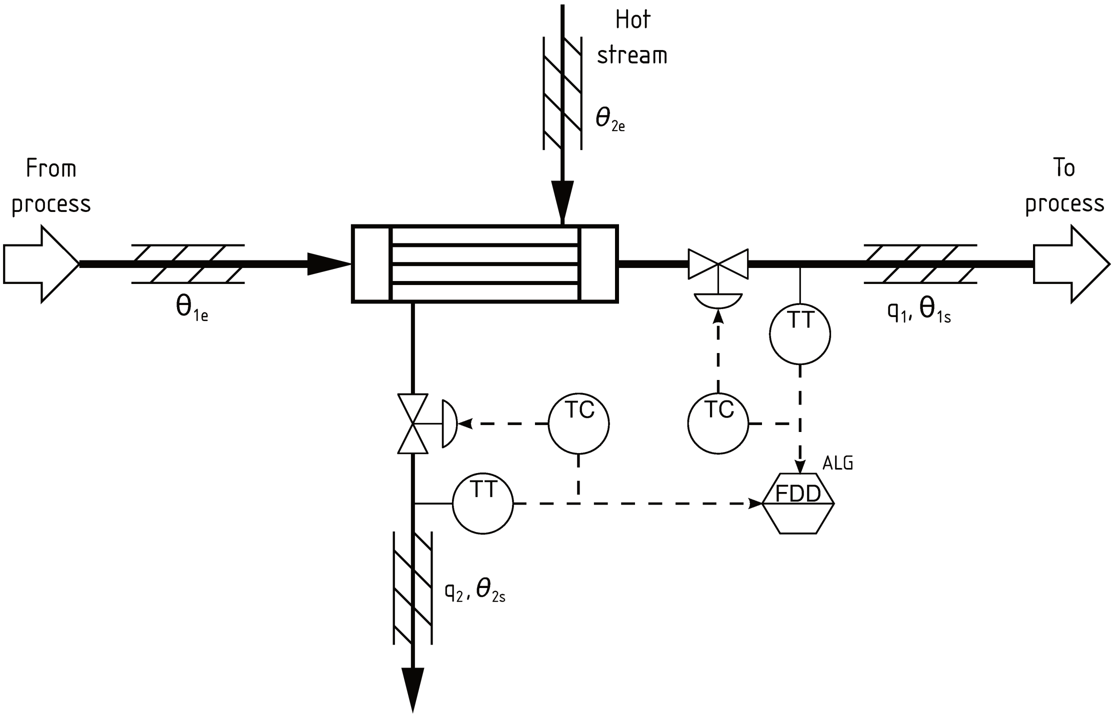


Figure 3.2: Diagram of a Heat Exchanger process.

To modelling its dynamics, the following hypotheses and conditions have been established:

- It is considered a system of concentrated parameters.
- Based on the total number of tubes, it is assumed that their mass can accumulate heat.
- Heat losses are negligible.
- The physical properties are considered constant and evaluated at the mean temperature between the inlet and outlet of each stream.
- Incompressible fluid, consequently there is no accumulation of matter in tubes and there is no phase change in fluids.
- Without loss of generality and in order to establish a direction of caloric flow, it is assumed that $\theta_{2s} > \theta_p > \theta_{1s}$.
- The temperature profile on the wall of the tubes is negligible. That is, it is assumed that the wall of the tubes has a very large heat transfer coefficient (λ)
- A single average caloric exchange area is considered ($A_1 = A_2 = A$).
- There is no chemical reaction ($r = 0$).
- The potential energy changes are negligible.

Based on these considerations, Table 3.1 shows its physical and operational parameters.

Table 3.1: Parameters of the HE process.

Name	**Description**	**Value**
ρ_1	Fluid 1 density	$1\,\mathrm{kg\,l^{-1}}$
ρ_2	Fluid 2 density	$1\,\mathrm{kg\,l^{-1}}$
ρ_p	Wall density	$7.874\,\mathrm{kg\,l^{-1}}$
C_{p_1}	Heat capacity 1	$1000\,\mathrm{cal\,kg^{-1}\,K}$
C_{p_2}	Heat capacity 2	$1000\,\mathrm{cal\,kg^{-1}\,K}$
C_{p_p}	Wall specific heat	$1075.53\,\mathrm{cal\,kg^{-1}\,K}$
A	Heat exchange area	$0.881\,\mathrm{m^2}$
h_1	Heat transfer 1	$32\,374\,\mathrm{cal\,min^{-1}\,K^{-1}\,m^{-2}}$
h_2	Heat transfer 2	$14\,716.67\,\mathrm{cal\,min^{-1}\,K^{-1}\,m^{-2}}$
V_1	Tube Volume	$16\,\mathrm{l}$
V_2	Case Volume	$2.11\,\mathrm{l}$
V_p	Wall Volume	$1.19\,\mathrm{l}$
θ_{1_e}	Input 1 Temp.	$450\,\mathrm{K}$
θ_{2_e}	Input 2 Temp.	$900\,\mathrm{K}$

Taking all the above into account, their non-linear model equations are [59, pp. 67-72]:

$$\begin{aligned} V_1\frac{d\theta_{1_s}(t)}{dt} &= q_1(t)(\theta_{1_e} - \theta_{1_s}(t)) - \frac{Ah_1}{\rho_1 C_{p_1}}(\theta_{1_s}(t) - \theta_p(t)) \\ V_2\frac{d\theta_{2_s}(t)}{dt} &= q_2(t)(\theta_{2_e} - \theta_{2_s}(t)) + \frac{Ah_2}{\rho_2 C_{p_2}}(\theta_p(t) - \theta_{2_s}(t)) \\ V_p\frac{d\theta_p(t)}{dt} &= \frac{Ah_1}{\rho_p C_{p_p}}(\theta_{1_s}(t) - \theta_p(t)) - \frac{Ah_2}{\rho_p C_{p_p}}(\theta_p(t) - \theta_{2_s}(t)) \end{aligned} \tag{3.42}$$

It should be note that, according to Figure 3.2, the state variable θ_{1_s} is controlled by the cold process flow rate q_1 and the hot stream flow rate q_2.

Furthermore, applying the PJL technique (see Appendix B) it is possible to rewrite the non-linear system from Equation (3.42) as an LTI model around the i-th operating point $x_i = \{\theta_{1s_i},\ \theta_{2s_i},\ \theta_{p_i}\}$,

$$A_i = \begin{bmatrix} -\frac{Ah_1+q_1\rho_1 C_{p_1}}{V_1\rho_1 C_{p_1}} & 0 & \frac{Ah_1}{V_1\rho_1 C_{p_1}} \\ 0 & -\frac{Ah_2+q_2\rho_2 C_{p_2}}{V_2\rho_2 C_{p_2}} & \frac{Ah_2}{V_2\rho_2 C_{p_2}} \\ \frac{Ah_1}{V_p\rho_p C_{p_p}} & \frac{Ah_2}{V_p\rho_p C_{p_p}} & -\frac{Ah_1+Ah_2}{V_p\rho_p C_{p_p}} \end{bmatrix},$$

$$B_i = \begin{bmatrix} \frac{\theta_{1_e}-\theta_{1_s}}{V_1} & 0 \\ 0 & \frac{\theta_{2_e}-\theta_{2_s}}{V_2} \\ 0 & 0 \end{bmatrix},\ C_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ D_i = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Finally, being the non-linear process model from Equation (3.42), it was develop an FDD scheme with the aim to detect, isolate y diagnose faults in multiple actuators and sensors. Therefore, based on the novelty FDD scheme, introduced through the Sections 3.2 to 3.5, the suitable matrices and involved functions were designed.

LPV-RUIO

According to Algorithm 3 the scheduling parameter vector, with dimension $N = 2$, was defined depending on the outputs, $\zeta(t) \coloneqq \begin{bmatrix} \theta_{1_s}(t) & \theta_{2_s}(t) \end{bmatrix}$. Moreover, to obtain a proper representation, $L = 3$ linearisation points per parameter was used. Then, based on Appendix B the non-linear system from Equation (3.42) was rewritten such as an LPV model. As a result, $M = L^N = 9$ weighting functions corresponding to each linear model was setted. Next, in the same way as Appendix B.3, the proper membership functions of each parameter were obtained.

At last, defining $F = \begin{bmatrix} \frac{\theta_{1_e} - \theta_{1_{s_i}}}{V_1} & 0 & 0 \end{bmatrix}^T$, it was possible to construct the matrices of the observer from Equation (3.8) that was used to estimate the faults of the valve q_1. On the other hand, defining $F = \begin{bmatrix} 0 & \frac{\theta_{2_e} - \theta_{2_{s_i}}}{V_2} & 0 \end{bmatrix}^T$, and repeating the previous procedure, the observer was constructed to estimate the faults of the valve q_2.

LPV-UIOO

Similarly, as was defined in the Algorithm 4, and furthermore using the same LPV model representation of the non-linear system from Equation (3.42), developed previously, the LPV-UIOO design is presented.

For that, firstly the output θ_{1_s} was selected as a non-monitored output (H_i corresponds with the θ_{1_s} row of the output matrix C_i), and $F = \begin{bmatrix} \frac{\theta_{1_e} - \theta_{1_{s_i}}}{V_1} & 0 & 0 \end{bmatrix}^T$ was defined, then the appropriate T_2 matrix was chosen (null space of H_i). Thus, if the existence conditions were verified, it is possible to construct the matrices of the observer from Equation (3.38), that was used to diagnose faults of the output sensor θ_{1_s}. On the other hand, choosing the output θ_{2_s} as non-monitored output, and defining $F = \begin{bmatrix} 0 & \frac{\theta_{2_e} - \theta_{2_{s_i}}}{V_2} & 0 \end{bmatrix}^T$, if the existence conditions were verified, and selecting another T_2 matrix, a new observer from Equation (3.38) was constructed to diagnose faults of the output sensor θ_{2_s}.

Numerical simulation

To evaluate the performance and effectiveness of the proposed FDD scheme, it was simulated on the non-linear system from Equation (3.42), during 8 h and using a sampling period $T_s = 3\,\mathrm{s}$. In particular, to show its dynamism a set of set-points changes were introduced.

1. A shift from the initial condition $\theta_{1_s}^0 = 495\,\mathrm{K}$, $\theta_{2_s}^0 = 695.915\,\mathrm{K}$ to $\theta_{1_s} = 497.32\,\mathrm{K}$ was applied between $t = 160\,\mathrm{min}$ and $t = 200\,\mathrm{min}$.
2. From $t = 340\,\mathrm{min}$ to $t = 380\,\mathrm{min}$ another change up to $\theta_{1_s} = 498\,\mathrm{K}$, was introduced.
3. Finally the output θ_{2_s} was carried up to 680 K from $t = 520\,\mathrm{min}$ to $t = 560\,\mathrm{min}$.

And the following faults were considered:

1. from 30 min to 130 min an abrupt leak in valve q_1 coupling, producing a loss of 5% from $q_{1_{\max}}$,
2. an exponential clog in valve q_2 of up to 5% from $q_{2_{\max}}$ income between 580 min and 680 min,
3. at 400 min to 500 min an exponential calibration error in sensor θ_{1_s} of up to 0.5% from $\theta_{1_{s\max}}$,
4. since 220 min up to 320 min an exponential accumulation of tartar in sensor θ_{2_s}, inducing an error of up to 0.5% from $\theta_{2_{s\max}}$.

It is important to note that, the high slopes in these faults were chosen to illustrate worst-case conditions and expose the capacity of the proposed FDD scheme to detect, isolate and diagnose such cases. Furthermore, to build a more realistic simulation, a white measurement noise, of considerable magnitude, was added.

Figures 3.3 to 3.7 show the simulation results when the actuators and sensors faults occur. Indeed, the estimated output signals of the non-linear system from Equation (3.42), and their output measures were depicted in Figure 3.3. There, it is possible to appreciate the convergence capacity, at the beginning. Moreover, it is easy to see how, when a sensor fault affects the system, the control loop is unable to reach the target set-point (due to the erroneous measurement of process variables). In contrast, when an actuator becomes faulty the control loop (*Proportional-Integral* (PI) controller) automatically compensates the tracking error, and as shown in Figure 3.3 small peaks at 30 min and 130 min are appreciated.

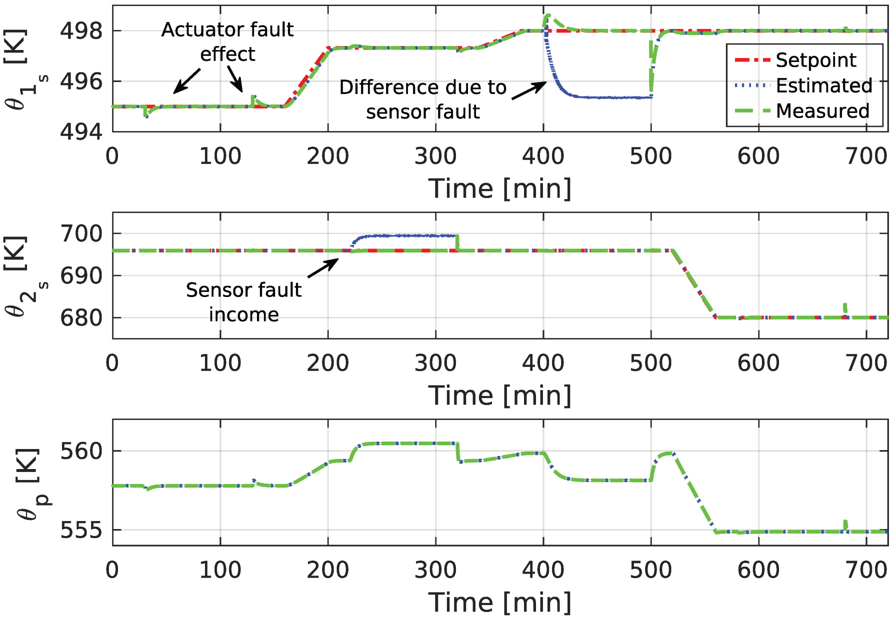


Figure 3.3: HE temperatures.

In addition, the Figure 3.4 exhibits the time behaviour of the weighting functions of both observers types. Here, it can be appreciated that the weight varies according to the set-points changes, composing a linear model that approximates the non-linear behaviour around its operation point.

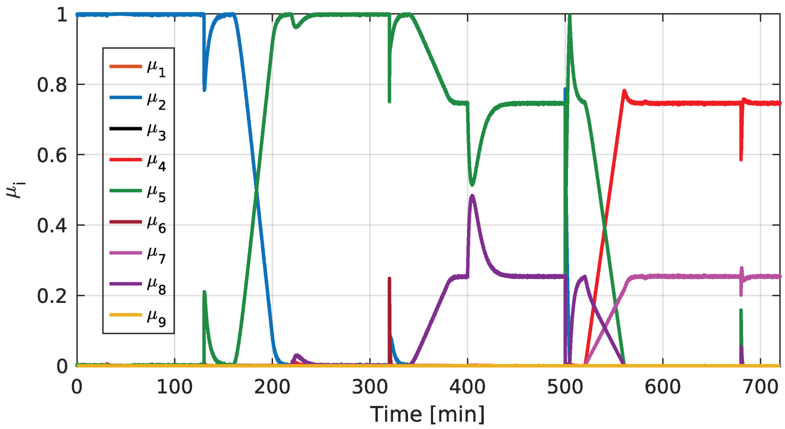


Figure 3.4: HE weighting functions.

Moreover, the Figure 3.5 plots the inputs with the actuators fault occurrence. Here, the controller overused the control inputs to compensate the actuators faults (this is only possible when the system has no restrictions)

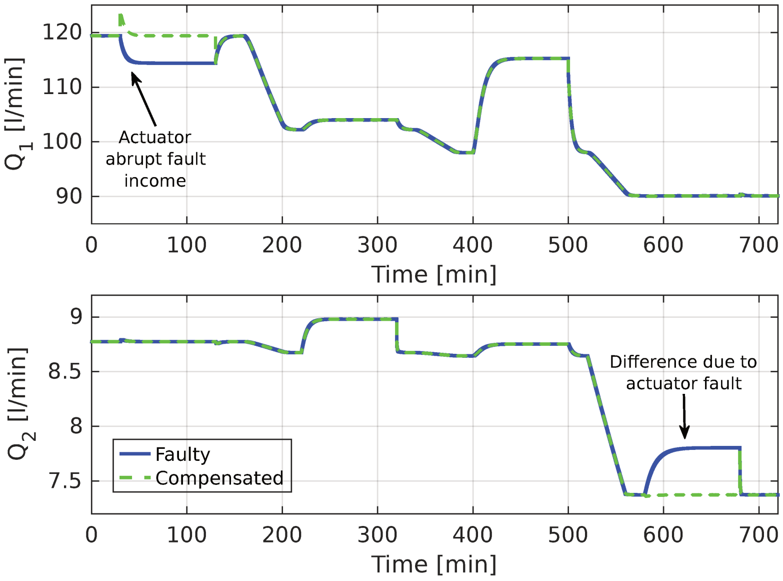
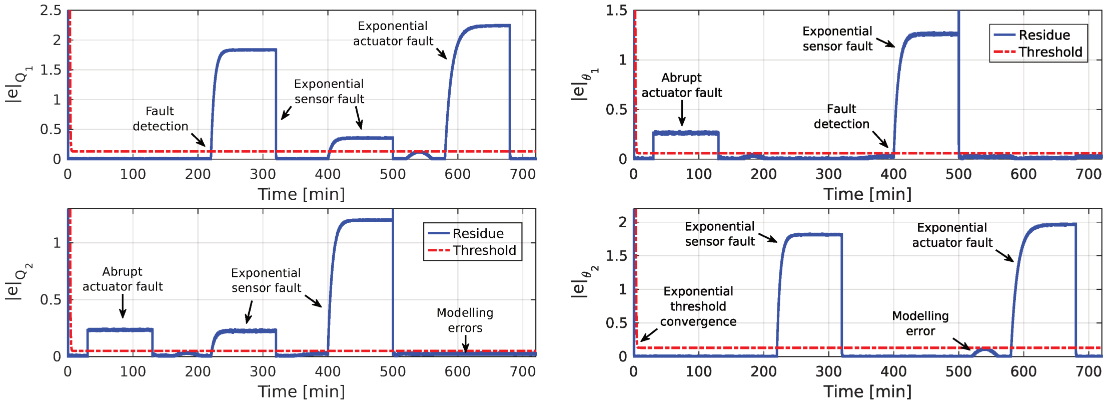


Figure 3.5: HE control inputs.

On the other hand, the Figure 3.6 presents the observers residues. Specifically, the Figure 3.6a shows the LPV-RUIOs errors, and the Figure 3.6b plots the LPV-UIOOs errors. In this, it is possible to see the exponential threshold convergence, the fault detection instant and the errors due to the model mismatch. In particular, bearing in mind the design of the residual structure, presented in Section 3.5, the observer errors are sensitive to a set of faults and insensitive to others. For example at $t = 300\,\text{min}$ all but one residues are activated, indicating that the hot stream temperature sensor (θ_{2_s}) is failing. Lastly, it is important to highlight that this scheme exhibits a short detection time and, as expected, it depends on the temporal shape of the faults (i.e. exponential or abrupt).

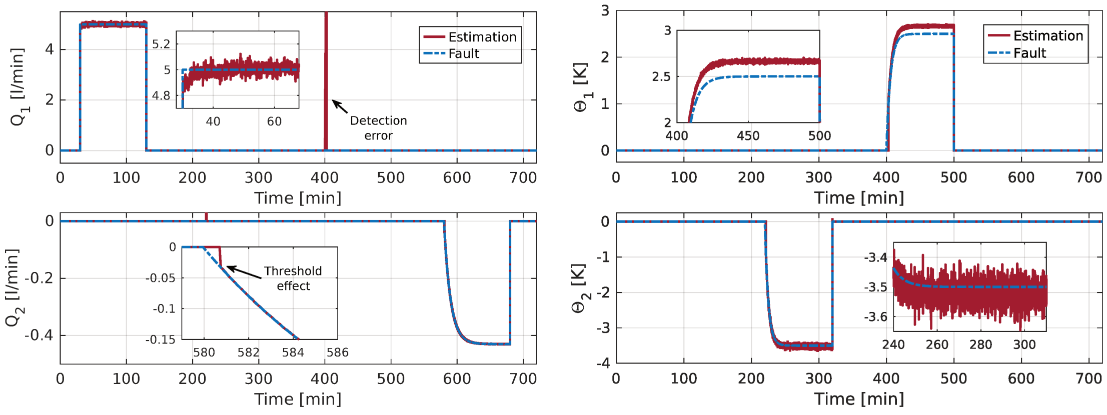


(a) HE LPV-RUIOs errors.

(b) HE LPV-UIOOs errors.

Figure 3.6: HE observers errors.

Besides, the Figure 3.7 shows the actuators (Figure 3.7a) and sensors (Figure 3.7b) faults estimation. There, the effects introduced due to the use of a detection strategy based on threshold, is appreciated. As well as, detection and estimation errors, all of these of small magnitude.



(a) HE actuator fault estimation.

(b) HE sensor fault estimation.

Figure 3.7: HE faults estimation.

At last, to verify that the computation effort required by the proposed strategy is not a problem to carry out on-line as a part of an AFTCS scheme, the overhead time percentage with respect of T_s was measured[1]. Being 0.108 %, 0.126 % and 3.151 % the minimum, the average and the maximum values, respectively. As can be appreciated there, it is not a problem to carry out these FDD methods on-line as a part of an AFTCS scheme.

[1] Using an i5-3337U CPU@2.7 GHz (2 Cores) with 6 GB of RAM.

3.7 Illustrative example II (CSTR)

A CSTR is a continuous-flow reactor equipped with an impeller or other mixing device to provide efficient mixing. They are the most basic reactors used in chemical processes.

Specifically, consider the CSTR process depicted in Figure 3.8. This model is a modified version of the CSTR example presented by Morningred et al. in [89]. In the original model, the system operates to constant volume.

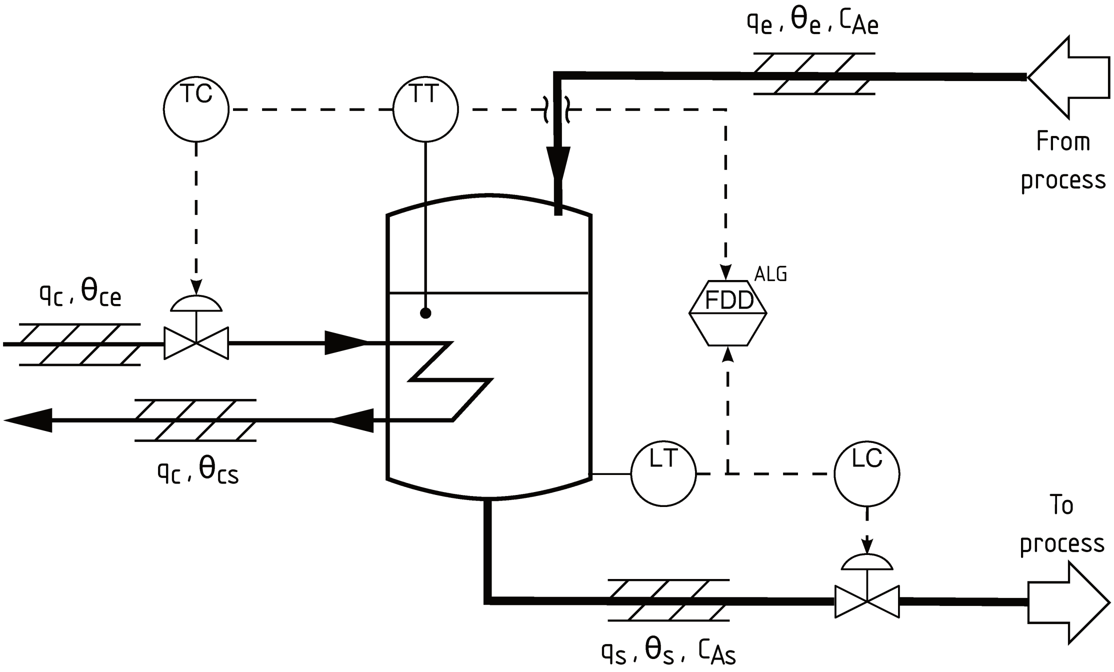


Figure 3.8: Diagram of a Continuous Stirred Tank Reactor process.

Meanwhile, to modelling its dynamics, the following hypotheses and conditions have been established:

- The physical properties are constant and independent of temperature and concentration of reactants and products.
- Time delays are not modelled.
- Reagents and products are in the liquid phase.
- The chemical reaction is not isothermal.
- Perfect mixing can be assumed in the reactor.
- The heat transfer from the reaction tank to the jacket is ideal, indicating that the energetic effects occurring between the tank wall and the jacket are assumed to be negligible.
- The resistance to the liquid at the outlet of the tank is not negligible, therefore, the volume of the liquid in the reactor cannot be considered constant in the face of changes in the inlet flow (q_e).
- The cross section (A) of the reaction tank is constant and therefore it can be assumed that $V = Ah$.

Based on these considerations, the Table 3.2 shows its physical and operational parameters. Therefore, the CSTR process consists of an irreversible, exothermic reaction, $A \rightarrow B$, in a variable volume reactor cooled by a single coolant stream which can be modelled by the following equations:

Table 3.2: Parameters of the CSTR process.

Name	Description	Value
q_e	Feed flow rate	$100\,\mathrm{l\,min^{-1}}$
T_e	Feed temperature	$350\,\mathrm{K}$
C_{Ae}	Feed concentration	$1\,\mathrm{mol\,l^{-1}}$
T_{ce}	Inlet coolant temp.	$350\,\mathrm{K}$
E/R	Activation energy	$1 \times 10^4\,\mathrm{K}$
$\Delta\mathrm{H}$	Heat of reaction	$-2 \times 10^5\,\mathrm{cal\,mol^{-1}}$
C_p, C_{pc}	Specific heats	$1\,\mathrm{cal\,g^{-1}\,K}$
ρ, ρ_c	Liquid densities	$1 \times 10^3\,\mathrm{g\,l^{-1}}$
h_A	Heat transfer term	$7 \times 10^5\,\mathrm{cal\,min^{-1}\,K}$
k_0	Reaction rate constant	$7.2 \times 10^{10}\,\mathrm{l\,min^{-1}}$
$q_{s_{\max}}$	Max. output flow rate	$110\,\mathrm{l\,min^{-1}}$
$q_{c_{\max}}$	Max. coolant flow rate	$110\,\mathrm{l\,min^{-1}}$

$$
\begin{aligned}
\frac{dV(t)}{dt} &= q_e - q_s(t) \\
\frac{dC_A(t)}{dt} &= \frac{q_e}{V(t)}(C_{Ae} - C_A(t)) - k_0 e^{\frac{-E}{RT(t)}} C_A(t) \\
\frac{dT(t)}{dt} &= \frac{q_e}{V(t)}(T_e - T(t)) - k_1 e^{\frac{-E}{RT(t)}} C_A(t) \\
&\quad + \frac{q_c(t)}{V(t)} k_2 (1 - e^{\frac{-k_3}{q_c(t)}})(T_{ce} - T(t))
\end{aligned}
\tag{3.43}
$$

where

$$
k_1 = \frac{\Delta H k_0}{\rho C_p}, \quad k_2 = \frac{\rho C_{pc}}{\rho_c C_p}, \quad k_3 = \frac{h_A}{\rho_c C_{pc}}.
$$

Notice that, according to Figure 3.8, the state variables V and C_A are controlled by the process flow rate q_s and the coolant flow rate q_c, respectively. It should be clarified that the state variable C_A is controlled indirectly from the state T.

Furthermore, applying the PJL technique (see Appendix B), it is possible to rewrite the system from Equation (3.43) as an LTI model around the i-th operating point $x_i = \{V_i,\ C_{A_i},\ T_i\}$,

$$A_i = \begin{bmatrix} 0 & 0 & 0 \\ \frac{q_e(C_{A_i}-C_{Ae})}{V_i^2} & -\frac{q_e}{V_i} - k_0 e^{\frac{-E}{RT_i}} & -\frac{EC_{A_i}k_0 e^{\frac{-E}{RT_i}}}{RT_i^2} \\ A_{31} & -k_1 e^{\frac{-E}{RT_i}} & A_{33} \end{bmatrix},$$

$$B_i = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & B_{32} \end{bmatrix},\ C_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ D_i = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

where

$$A_{31} = \frac{q_e(T_i - T_e)}{V_i^2} - \frac{k_2 q_{c_i}(e^{\frac{-k_3}{q_{c_i}}} - 1)(T_i - T_{ce})}{V_i^2}$$

$$A_{33} = \frac{k_2 q_{c_i}(e^{\frac{-k_3}{q_{c_i}}} - 1)}{V_i} - \frac{q_e}{V_i} - \frac{EC_{A_i}k_1 e^{\frac{-E}{RT_i}}}{RT_i^2}$$

$$B_{32} = \frac{k_2(T_i - T_{ce})\left(e^{\frac{-k_3}{q_{c_i}}} - 1\right)}{V_i} + \frac{k_2 k_3 (T_i - T_{ce}) e^{\frac{-k_3}{q_{c_i}}}}{V_i}.$$

Finally, following the design procedure used in the previous example, previously introduced through Sections 3.2 to 3.5, the suitable matrices and involved functions were designed.

LPV-RUIO design

According to Algorithm 3 the scheduling parameter vector, with dimension $N = 2$, was defined depending on the outputs, $\zeta(t) \coloneqq \begin{bmatrix} V(t) & T(t) \end{bmatrix}$. Moreover, to obtain a proper representation, $L = 3$ linearisation points per parameter was used. Then, based on Appendix B the non-linear system from Equation (3.43) was rewritten such as an LPV model. As a result, $M = L^N = 9$ weighting functions corresponding to each linear model was setted. Next, in the same way as Appendix B.3, the proper membership functions of each parameter were obtained.

Moreover, setting $F_i = \begin{bmatrix} -1 & 0 & 0 \end{bmatrix}^T$ and solving the LMI from Equation (3.11), the matrices to construct the observer from Equation (3.8) that are used to estimate faults over the valve q_s, were designed. At last, defining $F_i = \begin{bmatrix} 0 & 0 & B_{32} \end{bmatrix}^T$, and repeating the previous procedure, the observer was constructed to estimate faults on the valve q_c.

LPV-UIOO

Similarly, as was defined in the Algorithm 4, and furthermore using the same LPV model representation of the non-linear system from Equation (3.43), developed previously, the LPV-UIOO design is presented.

It is noteworthy, that the existing hard correlation over the states C_A and T hinder the correct fault detection and isolation task design. In consequence, firstly the output V was selected as a non-monitored output (H_i corresponds with the V row of the output matrix C_i), and $F_i = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & B_{32} \end{bmatrix}^T$ was defined, then the appropriate T_2 matrix was chosen (null space of H_i). Thus, if the existence conditions are verified, it is possible to construct the matrices of the observer from Equation (3.38) that is used to diagnose faults of the output sensor V. On the other hand, choosing the output T as non-monitored output, and defining $F_i = \begin{bmatrix} -1 & 0 & 0 \end{bmatrix}^T$, if the existence conditions are verified, and selecting another T_2 matrix, a new observer from Equation (3.38) was constructed to diagnose faults of the output sensor T.

Numerical Simulation

Again, to evaluate the performance and effectiveness of the proposed FDD scheme, it was simulated on the non-linear system from Equation (3.43), using a sampling period $T_s = 3\,\mathrm{s}$. In particular, to show its dynamism, a set of set-points changes were introduced.

1. A shift from the initial condition $V^0 = 96\,\mathrm{m}^3$, $T^0 = 440\,\mathrm{K}$ to $T = 445\,\mathrm{K}$ was applied between $t = 160\,\mathrm{min}$ and $t = 200\,\mathrm{min}$.
2. From $t = 340\,\mathrm{min}$ to $t = 380\,\mathrm{min}$ a displacement up to $V = 110\,\mathrm{m}^3$, was introduced.
3. Finally the output T was carried up to $450\,\mathrm{K}$ from $t = 520\,\mathrm{min}$ to $t = 560\,\mathrm{min}$.

And the following exponential faults were considered:

1. from 30 min to 130 min a leak in valve q_s coupling, producing a loss of 5% from $q_{s_{\max}}$,
2. a clog in valve q_c of up to 5% from $q_{c_{\max}}$ income between 400 min and 500 min,
3. at 220 min to 320 min an exponential calibration error in the V sensor of up to 1.5% from $V_{\max}$,
4. since 580 min up to 680 min an exponential accumulation of tartar in the sensor of T, inducing an error of up to 1% from $T_{\max}$.

The high slopes in these faults were chosen to illustrate worst-case conditions and expose the capacity of the proposed FDD scheme to detect, isolate and diagnose such cases. A white measurement noise was added, as well.

Figures 3.9 to 3.12 show the simulation results when the actuators and sensors faults occur. In fact, the estimated output signals of the non-linear system from Equation (3.43), and their output measures are depicted in Figure 3.9. It is easy to see that, when a sensor fault affects the system, its control loop is unable to reach the target set-point (due to the erroneous measurement of process variables). In contrast, when an actuator becomes faulty the control loop (PI controller) automatically compensates the tracking error, and as shown in Figure 3.9 only a small peak is appreciated (this compensation is not possible when the control input is subject to restrictions). On another note, it is possible to appreciate the convergence capacity at the beginning.

Furthermore, the Figure 3.10 exhibits the behaviour of the weighting functions. As expected, the weight varies in concordance with the set-points changes.

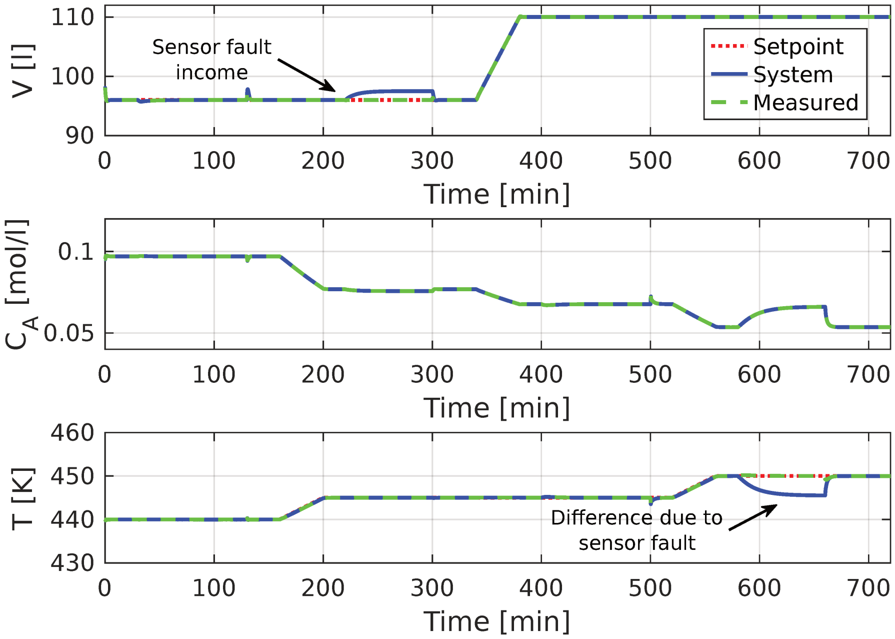


Figure 3.9: CSTR volume, concentration and temperature.

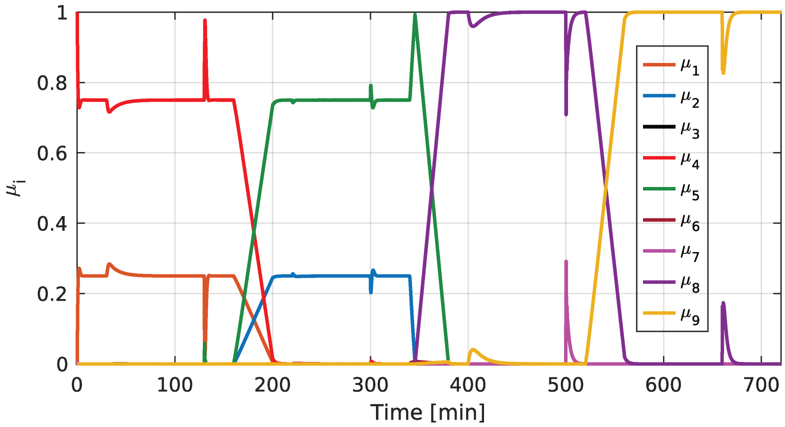


Figure 3.10: CSTR weighting functions.

Moreover, the Figure 3.11 plots the inputs with the actuators fault occurrence. Here, the controller overused the control inputs to compensate the actuators faults (this is only possible when the system has no restrictions)

Besides, the Figure 3.12 shows the actuators (Figure 3.12a) and sensors (Figure 3.12b) faults estimation. There, the effects introduced due to the use of a detection strategy based on threshold, is appreciated. As well as, estimation errors of small magnitude.

Finally, the overhead time percentage with respect of T_s was measured[2]. Being 0.105 %, 0.111 % and 1.915 % the minimum, the average and the maximum values, respectively. As can be appreciated there, it is not a problem to carry out these FDD methods on-line as a part of an AFTCS scheme.

[2] Using an i5-3337U CPU@2.7 GHz (2 Cores) with 6 GB of RAM.

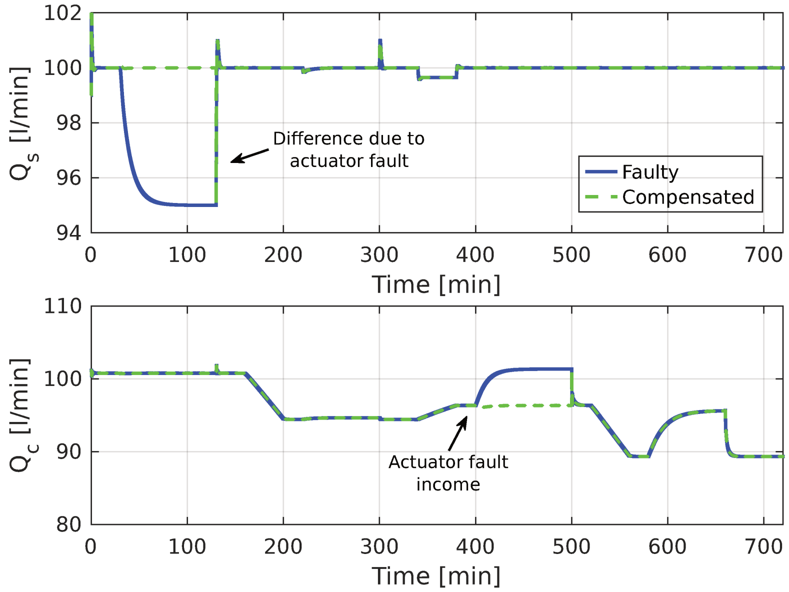
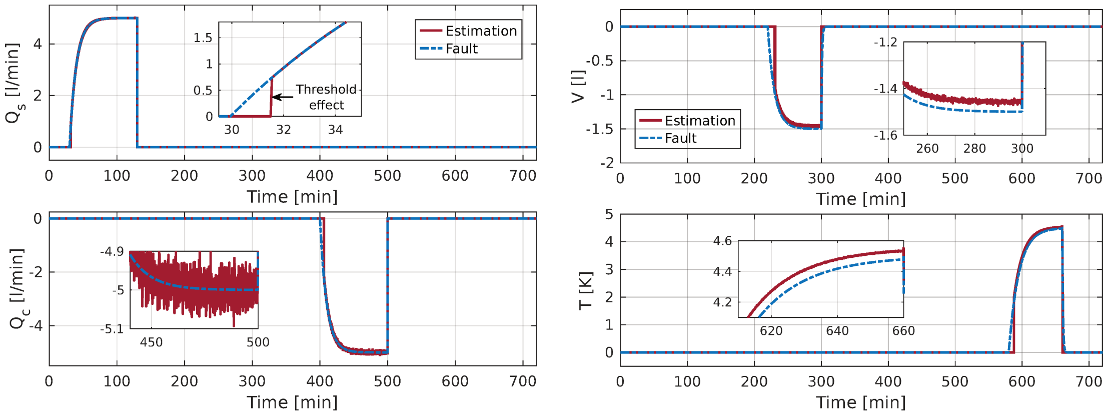


Figure 3.11: CSTR control inputs.



(a) CSTR actuator fault estimation.

(b) CSTR sensor fault estimation.

Figure 3.12: CSTR faults estimation.

3.8 Conclusion

This chapter presents the design of an FDD scheme formed by a bank of two types of observers, applied to LPV systems. The proposed observers and its stability conditions are based on the solving of an LMI problem, that had been performed using software elements.

It is important to note that the main result of this chapter is to address a novel design of observers (LPV-RUIO and LPV-UIOO) that allow multiple actuator and sensor fault detection and diagnosis, from the linear-like design tools and the proposition of LMI problems. Therefore, it enables the development of a particular observer bank to detect and diagnose faults in LPV systems, over continuous-time as much as discrete-time.

In addition, with the purpose of highlighting the behaviour of the proposed

FDD strategy, two numerical simulations of typical chemical industrial processes were given. Consequently, these simulations results confirm the performance and effectiveness of the proposed scheme over non-linear systems in the presence of external disturbance.

At last, in order to enable the reader to replicate and enhance the results shown here, the scripts used are available on a repository[3] under GPL-v3.0 license.

[3] https://github.com/ebernardi/FDD

Chapter 4

Linear Parameter Varying Model Predictive Control

In this chapter a predictive controller formulation is developed within an LPV formalism, which serves as a non-linear process model. The proposed strategy is an adaptive MPC, designed with terminal set constraints and considering the scheduling polytope of the model. At each sample time, two QPs programs are solved: the first QP considers a backward horizon to find a virtual model-process tuning variable that defines the best LTI prediction model, regarding the vertices of the polytopic system; then, the second QP uses this LTI model to optimise performances along a forward horizon. This chapter ends with a realistic *Solar-Thermal* (ST) collector process simulation, comparing the proposed MPC to other techniques from the literature. Discussions concerning the results, the design procedure and the computational effort are presented.

Finally, it should be noted that the main results of this chapter can be found in works [90, 91, 92].

4.1 Introduction

Bearing in mind the concepts from Chapter 2 where the traditional MPC method was shortly presented, it should be noted that due to its formulation, the prediction model is the cornerstone of MPC [70]. Thus, the inclusion of a non-linear process model is not a trivial task and heavily increases the complexity of the resulting optimisation problem [93], which makes it tougher for real-time modules. Accordingly, if sought to be really implementable, these *Non-linear Model Predictive Control* (NMPC) algorithms must be adapted, or approximated, by reducing complexity and resorting to some sub-optimality, as it is done with real-time iteration methods, such as ACADO [94], and gradient-based, such as GRAMPC [95].

In such a way, at the same time that literature presented many advances regarding MPC tools, many others have been devoted to design methods for LPV systems [61, 96]. Specifically, as introduced in Section 2.6, LPV models are used to represent non-linear systems as parameter dependent on known and bounded scheduling parameters. Furthermore, as recently reviewed in [97], MPC strategies based on LPV models for non-linear processes are a promising lead for real-time applicable NMPC algorithms. Accordingly, it becomes evident that fast MPC methods for non-linear systems are lacking. Therefore, the main motivation of this chapter is to propose an MPC scheme for non-linear processes.

Hence, the proposed scheme is an adaptive control method that determines the optimal control policy through two consecutive QP problems. The first QP is an MHE [98, 78], which uses available data from a backward horizon of N_e steps, to minimise the difference from the data and a polytopic LPV model with a fixed virtual tuning variable used to determine an LTI model which better represents the real process at the current operating point. Then, considering that the parameter remains constant throughout the following N_p steps, an MPC problem is solved, through a second QP. Moreover, this *Moving Horizon Estimation - Model Predictive Control* (MHE-MPC) method also includes terminal ingredients (stage cost and set constraints) to ensure stability despite the model simplifications.

4.2 Model statement

Evoking the discrete-time LPV model from Section 2.6:

$$\begin{aligned} x(k+1) &= \sum_{i=1}^{M} \mu_i\left(\zeta(k)\right)\left\{A_i x(k) + B_i u(k) + \Delta x_i\right\} \\ y(k) &= \sum_{i=1}^{M} \mu_i\left(\zeta(k)\right)\left\{C_i x(k) + D_i u(k) + \Delta y_i\right\}, \end{aligned} \tag{4.1}$$

assuming that

$$\Omega = \mathrm{Co}\left\{[A_1, B_1], [A_2, B_2], \ldots, [A_M, B_M]\right\},$$

and considering the goal of applying MPC to a non-linear process, the LPV framework offers some advantages that can be used to simplify this non-linear problem into a QP formulation. In a regular NMPC formulation, it would be imperious to know the exact behaviour of the process model along the prediction horizon. Besides, in the *quasi-Linear Parameter Varying* (qLPV) case, it is necessary to know the future values of the scheduling parameter along N_p, coupled as

$$\Gamma(k) = \mathrm{col}\left\{\rho(k+1), \rho(k+2), \ldots, \rho(k+N_p-1)\right\}. \tag{4.2}$$

Thus, the main advantage of LPV setting appears with regard to $\Gamma(k)$, since the LPV model can be described, for all future instants $k+j$, by a generic pair $[A(\rho(k+j)), B(\rho(k+j))]$ which belongs to the polytope Ω. Therefore, any pair can be represented as a convex combination of the LTI vertices of this polytope as follows:

$$A(\rho(k+j)) = \sum_{i=1}^{M} \mu_i(k+j) A_i \quad \text{and} \quad B(\rho(k+j)) = \sum_{i=1}^{M} \mu_i(k+j) B_i$$
$$\text{with} \ \sum_{i=1}^{M} \mu_i(k+j) = 1 \quad \text{and} \quad 0 \le \mu_i(k+j) \le 1, \quad i \in \mathbb{Z}_{1:M}.$$

Note that each $\mu_i(k+j)$ is a weighting variable that determines how much does the i-th vertex of the scheduling polytope (LTI model) represents the LPV model at a given future instant $k+j$ (see more in Appendix B.3).

In other words, due to this polytopic characteristic of the LPV model, $\Gamma(k)$ can be replaced in the MPC problem by the respective convex sum of the M LTI models, which are always known. Thus, if the M weighting variables $\mu_i(k+j)$ are assumed to be known along N_p, then the non-linear problem is converted into a QP version, which can be evaluated much faster than full-blown NMPC procedures.

Remark 4.1. *Instead of using the polytopic representation of the LPV model, it could also be admitted that ρ is known for all future instants inside the N_p horizon. However, to say that the complete future scheduling vector $\Gamma(k)$ is known is obviously false, since only the instantaneous value of ρ is known. Thus, in [99, 100] different estimation strategies are used to provide a guess $\hat{\Gamma}(k)$ at each instant k to substitute $\Gamma(k)$ in the MPC problem and render it as QP version of this control problem.*

4.3 Adaptive MHE-MPC method

With the previous formalities in mind, this section presents an adaptive MPC design procedure for non-linear systems modelled under an LPV formalism. This MHE-MPC regulation policy tries to find, at each sampling instant k, the best LTI prediction model for the next N_p steps, based on the previous N_e steps of data, and uses some terminal ingredients to guarantee stability. The basic idea behind this method is to consider the LPV polytopic combination variables μ_i as virtual weights that, at each sampling instant, indicate which is the best LTI combination model that can be used to momentarily describe the controlled process.

Therefore, from now on, the vector μ_i is treated as a new decision variable of the optimisation problem. In this sense, the MHE-MPC method adapts the process model to the uncertain system into a single LTI prediction model, at each sampling instant, weighting the LTI vertex of polytope Ω to find the *ideal* one for the prediction of state variables for the following N_p steps.

Due to the simplification of finding an LTI prediction model, the major consequence is having model-process mismatches regarding the control horizon. This means that the proposed method is obviously *a priori* sub-optimal, which does not mean that the achieved results will not be very near the optimal condition. Moreover, a great advantage of using a simplified prediction model is that the MHE-MPC frameworks yields one QP for *identification* purposes (regarding μ_i) and another QP for control purposes, which can be solved on-line with fast solvers and used for a real-time implementation of the MHE-MPC strategy for industrial processes.

4.3.1 Backward QP - The MHE

This problem is used to find a constant vector μ_i that optimally matches the polytopic LPV model to the past real data set. Indeed, this procedure minimises the model-data discrepancy with respect to μ and the variance of μ (ν_μ) throughout the backward horizon, at each sample time.

This virtual tuning variable is found with the solution of the following optimisation problem, considering x and u as measured data and $\mu(k-1)$ as the result from the previous iteration:

$$\begin{aligned}
\min_{\mu} \quad & \sum_{j=k+N_e-1}^{k} \left(e(j)^T Q_e e(j) + \nu_\mu^T Q_\nu \nu_\mu \right) \\
\text{s.t.} \quad & e(j+1) = x(j+1) - \left(Ax(j) + Bu(j) \right), \\
& A = \sum_{i=1}^{M} \mu_i A_i \quad \text{and} \quad B = \sum_{i=1}^{M} \mu_i B_i, \\
& \sum_{i=1}^{M} \mu_i = 1, \qquad 0 \leq \mu_i \leq 1, \\
& \mu = \operatorname{col}\{\mu_i(k)\}, \quad \mu = \mu(k-1) + \nu_\mu,
\end{aligned}$$

with $j \in \mathbb{Z}_{k-N_e:k-1}$ and $i \in \mathbb{Z}_{1:M}$. Moreover, N_e is the estimation horizon and the matrices Q_e and Q_ν are tuning weights of this optimisation procedure. For simplicity's sake, they are taken as weighted identity matrices.

Remark 4.2. *Note that the above QP works exactly as the MHE scheme for the estimation of time-varying parameters, proposed in the literature [78, 98] and briefly introduced in Section 2.9.*

4.3.2 Forward QP - The MPC

On the other hand, the forward QP is used to obtain a control law regarding constraints on the states, inputs and outputs. To achieve this, the *MPC for tracking* method [101] is considered for regulation purposes. Specifically, this control design ensures that the controller asymptotically lead the process to a steady-state reference x_s in an admissible trajectory from any feasible initial state x_0. The approach consists basically in adapting the standard MPC cost function (i.e. weighting the quadratic difference between output and reference).

Remark 4.3. *The MPC for tracking design embeds an artificial reference x_a to the optimisation problem and sets the system states to track this artificial variable. Altogether, it determines that this artificial set-point must be as close as possible to the real state reference x_s, which altogether ensures an enlarged domain of attraction. The target operation point $p_s = (x_s, u_s)$ is an admissible steady-state for the system.*

Assumption 5. *Consider $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$ as positive definite matrices; and $\kappa \in \mathbb{R}^{m \times n}$ as an arbitrary stabilising state-feedback control gain of the process model. For these matrices, it is implied that $A_i + B_i\kappa$ is quadratically Schur. Then, there exists another positive definite matrix $P \in \mathbb{R}^{n \times n}$ such that*

$$(A_i + B_i\kappa)^T P (A_i + B_i\kappa) - P \leq -(Q + \kappa^T R \kappa)$$

Moreover, as long as the Assumption 5 holds, the MPC problem is formulated with the following cost function, considering that μ represents the value obtained with the backward MHE scheme, i.e.:

$$V_N(x, x_s, \mu; \mathbf{u}, x_a, u_a) = V_N^d(x, \mu; \mathbf{u}, x_a, u_a) + V_N^t(x_s; x_a)$$

with

$$V_N^d(x, \mu; \mathbf{u}, x_a, u_a) = \sum_{j=0}^{N_p-1} \|x(j) - x_a\|_Q^2 + \|u(j) - u_a\|_R^2,$$

where N_p is the prediction horizon. The couple (x_a, u_a) is the so-called artificial reference, which represents the best steady state reachable by the system in N_p steps, starting from the initial condition $x = x(k)$. At last, $V_N^t(x_s; x_a)$ is the terminal cost function, which is defined as,

$$V_N^t(x_s; x_a) = \|x_a - x_s\|_\varphi^2 + \|x(N_p) - x_a\|_P^2,$$

being φ and P the appropriates weighting matrices. Then the term $\|x(N_p) - x_a\|_P^2$ is an offset that penalises the final state deviation from this target operation and the offset term $\|x_a - x_s\|_\varphi^2$ ensuring that the artificial variable tracks the real set-point variable, with the actual target goal p_s. Note that the inclusion of this suitable penalisation of the terminal state $x(N_p)$ can steer to asymptotic stability with good performances, as evidenced in [102].

Remark 4.4. *The objective of the inclusion of the artificial target point p_a works as follows. Consider that the system evolves as predicted (with μ_i representing the weight for the LPV model) and that the actual target point $p_s = (x_s, u_s)$ is an admissible point contained inside the tracking set $\mathbb{T} := \mathbb{X} \times \mathbb{U}$ and that it can be tracked within N_p steps. Then, p_a becomes an asymptotically stable point in closed-loop, since the MPC will ensure convergence to it. If the system cannot ensure that the target reference p_s is tracked within the horizon of N_p steps, then the artificial reference x_a enables it to stabilise at more options of closed-loop equilibria points, as close as possible to x_s, since x_a is set to converge to the actual target. Note that the weighting matrix φ is taken as a parameterised version of P, i.e. $\varphi = \alpha P$, with $\alpha \in \mathbb{R}$.*

Finally, at each time step $k = k_0$, the controller is found with the solution of the following optimisation problem:

$$\begin{aligned} \min_{\mathbf{u}, x_a, u_a} \quad & V_N(x, x_s, \mu; \mathbf{u}, x_a, u_a), \\ \text{s.t.} \quad & x(0) = x, \\ & x(j+1) = Ax(j) + Bu(j), \\ & x_a = Ax_a + Bu_a, \quad (u_a, x_a) \in \mathbb{Z}_s = \mathbb{X}_s \times \mathbb{U}_s, \\ & A = \sum_{i=1}^{M} \mu_i A_i \text{ and } B = \sum_{i=1}^{M} \mu_i B_i, \\ & x(j) \in \mathbb{X}, \; u(j) \in \mathbb{U}, \quad \forall j \in \mathbb{Z}_{0:N_p-1}, \\ & x(N_p) \in \mathbb{X}_f, \end{aligned} \tag{4.3}$$

where $i \in \mathbb{Z}_{0:N_p-1}$, and $\mathbb{X}_f$ is an adequate robust terminal invariant set that contains p_s. To determine this invariant set, some suggestions are also given in [103, 104, 105]. Synthetically, the essential idea on how to determine this robust invariant set for the LPV system from Equation (4.1), is to find $\mathbb{X}_f \subset \mathbb{X}$ so that for all possible $x(k_0) \in \mathbb{X}_f$ there must exist a feasible input $u = \kappa(x(k_0)) \in \mathbb{U}$ which guarantees that $x(k_0 + 1)$ lies inside $\mathbb{X}_f$ despite model-process mismatches, as shown in Section 4.3.3.

After all, by considering a receding horizon policy, the proposed regulation control policy is given by $\kappa(x(k_0)) = \mathbf{u}(0)$.

4.3.3 Stability analysis

In Section 2.8 the terminal ingredients was introduced as the key elements to ensure stability and recursive feasibility of state-feedback predictive control loops. Thus, the usual approach with terminal ingredients resides in ensuring that conditions are met by: (a) the terminal set $\mathbb{X}_f$ and (b) the terminal cost $V_f(x(N_p))$ with respect to a nominal state-feedback controller $u(k) = \kappa(x(k))$, which is usually the unconstrained solution of the MPC problem. Therefore, this section presents the development of a sufficient stability condition for the proposed MHE-MPC mechanism in order to verify these conditions. The approach here presented is based on the vast existing related-literature (i.e. [100, 106, 107, 108, 109, 110, 111, 112, among others])

Terminal Ingredients

Firstly, to define the terminal ingredients it must be considered that there exists a nominal state-feedback gain $\kappa : \mathbb{R}^n \to \mathbb{R}^{n \times m}$. For demonstration simplicity and notation lightness, a regulation problem will be considered. However, for the tracking case, the nominal feedback is given by $u(k) = \kappa\left(x(k) - x_s\right)$ (the terminal constraint $(x(N_p) - x_s) \in \mathbb{X}_f$ and the terminal cost $V(x(N_p) - x_s)$).

This nominal controller purpose is to demonstrate stability and recursive feasibility properties of the proposed MHE-MPC mechanism. Anyhow, it represents the infinite-horizon *Linear-Quadratic Regulator* (LQR) solution for an LPV system model from Equation (4.1), which verifies

$$\kappa_{LQR} = \arg\min_{\kappa_{LQR}} \sum_{j=0}^{\infty} \left(\|x(k+j)\|_Q^2 + \|u(k+j)\|_R^2 \right)$$

Besides, for regularity, the largest ellipsoidal invariant set is considered as the terminal constraint $\mathbb{X}_f$.

Definition 4.1 (Largest invariant ellipsoid [66]). *Given the discrete dynamical system from Equation* (4.1), *a subset*

$$\xi = \{x(k) \in \mathbb{R}^n \mid x(k)^T W x(k) \leq 1\}$$

of the state space $\mathbb{R}^n$ is said to be an asymptotically stable invariant ellipsoid, if it has the property that, whenever $x(k_1) \in \xi$, then all trajectories $x(k) \in \xi$ for all times $k \geq k_1$ and $x(k) \to 0$ as $k \to \infty$.

Which results in an ellipsoid centred at the origin,

$$\mathbb{X}_f = \left\{ x(N_p) \mid x(N_p)^T W x(N_p) \leq 1 \right\}. \tag{4.4}$$

Furthermore, this terminal set is a sub-level set of terminal cost $V(\cdot)$, which is taken as a Lyapunov function as follows:

$$V(x(N_p)) = x(N_p)^T P x(N_p).$$

Accordingly, this nominal feedback gain κ_{LQR} and the terminal ingredients verbalised through the symmetric Lyapunov matrix P are so that the following *Input-to-State Stability* (ISS) theorem is guaranteed.

Theorem 4.1 (Input-to-State Stability). *Assume that a nominal control law $u = \kappa(x(k)) = Kx(k)$ exists. Moreover, consider that the MPC is in the framework of the optimisation problem from Equation* (4.3)*, with a terminal state set given by $\mathbb{X}_f$ and a terminal cost $V(x(N_p))$. Then, the ISS is ensured if the following conditions are hold $\forall \mu_i(\cdot) \in \mathcal{P}$:*

(C1) The origin lies in the interior of $\mathbb{X}_f$.

(C2) Any consecutive state to $x(k)$, given by $(A_i + B_i K)\, x(k)$ lies within $\mathbb{X}_f$ (i.e. this is an invariant set).

(C3) The discrete algebraic Ricatti equation is verified within this invariant set, this is, $\forall x(k) \in \mathbb{X}_f$:

$$V(x(k+j+1)) - V(x(k+j)) \leq -x(k+j)^T Q x(k+j) - u(k+j)^T R u(k+j).$$

(C4) The image of the nominal feedback always lies within the admissible control input domain: $Kx(k) \in \mathbb{U}$.

*Assuming that the initial solution of the MPC problem **u**, computed with respect to the initial state $x(0)$, is feasible, the MPC algorithm is indeed recursively feasible, asymptotically stabilising the state origin [113].*

Proof. Consider an initial state condition $x(k_0)$. Assume that the optimal control sequence at time k_0 is

$$\mathbf{u}^\star(k_0) = \left[u^\star(k_0), u^\star(k_0+1), \ldots, u^\star(k_0+N_p-1) \right]^T.$$

Then, the next feasible solution (possibly sub-optimal) at instant k_0+1 is given by:

$$\mathbf{u}(k_0+1) = \left[u(k_0+1), u(k_0+2), \ldots, u(k_0+N_p-1), Kx(k_0+N_p) \right]^T.$$

Accordingly, the MPC cost function $V_N(\cdot)$ evolution is,

$$\begin{aligned} V_N(k_0) &= \sum_{j=0}^{N_p-1} \left(\|x(k_0+j)\|_Q^T + \|u(k_0+j)\|_R^T \right) + V(x(k_0+N_p)), \\ V_N(k_0+1) &= \sum_{j=0}^{N_p-1} \left(\|x(k_0+1+j)\|_Q^T + \|u(k_0+1+j)\|_R^T \right) + V(x(k_0+1+N_p)), \end{aligned}$$

and its decay between time instants k_0 and k_0+1 results,

$$\begin{aligned}\Delta V_N(k_0) = V_N(k_0+1) - V_N(k_0)\\ = - x(k_0)^T Q x(k_0) - u(k_0)^T R u(k_0) + V(x(k_0+1+N_p))\\ + x(k_0+N_p)^T (Q + K^T R K) x(k_0+N_p) - V(x(k_0+N_p)).\end{aligned}$$

Assuming that $u^\star(k_0)$ was indeed applied to the plant at instant k_0 (this input is the first entry of the optimal solution $\mathbf{u}^\star(k_0)$ at time k_0), it follows that:

$$\begin{aligned}\Delta V_N(k_0) = - x(k_0)^T Q x(k_0) - u^\star(k_0)^T R u^\star(k_0) + V(x(k_0+1+N_p))\\ + x(k_0+N_p)^T (Q + K^T R K) x(k_0+N_p) - V(x(k_0+N_p)).\end{aligned} \tag{4.5}$$

From condition (C3), it can pursue with the negativeness of the following term:

$$0 \geq V(x(k_0+N_p+1)) - V(x(k_0+N_p)) + x(k_0+N_p)^T \left(Q + K^T R K\right) x(k_0+N_p).$$

Thus, substituting this inequality into Equation (4.5), it results:

$$\Delta V_N(k_0) \leq -x(k_0)^T Q x(k_0) - u^\star(k_0)^T R u^\star(k_0).$$

Since Q and R are positive defined matrices by construction (they are the MPC tuning weights matrices) and $u^\star(k_0)$ and $x(k_0)$ are known[1], it follows that:

$$\Delta V_N(k_0) \leq 0,$$

which means that the MPC cost function decays along k.

From the optimality of the solution $\mathbf{u}^\star(k+1)$, it follows that the cost constructed with this sequence ($V_N^\star(k_0+1)$) holds as a lower-bound with respect to $V_N(k_0+1)$, this is $V_N^\star(k_0+1) \leq V_N(k_0+1)$. Then, using this argument on $\Delta V_N(k_0)$ it arrives at:

$$\begin{aligned}V_N(k_0+1) - V_N(k_0) \leq 0,\\ V_N^\star(k_0+1) \leq V_N(k_0+1) \leq V_N(k_0),\end{aligned}$$

which proves that $V_N^\star$ is a Lyapunov function and x will converge to the origin (due to condition (C1)), as long as the initial condition provides, in fact, a feasible starting point. Note that condition (C2) is necessary to map a feasible $x(k_0+1)$. Condition (C4) is necessary to ensure the admissible inputs u. □

Now, in order to find some nominal state-feedback gain K, some terminal set $\mathbb{X}_f$ and some terminal offset cost $V(\cdot)$, an offline LMI problem is proposed in the sequel. These LMI problems are such that a positive definite matrix P is found to ensure that the conditions of Theorem 4.1 are satisfied.

These LMI problems are provided through the following theorem, which aims to find the largest terminal set $\mathbb{X}_f$ that is invariant under the nominal control policy $u(k) = \kappa(x(k))$ for all k, while remaining admissible, i.e. $\kappa(x(k)) \in \mathbb{U}, \forall x(k) \in \mathbb{X}$. Note that the previously defined largest ellipsoidal set is posed through a maximisation problem.

[1] Note that $x(k_0+1) = A_i x(k_0) + B_i u^\star(k_0)$.

Theorem 4.2 (Terminal Ingredients)**.** *The conditions (C1)-(C4) of Theorem 4.1 are satisfied if there exist the symmetric positive definite matrices* $P : \mathbb{R}^n \to \mathbb{R}^{n\times n}$, $Z : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ *and a rectangular matrix* $L : \mathbb{R}^n \to \mathbb{R}^{m\times n}$ *such that* $P = \gamma Y^{-1}$, $K = LY^{-1}$, $W = Z^{-1}$ *and that the following LMIs hold for all* $\mu_i(\cdot) \in \mathcal{P}$.

Thus, in order to construct the suitable matrices K, P *and* W, *the following consecutive problems are considered:*

Feasibility problem:

$$\begin{aligned} \min_{Y,L,\gamma} \quad & \gamma \\ s.t. \quad & Y \succ 0, \\ & \begin{bmatrix} Y & YA_i^T + L^T B_i^T & YQ^{1/2} & L^T R^{1/2} \\ A_i Y + B_i L & Y & 0 & 0 \\ Q^{1/2} Y & 0 & \gamma I & 0 \\ R^{1/2} L & 0 & 0 & \gamma I \end{bmatrix} \succ 0. \end{aligned} \tag{4.6}$$

Terminal region problem:

$$\begin{aligned} \max_{Z} \quad & det(Z) \\ s.t. \quad & Z \succ 0, \\ & \begin{bmatrix} -Z & Z(A_i + B_i K)^T \\ (A_i + B_i K)Z & -Z \end{bmatrix} \prec 0, \\ & K_l Z K_l^T \leq \overline{u}_l^2. \end{aligned}$$

where K_l *denotes the* l*-th row of* K *and* $\overline{u}_l$ *refers to the bound of the* l*-th input.*

Proof. To demonstrate the validity of this theorem through its synthesis, all conditions of Theorem 4.1 will be satisfied.

Firstly, note that condition (C1) trivially holds due to the form of $\mathbb{X}_f$. As gives Equation (4.4), this terminal set is defined as an ellipsoid, which holds the origin of $\mathbb{R}^n$ by construction.

The condition (C2), control invariance of the terminal set $\mathbb{X}_f$, is verified as follows: suppose that the terminal constraint set is an ellipsoid (Definition 4.1). It is possible to apply the state feedback law $u = Kx(k)$ to any non-zero $x(k) \in \xi$, which according to the invariance condition results in

$$x(k+j+1)^T W x(k+j+1) \leq x(k+j)^T W x(k+j).$$

Thus, considering the system from Equation (4.1)

$$x(k+j)^T \left((A_i + B_i K)^T W (A_i + B_i K) - W \right) x(k+j) \leq 0,$$

that is

$$(A_i + B_i K)^T W (A_i + B_i K) - W \leq 0.$$

Then, pre and post-multiplying by Z and considering $W = Z^{-1}$,

$$Z - Z(A_i + B_i K) Z^{-1} (A_i + B_i K) Z \succ 0.$$

Finally, applying the Schur complement (see Appendix C), it results in:

$$\begin{bmatrix} Z & Z(A_i + B_iK)^T \\ (A_i + B_iK)Z & Z \end{bmatrix} \succ 0$$

ensuring that ξ is an asymptotically stable invariant ellipsoid. Proving the condition (C2).

The discrete Ricatti condition (C3) is verified through the solution of the LMI from Equation (4.6). That is, consider a quadratic function $V(x(k)) = x(k)^T P x(k)$, $P \succ 0$ at the state $x(k)$ of the system from Equation (4.1). At sampling k, suppose $V(\cdot)$ satisfies the following inequality for all $x(k+j)$, $u(k+j)$ with $j \geq 0$. As a consequence,

$$V(x(k+j+1)) - V(x(k+j)) \leq -\left(x(k+j)^T Q x(k+j) + u(k+j)^T R u(k+j)\right).$$

From the quadratic function $V(\cdot)$ required to satisfy condition (C3) and substituting $u(k+j) = Kx(k+j)$

$$\begin{aligned} x(k+j+1)^T P x(k+j+1) - x(k+j)^T P x(k+j) \leq & - x(k+j)^T Q x(k+j) \\ & - x(k+j)^T K^T R K x(k+j) \end{aligned}$$

moreover, considering the system from Equation (4.1) to replace $A_{i_{cl}} = A_i + B_iK$

$$x(k+j)^T \left(A_{i_{cl}}^T P A_{i_{cl}} - P + Q + K^T R K\right) x(k+j) \leq 0.$$

That is,

$$A_{i_{cl}}^T P A_{i_{cl}} - P + Q + K^T R K \leq 0$$

with $P = \gamma Y^{-1}$; $K = LY^{-1}$. Then,

$$A_{i_{cl}}^T \gamma Y^{-1} A_{i_{cl}} - \gamma Y^{-1} + Q + Y^{-1} L^T R L Y^{-1} \leq 0$$

pre and post-multiplying by Y (which leaves the inequality unaffected),

$$(A_iY + B_iL)^T \gamma Y^{-1} (A_iY + B_iL) - \gamma Y + YQY + L^T R L \leq 0,$$

using Cholesky factorisation and multiplying all terms by γ^{-1}

$$Y(Q^{1/2})^T \gamma^{-1} Q^{1/2} Y + L^T (R^{1/2})^T \gamma^{-1} R^{1/2} L - (A_iY + B_iL)^T Y^{-1} (A_iY + B_iL) - Y \succ 0.$$

Finally, applying the Schur complement (see Appendix C), it results in:

$$\begin{bmatrix} Y & YA_i^T + L^T B_i^T & YQ^{1/2} & L^T R^{1/2} \\ A_iY + B_iL & Y & 0 & 0 \\ Q^{1/2}Y & 0 & \gamma I & 0 \\ R^{1/2}L & 0 & 0 & \gamma I \end{bmatrix} \succ 0$$

proving the condition (C3)

Lastly, the condition (C4) is verified through the maximum norm of the projection $Kx(k)$ of any state point $x(k)$ that belongs to some ellipsoid $x(k)^T W x(k) \leq 1$, which is given by $\sqrt{KW^{-1}K^T}$ [113]. It means that the projection $K_l x(k)$ (i.e. l-th control signal) is bounded, in norm, by $\overline{u}_l$. That is,

$$K_l W^{-1} K_l^T \leq \overline{u}_l^2,$$

replacing $W = Z^{-1}$, it results in:

$$K_l Z K_l^T \leq \overline{u}_l^2.$$

Satisfying the condition (C4). □

Remark 4.5. *The LMIs of Theorem 4.2 are solved over the admissible parameter set $\forall \mu_i(\cdot) \in \mathcal{P}$. This choice is conservative since the conditions need only to hold within the terminal region $\mathbb{X}_f$. However, the terminal region is calculated only after the solutions of such LMIs. Further conservatism comes from considering constant unknown matrices Y, Z, K. The conditions can be relaxed by including parameter dependency with relevant basis functions.*

It must be noted that the above proof demonstrates that the solution of the LMIs presented in Theorem 4.2 ensures the existence of positive definite matrices P and W which are used to compute the MPC terminal ingredients $V(\cdot)$ and $\mathbb{X}_f$ such that ISS of the closed-loop is guaranteed, verifying the conditions of Theorem 4.1. Furthermore, when the MPC is designed with these terminal ingredients, for whichever initial condition $x(0) \in \mathbb{X}_f$, it remains recursively feasible for all consecutive discrete time instants $k > 0$.

4.4 quasi-Linear Parameter Varying Model Predictive Control method

One of the motivations of this chapter is to compare some predictive control methods that have been applied for ST collector systems. For this goal, the proposed MHE-MPC method will be compared to a *quasi-Linear Parameter Varying Model Predictive Control* (qLPV-MPC) technique, which is based on a sequence of control invariant sets to embed the non-linearities. Therefore, this section rapidly recalls the qLPV-MPC design procedure, from [90].

For this method the system model from Equation (4.1) is now considered as a qLPV model, so the application of this model to the standard MPC, from Equation (2.8), is in fact a non-linear problem. Because the values from Equation (4.2) are needed to solve the optimisation procedure, but they are unknown at the instant k, as discussed in Section 4.2.

Thus, the rationale of the following method is to convert the standard MPC problem (including the qLPV model) into a QP, which can be tackled by most standard solvers. To do so, a major relaxation is needed, which is to consider a guess for the evolution scheduling parameters, namely $\hat{\Gamma}(k)$. According to [99, 100], this guess is taken as $\hat{\rho}(k+j) = \rho(k)$ for $j \in \mathbb{Z}_{0:N_p-1}$, which translates the prediction model into:

$$x(k+j) = A^j(\hat{\Gamma}(k))x(k) + B^j(\hat{\Gamma}(k))\mathbf{u},$$

which is linear because the non-linear terms become constant matrices, dependent on the frozen $\hat{\Gamma}(k)$.

Performing this simple relaxation does not in itself guarantee feasibility nor optimality of the control solution $\kappa(x(k))$. Therefore, the *MPC for tracking* method, proposed by [101] (presented in Section 4.3.2), and a contractive terminal set constraint must be added. As a consequence, definitions and concerned aspects to the contractive terminal set are introduced in the sequel.

In [114], the notion of control-invariant reachable set is presented and must be recalled:

Definition 4.2. *A set $\Upsilon \subset \mathbb{R}^n$ is said to be control invariant for the system from Equation* (4.1)*, with $u(k) \in \mathbb{U}$, if, for all possible $x(k) \in \Upsilon$, there exists an admissible input $u = \kappa(x(k))$ so that $x(k+1)$ lies inside Υ.*

Definition 4.3. *The one-step set from Υ, namely $\mathcal{Q}_1\{\Upsilon\}$, is the set of states that the system can steered from within one step k to the target set Υ with an admissible control action $u(k) \in \mathbb{U}$. A given set Υ is control invariant if and only if $\Upsilon \subseteq \mathcal{Q}_1\{\Upsilon\}$.*

Definition 4.4. *A sequence of control invariant sets $\{\Upsilon_j\}$ is the set sequence through which x can be steered through, leaping from one set Υ_j to the following Υ_{j-1}, with feasible control actions, until finally reaching the target invariant set Υ. This is: $\{\Upsilon_j\}$ is a sequence s.t. each $\Upsilon_{j-1} \subseteq \mathcal{Q}_1\{\Upsilon_j\}$ $\forall j$.*

MPC design coupled to the use of control invariant set sequences is used to make sure that the algorithm guarantees asymptotic convergence [102]. Anyhow, to compute a reachable sequence of control invariant sets for the case of qLPV models, the bounds of the scheduling parameters and their derivatives must be taken into account. Note that, for the qLPV case, the one-step-ahead set from state $x(k)$ contains all states derived with scheduling parameter derivatives smaller than or equal to their min/max bounds ($\underline{d\rho}$ / $\overline{d\rho}$). This means that $x(k+1)$, for whichever $\rho(k+1) \in \mathcal{P}$, is, at most, equal to $x^\star(k+1) = A(\rho^\star)x_k + B_1(\rho^\star)u(k) + B_2 w(k)$, where $\rho^\star = \rho(k-1) + dp$, being dp either equivalent to $\underline{d\rho}$ or $\overline{d\rho}$, knowing that $(\underline{d\rho}, \overline{d\rho})$ are the bounds on the scheduling parameter variation rate.

For the qLPV case, the control invariant set Υ must be computed at the first iteration of the MPC algorithm, considering $x = x(k)$ and abiding to:

$$\Upsilon^{\text{Max}} \subseteq \Upsilon \subseteq \Upsilon^{\text{Min}} \quad \text{and} \quad p_s \in \Upsilon, \tag{4.7}$$

where p_s is the target operation point and Υ^{Max} is the set achieved with a sequence of N_r maximal scheduling parameter variations, i.e. $\{\rho(k) + \overline{d\rho}, \rho(k) + 2\overline{d\rho}, \dots\}$, and admissible control laws; Υ^{Min} is the set for a sequence for minimal variations of ρ.

Then, for each iteration k, the sequence of reachable sets is computed as the union of the worst-case wider sets, using the upper $\rho^{+\star} = \rho(k) + \overline{d\rho}$ and the lower $\rho^{-\star} = \rho(k) + \underline{d\rho}$. This is, for $j = \max\{N_r - j, 0\}, \ldots, 0$:

$$\{\Upsilon_j\} := \text{col}\{\Upsilon_j\}, \tag{4.8}$$

where $\Upsilon_j = (x(k+j+1) \in x^{+\star}) \cup (x(k+j+1) \in x^{-\star})$, with $x^{+\star} = A(\rho^{+\star})x(k+j) + B(\rho^{+\star})u(k+j)$ and $x^{-\star} = A(\rho^{-\star})x(k+j) + B(\rho^{-\star})u(k+j)$.

In order to guarantee that, within N_r fixed steps from k_0, the controlled system converges to the target control invariant set Υ in Equation (4.7), containing the steady-state equilibrium goal p_s, the following contractive terminal set constraint is used:

$$x(k_0 + N_r) \in \Upsilon_j, \qquad j = \max\{N_r - k, 0\},$$

with the sequence of control invariant sets given by Equation (4.8). Note that the terminal set Υ_j is equal to the largest Υ at the initial instant k_0 being shrank subsequently until, at $k_0 + N_r$, it becomes the smallest set Υ_{N_r}. Moreover, note that N_p is a sliding-horizon, while N_r is not, being fixed at the first iteration k_0. This means that k gets closer to $k_0 + N_r$ as it advances. This constraint makes the MPC method intrinsically time-varying, since, for N_r samples, the sets are contracting.

With the above constraint coupled to the MPC design, indeed there appear further holds on stability and feasibility, which are needed due to model-plant differences, i.e. $\Gamma(k) \neq \hat{\Gamma}(k)$. The sequence of control invariant sets makes sure that the terminal constraints contracts and the states converge to the desired target p_s (or as closely as possible, due to the pseudo-referencing tool).

4.5 Temperature control in Solar Collectors

The addition of renewable energy sources to power plants are a good way to reduce greenhouse gas emissions and environmental impact. Anyhow, an inherent problem to be solved is how to integrate these energy sources without loosing efficiency and dispatchability of energy plants. As discussed in the literature [115], current solar energy technologies are of two main kinds: photovoltaic systems, that directly covert solar radiance into electric energy, and solar-thermal systems, which usually generate heated fluid (or steam).

Solar radiance is an intermittent energy source. When there occurs a cloudy period of the day, for instance, energy might be running low if no compensation strategy is considered. A practical solution for this matter, adopted in the majority of modern ST systems [116], is to include accumulation tanks to store energy (hot fluid) while there is no process demand, and a complementary (auxiliary) energy source (say, for instance, a gas heater), that could be used when there is no sun or the accumulation tanks are not sufficient to meet the demand fully. A modern ST unit is usually a structure that integrates a solar-thermal collector field, some accumulation tanks and a gas heater. Figure 4.1 gives an illustration of such complete ST collectors.

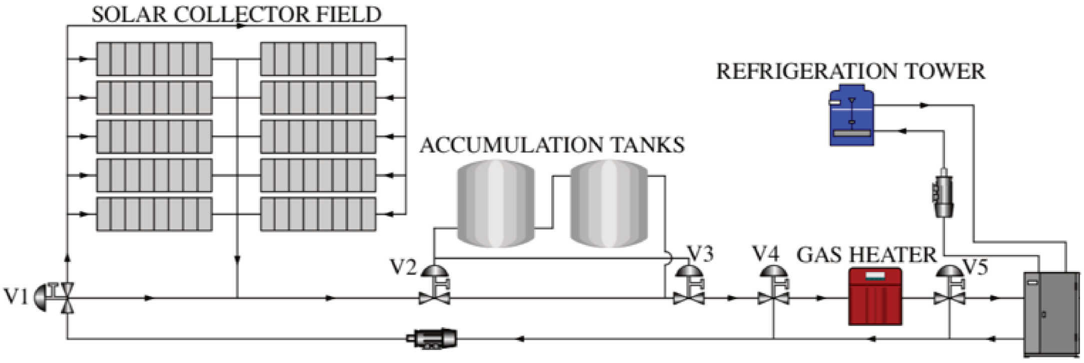


Figure 4.1: Schematic of a modern Solar-Thermal collector system.

Nonetheless, in this chapter, the focus is solely in the temperature regulation of the ST collector panel itself, as depicted in Figure 4.2.

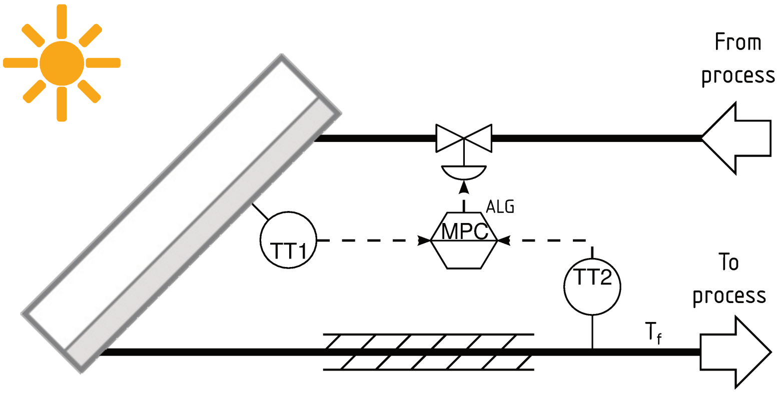


Figure 4.2: Solar-Thermal collector process.

4.5.1 The CIESOL ST plant

Complete phenomenological models have previously been derived for ST collector fields [117, 118], which is according to model-validation [119] and parameter identification procedures [103]. In this chapter, the model is based on the CIESOL ST plant, located in the *CIESOL-ARFR-ISOL R&D* Centre of the University of Almería, Spain. This testbed has a flat ST collector, used to regulate the temperature of the inlet fluid to guarantee a certain heat demand.

Regarding these phenomenological models, they are based on the following assumptions:

- The fluid flow through the solar collector is incompressible, with uniform pressure along the field;
- The heat transfer capacity of the collector plates is constant and denoted C_m; the density of these metal plates is also constant and denoted ϵ_m;
- The balance of energy equations assume a constant thermal loss coefficient ν, with respect to the thermal energy that derives from the incident solar

radiance;

- The heat transfer absorption coefficient (external temperature to plates), denoted h_0, is constant, while the heat transfer coefficient of the fluid (fluid to plates), denoted $h_i(\cdot)$, varies positively according to the temperature of the plates.

Then, the following partial-differential dynamics arise due to balance of energy equations, where t represents the time variable and s the space variable:

$$\begin{aligned}\epsilon_m C_m A_e \frac{dT_p}{dt}(t) &= d_e \pi \nu I(t) - d_e \pi h_0 (T_p(t) - T_e(t)) - d_i \pi h_i (T_p(t))(T_p(t) - T_f(t)) \\ \epsilon_f C_f A_i \frac{\partial T_f}{\partial t}(t,s) &= -u(t) \epsilon_f C_f \frac{\partial T_f}{\partial s}(t,s) + d_i \pi h_i (T_p(t))(T_p(t) - T_f(t)).\end{aligned} \tag{4.9}$$

In these temperature gradient dynamics of Equation (4.9), $I(t)$ stands for solar radiance focused upon the collectors (which is a load disturbance from a control viewpoint); T_p, T_e and T_f are, respectively, the collector plate, the external (load disturbance as well) and the fluid temperatures; u is the inlet fluid flow, which is the control input of the system; finally, A_i and A_e are, respectively, the internal and external surfaces of the pipes, that have (internal and external) diameters of d_i and d_e.

For application purposes, as seen in [117, 119], the space-derivative term $\frac{\partial T_f}{\partial s}(t,s)$ can be replaced by either a non-linear function or an *apparent* transport delay. In resemblance to the use of a transport delay function, it is consider that this term is sufficiently approximated by the following exponential non-linearity, which takes t_{ss} seconds to stabilised, for $T_f(t_{ss}) = T_f^{\max}$:

$$\frac{\partial T_f(t,s)}{\partial s} \approx \frac{1 - e^{-T_f(t)/T_f^{\max}}}{(1 - e^{-1})}, \tag{4.10}$$

which means that the diffusion of the thermal energy of the fluid flowing along the flat collectors increases with respect to its temperature $T_f(t)$ until a certain level is attained $T_f^{\max}$, after which the diffusion is constant, i.e. the whole fluid inside the flat collector is at the same temperature. This approximation is quite reasonable with respect to the ST application and in accordance with [117].

On the other hand, the heat transfer coefficient of the fluid $h_i\left(T_p(t)\right)$ is given according to the following non-linear equation:

$$h_i\left(T_p(t)\right) = \overline{h}_i \left(\frac{1 - e^{-T_p(t)/T_p^{\max}}}{1 - e^{-1}} \right), \tag{4.11}$$

where $\overline{h}_i$ is the maximal heat transfer coefficient of fluid, attained for $T_p(t) = T_p^{\max}$, see [117].

4.5.2 Model parameters

Regarding the non-linear model from Equation (4.9) with the relaxations of Equations (4.10) and (4.11), the parameters have been identified and adjusted for the

CIESOL tested [103]. The numerical values for these parameters are given in Table 4.1.

Table 4.1: Parameters of the ST process.

Name	**Value**	**Name**	**Value**
ϵ_m	$1100\,\mathrm{kg\,m^{-3}}$	C_m	$440\,\mathrm{J\,kg^{-1}\,{}^{\circ}C^{-1}}$
ϵ_f	$1000\,\mathrm{kg\,m^{-3}}$	C_f	$4018\,\mathrm{J\,kg^{-1}\,{}^{\circ}C^{-1}}$
A_e	$0.0038\,\mathrm{m^2}$	A_i	$0.0013\,\mathrm{m^2}$
d_i	$0.04\,\mathrm{m}$	d_e	$0.07\,\mathrm{m}$
h_0	11	$\overline{h_i}$	800
ν	3.655		

4.5.3 Performance goals and constraints

The goal of this ST system is to track outlet temperature references to cover a certain heat demand, which is done by varying the inlet fluid flow u. This collector field has a surface area of $160\,\mathrm{m^2}$, distributed in ten parallel rows composed of eight collectors per row.

In terms of performances, the temperature set-point tracking must be done as fast as possible, while respecting the maximal temperature of 300 °C that the inlet fluid can tolerate. Moreover, the temperature of the plates must not surpass 600 °C. These performances can be evaluated using usual reference-tracking indices (see Appendix D).

The inlet flow (control input) must be always positive (no fluid can be extracted from the ST units, only injected) and abide to a maximal value of $0.35\,\mathrm{m^3\,s^{-1}}$, which is the upper constraint of the injection pump. Moreover, the control policy must be evaluated within $T_s = 3\,\mathrm{s}$, which is the sampling period of this ST unit.

The disturbances to this system (the solar radiance and external temperature variables) are assumed to be measurable from a control viewpoint. This is quite reasonable, given that accurate estimations for the future behaviour (of some hours ahead) of these disturbances can be indeed obtained [115]. These estimation results, specially considering solar data, can be provided with deep and recurrent neural network tools, as seen in [120, 121]. It must be noted that this forecasting problem is, by itself, a complex task with its difficulties, but it is not the focus of this work.

Table 4.2 resumes the state and input constraints. Note that the fluid and plate temperatures are lower-bounded by external temperature to the ST system, $T_e(t)$. If there is no sun, the ST system will reach a thermal equilibrium with $T_e(t)$. For simplicity, since $T_e(t) > 0\,{}^{\circ}\mathrm{C}$, the lower bounds on T_p and T_f can be taken as 0 °C.

Table 4.2: Constraints of the considered ST system.

Variable	**Range**
$u(t) \in \mathbb{U}$	$\mathbb{U} := \{u \in \mathbb{R} \mid 0 \leq u \leq 0.35\,\mathrm{m}^3\,\mathrm{s}^{-1}\}$
$T_p(t) \in \mathbb{T}_p$	$\mathbb{T}_p := \left\{T_p \in \mathbb{R} \mid T_e(t) \leq T_p \leq T_p^{\max}\right\}$, $T_p^{\max} = 600\,°\mathrm{C}$
$T_f(t) \in \mathbb{T}_f$	$\mathbb{T}_f := \left\{T_f \in \mathbb{R} \mid T_e(t) \leq T_f \leq T_f^{\max}\right\}$, $T_f^{\max} = 300\,°\mathrm{C}$

4.5.4 Simulation results

Since the proposed MHE-MPC method has been thoroughly explained and the qLPV-MPC has also been recalled, this section presents realistic simulation results of the considered ST system. These control methodologies are compared against each other but also to a simpler linearisation-based MPC (denoted *standard Model Predictive Control* (sMPC)), which solves the QP in Equation (4.3) taking all weighting variables as $\mu_i = 1/M$, which is a linear model for the ST system, considering the M vertices of the polytope Ω. This last technique can be understood as a nominal, averaged LTI MPC paradigm, since it takes the tuning variables μ_i as equally weighted. As might be expected, this controller is not able to achieve successful results for a large operation region, and it might even lead to infeasibility, since it takes an average of the M model-tuning variables, which obviously does not represent the whole scheduling polytope Ω.

The following results comprise the constrained regulation of the outlet temperature x_2, despite variations upon the solar radiance (I) and outside temperature (T_e) disturbances w. Real meterological data from the region of the CIESOL testbed is used for the solar radiance and temperature disturbances, considering a full hour of simulation [122]. These disturbances are known from a control viewpoint; they are given in Figure 4.3.

Accordingly, the ST system is emulated using a realistic, high-fidelity model, considering the non-linear Equation (4.9). Remember that these non-linear equations have been previously validated to thoroughly emulate the CIESOL ST system [117]. The controllers were synthesised using Matlab with Yalmip toolbox [88]. Moreover, the MHE weighting matrices were set to $Q_e = 1 \times 10^5$ and $Q_\nu = 1$. Besides, the MPC weighting matrices were defined as $Q = \mathrm{diag}\{0\ ,\ 1 \times 10^4\}$ and $R = 1 \times 10^1$. However, it is important to highlight that all implemented MPC algorithms use the same set of weighting matrices, the same prediction horizon $N_p = 10$ ($N_e = 10$ for the MHE-MPC case), and operate with the same sampling period, i.e. $T_s = 3\,\mathrm{s}$.

The reference tracking goal is set as 97 °C for the fluid temperature $x_2(k)$, and 109.93 °C for the plate temperature $x_1(k)$. In fact, the hard-constrained setpoint is the one for x_2, which must be tracked with minimal error as possible. Notice that integral action is not necessary in the Equation (4.3), since ±0.5 °C is tolerated. If sought, integral action could be easily included by defining a new constraint $u(k+i) = u(k+i-1) + \delta u(k+i)$. Note that all methods guarantee that the constraints on x and u (given in Table 4.2) are respected.

Figure 4.4 exhibits the achieved performances for the simulation run in terms

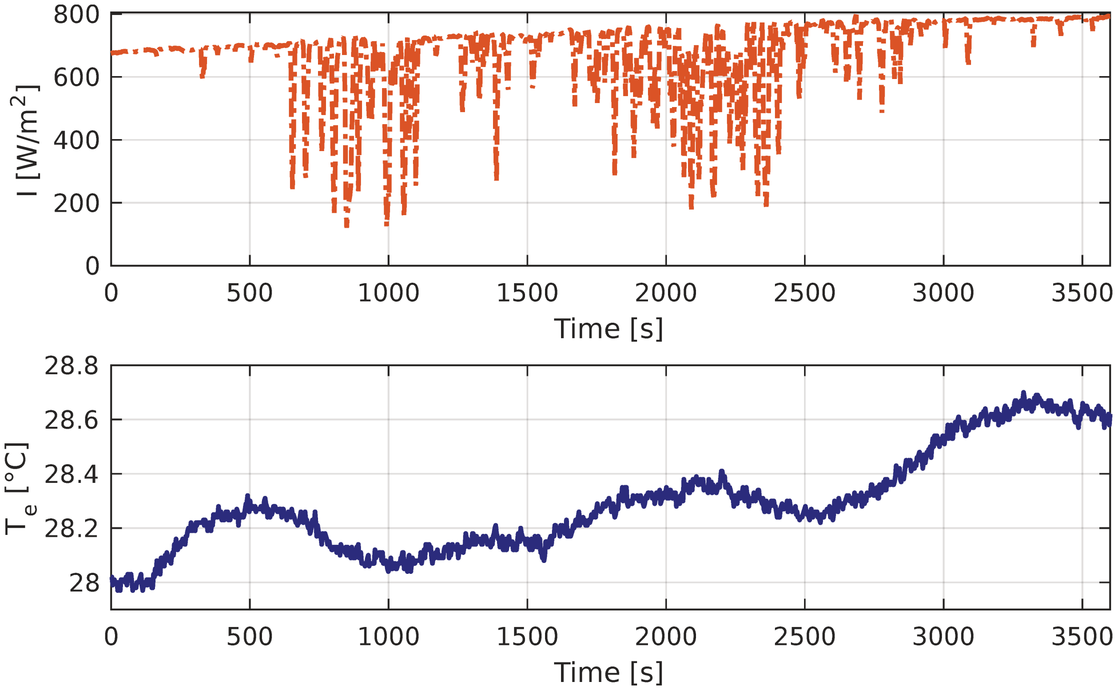


Figure 4.3: Disturbances $w(k)$: solar irradiance and external temperature.

of reference tracking and disturbance rejection, showing the evolution for the fluid and plate temperatures with the three controllers. All methods guarantee regulation and disturbances-to-state stabilisation. The respective applied control signal (fluid flow) is given in Figure 4.5.

As discussed, the MHE-MPC is able to find a better prediction model by adapting the LTI prediction model on-line through the polytopic variables μ_i, which are shown in Figure 4.6, along with the real values for μ_i.

On the one hand, it is evident that the MHE-MPC and qLPV-MPC methods are able to guarantee almost offset-free reference-tracking (within ±0.5 °C tracking error), despite the abrupt solar load disturbances, as it can be appreciated at $t = 700$ s and 2000 s in Figure 4.3. Note that the simulations consider a partially-cloudy day, because it is unreasonable to consider completely cloudy or nighttime periods due to the fact that the solar irradiance is the main energy source used by the process to heat the fluid (despite also being a disturbance). On the other hand, with respect to the sMPC method, as it holds an average model for prediction, it cannot yield almost offset-free reference-tracking and, moreover, it results heavily perturbed by the load disturbance (its rejection is very poor). Furthermore, it also experiments an operation region reduction which causes an infeasible start-point. However, to allow the comparison between these methods an admissible start-point was selected.

To further illustrate the results, Table 4.3 shows the *Integral Absolute Error* (IAE) index for the reference tracking and disturbance rejection results presented in Figure 4.4 (regarding x_2). Moreover, Table 4.3 also presents the *Total Variance* (TV) index for the three methods (see Appendix D). Bigger values for the TV index shows that more variation is applied to the control along the simulation.

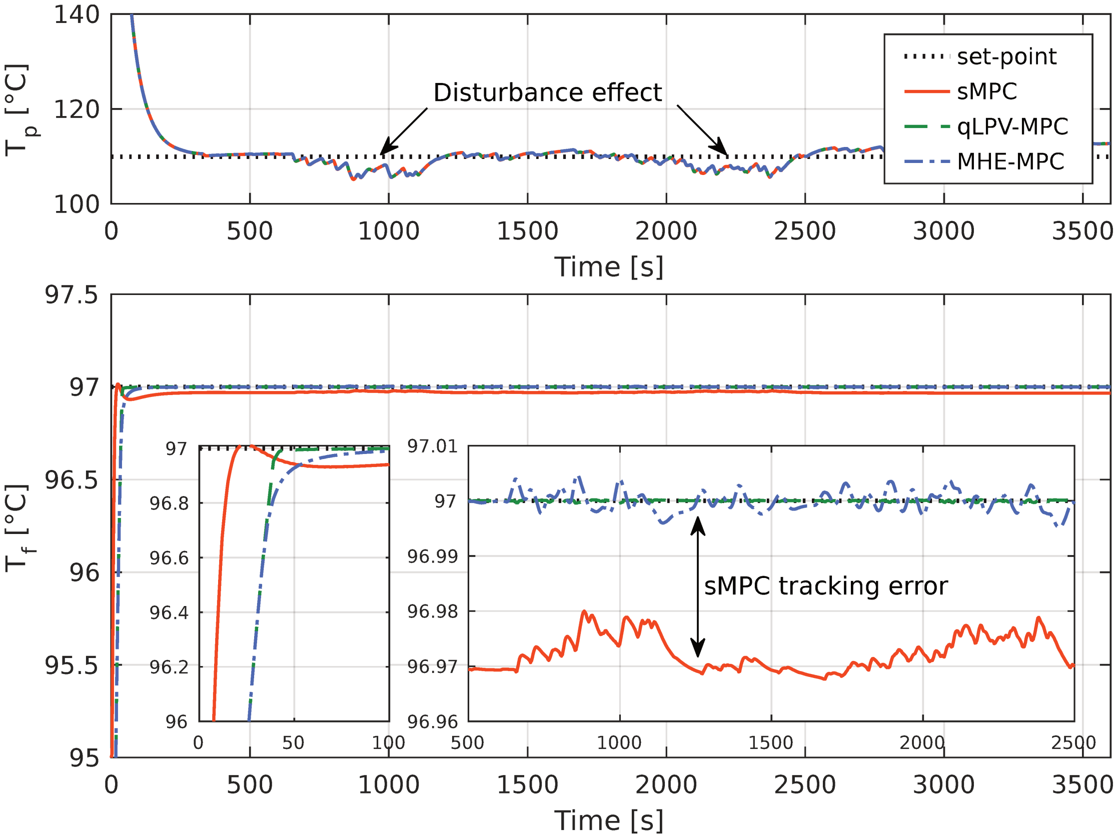


Figure 4.4: Fluid and plate temperature behaviours.

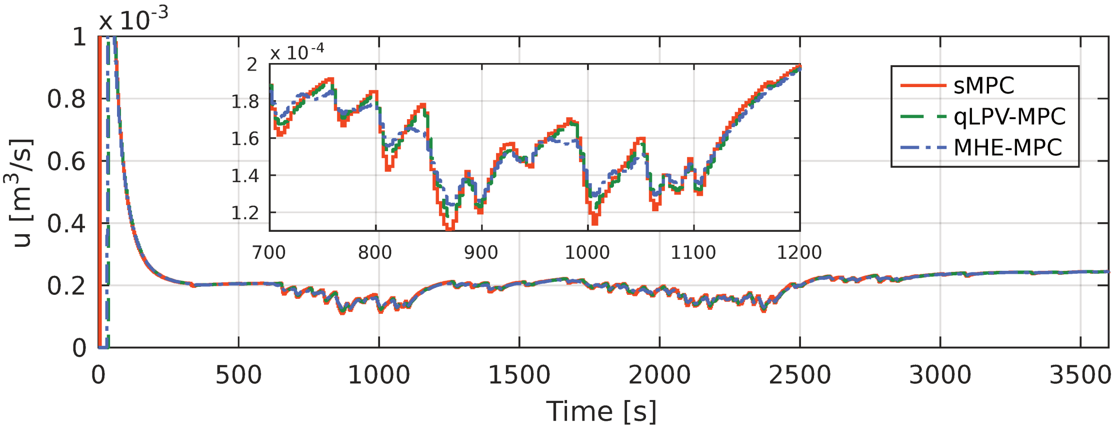


Figure 4.5: Control policies through the evolution of the fluid flow signal.

Therefore, values closer to zero indicate better (smoother) control strategies in terms of the use of the actuator. The analysis of the TV index is a very important issue from a practical point-of-view, since it means that the system actuators will have longer lifespan.

Clearly, the best results are obtained with the qLPV-MPC method: even without having the smallest TV value, which means that greater control effort was necessary, this tool enables a good tracking and disturbance rejection responses.

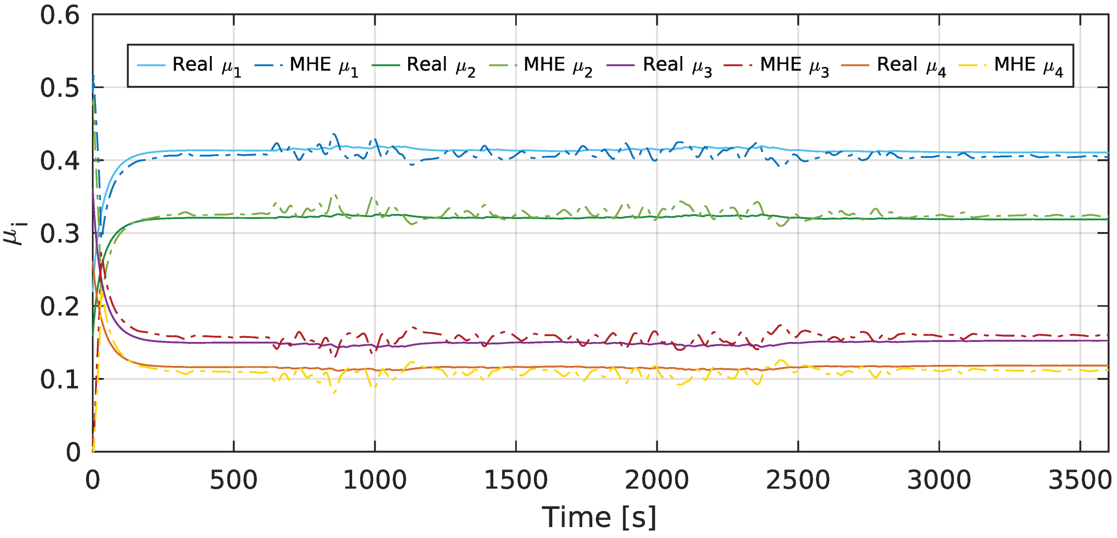


Figure 4.6: Evolution of the membership function (MHE-MPC method).

Table 4.3: Performance indices of the control methods.

	IAE tracking	**IAE rejection**	**TV**
MHE-MPC	80.1477	3.633	**0.0036** $\mathrm{m^3\,s^{-1}}$
qLPV-MPC	**76.9612**	**0.2058**	0.0046 $\mathrm{m^3\,s^{-1}}$
sMPC	–	95.8031	0.0081 $\mathrm{m^3\,s^{-1}}$

The IAE indices for this method are the ones closer to zero, which stands for excellent tracking/rejection. Note that the sMPC method has no value of IAE for tracking, it is due that this technique uses a different admissible start-point.

Besides, the Table 4.4 exposes the *On-line Computational Effort* (OCE) in terms of maximal, mean and minimal elapsed computational time measured as a percentage of the sample time[2] (see Appendix D). This table displays that sMPC requires less OCE, however all methods can be implemented for real-time purposes since the maximal elapsed time is less than the sample time. However, it is important to highlight that the qLPV-MPC method is performed using parallel computing tools, which allows an acceptable computation time (lesser than the T_s).

Table 4.4: Computation effort times with respect to the sample time.

	min	mean	max
MHE-MPC	0.7%	0.78%	**12.99%**
qLPV-MPC	70.76%	74.95%	99.41%
sMPC	**0.5%**	**0.6%**	48.28%

As a consequence, the quantitative results presented throughout Tables 4.3 and 4.4 expose that the better trade-off between performance and implementation complexity is offered by the proposed MHE-MPC method. That is, although

[2] In an i5-3337U CPU@2.7 GHz (2 Cores) with 6 GB of RAM.

the qLPV-MPC method has better IAE values, its application implies a heavy computational burden to solve their algorithms (i.e. QP plus sets calculation), so it becomes computationally unattractive. Moreover, the sMPC method cannot ensure stabilisation or recursive feasibility, since the average-weighted LTI model may not provide enough robustness under closed-loop (i.e. for some given conditions, it may run out of control). Anyhow, for practical purposes, the OCE needed to solve sMPC and MHE-MPC problems is roughly similar (although greater with the MHE-MPC). This issue can be critical for some real-time applications and must be taken into account by the designer. Nevertheless, the MHE-MPC takes almost the same time as the sMPC to be computed and performs much better.

Lastly, in the case of the MHE-MPC algorithm, it is interesting to decompose the total time elapsed for each of this two QP stages. With respect to this matter the Figure 4.7 exhibits the time consumed by the backward MHE and the forward MPC, and the total effort needed by the full MHE-MPC method.

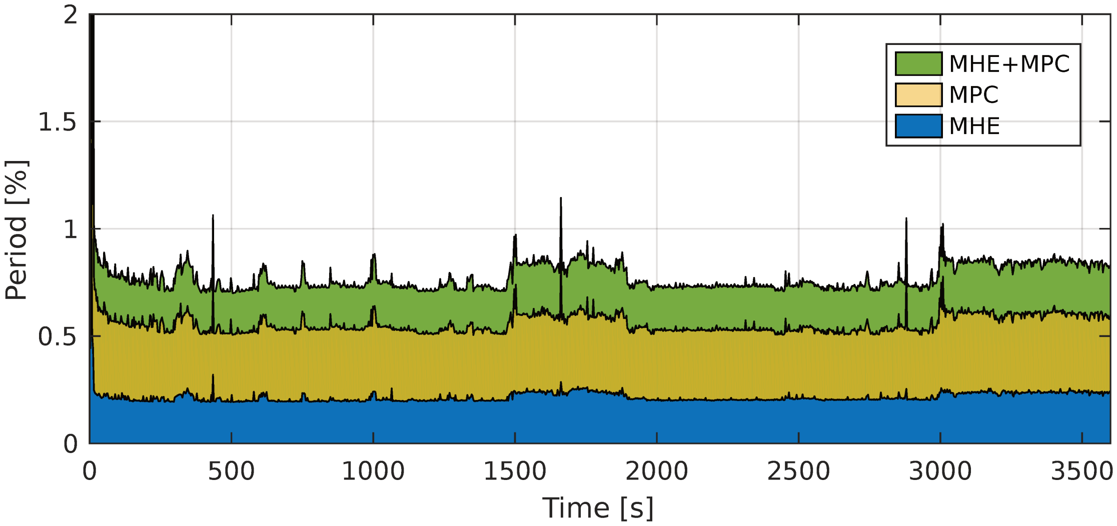


Figure 4.7: Period consumption.

4.6 Conclusions

The main result of this chapter is to present a novel state-feedback MPC strategy, based on an adaptive set-based approach for non-linear systems with LPV models. The adaptive method introduced in this work simplifies the non-linear scheduling into an LTI framework using a weighting variable for the prediction of the future system behaviour within the horizon. This adaptation variable is optimised on-line so as to minimise model-process mismatches. Thus, these results are undoubtedly strong, since the algorithm can guarantee feasibility, despite having model-process mismatches. There was no need for the non-linear programming solution which would have otherwise been set without the simplifications provided with the proposed method. Synthetically, a new, effective QP-based adaptive method for non-linear systems was developed in this chapter.

Then, with the purpose of highlighting the behaviour of the proposed strategy a realistic ST collector process simulation, comparing the proposed MPC to other techniques from the literature (qLPV-MPC and sMPC), was addressed. Consequently, these simulations results confirm the performance and effectiveness of the proposed scheme over non-linear systems in the presence of external disturbance. Some discussions regarding the results, the design procedure and the computational effort were presented.

For further works, parameter-dependent constraint and terminal set will be included into the MPC design procedure so that a larger domain of attraction could be obtained.

At last, in order to enable the reader to replicate and enhance the results shown here, the scripts used are available on a repository[3] under GPL-v3.0 license.

[3] https://github.com/ebernardi/LPVMPC

Chapter 5

Fault-Tolerant Linear Parameter Varying Model-based Predictive Control

In recent years, FTC has become a relevant research field and has attracted significant attention because of its applicability to industrial processes, which increases their security and reliability. Consequently, this chapter presents a model-based strategy for fault-tolerance in non-linear chemical processes. Specifically, the techniques addressed throughout Chapters 3 and 4 were integrated to develop an AFTCS strategy, which allows to track a reference even in the presence of actuator and sensor faults. Moreover, as previously stated, the observers convergence and the controller stability were guaranteed in terms of LMI problems. Finally, a numerical simulation based on a typical chemical industrial process (the highly non-linear CSTR) shows that the proposed method can achieve satisfactory performance in fault-tolerance.

It should be noted that the main results of this chapter can be found in works [8, 123, 124].

5.1 Introduction

Nowadays, control systems have evolved into sophisticated algorithm-based strategies. These advanced techniques allow high performance even to control unstable systems, optimising costs and control efforts. Nevertheless, in the event of unexpected scenarios or situations, which are not typically considered in the controller design process, the problem of achieving a suitable performance and stability requires a different strategy rather than just having a robust controller. As previously explained, this challenge has motivated the AFTCS strategy [2, 8]. This approach is assimilated as a variable structured technique because the controller is reconfigured when a fault occurs, so the stability and an acceptable system performance can be maintained.

As could be expected, the appropriate behaviour of an AFTCS depends on a solid FDD module to provide an early detection and sizing of faults, which can avoid system shutdown, breakdowns, and even worse, the occurrence of potential catastrophes. In that sense, previous works show that the modular approach (FDD and controller designed separately) presents its benefits, being this more flexible for practical applications and, therefore, easier to test and implement [2].

For the above mentioned reasons, this chapter presents the design of a nonlinear fault-tolerant MPC-based strategy. Thus, the structure of the proposed AFTCS is presented in Figure 5.1. There, the FDD stage uses banks of generalised observers to detect, isolate and estimate multiples faults. In particular, groups of LPV-RUIO and LPV-UIOO are used. Subsequently, this information is delivered to a reconfiguration mechanism that compensates the controller input data, in order to achieve an acceptable post-fault system performance. On the other hand, the selected controller, introduced in chapter 4, uses an MHE technique to update the MPC internal model at each iteration time, enhancing the controller fault-tolerance capabilities.

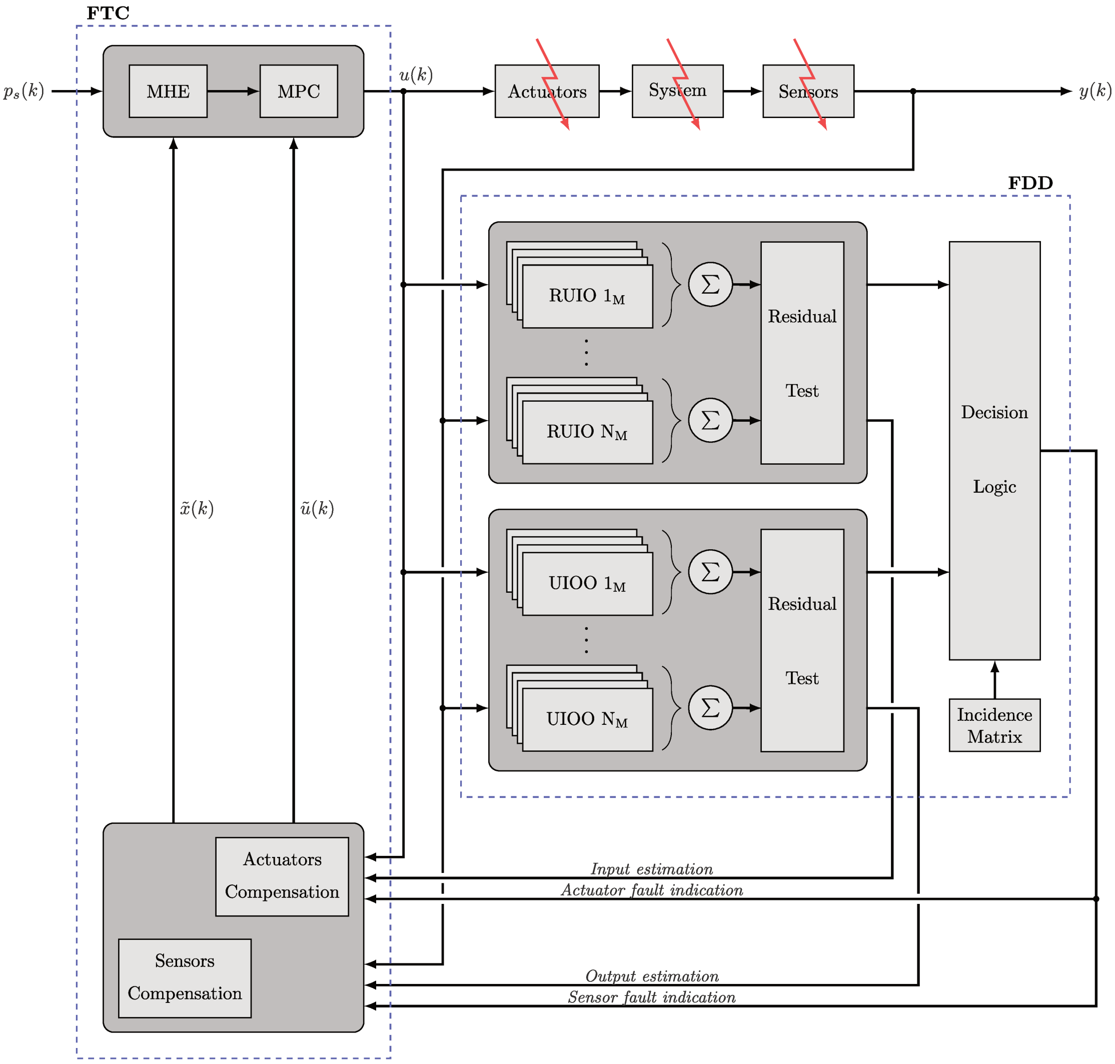


Figure 5.1: Fault-tolerant MPC scheme.

Remark 5.1. *Note that the main goal of this chapter is to build an AFTCS* *strategy. To do that, some concepts from the preceding chapters need to be recalled. As a consequence, previous concepts and formulations are specifically evoked with the purpose to create a clean procedure and expose a structured AFTCS design framework.*

5.2 Model statement

Recalling the discrete-time LPV model with additive input and output faults, from Equation (3.1). That is,

$$\begin{aligned} x(k+1) &= \sum_{i=1}^{M} \mu_i\left(\zeta(k)\right)\left\{A_i x(k) + B_i u(k) + F_i f_u(k) + \Delta x_i\right\} \\ y(k) &= \sum_{i=1}^{M} \mu_i\left(\zeta(k)\right)\left\{C_i x(k) + D_i u(k) + H_i f_y(k) + \Delta y_i\right\} \end{aligned} \tag{5.1}$$

where $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$, $f_u(k) \in \mathbb{R}^q$, $y(k) \in \mathbb{R}^p$, $f_y(k) \in \mathbb{R}^o$ are the state vector, the input vector, the fault input vector, the output vector and the fault output vector, respectively. Moreover, A_i, B_i, C_i, D_i, F_i, H_i, Δx_i and Δy_i are constant matrices of appropriate dimensions.

Furthermore, without loss of generality the Assumptions 2 to 4 are considered true, then the development of a novel AFTCS strategy is addressed below.

5.3 Fault detection and diagnosis

Bearing in mind the previous statements, the Sections 5.3.1 and 5.3.2 briefly present the design procedure for the discrete-time case of the LPV-RUIO and LPV-UIOO, respectively. Notice that a detailed version which explains these techniques step-by-step was introduced in Chapter 3.

5.3.1 Design of the LPV-RUIO

Firstly, if the term $Hf_y(k) = 0$ is considered (no sensors faults occur in the system), and under the assumption that the rank of $F_i = q$ (being F_i a specific column of B_i, that corresponds with the faulty input to diagnose), then, after performing some algebraic operations, the system from Equation (5.1) is transformed into a new LPV system without unknown inputs,

$$\begin{aligned} \bar{x}_1(k+1) &= \sum_{i=1}^{M} \mu_i(\zeta(k))\left\{\tilde{A}_{i_1}\bar{x}_1(k) + E_{i_1}y(k) + \bar{B}_{i_1}u(k) + \bar{\Delta x}_{i_1}\right\} \\ y(k) &= \sum_{i=1}^{M} \mu_i(\zeta(k))\left\{\tilde{C}_{i_1}\bar{x}_1(k)\right\} \end{aligned}$$

with $\tilde{C}_{i_1} = CN_i$, $\tilde{A}_{i_1} = \bar{A}_{i_{11}} - \bar{A}_{i_{12}}U_{i_1}CN_i$ and $E_{i_1} = \bar{A}_{i_{12}}U_{i_1}$.

At this point, if the pair $(\tilde{A}_{i_1},\ \tilde{C}_{i_1})$ is observable, it is possible to design a reduced-order observer for (5.2) as,

$$\Phi(k+1) = \sum_{i=1}^{M} \mu_i(\zeta(k))\left\{K_i\Phi(k) + \bar{B}_{i_1}u(k) + L_i^{*}y(k) + \bar{\Delta} x_{i_1}\right\} \tag{5.3}$$

where $\Phi(k) \in \mathbb{R}^{(n-q)}$ is the observer state vector, $L_i^{*} = L_i + E_{i_1}$ and $K_i = \tilde{A}_{i_1} - L_i\tilde{C}_{i_1}$. Moreover, the matrices $L_i \in \mathbb{R}^{(n-q)\times(p-q)}$ are the observer gains to be designed.

Consequently, the necessary existence conditions are:

1. $\tilde{A}_{i_1}$ is asymptotically stable.
2. $(\tilde{A}_{i_1},\ \tilde{C}_{i_1})$ is observable.
3. C and F_i are full row and column rank, respectively.
4. $\mathrm{rank}(CF_i) = \mathrm{rank}(F_i)$

Accordingly, through the Theorem 3.2 the sufficient LMI conditions for the LPV-RUIO synthesis, were presented.

Thereby, from Equation (5.3) with $\Phi(k) \to \hat{\bar{x}}_1(k)$ pursuant to $k \to \infty$. That is,

$$\begin{aligned} \hat{x}(k) &= \sum_{i=1}^{M} \mu_i(\zeta(k))T_i\hat{\bar{x}}(k) \\ &= \sum_{i=1}^{M} \mu_i(\zeta(k))T_i \begin{bmatrix} \Phi(k) \\ U_{i_1}y(k) - U_{i_1}CN_i\Phi(k) \end{bmatrix} \end{aligned}$$

with $\hat{x}(k) \to x(k)$, with $k \to \infty$.

5.3.2 Design of LPV-UIOO

On the other hand, considering that the matrix H corresponds with a row of C, which is a non-monitored output (faulty sensor), and choosing a transformation matrix T_2, such that $J = T_2C$, where $J \in \mathbb{R}^{p-o}$ only contains the C rows corresponding with the monitored outputs (non-faulty sensors). Thus, the output equation of the system model from Equation (5.1) is transformed to,

$$\tilde{y}(k) = Jx(k).$$

Thereby, an UIO with the purpose of estimating the state variables $x(k)$, even with unknown input presence, is formulated as,

$$\begin{aligned} z(k+1) &= \sum_{i=1}^{M} \mu_i(\zeta(k))\left\{N_iz(k) + G_iu(k) + L_i\tilde{y}(k) + \Delta z_i\right\} \\ \hat{x}(k) &= z(k) - E\tilde{y}(k) \\ \hat{y}(k) &= C\hat{x}(k) \end{aligned}$$

where $z(k)$ is the full-order observer state vector, and the matrices $N_i \in \mathbb{R}^{nxn}$, $G_i \in \mathbb{R}^{nxp}$, $L_i \in \mathbb{R}^{nxp}$ and $E \in \mathbb{R}^{nxp}$ are the observer gains to be determined. In addition, the weighting functions $\mu_i(.)$ are the same as used in Equation (5.1).

Thus,

$$e(k+1) = \sum_{i=1}^{M} \mu_i(\zeta(k)) N_i e(k)$$

if the following constraints hold

$$\begin{aligned}
&T_1 F_i = 0\\
&G_i - T_1 B_i = 0\\
&N_i T_1 + L_i J - T_1 A_i = 0\\
&\Delta z_i - T_1 \Delta x_i = 0
\end{aligned}$$

being $T_1 = (\mathbb{I} + EJ)$. In this way, if N_i is quadratically Schur, $e(k+1) \to 0$ asymptotically.

As a consequence, the necessary existence conditions are:

1. A_i is asymptotically stable.
2. (A_i, J) is observable.
3. J and F_i are full row and column rank, respectively.
4. $\mathrm{rank}(JF_i) = \mathrm{rank}(F_i)$.

Accordingly, through the Theorem 3.4 the sufficient LMI conditions for the LPV-UIOO synthesis, were presented.

Remark 5.2. *The Theorems 3.2 and 3.4 show that the observers designs are solved through LMI problems. For that, the Yalmip [88] and LMI Lab [65] packages of Matlab software are used.*

5.3.3 Detection and diagnosis scheme

As presented in Section 3.5, the residual pattern should be customised to follow a certain structure (*generalised residuals*). Thus, the structured residuals are characterised by selective fault responses. In other words, any residual is sensitive to one group of faults and insensitive to others.

Fault detection and isolation

Taking all the above into account, a threshold test may be performed separately for each residual, yielding to a decision incidence matrix, and then the isolation task is fulfilled using this matrix.

Fault estimation

When the fault detection and isolation tasks are successfully achieved, the actuator or sensor fault estimation is validated.

Actuators Hence, the actuator fault estimation task is performed using the corresponding LPV-RUIO to the associated fault (selected as unknown input signal).

Sensors Besides, the sensor fault estimation task is performed using the k-th non-monitored residue component from the corresponding LPV-UIOO to the associated fault (selected as non-monitored signal).

5.4 Fault-tolerant controller

The aim of this section is to develop an FTC to abide actuators and sensors faults. This strategy looks for, at each sampling time k, the best prediction model for the next N_p steps (based on the previous N_e steps) and compute the near-optimal sequence of control actions using some terminal ingredients to guarantee stability. As a consequence, this *Fault-Tolerant Model Predictive Control* (FT-MPC) strategy is divided into two QP problems: the backward MHE QP and the forward MPC QP. Notice that this FT-MPC is based on the MHE-MPC one, which was presented in detail in Chapter 4.

5.4.1 Backward QP - The MHE

As previously stated, the backward QP is used for *identification* purposes. The basic idea behind this, is to consider the interpolation variables μ_j to compose the best LTI combination model to be used to momentarily describe the controlled process. This procedure minimises the model discrepancy with respect of μ and its variance throughout that time. This is achieved with the solution of the following optimisation problem, from $k = k_0$, considering $\tilde{x} = x - x_f$ and $\tilde{u} = u + u_f$ as the fault-compensated input and output data, and μ_{k-1} as the result from the previous iteration:

$$\begin{aligned}
\min_{\mu} \quad & \sum_{j=k-N_e+1}^{k} \left(e(j)^T Q_e e(j) + \nu_\mu^T Q_\nu \nu_\mu \right) \\
\text{s.t.} \quad & e(j+1) = x(j+1) - (Ax(j) + Bu(j)), \\
& A = \sum_{i=1}^{M} \mu_i A_i \quad \text{and} \quad B = \sum_{i=1}^{M} \mu_i B_i, \\
& \sum_{i=1}^{M} \mu_i = 1, \qquad 0 \le \mu_i \le 1, \\
& \mu = \operatorname{col}\{\mu_i(k)\}, \quad \mu = \mu_{k-1} + \nu_\mu,
\end{aligned} \tag{5.5}$$

where $j \in \mathbb{Z}_{k-N_e:k-1}$ and $i \in \mathbb{Z}_{1:M}$. Moreover, N_e is the estimation horizon and the matrices Q_e and Q_ν are the appropriate dimension tuning weights of this optimisation procedure, considered as identity matrices.

5.4.2 Forward QP - The MPC

On the other hand, the forward QP is capable of obtaining a control law that takes into consideration constraints on the states, inputs and outputs. For that, the proposed cost function of the FT-MPC optimisation problem is given by:

$$V_N(\tilde{x}, x_s, \mu; \mathbf{u}, x_a, u_a) = V_N^d(\tilde{x}, \mu; \mathbf{u}, x_a, u_a) + V_N^t(x_s; x_a)$$

with

$$V_N^d(\tilde{x},\mu;\mathbf{u},x_a,u_a)=\sum_{j=0}^{N_p-1}\|\tilde{x}(j)-x_a\|_Q^2+\|\tilde{u}(j)-u_a\|_R^2,$$

where $Q\in\mathbb{R}^{n\times n}$ and $R\in\mathbb{R}^{m\times m}$ are positive definite matrices, and N_p is the prediction horizon. The couple (x_a,u_a) is the artificial reference, which represents the best reachable steady state by the system in N_p steps, starting from the initial condition $\tilde{x}=\tilde{x}(k)$. At last, $V_N^t(x_s;x_a)$ is the end cost function, which is defined

$$V_N^t(x_s;x_a)=\|x_a-x_s\|_\varphi^2+\|\tilde{x}(N_p)-x_a\|_P^2.$$

being φ and P the appropriates weighting matrices. Then the term $\|\tilde{x}(N_p)-x_a\|_P^2$ is an offset that penalises the final-state deviation from this target operation and the offset term $\|x_a-x_s\|_\varphi^2$ ensures that the artificial variable tracks the real set-point variable, with the actual target goal.

Thus, for any current compensated state $\tilde{x}$, the optimisation problem to be solved at each time step $k=k_0$ is given by

$$\begin{aligned}\min_{\mathbf{u},x_a,u_a}\quad & \sum_{j=0}^{N_p-1}V_N(\tilde{x},x_s,\mu;\mathbf{u},x_a,u_a),\\ \text{s.t.}\quad & x(0)=\tilde{x},\\ & x(j+1)=Ax(j)+B\tilde{u}(j),\\ & x_a=Ax_a+Bu_a,\quad (u_a,x_a)\in\mathbb{Z}_s=\mathbb{X}_s\times\mathbb{U}_s\\ & A=\sum_{i=1}^{M}\mu_iA_i\ \text{ and }\ B=\sum_{i=1}^{M}\mu_iB_i,\\ & x(j)\in\mathbb{X},\ \tilde{u}(j)\in\mathbb{U},\quad \forall j\in\mathbb{Z}_{0:N_p-1}\\ & x(N_p)=x(N_p)\in\mathbb{X}_f,\end{aligned}\tag{5.6}$$

considering that $\mu_\mathbf{i}$ represents the value obtained by the backward MHE QP for μ. Additionally, $\mathbb{X}_f$ is an adequate robust controlled positively terminal invariant set that contains the set-point $p_s=(x_s,u_s)$ (see Chapter 4).

Remark 5.3. *Note that this fault-tolerant predictive controller uses a feedforward compensation of actuator faults and a compensated version of the measurable outputs to compute the fault-tolerant control policy (reconfiguration mechanism).*

Remark 5.4. *The stability proof of the system from Equation* (5.1) *in closed-loop with the proposed fault-tolerant controller is equivalent to the one presented in Chapter 4 for the nominal MHE-MPC. It is due to the convergence of the compensated variables $\tilde{x}$ and $\tilde{u}$ depends purely from the FDD module, which is assumed to be fast enough.*

5.5 Illustrative example (CSTR)

Recalling the CSTR process, presented in Section 3.7, where its non-linear model is described in Equation (3.43) and its physical and operational parameters are

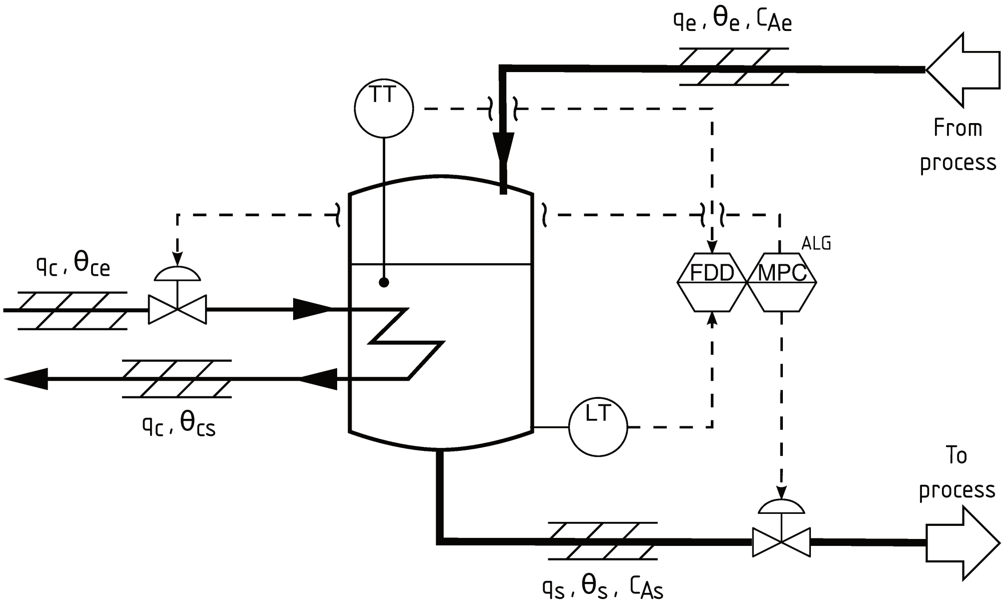


Figure 5.2: Diagram of the AFTCS strategy on a CSTR process.

presented in Table 3.2. Moreover, the application of the proposed AFTCS scheme on the CSTR process is depicted in Figure 5.2.

Notice that, according to this figure, the state variables V and C_A are controlled by the process flow rate q_s and the coolant flow rate q_c, respectively. It should be clarified that the state variable C_A is controlled indirectly from the state T.

5.5.1 AFTCS design

As was introduced in Section 5.2 the first step consists on the selection of a scheduling parameter vector (in this case $N = 2$) which depends on the outputs, i.e. $\zeta(t) \coloneqq \begin{bmatrix} V(t) & T(t) \end{bmatrix}$. Then, using the PJL technique (see Appendix B), the nonlinear system from Equation (3.43) was rewritten as a discrete-time LPV model, which considered additive input and output faults. Specifically, to obtain a proper model representation, $L = 3$ linearisation points per parameter was used. As a consequence $M = L^N = 9$ weighting functions $\mu_i(\cdot)$ corresponding to each affine model were derived. Next, following the same procedure as Chapter 3, the membership functions corresponding to each parameter were selected.

FDD design

According to the design method formulated in Section 5.3, the suitable observer matrices and functions are developed. To do that, a brief explanation for each subsystem is provided as follows.

LPV-RUIO On the one hand, setting $F_i = \begin{bmatrix} -1 & 0 & 0 \end{bmatrix}^T$ and solving the LMI from Equation (3.17), the matrices to construct the observer to estimate faults over the valve q_s, were designed. On the other hand, defining $F_i = \begin{bmatrix} 0 & 0 & B_{32} \end{bmatrix}^T$, and repeating the previous procedure, the observer to estimate faults on the valve q_c was constructed.

LPV-UIOO As previously stated, the existing hard correlation over the states C_A and T hinder the correct fault detection and isolation task design. As a consequence, firstly the output V was selected as a non-monitored output (H_i corresponds with the V row of the output matrix C_i), and $F_i = \begin{bmatrix} -1 & 0 & 0 \end{bmatrix}^T$ was defined, then the appropriate T_2 matrix was chosen (null space of H_i). Thus, it was possible to construct the matrices of the first observer that was used to diagnose faults of the output sensor V. Besides, choosing the output T as non-monitored output, defining $F_i = \begin{bmatrix} -1 & 0 & 0 \end{bmatrix}^T$, and selecting another T_2 matrix, another observer was constructed to diagnose faults of the output sensor T.

Controller design

On the other hand, based on the strategy presented in Section 5.4 the FTC is developed. To do that, firstly $Ne = N_p = 8$ was used. Secondly, the constraints $q_s \in [90, 110], q_c \in [90, 105]$, $V \in [90, 110]$, $C_A \in [0.03, 0.17]$, $T \in [440, 450]$ and the corresponding sets $\mathbb{U}$ and $\mathbb{X}$ were defined (Hardware constraints). Then, the weighting matrices $Q_e = 1 \times 10^5$, $Q_\nu = 1$, $Q = \text{diag}\{0\ ,\ 1 \times 10^4\}$ and $R = 1 \times 10^1$ were settled. Lastly, the optimisation problems from Equations (5.5) and (5.6) were properly deployed, using computer-aided tools.

5.5.2 Numerical simulation

To evaluate the performance and effectiveness of the proposed AFTCS strategy, it was simulated on the non-linear CSTR system model, from Equation (3.43), during 3 h and using $T_s = 3\,\text{s}$ (requirement imposed by the available hardware). In particular, to show its behaviour, a group of set-point changes were made from the initial condition $V^0 = 100\,\text{m}^3$, $T^0 = 447\,\text{K}$.

1. Firstly, a step change to $V = 98\,\text{m}^3$ and $T = 445\,\text{K}$ was applied at the start, i.e. $t = 0\,\text{min}$.
2. Then, an increment of $V = 12\,\text{m}^3$ was added at $t = 7\,\text{min}$.
3. At $t = 25\,\text{min}$ an abrupt displacement up to $V = 95\,\text{m}^3$, $T = 442\,\text{K}$ was introduced.
4. Another step change was applied at $t = 45\,\text{min}$ to $V = 98\,\text{m}^3$, $T = 445\,\text{K}$.
5. Finally, when $t = 65\,\text{min}$, the volume was carried up to $V = 91\,\text{m}^3$, keeping $T = 445\,\text{K}$.

Furthermore, with the purpose of exposing the fault-tolerance capacity, the following faults were considered:

1. since 10 min up to 20 min an exponential accumulation of tartar in the sensor of T, inducing an error of up to 0.5% from $T_{\max}$,
2. from 30 min to 40 min a leak in valve q_s coupling, producing a loss of 5% from $q_{s_{\max}}$,
3. at 50 min to 60 min an exponential calibration error in the V sensor of up to 2% from $V_{\max}$,

4. an exponential clog in valve q_c of up to 5% from $q_{c_{\max}}$ income between 70 min and 80 min.

It is important to note that, the high slopes in these faults and set-point changes were chosen to illustrate worst-case conditions and expose the capacity of the proposed AFTCS strategy to tolerate different types of faults in different situations. Furthermore, to create an even more realistic simulation, white measurement noises were added.

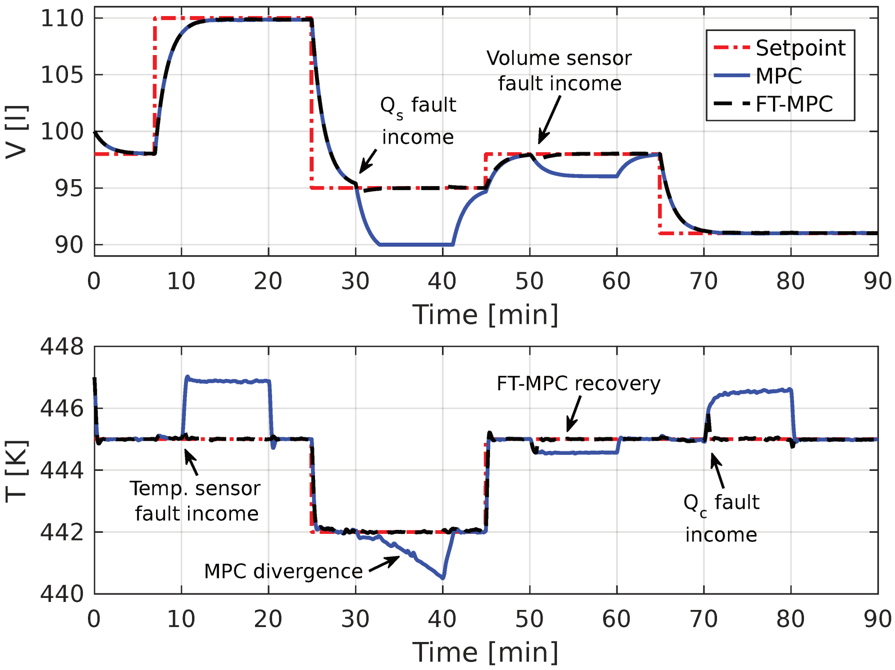


Figure 5.3: CSTR outputs.

Figures 5.3 to 5.5 show the simulation results when faults occur. The behaviour of the proposed FT-MPC and the nominal MPC are depicted in Figure 5.3. There, it is possible to appreciate the convergence capacity, at the beginning. Additionally, it is easy to see the fault occurrence, and the ability of the FT-MPC scheme to regulate the state variables to their desired values x_s, after the occurrence of faults. Specifically, from 10 min to 20 min it is noticeable how an accumulation of tartar in the temperature sensor leads to a significant error in the achievement of the temperature set-point, but it does not affect the tank volume. Furthermore, between 30 min to 40 min it is possible to see how a leak in valve q_s may conduct the nominal controller to instability. Instead, the superior performance of the FT-MPC scheme is able to guarantee stability and even a fault compensation. Lastly, the respective applied control signals are given in Figure 5.4, where the compensation efforts to mitigate the faults effects are shown.

On the other hand, the Figures 5.5a and 5.5b show the actuators and sensors faults estimation, respectively. There, the effects introduced due to the use of a detection strategy based on threshold is appreciated. As well as the detection and estimation errors, all of these of small magnitude.

For the sake of completeness, Table 5.1 shows the IAE and the *Integral of*

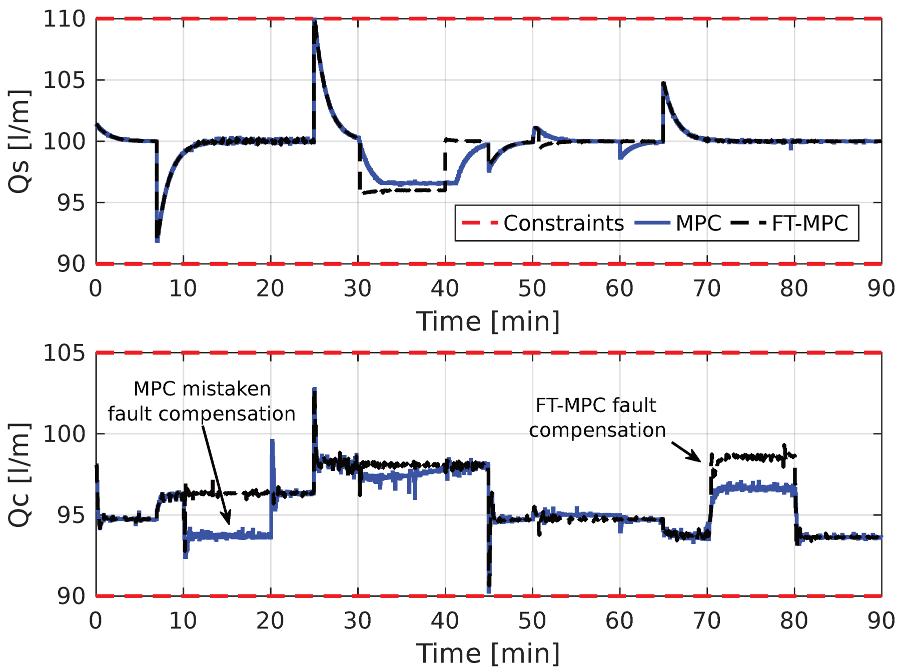
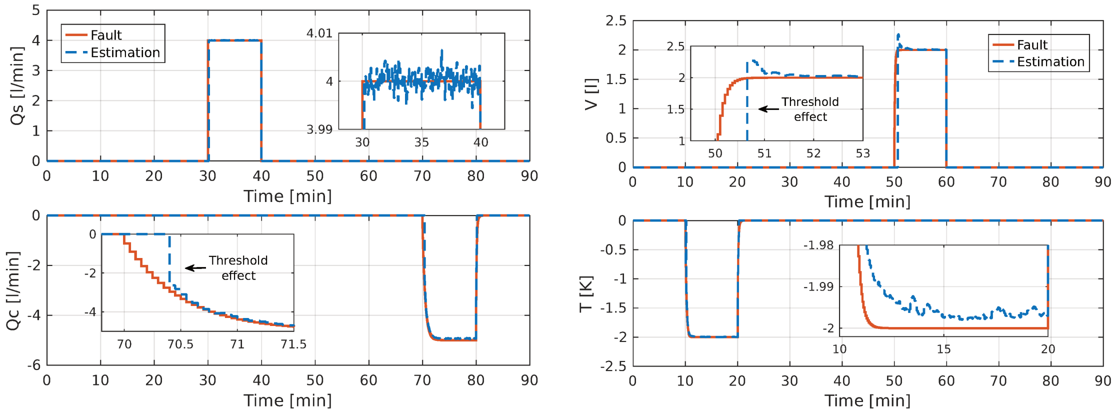


Figure 5.4: CSTR inputs.



(a) Actuator fault estimation.

(b) Sensor fault estimation.

Figure 5.5: CSTR fault estimation.

Squared Error (ISE) indices results (regarding C_A). Furthermore, Table 5.1 also presents the TV index for these methods (regarding q_c). Clearly, the best performance results are obtained with the FT-MPC method, even concerning TV values. In other words, this strategy combines a good tracking and disturbance rejection responses, even against faults.

Table 5.1: Performance indices of the control methods.

	IAE	**ISE**	**TV**
FT-MPC	**0.52**	**0.005**	**125.87** $\mathrm{l\,h^{-1}}$
MPC	0.656	0.009	134.091$\mathrm{l\,h^{-1}}$

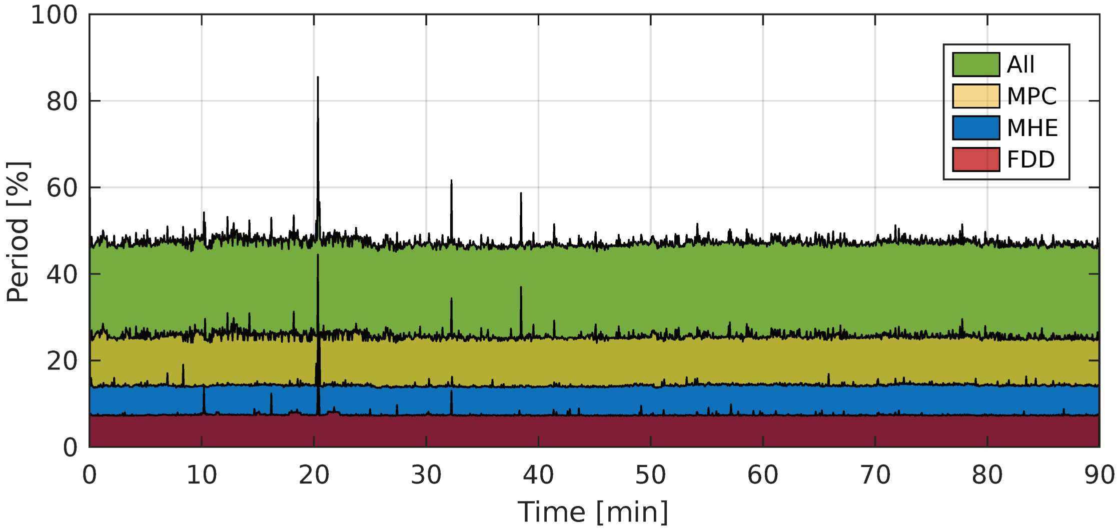


Figure 5.6: Period consumption.

Besides, the Figure 5.6 exposes the OCE[1] in terms of MHE, MPC and FDD modules, as well as all together (see Appendix D). Exposing that the proposed fault-tolerant method composes a computationally tractable algorithm.

5.6 Conclusion

This chapter presents the design of a novelty AFTCS to endure actuators and sensors faults upon non-linear industrial processes. For this purpose, an observer-based FDD scheme has been incorporated. In particular, a bank of generalised observers was implemented. Moreover, an MPC strategy has been used to maintain a desired fault-tolerance feature in the proposed AFTCS approach. The controller formulation, which is compound of two QP problems (MHE and MPC), is able to regulate the state variables for a satisfactory control recovery.

In addition, this AFTCS strategy was tested against a nominal controller. Simulation results are promising and illustrate the excellent performance of the presented approach.

At last, in order to enable the reader to replicate and enhance the results shown here, the scripts used are available on a repository[2] under GPL-v3.0 license.

[1] In an i5-3337U CPU@2.7 GHz (2 Cores) with 6 GB of RAM.

[2] `https://github.com/ebernardi/FT-LPVMPC`

Chapter 6

Fault-Tolerant Energy Management System

This chapter presents an optimisation-based control method to develop a fault-tolerant EMS for an industrial microgrid, based on a sugarcane power plant. The studied microgrid has several renewable energy sources, such as photovoltaic panels, wind turbines and biomass power generation, being subject to different operational constraints and load demands. The proposed management policy guarantees that these demands are met at every sampling instant, despite eventual faults. This law is derived from the solution of an optimisation problem that combines the formalism of an MHE scheme (to estimate faults) and an MPC loop (for fault-tolerant control goals); it chooses which energy source to use, seeking maximal profit and increasing sustainability. The predictive controller part of the scheme is based on a linear time-varying model of the process, which is scheduled with respect to the fault estimation brought up by the MHE. Furthermore, a realistic numerical simulation confirms the effectiveness and performance of the proposed control method.

Finally, it should be noted that the main results of this chapter can be found in work [39].

6.1 Introduction

The improvements to reduce the effect of climate changes necessarily involve a sustainable development plan, for which, the replacement of fossil fuels by renewable energy sources seems like the evident path. An important aspect to consider is the efficiency of energy production. Consequently, future (large scale) energy projects will have to include strategies to integrate the *renewables* coherently, with appropriate management policies which guarantee uninterrupted and efficient energy generation.

The smart-grid/microgrid concept has come to focus under this scope; these technologies allow the efficient use and generation of energy, intelligently managing the available energy sources together with storage units to increase the reliability of the renewable systems. Thus, the core idea of the concept is to “shift” electric load demands to periods when the *renewables* are generating energy, avoiding their intermittent characteristic, see [125, 126]. In addition, with the inclusion of intermediate storage units coupled to the energy generation (batteries, supercapacitors, fly-wheels and others), the excessive energy produced at a given instant can be stored to compensate, at a future moment, lacking production. As proposed in [127], a microgrid is henceforth understood as a set of generators, loads and storage units concatenated together as a single system. Note that many experimental implementations of optimised renewable smart-grids have been seen through the literature, such as in [128, 129].

Hence, the energy management problem of microgrids is a significant issue to be studied in order to allow an optimal and eco-friendly operation of future energy matrices. For such, MPC schemes have been successfully applied to a diverse set of recent renewable applications [130, 131, 132, 133, 134, among others].

However, as discussed in previous chapters, unexpected scenarios or unusual events sometimes occur, which are not typically considered in the standard MPC design process. As a consequence, the problem of guaranteeing load demands requires a different strategy rather than just having a robust controller.

This challenge has motivated several MPC-based AFTCS studies for microgrids. In terms of papers, some results are mentioned:

- A fault-tolerant predictive strategy is developed in [135, 136] to ensure the proper amount of energy in the storage devices is kept so that consumer demand is always covered. In these works, the fault term is included in the optimisation procedure, therefore finding a time-varying control policy that incorporates the variability of the energy plant's behaviour due to faults in the nominal process model;
- Based on a similar paradigm, in [137] the authors develop a predictive controller that mitigates the effects of faults in a microgrid based on a diagnosis level. The reconfiguration block acts to make changes on constraints as well as references passed to the MPC.
- Recently, in [97] an energy coordination AFTCS has been proposed. There the AFTCS was synthesised based on several diagnosis observers. The main drawback, as discussed therein, is that the approach was not unified with respect to fault estimation, the implementation complexity growing linearly according to the number of control inputs.

As a result, the integration of AFTCS requires a solid detection and an accurate estimation of the level of faults, so that smooth performance degradation can be obtained. It must be commented that many papers have detailed the ability of the MHE optimisation to be adapted to estimate faults (and level of faults) in a controlled process, as if it was embedded into some concentrated time-varying model parameter [81, 33]. Consequently, the MHE method is an interesting option to be directly combined to an FTC layer, since it shares the same usual quadratic, constrained optimisation formalism with MPC.

It seems very reasonable that both MPC and MHE could be combined to design an AFTCS, yet this has not been formally conducted in the literature for the case of renewable smart-grids. Therefore, it is the main motivation of this chapter, which aims in providing high-fidelity results of such kind of technique, demonstrating possible outcomes and advantages with respect to simpler EMSs (in which the effect of faults is not included).

6.2 Preliminaries: sugarcane microgrid

Brazil is a country with a large share of renewable energy (up to 43 % of its primary matrix; the average in the rest of the world is 15 %); the sugarcane industries play an important role in this scenario: their contribution reaches 40 % of the renewable share (with bio-ethanol production and bagasse burning).

These Brazilian sugarcane plants are indeed, renewable microgrids, since they use the sub-products of the sugarcane processing (bagasse, straw and vinasse) as sources to generate electric energy, steam and cold water, as schematised by Figure 6.1. The three outputs are used to sustain the needs of the sugar/ethanol production process[1]. In this work, following the lines of [129], mutatis mutandis, auxiliary renewable sources are coupled to this system (wind turbine, solar panels) so that it can generate energy throughout the whole year (even out of the harvest period) and sell the exceeding energy to the local *Distribution Network Operator* (DNO).

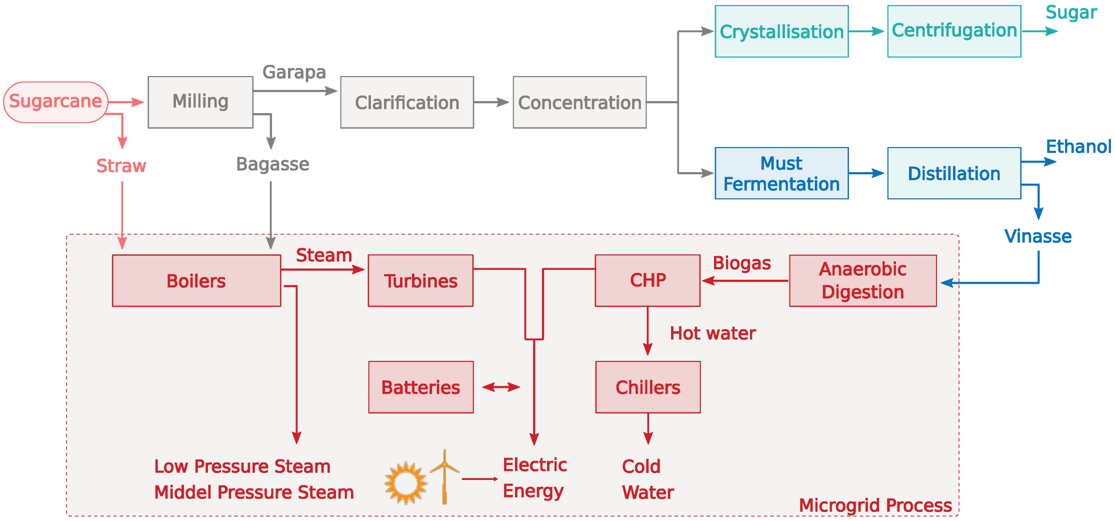


Figure 6.1: Sugarcane industry process.

The particular plant studied in this chapter is composed of the following subsystems, as illustrated in Figure 6.2: two boilers, with different efficiencies; two steam turbines, with different efficiencies too; a *Combined Heat and Power* (CHP) system; a water chiller; a hot water tank; photovoltaic panels; *Water Heating Solar* (WHS) panels; a wind turbine; two pressure reduction valves; an HE; stocks of bagasse, straw and compressed biogas, and a battery bank. The use of intermediate storage units (battery bank and biomass stocks) allows the system to

[1] Steam is used to boost shredders, spray pumps and other equipment; cold water is due to internal refrigeration needs; electric energy guarantees the operation of the process.

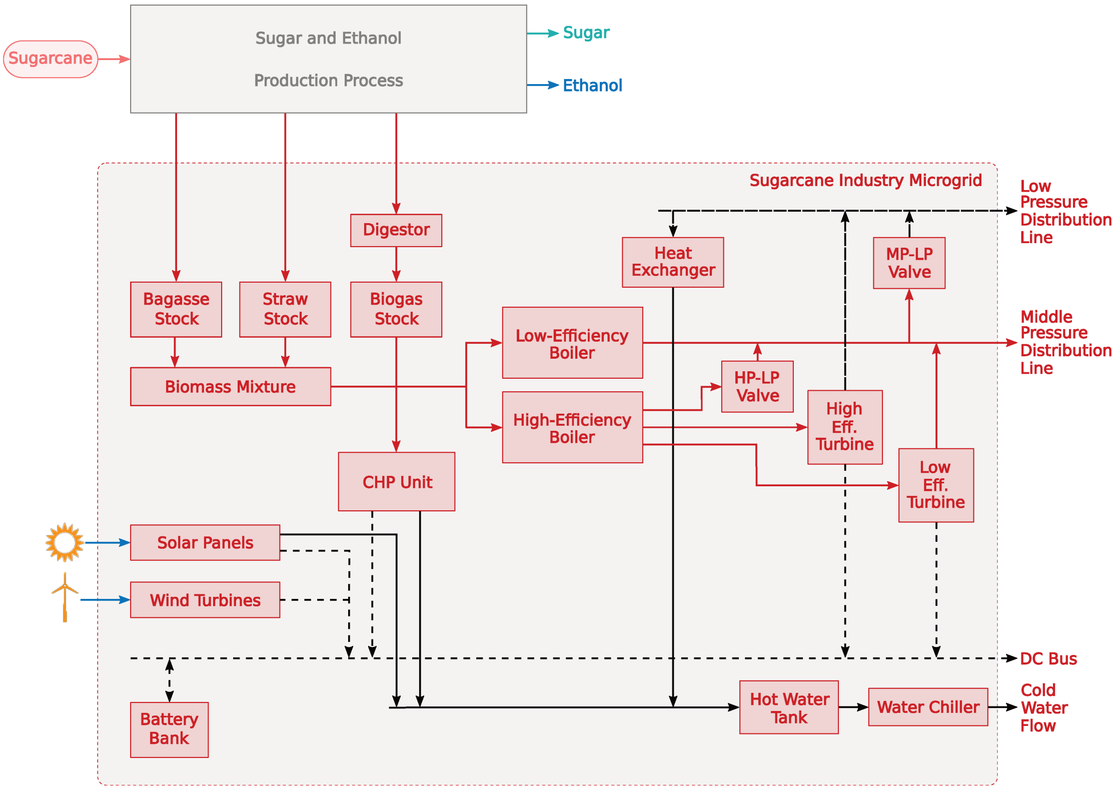


Figure 6.2: The considered microgrid and its subsystems.

accumulate energy (or biomass, that can be converted into energy) when the renewable generation is high and, then use it when there is no renewable production.

The different subsystems in this microgrid must be coordinated by a central EMS. This system is responsible for determining the set-points that are passed to the local controllers of each subsystem so that, at the end of each hour (sampling period), the microgrid is coordinated such that the demands on the controlled outputs are addressed. For instance, at a given moment, the sugar and ethanol production process requires $60\,\mathrm{m}^3\,\mathrm{h}^{-1}$ cold water flow, $90\,\mathrm{Mg\,h}^{-1}$ middle-pressure steam flow, $130\,\mathrm{Mg\,h}^{-1}$ low-pressure steam flow and 7 MW of electric power; then, the central controller operates to find the best possible coordination rule for the microgrid (with respect to some efficiency performance objective), choosing the set-points that are passed to the local controllers. Within the following hour, the local controllers track these set-points and thus, the demands from the sugar and ethanol are delivered.

This microgrid is modelled from a supervisory control viewpoint (with T_s = 1 h sampling period). A high-fidelity model is found in [129]. As depicted in Figure 6.3, it works from the tertiary level, the control inputs are the set-points to the (locally controlled) subsystems u_j[2]. The controlled outputs y_j, stand for the transmission lines/outlets[3], that must agree with demand requirements.

[2] Set-points to the internal energy generation/conversion subsystems, such as boilers, turbines, *etc.*

[3] Electric network, gas pipes, steam lines, *etc.*

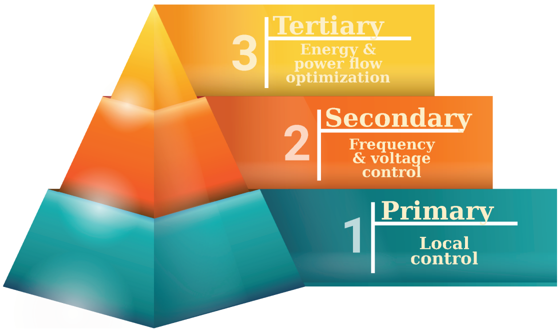


Figure 6.3: Hierarchical control of microgrid.

Therefore, the control problem is the following: a supervisory controller must set hourly set-points for the plant's internal (local) controllers, so that each global demand is met and the energy generation is maximised (together with the sustainability, i.e. renewable quota). This control problem has been treated with optimal predictive schemes in previous works [120, 138].

6.2.1 Faulty plant model

It is also considered that this microgrid is subject to several faults that impair the energy generation. These faults can represent different situations in the plant, such as: the accumulation of debris and residues on the boilers, a pressure valve malfunction, and so forth, as extensively discussed in [33]. As suggested there, faults are represented with multiplicative *loss of effectiveness* factors λ_j, upon the output of each controlled subsystem j. Roughly speaking, it is assumed that the supervisory hourly controller determines set-points u_j for the local subsystems that are not correctly tracked and, the actual output from each of these subsystems are given, individually, by $\lambda_j u_j$.

The control-oriented (hourly-discrete) state-space model of this (faulty) microgrid is:

$$\begin{aligned} \mathbf{x}(k+1) &= A\mathbf{x}(k) + B_1\mathbf{w}(k) + B_2\overbrace{\Lambda(k)\mathbf{u}(k)}^{\mathbf{u_f}(k)}, \\ \mathbf{z}(k) &= C_1\mathbf{x}(k) + D_{11}\mathbf{w}(k) + D_{12}\Lambda(k)\mathbf{u}(k), \\ \mathbf{y}(k) &= C_2\mathbf{x}(k) + D_{21}\mathbf{w}(k) + D_{22}\Lambda(k)\mathbf{u}(k), \end{aligned} \tag{6.1}$$

with:

$$u_{f_j}(k) = \lambda_j(k)u_j(k),$$

where $\lambda_j(k)$ represents the faults upon subsystem j.

Remark 6.1. *The diagonal matrix Λ stands for the collection of faults (on each subsystem) λ_j. The chosen fault representation implies that if $\lambda_j = 1$, subsystem j is healthy, whereas if $\lambda_j = 0$, a complete breakdown has occurred in subsystem j.*

Remark 6.2. *The model above is conceived following an Energy Hubs framework, which is based on mass and energy balance relationships. The inputs to the system are those passed as set-points to the subsystems of the microgrid (detailed in Figure 6.2). The model has inherently more inputs than outputs, since it has multiple energy inputs which are linearly combined to produce a single energy demand outlet. The subsystems (assumed to be regularly controlled by local PI and Proportional-Integral-Derivatives (PIDs)) are viewed as static gains between their inlet variables and output flows. As an example, the CHP unit gives the relationship between the inlet of compressed biogas (output to the biogas stock X_{Bg}) and the outlets of electric power and heated water.*

The considered microgrid has five states (stocks), thirteen control inputs (set-points to the subsystems), five disturbances (renewable inputs), five measured outputs (states) and five controlled outputs (which relate to the demands). Therefore, according to [129], the dimensions of the matrices in the above are: $A \in \mathbb{R}^{5\times5}$, $B_1 \in \mathbb{R}^{5\times5}$, $B_2 \in \mathbb{R}^{5\times13}$, $C_1 \in \mathbb{R}^{5\times5}$, $D_{11} \in \mathbb{R}^{5\times5}$, $D_{12} \in \mathbb{R}^{5\times13}$, $C_2 \in \mathbb{R}^{5\times5}$, $D_{21} \in \mathbb{R}^{5\times5}$ and $D_{22} \in \mathbb{R}^{5\times13}$.

The system state vector $\mathbf{x}$ is defined in Equation (6.2), where each entry stands for the normalised percentage of the storage units: battery bank, bagasse stock, straw stock, biogas stock and hot water tank. The system's measured outputs are also these states: $\mathbf{y}(k) = \mathbf{x}(k)$.

$$\mathbf{x}(k) = \begin{bmatrix} X_{Bat}(k) & X_{Bag}(k) & X_{Str}(k) & X_{Bg}(k) & X_T(k) \end{bmatrix}^T . \tag{6.2}$$

The input vector $\mathbf{u}$ stands for hourly set-points for the internal subsystems[4]. The complete control vector is given by Equation (6.3), where each entry stands for set-points of: energy boiler, lower efficiency turbine, higher efficiency turbine, energy flow to (from) the battery bank, CHP, water chiller, heat exchanger, high-to-middle pressure reduction valve, middle-to-low pressure reduction valve, hot water escape flow, middle pressure steam escape flow, low pressure steam escape flow, and the power available to the external network, respectively.

$$\begin{aligned}\mathbf{u}(k) = \big[&SP_{TU}^{A}(k) \quad SP_{TU}^{B}(k) \quad Pot_{Net}(k) \quad SP_{C}^{B}(k) \ldots \\ &Q_V^{Out}(k) \quad Q_{Esc}^{M}(k) \quad Q_{Esc}^{B}(k) \quad SP_{CHP}(k) \ldots \\ &SP_{Ch}(k) \quad SP_{TC}(k) \quad Pot_{Bat}(k) \quad Q_V^{MB}(k) \quad Q_{Esc}^{T}(k)\big] .\end{aligned} \tag{6.3}$$

The system's controlled outputs $\mathbf{z}$ are defined in Equation (6.4), being P_P the electric power produced to supply the sugarcane processing demand (kW); Q_V^M the flow of middle pressure steam (Mg h^{-1}); Q_V^L the flow of low pressure steam (Mg h^{-1}); Q_{CW} the flow of chilled water required by the fermentation and distillery process (m^3 h^{-1}); finally, P_S represents the electric power available to the external network (kW).

$$\mathbf{z}(k) = \begin{bmatrix} P_P(k) & Q_V^M(k) & Q_V^L(k) & Q_{CW}(k) & P_S(k) \end{bmatrix}^T . \tag{6.4}$$

[4] From an hourly controller's point-of-view, it is assumed that the sampling period is large enough so that all these set-points have been accurately tracked by their respective subsystems (if no fault occurs).

The load disturbances $\mathbf{w}$ are defined in Equation (6.5), being Wnd the speed of the wind ($\mathrm{km\,h^{-1}}$), present in the microgrid's area, $Irrd$ the amount of solar irradiance ($\mathrm{W\,m^{-2}}$) upon the solar panels, Bag, Str and Bg represent the income ($\mathrm{Mg\,h^{-1}}$) of bagasse, straw and biogas to the plant[5], respectively.

$$\mathbf{w}(k) = \begin{bmatrix} Wnd(k) & Irrd(k) & Bag(k) & Str(k) & Bg(k) \end{bmatrix}^T . \tag{6.5}$$

Assumption 6. *The external disturbances* $\mathbf{w}$ *are dependent on the weather conditions on the sugarcane field (solar irrandiance and wind speed). It is assumed that the proposed AFTCS has access to estimations of these disturbances,* $\hat{w}$*, that can be used for feedforward compensation. Neural Network methods can be used for their estimation, as done in [120].*

6.2.2 Demands and operational constraints

To sustain the sugar and ethanol production process, the following demands must always be guaranteed:

- electric power to sustain the plant (in average, 8000 kW) and to sell the surplus to a local DNO (a contract of 11.52 GW h per month has to be respected). This power generation can be continuous, maintaining a constant rate value, or time-varying, with different values at each hour of the month. Moreover, a tolerance of $\pm\psi$ (1.728 GW h) is allowed upon this goal;
- steam in different pressures;
- cold water.

The operational constraints are given by the physical requirements of each subsystem: upper and lower limit bounds of states x and internal set-points u_j.

6.3 Proposed AFTCS/EMS

The main contribution of this chapter is to propose a novel AFTCS for (a sugarcane) renewable microgrid, as follows:

Problem 1. *Find an accurate fault estimation vector* $\hat{\Lambda}(k)$*, and an optimal controller that determines the set-points* $\mathbf{u}(k)$ *in order to guarantee that the outputs* $\mathbf{z}(k)$ *are equal to the (known) demands* $\mathbf{z}_r(k)$*, despite the presence of faults and the renewable disturbances.*

This chapter proposes a modular optimisation-based procedure to solve Problem 1. The fault estimation part of the solution is derived from an MHE scheme, whilst the control part is handled by an (model-update) MPC. The outline of the proposed method is illustrated by Figure 6.4, where the process model block stands for what is given by Equation (6.1)*. In the sequel, both MHE and MPC are explained more minutely.*

[5] These three incomes are proportional to the sugarcane crop, harvest and milling.

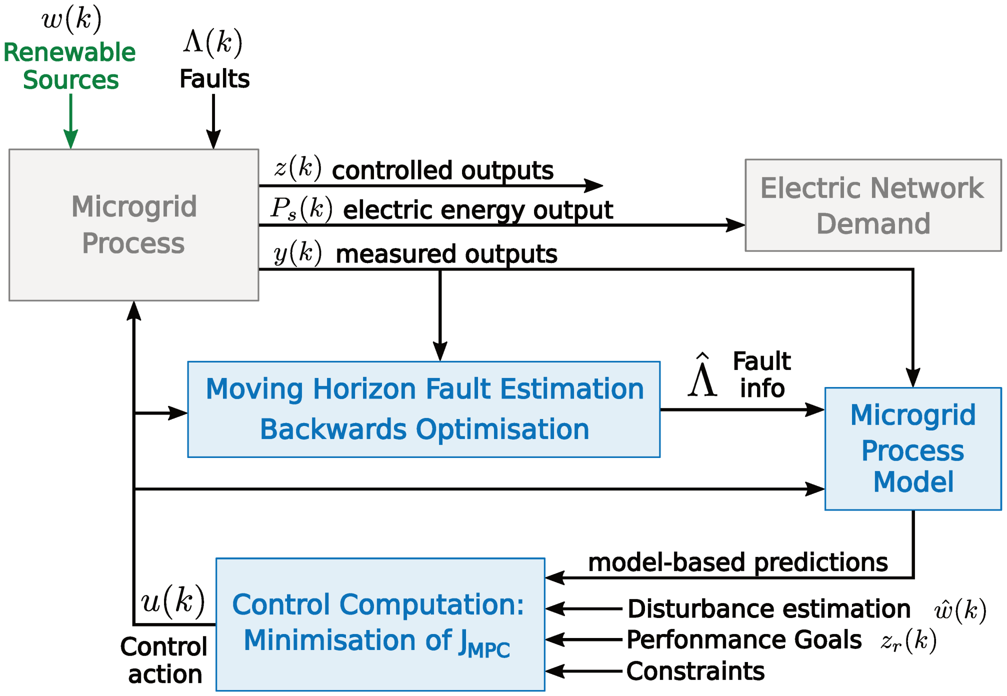

$w(k)$
Renewable Sources
$\Lambda(k)$
Faults
Microgrid Process
$z(k)$ controlled outputs
$P_s(k)$ electric energy output
$y(k)$ measured outputs
Electric Network Demand
Moving Horizon Fault Estimation Backwards Optimisation
$\hat{\Lambda}$ Fault info
Microgrid Process Model
model-based predictions
$u(k)$
Control action
Control Computation: Minimisation of J_{MPC}
Disturbance estimation $\hat{w}(k)$
Perfonmance Goals $z_r(k)$
Constraints

Figure 6.4: Outline of studied AFTCS problem.

6.3.1 Moving horizon fault estimation

Based on the considered microgrid model given in Equation (6.1), the operational constraints of each subsystem, and on the data set of previous information of y, u and disturbances w (considering a backward window of N_e steps), the MHE has the goal to estimate the collection of loss of effectiveness fault terms, that measure the amount of fault in each subsystem, denoted as $\hat{\Lambda}$. The setup of the MHE also embeds state estimation, denoted $\hat{x}$, in its formulation. The cost function for the MHE optimisation is given by[6]:

$$J_{MHE} = l_{ac}(\hat{x}(k-N_e)) + \sum_{i=k-N_e}^{k-1} \left(l(\mu(i), \nu(i)) + \|\nu(i)\|_R^2 \right),$$

where $l_{ac}(\cdot)$ is an arrival cost offset and $l(\cdot)$ denotes the main optimisation cost along the backward horizon.

The data set of the backward horizon is expressed through $\mathbf{w}(i)$, $\mathbf{u}(i)$ and $\mathbf{y}(i)$ for $i = k-N_e, \ldots, k-1$ as known, previously measured variables. The new variable μ collects variance upon the states w_x and upon the fault terms w_λ, which the MHE is set to minimise; ν represents the output measurement noise. Therefore, the main optimisation function is given as follows:

$$l(\mu(i), \nu(i)) = \|\mu(i)\|_Q^2 + \|\nu(i)\|_R^2,$$

where Q and R are adequate weighting matrices.

[6] The weighted norm $||f(i)||_M^2$ denotes $f(i)^T M f(i)$

Finally, the arrival term l_{ac} included in the cost function is used to penalise the distance between the state estimate at the start of the horizon backward $(k-N_e)$ and a previous state estimate $\bar{x}_a$. As previously introduced in Section 2.9 this term is known as an "arrival cost" because it weights the influence of the past information outside the estimation horizon, i.e. from $t=0$ to $t=k-N_e$, in a single concatenated term. This offset is given by:

$$l_{ac}(\hat{x}_a(k-N_e)) = \|\hat{x}_a(k-N_e) - \bar{x}_a\|_P^2$$

where $x_a = [\hat{x}' \ \hat{\Lambda}']'$ and P is a positive definite penalty matrix. The estimate $\hat{x}_a(k-N_e)$ is given by a simple EKF.

At each sampling instant k, the optimisation problem defined in Equation (6.6), below, is solved and a current estimation of states and faults (i.e. $\hat{x}(k)$, $\hat{\Lambda}(k)$) becomes available for control and monitoring purposes. At the next sampling instant, the following arrival term $l_{ac}(\cdot)$ is updated by an EKF guess and the horizon window N_e slides forward by one unit.

$$\begin{aligned} \min_{\hat{\Lambda}} \quad & J_{MHE} \\ \text{s.t.} \quad & \hat{x}(i+1) = A\hat{x}(i) + B_1\mathbf{w}(i) + B_2\hat{\Lambda}(i)\mathbf{u}(i) + w_x(i), \\ & \hat{\Lambda}(i+1) = \hat{\Lambda}(i) + w_\lambda(i), \qquad\qquad i \in \mathbb{Z}_{k-N_e:k-1}, \\ & \mathbf{y}(i) = C_2\hat{x}(i) + D_{21}\mathbf{w}(i) + D_{22}\hat{\Lambda}(i)\mathbf{u}(i) + \nu(i), \\ & \hat{\Lambda}_{\min} \leq \hat{\Lambda}(i) \leq \hat{\Lambda}^{\max}, \\ & \hat{x}_{\min} \leq \hat{x}(i) \leq \hat{x}^{\max}, \qquad\qquad i \in \mathbb{Z}_{k-N_e:k}, \end{aligned} \tag{6.6}$$

Let the steps of the MHE fault estimation be rapidly recalled:

(a) Collect the data set based on the available information from the last N_e steps, through $\mathbf{w}$, $\mathbf{u}$ and $\mathbf{y}$;

(b) Compute an state estimate at the arrival moment, i.e. N_e steps behind k, through an EKF guess, given by $\hat{x}_a(k-N_e)$;

(c) Based on the previous state estimate, from the last MHE iteration, given by $\bar{x}_a$, compute the arrival term offset $l_{ac}(\cdot)$;

(d) Solve the constrained optimisation problem, resolving the QP given by Equation (6.6);

(e) Pass the output of the QP solution, the vector of estimated fault terms $\hat{\Lambda}$, to the MPC algorithm, which is detailed in the sequel.

6.3.2 Fault-tolerant model predictive control

The complete AFTCS is derived with an MPC algorithm for fault-tolerant control purposes. This MPC includes performance goals (such as demand deliverance) and process operational constraints. To design this model-based predictive controller, that takes the $\hat{\Lambda}(k)$ fault information from the MHE, a cost function J_{MPC} is minimised subject to constraints and the load demands variables $\mathbf{z}_r(k)$, previously detailed in Section 6.2.2.

For the energy plant to abide to the energy production rules, supplying the local DNO with 11.52 GW h per month, the following cost function is established:

$$J_{MPC} = \sum_{i=0}^{N_p-1} \|E_D(i)\|^2_{Q_s} + \|\mathbf{z}(i) - \mathbf{z}_r(i)\|^2_{Q_z} + \|\epsilon(i)\|^2_{Q_\epsilon} + \|\mathbf{x}(i) - x_{ref}\|^2_{Q_x} + q_u u(i),$$

where $E_D(i)$ denotes the difference from the energy delivered to the external grid and the energy contract generation goal, weighted through Q_s; the demand compliance error term is expressed through $\mathbf{z}(i) - \mathbf{z}_r(i)$, which is weighted through Q_z; a slack variable ϵ, weighted through Q_ϵ, is introduced to the optimisation procedure to account for the contract tolerance; the deviance $\mathbf{x}(i) - x_{ref}$ is optimised to bring the states close to a certain reference value, weighted through Q_x; finally, q_u linearly weights the control inputs - note that linearity is allowed because the control vector u is always positive.

The delivered energy deliverance is measured through:

$$E_D(i) = \left[T_s P_S(i) - \epsilon(i) - \frac{E_{con} - E_{sum}}{T_{goal}} \right],$$

where E_{sum} represents the electric energy that has already been produced (until instant k), E_{con} stands for the energy sales contract value; $(E_{con} - E_{sum})$ represents how much electric energy the microgrid still has to produce until the end of the month, due to contract requirement, taking T_{goal} as the remaining time until the end of the month, in hours.

The reference to the states $\hat{x}_{ref}$, represents a vector of 50 % references, used so that the states stay close to this mean value. The prediction horizon of the MPC algorithm is denoted N_p. Note that z^r stands for the load demands the EMS must guarantee, to maintain the sugar/ethanol production process running.

The MPC synthesis, then, resides in minimising J_{MPC}, with $\mathbf{x}(0) = x(k)$ and considering a frozen LTI version of the model in Equation (6.1), updated with $\Lambda(i) = \hat{\Lambda}(k)$ (informed by the MHE), subject to the following operational constraints:

$$\begin{aligned} \mathbf{x}(i+1) &= A\mathbf{x}(i) + B_1\hat{w}(i) + B_2\hat{\Lambda}\mathbf{u}(i), \\ \mathbf{z}(i) &= C_1\mathbf{x}(i) + D_{11}\hat{w}(i) + D_{12}\hat{\Lambda}\mathbf{u}(i), \\ x_{\min} &\le \mathbf{x}(i+1) \le x^{\max} \\ u_{\min} &\le \mathbf{u}(i) \le u^{\max} \\ 0 &\le \mathbf{z}(i) \\ 0 &\le \epsilon(i) \le \psi, \qquad i \in \mathbb{Z}_{0:N_p-1}. \end{aligned}$$

Note that the slack variable ϵ is included directly into J_{MPC}, since it is evident that the energy generation has the production as close as possible to the contract goal E_{con}, which means that ϵ has to be as close to zero as possible.

The cost function J_{MPC} is very similar to the one used in [138], embedding contract generation goals through E_D and demand compliance error term through $(\mathbf{z} - \mathbf{z}_r)$, while altogether maintaining the stocks (states) close to a half-full reference. The values for the weighting matrices Q_s, Q_z, Q_ϵ, Q_x and q_u can also be found in [138], taken to ensure a trade-off between these goals; it is important to mention that the MPC is set to, most importantly, generate the contract goal and, secondly, comply with the demands, which means $||Q_s||_\infty > ||Q_z||_\infty$.

6.3.3 Integrated AFTCS

One of the advantages of the proposed AFTCS method based on the MHE and MPC optimisation procedures is that, even though it has a systematic modular design and implementation procedure, it can also be formulated as a single-step joint estimation and control optimisation procedure. The original formulation is the one illustrated in Figure 6.4, where the MHE and MPC compose separate steps. This joint formulation is illustrated below, in Figure 6.5, where $J_{full} = a_{MHE}(i)J_{MHE} + a_{MPC}(i)J_{MPC}$. Note that the joint formulation is indeed possible due to the fact that both MPC and MHE procedures share a common mathematical formalism. The joint approach is an elegant form of embedding the proposed AFTCS in a single micro-controller that performs the complete AFTCS procedure.

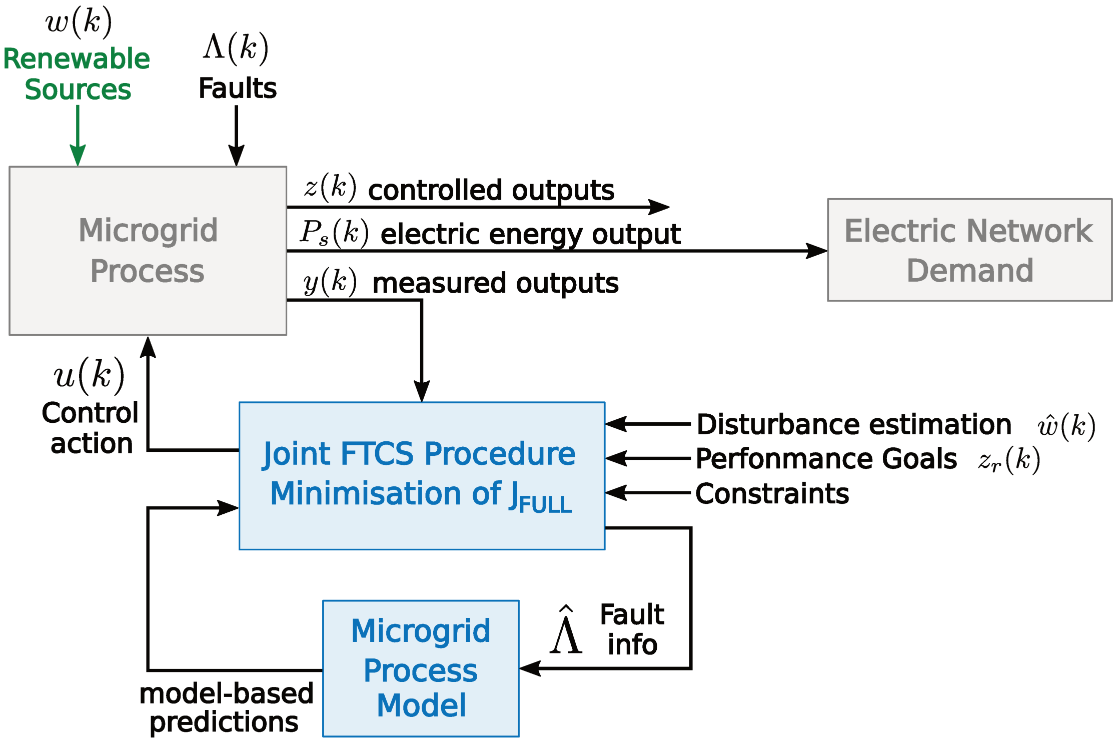

$w(k)$
Renewable Sources
$\Lambda(k)$
Faults
Microgrid Process
$z(k)$ controlled outputs
$P_s(k)$ electric energy output
$y(k)$ measured outputs
Electric Network Demand
$u(k)$
Control action
Joint FTCS Procedure Minimisation of J_{FULL}
Disturbance estimation $\hat{w}(k)$
Perfonmance Goals $z_r(k)$
Constraints
Microgrid Process Model
$\hat{\Lambda}$ Fault info
model-based predictions

Figure 6.5: Outline of the joint AFTCS formulation (single QP).

The joint AFTCS optimisation is done by solving the following QP, which has a concatenated decision variable ϱ which concatenates the estimation of the fault terms $\hat{\Lambda}$ as well as the control signal u, i.e. $\varrho = \text{col}\{\hat{\Lambda}, u\}^T$:

$$
\begin{aligned}
\min_{\varrho} \quad & a_{MHE}(i)J_{MHE} + a_{MPC}(i)J_{MPC}, \\
\text{s.t.} \quad & a_{MHE}(i) = 1 \quad \text{and} \quad a_{MPC}(i) = 0, \quad \forall i \in \mathbb{Z}_{k-N_e:k-1} \\
& \hat{x}(i+1) = A\hat{x}(i) + B_1\mathbf{w}(i) + B_2\hat{\Lambda}(i)\mathbf{u}(i) + w_x(i), \quad \forall i \in \mathbb{Z}_{k-N_e:k-1} \\
& \hat{\Lambda}(i+1) = \hat{\Lambda}(i) + w_\lambda(i), \quad \forall i \in \mathbb{Z}_{k-N_e:k-1} \\
& \mathbf{y}(i) = C_2\hat{x}(i) + D_{21}\mathbf{w}(i) + D_{22}\hat{\Lambda}(i)\mathbf{u}(i) + \nu(i), \quad \forall i \in \mathbb{Z}_{k-N_e:k-1} \\
& x_{\min} \le \hat{x}(i+1) \le x^{\max}, \quad \forall i \in \mathbb{Z}_{k-N_e:k-1} \\
& \hat{\Lambda}_{\min} \le \hat{\Lambda}(i) \le \hat{\Lambda}^{\max}, \quad \forall i \in \mathbb{Z}_{k-N_e:k-1} \\
& a_{MHE}(i) = 0 \quad \text{and} \quad a_{MPC}(i) = 1, \quad \forall i \in \mathbb{Z}_{k:k+N_p-1} \\
& \mathbf{x}(i+1) = A\mathbf{x}(i) + B_1\hat{w}(i) + B_2\hat{\Lambda}(k-1)\mathbf{u}(i), \quad \forall i \in \mathbb{Z}_{k:k+N_p-1} \\
& \mathbf{z}(i) = C_1\mathbf{x}(i) + D_{11}\hat{w}(i) + D_{12}\hat{\Lambda}(k-1)\mathbf{u}(i), \quad \forall i \in \mathbb{Z}_{k:k+N_p-1} \\
& x_{\min} \le \mathbf{x}(i+1) \le x^{\max}, \quad \forall i \in \mathbb{Z}_{k:k+N_p-1} \\
& u_{\min} \le \mathbf{u}(i) \le u^{\max}, \quad \forall i \in \mathbb{Z}_{k:k+N_p-1} \\
& 0 \le \mathbf{z}(i), \quad \forall i \in \mathbb{Z}_{k:k+N_p-1} \\
& 0 \le \epsilon(i) \le \psi, \quad \forall i \in \mathbb{Z}_{k:k+N_p-1}
\end{aligned}
\tag{6.7}
$$

Note that this single QP is subject to the whole set of constraints (those of both MHE and MPC). The new horizon is, thus, $N_e + N_p$. In this problem, $\mathbf{u}$ is fixed for the first part (past values) and $\Lambda(N_e + i) = \hat{\Lambda}(N_e)$ for $i = 1$ to N_p.

6.3.4 Robustness remarks

Some few words must be stated about robustness, stability and feasibility of the proposed solution:

- With respect to the MHE problem for estimating faults, since the open-loop microgrid plant is *a priori* stable, and the optimisation problem is data-based, local asymptotic convergence of the fault vector $\hat{\lambda}(k)$ is always ensured. Moreover, since the variability term $w_\lambda(k)$ of the optimisation problem is not constrained (but minimised), recursive feasibility is also always ensured. Demonstrating robustness of the MHE scheme is not the focus of this work, and has been discussed in other papers [77, 139]. Note that the robustness and uncertainty considerations of fault estimation schemes for microgrid have also been previously discussed in [140], considering the same sugarcane-based system.
- Regarding the MPC problem of the considered microgrid, robustness, feasibility and stability remarks have been discussed in [138, 97]. Essentially, robustness and recursive feasibility of this optimisation loop can be verified regarding how the optimisation function J_{MPC} is formulated. Synthetically, it must be $\mathcal{K}$-class lower-bounded with respect to x, with its offset term, if present, $\mathcal{K}$-class upper-bounded with respect to x. Moreover, the offset cost should be decreasing along the horizon. Indeed, all these conditions are verified and the MPC solution is recursively feasible for the considered microgrid

problem, which means that for whichever starting condition $x(0)$ within the operational set (delimited by x_{min} and x^{max}), the proposed AFTCS will ensure convergence to some stable equilibrium as $k \to \infty$ through a sequence of admissible control policies.

- The robustness of the MHE and MPC verified separately does not *a priori* ensure the robustness of the joint AFTCS version of Equation (6.7). Therefore, if one is to implement the joint AFTCS formulation, the total cost function J_{full} should be analysed: This cost must verify the quadratic boundedness and Lyapunov-decay along the horizon (the same conditions stated above regarding J_{MPC}).
- It must also be commented that the best performance solution of the considered problem is the one with a Fault-Forecast-MPC, which embeds to the model the future behaviour of Λ (along the horizon). Of course, such formulation is not possible, since only the instantaneous values of Λ are available (from the MHE layer) at each sampling instant k.

6.4 Results and analysis

In this Section, high-fidelity simulation results are given in order to illustrate the proposed Fault-Tolerant EMS. The strategies were synthesised using Yalmip package [88], Gurobi optimisation solver [141] and Matlab software.

6.4.1 Simulation model

The considered simulation model for the sugarcane plant microgrid is the one proposed and validated in [129]. This model describes the dynamics of the microgrid from a tertiary-level control viewpoint, which means that it is assumed that local controllers are acting to ensure that each subsystem tracks its operational set-point determined by the EMS (the proposed AFTCS) withing one hour, which is the sampling period of the model.

These local controllers are those that regulate the Boilers, Turbines, CHP, Batteries and so forth (the thirteen subsystems presented in Figure 6.2). These lower control layers consist on classical PI and PID controllers. These controllers are not concerned with loss of effectiveness faults (gain decreases). From the viewpoint of the MPC, each (locally-controlled) subsystem operates as a static gain between input and output (set-point), meaning that transient dynamics are not present, just the integration dynamics that derive from balance of energy and mass relationships.

6.4.2 QP criteria selection

Firstly, Table 6.1 synthesises the general performances obtained with the proposed AFTCS framework with respect to the adapted selection criteria of the (MHE and MPC) QPs. This is an important feature to be analysed, because the execution time of the AFTCS also changes according to the size of the horizons (N_e and N_p).

The performance results comprised in this table are measured through the average *Root Mean Square* (RMS) index of the demand tracking error, i.e. RMS$\{\mathbf{z} - \mathbf{z}_r\}$, where RMS$\{\cdot\}$ measures the RMS value of the discrete-time variable over the whole simulation span. The "H_RMS" index measures the RMS value for the first N_p samples of $(\mathbf{z} - \mathbf{z}_r)$ (inside the horizon). The IAE is a well-known time-domain index; it measures the integral of the average tracking error, i.e. of $(\mathbf{z} - \mathbf{z}_r)$ along the simulation (see Appendix D). These results exhibit that, when smaller horizons are tested, the control strategy forces the convergence of the outputs to the demands at the first instants, while for larger horizons this condition is relaxed. Moreover, as expected, the average computational stress increases with the size of the horizons, since the generated QPs have larger orders. Analysing the obtained results, it seems that an adequate trade-off seems to be the choice of $N_p = N_e = 10\,$h, which is coherent with the discussion in [138]. For this reason, through the sequel, the results consider $N_p = N_e = 10\,$h.

Table 6.1: MHE-MPC performances vs. length of the horizon.

Horizon	MHE	MPC	RMS	H_RMS	IAE
5 h	0.0652 s	0.0113 s	0.3308	0.2440	6.8814
10 h	0.1036 s	0.0151 s	0.3417	0.2205	7.3599
15 h	0.1482 s	0.0197 s	0.3689	0.2791	8.2567
20 h	0.1841 s	0.0257 s	0.3887	0.3422	8.8167
24 h	0.1967 s	0.0261 s	0.4579	0.4120	10.7764

Note that, for all these possible horizon sizes (for both N_p and N_e), the yielded QPs were evaluated within 1 s (this is the OCE index[7]). Note that the sampling period of the process is of a full hour, which means that the computational burden of the QPs is not an implementation issue. Larger horizons (greater than 24 h) have also been tested; anyhow, performances were not enhanced. This is coherent with the average settling period of the considered fault terms, which is smaller than 10 h.

Remark 6.3. *Many other horizon values were tested. There are only subtle differences between the results (indices) achieved with intermediate horizons (between* 5 *and* 10 h, 10 *and* 15 h *and so on). Therefore, Table 6.1 concatenates the horizons which presented the best results in terms of the performance indices.*

6.4.3 Simulation

The following simulation scenario is considered: a whole month of energy generation and demand subject to average sun and wind availability, as presented in [120]. Moreover, as done in [33], faults are emulated to occur in several subsystems, as displays Figure 6.6: Turbine A (6.6a), high-to-middle pressure (HP-MP) reduction valve (6.6b), CHP (6.6c) and water chiller (6.6d).

[7] In an i5-3337U CPU@2.7 GHz (2 Cores) with 8 GB of RAM

Furthermore, in Figure 6.6, the estimation provided through the MHE optimisation is also shown. Clearly, the estimation results are quite good and effective. It is important to note that the high slopes in these faults (as well as the concomitant occurrence) were chosen to illustrate worst-case conditions and expose the capacity of the proposed AFTCS method to tolerate such harshness.

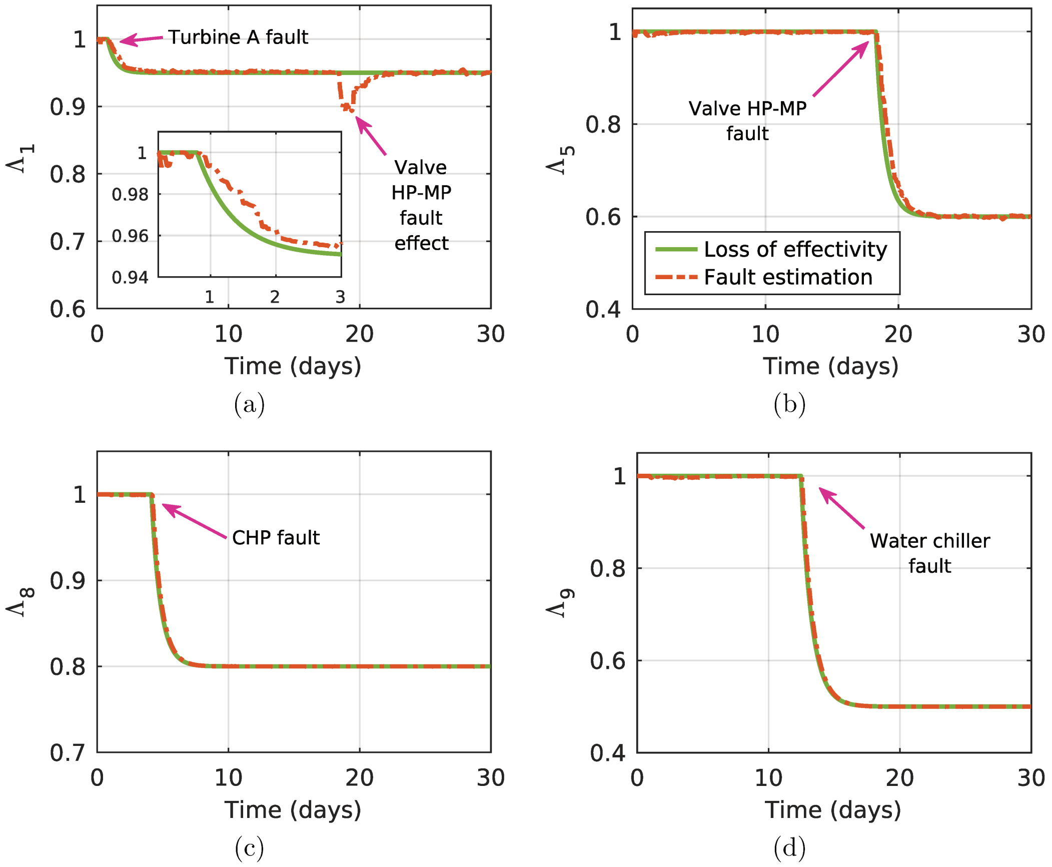


Figure 6.6: Simulated faults: (a) Turbine A; (b) HP-MP pressure reduction valve; (c) CHP; (d) Water chiller.

It must be reaffirmed that a great advantage of the proposed MHE method to estimate these loss of effectiveness faults in the subsystems of the considered microgrid is that it is a unified structured. This MHE optimisation procedure solves on-line, a single QP to estimate all fault terms on all thirteen subsystems, which is rather convenient. With respect to the fault estimation scheme proposed in [33], it can be designed and run with much more ease; the previous technique requires one state observer for each control variable, i.e. a bank of thirteen observers, each individually synthesised and tuned. The tuning step for the MHE scheme is unified and done in a single step, through the weighting matrices Q, R and P.

These four effectiveness losses faults (depicted in Figure 6.6) were chosen to represent:

(a) an increase of the bearing temperature on turbine A, starting at 20 h;

(b) from 100 h, a gas leakage and partial clogging on hot water flow duct occur on the CHP unit;

(c) partial clogging in the expansion valve and an increased bearing temperature of the compressor motor of the water chiller income at 300 h;

(d) at 440 h, a steam leakage, implying a loss of mass flow, occurs in the pressure reduction valve (HP-MP).

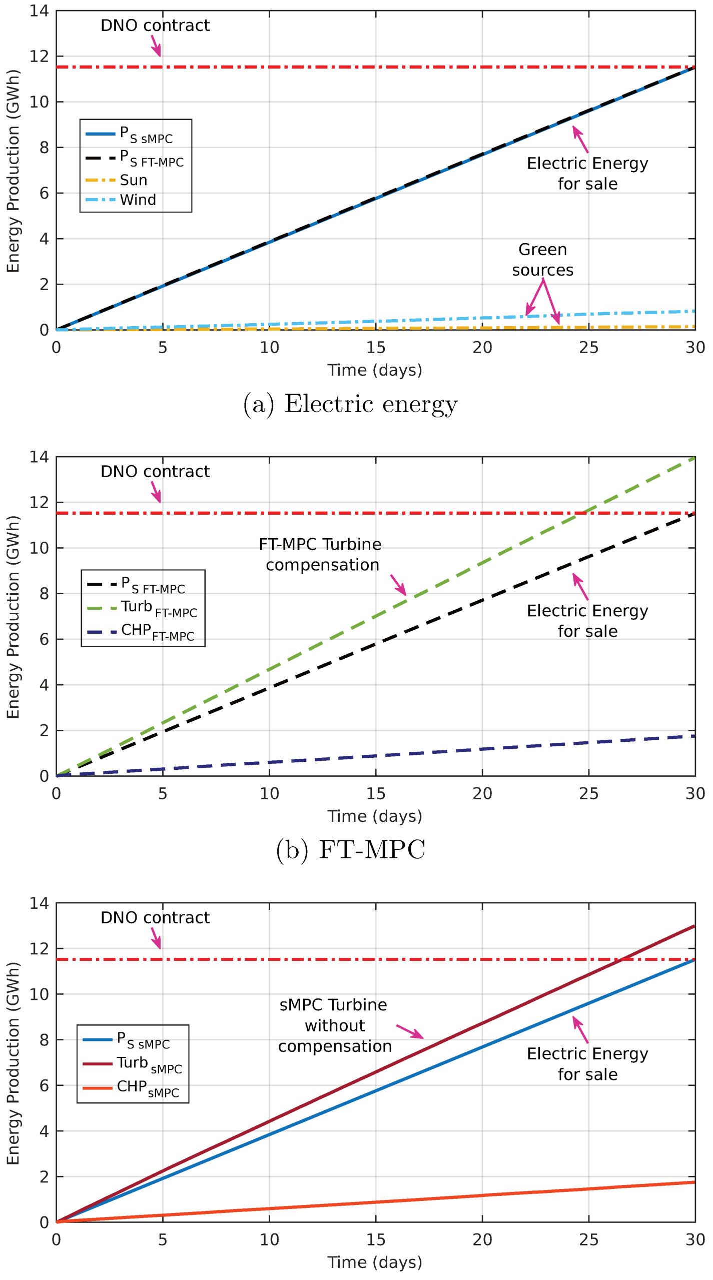
Now, the control results achieved with the proposed AFTCS are given in Figures 6.7 to 6.9. These figures stand for the total energy generation, the electric power and cold water production and the two steam demands, respectively. Therein, FT-MPC (dashed bold line) stands for the proposed AFTCS and, for comparison goals, sMPC (full blue line) stands for a "standard" MPC, taken from [138]. This sMPC solves the same MPC optimisation problem, minimising J_{MPC}, but disregards the effect of the faults. Consequently, if no faults occur ($\Lambda(k) = \mathbb{I}_{13}$) the operation of both FT-MPC and sMPC would be the equivalent.

Firstly, regarding what Figure 6.7 shows, it must be commented that the DNO contract is accomplished by controlling both strategies. This can be explained since the MPC optimisation weights (Q_s through q_u) were chosen so that the most important goal is to achieve the energy sales contract (as explained, $||Q_s||_\infty$ is the biggest norm of these weights). Figure 6.7a depicts this fact, where the energy generation by the green sources[8] (the same happens with both control strategies, since these variables are not controllable) is shown together with the energy sold by the microgrid to the network, when controlled via FT-MPC or sMPC. Then, Figures 6.7b and 6.7c further detail how this total energy generation is performed with each strategy, FT-MPC and sMPC, respectively. The energy generation by the turbines[9] plays the biggest role with regard to this matter, since the AFTCS further boosts its generation to account for the losses implied by the faults; it is the turbine generation that is adapted to produce more energy that is not sold, needed to sustain internal process, i.e. demand upon $\mathbf{z}_1$. The difference between the strategies grows roughly 1 GW until the end of the month. With both control strategies, the energy produced in the CHP unit is the same, since this subsystem is boosted to its maximal production (the set-point $SP_{CHP}(k)$ defined by the controllers is always saturated), since it is the most energy-efficient subsystem of the considered microgrid.

It is also important to notice that the slack variable ϵ in the MPC QP allows to the AFTCS controller overachieve the DNO quota, in the case of sudden (last-minute) disturbances. Such phenomenon is allowed by the tolerance factor in this DNO contract, as explained in Section 6.2.2.

[8] Note that the green sources only provide a small percentage of the entire generation, for both cases, as expected, regarding the considered scenario, as seen in [120].

[9] The main subsystem used by the AFTCS to tolerate the faults is the (highly efficiency) turbine A.



(a) Electric energy

(b) FT-MPC

(c) sMPC

Figure 6.7: Energy production.

Despite producing the contract energy goal, it occurs that the sMPC strategy, which has no knowledge of the fault, must sacrifice the accomplishment of the demands to generate enough energy for sale. This fact is shown through Figures 6.8 and 6.9, which depict the controlled outputs $\mathbf{z}$ and the demands $\mathbf{z}_r$. This kind of situation would be unacceptable in practice, since it leads to a disruption on the ethanol/sugar production line.

Figure 6.8 enlightens that the faults highly disrupt the fulfilment of the electric power demand, especially due to the gain losses in turbine A - only the AFTCS method can overlap this issue, which is quite impressive. Besides, since all subsystems are physically interconnected, there is an inherent coupling effect between the faults (and their estimation) and, thereby, upon the demand compliance (see, for instance, the fault that occurs on day 18). Evidently, the sMPC does not comply with these electric power demands already from the second day, when the turbine A fault occurs. From day 13, a fault occurs on the water chiller, which also disrupts the cold water demand compliance by the sMPC. With the AFTCS, this fault is almost unnoticeable (the compensation is quite fast).

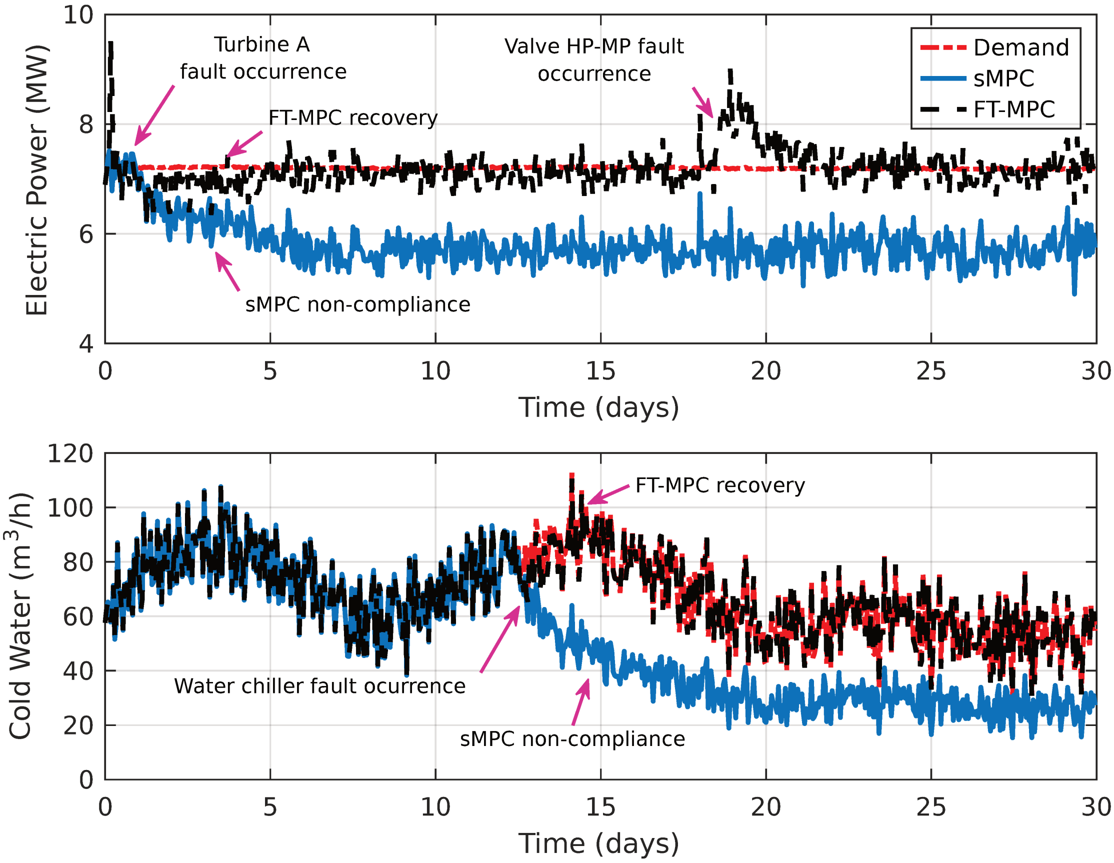


Figure 6.8: Electric energy and cold water flow demand.

With regard to Figure 6.9, it must be noted that the loss of effectiveness fault in the turbine A directly affects the low pressure steam generation, which is demanded by the production of sugar and bio-ethanol. Besides, the requirements of the MP steam is mainly affected by a fault on the reduce pressure HP-MP valve, because it is the main generation source. It should be noted that the sMPC controller, having no knowledge of the fault, is unable to reconfigure the steam generation, and therefore does not achieve the plant demands. To further assess this fact, Table 6.2 gives the mean values for these four demand curves, and the mean values of the production with the FT-MPC and sMPC strategy. Clearly, the average values achieved with the AFTCS are much closer to the expected amount (perfect compliance). The only moment when the demands are not met is when the AFTCS is compensating the fault or due to coupling effects between the thirteen subsystems.

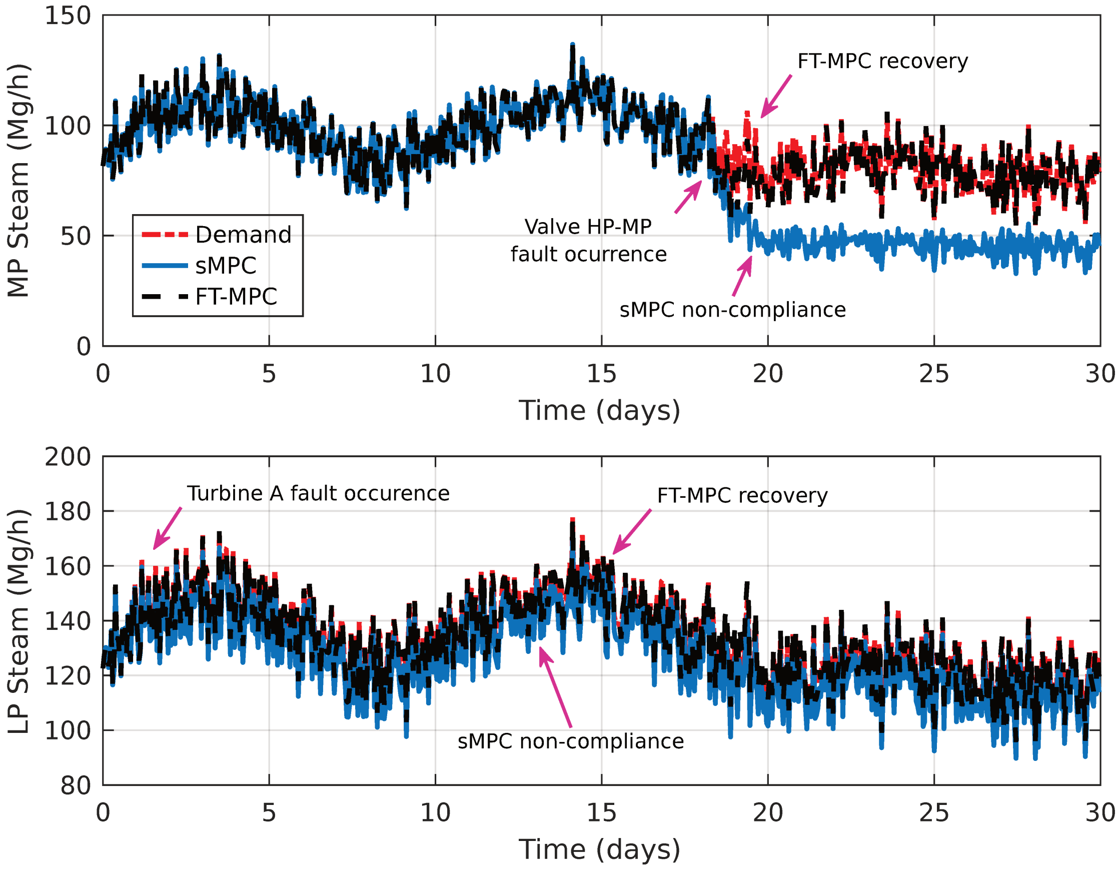


Figure 6.9: Steam demands.

Table 6.2: Average demand production.

	Electric Power	Cold Water	MP Steam	LP Steam
	(MW)	($\mathrm{m}^3\,\mathrm{h}^{-1}$)	($\mathrm{Mg\,h}^{-1}$)	($\mathrm{Mg\,h}^{-1}$)
Demand	7.19	67.93	91.93	132.93
FT-MPC	7.16	67.68	91.48	133.15
sMPC	5.83	50.37	79.45	127.26

Finally, Figure 6.10 shows the control set-points and the actual outputs of the faulty systems (whose outputs are corrupted due to the loss of effectiveness faults). Complementary, Table 6.3 gives the mean values for $u_{f_j}(k) = \lambda_j(k)u_j(k)$ for the fault-affected subsystems, before and after a fault occurs.

In this Figure, it becomes clear how the AFTCS (according to the scheduled faulty process model) increases these set-points so as to find a better way to achieve the control energy management goals. This change on the control policies is also clearly seen in the mean values presented in Table 6.3; note that prior to fault events, the mean values of the control actions with both strategies are roughly the same. The sMPC method, on the other hand, can do no such fault reconfiguration mechanism since it does not know the level of faults on the system.

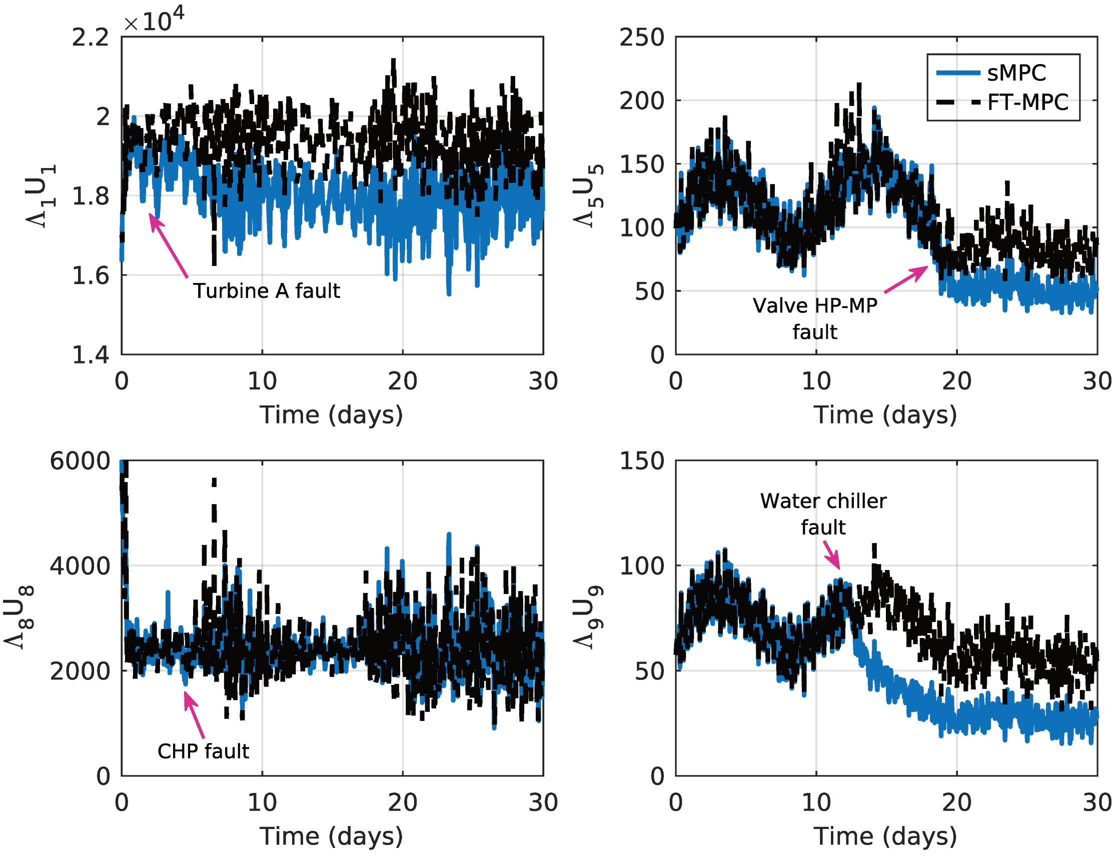


Figure 6.10: Manipulated variables.

Table 6.3: Average input data.

	$\Lambda_1 U_1$		$\Lambda_5 U_5$		$\Lambda_8 U_8$		$\Lambda_9 U_9$	
Hours	< 20	≥ 20	< 440	≥ 440	< 100	≥ 100	< 300	≥ 300
FT-MPC	18694.68	19416.34	123.57	83.23	2604.79	2405.49	73.33	63.66
sMPC	18452.14	18050.87	122.54	53.78	2610.11	2403.00	73.30	34.08

Evidently, as seen and discussed with respect to the above figures and tables, the proposed AFTCS approach is able to overcome the effect of faults (changing set-points as soon as faults are estimated), being able to produce the contract energy goal, at the end of the month, and fulfilling the plant demands throughout all days. The proposed method is quite strong, residing in solely two QPs. The tuning parameters are some matricial weights for each QP and no more; there is no need for harsh design procedures (computing matrix inequalities or non-linear programs) and the implementation steps are quite direct.

6.5 Conclusion

This chapter presented the issue of controlling and managing a sugarcane plant based microgrid system that is subject to multiple and simultaneous faults. For such, an integrated optimisation-based AFTCS was designed to cope with the presence of faults, while always attending to the load demands and producing the defined amount of energy. The presented approach highlights the interest of the proposed MHE/MPC paradigm for the development of Fault-Tolerant EMSs. Simulation results show that the proposed scheme can accurately re-adjust the control law so that effects of faults are mitigated.

Chapter 7

Conclusions and future works

This final chapter contains a summary of the thesis proposals, a detailed discussion of the achieved results and a perspective of possibles future research directions.

7.1 Summary

This thesis addressed the problem of analysing and designing control methods to mitigate, as well as possible, the fault occurrence of actuator and sensor elements on industrial processes, where the fault presence can lead the system to a degraded performance, or in worst cases to instability.

To achieve this, firstly, Chapter 1 introduced the reader on the origin and motivation, as well as the research area state-of-the-art and objectives of this thesis. Secondly, some existing and basic concepts required by the AFTCS design were recalled in Chapter 2.

Furthermore, two different model-based approaches were presented on this thesis: on the one hand, throughout Chapters 3 to 5 the design of a novel observer-based FDD module and an innovative QP-based adaptive FTC method were combined to develop a non-linear AFTCS, on the basis of LPV system representation. A sketched procedure about its application on a non-linear chemical industrial process was given. On the other hand, on Chapter 6, an optimisation-based AFTCS strategy was proposed to develop a tertiary-level EMS, based on a sugarcane distillery power plant. Where the availability of several renewable energy sources exposes the capacity of proposed integrated AFTCS to mitigate faults, seeking maximal profit and increasing sustainability.

Lastly, it is noteworthy that for each studied method a detailed computer-aided simulation scenario was exhibited. Enabling qualitative and quantitative discussions about its performance and effectiveness, via graphs and metric indices respectively.

7.2 Contributions

The main contributions of this thesis are highlighted as follows:

- The development of a novel observer-based FDD strategy for non-linear systems was presented. The observers design and its stability conditions are based on terms of LMI.
- An MHE procedure coupled to an MPC formulation was addressed to develop a non-linear predictive controller. Terminal ingredients to guarantee MPC stability are based on the resolution of off-line LMI problems.
- A model-based AFTCS for non-linear industrial processes was developed.
- An integrated fault-tolerant EMS MPC was addressed. An MHE method is proposed to estimate both system states and incipient faults on renewable microgrids.
- Realistic numerical simulation scenarios were presented, showing the effectiveness of each method. Examples of typical chemical industrial processes are given, as well as a sugarcane industrial microgrid.
- Performance indices were used to demonstrate and quantify the effectiveness of the proposed methods.

7.3 Future works

The fault-tolerance control system remains an interesting and open research area involving different fields of knowledge. Accordingly, with respect to the solutions presented in this thesis remarkable future research topics are:

- The study and development of FDD schemes to achieve simultaneous fault detection and estimation.
- The research and design of simultaneous integrated AFTCS for non-linear systems. With the purpose of reducing the undesirable effects introduced by FDD-FTC interactions.
- Consider the inclusion of parameter-dependent constraints and terminal set into the MPC design procedure, to obtain a larger domain of attraction.
- Evaluate the economic impacts of stopping subsystems for repairs. Planning the extension of the plant model knowledge, considering the maintenance actions, that could allow a more realistic scenario.

List of publications

The development presented in this thesis enabled the publication of several scientific works. These are listed below:

Journal papers

- Emanuel Bernardi and Eduardo J. Adam. "Fault-tolerant Linear Parameter Varying Model Predictive Control Scheme for Industrial Processes". In: *Journal of the Taiwan Institute of Chemical Engineers* (2021). 1876-1070. DOI:10.1016/j.jtice.2021.10.003
- Emanuel Bernardi, Marcelo M. Morato, Paulo C. Mendes, Julio E. Normey-Rico and Eduardo J. Adam. "Fault-Tolerant Energy Management for an Industrial Microgrid: A Compact Optimization Method". In: *International Journal of Electrical Power and Energy Systems* 124 (2021), p. 106342. ISSN: 0142-0615. DOI: 10.1016/j.ijepes.2020.106342.
- Hugo. A. Pipino, Marcelo M. Morato, Emanuel Bernardi, Eduardo J. Adam, and Julio E. Normey-Rico. "Nonlinear Temperature Regulation of Solar Collectors with a Fast Adaptive Polytopic LPV MPC Formulation". In: *Solar Energy* 209C (Oct. 2020), pp. 214-225. ISSN: 0038-092X. DOI: 10.1016/j.solener.2020.09.005.
- Emanuel Bernardi and Eduardo J. Adam. "Observer-based Fault Detection and Diagnosis Strategy for Industrial Processes". In: *Journal of the Franklin Institute* 357 (14 2020), pp. 9895-9922. ISSN: 0016-0032. DOI: 10.1016/j.jfranklin.2020.07.046.

Conference papers

- Emanuel Bernardi, Hugo A. Pipino, Carlos A. Cappelletti and Eduardo J. Adam. "Adaptive Predictive Control for Industrial Processes". In: *XIX Workshop on Information Processing and Control (RPIC)*. San Juan, Argentina. (2021).
- E. Bernardi and E. J. Adam. "Nonlinear Fault-tolerant Model Predictive Control Strategy for Industrial Processes". In: *2020 Argentine Conference on Automatic Control (AADECA)*. Ciudad Autónoma de Buenos Aires, Argentina (2020) pp. 105-110. ISBN: 978-987-46859-2-6. DOI: 10.23919/AADECA49780.2020.9301630.

- Hugo A. Pipino, Emanuel Bernardi, Marcelo M. Morato, Carlos A. Cappelletti, Eduardo J. Adam and Julio E. Normey-Rico. "Formulación LPV-MPC Adaptativo para Procesos Industriales No Lineales". In: *2020 Argentine Conference on Automatic Control (AADECA)*, Buenos Aires, Argentina (2020) pp. 195-200. `ISBN: 978-987-46859-2-6`.
- Emanuel Bernardi, Hugo A. Pipino and Eduardo J. Adam. "Full-order Output Observer Applied to a Linear Parameter Varying System with Unknown Input". In: *2020 IEEE Biennial Congress of Argentina (ARGENCON).* Resistencia, Argentina (2020). `ISBN: 978-1-7281-5957-7`.
- Hugo A. Pipino, Emanuel Bernardi, Carlos A. Cappelletti and Eduardo J. Adam. "Predictive Control for Multi-Model Systems". In: *2020 IEEE Biennial Congress of Argentina (ARGENCON).* Resistencia, Argentina (2020). `ISBN: 978-1-7281-5957-7`.
- Emanuel Bernardi and Eduardo J. Adam. "Observador de Salida de Tiempo Discreto para Detección y Diagnóstico de Fallas en Sensores". In: *Jornadas de Ciencia y Tecnología 2020 de la UTN Facultad Regional San Francisco.* San Francisco, Argentina. (2020) pp. 107-108. `ISBN: 978-950-42-0201-1`.
- Marcelo M. Morato, Hugo A. Pipino, Emanuel Bernardi, Diego M. Ferreyra, Eduardo J. Adam and Julio E. Normey-Rico. "Sub-optimal Linear Parameter Varying Model Predictive Control for Solar Collectors". In: *2020 IEEE International Conference on Industrial Technology (ICIT).* Buenos Aires, Argentina. (2020) pp. 95-100. `ISBN: 978-1-7281-5755-9. DOI: 10.1109/ICIT45562.2020.9067139`.
- Emanuel Bernardi, Carlos A. Cappelletti and Eduardo J. Adam. "Fault-tolerant Model Predictive Control Strategy Applied to Industrial Processes". In: *XVIII Workshop on Information Processing and Control (RPIC).* Bahía Blanca, Argentina. (2019) pp. 31-36. `ISBN: 978-987-1648-44-3. DOI: 10.1109/RPIC.2019.8882143`.
- Emanuel Bernardi and Eduardo J. Adam. "Esquema de Detección y Diagnóstico de Fallas Basado en Observadores de Tiempo Discreto". In: *Jornadas de Ciencia y Tecnología 2019 de la UTN Facultad Regional San Francisco.* San Francisco, Argentina. (2019) pp. 110-111. `ISBN: 978-950-42-0193-9`.
- Carlos A. Cappelletti, Emanuel Bernardi, Hugo A. Pipino and Eduardo J. Adam. "Optimum Multiobjective Regulator with Variable Gain Matrix Applied to an Industrial Process". In: *2018 Argentine Conference on Automatic Control (AADECA).* Buenos Aires, Argentina (2018) pp. 67-72. `ISBN: 978-987-46859-0-2. DOI: 10.23919/ AADECA.2018.8577354`.
- Emanuel Bernardi and Eduardo J. Adam. "Reduced Order Observer Applied to a Linear Parameter Varying System with Unknown Input". *2018 Argentine Conference on Automatic Control (AADECA).* Buenos Aires, Argentina (2018) pp. 217-222. `ISBN: 978-987-46859-0-2. DOI: 10.23919/ AADECA.2018.8577337`.

Appendix A

State observer

A.1 Observability

This concept was introduced by Rudolf E. Kalman in [142], and it is related to the condition of observing or estimating the system state variables from its outputs, generally measurable. Essentially, a system is completely observable if every system state variable affects some of its outputs.

In practice, one of the main difficulties in a state feedback control implementation lies in the fact that the state variables are not accessible by direct measurement, therefore, to construct the corresponding control signals, it becomes necessary the estimation of non-measurable state variables. Such estimates are possible if, and only if, the system is completely observable [47, 58].

A state observer estimates the state variables based on the measurements of the output variables $y(t)$, and on the knowledge of its input variables $u(t)$. If the state observer captures all system state variables, regardless of whether some are available for direct measurement, it is called *full-order state observer.* On the other hand, if it does not estimate all system state variables, it is called *reduced-order state observer.*

Definition A.1 (Observability). *Considering an LTI system described by*

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{A.1}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}^p$ are the vector of states, inputs and outputs, respectively. Moreover, A, B, C and D are constant matrices of appropriate dimensions. The state $x(t_0)$ is said to be observable if given any input $u(t)$, there exists a finite time $t_f \geq t_0$, such that the knowledge of $u(t)$ for $t_0 \leq t \leq t_f$, the matrices A, B, C and D; and the output $y(t)$ for $t_0 \leq t \leq t_f$, are sufficient to determine $x(t_0)$. If every state of the system is observable for a finite t_f, the system is said to be completely observable, or simply, observable [58].

The Definition A.1, is the formal statement of observability. The following theorem introduces the necessary and sufficient conditions for the system to be observable.

Theorem A.1. *For the system described by Equation* (A.1) *to be completely observable, it is necessary and sufficient condition that the observability matrix* $\mathcal{O} \in \mathbb{R}^{np \times n}$ *has a row rank of* n*,*

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}. \tag{A.2}$$

The condition also refers that the pair (A, C) *is observable.*

Proof. As generally known, the solution of the Equation (A.1) is

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau,$$

and consequently, the output is,

$$y(t) = Ce^{At}x(0) + C\int_0^t e^{A(t-\tau)}Bu(\tau)d\tau + Du(t).$$

Thus, for observability evaluation purposes, it can be considered that the input $u(t)$ is null and by means of the Cayley-Hamilton theorem the output is,

$$y(t) = \sum_{k=0}^{n-1} \alpha_k(t)CA^k x(0),$$

or,

$$y(t) = \begin{bmatrix} \alpha_0(t) & \alpha_1(t) & \cdots & \alpha_{n-1}(t) \end{bmatrix} \begin{bmatrix} CA^0 \\ CA^1 \\ \vdots \\ CA^{n-1} \end{bmatrix} x(0).$$

Therefore, if all the state variables affect some of its outputs $y(t)$ (the system is completely observable) the rank of the matrix

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix},$$

should be n. □

A.2 Full-order state observer.

Given an LTI system described by the Equation (A.1), and assuming that the pair (A, C) is observable, it is possible to formulate a full-order state observer, proposed by Luenberger [86] and depicted in Figure A.1, as:

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L(y(t) - C\hat{x}(t)) \tag{A.3a}$$

$$\hat{y}(t) = C\hat{x}(t) \tag{A.3b}$$

being $\hat{x}(t)$ e $\hat{y}(t)$ the estimated state vectors and outputs, respectively. On the other hand, the L matrix corresponds to the feedback gain, to be suitable designed to achieve the desired observer behaviour. Thus, in the Equation (A.3a), the last term continuously corrects the difference between the measurable output $y(t)$ and its estimation $\hat{y}(t)$.

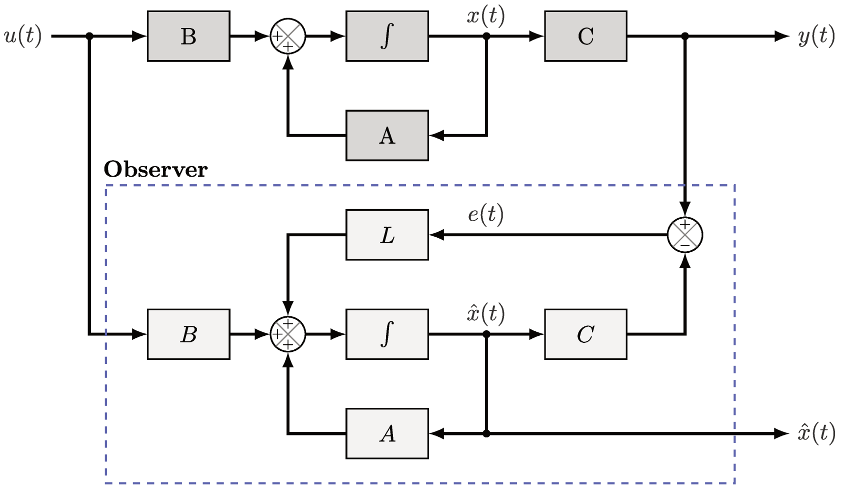


Figure A.1: Full-order state observer.

As a consequence, from combining the Equations (A.1) and (A.3), and defining the state estimation error vector $e(t) = x(t) - \hat{x}(t)$, the dynamic of the estimation error is obtained,

$$\dot{e}(t) = (A - LC)\, e(t) \tag{A.4}$$

thus, the matrix $(A - LC)$ determines the error vector dynamic behaviour. Consequently, if this matrix is stable (Hurwitz) the observer is stable, and the error vector converges asymptotically to zero,

$$\lim_{t\to\infty} e(t) = 0 \tag{A.5}$$

from any initial vector $e(0)$.

Appendix B

Parameterised Jacobian Linearisation

This technique is one of the most used methodologies to obtain LPV models [60, 143]. Specifically, considering a non-linear system where the following differential equations govern the system state variables evolution,

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) \\ y(t) &= g(x(t), u(t)) \end{aligned} \tag{B.1}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}^p$ are the state, input and output vectors, respectively. Moreover, $f(x(t), u(t)) : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ and $g(x(t), u(t)) : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^p$ are continuous smooth non-linear functions. The PJL technique is used to build an LPV model from the non-linear system equations. As a consequence, the arising model is formed by local approximations of the non-linear plant dynamics, around selected operating points. Therefore, the foundation of this method is the development of the first-order Taylor series expansion, with respect to the operating points $(\bar{x},\ \bar{u})$ (see Appendix B.1). Then, the combination of the resulting affine models compound a polytopic representation of the non-linear plant, not only on the set of linearisation points, but also on those contained (see Appendix B.3).

B.1 Taylor series expansion

Definition B.1 (Taylor series). *Let be $f(x)$ an analytic function defined on the open disk $|x - x_0| < R_0$ (centred at x_0 with radio R_0) in $\mathbb{C}$. Then, $f(x)$ admits power series representation as:*

$$f(x) = \sum_{n=0}^{\infty} a_n (x - x_0)^n \qquad (|x - x_0| < R_0),$$

where

$$a_n = \frac{f^{(n)}(x_0)}{n!} \qquad (n = 0,\ 1,\ 2,\ \ldots).$$

Thus, this power series representation is known as Taylor series expansion of n-order from the function $f(x)$ on the open disk $|x - x_0| < R_0$ centred at x_0. The special case $x_0 = 0$, is also called MacLaurin series.

B.2 Linearisation of non-linear systems

Based on the first-order Taylor series expansion (Definition B.1) and being $(\bar{x}, \bar{u})$ a fixed point. Therefore, the matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$ and $D \in \mathbb{R}^{p \times m}$ are defined by:

$$\begin{aligned} A &= \frac{\partial f(x(t), u(t))}{\partial x}|_{(\bar{x},\bar{u})}, & B &= \frac{\partial f(x(t), u(t))}{\partial u}|_{(\bar{x},\bar{u})}, \\ C &= \frac{\partial g(x(t), u(t))}{\partial x}|_{(\bar{x},\bar{u})}, & D &= \frac{\partial g(x(t), u(t))}{\partial u}|_{(\bar{x},\bar{u})}. \end{aligned}$$

Accordingly, the non-linear system from Equation (B.1) results in the following related system:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + \Delta_x \\ y(t) &= Cx(t) + Du(t) + \Delta_y \end{aligned} \tag{B.2}$$

where $\Delta_x = f(\bar{x}, \bar{u}) - (A\bar{x} + B\bar{u})$ and $\Delta_y = g(\bar{x}, \bar{u}) - (C\bar{x} + D\bar{u})$. Note that, when the fixed point $(\bar{x},\ \bar{u})$ corresponds with an equilibrium, the system from Equation (B.2) is linear. That is, $\Delta_x = 0$ and $\Delta_y = 0$. Lastly, it is also noted that the Taylor series expansion transforms non-linear terms, not related to vectors of inputs and states, into terms related to any of them.

B.3 Weighting functions

As previous sections stated, it is possible to represent, a non-linear system through an LPV model, using the PJL technique. For that, firstly a variable parameter vector $\zeta(t) \in \mathbb{R}^N$ must be defined, on which the system matrices depend. Secondly, the number of linearisation points per parameter L is adopted. It is important to note that the N and L dimension are closely related to the non-linear model representation capacity. That is, the amount of affine models need to proper represent the non-linear system through an LPV model, is $M = L^N$.

As an example, the development of weighting functions with dimensions $N = 2$ and $L = 3$ is presented. Thus, the L linearisation points per parameter are selected,

$$\rho_{j,1} = \min\{\zeta_j(t)\}, \qquad \rho_{j,2} = \mathrm{mid}\{\zeta_j(t)\}, \qquad \rho_{j,3} = \max\{\zeta_j(t)\}.$$

being $j \in \mathbb{Z}_{1:N}$. Then, the membership functions for each parameter are obtained,

$$M_{j,1}(\zeta_j(t)) = \frac{\rho_{j,1} - \zeta_j(t)}{\rho_{j,2} - \rho_{j,1}}, \quad M_{j,2}(\zeta_j(t)) = \frac{\rho_{j,2} - \zeta_j(t)}{\rho_{j,1} - \rho_{j,3}}, \quad M_{j,3}(\zeta_j(t)) = \frac{\zeta_j(t) - \rho_{j,3}}{\rho_{j,3} - \rho_{j,2}}.$$

Following, the M weighting functions corresponding to each affine model are defined as,

$$\mu_{k+L(l-1)}(\zeta(t)) = M_{1,l}(\zeta_1(t))M_{2,k}(\zeta_2(t)), \qquad \text{with } \ k \wedge l \in \mathbb{Z}_{1:L}.$$

Lastly, the polytopic LPV model is obtained,

$$\begin{aligned} \dot{x}(t) &= \sum_{i=1}^{M} \mu_i\left(\zeta(t)\right)\{A_i x(t) + B_i u(t) + \Delta x_i\} \\ y(t) &= \sum_{i=1}^{M} \mu_i\left(\zeta(t)\right)\{C_i x(t) + D_i u(t) + \Delta y_i\} \end{aligned} \tag{B.3}$$

where $\Delta_{x_i} = f(x_i, u_i) - (A_i x_i + B u_i)$ and $\Delta_{y_i} = g(x_i, u_i) - (C x_i + D u_i)$. Moreover, the variable indicated with a subscript i corresponds to its value at the i-th linearisation point, being $i \in \mathbb{Z}_{1:M}$.

Appendix C

Linear Matrix Inequalities

C.1 Introduction

Historically, the first LMI appeared in 1982, when the Russian mathematician Aleksandr Lyapunov presented his doctoral dissertation, titled: *"General Problem of the Stability of Motion"*[1]. There, the main theoretical basis of almost all controller design was introduced. Being this the well-known Lyapunov stability theory [66, 65]. Specifically,

Definition C.1. *Considering an autonomous continuous-time LTI system*

$$\dot{x}(t) = Ax(t) \tag{C.1}$$

with $x(t) \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n \times n}$. It is said to be asymptotically stable if and only if there exists a positive definite matrix P such that the following Lyapunov equation holds:

$$A^T P + PA < 0 \tag{C.2}$$

The requirement $P > 0$, $A^T P + PA < 0$ is usually called as Lyapunov inequality on P, which is a special form of an LMI. Lyapunov also showed that this LMI could be explicitly solved. Indeed, it is possible to choose any $Q = Q^T > 0$ and then solve the linear equation $A^T P + PA = -Q$ for the matrix P, which is guaranteed to be positive-definite if the system from Equation (C.1) is stable.

C.2 LMI regions

The concept of LMI region is also useful to formulate pole placement problems in terms of LMI, which can be solved efficiently using many existing software.

Definition C.2. *Let $\mathcal{D}$ be a region on the complex plane. If there exist matrices $L = L^T \in \mathbb{R}^{mxm}$ and $M \in \mathbb{R}^{mxm}$ such that:*

$$\mathcal{D} = \{s \mid s \in \mathbb{C} : L + sM + \bar{s}M^T < 0\}, \tag{C.3}$$

[1] Translated from Russian

then $\mathcal{D}$ is called LMI region and it is usually denoted by $\mathcal{D}_{(L,M)}$. Moreover, $f_{\mathcal{D}}(s) = L + sM + \bar{s}M^T$[2] *is called characteristic function of the LMI region $\mathcal{D}_{(L,M)}$.*

The following proposition reveals a fundamental property of LMI regions.

Proposition 1. *An LMI region is convex and symmetric about the real axis.*

Proof. The symmetry property is obvious. To show the convexity, two points $s_1, s_2 \in \mathcal{D}$ are arbitrarily chosen,

$$s^* = \theta s_1 + (1-\theta)s_2, \qquad 0 \le \theta \le 1.$$

Therefore, it has

$$L + s_1 M + \bar{s}_1 M^T < 0, \quad \text{and} \quad L + s_2 M + \bar{s}_2 M^T < 0.$$

Using this relationship it is derived in,

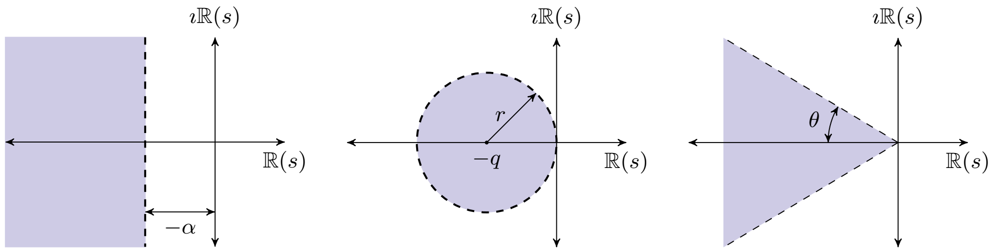
$$\begin{aligned} L + s^* M + \bar{s}^* M^T &= L + (\theta s_1 + (1-\theta)s_2)M + (\theta \bar{s}_1 + (1-\theta)\bar{s}_2)M^T \\ &= L + \theta s_1 M + \theta \bar{s}_1 M^T + (1-\theta)s_2 M + (1-\theta)\bar{s}_2 M^T \\ &= \theta(L + s_1 M + \bar{s}_1 M^T) + (1-\theta)L + s_2 M + \bar{s}_2 M^T \\ &< 0, \qquad\qquad\qquad \forall\ 0 \le \theta \le 1. \end{aligned}$$

Showing the convexity of the set $\mathcal{D}$ in the Equation (C.3). □

The LMI regions are quite general since its closure is the set of symmetrical convex regions with respect to the real axis. Indeed, LMI regions include relevant regions such as disks, cones, strips, etc., as well as any intersections of the above.

Examples

As the Definition C.2 states, an LMI region is a complex-plane region characterised by a function of s and $\bar{s}$. To illustrate this concept consider the following three complex-plane regions, depicted in Figure C.1.



(a) Half-plane.

(b) Disk centred at $(-q, 0)$ with radius r.

(c) Cone centred at the origin, with inner angle θ.

Figure C.1: LMI regions.

[2] $\bar{s}$ denotes the complex conjugate of s.

Firstly, the complex-plane region from the Figure C.1a represented by

$$\mathcal{D}_\alpha = \{x + \imath y \mid x < -\alpha < 0\},$$

is an LMI region because

$$\begin{aligned}
\mathcal{D}_\alpha &= \{x + \imath y \mid x < -\alpha\}\\
&= \{s \mid \mathbb{R}(s) < -\alpha\}\\
&= \left\{s \mid \frac{1}{2}(s + \bar{s}) < -\alpha\right\},
\end{aligned}$$

that is, $\mathcal{D}_\alpha$ is an LMI region with $L = 2\alpha$ and $M = 1$. Thus, the characteristic function results

$$f_{\mathcal{D}_\alpha}(s) = 2\alpha + s + \bar{s}.$$

Secondly, the complex-plane region from the Figure C.1b given by

$$\mathcal{D}_{q,r} = \{x + \imath y \mid (x + q)^2 + y^2 < r^2\},$$

is also an LMI region because

$$\begin{aligned}
\mathcal{D}_{q,r} &= \{x + \imath y \mid (x + q)^2 + y^2 < r^2\}\\
&= \{s \mid (s + q)(\bar{s} + q) < r^2\}\\
&= \left\{s \;\middle|\; \begin{bmatrix} -r & s + q \\ \bar{s} + q & -r \end{bmatrix} < 0\right\}\\
&= \left\{s \;\middle|\; \begin{bmatrix} -r & q \\ q & -r \end{bmatrix} + s\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + \bar{s}\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}^T < 0\right\},
\end{aligned}$$

that is, $\mathcal{D}_{q,r}$ is an LMI region with

$$L = \begin{bmatrix} -r & q \\ q & -r \end{bmatrix}, \qquad M = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

Lastly, the complex-plane region from the Figure C.1c performed by

$$\mathcal{D}_\theta = \{x + \imath y \mid |y| < -x\tan\theta\},$$

with $0 < \theta < \frac{\pi}{2}$, is an LMI region. In fact,

$$\begin{aligned}
\mathcal{D}_\theta &= \{x + \imath y \mid |y| < -x\tan\theta\}\\
&= \big\{\{x + \imath y \mid y^2 < x^2\tan^2\theta, \quad x\tan\theta < 0\big\}\\
&= \big\{x + \imath y \mid y^2\cos^2\theta < x^2\sin^2\theta, \quad x\sin\theta < 0\big\}\\
&= \left\{x + \imath y \;\middle|\; \begin{bmatrix} x\sin\theta & \imath y\cos\theta \\ -\imath y\cos\theta & x\sin\theta \end{bmatrix} < 0\right\}\\
&= \left\{s \;\middle|\; \begin{bmatrix} (s + \bar{s})\sin\theta & \imath(s - \bar{s})\cos\theta \\ (-s + \bar{s})\cos\theta & (s + \bar{s})\sin\theta \end{bmatrix} < 0\right\}\\
&= \left\{s \;\middle|\; s\begin{bmatrix} \sin\theta & \cos\theta \\ -\cos\theta & \sin\theta \end{bmatrix} + \bar{s}\begin{bmatrix} \sin\theta & \cos\theta \\ -\cos\theta & \sin\theta \end{bmatrix}^T < 0\right\},
\end{aligned}$$

that is, $\mathcal{D}_\theta$ is an LMI region with

$$L = 0, \qquad M = \begin{bmatrix} \sin\theta & \cos\theta \\ -\cos\theta & \sin\theta \end{bmatrix}.$$

C.3 Schur complement

Convex quadratic inequalities are converted to LMI form using Schur complements.

Definition C.3. *Let $Q(x) = Q(x)^T$, $R(x) = R(x)^T$, and $S(x)$ depend affinely on x. Then the LMI*

$$\begin{bmatrix} Q(x) & S(x) \\ S(x)^T & R(x) \end{bmatrix} > 0 \tag{C.4}$$

is equivalent to the matrix inequality

$$R(x) > 0,\ Q(x) - S(x)R(x)^{-1}S(x)^T > 0 \tag{C.5}$$

or equivalently,

$$Q(x) > 0,\ R(x) - S(x)^T Q(x)^{-1} S(x) > 0. \tag{C.6}$$

In other words, the set of non-linear inequalities from Equations (C.5) and (C.6) can be represented as the LMI of the Equation (C.4).

Appendix D

Performance Indices

The design of modern control systems requires a metric to evaluate its performance. Accordingly, the error and the time at which it occurs are the valuable factors to be considered. Thus, a performance index is a single measure of a system's performance that emphasises those characteristics of the response that are deemed to be important. In short, a performance index is a quantitative measure of the performance of a system and it is chosen so that emphasis is given to the important system specifications. Some of the most important indices are listed below:

- Integral Absolute Error (IAE)

$$\text{IAE} = \int_0^\infty |e(t)|dt,$$

- Integral of Squared Error (ISE)

$$\text{ISE} = \int_0^\infty e(t)^2 dt,$$

- Root Mean Square (RMS)

$$\text{RMS} = \sqrt{\frac{1}{n}\left(e(k+1)^2 + e(k+2)^2 + \ldots + e(k+n)^2\right)},$$

- Total Variance (TV)

$$\text{TV} = \sum |\delta u(k)| = \sum |u(k) - u(k-1)|,$$

- On-line Computational Effort (OCE)

$$\text{OCE} = \frac{\text{elapsed time}}{T_s} 100\%.$$

Bibliography

[1] Youmin Zhang and Jin Jiang. “Bibliographical Review on Reconfigurable Fault-Tolerant Control Systems”. In: *Annual Reviews in Control* 32.2 (2008), pp. 2–252. ISSN: 13675788. DOI: 10.1016/j.arcontrol.2008.03.008 (cit. on pp. 2, 4).

[2] Mogens Blanke, Michel Kinnaert, Jan Lunze, and Marcel Staroswiecki. *Diagnosis and Fault-Tolerant Control*. 3rd ed. Springer-Verlag, 2016. ISBN: 978-3-662-47942-1. DOI: 10.1007/978-3-662-47943-8 (cit. on pp. 2, 5, 12, 73, 74).

[3] Prashant Mhaskar, Jinfeng Liu, and Panagiotis D. Christofide. *Fault-Tolerant Process Control Methods and Applications*. Vol. 22. 1. Springer-Verlag, 2013. ISBN: 9781447148074 (cit. on p. 2).

[4] M. G. Mehrabi, A. G. Ulsoy, Y. Koren, and P. Heytler. “Trends and Perspectives in Flexible and Reconfigurable Manufacturing Systems”. In: *Journal of Intelligent Manufacturing* 13 (2002), pp. 135–146. ISSN: 09565515. DOI: 10.1023/A:1014536330551 (cit. on p. 2).

[5] Mufeed M. Mahmoud, Jin Jiang, and Youmin Zhang. *Active Fault Tolerant Control Systems-Stochastic Analysis and Synthesis*. Springer-Verlag, 2003. ISBN: 978-3-540-36283-8. DOI: 10.1007/3-540-36283-5 (cit. on p. 2).

[6] H. Rezaei and M. J. Khosrowjerdi. “A Polytopic LPV Approach to Active Fault Tolerant Control System Design for Three Phase Induction Motors”. In: *International Journal of Control* 90.10 (2017), pp. 2297–2315. ISSN: 13665820. DOI: 10.1080/00207179.2016.1244730. URL: http://dx.doi.org/10.1080/00207179.2016.1244730 (cit. on p. 2).

[7] Arslan Ahmed Amin and Khalid Mahmood Hasan. “A review of Fault Tolerant Control Systems: Advancements and applications”. In: *Measurement: Journal of the International Measurement Confederation* 143 (2019), pp. 58–68. ISSN: 02632241. DOI: 10.1016/j.measurement.2019.04.083 (cit. on p. 2).

[8] Emanuel Bernardi, Carlos Alberto Cappelletti, and Eduardo J. Adam. “Fault-tolerant Model Predictive Control Strategy Applied to Industrial Processes”. In: *2019 XVIII Workshop on Information Processing and Control (RPIC)*. Bahía Blanca, Argentina: Universidad Nacional del Sur, Sept. 2019, pp. 193–198. ISBN: 978-987-1648-44-3. DOI: 10.1109/RPIC.2019.8882143 (cit. on pp. 2, 3, 5, 73).

[9] Youmin Zhang and Jin Jiang. “Issues on integration of fault diagnosis and reconfigurable control in active fault-tolerant control systems”. In: *IFAC Proceedings Volumes (IFAC-PapersOnline)* 6.PART 1 (2006), pp. 1437–1448. ISSN: 14746670. DOI: `10.3182/20060829-4-cn-2909.00240` (cit. on p. 2).

[10] Karl Johan Aström and Bjorn Wittenmark. *Adaptive Control.* Dover Publications, 2008. ISBN: 9780486462783 (cit. on p. 3).

[11] Martín F. Picó and Eduardo J. Adam. “Fault Diagnosis and Tolerant Control Using Observer Banks Applied to Continuous Stirred Tank Reactor”. In: *Advances in Science, Technology and Engineering Systems Journal* 2.3 (2017), pp. 171–181. ISSN: 24156698. DOI: `10.25046/aj020322` (cit. on pp. 3, 5, 34).

[12] Youmin Zhang and Jin Jiang. “Active Fault-Tolerant Control System Against Partial Actuator Failures”. In: *IEEE Proceedings - Control Theory and Applications* 149.1 (2002), pp. 95–104 (cit. on p. 3).

[13] James B. Rawlings, David Q. Mayne, and Moritz M. Diehl. *Model Predictive Control: Theory, Computation, and Design.* Nob Hill Publishing, 2017. ISBN: 9780975937730 (cit. on pp. 3, 5, 18, 20, 21).

[14] David Q. Mayne. “Model predictive control: Recent developments and future promise”. In: *Automatica* 50.12 (2014), pp. 2967–2986. ISSN: 00051098. DOI: `10.1016/j.automatica.2014.10.128` (cit. on pp. 3, 19).

[15] A. Gambier, T. Miksch, and E. Badreddin. “Fault-Tolerant Control of a Small Reverse Osmosis Desalination Plant with Feed Water Bypass”. In: *2010 American Control Conference.* 2010, pp. 3611–3616. ISBN: 978-1-4244-7427-1. DOI: `10.1109/ACC.2010.5530596` (cit. on p. 3).

[16] R. Keller, Steven X. Ding, M. Müller, and D. Stolten. “Fault-Tolerant Model Predictive Control of a Direct Methanol-Fuel Cell System with Actuator Faults”. In: *Control Engineering Practice* 66.October 2016 (2017), pp. 99–115. ISSN: 09670661. DOI: `10.1016/j.conengprac.2017.06.008` (cit. on pp. 3, 5).

[17] Bin Yu, Youmin Zhang, and Yaohong Qu. “MPC-based FTC with FDD Against Actuator Faults of UAVs”. In: *ICCAS 2015 - 2015 15th International Conference on Control, Automation and Systems, Proceedings* (2015), pp. 225–230. DOI: `10.1109/ICCAS.2015.7364911` (cit. on p. 3).

[18] Rolf Isermann. *Fault-Diagnosis Systems: an Introduction from Fault Detection to Fault Tolerance.* 1st ed. Springer, 2006, pp. 1–475. ISBN: 3540241124. DOI: `10.1007/3-540-30368-5` (cit. on pp. 4, 5, 9, 16, 34).

[19] M. Steinberg. “Historical overview of research in reconfigurable flight control”. In: *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 219.4 (2005), pp. 263–275. ISSN: 09544100. DOI: `10.1243/095441005X30379` (cit. on p. 4).

[20] Jack McMahan. “Flight 1080”. In: *AIR LINE PILOT* (1978) (cit. on p. 4).

[21] R. J. Montoya, W. E. Howell, W. T. Bundick, A. J. Ostroff, R. M. Hueschen, and Christine M. Belcastro. *Restructurable Controls.* Tech. rep. 1983 (cit. on p. 4).

[22] Jan M. Maciejowski and Colin N. Jones. "MPC Fault-Tolerant Flight Control Case Study: Flight 1862". In: *Proceedings of the 5th IFAC symposium on fault detection, supervision and safety for technical processes* 36.5 (2003), pp. 119–124. ISSN: 14746670. DOI: `10.1016/S1474-6670(17)36480-7` (cit. on pp. 4, 5).

[23] Venkat Venkatasubramanian, Raghunathan Rengaswamy, Kewen Yin, and Surya N. Kavuri. "A Review of Process Fault Detection and Diagnosis Part I: Quantitative Model-Based Methods". In: *Computers and Chemical Engineering* 27.3 (2003), pp. 293–311. ISSN: 00981354. DOI: `10.1016/S0098-1354(02)00160-6` (cit. on p. 4).

[24] Venkat Venkatasubramanian, Raghunathan Rengaswamy, and Surya N. Kavuri. "A Review of Process Fault Detection and Diagnosis Part II : Qualitative Models and Search Strategies". In: *Computers & Chemical Engineering* 27.3 (2003), pp. 313–326. ISSN: 00981354. DOI: `10.1016/S0098-1354(02)00161-8` (cit. on p. 4).

[25] Venkat Venkatasubramanian, Raghunathan Rengaswamy, Surya N. Kavuri, and Kewen Yin. "A Review of Process Fault Detection and Diagnosis Part III: Process History Based Methods". In: *Computers & Chemical Engineering* 27.3 (2003), pp. 327–346. ISSN: 00981354. DOI: `10.1016/S0098-1354(02)00160-6` (cit. on p. 4).

[26] Dubravko Miljković. "Fault Detection Methods: A Literature Survey". In: *Proceedings of the 34th International Convention MIPRO* (2011), pp. 750–755 (cit. on pp. 4, 16).

[27] R. S. Sharma, L. Dewan, and S. Chatterji. "Fault Diagnosis Methods in Dynamic Systems: A Review". In: *International Journal of Electronics and Electrical Engineering* 3.6 (2015), pp. 465–471. ISSN: 2301380X. DOI: `10.12720/ijeee.3.6.465-471` (cit. on p. 4).

[28] Zhiwei Gao, Carlo Cecati, and Steven X. Ding. "A survey of fault diagnosis and fault-tolerant techniques-part I: Fault diagnosis with model-based and signal-based approaches". In: *IEEE Transactions on Industrial Electronics* 62.6 (2015), pp. 3757–3767. ISSN: 02780046. DOI: `10.1109/TIE.2015.2417501` (cit. on p. 4).

[29] Zhiwei Gao, Carlo Cecati, and Steven X. Ding. "A survey of fault diagnosis and fault-tolerant techniques-part II: Fault diagnosis with knowledge-based and hybrid/active approaches". In: *IEEE Transactions on Industrial Electronics* 62.6 (2015), pp. 3768–3774. ISSN: 02780046. DOI: `10.1109/TIE.2015.2419013` (cit. on p. 4).

[30] You-jin Park, Shu-kai S. Fan, and Chia-Yu Hsu. "A Review on Fault Detection and Process Diagnostics in Industrial Processes". In: *Processes* 8.9 (2020), p. 1123. ISSN: 2227-9717. DOI: `10.3390/pr8091123` (cit. on p. 4).

[31] Jie Chen and Ron J. Patton. *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Springer, 2012. ISBN: 9781461373445 (cit. on pp. 5, 34).

[32] Inseok Hwang, Sungwan Kim, Youdan Kim, and Seah Chze Eng. “A Survey of Fault Detection Isolation and Reconfiguration Methods”. In: *IEEE Transactions on Control Systems Technology* 18.3 (2010), pp. 363–653 (cit. on p. 5).

[33] Marcelo Menezes Morato, Daniel J. Regner, Paulo Renato Costa Mendes, Julio Elias Normey-Rico, and Carlos Bordons. “Fault Analysis, Detection and Estimation for a Microgrid via H2/H_inf LPV Observers”. In: *International Journal of Electrical Power and Energy Systems* 105 (2019), pp. 823–845. ISSN: 01420615. DOI: `10.1016/j.ijepes.2018.09.018` (cit. on pp. 5, 86, 89, 98, 99).

[34] Damiano Rotondo, Francisco Ronay López-Estrada, Fatiha Nejjari, Jean-Christophe Ponsart, Didier Theilliol, and Vicenç Puig. “Actuator multiplicative fault estimation in discrete-time LPV systems using switched observers”. In: *Journal of the Franklin Institute* 353.13 (2016), pp. 3176–3191. ISSN: 00160032. DOI: `10.1016/j.jfranklin.2016.06.007` (cit. on pp. 5, 16).

[35] T. Youssef, Mohammed Chadli, Hamid Reza Karimi, and R. Wang. “Actuator and sensor faults estimation based on proportional integral observer for TS fuzzy model”. In: *Journal of the Franklin Institute* 354.6 (2016), pp. 2524–2542. ISSN: 00160032. DOI: `10.1016/j.jfranklin.2016.09.020` (cit. on p. 5).

[36] Huayun Han, Ying Yang, Linlin Li, and Steven X. Ding. “Observer-based fault detection for uncertain nonlinear systems”. In: *Journal of the Franklin Institute* 355.3 (2018), pp. 1278–1295. ISSN: 00160032. DOI: `10.1016/j.jfranklin.2017.12.021` (cit. on p. 5).

[37] Ye Wang, Vicenç Puig, Feng Xu, and Gabriela Cembrano. “Robust fault detection and isolation based on zonotopic unknown input observers for discrete-time descriptor systems”. In: *Journal of the Franklin Institute* 356.10 (2019), pp. 5293–5314. ISSN: 00160032. DOI: `10.1016/j.jfranklin.2019.04.014` (cit. on p. 5).

[38] Masoud Pourasghar, Vicenç Puig, and Carlos Ocampo-Martinez. “Characterisation of interval-observer fault detection and isolation properties using the set-invariance approach”. In: *Journal of the Franklin Institute* 357.3 (2019), pp. 1853–1886. ISSN: 00160032. DOI: `10.1016/j.jfranklin.2019.11.027` (cit. on p. 5).

[39] Emanuel Bernardi, Marcelo Menezes Morato, Paulo Renato Costa Mendes, Julio Elias Normey-Rico, and Eduardo J. Adam. “Fault-Tolerant Energy Management for an Industrial Microgrid : A Compact Optimization Method”. In: *International Journal of Electrical Power and Energy Systems* 124 (2021), p. 106342. ISSN: 0142-0615. DOI: `10.1016/j.ijepes.2020.106342` (cit. on pp. 5, 85).

[40] Bruno M. Lima, Marcelo Menezes Morato, Paulo Renato Costa Mendes, and Julio Elias Normey-Rico. "Moving horizon estimation of faults in renewable microgrids". In: *IFAC-PapersOnLine* 52.1 (2019), pp. 311–316. ISSN: 24058963. DOI: `10.1016/j.ifacol.2019.06.080` (cit. on p. 5).

[41] Lei Zou, Zidong Wang, Jun Hu, and Donghua Zhou. "Moving Horizon Estimation with Unknown Inputs under Dynamic Quantization Effects". In: *IEEE Transactions on Automatic Control* 9286.c (2020), pp. 1–1. ISSN: 0018-9286. DOI: `10.1109/tac.2020.2968975` (cit. on p. 5).

[42] IFAC SAFE-PROCESS. *Bibliography Database.* 2013. URL: `https://tc.ifac-control.org/6/4/bibliography-database` (cit. on pp. 5, 9).

[43] Martín F. Picó. "Diseño de Controladores Tolerante a Fallas Aplicados a Procesos de la Industria Química". PhD thesis. Universidad Nacional del Litoral, 2015 (cit. on pp. 9, 16).

[44] Rolf Isermann. *Fault-Diagnosis Applications Model-Based Condition Monitoring: Actuators, Drives, Machinery, Plants, Sensors, and Fault-tolerant Systems.* Springer Science & Business Media, 2011. ISBN: 9783642127663 (cit. on p. 9).

[45] Janos J. Gertler. *Fault Detection and Diagnosis in Engineering Systems.* 1st ed. CRC Press, 1998. ISBN: 978-0-8247-9427-9 (cit. on pp. 10–12, 14, 16).

[46] Jian Zhang, Akshya Kumar Swain, and Kiong Sing Nguang. *Robust Observer-Based Fault Diagnosis for Nonlinear Systems Using MATLAB.* Springer International, 2016. ISBN: 978-3-319-32323-7 (cit. on p. 10).

[47] Katsuhiko Ogata. *Modern Control Engineering.* 5th ed. Pearson Education, 2010. ISBN: 9780136156734. DOI: `10.1109/TAC.1972.1100013` (cit. on pp. 11, 16, 18, 111).

[48] Scott D. Sagan. "The Problem of Redundancy Problem: Why More Nuclear Security Forces May Produce Less Nuclear Security". In: *Risk Analysis* 24.4 (2004), pp. 935–946. ISSN: 02724332. DOI: `10.1111/j.0272-4332.2004.00495.x` (cit. on p. 12).

[49] Janos J. Gertler. "Fault Detection and Isolation Using Parity Relations". In: *Control Engineering Practice* 5.5 (1997), pp. 653–661. ISSN: 0967-0661 (cit. on p. 14).

[50] Rolf Isermann, Ralf Schwarz, and Stefan Stölzl. "Fault Tolerant Drive-by-Wire Systems". In: *IEEE Control Systems Magazine* 22.October (2002), pp. 64–81. ISSN: 02721708. DOI: `10.1109/MCS.2002.1035218` (cit. on p. 16).

[51] Janos J. Gertler. "Survey of Model-Based Failure Detection and Isolation in Complex Plants". In: *IEEE Control Systems Magazine* 8.6 (1988). ISSN: 0272-1708. DOI: `10.1109/37.9163` (cit. on p. 16).

[52] Jie Zhang. "Expert Systems in On-line Process Control and Fault Diagnosis". PhD thesis. City University London, 1991 (cit. on p. 16).

[53] T. Jiang, K. Khorasani, and S. Tafazoli. "Parameter Estimation-Based Fault Detection, Isolation and Recovery for Nonlinear Satellite Models". In: *IEEE Transactions on Control Systems Technology* 16.4 (2008), pp. 799–808. ISSN: 1063-6536. DOI: 10.1109/TCST.2007.906317 (cit. on p. 16).

[54] Halim Alwi, Christopher Edwards, and Chee Pin Tan. *Fault Detection and Fault-Tolerant Control Using Sliding Modes*. Springer-Verlag, 2011. ISBN: 978-0-85729-649-8. DOI: 10.1007/978-0-85729-650-4 (cit. on p. 16).

[55] Emanuel Bernardi and Eduardo J. Adam. "Observer-based Fault Detection and Diagnosis Strategy for Industrial Processes". In: *Journal of the Franklin Institute* 357.14 (2020), pp. 9895–9922. ISSN: 0016-0032. DOI: 10.1016/j.jfranklin.2020.07.046 (cit. on pp. 16, 23).

[56] Hongwen He, Zhentong Liu, and Yin Hua. "Adaptive Extended Kalman Filter Based Fault Detection and Isolation for a Lithium-Ion Battery Pack". In: *Energy Procedia* 75 (2015), pp. 1950–1955. ISSN: 18766102. DOI: 10.1016/j.egypro.2015.07.230 (cit. on p. 16).

[57] Behrooz Safarinejadian and Elham Kowsari. "Fault detection in non-linear systems based on GP-EKF and GP-UKF algorithms". In: *Systems Science and Control Engineering* 2.1 (2014), pp. 610–620. ISSN: 21642583. DOI: 10.1080/21642583.2014.956843 (cit. on p. 16).

[58] Farid Golnaraghi and Benjamin C. Kuo. *Automatic Control Systems*. 9th ed. Wiley, 2009, p. 944. ISBN: 9780470048962 (cit. on pp. 16, 111).

[59] Eduardo J. Adam. *Instrumentación y Control de Procesos. Notas de Clase.* 3rd ed. Santa Fe: Ediciones UNL, 2018, p. 796. ISBN: 978-987-749-122-7 (cit. on pp. 16, 39).

[60] Corentin Briat. "Commande et Observation Robustes des Systèmes LPV Retardés". PhD thesis. Grenoble Institute of Technology, 2008 (cit. on pp. 16, 115).

[61] Javad Mohammadpour and Carsten W. Scherer. *Control of Linear Parameter Varying Systems with Applications*. Springer, 2012. ISBN: 9781461418320. DOI: 10.1007/978-1-4614-1833-7 (cit. on pp. 16, 17, 51).

[62] Guang-Ren Duan and Hai-Hua Yu. *LMIs in Control Systems*. Taylor & Francis, 2013. ISBN: 978-1-4665-8300-9 (cit. on p. 16).

[63] C. M. Astorga-Zaragoza, Didier Theilliol, Jean-Christophe Ponsart, and Mickael Rodrigues. "Observer synthesis for a class of descriptor LPV systems". In: *Proceedings of the American Control Conference*. 2011, pp. 722–726. ISBN: 9781457700804. DOI: 10.1109/acc.2011.5990835 (cit. on p. 16).

[64] Habib Hamdi, Mickael Rodrigues, Chokri Mechmeche, Didier Theilliol, and Naceur Benhadj Braiek. "Fault detection and isolation in linear parameter-varying descriptor systems via proportional integral observer". In: *International Journal of Adaptive Control and Signal Processing* 26.3 (2012), pp. 224–240. DOI: 10.1002/acs (cit. on p. 16).

[65] Pascal Gahinet, Arkadi Nemirovski, Alan J. Laub, and Mahmoud Chilali. *LMI Control Toolbox For Use with MATLAB.* Tech. rep. MathWorks, Inc., 1995 (cit. on pp. 16, 28, 29, 77, 119).

[66] Stephen P. Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear Matrix Inequalities in System and Control Theory.* Vol. 15. Society for Industrial and Applied Mathematics, 1994, p. 203. ISBN: 089871334X. DOI: `10.1109/TAC.1997.557595` (cit. on pp. 16, 27, 28, 32, 33, 56, 119).

[67] Damiano Rotondo. *Advances in Gain- Scheduling and Fault Tolerant Control Techniques.* Springer International, 2018. ISBN: 9783319629018 (cit. on p. 17).

[68] Jie Chen and Ron J. Patton. *Robust Model-Based Fault Diagnosis for Dynamic Systems.* Kluwer Academic Publishers, 1999. ISBN: 978-0-7923-8411-3 (cit. on p. 18).

[69] Julio Elias Normey-Rico and Eduardo F. Camacho. *Control of Dead-time Processes.* Springer, 2007. ISBN: 978-1-84628-828-9 (cit. on p. 18).

[70] Eduardo F. Camacho and Carlos Bordons. *Model Predictive Control.* Vol. 53. 9. Springer, 2007. ISBN: 978-1-85233-694-3. DOI: `10.1007/978-0-85729-398-5` (cit. on pp. 18, 51).

[71] David Q. Mayne and Paola Falugi. "Stabilizing conditions for model predictive control". In: *International Journal of Robust and Nonlinear Control* 29.4 (2019), pp. 894–903. ISSN: 10991239. DOI: `10.1002/rnc.4409` (cit. on p. 20).

[72] Antonio Ferramosca. "Model Predictive Control of Systems with Changing Setpoints". PhD thesis. Universidad de Sevilla, 2011, p. 255 (cit. on pp. 20, 21).

[73] Graham C. Goodwin, María M. Seron, and José A. De Dona. *Constrained Control and Estimation An Optimisation Approach.* Springer, 2004. ISBN: 1-85233-548-3 (cit. on pp. 20, 21).

[74] Marshall D. Rafal and William F. Stevens. "Discrete dynamic optimization applied to on-line optimal control". In: *AIChE Journal* 14.1 (1968), pp. 85–91. ISSN: 15475905. DOI: `10.1002/aic.690140117` (cit. on p. 21).

[75] Charles R. Cutler and L. B. Ramaker. "Dynamic Matrix Control - A Computer Control Agorithm". In: *86th National American Institute of Chemical Engineers* (Apr. 1979) (cit. on p. 21).

[76] Christopher V. Rao, James B. Rawlings, and Jay H. Lee. "Constrained linear state estimation - A moving horizon approach". In: *Automatica* 37.10 (2001), pp. 1619–1628. ISSN: 00051098. DOI: `10.1016/S0005-1098(01)00115-7` (cit. on p. 21).

[77] Angelo Alessandri and Moath Awawdeh. "Moving-horizon estimation with guaranteed robustness for discrete-time linear systems and measurements subject to outliers". In: *Automatica* 67 (2016), pp. 85–93. ISSN: 00051098. DOI: `10.1016/j.automatica.2016.01.015` (cit. on pp. 21, 96).

[78] Peter Kühl, Moritz Diehl, Tom Kraus, Johannes P. Schlöder, and Hans Georg Bock. "A real-time algorithm for moving horizon state and parameter estimation". In: *Computers and Chemical Engineering* 35.1 (2011), pp. 71–83. ISSN: 00981354. DOI: `10.1016/j.compchemeng.2010.07.012` (cit. on pp. 21, 52, 54).

[79] Christopher V. Rao and James B. Rawlings. "Constrained process monitoring: Moving-horizon approach". In: *AIChE Journal* 48.1 (2002), pp. 97–109. ISSN: 00011541. DOI: `10.1002/aic.690480111` (cit. on p. 21).

[80] Edward P. Gatzke and Francis J. Doyle. "Use of multiple models and qualitative knowledge for on-line moving horizon disturbance estimation and fault diagnosis". In: *Journal of Process Control* 12.2 (2002), pp. 339–352. ISSN: 09591524. DOI: `10.1016/S0959-1524(01)00037-3` (cit. on p. 22).

[81] Alberto Bemporad, Domenico Mignone, and Manfred Morari. "Moving horizon estimation for hybrid systems and fault detection". In: *Proceedings of the American Control Conference.* San Diego, 1999, pp. 2471–2475 (cit. on pp. 22, 86).

[82] Emanuel Bernardi and Eduardo J. Adam. "Reduced Order Observer Applied to a Linear Parameter Varying System with Unknown Input". In: *2018 Argentine Conference on Automatic Control (AADECA).* IEEE, Nov. 2018, pp. 1–6. ISBN: 978-9-8746-8591-9. DOI: `10.23919/AADECA.2018.8577337` (cit. on pp. 23, 25).

[83] Emanuel Bernardi, Hugo A. Pipino, and Eduardo J. Adam. "Full-order Output Observer Applied to a Linear Parameter Varying System with Unknown Input". In: *2020 IEEE Biennial Congress of Argentina (ARGENCON).* Resistencia, 2020. ISBN: 9781728159577 (cit. on p. 23).

[84] Pierre Apkarian, Pascal Gahinet, and Greg Becker. "Self-scheduled H_oo Control of Linear Parameter-varying Systems: a Design Example". In: *Automatica* 31.9 (1995), pp. 1251–1261. ISSN: 00051098. DOI: `10.1016/0005-1098(95)00038-X` (cit. on p. 24).

[85] Ming Hou and Peter C. Muller. "Design of Observers for Linear Systems with Unknown Inputs". In: *IEEE Transactions on Automatic Control* 37.6 (June 1992), pp. 871–875. DOI: `10.1109/9.256351` (cit. on p. 25).

[86] David G. Luenberger. "Observing the State of a Linear System". In: *IEEE Transactions on Military Electronics* 8.2 (1964), pp. 74–80. ISSN: 05361559. DOI: `10.1109/TME.1964.4323124` (cit. on pp. 26, 112).

[87] Mohamed Darouach, Michel Zasadzinski, and Shi Jie Xu. "Full-order observers for linear systems with unknown inputs". In: *IEEE Transactions on Automatic Control* 39.3 (1994), pp. 606–609. ISSN: 0018-9286 (cit. on p. 30).

[88] Johan Löfberg. "YALMIP : a Toolbox for Modeling and Optimization in MATLAB". In: *2004 IEEE International Conference on Robotics and Automation* (2004), pp. 284–289. ISSN: 03014215. DOI: `10.1109/CACSD.2004.1393890` (cit. on pp. 32, 34, 67, 77, 97).

[89] J. Duane Morningred, Bradley E. Paden, Dale E. Seborg, and Duncan A. Mellichampt. "An Adaptive Nonlinear Predictive Controller". In: *Chemical Engineering Science* 47.4 (1992), pp. 755–762 (cit. on p. 44).

[90] Marcelo Menezes Morato, Hugo A. Pipino, Emanuel Bernardi, Diego M. Ferreyra, Eduardo J. Adam, and Julio Elias Normey-Rico. "Sub-optimal Linear Parameter Varying Model Predictive Control for Solar Collectors". In: *2020 IEEE International Conference on Industrial Technology (ICIT).* Buenos Aires: IEEE, Feb. 2020, pp. 95–100. ISBN: 978-1-7281-5754-2. DOI: `10.1109/ICIT45562.2020.9067139` (cit. on pp. 51, 61).

[91] Hugo A. Pipino, Emanuel Bernardi, Marcelo Menezes Morato, Carlos Alberto Cappelletti, Eduardo J. Adam, and Julio Elias Normey-Rico. "Formulación de un LPV-MPC Adaptativo para Procesos Industriales No Lineales". In: *2020 Argentine Conference on Automatic Control (AADECA).* 2020, pp. 195–200. ISBN: 978-987-46859-3-3 (cit. on p. 51).

[92] Hugo A. Pipino, Marcelo Menezes Morato, Emanuel Bernardi, Eduardo J. Adam, and Julio Elias Normey-Rico. "Nonlinear Temperature Regulation of Solar Collectors with a Fast Adaptive Polytopic LPV MPC Formulation". In: *Solar Energy* 209 (2020), pp. 214–225. ISSN: 0038-092X. DOI: `10.1016/j.solener.2020.09.005` (cit. on p. 51).

[93] Frank Allgöwer and Alex Zheng. *Nonlinear Model Predictive Control.* Birkhäuser, 2000. ISBN: 978-3-0348-9554-5 (cit. on p. 51).

[94] Boris Houska, Hans Joachim Ferreau, and Moritz Diehl. "ACADO toolkit - An open-source framework for automatic control and dynamic optimization". In: *Optimal Control Applications and Methods* 32.3 (2011), pp. 298–312. DOI: `10.1002/oca.939` (cit. on p. 51).

[95] Bartosz Kapernick and Knut Graichen. "The gradient based nonlinear model predictive control software GRAMPC". In: *2014 European Control Conference, ECC 2014* 3 (2014), pp. 1170–1175. DOI: `10.1109/ECC.2014.6862353` (cit. on p. 51).

[96] Jeff S. Shamma. "An Overview of LPV Systems". In: *Control of Linear Parameter Varying Systems with Applications.* 2012, pp. 3–26. ISBN: 9781461418320 (cit. on p. 51).

[97] Marcelo Menezes Morato, Paulo Renato Costa Mendes, Julio Elias Normey-Rico, and Carlos Bordons. "LPV-MPC fault-tolerant energy management strategy for renewable microgrids". In: *International Journal of Electrical Power and Energy Systems* 117 (2020), p. 105644. ISSN: 01420615. DOI: `10.1016/j.ijepes.2019.105644` (cit. on pp. 52, 86, 96).

[98] James B. Rawlings and Bhavik R. Bakshi. "Particle filtering and moving horizon estimation". In: *Computers and Chemical Engineering* 30.10-12 (2006), pp. 1529–1541. ISSN: 00981354. DOI: 10.1016/j.compchemeng.2006.05.031 (cit. on pp. 52, 54).

[99] Marcelo Menezes Morato, Paulo Renato Costa Mendes, Julio Elias Normey-Rico, and Carlos Bordons. "LPV-H_inf Fault Estimation for Boilers in Sugarcane Processing Plants". In: *IFAC-PapersOnLine* 51.26 (2018), pp. 1–6. ISSN: 24058963. DOI: 10.1016/j.ifacol.2018.11.177 (cit. on pp. 53, 61).

[100] Pablo Gonzalez Cisneros, Aadithyan Sridharan, and Herbert Werner. "Constrained Predictive Control of a Robotic Manipulator using quasi-LPV Representations". In: *IFAC-PapersOnLine* 51.26 (2018), pp. 118–123. ISSN: 24058963. DOI: 10.1016/j.ifacol.2018.11.158 (cit. on pp. 53, 56, 62).

[101] Daniel Limon, Ignacio Alvarado, Teodoro Alamo, and Eduardo F. Camacho. "MPC for tracking piecewise constant references for constrained linear systems". In: *Automatica* 44.9 (2008), pp. 2382–2387. ISSN: 00051098. DOI: 10.1016/j.automatica.2008.01.023 (cit. on pp. 54, 62).

[102] Antonio Ferramosca, Daniel Limon, Ignacio Alvarado, Teodoro Alamo, and Eduardo F. Camacho. "MPC for tracking of constrained nonlinear systems". In: *Proceedings of the IEEE Conference on Decision and Control* May 2014 (2009), pp. 7978–7983. ISSN: 01912216. DOI: 10.1109/CDC.2009.5400618 (cit. on pp. 55, 62).

[103] Marcelo Menezes Morato, Julio Elias Normey-Rico, and Olivier Sename. "Novel qLPV MPC Design with Least-Squares Scheduling Prediction". In: *IFAC-PapersOnLine* 52.28 (2019), pp. 158–163. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2019.12.366 (cit. on pp. 56, 64, 66).

[104] Sammyak Mate, Hariprasad Kodamana, Sharad Bhartiya, and P. S.V. Nataraj. "A Stabilizing Sub-Optimal Model Predictive Control for Quasi-Linear Parameter Varying Systems". In: *IEEE Control Systems Letters* 4.2 (2019), pp. 402–407. ISSN: 24751456. DOI: 10.1109/LCSYS.2019.2937921 (cit. on p. 56).

[105] Mircea Lazar, W. P.M.H. Heemels, S. Weiland, and Alberto Bemporad. "Stabilizing Model Predictive Control of Hybrid Systems". In: *IEEE Transactions on Automatic Control* 51.11 (2006), pp. 1813–1818. ISSN: 00189286. DOI: 10.1109/TAC.2006.883059 (cit. on p. 56).

[106] Franco Blanchini and Stefano Miani. *Set-Theoretic Methods in Control.* 9783319179322. 2015, pp. 1–630. ISBN: 9783319179322 (cit. on p. 56).

[107] Mayuresh V. Kothare, Venkataramanan Balakrishnan, and Manfred Morari. "Robust Constrained Model Predictive Control using Linear Matrix Inequalities". In: *Automatica* 32.10 (1996), pp. 1361–1379. ISSN: 0005-1098 (cit. on p. 56).

[108] Zhaoyang Wan and Mayuresh V. Kothare. “An efficient off-line formulation of robust model predictive control using linear matrix inequalities”. In: *Automatica* 39.5 (2003), pp. 837–846. ISSN: 00051098. DOI: 10.1016/S0005-1098(02)00174-7 (cit. on p. 56).

[109] Mircea Lazar. “Model Predictive Control of Hybrid Systems: Stability and Robustness”. PhD thesis. Technische Universiteit Eindhoven, 2006. ISBN: 9789038618234 (cit. on p. 56).

[110] Pablo Gonzalez Cisneros, Sophia Voss, and Herbert Werner. “Efficient Nonlinear Model Predictive Control via quasi-LPV representation”. In: *2016 IEEE 55th Conference on Decision and Control, CDC 2016* (2016), pp. 3216–3221. DOI: 10.1109/CDC.2016.7798752 (cit. on p. 56).

[111] Marcelo Menezes Morato, Julio Elias Normey-Rico, and Olivier Sename. “Model predictive control design for linear parameter varying systems: A survey”. In: *Annual Reviews in Control* 49.June (2020), pp. 64–80. ISSN: 13675788. DOI: 10.1016/j.arcontrol.2020.04.016 (cit. on p. 56).

[112] Carlos Alberto Cappelletti, Emanuel Bernardi, Hugo A. Pipino, and Eduardo J. Adam. “Optimum Multiobjective Regulator with Variable Gain Matrix Applied to an Industrial Process”. In: *2018 Argentine Conference on Automatic Control (AADECA)*. IEEE, Nov. 2018, pp. 1–6. ISBN: 978-9-8746-8591-9. DOI: 10.23919/AADECA.2018.8577354 (cit. on p. 56).

[113] Johan Löfberg. “Oops! i cannot do it again: Testing for recursive feasibility in MPC”. In: *Automatica* 48.3 (2012), pp. 550–555. ISSN: 00051098. DOI: 10.1016/j.automatica.2011.12.003 (cit. on pp. 57, 61).

[114] Franco Blanchini. “Set invariance in control”. In: *Automatica* 35.11 (1999), pp. 1747–1767. ISSN: 00051098. DOI: 10.1016/S0005-1098(99)00113-2 (cit. on p. 62).

[115] Eduardo F. Camacho, Manuel Berenguel, Francisco R. Rubio, and Diego Martinez. “Control Issues in Solar Systems”. In: *Control of Solar Energy Systems*. Springer, London, 2012, pp. 25–47. ISBN: 9780857299161. DOI: 10.1007/978-0-85729-916-1{_}2 (cit. on pp. 63, 66).

[116] R. M. Dell and D. A.J. Rand. “Energy storage - A key technology for global energy sustainability”. In: *Journal of Power Sources* 100.1-2 (2001), pp. 2–17. ISSN: 03787753. DOI: 10.1016/S0378-7753(01)00894-1 (cit. on p. 63).

[117] M. Pasamontes, J. D. Álvarez, J. L. Guzmán, Manuel Berenguel, and Eduardo F. Camacho. “Hybrid modeling of a solar-thermal heating facility”. In: *Solar Energy* 97 (2013), pp. 577–590. ISSN: 0038092X. DOI: 10.1016/j.solener.2013.09.024 (cit. on pp. 64, 65, 67).

[118] A. J. Gallego, F. Fele, Eduardo F. Camacho, and L. Yebra. “Observer-based Model Predictive Control of a parabolic-trough field”. In: *Solar Energy* 97 (2013), pp. 426–435. ISSN: 0038092X. DOI: 10.1016/j.solener.2013.09.002 (cit. on p. 64).

[119] Gary Ampuño, Lidia Roca, Juan D. Gil, Manuel Berenguel, and Julio Elias Normey-Rico. “Apparent delay analysis for a flat-plate solar field model designed for control purposes”. In: *Solar Energy* 177.November 2018 (2019), pp. 241–254. ISSN: 0038092X. DOI: 10.1016/j.solener.2018.11.014 (cit. on pp. 64, 65).

[120] José D. Vergara-Dietrich, Marcelo Menezes Morato, Paulo Renato Costa Mendes, Alex Amadeu Cani, Julio Elias Normey-Rico, and Carlos Bordons. “Advanced chance-constrained predictive control for the efficient energy management of renewable power systems”. In: *Journal of Process Control* 74 (2019), pp. 120–132. ISSN: 09591524. DOI: 10.1016/j.jprocont.2017.11.003 (cit. on pp. 66, 89, 91, 98, 100).

[121] S. Rosiek, J. Alonso-Montesinos, and F. J. Batlles. “Online 3-h forecasting of the power output from a BIPV system using satellite observations and ANN”. In: *International Journal of Electrical Power and Energy Systems* 99.November 2017 (2018), pp. 261–272. ISSN: 01420615. DOI: 10.1016/j.ijepes.2018.01.025 (cit. on p. 66).

[122] César Hernández-Hernández, Francisco Rodríguez, José Carlos Moreno, Paulo Renato Costa Mendes, and Julio Elias Normey-Rico. “The use of model predictive control (MPC) in the optimal distribution of electrical energy in a microgrid located in southeastern of Spain: A case study simulation”. In: *Renewable Energy and Power Quality Journal* 1.15 (2017), pp. 221–226. ISSN: 2172038X. DOI: 10.24084/repqj15.278 (cit. on p. 67).

[123] Emanuel Bernardi and Eduardo J. Adam. “Nonlinear Fault-tolerant Model Predictive Control Strategy for Industrial Processes”. In: *2020 Argentine Conference on Automatic Control (AADECA)*. 2020, pp. 105–110. ISBN: 978-987-46859-3-3. DOI: 10.23919/AADECA49780.2020.9301630 (cit. on p. 73).

[124] Emanuel Bernardi and Eduardo J. Adam. “Fault-tolerant predictive control based on linear parameter varying scheme for industrial processes”. In: *Journal of Taiwan Institute of Chemical Engineers* 10.3 (2021). DOI: 10.1016/j.jtice.2021.10.003 (cit. on p. 73).

[125] Peter Palensky and Dietmar Dietrich. “Demand side management: Demand response, intelligent energy systems, and smart loads”. In: *IEEE Transactions on Industrial Informatics* 7.3 (2011), pp. 381–388. ISSN: 15513203. DOI: 10.1109/TII.2011.2158841 (cit. on p. 86).

[126] Pierluigi Mancarella. “MES (multi-energy systems): An overview of concepts and evaluation models”. In: *Energy* 65 (2014), pp. 1–17. ISSN: 03605442. DOI: 10.1016/j.energy.2013.10.041 (cit. on p. 86).

[127] Robert H. Lasseter and Paolo Piagi. “Microgrid: A Conceptual Solution”. In: *2004 IEEE 35th Annual Power Electronics Specialists Conference*. 2004. DOI: 10.1109/PESC.2004.1354758 (cit. on p. 86).

[128] Esther Mengelkamp, Johannes Gärttner, Kerstin Rock, Scott Kessler, Lawrence Orsini, and Christof Weinhardt. “Designing microgrid energy markets: A case study: The Brooklyn Microgrid”. In: *Applied Energy* 210 (2018), pp. 870–880. ISSN: 03062619. DOI: 10.1016/j.apenergy.2017.06.054 (cit. on p. 86).

[129] Marcelo Menezes Morato, Paulo Renato Costa Mendes, Alex Amadeu Cani, Julio Elias Normey-Rico, and Carlos Bordons. “Future Hybrid Local Energy Generation Paradigm for the Brazilian Sugarcane Industry Scenario”. In: *International Journal of Electrical Power and Energy Systems* 101 (2018), pp. 139–150. ISSN: 01420615. DOI: 10.1016/j.ijepes.2018.03.024 (cit. on pp. 86–88, 90, 97).

[130] Mario Petrollese, Luis Valverde, Daniele Cocco, Giorgio Cau, and José Guerra. “Real-time integration of optimal generation scheduling with MPC for the energy management of a renewable hydrogen-based microgrid”. In: *Applied Energy* 166 (2016), pp. 96–106. ISSN: 03062619. DOI: 10.1016/j.apenergy.2016.01.014 (cit. on p. 86).

[131] Paulo Renato Costa Mendes, Jose M. Maestre, Carlos Bordons, and Julio Elias Normey-Rico. “A practical approach for hybrid distributed MPC”. In: *Journal of Process Control* 55 (2017), pp. 30–41. ISSN: 09591524. DOI: 10.1016/j.jprocont.2017.01.001. URL: http://dx.doi.org/10.1016/j.jprocont.2017.01.001 (cit. on p. 86).

[132] Pablo Báez-González, Alejandro J. del Real, Miguel A. Ridao Carlini, and Carlos Bordons. “Day-ahead economic optimization of energy use in an olive mill”. In: *Control Engineering Practice* 54 (2016), pp. 91–103. ISSN: 09670661. DOI: 10.1016/j.conengprac.2016.05.019 (cit. on p. 86).

[133] Alessandra Parisio, Christian Wiezorek, Timo Kyntäjä, Joonas Elo, Kai Strunz, and Karl Henrik Johansson. “Cooperative MPC-Based Energy Management for Networked Microgrids”. In: *IEEE Transactions on Smart Grid* 8.6 (2017), pp. 3066–3074. ISSN: 19493053. DOI: 10.1109/TSG.2017.2726941 (cit. on p. 86).

[134] Yan Zhang, Tao Zhang, Rui Wang, Yajie Liu, and Bo Guo. “Optimal operation of a smart residential microgrid based on model predictive control by considering uncertainties and storage impacts”. In: *Solar Energy* 122 (2015), pp. 1052–1065. ISSN: 0038092X. DOI: 10.1016/j.solener.2015.10.027 (cit. on p. 86).

[135] Ionela Prodan, Enrico Zio, and Florin Stoican. “Fault tolerant predictive control design for reliable microgrid energy management under uncertainties”. In: *Energy* 91 (2015), pp. 20–34. ISSN: 03605442. DOI: 10.1016/j.energy.2015.08.009 (cit. on p. 86).

[136] Ionela Prodan and Enrico Zio. “A model predictive control framework for reliable microgrid energy management”. In: *International Journal of Electrical Power and Energy Systems* 61 (2014), pp. 399–409. ISSN: 01420615. DOI: 10.1016/j.ijepes.2014.03.017 (cit. on p. 86).

[137] J. J. Marquez, A. Zafra-Cabeza, and C. Bordons. "Diagnosis and fault mitigation in a microgrid using model predictive control". In: *2018 International Conference on Smart Energy Systems and Technologies, SEST 2018 - Proceedings.* IEEE, 2018, pp. 1–6. ISBN: 9781538653265. DOI: `10.1109/SEST.2018.8495846` (cit. on p. 86).

[138] Marcelo Menezes Morato, Paulo Renato Costa Mendes, Julio Elias Normey-Rico, and Carlos Bordons. "Optimal operation of hybrid power systems including renewable sources in the sugar cane industry". In: *IET Renewable Power Generation* 11.8 (2017), pp. 1237–1245. ISSN: 17521424. DOI: `10.1049/iet-rpg.2016.0443` (cit. on pp. 89, 94–96, 98, 100).

[139] Tor A. Johansen, Dan Sui, and Roar Nybo. "Regularized nonlinear moving-horizon observer with robustness to delayed and lost data". In: *IEEE Transactions on Control Systems Technology* 21.6 (2013), pp. 2114–2128. ISSN: 10636536. DOI: `10.1109/TCST.2012.2228652` (cit. on p. 96).

[140] Marcelo Menezes Morato, Paulo Renato Costa Mendes, Julio Elias Normey-Rico, and Carlos Bordons. "Robustness conditions of LPV fault estimation systems for renewable microgrids". In: *International Journal of Electrical Power and Energy Systems* 111 (2019), pp. 325–350. ISSN: 01420615. DOI: `10.1016/j.ijepes.2019.04.014` (cit. on p. 96).

[141] LLC Gurobi Optimization. *Gurobi Optimizer Reference Manual.* 2020. URL: `http://www.gurobi.com` (cit. on p. 97).

[142] Rudolf Emil Kalman. *On the General Theory of Control Systems.* 1959. DOI: `10.1109/TAC.1959.1104873` (cit. on p. 111).

[143] Kazuo Tanaka and Hua O. Wang. *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach.* Wiley, 2001. ISBN: 0471323241. DOI: `10.1002/0471224596` (cit. on p. 115).