ENHANCED MODEL REFERENCE ADAPTIVE CONTROLLERS FOR FAULT-TOLERANT CONTROLS IN INDUSTRIAL APPLICATIONS

PhD Thesis presented by

JORGE RODRÍGUEZ-GUERRA

Directed by

Dr. CARLOS CALLEJA ELCORO
Dr. ANA MARÍA MACARULLA ARENAZA

ikerlan
MEMBER OF BASQUE RESEARCH & TECHNOLOGY ALLIANCE

Universidad de Deusto
University of Deusto
Deusto

September 25, 2020

Abstract

Industrial systems, similarly to natural ecosystems, are continually fluctuating. Both environments are moulded by internal and external factors, conditioning the inhabiting species lifestyle. The manufacturing machines and the living beings populating factories and landscapes, respectively, suffer continuous transformations to accommodate their features to the locations they occupy. This constant adaptation process has promoted the emergence of characteristics in the individuals that strengthen their survivability.

These advantages increase populations lifespan, especially the mechanisms diagnosing dangerous situations to draw up a contingency plan. Biological species present several techniques to ensure that their individuals survive despite the emergence of a dire circumstance. Manufacturing machines have imitated these organic procedures to maintain the production stable despite the emerge of malfunctioning sources on their components.

Fault-Tolerant Control techniques introduce a revolutionary approach to detect, measure and isolate faults on a first instance during a phase called Fault Detection and Isolation, and, to reconfigure the control algorithm in a second instance during a phase called Control Redesign. Both phases represent the mechanism previously mentioned, as this technique analyses the drawback source and updates the system performance accordingly to bypass the injury. This reconfiguration procedure encapsulates an Adaptive Control algorithm that modifies the controller response attending to the new dynamics or faults in the manufacturing machine.

Between the multiple configurations for these algorithms, Model Reference Adaptive Controllers offers the most suitable platform. However, despite their benefits, industrial systems require higher levels of flexibility, robustness and upgradability hardly achievable by conventional MRAC architectures. This research project has been focused on improving the current structure, enhancing its three major components:

1. **Model Based Technology:** Biological species learn from their parents the abilities to survive in the environment. Similarly, manufacturing machines follow the signals from the reference model to reduce the tracking error.

This accommodation process substitutes the transfer function describing the closed-loop system response with a Digital-Twin connected to a replica of the original control algorithm. This enhanced model increases MRACs flexibility, as they adapt to a more extensive number of fault scenarios.

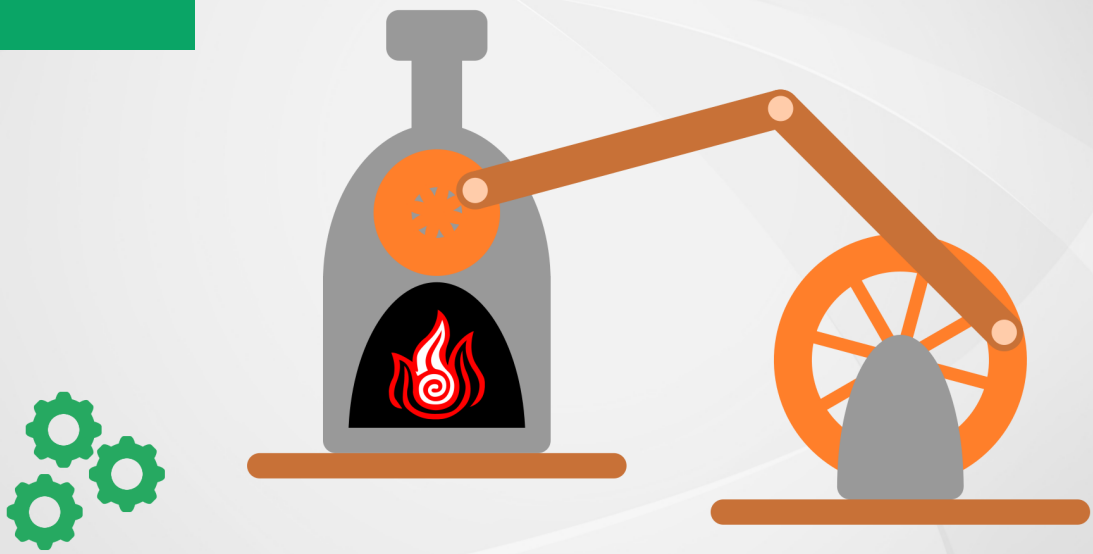
2. **Adaptive Control Algorithms:** Biological species present healing capabilities, recovering their bodies from injuries and wounds after short-term periods. Similarly, manufacturing machines controllers adjust their performance to the fault dynamics in the plant through the adaptive gains. Across each iteration in the production cycle, these gains accommodate the controller responses to minimise the tracking error, recovering the system optimal performance. Besides, an additional loop ensures the reference model is adapted to the faulty mechanical limits. This enhanced algorithm increases MRACs robustness, as it maintains the system stable without interrupting the production independently of the fault harm grade.

3. **Advance Manufacturing Techniques:** Biological species inhabit ecosystems in communities, distributing their roles accordingly to the individuals' particular features. Similarly, manufacturing machines are affected by multiple faults with heterogeneous sources. Instead of designing a unique adaptive gain to surpass each incident, a Bank of Controllers structure switches between these parameters to the optimal situation. This enhanced platform increases MRACs upgradability, as the system increases the number of fault scenarios surpassed optimally introducing additional adaptive gains into the Bank of Controllers.

Through these enhances, MRACs present an architecture compatible with industrial systems. This novel methodology has been validated against a Hydraulic-Press explicitly designed for this research project. The experiments carried out to corroborate the investigation confirm the initial statement, in the presence of faults, the control algorithm detects, measure and isolate its source, reconfiguring the controller afterwards to surpass it without spreading its damage.

Contents

1 Introduction



Biologic species have evolved from a common ancestor that populates the Earth around 3800 and 3500 million years ago. Each descendant inherits the genotype from their predecessors, the code that entangles the phenotype or physical characteristics of the individual. This cyclical process conceals the engine that has fueled the natural environment progress; only the best-adapted individuals survive and transfer their advantages to the next generation.

Similarly to Sisifo from Efira myth, this evolution and adaptation process generates an endless loop, where the previous generation has to survive in an aggressive environment, transferring their genetic advantages to the following generation. This ouroboros is deeply tattooed in biologic species survivability mechanism, becoming a natural source of knowledge for humankind. Guided by Palas Atena wisdom, they have investigated their features, and inspired by Apolo, they have replicated their virtues on their advantage. As a result, Hephaestus has forged the factories, the human counterpart to the natural environment.

Despite the rhetoric or over-complicated the previous ideas sound, they hide a

more straightforward concept, factories have suffered an evolution process prompted by the social necessities. Humans have imitated the behaviour of natural species, adapting their performance to the artificial world. If molars smash food to convert it into a digestible viscous paste, hydraulic-presses compress grapes to extract their juice.

This research project has gone further into this idea, presenting a novel methodology to confer regeneration properties to industrial machines. Athletes are subjected to high physical and psychological pressure during competitions, pushing their bodies to the limit. Every step matters to reach the finishing line. They train their physiology to continue performing under their maximum capacity although they are fatigued. Similarly, the enhances proposed into manufacturing machines maintain the production despite the emergence of faults that would neglect their proper performance.

The regeneration mechanism has been replicated through Fault-Tolerant Control (FTC) techniques, presenting an industrial controller based on Model Reference Adaptive Techniques (MRAC). This algorithm has suffered several enhances, such as substituting the mathematical models for Digital-Twins (DTs), improving the reliability through adaptation rules that accommodate the mechanical behaviour to the fault status and a Bank of Controllers (BCs) architecture to switch the controller accordingly to the fault case.

1.1 Hypothesis

On real-life, living species are able to survive when they suffer injuries due to their auto-healing capacities, and thus, they ensure their survivability and avoid being hunted by predators. Similarly, in the industrial world, machines are threatened by several factors that harm them and neglect their execution. Hence, the ultimate chore of this thesis is to replicate the regeneration mechanism of living creatures within industrial machines, adapting their performance to ensure they can survive their menaces.

Fault-Tolerant Control theory take care of industrial species altering their controllers accordingly to the wound suffered, minimising their harming effect and recovering an optimal performance without slowing-down the manufacturing process. They reinforce machines detecting, grading and isolating the faults during Fault Detection and Isolation (FDI) phase and adjusting the controller signal during Control Redesign (CR) phase [5]. This thesis has encountered a research niche on Model Reference Adaptive Control algorithms to alter machine controllers and recover the optimal performance without interrupting the production.

Conventional control architectures, such as PID algorithms, already reduce some adversities while they continue tracking the reference. However, faults alter machine behaviour beyond the controller recovery point, as they transform the plant dynamics, reduce the production and neglect operators safety. Adaptive Controls (ACs) offer a suitable alternative to deal with faults without spreading their harming effect [53, 57, 63]. This thesis sought an FTC methodology, whose

CR phase integrates AC algorithms based on MRACs. The following research question summarised this objective:

How Model Reference Adaptive Controllers are enhanced to surpass faults, introducing them into the Control Redesign phase of a Fault-Tolerant Control for Industrial Machines?

To answer this research question, three main hypothesis have been stated:

1. **Hypothesis:** Bringing in Digital-Twins into the Model Reference stage of MRACs increases its *flexibility* as they provide a modular platform adaptable to mechanical loads and parameters irregularities.
2. **Hypothesis:** Improving the Adaptation Mechanism of MRACs with novel approaches based on Anti-Windup (AW) techniques increase their *robustness* for industrial environment.
3. **Hypothesis:** Integrating Bank of Controllers algorithms into Adaptive Controllers improve their performance as they would have an *upgradable* platform that selects the proper adaptive gains to surpass the fault.

Each hypothesis has its doppelganger on MRACs structure (see Fig. 1.1):

1. **Model Reference:** Composing an adaptable platform that alters its behaviour accordingly to the current production and the manufacturing cycle. The model replicates the machine components to reproduce the plant responses in Real-Time (RT).
2. **Adaptation Mechanism:** Avoiding the fault's spreading to other components or machines through a controller. This algorithm is prepared to maintain the system operating under acceptable conditions without exceeding actuators mechanical limits.
3. **Adjustable Controller:** Implementing a compatible platform with industrial machines, switching between Adaptive Controllers without interrupting the production. The platform selects the optimal control loop to surpass the fault.

Conventional MRAC resembles natural species adaptation mechanism, transforming the controller behaviour accordingly to the current plant dynamics. Nonetheless, these techniques become obsolete under the hostile industrial environment. On this threatening atmosphere, standalone AC algorithms attempt to retrieve the optimal behaviour against several fault sources simultaneously. FTC paradigm concentrates the recovery process against the most harmful situation.

This thesis focuses on the fault's prevention but also tackles other conventional MRACs vulnerabilities. The MRACs adaptive mechanism attains the minimum error between plant and model responses. The *flexibility* enhancement substitutes the mathematical model and describes the system closed-loop responses with

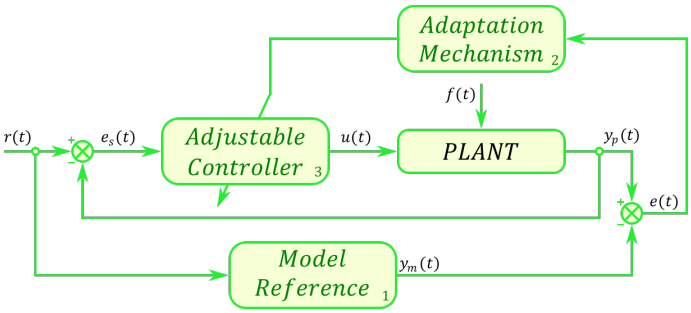


Figure 1.1: Conventional MRAC structure.

a dynamical platform. As a consequence, the original behaviour is identically replicated in the Model Reference platform.

When a fault emerges on an industrial system, their constructive properties also suffer a deviation. Controller positions previously reachable become the current mechanical limits. The *robustness* enhancement sought an adaptive controller that adjusts automatically their performance attending to the latest plant dynamics and limitations.

Due to machines prolonged lifespan, they require maintenance policies to ensure proper behaviour. During these examinations, operators adjust the mechanical components, altering their response slightly. They also customise controllers parameters accordingly to the current machine status. These alterations also affect the adaptation mechanism, needing an *upgradable* platform that transforms the control algorithm quickly to minimise the time spent under maintenance.

The enhances proposed to MRACs maintain their essence, adapting controllers signals accordingly to the latest plant dynamics, while they introduce them into FTC paradigm:

1. **Model Based Technology:** In the Model Reference stage, a Digital-Twin represents the desired system closed-loop performance, reproducing in Real-Time identical responses as the ones offered by the plant. The model operates under a Hardware in the Loop (HiL) validation platform compatible with the industrial environment.

2. **Adaptive Control Algorithms:** In the Adaptation Mechanism stage, the rules of the Massachusetts Institute of Technology (MIT) or Lyapunov lessen the fault spread adapting the controller signal accordingly to the latest plant dynamics. Besides this principal branch, an additional one adjusts the reference signal attending to the plant mechanical limits under the fault effect.

3. **Advanced Manufacturing Techniques:** In the Adjustable Controller stage, a Bank of Controllers switches automatically between MRACs adaptive gains attending to the current fault. This structure divides the controllers between branches coordinated by a common mechanism, expanding them when operators design new algorithms for the system.

After these renewal process (see Fig. 1.2), MRACs would maintain their chore, but they are also compatible with the industrial environment. This thesis reviews the literature to understand the current status of FTC techniques and AC algorithms [96, 116, 56], deciding the departure point (see Section 2). Afterwards, the research carried out to improve MRACs accordingly to the previous enhances is presented (see Section 3), testing their viability against a Hydraulic-Press (HP) (see Section 4). Finally, the conclusion drawn from the research work are expounded (see Section 5).

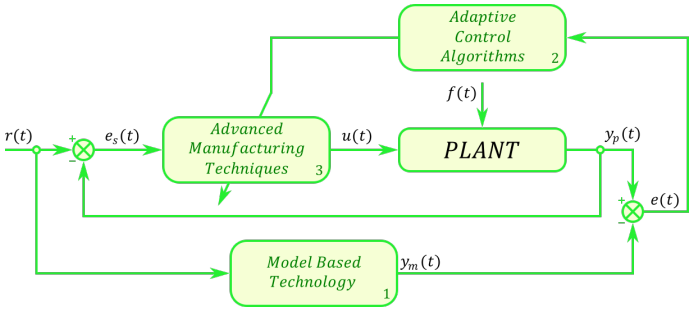


Figure 1.2: MRAC structure updated with the novel features presented in this thesis.

1.2 Scope

1.2.1 Fault-Tolerant Control

Conventional MRACs maintain the system controlled despite variances on the plant dynamics; however, factories have a more hostile environment threatening the system stability. The enhances brought to MRACs reduce these uncertainties, opening the gate to include these techniques into other control algorithms, such as the CR phase of FTCs. The design procedure for industrial controllers has influenced this thesis, so an expert manufacturer has provided their knowledge regarding this subject.

Fagor Arrasate S. Coop., a leading hydraulic-press manufacturer installed in Basque Country, has supported this thesis contributing with their knowledge in this kind of industrial systems. Their expertise in the field of heavy industrial machines has settled the bases for the MRACs enhancements. They have offered their material and installations to generate a testbench in the laboratory to study the controller performance under identical conditions as they would have on a factory.

On this context, the research process has been initiated analysing the latest FTC techniques. Similarly to MRACs, in order to export their features into industrial machines, the controllers have to ensure the safety protocols and prevent operators neglects. This thesis has successfully combined the FTC features with the industrial environment requirements, improving their main phases:

- **Fault Detection and Isolation:** Classical approaches detect, quantify and isolate faults through mathematical equations. This thesis introduces a novel feature based on Neural-Networks (NNs), whose neurons have been trained through a fault database in a controlled experiment employing the validation platform, recreating these adverse conditions, storing the machine behaviour for each situation.

- **Control Redesign:** Conventional approaches divide this phase attending to the controller behaviour against the fault. An external algorithm masks the drawback or transforms its internal structure, distinguishing between control accommodation or reconfiguration, respectively. MRACs adjust the controller signals without altering its initial gains, that is to say, they accommodate the control algorithm response.

Factories have several machines running in parallel, accomplishing their assignment in the manufacturing process through their unique features. This heterogeneous environment challenges the discovery of a standard FTC methodology harmonious with each system. This thesis sought a rule to consolidate industrial controllers under this paradigm, that is to say, the methodology has to be compatible with industrial machines independently of their original control algorithm. The approach proposed bases its control redesign phase onto MRACs improved with model-based technology.

Rather than replicating mathematical models simulating the whole plant dynamics, this thesis has researched a manufacturing component repository easily configurable through data-sheet documents. Instead of complex laboratory experiments to determine constructive characteristics, operators configure the model behaviour replicating the parameters provided in these technical documents (standard for each component).

Operators configure low-level industrial components, instead of high-level manufacturing machines. After parametrising them, they proceed to connect their terminals replicating the machine assembling process on factories. This procedure provides them with a Digital-Twin that replicates the industrial system performance. This software platform also serves as an initial testbench for the model behaviour, contrasting the responses obtained against its real counterpart.

After validating the DT signals, operators continue the study exporting it into a Hardware in the Loop validation platform. This thesis has presented solutions based on common manufacturing platforms, such as the Industrials PCs or PLCs. This environment provided operators with a hardware testbench, where they attach the current controller installed on the machine directly to the DT. Due to the Real-Time performance, this methodology substitutes the physical manufacturing machines.

This validation platform standardises MRACs for manufacturing machines. Independently of their features, operators would tune the machine controllers in a virtual commissioning stage. The methodology also improves conventional MRACs, as they would examine the current system performance against this upgradeable DT instead of the mathematical approximation replicating its dynamics.

This thesis has studied the limitations of this methodology against a Hydraulic-Press placed on Fagor Arrasate's factory. This manufacturing machine would become the system under test, upgrading their current PID controllers with MRACs configured following this methodology. The study would attempt to prove how the novelties introduced into FTC techniques reduce the fault impact on manufacturing machines without losing feasibility, endangering operators safety or lessening the production.

1.2.2 Fault Recovery Process

Factories present a hostile environment for machine controllers, as they are prone to suffer from noise disturbance, static electricity, mechanical collision and other drawbacks that diminish their effectiveness. Conventional controllers, such as PID architectures, mitigate these adverse effects, maintaining the system in proper performance. Nonetheless, these structures are inefficient when faults spread their harming effect, degrading the machine performance from nominal behaviour to failure status.

When a disturbance appears on the system, PID architectures maintain the stability and recover the optimal performance integrating the error between the plant output and the reference signal in a closed-loop configuration. Although this architecture mitigates the harming effect of the external signals, they are unable to regain stability when there are alterations on the plant dynamics. In these cases, instead of recovering an optimal performance, the controller pushes the system towards the instability, harming the machine and endangering operators safety.

This thesis proposes a four-stage process to surpass faults and maintain the system operating without pausing the machine. The method presented in the subsequent sections mimics the stages in FTC techniques. It sought to avoid halting the production and reducing products quality, as it keeps the machine under optimal conditions (see Fig. 1.3). The following points describe the recovery process through the performance of a hypothetical Hydraulic-Press.

1.2.2.1 Nominal Behaviour

Initially, the machine performs its cycle without suffering any disturbance on the so-called nominal behaviour. During this stage, the system remains unaffected by faults, so both control architectures, PID algorithms and FTC techniques maintain the stability.

The HP cycle matches the optimal performance, behaving as the model reference signal for MRAC in the subsequent stages of the recovery process. The Digital-Twin replicates this non-faulty behaviour, as it reproduces the closed-loop responses of the original plant and controller. The inertia maintains the system under this status until any external or internal condition alters it.

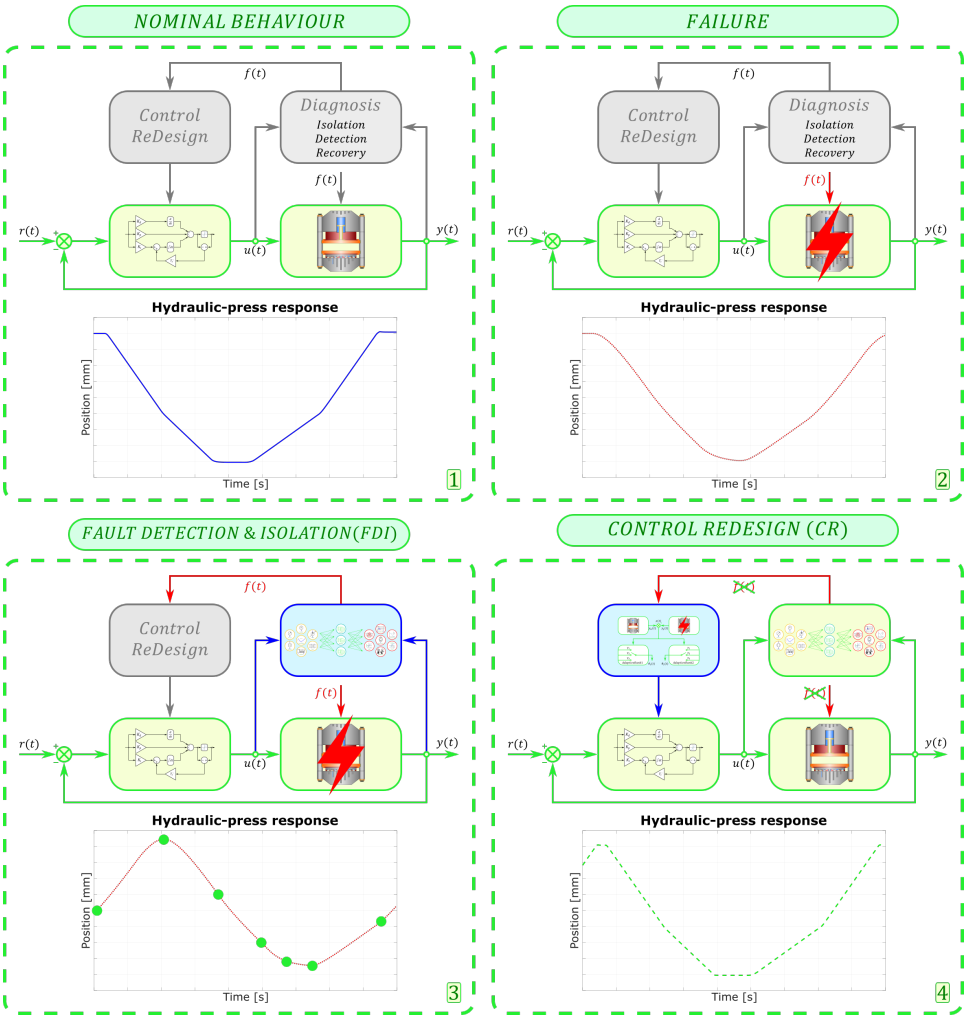


Figure 1.3: Stages followed in the fault recovery process.

1.2.2.2 Failure

Machines are continually evolving, adapting their production to the manufacturing cycle requirements. Although these adjustments modify their performance, only when this change has negative consequences on the production, it is considered a fault. These variations have a broad harm grade, as they englobe from soft faults repaired by the PID architecture to more severe ones that can even break the machine. This thesis focuses on the intermediate range, faults that degrade the system performance without compromising its stability, such as mechanical fatigue and internal leaks between components.

These faults typically emerge on the machine components, degrading its per-

formance slightly at the beginning and spreading their harming effect until the system ceases providing service. Maintenance policies reduce this drawback, as operators analyse the machine components, repairing them when required. Although these practices reduce the fault emergence ratio, they interrupt the manufacturing process. Predictive maintenance policies decrease the unproductive periods planning the most favourable opportunity to accomplish these assignments.

Despite there are mathematical methods that prognosticate the fault emergence with high accuracy, a spontaneous development ruins these methodologies. As an example, this thesis studies the effect of faults on the HP actuators, the proportional valves. Their opening degradation rate is well-known by manufacturers and engineers, due to the laboratory experiments carried out. The curve estimates the number of cycles before the mechanical fatigue starts reducing the proportional valve opening, establishing the maintenance policies accordingly to these diagrams. As additional factors are amplifying this adverse effect, such as humidity or temperature, these forecasts tend to be unsuccessful, triggering the fault earlier than predicted [114, 54, 30].

Mechanical fatigue faults degrade the system performance slowly, as they deteriorate the proportional valve opening continuously (see Fig. 1.3). During the first cycles, its adverse effect remains unnoticeable by the controller, but after a certain amount of iterations, it neglects the control algorithm attempts to recover the optimal behaviour. Fault-Tolerant Controls lessens this adverse effect, including on the recovery process two additional phases, Fault Detection and Isolation (see Section 1.2.2.3) and Control Redesign (see Section 1.2.2.4).

1.2.2.3 Fault Detection and Isolation

Factories have several mechanisms involved in product quality control, such as artificial vision cameras inspecting for defects during the production or scientist detecting anomalies in the merchandise through laboratory experiments. These mechanisms ensure an identical production rate. Analogical methods identify relatively easy alterations on the stock; however, they are unable to discriminate between product deformity or machine fault.

Whereas the first situation only requires to discard the deformed product, the second case implies that the machine is malfunctioning. The time spent under these degraded conditions is crucial to avoid spreading the fault. The Fault Detection and Isolation phase englobes the period since the fault emerges until the controller addresses it. Traditionally, these techniques based their discovery mechanism in mathematical algorithms; however, they have a reduced efficiency for industrial systems due to the continuous variations on its manufacturing cycles.

Instead of finding a unique mathematical expression that discriminates faults from nominal behaviour, this thesis presents a novel approach based on Neural-Networks. This technique compares the current machine performance against a stored version of the system under the fault effect. This methodology is feasible since industrial processes replicate a cyclical behaviour, that is to say, they reproduce similar responses when they cross concrete points or stages during the product manufacture.

Despite their benefits, this methodology only has fruitful responses after training the neurons with realistic data of the machine performance under the fault status. This drawback has been mitigated introducing Digital-Twins to replicate the system behaviour with high accuracy. Due to these models, operators generate the fault database storing the information gathered from these virtual sensors under the Hardware in the Loop validation platform.

For each fault candidate, operators repeat several experiments until obtaining an approximate model of the system behaviour under the fault status. This cloud of possible machine fault situations is employed to train the NN. During the detection and isolation algorithm execution, the manufacturing machine compares the current measures obtained from its sensors against their analogues stored during the experiments to determine if the system is under a fault. When this condition is reached, the mechanism proceeds to isolate its source and generate a report for the following stage.

As a recapitulation, when the system reaches these control points, an initial NN detects if the measure belongs to the nominal behaviour or a fault case. After ending each complete iteration, a second NN verifies if there are false positives or a fault has really emerged. In the first scenario, the machine continues manufacturing pieces. While in the second scenario, the FDI technique detects, measures and isolates the fault, collecting this information and beginning the Control Redesign phase.

1.2.2.4 Control Redesign

Faults are far from being an unknown drawback for control designers. They have tuned conventional control architectures, such as PIDs, to deal with uncertainties, fluctuations and noise in plants, at least to some extent. When these effects emerge on the machine, the closed-loop architecture integrates the error between the operator's reference signal and the plant output, recovering the nominal behaviour in a redesign approach known as *Passive Fault-Tolerant Control*.

Although this methodology deals effectively with soft faults, these techniques are inefficient when the drawback alters the plant dynamics, as the controller favours the error instead of mitigating its adverse effect. The emergence of these dynamics in the plant alters the closed-loop poles, obtaining a system that makes obsolete the gains tuned by control designers to compensate for the tracking error. This situation leads the system to an instability status, being necessary to introduce novel approaches to redesign the controller gains, such as the *Active Fault-Tolerant Control*.

The latest CR phase has two alternatives, accommodation and reconfiguration. The first one masks the control signal, adapting the standard controller output to maintain the stability notwithstanding the fault behaviour. In contrast, the second one tunes new controller gains to balance the current plant dynamics. Control designers pick up the alternative that better suits their application.

This thesis has studied a solution based on the first approach, where Adaptive Controllers play a crucial role. The system recovers from the fault through a Model Reference Adaptive Control algorithm. This technique has a versatile

platform prepared to minimise the error between industrial and model plant, altering the controller response accordingly. Due to the hostile industrial environment, this thesis has enhanced the conventional MRAC structure with DT models (see Section 1.2.1).

When the FDI phase detects an emerging fault, the MRAC mechanism initiates the adaptive process, adjusting the controller response through the adaptive gains to recover an optimal performance. After some cycles, the system recovers from the fault (see Fig. 1.3), surpassing it without human interaction and without spreading their harmful effect to other components.

1.3 Thesis Outline

The thesis has been organised into five chapters, whose main features have been summarised in the subsequent sections.

1.3.1 Introduction

Industrial systems suffer from mechanical fatigue, component malfunction and other drawbacks considered as faults. They modify nominal machine behaviour, reducing the controller effectiveness or, in extreme cases, cancelling them. Fault-Tolerant Control theory presents an alternative to avoid interrupting the production detecting, grading and isolating the faults during Fault Detection and Isolation phase and transforming the controller during Control Redesign phase. This thesis focuses on this last phase, introducing Adaptive Control (AC) techniques to surpass faults and retrieve the optimal system performance.

Across this chapter, the dissertation objectives, scope and outline are presented, answering the research hypothesis in the process. The CR phase of FTCs in industrial systems is improved through a Model Reference Adaptive Control enhanced with new features that increase its flexibility, robustness and upgradability.

1.3.2 State of Art

Before introducing novelties into Fault-Tolerant Controllers, the current techniques have been reviewed, presenting the information gathered in the State of Art section. This study discriminates between three main areas of expertise. The first notion presented submits general concepts and terminology of FTCs. The subsequent notion analyses the features of their principal phases, FDI and CR, while the latest notion proposes multiple AC architectures, explaining why MRACs have been chosen as a suitable technique to surpass faults in industrial systems.

1.3.3 Methodology and Research

This chapter has presented the techniques, materials, research and methodology required during this thesis to accomplish with the investigation, dividing the information into each enhanced MRAC stage.

Part 1: *Model Based Technology*

This section presents a new library designed to reproduce in Real-Time the behaviour of industrial components, parametrising them through the information brought by manufacturers in data-sheets. Generating heavy machinery Digital-Twins consumes a significant amount of project budget. With the novel techniques presented, rather than creating mathematical models representing the system closed-loop response, control designers assemble virtual machines dragging components from the library and plugging them in the virtual environment. Finally, they test its behaviour in a Software in the Loop (SiL) validation platform, subsequently verifying the design in a Hardware in the Loop validation platform.

Part 2: *Adaptive Control Algorithms*

This section constitutes the dissertation research chore, explaining the enhances introduced into MRACs to make them compatible with industrial machines. MRACs reduce the tracking error between model and system response altering the controller signal accordingly to the faulty plant dynamics. The control algorithm modifies its signal attending to the adaptive gains, a predefined variable whose value is determined by *Massachusetts Institute of Technology* (MIT) or *Lyapunov* rules. Also, several limitations of MRACs regarding unreachable positions or stability quandaries are studied. These drawbacks are surpassed when an Adaptive Reference Model is included into MRACs algorithm.

Part 3: *Advanced Manufacturing Techniques*

Manufacturing machines are subjected to multiple fault sources, such as mechanical fatigue, internal leaks or noise dissonance. Although MRACs treat faults as uncertainties, each case requires their adaptive gains to be surpassed efficiently. Instead of a unique mechanism for each fault case, this thesis presents a novel methodology based on a Bank of Controllers. This mechanism ensures that faults are surpassed by the adaptive gains that minimise their harming effect more efficiently and in the shortest periods.

1.3.4 Novel MRAC Features

This chapter corroborates the methodology and the enhances previously presented, testing its performance against an industrial machine.

Part 1: *Model Based Technology*

This part presents a Digital-Twin (DT) prepared to reproduce in Real-Time (RT) the behaviour of industrial systems. The virtual model substitutes the mathematical equation describing the system closed-loop behaviour on MRAC's Model Reference stage. The enhance compares the current machine response against the DT signals, determining the error between the desired output and the system response.

Part 2: *Adaptive Control Algorithms*

Depending on the system properties, there are multiple approaches to introduce the Adaptation Mechanism. Across these pages, the author has determined the adaptive gains for MIT and Lyapunov rules, studying the fastest alternative to recover which one offers a faster recovery attending to each fault. Also, this part covers the system performance against robustness issues after including an adaptive mechanism to the reference signal.

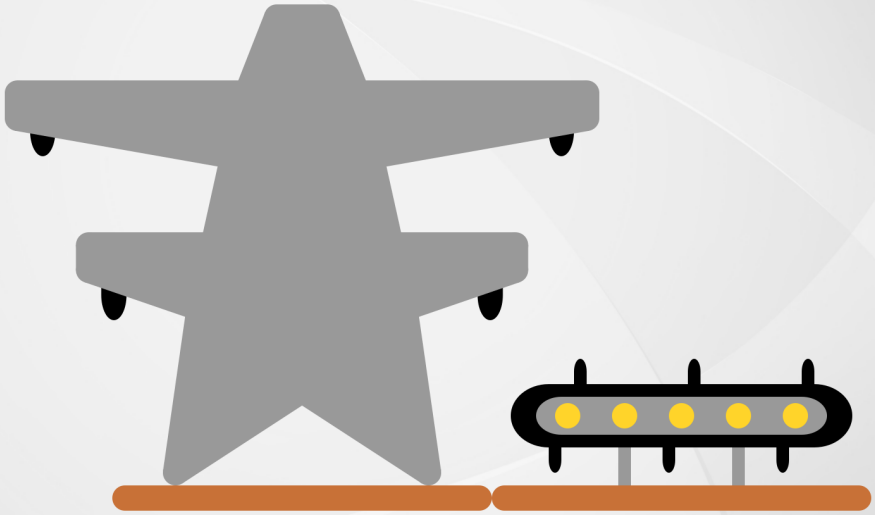
Part 3: *Advanced Manufacturing Techniques*

After including a Bank of Controllers into the adaptation mechanism, the overall system performance against faults increases, tuning the adaptive gains accordingly to the fault. Due to this mechanism, the controller adapts its behaviour accordingly to the fault, encountering the gains that surpass the fault more efficiently. Besides, this platform allows testing novel features introduced to the controllers quickly, substituting the older gains for the new tuned ones.

1.3.5 Conclusions and Future Work

On this final chapter, the conclusions drawn from the thesis are presented, studying the benefits brought by the improvements to the Model Reference Adaptive Control structure in the field of Fault-Tolerant Controllers. This research project reduces the fault effect in a single machine; nonetheless, modern production lines are composed of several ones. This situation offers a new field of study, introducing, for example, Game Theory (GT) algorithms as a novel feature for FTC techniques enabling production optimisation even when the industrial systems are under the effect of faults.

2 State of Art



Natural environments are shifting continuously, forcing in their fauna and flora a progressive evolution to ensure their survivability. Industrial environments, despite their artificial inception, suffer a similar adaptation process, reducing production times and charges at the expense of enhancing machines. These novel features require sophisticated controllers, opening the gate to advance control procedures.

Across this chapter, the reader would encounter two advance control procedures, fault-tolerant techniques (see Section 2.1 and 2.2) and adaptive algorithms (see Section 2.3). The first technique prepares manufacturing machines to detect unanticipated malfunctions, adjusting their controllers automatically to surpass this drawback without interrupting the production. The second algorithm adapts the controller signal progressively to the faulty behaviour.

2.1 Fault-Tolerant Control

Modern industry sought for advanced manufacturing techniques, where Fault-Tolerant Controllers would play a crucial role. These algorithms reduce the unproductive times as they maintain the machine operating despite the fault emergence.

2.1.1 Overview

Industrial systems have suffered multiple improvements across the years to increase productivity without reducing quality. Scientist and engineers have researched for more reliable controllers to maintain the machine operating even when a failure appears in the manufacturing process [57, 5]. In this context, Fault-Tolerant Control offers a two-phases methodology prepared to discern faults when they emerge, correcting them without human interaction and decreasing the unproductive periods [135, 5, 50].

These techniques originate multiple benefits to industries. Concerning the economic aspect, as machine failures do not completely shut-down manufacturing lines, the production losses are reduced. Furthermore, FTC techniques lessen interruptions in primary services, such as power grids, transportation systems, water supplies and communications [75, 62, 59, 66, 68]. Not only do these techniques occasion benefits beyond the financial perspective, but also they increase machines security and reliability, neglecting jeopardising operators safety.

FTC techniques strengthen safety protocols of critical manufacturing processes, such as nuclear and chemical plants. They have demonstrated their efficacy lessening the negative effect of non-critical faults; however, they are inefficient for complete system shut-downs. Therefore, they complement safety protocols, never replacing them. As a summary, the following points resume their benefits [135]:

- Providing system availability when a fault occurs.
- Preventing a single fault from turning into system failure.
- Reducing hardware redundancy in favour of duplicating information to detect faults.
- Reconfiguring system components assisted by the fault accommodation mechanism.
- Maintaining the system reliability despite having a degraded performance.
- Decreasing hardware investment substituting mechanical elements with virtual ones.

These techniques have engaged multiple sectors, such as flight controllers, aerospace industry or automotive engine manufactures [130, 137, 151, 36]. They have applied FTCs as a supervising technique to monitor the system behaviour and to bypass its failure. Despite their benefits, these enhance also deploys more resources during the commissioning stage [132, 140]. Developing an early study to answer the following questions reduces this financial breach:

- How critical is the component?
- How likely is the component to fail?
- How expensive is to make the component fault-tolerant?

This study reflects which components would benefit the most from these improvements, so they are prone to include FTC techniques when they receive positive feedback in each question [5]. For instance, sensors electromagnetic noise is a widely extended fault for industrial machines; nonetheless, conventional PID control architectures already surpass these drawbacks. On the contrary, pipeline leaks are infrequent in water supply circuits but could have a devastating effect on the system.

Attending to how the fault compromises the system performance, components become fault-tolerant through one of the following approaches [5]:

- **Replication:** The controller gathers information from multiple sources (actuators and sensors) and compares these signals against a model replicating the optimal system behaviour. When the responses have a discordance, the control algorithm understands that a fault has emerged and varies its gains to surpass it.
- **Redundancy:** Multiple identical instances of the system operate in parallel, comparing their signals continuously to detect the fault emergence. When this condition arises, the controller isolates the harmed component, switching automatically to their replica.
- **Diversity:** Multiple measure mechanisms analyse the system behaviour in parallel for early fault detection. When they discern a component malfunctioning, they modify the performance of another element to assume its assignments without spreading the fault.

Independently of the approach selected, systems become fault-tolerant when they accomplish these properties. This thesis states a novel methodology for FTC basing the redesign mechanism in Adaptive Controls (see Section 2.3).

2.1.2 Terminology

Despite the first approach of FTC techniques go back to 1970, they have only brought the attention of control designers in recent years, appealing the scientific community to initiate the research on this topic. Both groups have established a common terminology [5, 68]:

- **Faults:** This term specifies any source of malfunctioning in a machine, becoming the base element in FTC techniques. Multiple causes trigger faults in systems, being necessary to classify them accordingly (see Section 2.2.1).

- **Failures:** A system is under this status when they perform below the optimal conditions, finishing the service provided. If faults refer to the component level, failures affect at the system level. Preventing component or subsystems faults from becoming system failures is the FTC purpose.

- **Fault-Tolerance:** Systems have this property when their controller continues granting a proper service until its maintenance despite the fault emergence. Even when fault compromises system behaviour, they are still considered as Fault-Tolerance, providing that the safety protocols and production remains in the region of degraded performance (see Section 2.1.4).

- **Fault-Tolerant Control:** Fault diagnosis and controller redesign are the two conceptual steps for this property (see Section 2.2.2 and 2.2.3). During the diagnosis phase, systems detect, quantify and isolate faults online; while the redesign phase comprehends techniques to adjust the control algorithm to attenuate the fault effect.

2.1.3 Requirements and properties

Faults, as the machine malfunctioning source, degrade system performance, reducing its reliability and jeopardising operators life [5]. When control designers develop an FTC, they ensure these four notions to consider the system secure.

- **Safety:** Systems are under this property when operators handle them in the absence of danger. This security comes from software or hardware protections, such as code restrictions or emergency buttons, respectively. A system is fail-safe when, in response to a critical signal, triggers a controlled shut-down to protect the machine.

- **Reliability:** Systems are reliable when they maintain the optimal behaviour without suffering degradation during a specified period. FTCs minimise the fault effect on systems, extending the time spent between fault emergence and system performance degradation.

- **Availability:** Systems are available when the time spent between the user request and the machine response is minimum. Maintenance policies affect this property indirectly, as systems remain inoperative during these processes.

- **Dependability:** Joins together the previous properties: safety, reliability and availability, that is to say, a system is dependable when its fail-safe with high reliability and availability.

When a fault emerges on a system, their performance becomes degraded. Attending to the previous notions, FTCs should ensure the system dependability through one of the following situations (see Section 2.1.4):

- **Fail-operational:** In a first instance, faults degraded the system behaviour when they emerge, but after the controller redesign, the machine recovers the optimal performance.
- **Fail-graceful:** Although the controller ensures the dependability condition and the machine remains in operation, the system has a degraded performance below the optimal point.

2.1.4 Regions

The system performance is divided into four regions attending to the degradation suffered [5] (see Fig. 2.1). Initially, the system is in the *Region of Required Performance*, where the machine reflects the nominal behaviour. When a fault emerges, the performance becomes compromised, entering the *Region of Degraded Performance*, where the system is still able to operate, but below the nominal behaviour.

When the system penetrates this region, the FTC automatically initiates the recovery process. If the restoration is successful and the machine returns to the area of required performance, the system remains in a **fail-operational** status. However, if the controller was unable to recover, but the fault spread is contained, the system remains in a **fail-graceful** condition.

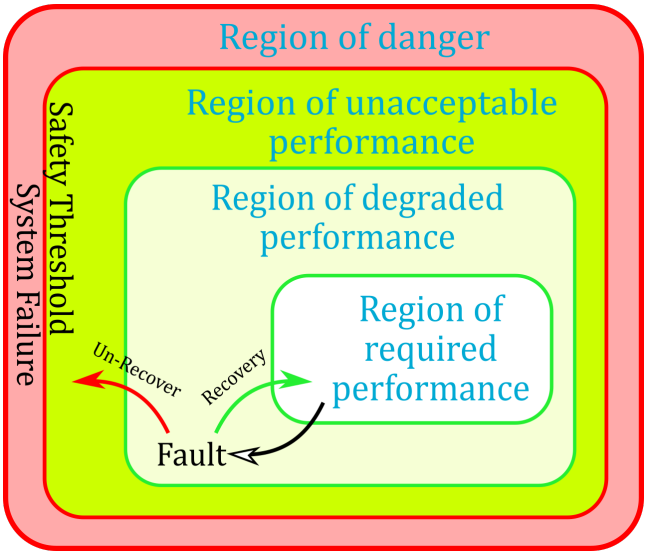


Figure 2.1: Regions of performance defined in a system under the effect of faults.

If the controller does not address the fault on time, their harming effect continues spreading to the *Region of Unacceptable Performance*. The recovery process has failed, compromising the system stability and reducing the production below the optimal behaviour. Besides, due to the harm suffered in the machine components, the controller does not ensure the reliability and safety levels. FTCs

mission is the avoidance of this region, preventing the system entrance into this under-optimal behaviour.

After trespassing the safety criterion threshold, the system enters the *Region of Danger*. Under this region, the fault has damaged the system components beyond the recovery point, initiating the failure status. Besides, the fault has neglected the FTC effect, containing the damage through the safety protocols, which ensure that the system is harmless and does not entail any threat for operators. When this critical situation is reached, the system requires human interaction to repair it and regain the nominal behaviour.

2.1.5 Classification

Control designers are well-aware of FTC techniques, as they have applied them traditionally in the design of industrial controllers. They tune the gains accordingly to the most disadvantageous position to ensure a proper function. Recently, in addition to this traditional control technique, another approach has become more appealing [5, 150, 100, 118, 72, 116, 56, 22] (see Fig. 2.2):

- **Passive Fault-Tolerant Control (PFTC):** Control designers detect the most dangerous position in the machine, tuning the controller accordingly to this situation. On this technique, the controller remains invariant when faults emerge, requiring less time to surpass it at the expense of limiting the number of faults susceptible to be recovered [150, 100].

- **Active Fault-Tolerant Control (AFTC):** On a first instance, the controller detects, quantify and isolate faults when they emerge (Fault Detection and Isolation phase). On a second instance, the controller retunes their algorithm to surpass the fault (Control Redesign phase). This principle is particularly efficient for machines composed of multiple subsystems [96, 116, 56].

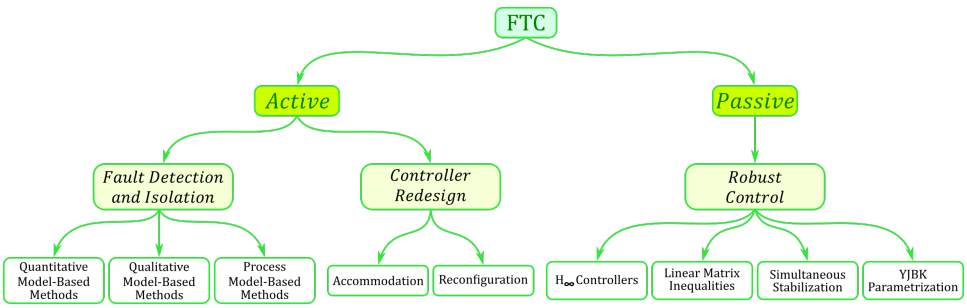


Figure 2.2: Classification of the Fault-Tolerant control techniques (expanded D.1).

The previous classification introduced the concepts of PFTCs or Robust Control, as a sort of controllers prepared to continue operating even when the fault

emerges without further modifications. Their design guarantees the system stability and performance against the most harmful and common fault, dividing them attending to their methodology into:

- H_∞ **Controller:** These controllers minimise the H-infinity-norm to optimise the worst performance case through a robust design based on stable dynamical compensators, such as the Kalman filters [19, 17]. Introducing an FDI mechanism allows control designers to upgrade their recovery characteristics from PFTC to AFTC [47, 21, 73].

- **Linear Matrix Inequalities (LMIs):** The tuned controller sought for the robustness against actuator and sensor faults. They solve convex problems with precise matrix constraints through fuzzy logic [18, 16, 150, 104, 103].

- **Simultaneous Stabilisation:** When the system is under the effect of multiple faults, this solution maintains stability through a unique controller tuned accordingly to this situation [121, 120].

- **Youla-Jabr-Bongiorno-Kucera (YJBK) parameterisation:** This methodology guarantees the system stability through controllers tuned representing feedback control algorithm architectures [84, 83, 82].

In contraposition to PFTC, AFTC deals with a broader number of faults due to their two phases process. As they are the most appealing control algorithm for contemporary industries, this thesis has focused on them. Due to this fact, the classification and study continues in Section 2.2 rather than on the following paragraphs.

2.2 Active Fault-Tolerant Control

Natural systems are continually evolving, adapting the fauna and flora attributes to the volatile environment. Similarly, industrial machines are far from conserving their behaviour, suffering modifications from multiple sources, such as fluctuations in the productive cycle, mechanical fatigue degradation in the components and maintenance policies in the factory. Controllers maintain the system stable despite these inconstancies; nonetheless, the status becomes compromised when the adverse impact persists and spreads their harmful effect.

Active Fault-Tolerant Control deals with this adverse behaviour, recovering the system stability. It surpasses the fault, retrieves the optimal response and avoids spreading the effect between components in a two-phase process. On the initial phase, the Fault Detection and Isolation mechanism detect, quantify and isolate the fault; while on the last phase, the Control Redesign algorithm recovers the production.

2.2.1 Classification

Each phase in AFTC has its own set of techniques and algorithms (see Fig. 2.2). FDI phase has several mathematical methods that approach the detection, the gradation and the isolation problematic, being the most relevant (see Fig. 2.3):

- **Quantitative Model-Based:** The FDI mechanism bases their algorithms in empirical mathematical equations, whose variables are obtained dynamically from the system responses. Through these measures, estimative methods, such as Kalman filters, calculate the unknown parameters [67, 48].
- **Qualitative Model-Based:** The mathematical equations determine the fault comparing the current system signals with the theoretical responses it should have attended to its physical properties. The technique has two main approaches, causal models, where the patterns focus on the degraded component performance and the fault effect; and abstraction hierarchies, where diagrams, fault trees and qualitative physics reveal the fault interaction between components [33, 112].
- **Process History-Based:** Instead of identifying the fault through mathematical equations, the current system responses compared their value against previously-stored samples. These measures come from empirical signals recorded during laboratory experiments. The *a priori* identified dynamics isolate the fault through quantitative and qualitative approaches, the first one basing the estimation in Artificial Intelligence algorithms and the second one in statistical methods [153, 154].

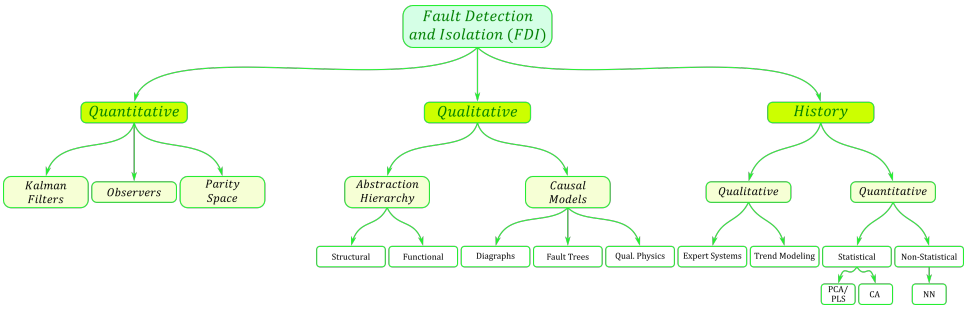


Figure 2.3: Fault Detection and Isolation (FDI) methodologies (expanded D.2).

On the CR phase behalf, there are two main approaches [135, 5, 68], fault accommodation and reconfiguration (see Fig. 2.4):

In **fault accommodation**, the controller input and output signals remain unaltered, as the redesign algorithm only modifies the controller gains. For instance, considering a PID architecture, this approach only modifies the proportional, derivative and integrative gains. Two primary methodologies have approached this technique:

- **Adaptive Control:** When the plant parameters suffer an alteration, the control law adapts its gains accordingly to recover the optimal performance. These controllers reduce the fault effect without pre-information about it, minimising the difference between the current and the desired behaviour. They are optimal for linear plants as the adaptation process modifies the controller slowly; however, their performance becomes degraded for abrupt faults [39, 70].
- **Switched Control:** These controllers state multiple scenarios (conventional or fault), designing a set of controllers gains for each one. The adaptation law switches automatically to the controller tracking the reference more efficiently [91, 143, 149, 64]. Despite each control algorithm has its gains, the input and output signals remain identical.

In **fault reconfiguration**, in addition to the controller gains, the input and output signals connected to sensors and actuators, respectively, also vary, modifying the whole control algorithm in response to faults. The following methodologies have approached this technique:

- **Physical Redundancy:** Instead of designing complex logic decision processes, they obtain the fault-tolerant behaviour through hardware redundancy, that is to say, installing identical sensors and actuators in parallel. The control algorithm switches from the faulty component to their identical replica automatically when the fault emerges [125, 40].
- **Projection Based Methods:** Control designers establish a limited number of fault scenarios producing a control architecture for each one. They create a bank of controllers and a bank of observers, selecting between the closest predefined scenario when a fault emerges. In contraposition to Switched Control, the controller shift their gains in conjunction with the input and output signals from sensors and actuators [108, 80, 49].
- **Controller Redesign:** When a fault emerges on the system, the controller detects the source and modifies the controller accordingly to continue achieving its objective through several approaches, such as pseudo-inverse methods, model following and control optimisation [55, 29, 69, 119].
- **Fault Hiding Methods:** Instead of redesigning the control algorithm gains, a reconfiguration block placed between the plant and the controller hides the fault. This mathematical structure masks the faulty signal, allowing the controller to continue operating as if no drawbacks exist. Their most successful strategy employs virtual actuators and sensors [89, 85, 93, 92].
- **Learning Control:** The controllers, instead of behaving passively, anticipate the fault and adapt their algorithm regularly to prevent its emergence. This continuous process requires Artificial Intelligence techniques, such as Neural-Nets [115, 105], Fuzzy Logic [39, 64, 122] and Genetic Algorithms [153, 154, 1], to maintain the system controlled. Their ability to detect and identify the fault determines the successfulness of these methods.

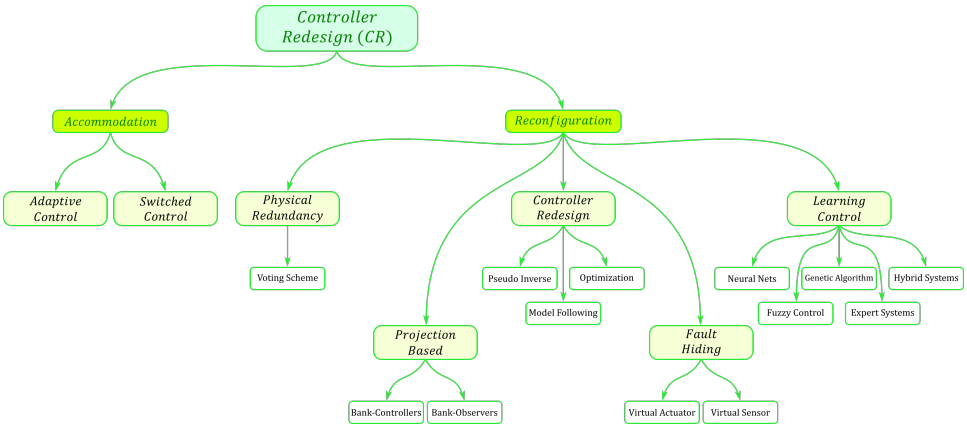


Figure 2.4: Controller Redesign (CR) methodologies (expanded D.3).

2.2.2 Fault Detection and Isolation

When something hurts an animal, it suffers an injury that slows down their behaviour. For instance, a wounded deer would travel slower than a healthy specimen; nevertheless, if the bruise is superficial, it would still be able to walk. Faults alter the nominal behaviour of systems similarly, as they prevent them from operating properly. In the cases studied in this thesis, faults would behave similarly as the wounded deer case, where they reduce the production without neglecting the machine performance completely.

If a fault emerges on the system the machine performance is affected negatively, altering the actuator's behaviour, considered as the input (U), and the sensor's response, regarded as the output (Y) [68, 23, 152]. The diagnostic algorithm, or Fault Detection and Isolation technique, discovers this abnormal conduct, gathers the system status and provides a report to the redesign mechanism (see Fig. 2.5).

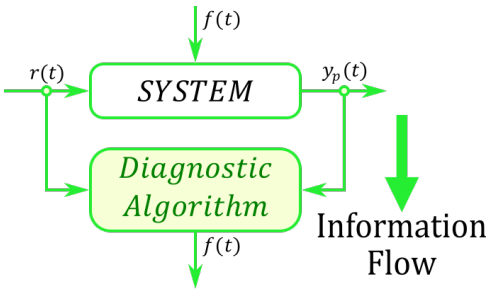


Figure 2.5: Graphical description representing the flow information diagram when a fault emerges on the system.

The diagnostic algorithm procedure has three steps to quantify the magnitude and locate the faulty system:

- **Fault detection:** Discerns when a fault has emerged on the system. The time spent on this step is crucial, as the system would remain in a faulty status assuming a non-existent drawback.
- **Fault quantification:** Estimates the fault magnitude, discerning its spread pattern and how it would affect other components and systems.
- **Fault isolation:** Discriminates the components affected by the fault, discovering its location and isolating them from the system.

Multiple approaches have analysed and solved this procedure through a shared universal principle, the consistency-based diagnosis [5]. System behaviour theory addresses this abstract concept establishing the bases for the fault discovery procedure (see Fig. 2.6). Assuming that every input signal $U(s)$ has its corresponding output $Y_p(s)$ value, the set of all the possible output values for every potential input signal defines a region entitled as the system behaviour (β).

Control designers have modelled the system behaviour (β) obtaining the set of equations that matches the studied region of input/output $(U(s), Y_p(s))$ signals. During the system execution, the diagnosis algorithm compares the responses obtained from the machine against the modelled system behaviour (β). When the performance is under nominal conditions, both values coincide, nonetheless, if a fault has emerged the response diverges:

$$(U(s), Y(s)) \notin \beta \tag{2.1}$$

This deviation determines that a fault (f) has emerged on the system, defining a new faulty behaviour (β_f) described by a new pair of signals $(U_f(s), Y_{p_f}(s))$. In these cases, f is a fault candidate, obtaining its characteristics during the analysis carried out in the FDI phase.

As an example, Fig. 2.6 brings the nominal behaviour and three fault cases (f_0, f_1, f_2) defined by their behaviour $(\beta_{f_0}, \beta_{f_1}, \beta_{f_2})$. Assuming that the system performance belongs to the points A, C or D, the discovery mechanism easily discern the faults f_0, f_1 or f_2, respectively. However, point B presents an overlap between two faulty situations (f_0 or f_1). Point E even exacerbates this problem, as the discovery mechanism has to discern between faulty behaviour (β_{f_0}) or nominal performance (β).

Compare the measured signals does not bring enough information to discern this ambiguity, questioning the methodology selected to discover faults. System diagnosability or fault detectability established a novel discovery mechanism through the *Consistency-Based Diagnosis* (CBD). This principle states that systems are subjected to faults f and describes faulty behaviours β_f when another input $U(s)$ in the same range also defines a faulty output $Y_{p_f}(s)$.

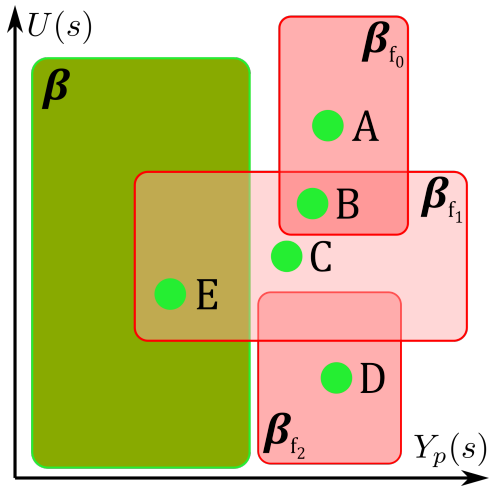


Figure 2.6: Graphical representation of the system behaviour.

$$(U(s), Y_{p_f}(s)) \in \beta_f \tag{2.2}$$

Systems determine the fault category considering these assumptions during the conduct of further tests:

- The fault is detectable without *a priori* information about its behaviour, basing the fault detection algorithm on nominal plant models.
- The fault is recognisable through a model that brings information about its spread in the system, allowing fault isolation and identification.
- Each fault belongs to a set; otherwise, it is excluded as a fault candidate to preserve the Consistency-Based Diagnosis.
- The fault has to be measurable. Indistinguishable cases should recognise it through other mathematical approaches.

Active Fault-Tolerant Control benefits from this system diagnosability process and the Consistency-Based Diagnosis, as the FDI phase discovers the fault source allowing a more personalised Control Redesign phase.

2.2.3 Control Redesign

The symmetries between natural and industrial systems also have their spot in the Control Redesign phase. In the example stated previously (see Section 2.2.2), the deer learns to survive despite its injury. As an example, this thesis compares the Control Redesign phase with animal behaviour after being shot on its hindquarters. The bullet neglects its movement partially; however, it is still able to continue

escaping. This task is accomplished when the deer manages to understand its wounds, discovering how to mitigate their effects, encountering an alternative to escape, for instance, employing their further limbs.

Similarly, control algorithms process the information gathered during the FDI phase to update the controller automatically and surpass the fault. In the fault detectability methodology, control design sought the position where plant β_P and controller β_C behaviour converges in the region of optimal performance β_{SPEC}. Faultless systems share this principle, as they intersect into a set of optimal values that satisfies the control laws, named as the nominal system behaviour (see Fig. 2.7).

$$(\beta_P \cap \beta_C) \subset \beta_{SPEC} \tag{2.3}$$

When a fault arises, the plant behaviour becomes degraded deviating its value from this optimal performance. Although the faulty plant β_{P_f} alters the closed-loop system, the region of optimal performance β_{SPEC} remains identical. The system would recover this performance after redesigning the controller algorithm β_{C_f} to converge with the faulty plant β_{P_f} dynamics inside the region of optimal performance β_{SPEC}:

$$\left(\beta_{P_f} \cap \beta_{C_f}\right) \subset \beta_{SPEC} \tag{2.4}$$

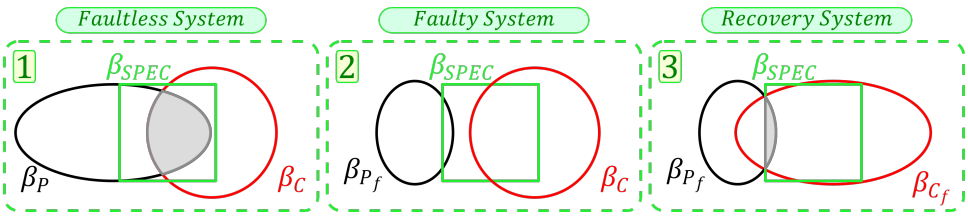


Figure 2.7: Stages followed from fault appearance to fault recovery.

Control Redesign algorithms discriminate two approaches attending to the relation between controlled β_C and faulty plant β_{P_f} region inside the optimal performance β_{SPEC}:

- **Fault Accommodation:** When the fault emerges on the system, controller β_C and faulty plant β_{P_f} behaviour still share some common region inside the optimal performance β_{SPEC}. There are input U and output Y signals that could achieve the desired behaviour, so CR phase tune the controller gains accordingly to the fault behaviour without modifying the control algorithm [46, 39, 70, 65, 142, 143, 149] (see Fig. 2.8).
- **Fault Reconfiguration:** When the fault emerges on the system, the faulty plant behaviour β_{P_f} matches the optimal performance region β_{SPEC} but not

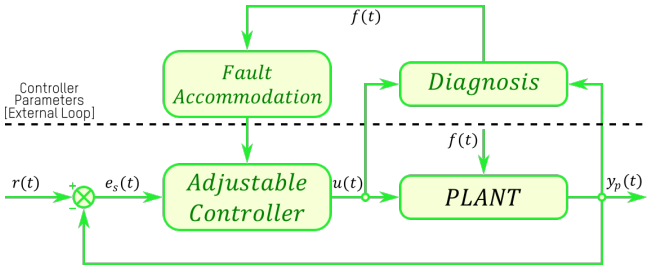


Figure 2.8: Closed-loop schematic for fault accommodation.

the controller β_C. During the CR phase, the technique not only tune additional controller gains, it also modifies the control algorithm input U and output signals Y to match the faulty plant β_{P_f} [40, 49, 119, 105, 89] (see Fig. 2.9).

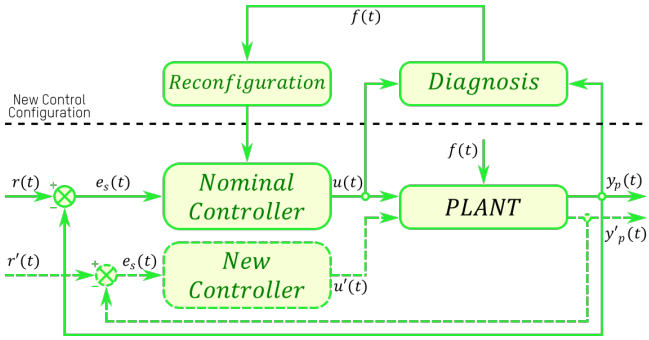


Figure 2.9: Closed-loop schematic for control reconfiguration.

Control Redesign phase is viable regarding that the faulty plant behaviour region β_{P_f} at least remains partially inside the optimal performance β_{SPEC}. Critical faults alter the plant behaviour beyond this point, rendering the controller useless, leading the system to collapse. This thesis has stated a Hydraulic-Press subjected to non-critical faults as the case study, degrading the system performance without entering the failure status.

Between the CR phase approaches, fault accommodation offers a suitable solution, as these non-critical faults modify the plant behaviour without altering the control algorithm structure completely [99, 98]. Several techniques are suitable for fault accommodation (see Section 2.2.1), implementing the ones based on Adaptive Control (see Section 2.3).

2.2.4 Real-Time problematic

AFTC redesigns the controller online, bringing a powerful and adaptable mechanism to deal with faults. Despite these advantages, they require to retune the

control algorithm without shutting down the machine and maintaining a constant production. There are several mathematical algorithms capable of auto-tuning the controllers' gains in offline status; nonetheless, tuning them in online status have additional requirements [5]:

- The mathematical algorithms have to obtain the new control algorithm automatically without human interaction.
- The mathematical algorithms must guarantee the safety criteria even if they sacrifice the optimal machine performance.
- The mathematical algorithms tune the new gains without pausing the production, requiring Real-Time capabilities to avoid instability.

FTC methodology considers a control algorithm tuned in Real-Time when the redesigned controller satisfies the control objectives in a known, constant and fixed time. This requirement ensures a constant time between the fault emergence and the system recovery. The following steps reflect this process:

1. Until the fault emerges, the nominal controller regulates the system, accomplishing with the original control objective.
2. When the fault emerges, the FTC initiates the recovery process. During the FDI phase, the older controller remains in operation, ensuring the system always has a controller in action. Afterwards, the control algorithm switches from the previous distribution to the new one during the CR phase.
3. After recovery from the fault, the latest controller algorithm commands the system, satisfying once again with control objectives.

This process is critical for AFTCs, as there are phases where the nominal controller manages a faulty plant. This situation induce harmful conditions in the machine, as the controller would seek references signals no longer achievable. The longer this situation prevails, the harder it is for the system to recover its previous stability. This thesis has considered this problematic, addressing it through a methodology that reduces the time spent in the FDI phase, while adapting the controller gradually in the CR phase.

2.3 Adaptive Control

2.3.1 Introduction

Natural fauna and flora are continually adapting their features to the surrounding environment. There are several mechanisms involved in this process; for instance, evolution ensures species survivability modifying their genes across multiple generations. Instead of this long term mechanisms, this thesis sought to alter the current patterns to induce a change in the system behaviour.

Adaptive Control adjusts the controller gains when plants present alterations in their dynamics. This technique accommodates the closed-loop response without *a priori* information about the current system behaviour. Several sectors benefit from this mechanism; for instance, drone controllers adapt the blade angle automatically to the prevailing wind velocity or video game controllers accustom their buttons to disabled persons.

The standard AC architecture presents an **adjustable controller** with tunable gains and an **adaptation mechanism** with the algorithms to alter the variables mentioned above (see Fig. 2.10). This mechanism, called the *Certainty Equivalence Principle* (CEP) [46, 79, 20, 76, 77], join both structures, ensuring that despite the plant behaviour fluctuations, the controller establishes the signal and tracks the reference.

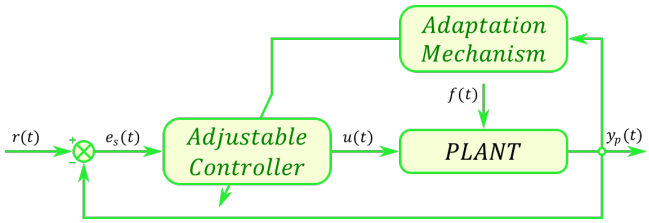


Figure 2.10: Standard architecture presented on Adaptive Controls.

On this methodology, control algorithms alter their dynamics accordingly to the plant behaviour fluctuations. The controller gains obtain this mutable peculiarity combining static information collected during the commissioning stage with dynamical signals received from the plant responses. This mixture brings adaptable controllers capable of modifying their performance independently of uncertainties in the plant.

2.3.2 Adaptation Law

The adaptation mechanism is far from being random, as a complex set of equations and mathematical rules named as **adaptation laws** adjusts the controller. The literature brings robust theories approaching this problematic [46]:

- **Sensitivity Methods:** A cost function estimates the controller gains through the *sensitivity function* partial derivative. This equation refers to the square error between the desired and measured plant responses. Although they solve the adaptation problem, they are prone to reach local minimums. *Massachusetts Institute of Technology* (MIT) rules commonly approach them [101, 42, 123].

- **Positivity Design:** A differential equation, named as the Lyapunov function V, solves the stability problem when its first derivative is negative $\dot{V} < 0$. This transfer function represents the system adaptability potential. Instead

of reaching local minimums, the equation sought to reduce the error between plant output and optimal model behaviour [128, 24, 109].

- **Minimum Square Error:** A sensitive function estimates the error between physical measures obtained from the plant and virtual signals collected from the model. This function should describe a convex distribution whose gradient is decreasing [3, 79, 27, 14, 6, 38].

2.3.3 Classification

Attending to the adaptation mechanism, the controller updates its gains directly, indirectly or hybridising both techniques (see Fig. 2.11). The estimation method approximates the plant and controller parameters to obtain an invariant adaptation law regardless of the system dynamics fluctuations. On every case, the control algorithm remains intact, varying the adaptation law parameters exclusively.

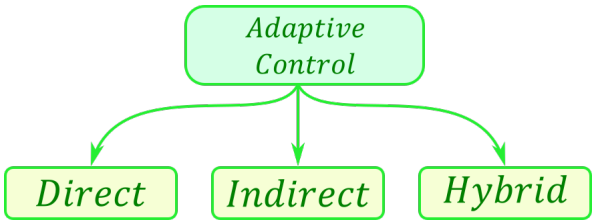


Figure 2.11: AC methodologies classified attending to the adaptation mechanism.

2.3.3.1 Direct Adaptive Control

In the **Direct Adaptive Control** (DAC) or implicit [94, 117, 129], the adaptive controller parameters estimate their value directly before determining the plant dynamics. Although the mechanism reaches the adaptive parameters faster than other techniques, they have lower precision (see Fig. 2.12).

This technique estimates the plant dynamics through the control parameters vector θ_c^*, provided that, the control law $C(\theta_c^*)$ accomplishes the requirements to obtain the plant model $P_m(\theta_c^*)$. This virtual model represents the physical properties defining the original plant $P(\theta^*)$, having similar input and output signals.

Instead of designing the online parameter estimator θ_c^* through the original plant dynamics $P(\theta^*)$, the algorithm determines its value directly basing their analysis in the input and output signals registered in the plant model $P_m(\theta_c^*)$. This estimation updates the control parameters vector θ_c^* without further calculus.

2.3.3.2 Indirect Adaptive Control

In the **Indirect Adaptive Control** (IAC) or explicit [136, 61, 146], an adjustable predictor estimates plant dynamics and controller gains. Although the mechanism determines these parameters slower than other techniques, they have higher precision (see Fig. 2.13).

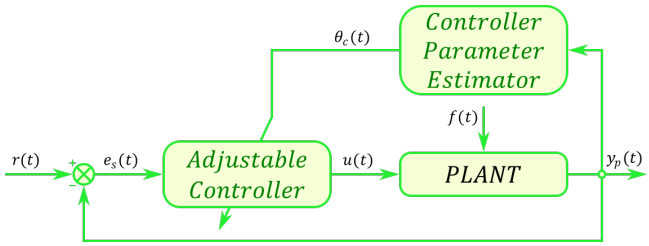


Figure 2.12: Generic schematic followed for Direct Adaptive Control.

This technique estimates the parameters online θ at a fixed sample rate t through a three-stage process:

1. Initially, the estimator characterises a model reflecting the plant input $u(t)$ and output $y_p(t)$ relation.
2. Afterwards, the estimation process determines the control parameter vector θ_c solving an algebraic equation $\theta_c(t) = F\left(\theta(t)\right)$, following an analogue process as the one stated for the DAC (see Section 2.3.3.1).
3. Finally, the control law $C(\theta_c)$ tune their gains accordingly to this function, matching the current algebraic equation $\theta_c(t) = F\left(\theta(t)\right)$ with their counterpart designed under nominal conditions $\theta^* = F(\theta^*)$. Obtaining the control parameter vector θ_c^* simulating an already-known plant $P(\theta^*)$.

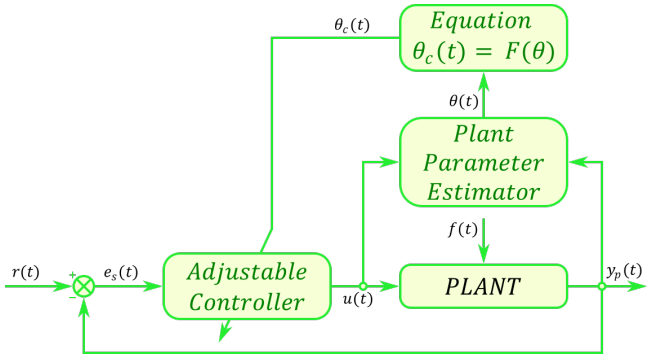


Figure 2.13: Generic schematic followed for Indirect Adaptive Control.

The online plant parameter estimator (when it is in discrete time) has two structures (see Fig. 2.14):

1. The adjustable predictor prognosticates the model output signals for each sample rate. The mechanism compares the current control command and the plant signal against the equations gathered during the commissioning stage with the original closed-loop system behaviour.

2. The adaptation mechanism determines the control parameters vectors integrating the estimation error between model and real plant responses through an adaptation law.

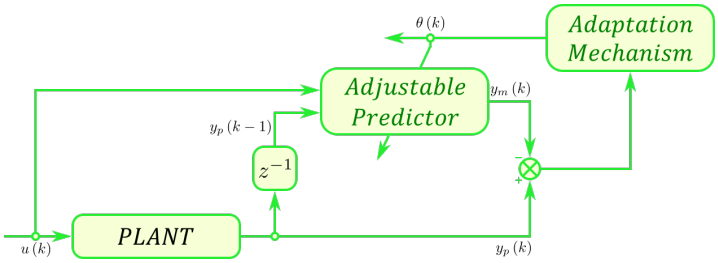


Figure 2.14: Basic schematic for plant parameter estimator.

2.3.3.3 Hybrid Adaptive Control

In the Hybrid Adaptive Control [88, 41, 43], the adaptation control algorithm inherits the adaptive system and the controller vector parametrisation from indirect and direct methods, respectively (see Fig. 2.15). The estimator parameters follow a similar distribution as the indirect AC, while their output behaves similarly as the plant model $P_c(\theta_c^*)$ in direct AC.

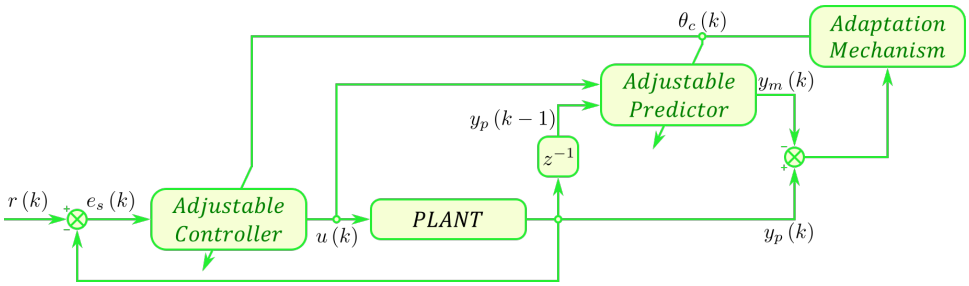


Figure 2.15: Generic schematic followed for Hybrid Adaptive Control.

2.3.4 Topology

Control designers have classified Adaptive Controls into Direct, Indirect and Hybrid methods, attending to their adaptation mechanism. Besides this distinction, they also state another separation regarding their control topology. This differentiation studies the mathematical equations solving the control algorithm and the position of the adaptive gains inside the controller, that is to say, it discriminates the AC architecture.

These architectures develop a DAC, IAC or HAC behaviour attending to the adaptation mechanism operation mode. They also differentiate between *feedback*

or *feedforward* structures [35, 12, 107]. The first case operates best when the plant dynamics evolve slowly, while the second case cancels easier sudden perturbations. Both cases maintain the system stable while the adaptation mechanism is still accommodating the controller. Besides, some approaches mixed both structures to maximise the benefits.

ACs distribution determines the adaptation rate, affecting directly to the interval employed on the controller adjustment process. Although the architecture selected defines the adaptation mechanism, control designers prepare them following an identical pattern [53]:

1. Select the measured signal that describes better the system behaviour.
2. Compare the system current behaviour against their ideal performance.
3. Modify the controller parameters adapting the process to the desired behaviour.

There are several architectures involved in the adaptation process, studying the most relevant ones on this Section (see Fig. 2.16):

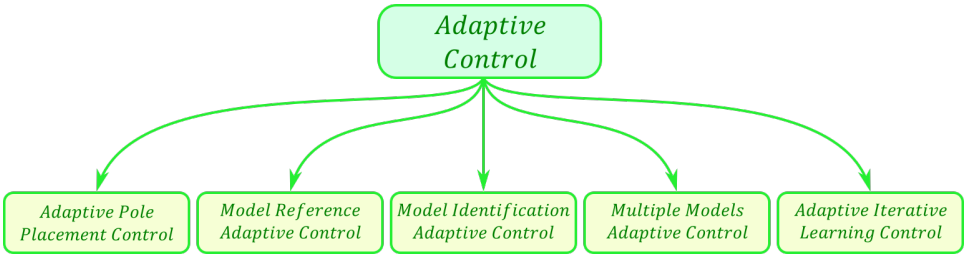


Figure 2.16: AC methodologies classification attending to its topology (expanded D.4).

2.3.4.1 Adaptive Pole Placement Control

The *Adaptive Pole Placement Control* (APPC) has a controller structure $C(\theta_c^*)$ and a parameter vector θ_c^* selected to match the closed-loop system poles, obtaining a feedback loop with identical dynamics as the desired response. Control designers obtain $C(\theta_c^*)$ and θ_c^* through the plant parameter vector θ^*. The CEP substitutes the vector θ_c^* for its estimation θ_c, reaching the standard APPC schematic (see Fig. 2.17).

Several authors have studied this methodology on both trends, their direct [14] or indirect [79, 20, 27, 106, 78] approach. They excel in controlling non-minimum phase systems when there are poles or zeros in the negative semi-plane, except for systems that require a pole-zero cancellation. Control designers formulate these algorithms when the system is controllable and they know the plant grade.

They also have additional advantages, such as their flexible design, which only depends on the control algorithm distribution and the parameter estimator. Commonly, authors based estimation algorithms on the least square method

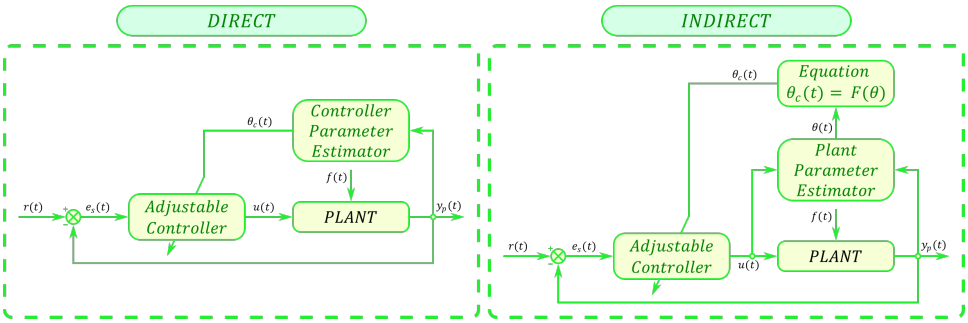


Figure 2.17: Direct (left) and Indirect (right) version of the Adaptive Pole Placement Control.

[3, 79, 27, 14, 6, 38]. There are other alternatives, such as the APPC running through a Neural-Net trained with batch learning methodologies, providing the controller with a model of the system [32]. Regarding the control law, control designers have currently approached it by a two-step methodology. Initially, the control designer selects a suitable dynamic, and, afterwards, decide the optimal architecture. Finally, it states and solves an equation connecting the control parameters with the desired poles and zeros, obtaining the desired control-loop dynamic.

2.3.4.2 Model Reference Adaptive Control

Model Reference Control (MRC) sought a control law that alters the controller structure and dynamics until plant and model outputs are identical, where the model algorithm is defined by a set of mathematical equations describing the desired input/output closed-loop dynamics. The feedback control loop cancels the plant dynamics, substituting its transfer function for the model reference equations. This action ensures that for any reference, the tracking error $e(t) = y_p(t) - y_m(t)$ converge to zero. This cancellation restricts the controllers' practice to non-minimum phase systems; otherwise, the methodology would destabilised the signal.

When the factor θ^* is unknown, the parameter θ_c^* is also unobtainable. There are alternatives to estimate its value directly or indirectly, substituting this missing information with the predicted parameter θ_c obtained after applying the CEP. The schematic, known as the Model Reference Adaptive Control (see Fig. 2.18), responds actively to alterations on the plant parameters, estimating the latest dynamics and adapting the controller accordingly to continue tracking the reference.

These methodology design controllers attending to the following considerations [3, 128, 26]:

- Plant transfer function should be minimum phase.
- Plant relative grade is known.

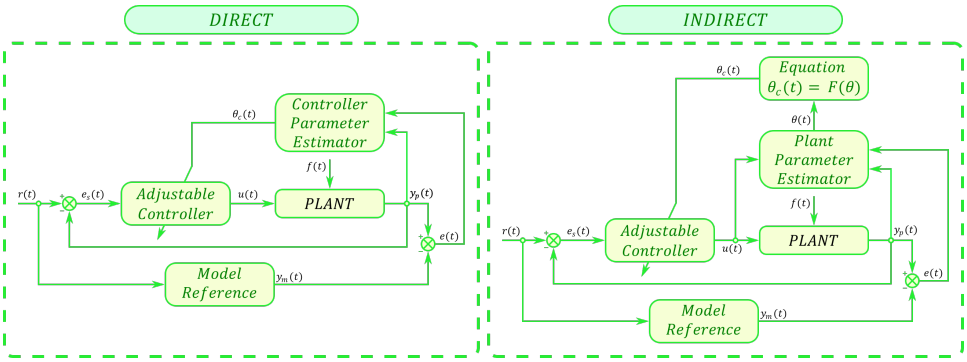


Figure 2.18: Direct (left) and Indirect (right) version of the Model Reference Adaptive Control.

- Plant static gain sign is known.

Besides, control designers consider additional requirements when they design the reference model. The outstanding consideration is that they encounter a transfer function stable and non-minimum phase, as it would define the desired output. There are additional conditions, as the feasibility restrictions, that is to say, the model has to ensure that the controller is realisable [128].

Regarding the controller parameter estimator methodology, there are approaches based on the MIT Rule [101, 42], on the Lyapunov stability [128, 24, 109] or on the minimum root strategies [6, 38]. There are additional options [3], such as models based on multipliers, that minimise a cost function through steepest-descent methods. Similarly to the previous cases, this methodology ensures system stability properties.

There are several considerations when control designers apply this methodology to industrial systems. MRACs are unable to deal with saturation, requiring to improve their algorithms with Anti-Windup techniques. Saturation is a widely extended problem in conventional control architectures, such as PIDs. On these cases, the designers disable the integration action when the system reaches its mechanical limits.

MRACs based their adaptation mechanism on dynamical models, preventing the direct translation of these techniques [128]. Some authors have approached these problematic, modifying the adaptation law [109] or limiting the reference signal [7, 60]. This thesis has initiated a novel research lane, where the reference model response varies accordingly to the saturation effect.

Another MRAC problematic resides in the requirement to understand the plant dynamics to elaborate the control algorithm. Control designers lessen this situation by applying Minimal Control Synthesis (MCS), [35, 12, 107] an extension of the MRAC methodology based on a feedback and feedforward control with adaptive gains, reducing the restrictions, assuming null initial conditions for the control

gains and maintaining the robustness against uncertainties, variations and perturbations in the plant parameters [46, 53].

2.3.4.3 Model Identification Adaptive Control

Their strategy identifies a novel plant model or redesigns the current dynamics of an older version without interrupting the performance (see Fig. 2.19). The adaptation mechanism transforms the control algorithm accordingly to the latest reference model without destabilising the system. Their architecture comprehends two modules, one for the plant identification and the other for the controller update.

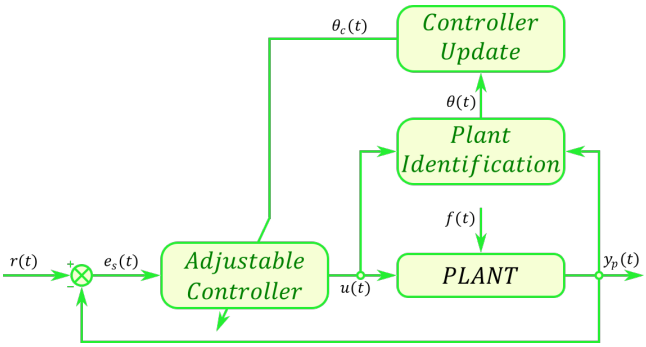


Figure 2.19: General schematic for the Model Identification Adaptive Control.

Their main difference with other adaptive methodologies resides on its capacity to update the model topology in addition to their parameters. This feature increases its relevance as an FTC technique, covering a more extensive range of faults. The identification mechanism distinguishes systems through rules and methods, adjusting the model accordingly to the stockpiled information. The following points factorise this mechanism [37]:

- **Structure theory:** Analyses the connection between *external behaviour* and *internal parameters* against the relevant properties, identifying a reliable plant model.
- **Real value parameter estimation:** Studies the plant fluctuating parameters, estimating their current status through parametric procedures.
- **Model selection:** Obtains the real parameters defining model subclasses, that is to say, their internal physical equations.

There are multiple identification strategies, such as Genetic Algorithms [58] or Recursive Least squares (RLS) estimators [102, 111]. Control designers encounter more commonly the latest strategy, subjecting the identification process to the plant dynamics. The literature also has studied several algorithms dedicated to understanding the plant behaviour, for instance, the Preysach operator [124].

2.3.4.4 Multiple Models Adaptive Control

Previous AC strategies adapt the controller attending to a unique plant model, updating their dynamics accordingly to the latest behaviour. Nonetheless, when the system operates in unstable environments, control designers hardly encounter a unique model embracing all the alternatives. This context prompts the emergence of the Multiple Models Control (MMC) strategies, where several environments define plant status. The controller switches between these scenarios until it encounters the most similar case to the current plant dynamics.

Traditionally, these techniques design the controllers without upgrading the plant parameters [95, 71], tuning imprecise controllers, especially against non-modelled faults or perturbations (see Fig. 2.20). Afterwards, control designers have improved these techniques, including ACs on their models. This approach called Multiple Models Adaptive Control (MMAC), also adapts the model dynamics, being able to control a more extensive number of scenarios [13, 145, 8]. Despite its benefits, this approach consumes more resources, slowing down the adaptation mechanism. Besides, they are only efficient when the behaviour suffers gradual modifications [76, 77].

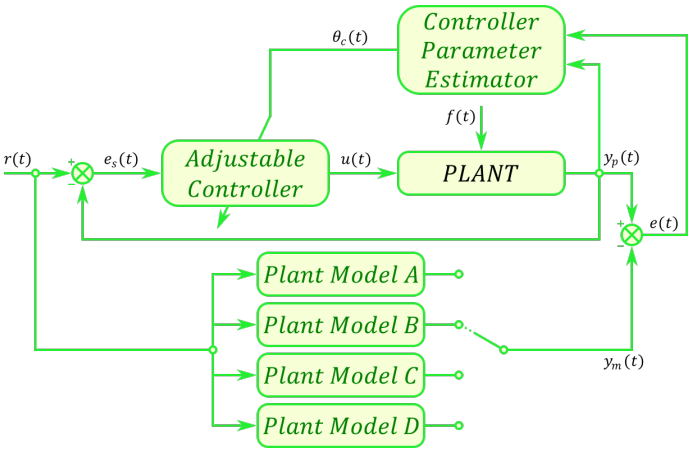


Figure 2.20: Control strategy for Multiple Models Adaptive Control combined with MRACs.

This strategy only modifies the plant characterisation, that is to say, it swaps between previously-stored dynamics that could potentially match the plant. Due to this property, this strategy is compatible with other AC techniques, such as MRACs [76], extending their usability to the fault detection and isolation mechanism [96, 95]. Control designers tune multiple algorithms, toggling between them until encountering the controller that surpass the fault more efficiently.

2.3.4.5 Adaptive Iterative Learning Control

Iterative Learning Control (ILC) algorithms are an adaptive strategy prepared to improve the transitory response in systems performing an iterative cycle across a fixed time interval. Conventional ILC algorithms recover information about the error between the controller output and the reference signal, adapting the coefficients to minimise this difference through a learning mechanism. Its basic algorithm has the following equation (see Fig. 2.21) [74, 9]:

$$u_{j+1}(k) = Q(z)\left[u_j(k) + l(k)e_j(k+1)\right] \tag{2.5}$$

where:

- **Q-Filter $Q(z)$:** Improves the controller robustness and its behaviour.
- **Learning Function $l(k)$:** Determines the tracking error measurements, updating the controller accordingly on the subsequent iterations. The following strategies exist attending to their structure [9]:
  - *PD Type:* Based on the proportional and derivative error. They do not require the knowledge of a precise model, just tune them.
  - *Plant Inversion:* They employ inverse plant models as learning functions. This methodology is unable to accommodate non-minimum phase systems.
  - H_∞ *Methods:* They search for the learning function that reaches fastest the convergence for the current $Q - filter$.
  - *Quadratically Optimal Design:* The learning functions and the Q-filter are designed to minimise the quadratic cost criteria on the next iteration.

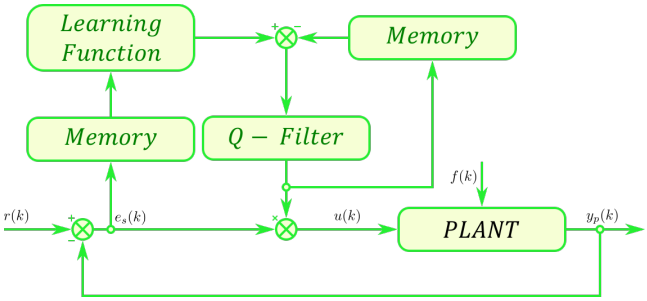


Figure 2.21: General schematic for the Iterative Learning Control.

ITCs are a feedforward strategy, but control designers commonly upgrade their algorithm with an additional feedback loop to improve its behaviour [126, 15, 141]. This mechanism compensates non-repetitive perturbations, discerning between the following architectures (see Fig. 2.22):

- **Series:** The feedback loop modifies system reference.
- **Parallel:** The feedback loop varies directly the control signal.

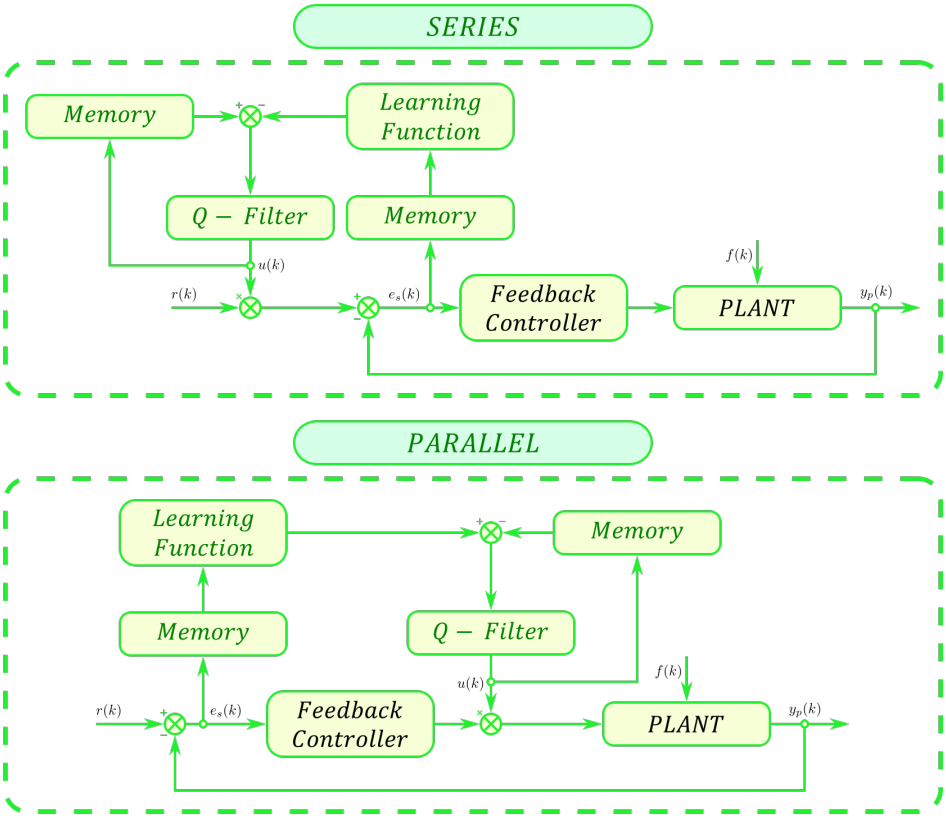


Figure 2.22: ILC architectures for the feedback loop: series (left) and parallel (right).

Initially, the ILC algorithms maintain the stability through *contraction mapping* methodologies, requiring a deep understanding of the system dynamics. These techniques present several restrictions that difficult their implementation on non-linear systems. Current algorithms include adaptive techniques to implement fuzzy logic systems and Neural-Network interfaces to deal with the non-linear uncertainties. This improved version called *Adaptive Iterative Learning Control* (AILC) solves system uncertainties through the whole system structure and the repetitive operation pattern [141].

The Lyapunov rule controllers also increase AILC performance [126, 141, 138], adjusting the input reference signal indirectly through the control law parameters. Besides, there are additional estimation methods, such as Kalman filters [86].

The literature discriminates between AILC methods attending to the mechanism employed to adjust the control law parameters (see Fig. 2.23). This separation divides the methodology regarding their axis into iteration, time or their

combination [126, 127, 15]. There are additional distributions for non-linear systems with delays [138] or restrictive input systems [148].

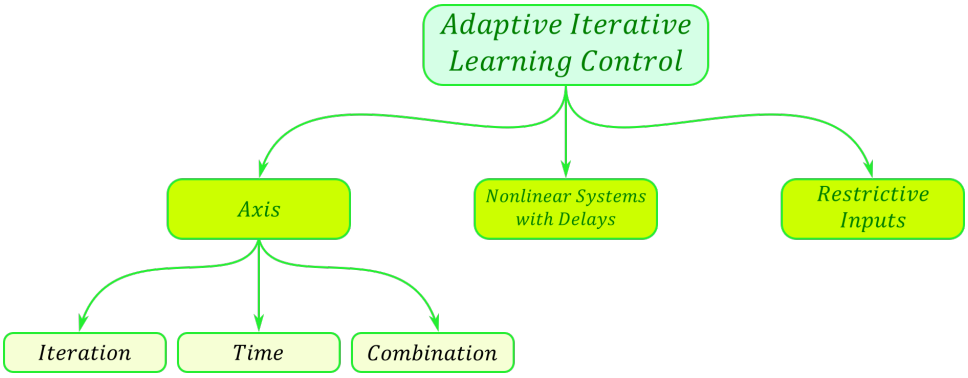
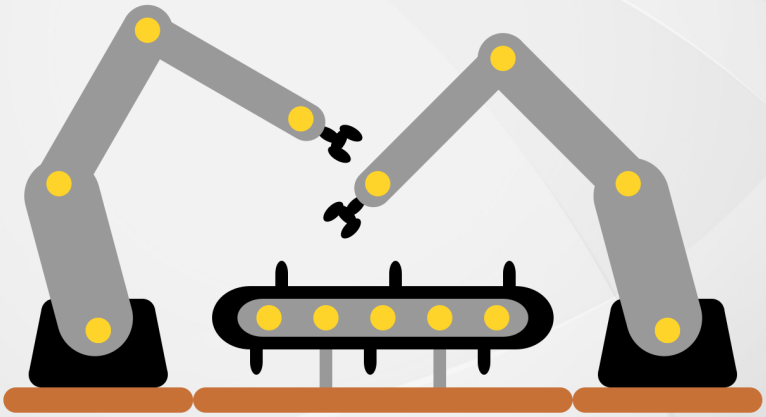


Figure 2.23: ILC division attending to their iteration with the axis and their linearity.

3 Methodology and Research



Industry behaves as a living entity, continuously evolving their manufacturing processes reducing production times without decreasing product quality. This search has lead to improve the manufacturing cycles, introducing novel techniques to maintain the machine working despite the appearance of faults [57], [63]. Due to this tendency, industrial systems have increased their productivity avoiding downtime periods.

Predictive maintenance brought the first step into this direction [11]. This technique estimates the machine break down, scheduling a maintenance before this drawback occurs to avoid the inner cost of a malfunctioning tool without stopping periodically the production cycle. Despite control designers are able to anticipate the fault emergence with high precision, their unpredictable nature makes the inversion effortless when they appear earlier than expected.

Fault-Tolerant Control (FTC) has been postulated as a solution to avoid these troubles, as they prepare the controller to surpass the fault and keep the system working without decreasing the product quality [5]. Control designers have a wide

trajectory implementing one of its approaches, the Passive Fault-Tolerant Control (PFTC) [150], [100], nonetheless, these controllers lack the flexibility needed to avoid different faults, specially when they have a wide harm-grade. Opposite to that tendency, Active Fault-Tolerant Control (AFTC) approach [96], [116], [56] offers a versatile environment as they are designed to detect, grade and isolate the fault source in a first instance during the Fault Detection and Isolation (FDI) phase and, afterwards, modify the control law to overtake the fault during Control Redesign (CR) phase without stopping the system.

The AFTC approach presented on this thesis sought a *robust* control algorithm prepared to work in industrial conditions, *flexible* enough to be adapted when the manufacturing cycle conditions vary and prepared to be *upgraded* easily without stopping the production cycle. These properties are accomplished through a novel methodology based on Advanced Process Control (APC). On the one hand, FDI phase has been improved introducing Neural-Networks into the detection, grading and isolation mechanism. On the other hand, CR phase has been upgraded with Adaptive Controllers in order to modify dynamically the controller response.

ACs have multiple approaches, being the Model Reference Adaptive Control the one implemented in this thesis. This algorithm improves conventional PID controllers introducing into the control loop three major improvements: a model reference, an adaptation mechanism and an adjustable controller. The study developed for the first one substitutes the mathematical equations employed as the reference model with a Digital-Twin, while the second one replicates the machine behaviour, and, the last one sought to acquire compatibility between the previous control designs and the new techniques. Lastly, MIT and Lyapunov adaptive rules [13], [90], [133], [70] have been analysed and improved introducing an Anti-Windup mechanism capable of adjusting the reference signal to the new saturation in the faulty system.

The control techniques used in the adaptive rules increase the overall robustness of industrial systems, while the DT implemented in the reference model brings MRACs with a flexible platform prepared to modify these controllers attending to the production cycle requirements. Despite these benefits, the adaptation mechanism reduces its efficiency when the fault harm-grade increases [64], [142], [143], [149]. Integrating a Bank of Controllers structure in the CR phase allows to overcome a wider number of faults, as the information provided during FDI phase is used to pick the most efficient adaptive gains to get over the fault.

These MRACs are suited for industrial applications, such us hydraulic-actuators, vacuum pumps and conveyor belts, as the enhances have been designed to be compatible with this environment [113]. Across this chapters, the methodology and techniques introduced into AFTC are presented, where the following three sections, 3.1, 3.2 and 3.3 research the Model Based Technology, Adaptive Control Algorithms and Advanced Manufacturing Techniques, respectively.

3.3 Advanced Manufacturing Techniques

Endemic species describe endless loops, where they born, grow, reproduce and die, abandoning their descendants with the task to perpetuate this cycle. The hostile environment ensures only the adapted fathers give birth to the next generation, surviving only the individuals with the optimal features. Litters conquer natural ecosystems through this infinite process, expanding within the territory and perpetuating their families.

Industrial environments are far from this evolution procedure. Their controllers are designed during the machine dawn, experiencing from small improvements until the system dusk. Across the company production lifetime, their manufacturing lanes suffer several adjustments; however, machine controllers remain identical with fewer modifications on their algorithms. Instead of perpetuating the cycle, the control techniques become obsolete and antiquated.

Operators hardly upgrade machine control algorithms, as they encounter unintelligible codes. Instead of an artificial evolution process prompted by them, the controllers become obsolete programs. They are born with a single commitment, incapable of adapting their current features to the modern manufacturing requirements. This single cycle methodology decreases machine productivity and increases manufacturing costs, lessening the factory profits.

In addition to the evolution process, species survive in the environments creating communities. For instance, when the resources became scarce on Galapagos island, Darwin's finches specialise in feeding through several sources. They accomplish these biological function through the unique features that they have heritage from their parents. However, their success was tied to the societies they create on each island, as they cooperate to encounter and extract resources, transferring these advantages to the next generation.

These societies have individuals sharing an identical feature and additional self representative characteristics. On an island, the finches of the shame community have long sharped beaks to catch worms, but they have distinct wingspan. This thesis proposes a novel approach where machine controllers behave as natural species, that is to say, they create a cluster of controllers participating in a common objective, but each one has their particular algorithms. This methodology has implemented a Bank of Controllers, where the enhances brought to MRACs algorithm are maximized, as several adaptation gains coexists simultaneously, creating an upgradable platform for controllers.

Due to this technique, machine controllers have several algorithms available simultaneously, selecting automatically the distribution that maximises the production. This context fosters the introduction of FTC techniques into manufacturing machines, as each fault case is surpassed by their particular controller optimised for this situation. Instead of a general MRAC for each possible fault circumstance, each case has its own adaptive gains. This platform favours the artificial evolution of controllers, upgrading each branch individually without altering the previous algorithms. Controllers populations growth at a constant rate, as additional adaptive gains are introduced without rejection.

3.3.1 Industrial Environment

When manufacturers invest in a factory, they purchase the equipment to produce merchandise that fulfils markets necessities. During this initial inversion, they acquire all the direct and indirect assets required for the manufacturing lane, such as machines, operators, controllers, actuators and sensors. Afterwards, they initiate the commissioning process, where operators install and configure this machinery. When this procedure concludes, the system is ready to produce.

The commissioning process involves a considerable amount of entities, as each distributor configure its machine. This layout requires at least a technician that sets up the machine manually, being necessary to displace this resource to the factory location. Besides, this installation usually lasts for several weeks, significantly increasing the economic charge of this stage.

This process consumes a significant amount of the manufacturing lane project resources, reducing considerable investors attitude towards improving machine controllers before their initial tuning. Furthermore, the distributor programmer tunes the machine controller on a first instance, but after this commissioning stage, the subsequent redesign is accomplished by the factory operators with fewer knowledge about the original program.

Despite their knowledge in control techniques and machine behaviour, before improving the current program, they have to understand its logic. They should avoid interrupting the cycle while the manufacturing lane is active, restricting these assignments to the maintenance periods. Besides, during this halting stages, they perform additional assignments, decreasing the margin to accomplish these supplementary duties. On this conditions, operators have to suspend the manufacture lane for more elongated sessions to adjust controllers, reducing the economic benefits.

Due to the difficulties presented to upgrade industrial controllers, commonly, machines maintain an identical control algorithm as their initial tuning across its lifetime. Instead of progressive evolution, factories grow over the years without optimising their manufacturing lanes until their performance is below their competitors. This practice matures the industrial park, as massive incomes of capital are required to upgrade control algorithms.

3.3.2 Bank of Controllers

Society gluttony devours natural resources in an endless cycle where they are continuously growing and accentuating this loop. Industrial processes satisfy this craving partially, exploiting the ecosystems to manufacture products vertiginously before consumers digest their former belongings. Control designers have sought novel manufacturing techniques to stabilise this infinity process and regain the balance, shortening production rates and consuming fewer resources to deliver high-quality commodities.

Although these techniques alleviate the exorbitant amount of products demanded by society, they extenuate operators and machines. The former has to research the current controller program, altering it during shorter non-productive

periods, such as maintenance policies, while the latter has to force its actuators, adapting their performance to the new controller requirements. This situation tires production lanes, increasing their inclination to suffer from faults. Adaptive Controls patch the machine wounds until operators treat them correctly, but they are far from being an adequate remedy.

This thesis has extended another nature concept into the industrial environment to increase ACs feasibility. Endemic species evolve accordingly to their surroundings, adapting their features to ensure their survivability. However, they also present divergences between their members. These ramifications reach their zenith when their individuals dwell similar environments with a significant variance in at least one characteristic. This situation gives birth to the races, where individuals of the same species differ on some features, but they collaborate as a single unit to ensure their survivability rate against a broader range of environments.

In this concept, individuals have their particular characteristics, but they share a common objective. Each member specialises in an assignment accordingly to their features, learning how to solve a singular problem faster. As each member performs their task optimally, these communities grow faster, increasing the overall welfare passively. The industrial doppelganger is a platform that has a standard control objective with several ramifications attending to the current machine behaviour, that is to say, a controller that switches its algorithm automatically when new dynamics emerge on the plant.

The Bank of Controllers architecture fits these requirements, as they introduce platforms that are compatible with several algorithms. Distributors program their machines with an initial control architecture, for instance, a PID algorithm. During the commissioning stage, operators tune their gains accordingly to the manufacturing lane requirements. Similarly to native species, the BCs technique generates a common control architecture were several races, or in this case, algorithms, coexisting to maintain the production optimised, as they switch between them automatically attending to the production requirements. Instead of a unique architecture, factories continuously adapt their manufacturing lanes accordingly to social demands. Their main advantage over traditional controllers resides on the fact that they preserve previous algorithms, but simultaneously, they are compatible with multiple techniques at once, rearranging them automatically.

BCs techniques are also extendible to other fields, for example, into ACs. When a source alters the plant dynamics, the adaptive gains adjust the controller signal accordingly to continue tracking the reference without destabilising the system. This thesis understands faults as the source altering the plant dynamics, so the adaptive mechanism sought to reduce their harming effect without spreading it or compromising the production.

This adaptation mechanism optimised the controller signal through the adaptive gains. They determine the adaptation rate, as they contribute fastening or relaxing the time spent until nullifying the error between plant and model responses (see Section 3.2). This thesis presents a novel BCs focused on the adaptive gains, allowing an automatic switch between them accordingly to the fault.

Industrial machines present several fault incidents, altering their behaviour in contradictory tendencies. Despite the adaptive gains effort to regain the optimal

performance, under this dissimilar conduct, they are unable to recover the system efficiently with a single collection. This thesis proposes a novel methodology, where rather than configuring a unique adaptive gain, each fault case has its particular assemblage.

During the FDI phase, operators consider several fault cases, replicating the machine behaviour under these conditions in the Digital-Twin to generate the fault database. Afterwards, they applied this information to train the NNs detection mechanism (see Section **??**). After these experiments, they have recorded the system performance under the fault effect, tuning the adaptive gains accordingly to each one of these conditions. BCs assemble these configurations into a universal platform, switching automatically to the gain that surpasses the fault more efficiently (see Fig. 3.18).

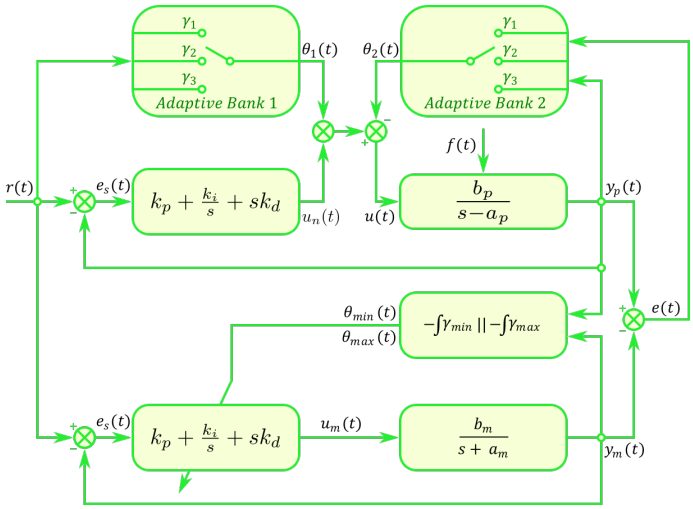
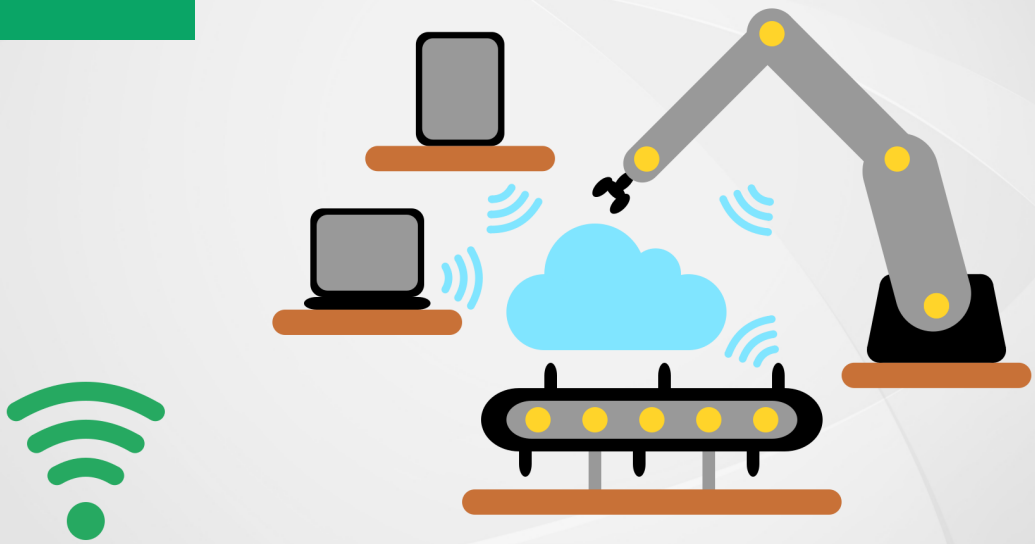


Figure 3.18: Improved MRAC structure through a Bank of Controllers.

This methodology dramatically improves MRACs performance against mechanical faults, as they always attempt the recovery process with the optimal adaptive gains. Due to the inclusion of the BCs technique, machines have an upgradable MRAC that responds actively to surpass faults, redesigning the controller accordingly to the source, maintaining the system stable without interrupting the production.

4 Experimental Study of the Enhanced MRAC



Natural habitats present an aggressive environment where species are continually fighting for their survivability. They are threatened by multiple entities, from their direct predators to other members of their classes, prompting a race where only the most adapted individuals survive. As an example, a reindeer has to compete for food against other members of its species, but it also has to escape from wolves. On this competition, the capacity to overcome unexpected situations is crucial, as only a rapid response against critical conditions ensures the persistence.

Despite factories are artificial ecosystems built by humans, machines suffer from a similar survivability process in this aggressive environment. Instead of escaping from predators, they are hunted by faults. These sources modify the manufacturing machine performance negatively, as they reduce their efficiency. Due to this obsolescence, the production cadence diminishes alongside the industry benefits; or expressed as a natural analogue, the factory is devoured by its competitors.

This thesis has minimised the fault effect over manufacturing machines improv-

ing their current controllers with Fault-Tolerant Control techniques (see Chapter 3). They differentiate from the current procedures on its Control Redesign stage, which has been updated through Adaptive Controls. From the multiple alternatives they offer (see Section 2.3.4), MRACs have been selected as the case under study. These control algorithms offer a versatile platform to introduce enhanced into three major areas (see Fig. 4.1):

1. **Model Based Technology:** In the natural environment, individuals learn their behaviour from their parents, which taught them how to survive in an aggressive environment. Similarly, in the MRAC, the reference model illustrated the optimal performance without fault to the primary system. Instead of a transfer function defining the plant dynamics, this stage has been enhanced with a Digital-Twin imitating the plant behaviour (see Section 4.1).
2. **Adaptive Control Algorithms:** In the natural environment, when individuals suffer from an injury, they attempt to minimise the pain while they are healing to continue escaping from the predators. Similarly, the MRAC adapt the controller performance to the new plant dynamics or faults to minimise the disturbances and continue tracking the reference. These algorithms have been upgraded even further implementing an additional adaptive loop in the reference model to increase the system stability (see Section 4.2).
3. **Advanced Manufacturing Techniques:** In the natural environment, individuals live in communities to increase their possibilities to survive against the predators. Similarly, the adaptive algorithm has been upgraded with a Bank of Controllers. This technique switches between multiple adaptive gains until encountering the situation that minimises the tracking error faster and with fewer disturbances (see Section 4.3).

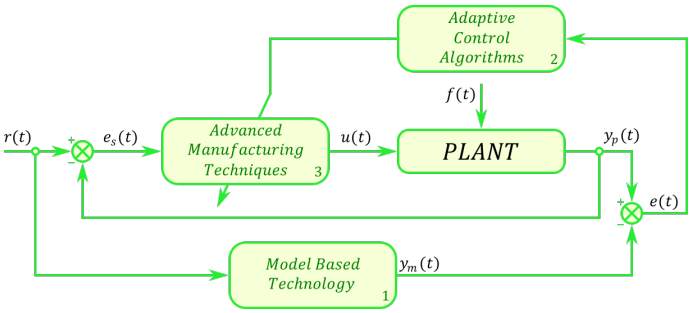


Figure 4.1: Schematic view of the novel MRAC features.

Previously, these improvements have been intensely studied, presenting across this chapter a case of study based on an industrial Hydraulic-Press for metal sheet stamping. This system has been designed from zero, that is to say, the subsequent pages present the designing procedure of a manufacturing machine accordingly to the novel methodology introduced in this thesis. When a fault emerges

on the system, the enhanced MRAC mitigates its adverse effect and recovers an optimal performance in the machine (see Fig. 4.2).

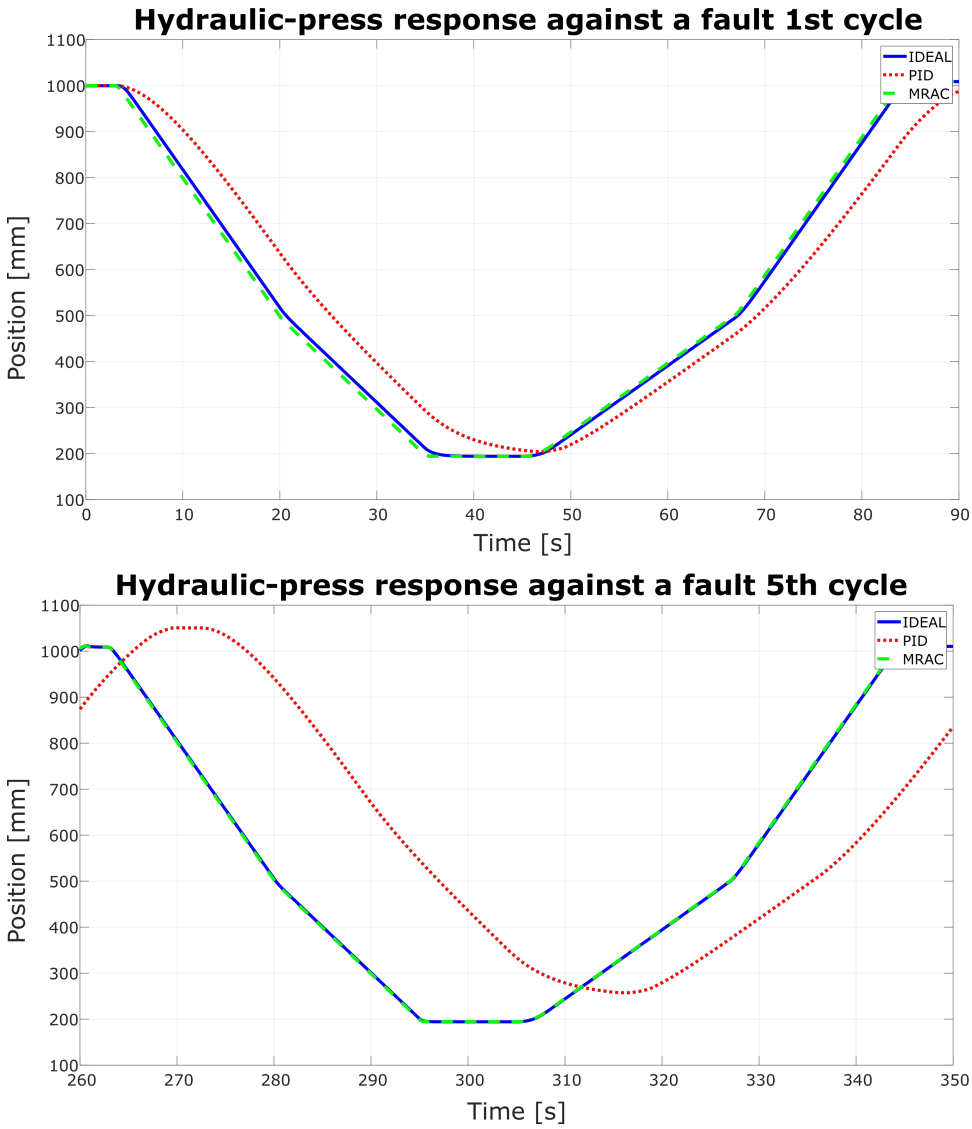


Figure 4.2: Overview about the Fault-Tolerant Controller proposed on this thesis.

4.1 Control Redesign: Model Based Technology

Control design techniques are constantly evolving, adapting their older versions to surpass previous quandaries or researching for novel algorithms to maintain an heterogeneous amount of systems stable. Adaptive Control techniques belong to this last improvement, as they introduce algorithms designed to minimise the effect of emerging unmodeled dynamics on plants. They have several topologies (see Section 2.3.4), studying in this dissertation Model Reference Adaptive Controls applied into the Control Redesign phase of Fault-Tolerant Control. The research sought to maintain industrial systems stable despite the emergence of unknown dynamics, that is to say, minimise the adverse effect of faults in manufacturing machines.

Despite their multiple benefits, these control algorithms lack the flexibility, robustness or upgradability required in manufacturing systems to overcome faults without spreading their harming effect or decreasing the production. This thesis presents three enhances developed over MRACs to introduce them into the FTC methodology. From the enhances, this section proves the research done to increase MRACs **flexibility** (see Section 3.1), introducing Digital-Twins into its reference model stage.

This Model Based Technology enhances conventional MRACs, substituting the current model reference stage for a flexible platform compounded of the DT and the controller (see Fig. 4.3). This Real-Time model reproduces identical responses as the industrial machine, emulating the system performance in the factory, as it has been connected to a replica of the original system control algorithm. This new approach substitutes the mathematical expressions describing the system closed-loop behaviour with a flexible platform that adjusts its performance automatically accordingly to the production requirements.

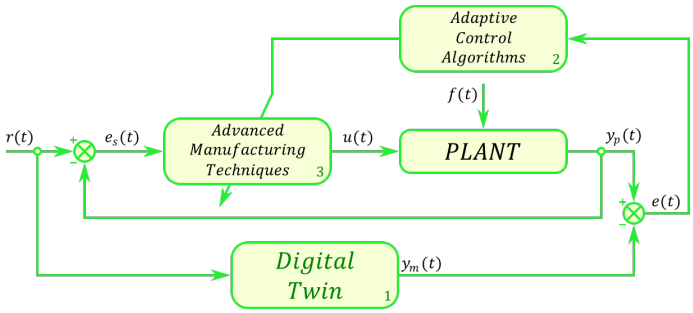


Figure 4.3: Schematic view of the novel MRAC features: Model Based Technology.

The following pages present the system design process, from the machine schematics to a Digital-Twin (see Section 4.1.1). The study continues comparing the Real-Time responses from Simscape™ library against the proposed ikSimscape counterpart (see Section 4.1.2), analysing how the current machine controllers are integrated into the model (see Section 4.1.3). Lastly, this Digital-Twin

is exported into the Beckhoff Hardware in the Loop platform to validate its performance (see Section 4.1.4).

4.1.1 Digital-Twin

Digital-Twins have brought new possibilities in the field of model design, as they reproduce an identical performance as their real counterpart. Nonetheless, several issues have been addressed before this responsive models are optimal for industrial environment. The heterogeneous nature of the physical equation defining these models difficult their simulation in Real-Time. Through the novel modelling techniques presented in this thesis (see Section 3.1), DTs become a feasible solution to reproduce the behaviour of manufacturing machines.

Heavy machinery, such as Hydraulic-Presses, combine several elements with different physical properties. For instance, hydraulic-actuators have a mechanical piston pumped by hydraulic fluid, with an elongation determined by electrical limit switches. This thesis proposes a novel methodology to join these different attributes under a unique Digital-Twin (see Section 3.1.4).

The proposed methodology divides the DT design into three stages. During the first one, operators select the components from data-sheets to previsualize the machine distribution on its schematics. Across the second stage, they pick the components from an existent library, ikSimscape (see Section 3.1.3), plugging them in a software environment to assemble and configure the machine segments attending to the parameters extracted from data-sheets. Finally, during the third stage, the DT is exported from the software environment into a hardware platform, connecting model and controller to replicate the factory conditions.

Across the subsequent paragraphs, this thesis exhibits the process followed by a Hydraulic-Press during the second and third stages (see Sections 4.1.1.1 and 4.1.1.2). The proposed system merges hydraulic, electrical and mechanical parts successfully. The component library designed for industrial Digital-Twins, ikSimscape, combines these distinct physical properties into a unique model to reproduce in Real-Time the machine performance.

4.1.1.1 Software Design

Hydraulic-Presses combine multiple physical properties, as they are assembled by a hydraulic feed circuit, translational mechanical cylinders and electric actuators and sensors. These manufacturing machines are widely extended in industries, as they have multiple applications, such as slice rolls, displace loads or mould pieces. Fagor Arrasate, the company supporting this research, has offered the schematics of several pre-existent manufacturing machines, combining their characteristics in the Hydraulic-Press presented as a case of study.

These machine schematics have provided information about the Hydraulic-Press structure (see Append A). Similarly to them, the model presented distributes the components accordingly to these drawings, parametrising them through the information gathered from data-sheets. In the case of study, the Hydraulic-Press has two cylinders, an upper and lower part that compress metal-sheets to mould

pieces for vehicles (see Fig. 4.4). The upper part or slide applies the compression, while the bottom part or cushion hold on the piece until obtaining their final appearance.

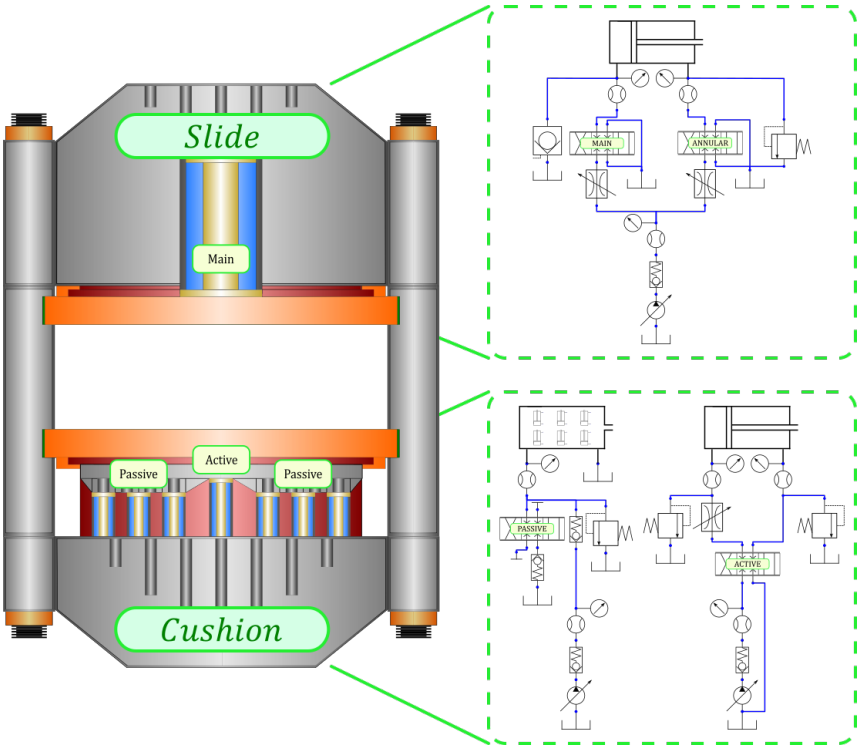


Figure 4.4: Overview of the Hydraulic-Press: Slide and Cushion schematics.

The design stage concludes when the Hydraulic-Press schematics are available, beginning the Digital-Twin modelling (see Section 3.1.4.1). This software design stage initiates selecting the components from ikSimscape library and placing them into a blank Simulink® work-sheet. The schematics dictate their position, assembling them to replicate the Hydraulic-Press distribution and generating the Digital-Twin. Lastly, each component configures its parameters with the information brought from data-sheets. Although the Digital-Twin behaves as an entire manufacturing machine, two main modules compound it, slide and cushion.

The **slide** (see Fig. 4.5) is the upper mobile part of the Hydraulic-Press commanding the top compression force over the metal sheet. Two proportional valves, one for each chamber (main and annular), order the cylinder. The controller ensures they have an opposite behaviour bypassing their overlap. When the proportional valve connected to the main chamber moves to the fill status, their counterpart in the annular chamber steps into the vacant position and *vice-versa*.

The **cushion** (see Fig. 4.6) is the lower mobile part of the Hydraulic-Press in charge of the bottom compression force over the metal sheet. Its base conglomer-

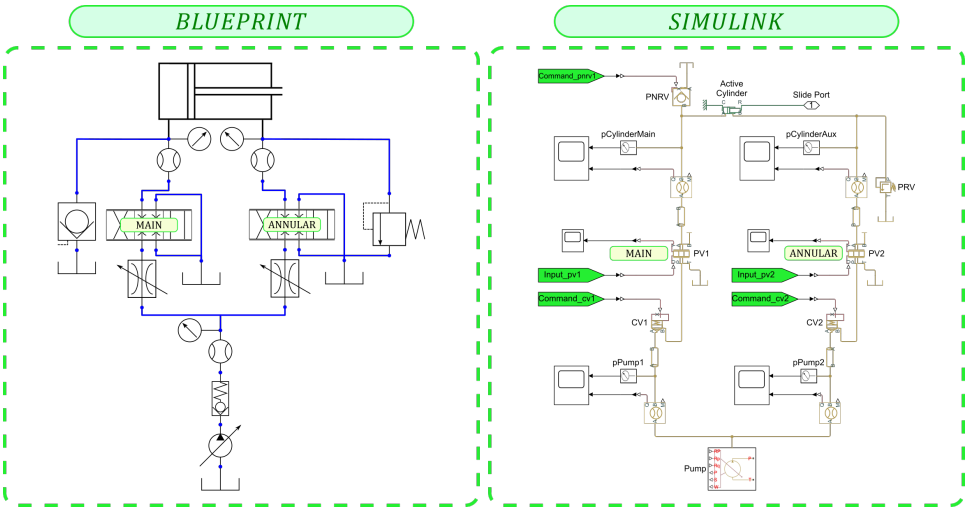


Figure 4.5: From blueprint to Simulink®, generation of the slide Digital-Twin, compounded of one active cylinder

ates seven cylinders divided into two categories: active or passive. Regardless of their trait, they have a proportional valve commanding the flow rate through their chambers. Their divergence resides on how they behave during the compression stage. The active cylinder displaces the base compressing the passive cylinders until the force exerted matches the requirements to mould the piece.

Before finishing the software design stage, this dissertation validates the Digital-Twin performance connecting the model to a PID control algorithm identical to the one available in the factory. During this validation, the Hydraulic-Press components have suffered adjustments to their parameters until the virtual sensors signals coincide with the manufacturing machine responses (see Section 4.1.4). This iterative process adapts the parameters attending to the information brought by data-sheets, selecting the point in the diagram representing with higher accuracy the real component behaviour.

4.1.1.2 Hardware Design

As previously explained, the Digital-Twin corroborates its behaviour under a Software in the Loop validation platform. These experiments ensure an identical performance between the HP model and the manufacturing machine installed on the factory. When this validation is over, the DT is exported to a Hardware in the Loop platform to initiate the second validation phase.

This dissertation presents a validation platform based on Beckhoff technology, embedding the system under test, the Hydraulic-Press DT, into their Industrial PC. This HiL platform has two well-defined stages:

- **Controller:** This module encapsulates an identical copy of the original con-

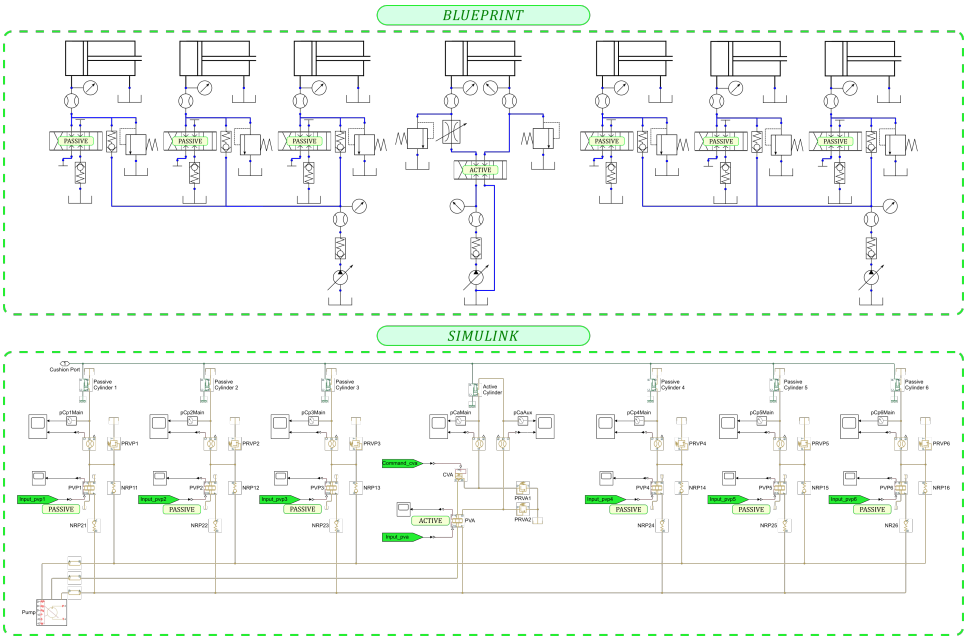


Figure 4.6: From blueprint to Simulink®, generation of the cushion Digital-Twin, compounded of six passive cylinders and one active.

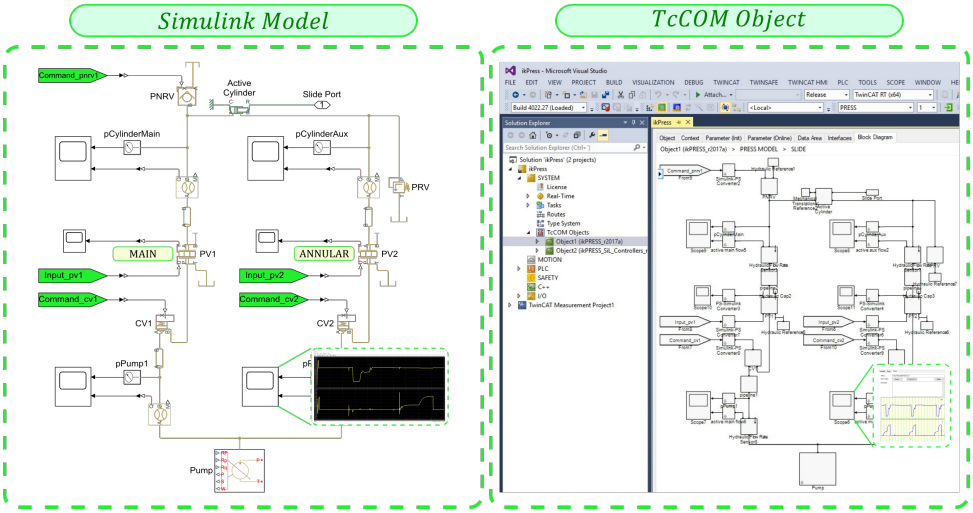
trol algorithm available in the manufacturing machine. Fagor Arrasate has offered authentic machines controllers tuned in a Beckhoff PLC, integrating them perfectly into the HiL validation platform.

- **Plant:** The Simulink® compiler exports the DT from this software into a TcCOM Object, a TwinCAT 3 programming block that encapsulates the physical equations describing the model behaviour. The exported DT reproduces in Real-Time the Hydraulic-Press performance without further reconfiguration of its parameters.

This Hardware in the Loop architecture replicates the manufacturing machine conditions in the factory, substituting it for a DT with identical performance. The platform brings a harmless environment to examine controllers, tuning its gains in the laboratory and exporting the architecture afterwards without further modifications. Both systems, plant and controller, exchange their status through a deterministic communication protocol, ensuring the Real-Time capabilities during the validation phase.

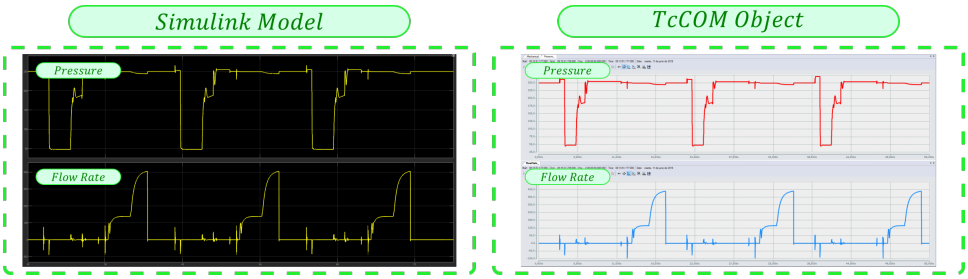
Across the exportation process, the DT configuration remains identical independently of the simulation platform (see Fig. 4.7). The TcCOM Object translates the Hydraulic-Press components directly, maintaining their parameters, position and connection. Due to this replication process, the model retains the physical constraints establishing the Hydraulic-Press operation. On this context, operators

manipulate the DT similarly as they would execute the manufacturing machine, configuring their characteristics attending to the production cycle.



(a) Simulink® Hydraulic-Press Digital-Twin

(b) TwinCAT 3 Hydraulic-Press Digital-Twin



(c) Pressure and flow rate responses

Figure 4.7: Comparison between the CPS generated in Simulink® against the TcCOM Object from TwinCAT 3.

The studies carried out in this section proves the presented design methodology. Digital-Twins configure their components parameters in an iterative process under a Software in the Loop platform, exporting them after their validation into a Hardware in the Loop platform. On this final stage, the Digital-Twin behaves as the system plant, connecting it to the controller through a deterministic communication protocol.

4.1.2 Real-Time Capabilities

The novel design methodology presented in this thesis brings a significant update over traditional commissioning, tuning the controller gains against a Digital-Twin rather than the manufacturing machine. Developing experiments under laboratory

conditions reduces the controller behaviour uncertainties drastically, as programmers obtain their performance in a harmless environment. However, these benefits require the Real-Time synchronisation between controller and DT signals.

Although simpler versions of Digital-Twins reach the equation convergence in a fixed time, the various physical properties involved on the industrial machine performance ruins their Real-Time capabilities. Some solutions provided slow-down simulations obtaining an optimal performance; however, these techniques fall apart when they synchronized their signal with the controller.

This dissertation presents a new library, called ikSimscape, prepared with industrial machine components designed at the data-sheet level (see Section 3.1.2). The proposed solution reduces the simulation time through Digital-Twins emulating the responses in the data-sheets, rather than complex physical properties. For instance, instead of defining the proportional valve opening ratio between ports, ikSimscape model components parametrize their behaviour through the flow rate and pressure curves.

Instead of requiring laboratory experiments to determine their constructive parameters, operators configure the Digital-Twin through simple data-sheet inspection. Although the simulation reaches the convergence faster than their counterpart, the DT replicates trustworthy the manufacturing machine behaviour (see Section 4.1.4). The following paragraphs demonstrate these assumptions experimenting with the performance of two hydraulic DTs.

The experiment carried out compares a hydraulic-actuator (see Fig. 3.4) completely designed with ikSimscape library against their Simscape^{TM} counterpart. Before presenting these measures, intermediate research has been carried out to determine how ikSimscape library modifies the equation convergence time.

Rather than duplicate the Digital-Twin with Simscape^{TM} components, both copies have the ikSimscape elements except for the proportional valve. The simulation starts under similar configuration parameters: an integration method based on Backward Euler algorithms and a fixed step size of two milliseconds. The response obtained from the hydraulic-actuator fully designed with the new library reaches equation convergence approximately 4% faster than their conventional counterpart (see Fig. 4.8).

On the next validation, the hydraulic-actuator components have been fully substituted for their Simscape^{TM} version. Including more elements from this library increases the bridge between both simulations times, reaching a difference of nearly a 14% (see Fig. 4.9). This experiment proves the previous assumptions, showing how Simscape^{TM} components require more time to attain equation convergence than its ikSimscape counterparts.

This factor becomes exacerbated in industrial machines, as they are compounded of more than fifty components. Simulating the models of the massive machines designed with traditional libraries requires step sizes five times bigger than the DTs generated with the proposed ikSimscape library.

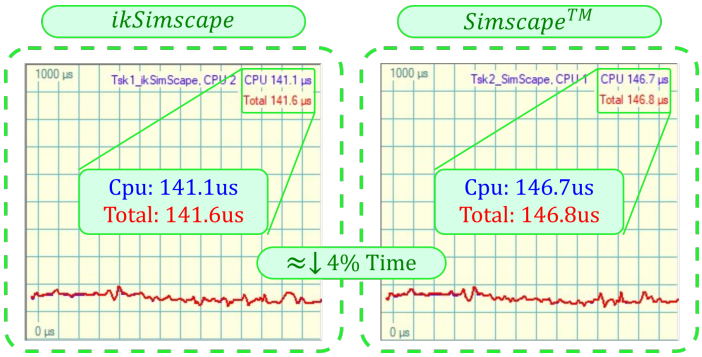


Figure 4.8: Comparison between the time required to achieve equation convergence in a Simscape™ proportional valve against its counterpart designed with the library presented on this paper.

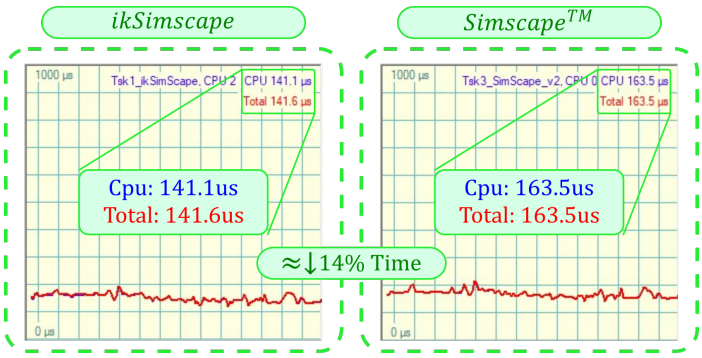


Figure 4.9: Comparison between the time required to achieve equation convergence in a Simscape™ proportional valve against its counterpart designed with the library presented on this paper.

4.1.3 Industrial Controllers

Since the early beginning of the factory automatization era, PID algorithms have offered a quick, reliable and economical solution for controlling the performance of industrial machines. Their auto-tune feature brings control designers with an effortless initial version of their gains, obtaining the final controller just by adapting the gains to the manufacturing machine specifications [52].

The Hydraulic-Press presented in this dissertation is not oblivious to this tendency, basing the control algorithm into a PID architecture. Across this research project, the controller has upgraded its features from a conventional PID to MRAC architecture. However, to validate the scope of the novel design process in similar conditions as the current manufacturing machines, this section replicates the conventional control design steps for a PID architecture.

During Software in the Loop validation phase, a primitive PID controller has

been tuned. This control algorithm benefits from Simulink® features, connecting it directly with the Digital-Twin previously assembled. Instead of tuning new controller gains, they have been approximated accordingly to the ones available in the manufacturing machines. The experiments validate the DT performance, studying their responses against their real counterpart.

The study has continued in the Hardware in the Loop validation phase, dividing the research into two experiments. The initial investigation has validated the DT behaviour replicating the factory conditions. Instead of the manufacturing machine, operators have connected the controller to the DT without suffering any adjustment. The new system replicates the original performance, validating the model and, therefore, the design process.

Although this study has focused on validating the DT performance against a PID controller, further streamlining is worth seeking. The HiL investigation continues integrating novel features into TwinCAT 3 control algorithms. Rather than create the controller entirely in Beckhoff's platform, the algorithm has been split between TwinCAT 3 and Simulink®.

Hydraulic-Press controllers are prone to this division, as they have three well-differentiated segments, whose characteristics are resumed on these points:

1. **PID Gains:** Traditionally, control designers tune the PID gains directly into TwinCAT; however, this dissertation introduces a new approach based on Simulink®. This second platform offers a set of toolboxes optimized for control design, allowing to tune the PID directly during the SiL validation stage. Afterwards, instead of exporting the DT exclusively, this procedure also extrapolates the controllers employed on the validation.

2. **State Machine:** The second unit defining the Hydraulic-Press performance belongs to a state machine. This control algorithm reproduces the slide and cushion movements across the Hydraulic-Press cycle stages (see Section 3.1.4). TwinCAT 3 belongs to the PLC programming languages family, that is to say, it executes their programs cyclically ensuring reaching all their instructions before the watchdog time-stamp.

3. **Commands:** Between the state machine and the PID architecture, there is an intermediate module controlling the proportional valve commands. This module regulates these valves behaviour, avoiding the chamber de-pressurization. Besides, it regulates the command signals for the safety valves, ensuring their proper performance when required.

Splitting the controller configuration between Simulink ® and TwinCAT 3 reduces the process design time and increases its feasibility. Lastly, the control algorithms are embedded into Beckhoff's hardware, synchronizing them with the DT through a deterministic communication protocol (see Fig. 4.10). This study reproduces, in a laboratory environment, the factory conditions, ensuring the controller compatibility with manufacturing machines.

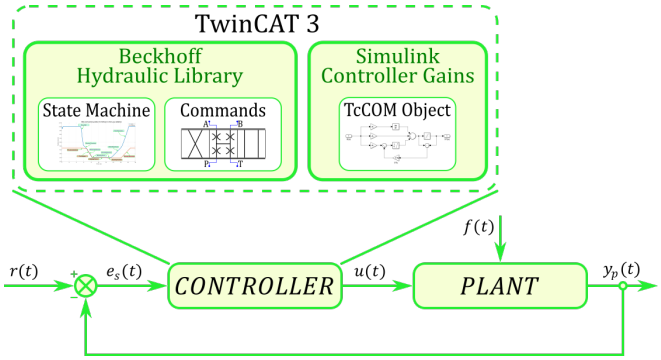


Figure 4.10: Control loop combining algorithms designed with MATLAB® control toolbox and Beckhoff hydraulic library.

4.1.4 Hydraulic-Press Performance

Across the following paragraphs, the Hydraulic-Press performance under the HiL validation platform has been presented. The DT has gone through every design process stage, verifying its behaviour initially under the software platform, and, later in the hardware. Controller and DT have synchronised their signals through a deterministic communication protocol, being EtherCAT the one picked for the case of study. This layout has been connected in the laboratory, configuring a testbench conformed of the DT model embedded in a Beckhoff IPC and the control algorithm from a PLC employed in similar HP (see Fig. 4.11).

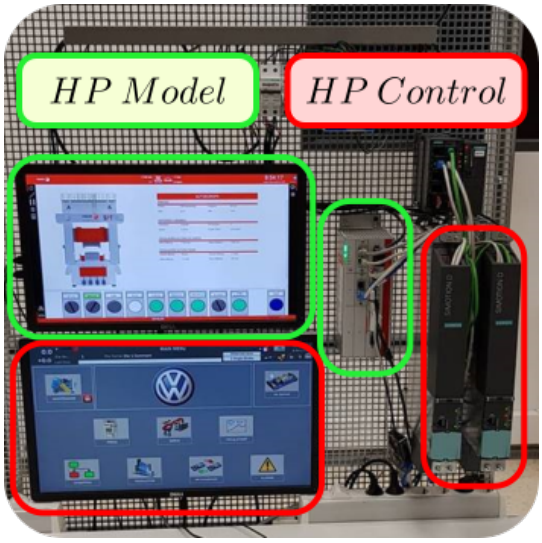


Figure 4.11: Layout of the HP model and control algorithm connected in the laboratory.

The DT performance study has been deployed through two strategies. The first round of experiments compares the performance of the manufacturing machine and DT studying the cushion force responses when the slide collides. The second round shows the Hydraulic-Press cycle around one iteration, altering the cushion recovery process. In both studies, the Hydraulic-Press model remains embedded inside the hardware platform to validate the proposed methodology.

On the first round of experiments, the Hydraulic-Press reproduces the force response when slide and cushion collide, whose value coincides with the compression strength exerted over the metal-sheet. The experiment is divided into three identical stages to examine the disparity between the manufacturing machine performance and its DT counterpart (see Fig. 4.12). Since the responses collected are identical, it is safe to assume that the Hydraulic-Press model replicates trustworthy its real counterpart behaviour.

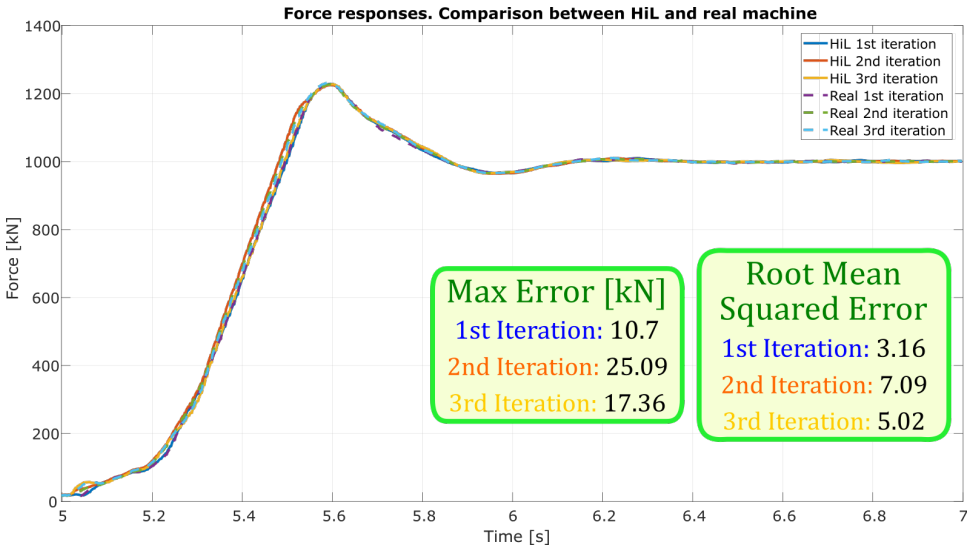


Figure 4.12: Comparison between the response obtained in the real system and the virtual model generated with ikSimscape library.

The experiments continue studying the Hydraulic-Press cycle. As previously explained (see Section 4.1.1), slide and cushion perform a repetitive cycle from their upper position to the bottom one where they collide exerting the force. After compressing and moulding the piece, they recover their rest status. Despite the slide always reports the same recovery movement, the cushion regains its spot in three options:

- **Normal Recovery:** After making force stage, the cushion descends its position slightly to decompress the piece. The moulded component would remain on its base while the system recovers its rest position (see Fig. 4.13).

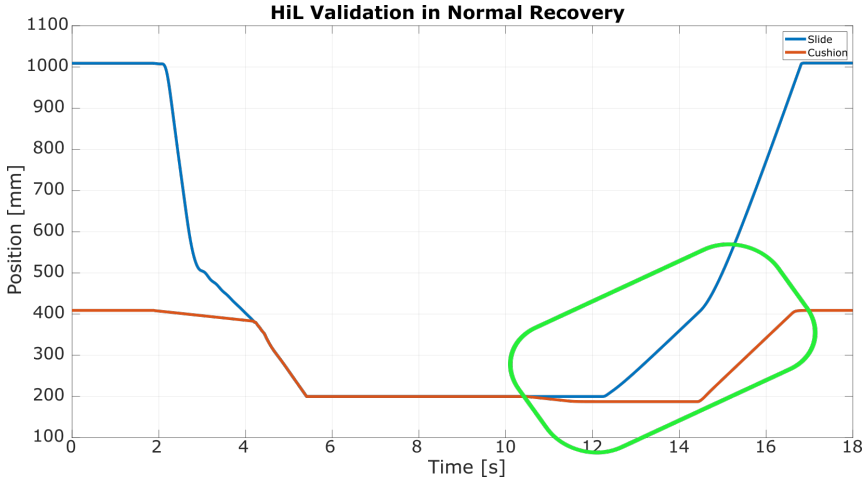


Figure 4.13: Hydraulic-press cycle obtained in the HiL validation platform with Cushion in Normal recovery mode.

- **Pick-up Recovery:** On this recovery, slide and cushion remain attached until arriving in an intermediate position, where a robot waits to pick the piece and displace it to the next stage in the productive cycle (see Fig. 4.14).

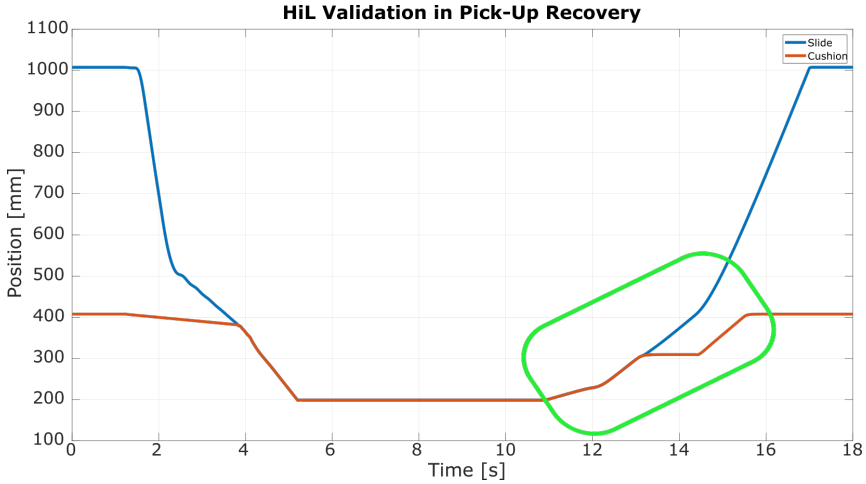


Figure 4.14: Hydraulic-press cycle obtained in the HiL validation platform with Cushion in Pick-Up recovery mode.

- **Accompaniment Recovery:** The last recovery maintain both cylinders attached until reaching the cushion rest position. During the restoration, they continue moulding the piece exerting a force over it (see Fig. 4.15).

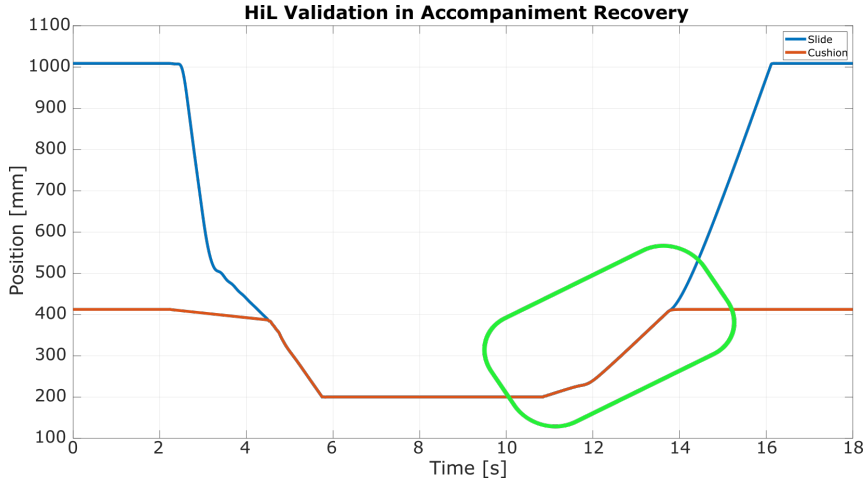


Figure 4.15: Hydraulic-press cycle obtained in the HiL validation platform with Cushion in Accompaniment recovery mode.

After completing the previous experiments, a supplementary study has been developed to determine the Digital-Twin performance across multiple iterations. Instead of a standalone simulation, the experiment reproduces the productive cycle, replicating the Hydraulic-Press factory conditions to simulate the manufacturing machine performance while it is being executed (see Fig. 4.16).

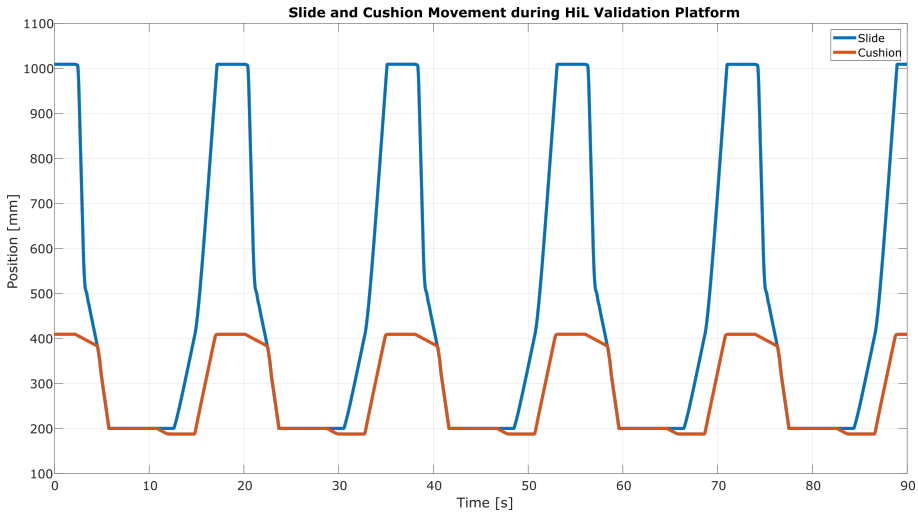


Figure 4.16: Hydraulic-press cycle simulated constantly under the Hardware in the Loop validation platform.

4.2 Control Redesign: Adaptive Control Algorithms

Native species ensure their survivability adjusting their behaviour proactively to the environment surrounding them. They are frequently searching for threats, as an early detection bring them enough time to alter their conduct and escape. Fault-Tolerant Control mimics this mechanism, as the Fault Detection and Isolation phase discern the faults while the Control Redesign phase surpasses them.

Model Reference Adaptive Controllers brought several benefits into the CR phase, as they adapt the control algorithm automatically to the faulty plant. However, industrial systems require robustness levels beyond the scope of this technique. This thesis proposes a novel approach, searching to increase MRACs **robustness** through an adaptable model reference stage (see Section 3.2).

The second improvement brought to the control algorithm enhance the conventional Adaptive Control Algorithms of MRACs. These advantages are split between system and reference adaptation mechanism (see Fig. 4.17). The share a similar principle, accommodate the control algorithm when faults emerge. However, while the first study reconfigures the machine controller, the second study adapts the Digital-Twin controller signal to the faulty plant mechanical boundaries.

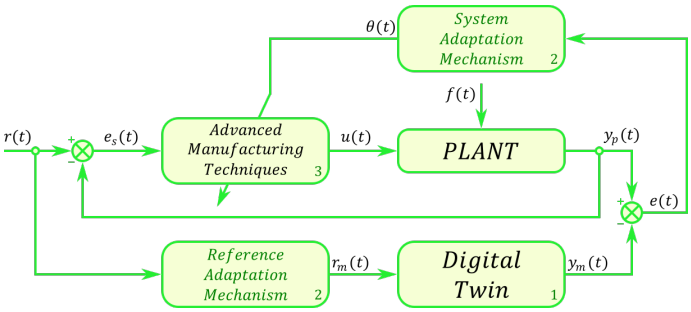


Figure 4.17: Schematic view of the novel MRAC features: Adaptive Control Algorithms.

This thesis has studied several faults (see Append B); nonetheless, this section presents the Hydraulic-Press behaviour against three cases, mechanical fatigue in the proportional valves (actuator), hydraulic losses inside the cylinder chambers (plant) and noise dissonances in the manometers (sensor). Each experiment compares the previously configured PID controller performance against the novel MRAC algorithm.

The research presents a similar structure (see Sections 4.2.1, 4.2.2 and 4.2.3) to manifest the Hydraulic-Press position dissonances. Each controller compares their performance against the faultless case, determining the maximum and root mean square deviation. The study conclude examining the control action necessary to maintain the system stable. An additional research (see Section 4.2.4) exhibits the adaptive reference model benefits, carrying out an identical investigation as the previous cases.

4.2.1 1^{st} Experiment: Actuator Fault

Across their lifetime, actuators frequently suffer from internal and external forces, such as mechanical contact, erosion and torque. These constant attacks wear away the component, materialising an adverse effect known as mechanical fatigue. Despite the actuator behaves appropriately at the beginning, after a certain amount of cycles this fault triggers a gradual detriment on its performance, degrading the component as its usage continues.

During the early stages, conventional PID controllers contain the fault harming effect, altering their behaviour to surpass it and to recover the nominal performance. However, prolong the fault reduces the controller reliability to a point where they end providing a proper service. When the system reaches this situation, it has crossed the region of degraded performance, compromising its stability and spreading the fault to other components.

Under identical conditions, the MRAC initiates the recovery process, that is to say, instead of spreading the fault, the controller detects its source and redesign its algorithm accordingly to mitigate the adverse effect and to maintain constant production. The performance of both controllers has been tested outlining a two phases experiment. The first phase presents the system behaviour degraded at 10% due to the fault, while the second one introduces a 90% degradation.

4.2.1.1 Hydraulic-Press Overview

The first point where operators advertise the fault effect is on the hydraulic-cylinder position. Under the faultless situation, the system performs an identical cycle independently of the controller (see Section 4.1.4). Nonetheless, the cylinder positions diverge from this optimal behaviour when the fault emerges. After detecting the mismatched performance, the controllers initiate the recovery process.

There are three architectures involved in the recovery process, testing on their limitations on this thesis. The first structure belongs to the PID distribution, while the second and third structures represent MRACs based on MIT or Lyapunov rules, respectively. These last controllers have their adaptive gains tuned accordingly to this fault situation. Each control structure has their own experiments, comparing the faultless response, named with the subscript Model, and the system response under the fault effect, named with the subscript Plant.

On the first study, the system subjects to a mechanical fatigue fault in the proportional valve. This drawback reduces in a 10% the component opening, decreasing the flow rate reaching the cylinder chambers from the pump. The performance is below the optimal point; however, the difference is practically nonexistent, and each controller maintains the production constant.

After the fault emergence, the PID controller performs a similar cycle as the nominal behaviour, remaining in the region of required performance. The cylinder velocity also endorses this behaviour, maintaining nominal values without significant disturbances or anomalies on the signal (see Fig. 4.18).

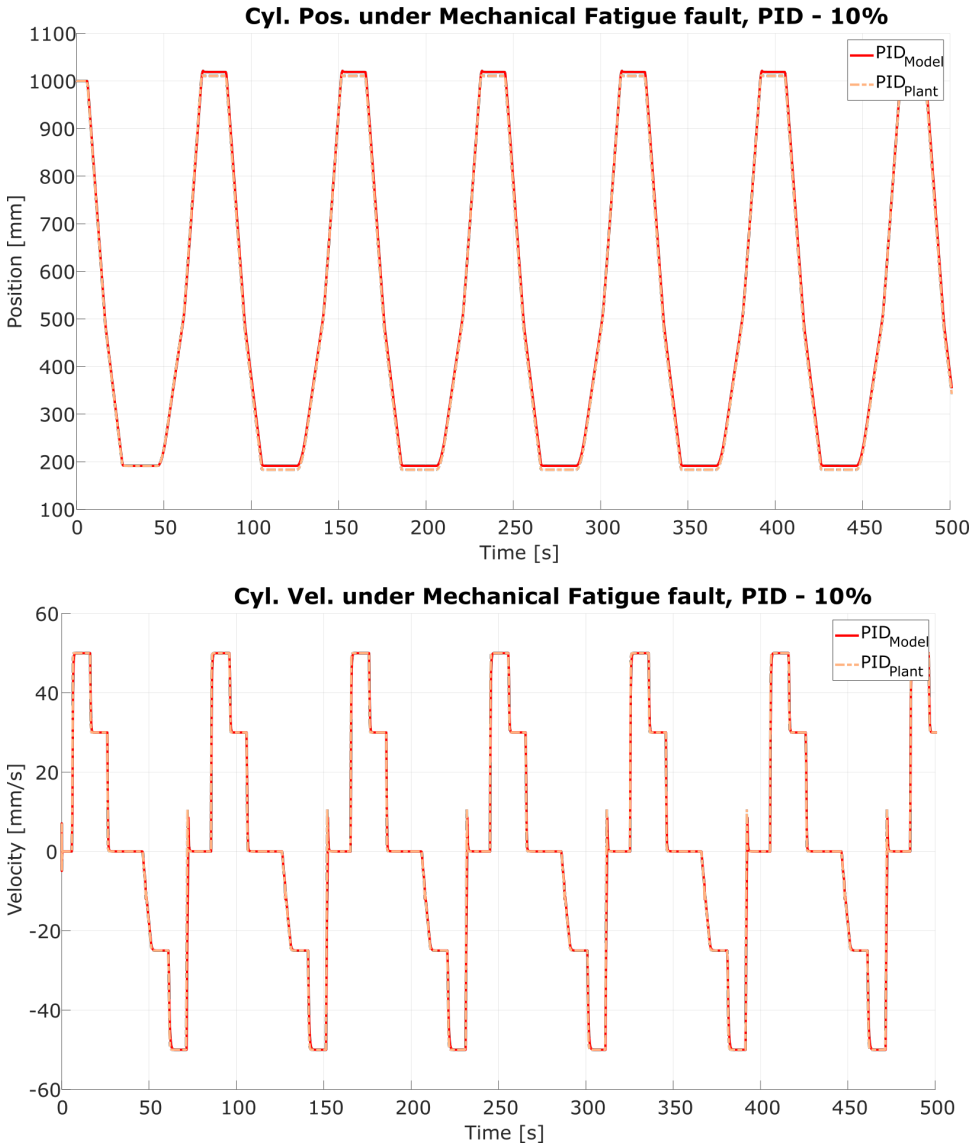


Figure 4.18: Hydraulic-Press Position and Velocity under *Mechanical Fatigue* fault for the PID controller, performance against a 10% fault.

The early interaction of the MRAC controller based on MIT rules ensures system stability. The Hydraulic-Press continue performing its cycle without any alteration, only manifesting a small disturbance on the first cycles due to the fault emergence (see Fig. 4.19).

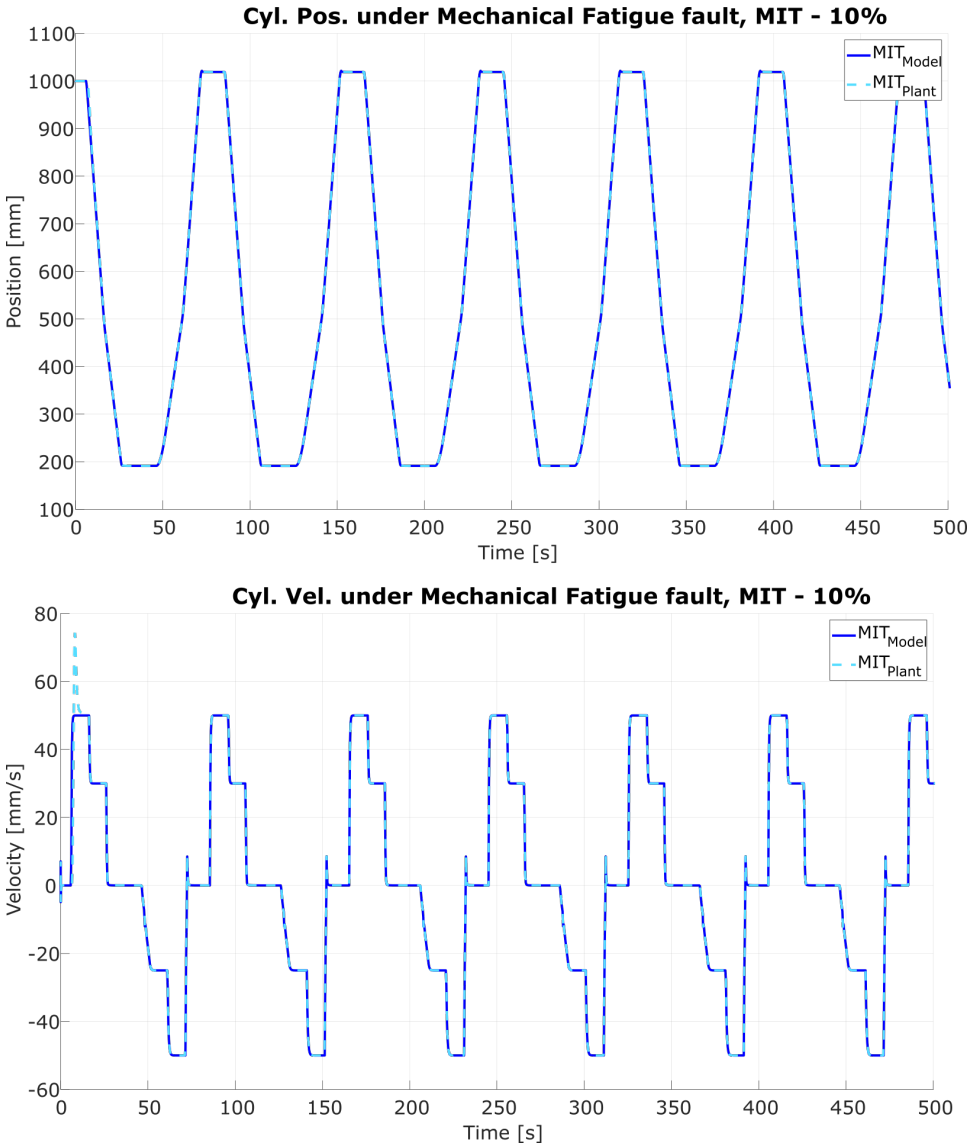


Figure 4.19: Hydraulic-Press Position and Velocity under *Mechanical Fatigue* fault for the MIT rule controller, performance against a 10% fault.

Lyapunov rules are also competent to surpass this fault, as they maintain the system stable without any alterations on its position or velocity (see Fig. 4.20). Their effect palliates the early disturbances produced on their MIT counterpart, adapting the signal smoothly.

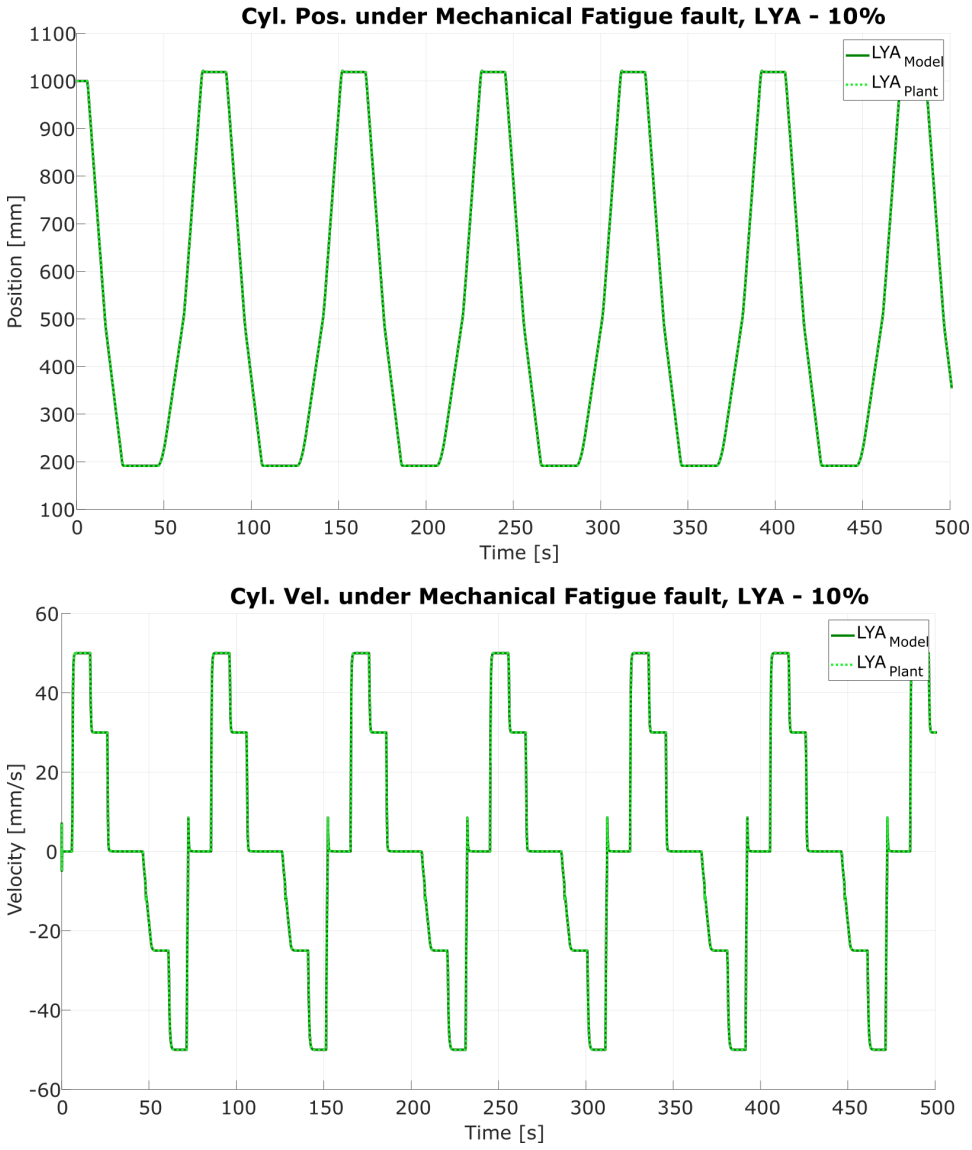


Figure 4.20: Hydraulic-Press Position and Velocity under *Mechanical Fatigue* fault for the Lyapunov rule controller, performance against a 10% fault.

On the second study, the fault has spread from 10% to 90% opening. The PID controller attempts to recover the optimal performance, but the lack of fluid in the chambers degrades its behaviour gradually. MRACs regain this status after a certain amount of cycle, but MIT rule requires longer periods to attain the optimal performance, as it has to auto-adjust properly its adaptive gains.

Although the PID controller surpasses the early stages of the fault, their performance becomes compromised when their harmful effect increases (see Fig. 4.21). The valve opens its channels below the optimal range, as the controller requires more time to reach the setpoints.

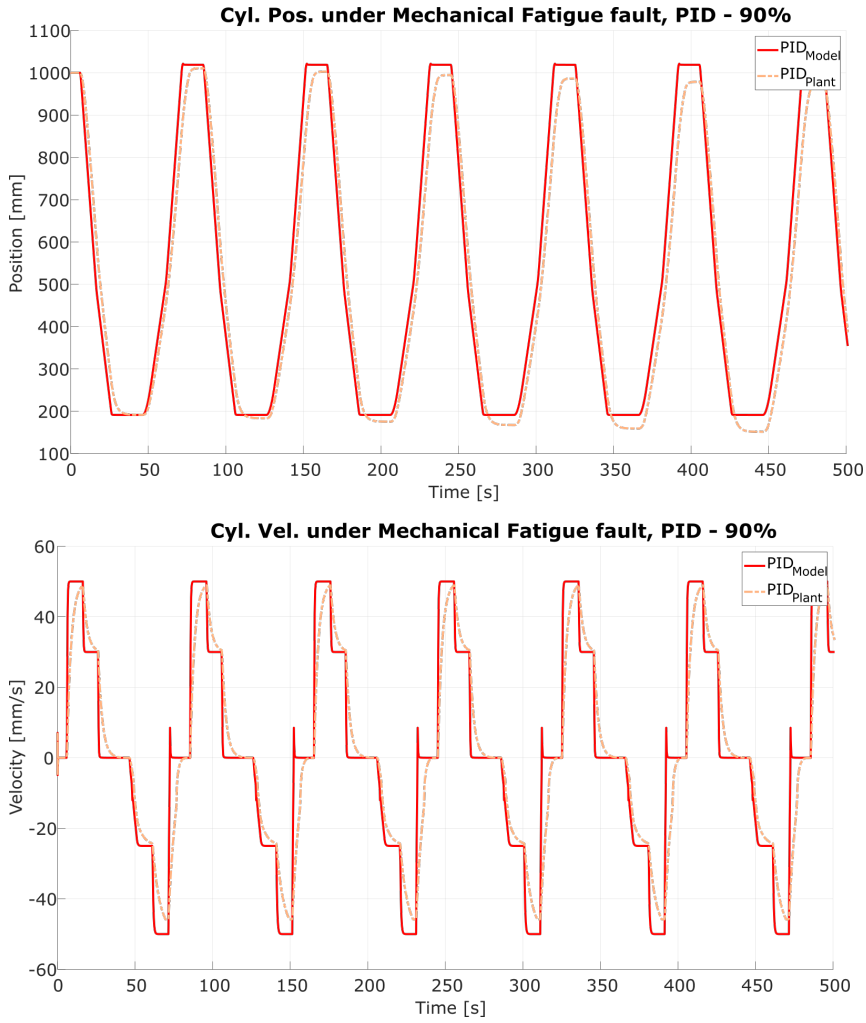


Figure 4.21: Hydraulic-Press Position and Velocity under *Mechanical Fatigue* fault for the PID controller, performance against a 90% fault.

When the fault emerges, MRACs based on MIT rule suffer to recover the signal while the control algorithm adjusts the adaptive gains. After this first cycle, the controller has recovered an optimal behaviour, entering the region of required performance and surpassing the fault (see Fig. 4.22).

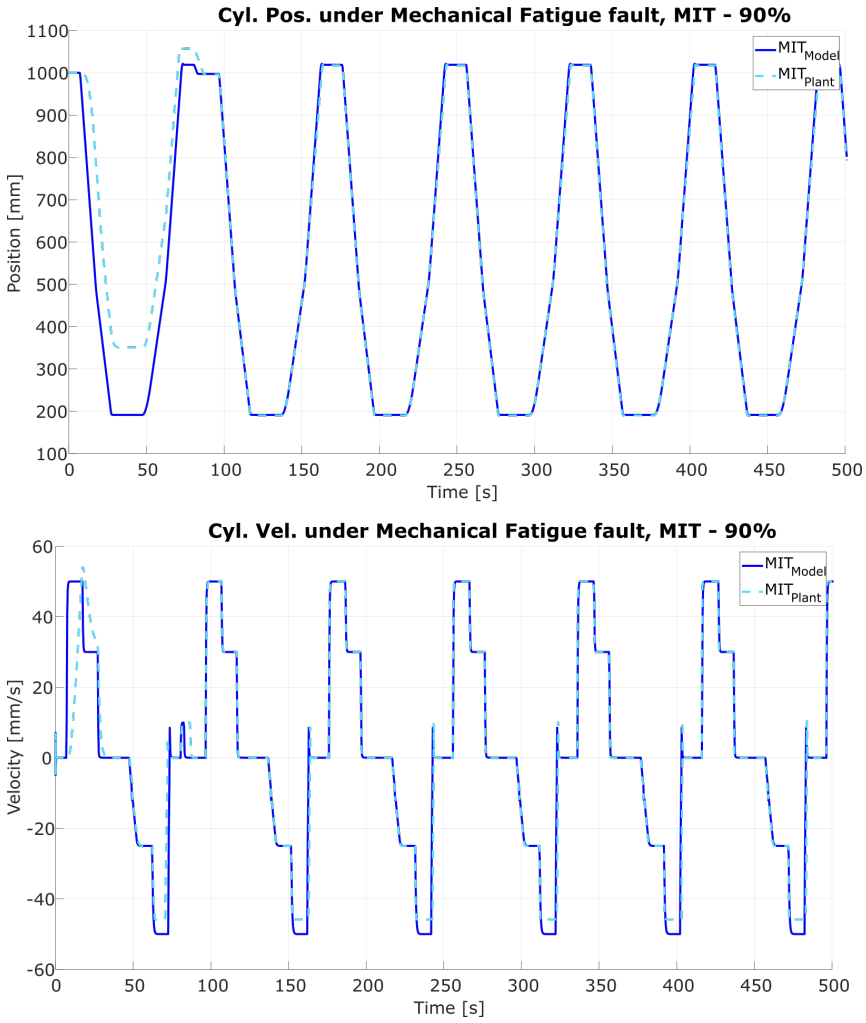


Figure 4.22: Hydraulic-Press Position and Velocity under *Mechanical Fatigue* fault for the MIT rule controller, performance against a 90% fault.

Lastly, the controller based on Lyapunov rules also recovers the nominal performance. They even maintain a closer response to the original Hydraulic-Press performance since the fault emergence (see Fig. 4.23). Although the control action maintains the stability, the fault has modified the valve opening limits below the controller saturation point, neglecting the velocity to reach lower values.

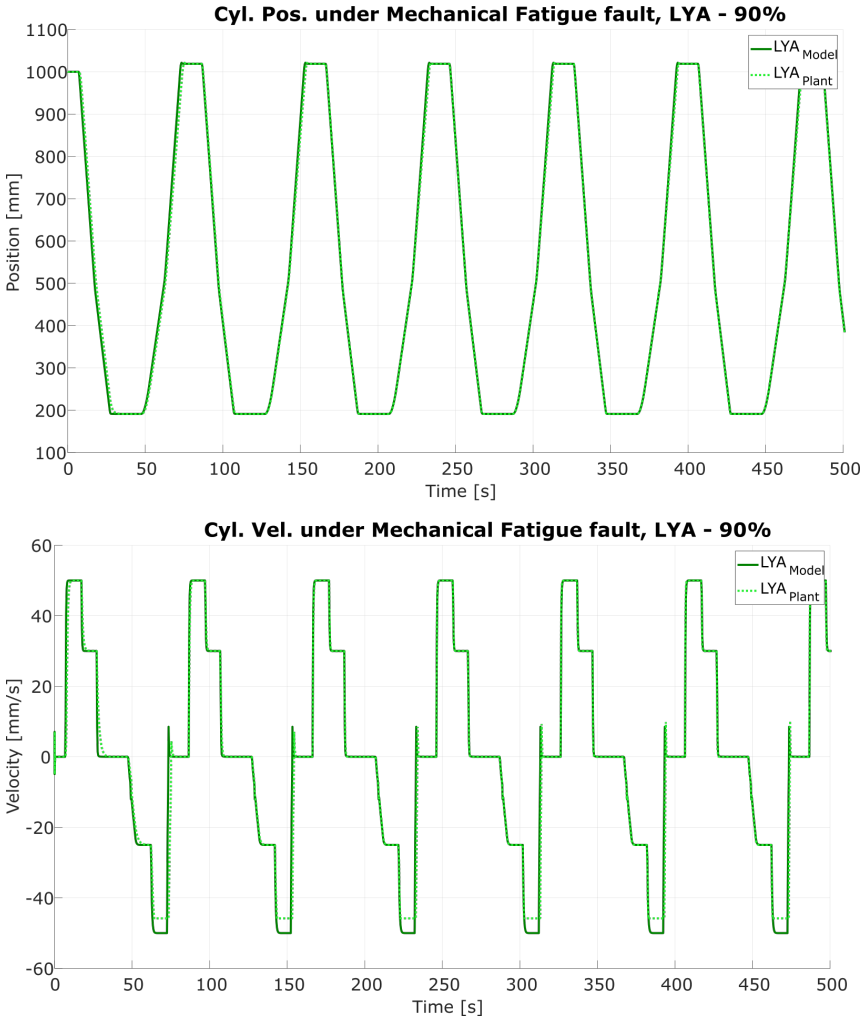


Figure 4.23: Hydraulic-Press Position and Velocity under *Mechanical Fatigue* fault for the Lyapunov rule controller, performance against a 90% fault.

Although conventional controllers maintain the system stable when actuators are under low impactful faults, their performance gets swiftly degraded when their adverse effect increase. Although conventional architectures based on PID controllers lack the robustness to maintain the stability when faults emerge, MIT and Lyapunov rules recover the optimal performance independently of their harm grade (see Fig. 4.24 and 4.25).

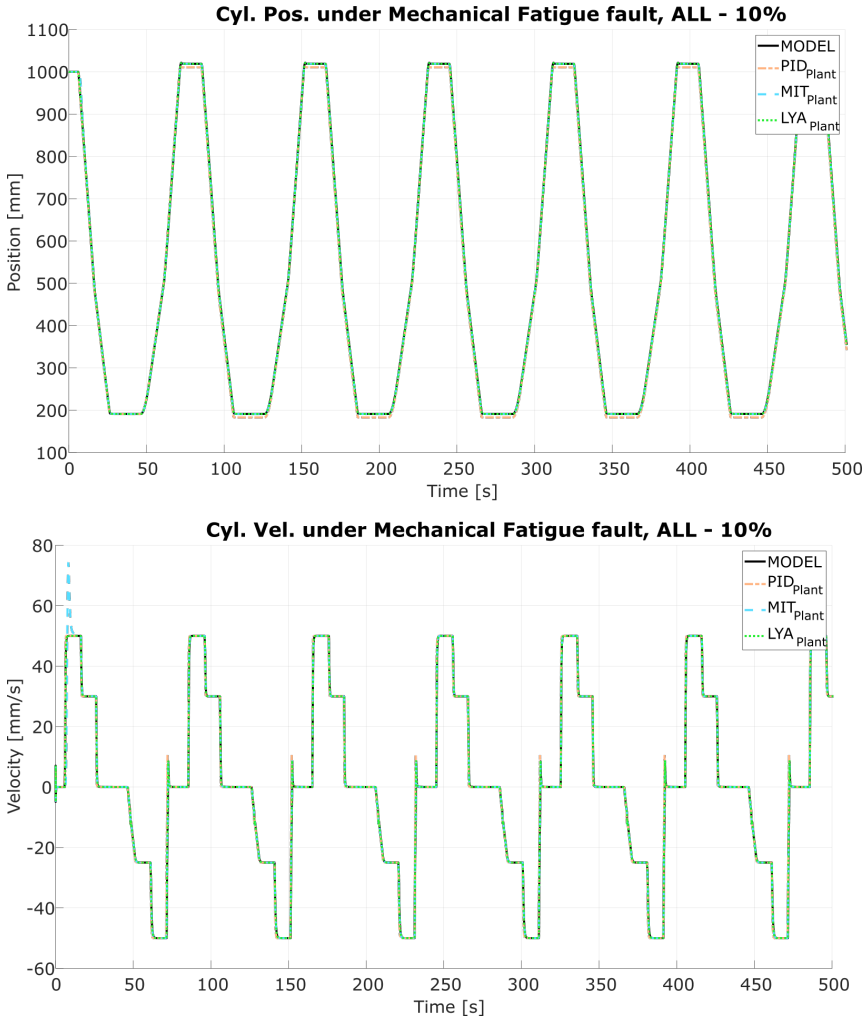


Figure 4.24: Hydraulic-Press Position and Velocity under *Mechanical Fatigue* fault for all the controllers, performance against a 10% fault.

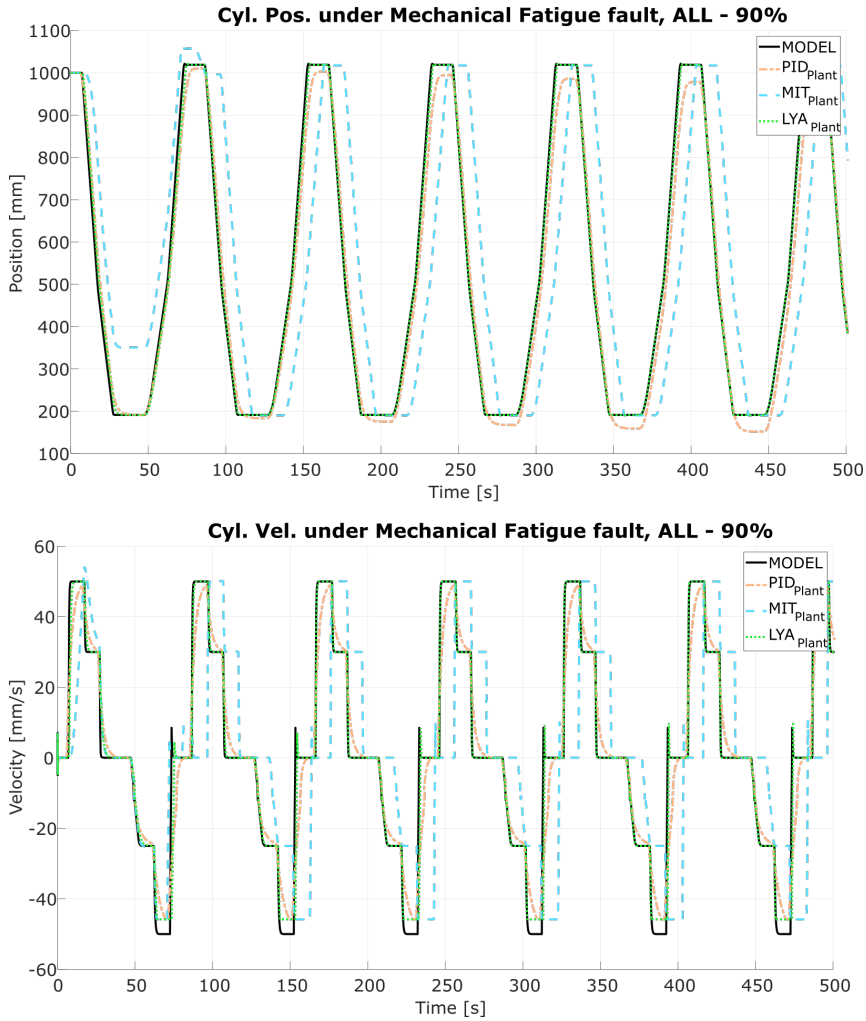


Figure 4.25: Hydraulic-Press Position and Velocity under *Mechanical Fatigue* fault for all the controllers, performance against a 90% fault.

4.2.1.2 Analytical Approach

The study continues analysing the maximum separation and the Root Mean Square Error (RMSE) between the optimal performance, represented by the model responses, and each control architecture. The measurements presented belong to the first ten iterations of the Hydraulic-Press cycle under both fault effects.

When the fault is under a 10%, the PID recovers an optimal performance, but the deviation remains identical on every cycle. MRACs also present a divergence, although they gradually minimise this error. The adaptive architecture lessens this deviation as the system continues performing the cycle. The subsequent results

corroborate the previous graphical study (see Section 4.2.1.1), providing analytical information about the maximum separation and RMSE in the first ten cycles after the fault emergence.

In the 90% fault experiment, MRAC controllers benefit across the first ten cycles is easily noticeable (see Fig. 4.26). While the PID maximum separation remains constant after the first reconfiguration attempt, the controllers based on MIT and Lyapunov rules reduce this error gradually. Also, results show the initial divergence between model and controller for MIT rule cases, where the system performance is below the optimal point.

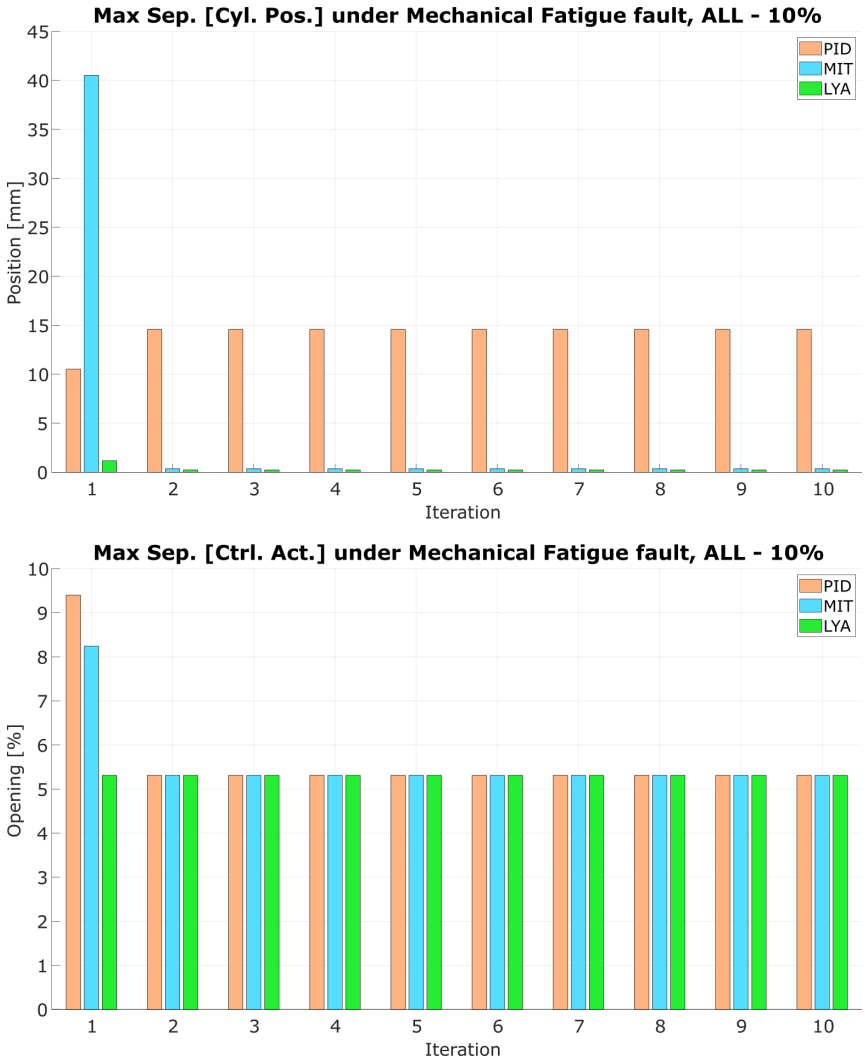


Figure 4.26: Maximum separation between real system and model in a *Mechanical Fatigue* fault for all the controllers, performance against a 10% fault.

Similarly, the Root Mean Square Error also presents a decreasing tendency (see Fig. 4.27). After the first cycle, controllers based on PID architectures are unable to reduce the separation between optimal behaviour and system response, maintaining a constant error across the subsequent cycles. MIT rule algorithms consume a significant amount of control action on the first cycle; however, after surpassing this initial divergence, they maintain a descending tendency. Lyapunov rule algorithms also have this tendency, presenting a better response than their MIT counterpart. They nullify the error even in the first cycle, adapting the signal in the subsequent ones until both signals converge.

Figure 4.27: Root Mean Square Error between real system and model in a *Mechanical Fatigue* fault for all the controllers, performance against a 10% fault.

These divergences increase when the fault spreads to 90% (see Fig. 4.28). The proportional valve barely opens, leading the PID controller to a suboptimal behaviour. The maximum separation in the adaptive algorithms presents the same tendency as the one described for the 10% fault in the MIT and Lyapunov studies, as they minimise the tracking error. Although they attain better results than the PID counterpart, the divergence is still considerable after the first ten cycles.

Figure 4.28: Maximum separation between real system and model in a *Mechanical Fatigue* fault for all the controllers, performance against a 90% fault.

The study continues presenting the RMSE, highlighting the fact that the control action consumed by the adaptive controllers is less similar to the original behaviour than the PID counterpart (see Fig. 4.29). Each iteration, Lyapunov controllers reduce this divergence, minimising the fault adverse effect over the system. Conventional controllers present a more erratic behaviour, closer to the instability as their control action presents a continuous maximum deviation across every cycle, but the cylinder has extreme position variations.

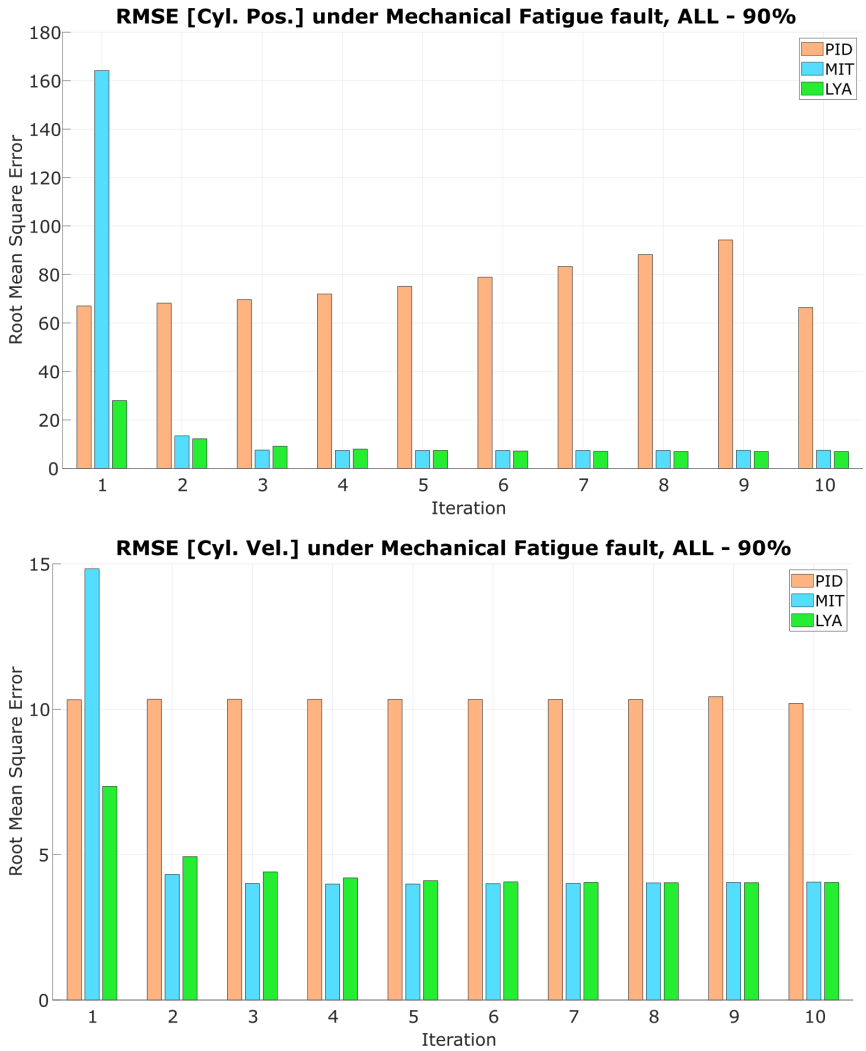


Figure 4.29: Root Mean Square Error between real system and model in a *Mechanical Fatigue* fault for all the controllers, performance against a 90% fault.

As anticipated, the separation between optimal behaviour and PID controller performance grows every cycle (see Fig. 4.30). After the ninth iteration, the cylinder position recovers a closer performance to the optimal behaviour, but this effect occurs due to a disturbance in the control action. The controller has still a degraded performance, who tends to destabilize the system on the subsequent cycles.

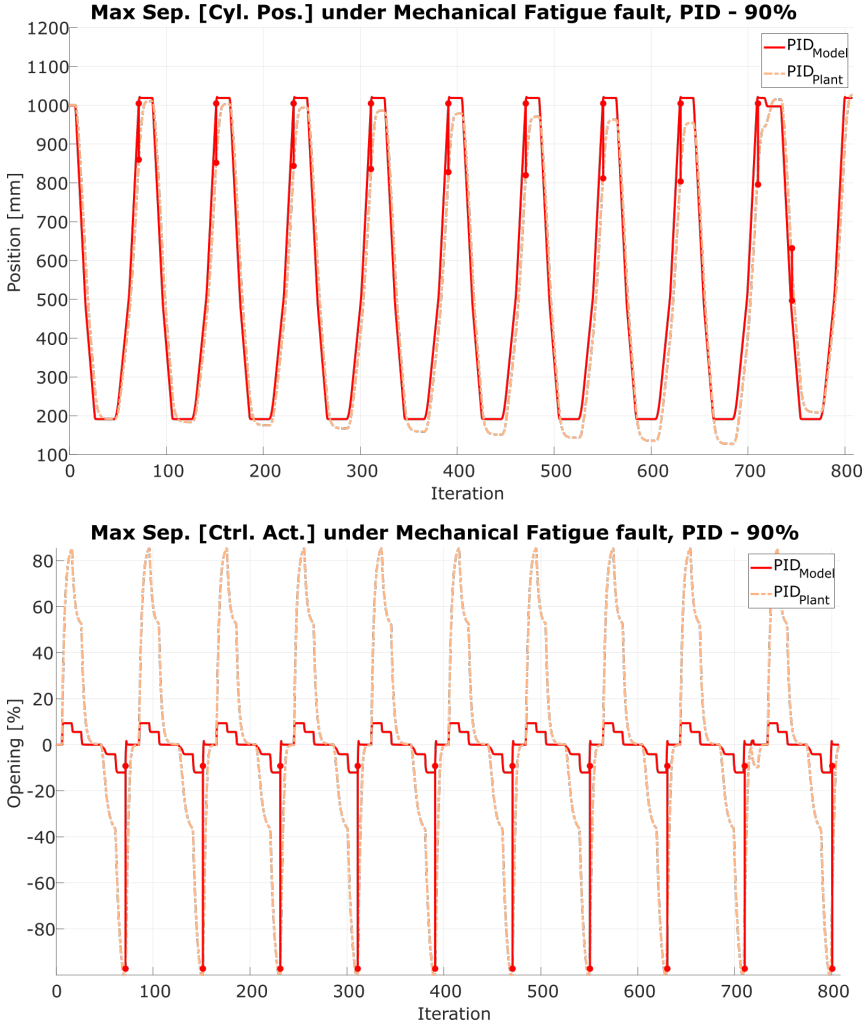


Figure 4.30: Maximum separation between real system and model in a *Mechanical Fatigue* fault for the PID controller, performance against a 90% fault.

The study continues presenting this unstable behaviour in the cylinder position and velocity through the RMSE (see Fig. 4.31). This property grows every iteration descending in the tenth cycle drastically due to the unexpected controller position. Although this alteration could be understood as an adaptation process, the RMSE in the cylinder velocity remains identical without improving its responses.

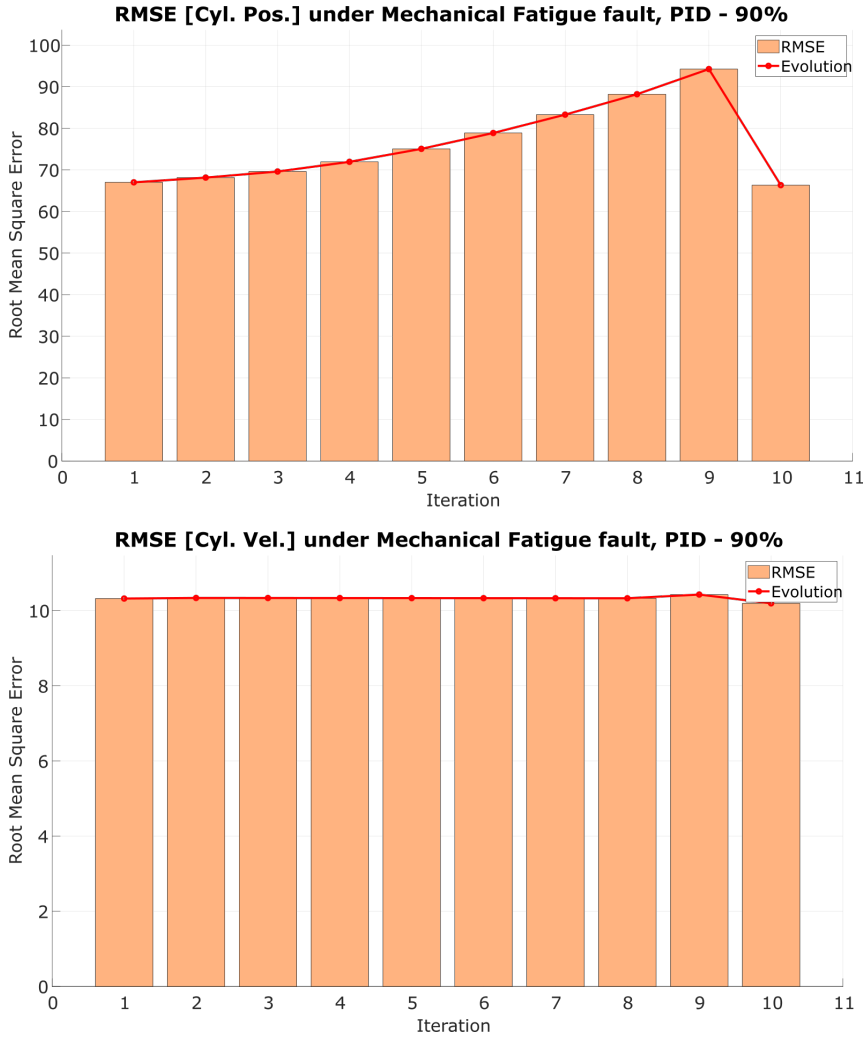


Figure 4.31: Root Mean Square Error between real system and model in a *Mechanical Fatigue* fault for the PID controller, performance against a 90% fault.

Controllers based on the MIT rule perform an unproductive first cycle after the fault emergence (see Fig. 4.32). Nonetheless, after the adaptive gains initiate the adaptation process, the controller behaviour maintains a similar tendency as the optimal performance. The position error is minimum across the cycle; however, there is a divergence on the control action.

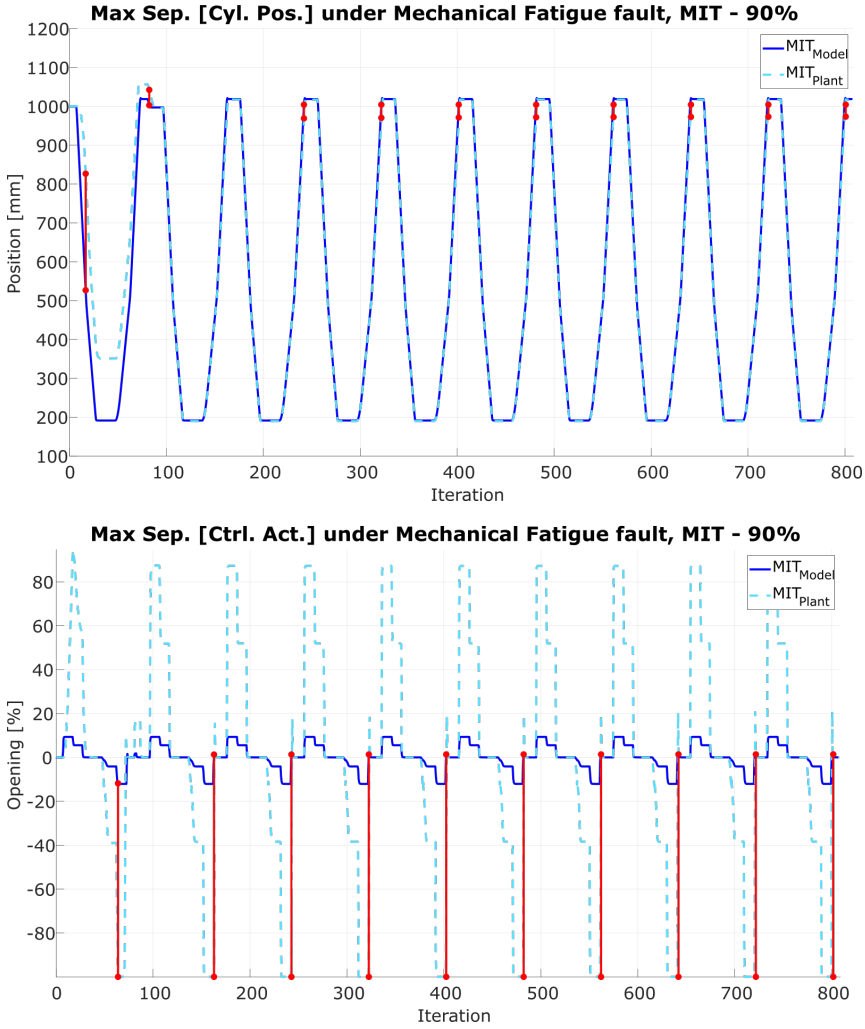


Figure 4.32: Maximum separation between real system and model in a *Mechanical Fatigue* fault for the MIT rule, performance against a 90% fault.

RMSE study corroborates the early assumptions, as this property reduces the divergence continuously in the cylinder position and velocity (see Fig. 4.33). This tendency demonstrates the adaptation process, but it is not conclusive regarding the maximum divergence in the controller response.

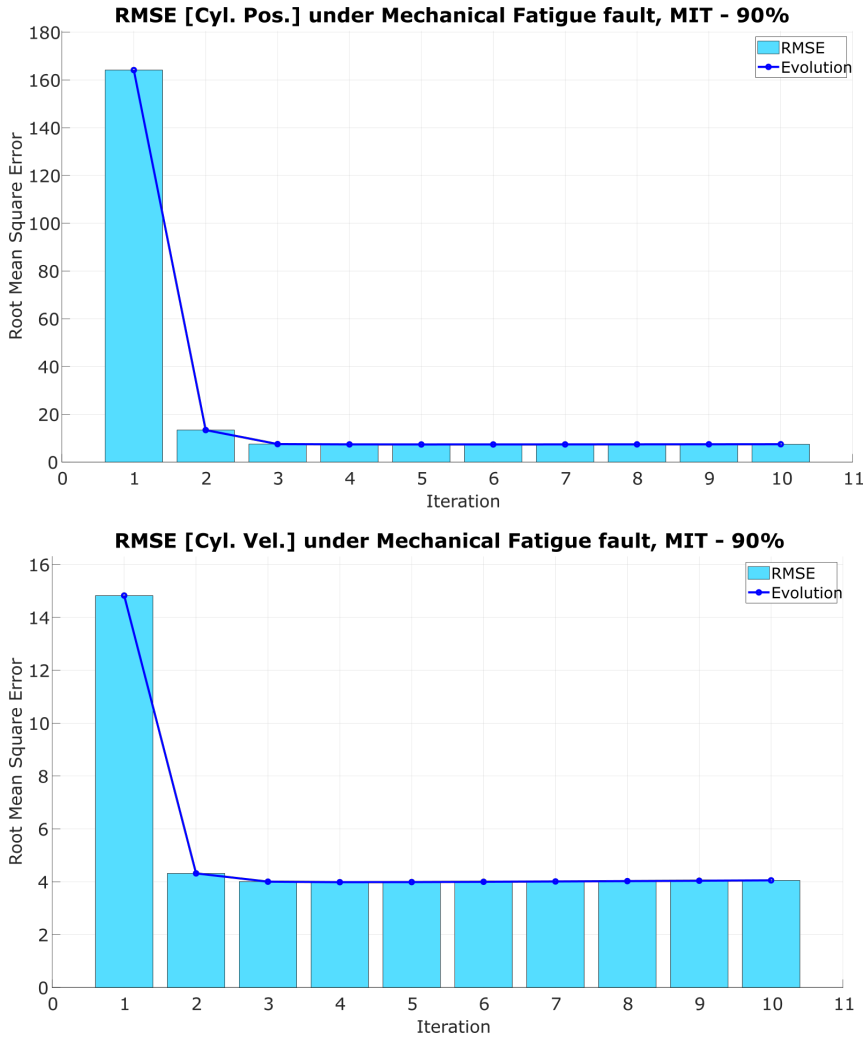


Figure 4.33: Root Mean Square Error between real system and model in a *Mechanical Fatigue* fault for the MIT rule, performance against a 90% fault.

Lyapunov rule controllers perform a smoother adaptation process than their MIT counterpart; nonetheless, they present a similar effect (see Fig. 4.34). The maximum separation between the cylinder position in optimal and fault performance is minimum, but the control action presents a single position where this value is extremely high.

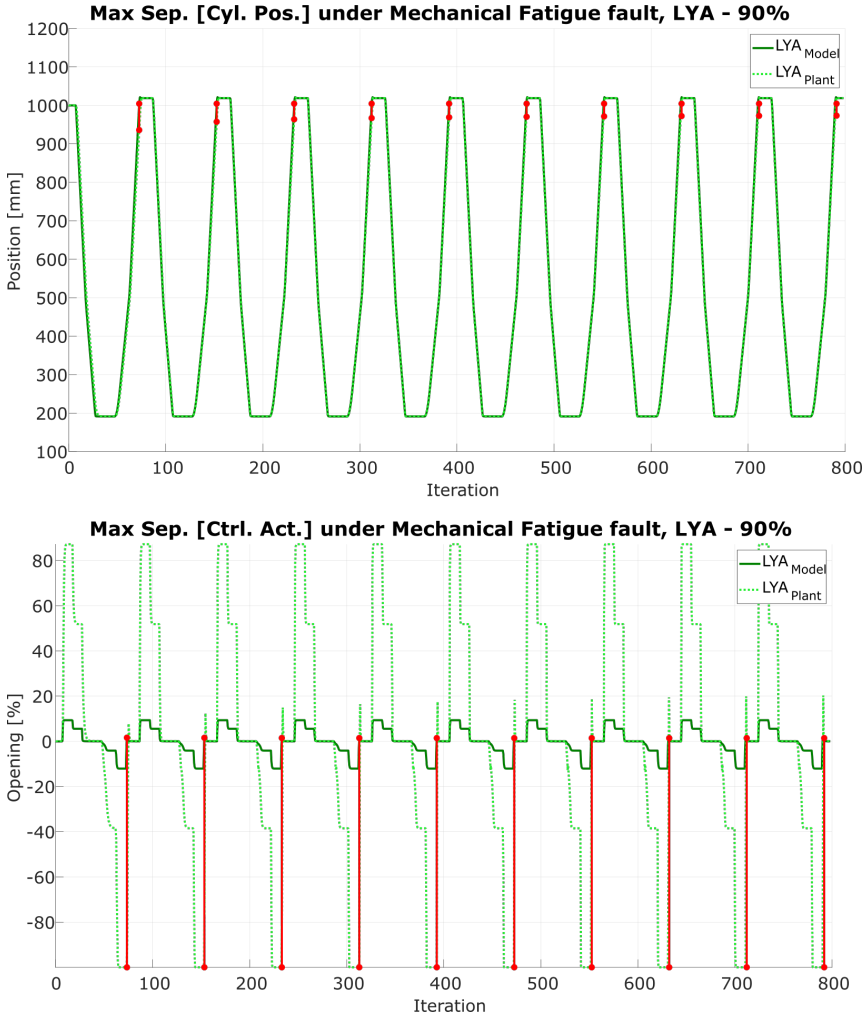


Figure 4.34: Maximum separation between real system and model in a *Mechanical Fatigue* fault for the Lyapunov rule, performance against a 90% fault.

The RMSE study corroborates these assumptions, as both position and controller, present a continuous adaptation process (see Fig. 4.35). After examining more carefully the maximum separation in the control action, the source of this divergence has been encountered. Even though the adaptive algorithm improves the controller response, they are unable to nullify the error as the fault has altered the proportional valve saturation limits.

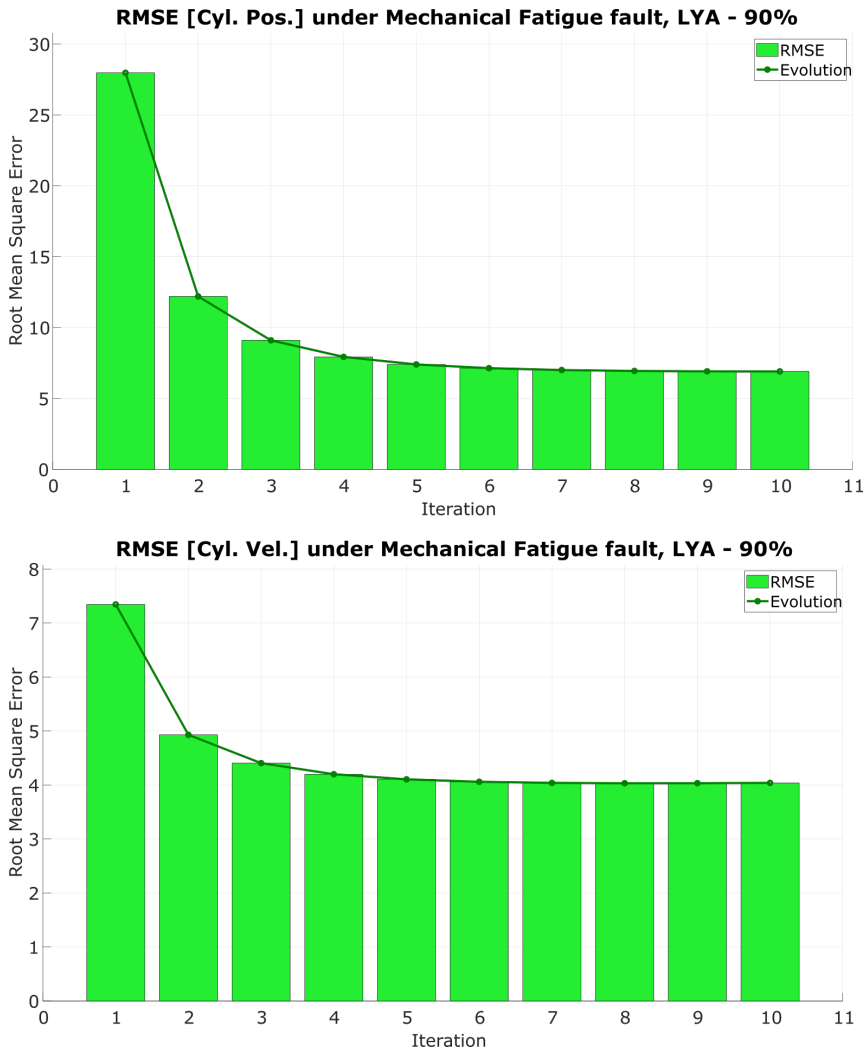


Figure 4.35: Root Mean Square Error between real system and model in a *Mechanical Fatigue* fault for the Lyapunov rule, performance against a 90% fault.

When the fault increases to a 90%, the PID architecture cease providing service, and the system becomes unstable. Against this degraded behaviour, MRACs

ensure stability. Their maximum separation and RMSE have increased regarding the 10% study, but the controller continues reducing the fault effect gradually. The analytical comparison corroborates the assumptions previously stated (see Table 4.1), considering that despite the improvements obtained, the fault still alters the proportional valve saturation limits, reducing the adaptation mechanism potential.

$FAULT$		$MechanicalFatigue$					
$VALUE$		10%					
$CONTROL$		PID		MIT		LYA	
$CYCLE$		1^{st}	10^{th}	1^{st}	10^{th}	1^{st}	10^{th}
CP	MAX	10.54	14.58	40.52	0.35	1.17	0.24
	$RMSE$	4.81	9.04	5.28	0.11	0.55	0.07
CA	MAX	26.34	21.09	44.05	1.49	12.68	1.49
	$RMSE$	2.52	1.81	5.00	0.09	0.42	0.09
$VALUE$		90%					
$CONTROL$		PID		MIT		LYA	
$CYCLE$		1^{st}	10^{th}	1^{st}	10^{th}	1^{st}	10^{th}
CP	MAX	144.83	135.11	300.01	30.85	68.57	31.08
	$RMSE$	67.03	66.35	164.14	7.45	27.97	6.91
CA	MAX	45.33	45.33	55.70	42.50	54.32	42.64
	$RMSE$	10.33	10.20	14.83	4.05	7.34	4.04

Table 4.1: Maximum separation and RMSE between the desired signal and the AC for *Mechanical Fatigue*.

As a recapitulation, PID controllers are incapable of maintaining the system stable when faults emerge, having a constant error between optimal and current performance. The adaptive controller presents a feasible solution to deal with faults where Lyapunov controllers present a slightly better response on the first iterations than their MIT counterpart; however, both have a faster adaptation rate to reduce the breach between optimal and current performance earlier. Across the following studies (see Section 4.2.4), the adaptive reference model sought to reduce the divergence between optimal and current responses, bridging the latest techniques.

4.2.2 2^{nd} Experiment: Plant Fault

Hydraulic-actuators have two isolated chambers behaving as the accumulators, called as primary and annular, and a rod that describes the movement. Propor-

tional valves control the rod displacement regulating the amount of fluid reaching each chamber. They managed the cylinder attending to a mono-directional or a bi-directional configuration, commanding one or both chambers, respectively.

Proportional valves only ensure proper control when both chambers are isolated, as connect them would equalise flow rate and pressure neglecting the rod displacement. Despite the efforts of manufacturers hermetically sealing the chambers, the movable parts prevent complete isolation. The control algorithm considers these small overtures, regulating the cylinder accordingly.

While these internal leaks between chambers are petite, PID controllers can manage them and maintain the cylinder controlled. However, the rod regularly contacts chamber walls, eroding them in the process and increasing the pores radius. These actions support the internal fluid exchange, spreading the fault harming effect and neglecting PID controllers regulation.

On contraposition, MRACs offer a viable alternative to regain stability, as they understand these faults as variations in the plant dynamics. They alter the controller signal accordingly to the current cylinder status, trying to minimise the harming effect. Similarly to the actuator faults, two experiments have been stated to test controllers performance, the first one with the fault at 10% and the second one at 90%.

4.2.2.1 Hydraulic-Press Overview

Operators quickly detect this fault, as it alters the piston rod displacement directly. Similarly, when controllers notice the changes in the cylinder behaviour, they alter their signal accordingly. When the fault harming effect has barely degraded the performance, controllers recover the optimal situation. Nonetheless, this fault modifies the pressure and flow rate in the chambers, the variables regulated by the control algorithm. After a certain point, the controller ends providing service as both chambers are being filled simultaneously.

The study focuses on faults below this range, that is to say, internal leaks in the cylinder chamber through small pores or overtures. Faults at 10% range represent a degradation in the joints isolating both chambers. There is barely any fluid transmission, deviating briefly the flow rate and pressure from the optimal values. When the fault spread at 90% degradation, these overtures enlarge their diameter decanting the oil through chambers.

The experiments resemble the mechanical fatigue fault (see Section 4.2.1), studying the cylinder position across several cycles and the controller performance through various algorithms. The first architecture belongs to the current PID controller instaled on the machine, while the second and third architectures belong to MRACs based on MIT or Lyapunov rules, respectively. Each control algorithm has been submitted to the fault, initially at 10% degradation, and, afterwards, at 90%.

While the fault remains at low levels, 10%, the PID controller maintains the system stable deviating the performance minimally from the original behaviour (see Fig. 4.36). The cylinder velocity also has values closer to the desired performance, ensuring a constant production.

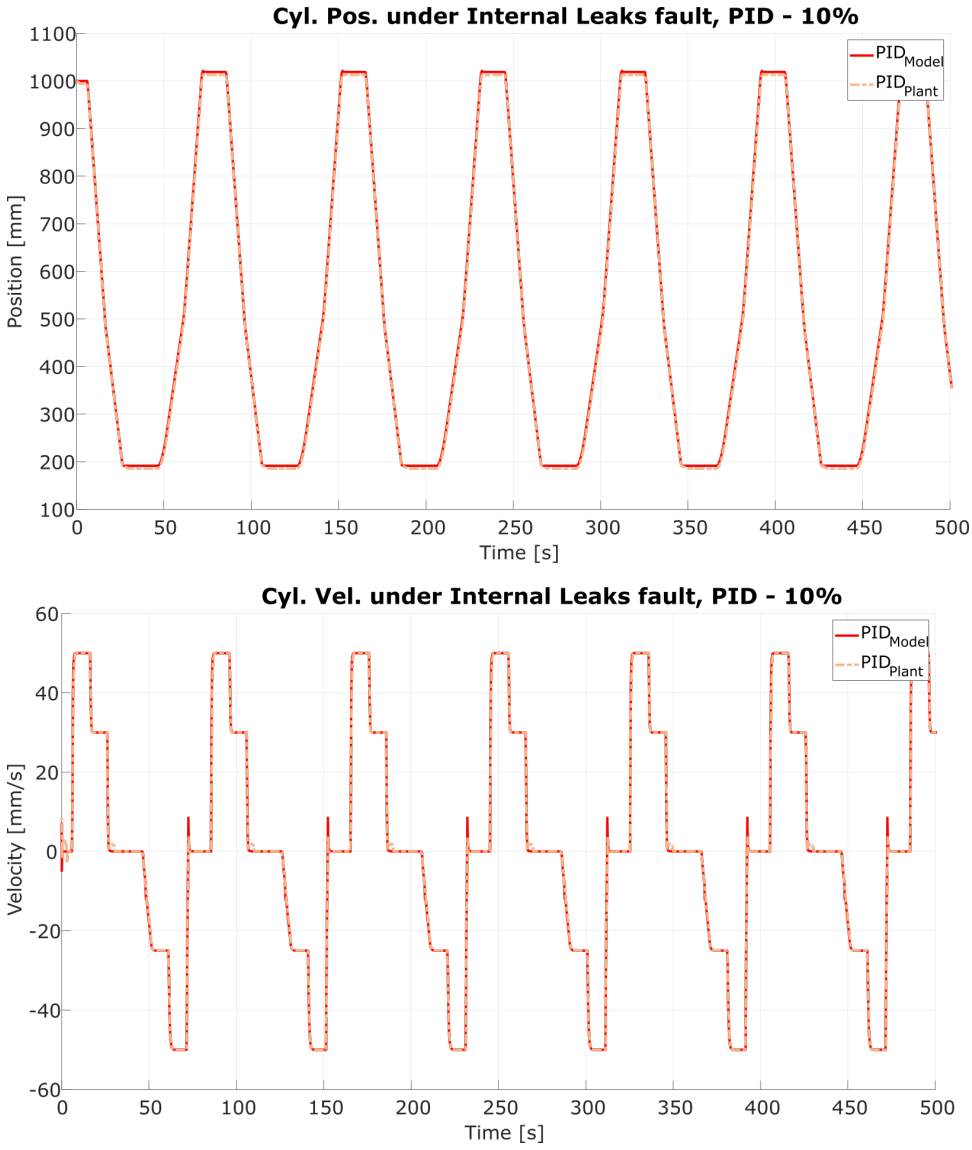


Figure 4.36: Hydraulic-Press Position and Velocity under *Internal Leaks* fault for the PID controller, performance against a 10% fault.

The adaptive controller based on MIT rules ensures proper performance, adapting the signal accordingly to the latest dynamics (see Fig. 4.37). Although the position is stable across all the cycles, the cylinder velocity suffers a slight imbalance during the first iteration until the adaptive gains attain stability.

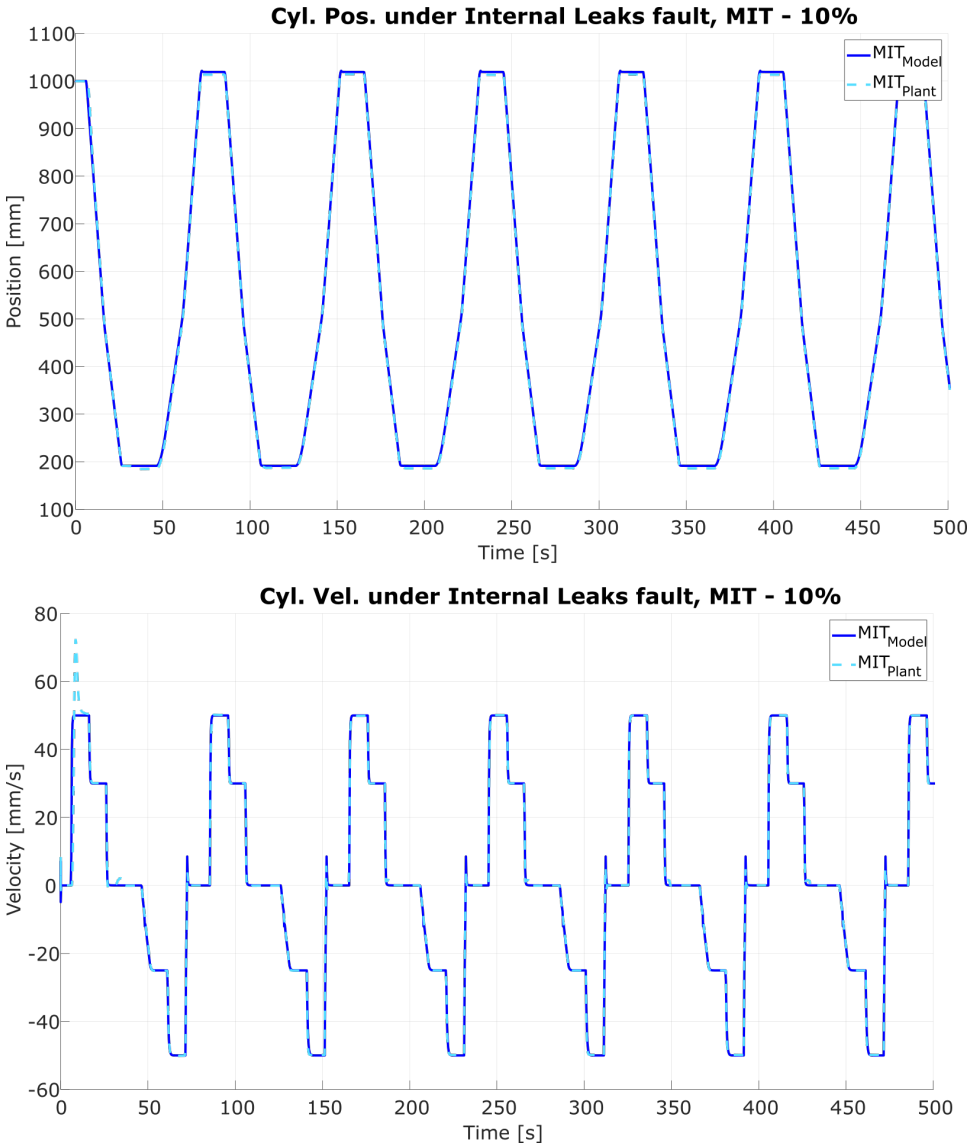


Figure 4.37: Hydraulic-Press Position and Velocity under *Internal Leaks* fault for the MIT rule controller, performance against a 10% fault.

Lyapunov rules present an identical behaviour as their MIT counterpart, with the exception that they maintain the cycle stable during the first iterations (see Fig. 4.38). Their smoothness adaptation process searching for the global minimum allows them to maintain the system stable across every cycle iteration.

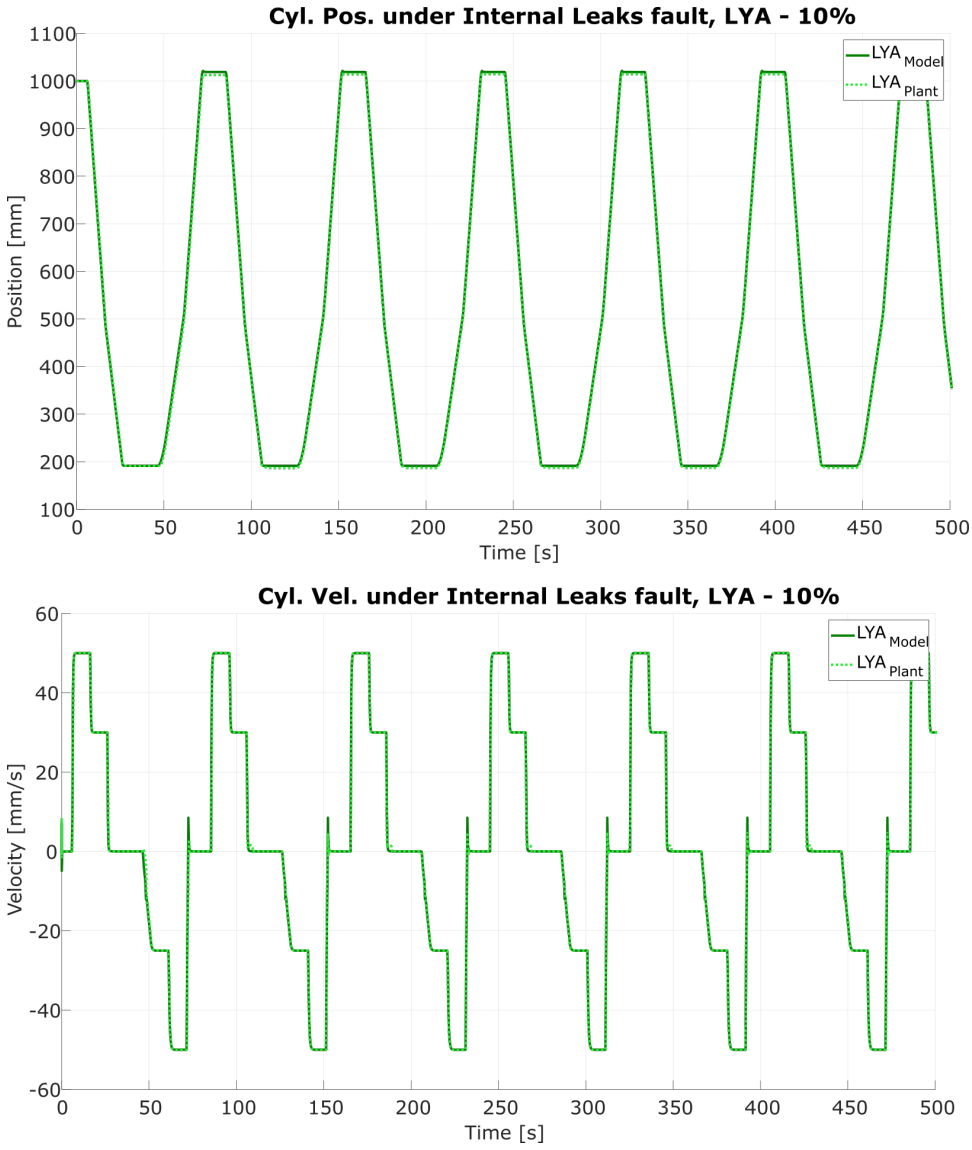


Figure 4.38: Hydraulic-Press Position and Velocity under *Internal Leaks* fault for the Lyapunov rule controller, performance against a 10% fault.

When the fault spreads at 90%, the cylinder behaviour suffers a dissonance on the cylinder position (see Fig. 4.39). Despite the control algorithm attempts to perform the cycle, the internal fluid exchange between chambers neglects the rod displacement. This effect creates a malfunction on the position that remains unnoticed by the controller, as the velocity has only suffered a brief degradation.

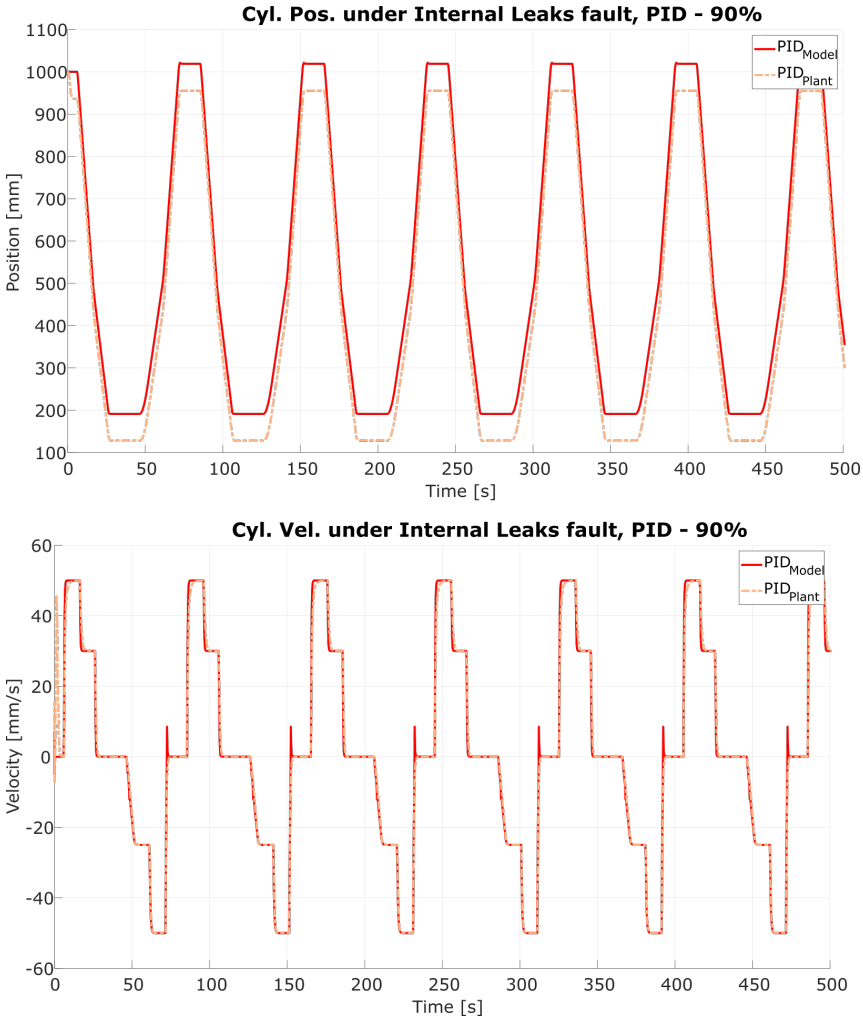


Figure 4.39: Hydraulic-Press Position and Velocity under *Internal Leaks* fault for the PID controller, performance against a 90% fault.

MIT rule controller presents a similar effect. The control action resembles the original performance, but the cylinder position diverges from the model response (see Fig. 4.40). Even though there are dissonances in the signals, they are better than their PID counterpart as adaptive algorithms minimise the error between plant current performance and model responses.

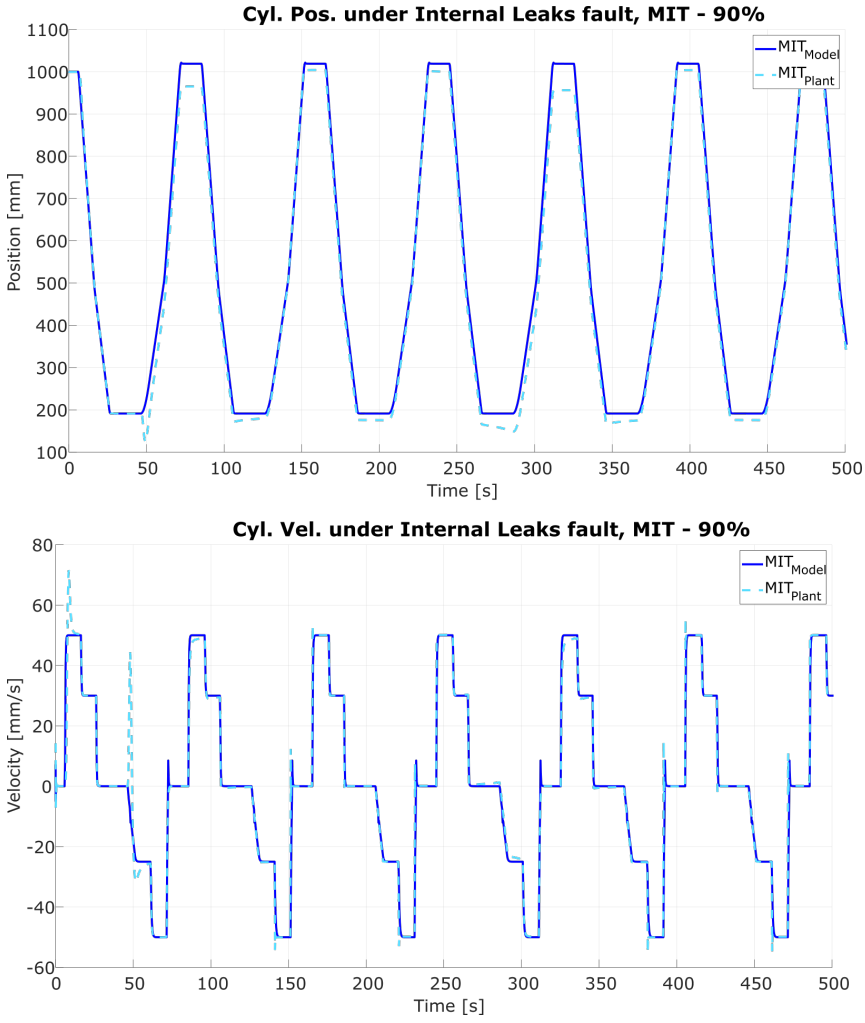


Figure 4.40: Hydraulic-Press Position and Velocity under *Internal Leaks* fault for the MIT rule controller, performance against a 90% fault.

The Lyapunov simulation presents a similar behaviour, but the cylinder position signal is closer to the optimal performance than their predecessors (see Fig. 4.41). They reduce the fault effect even more than the MIT rule as the algorithm search for the global minimum instead of local ones.

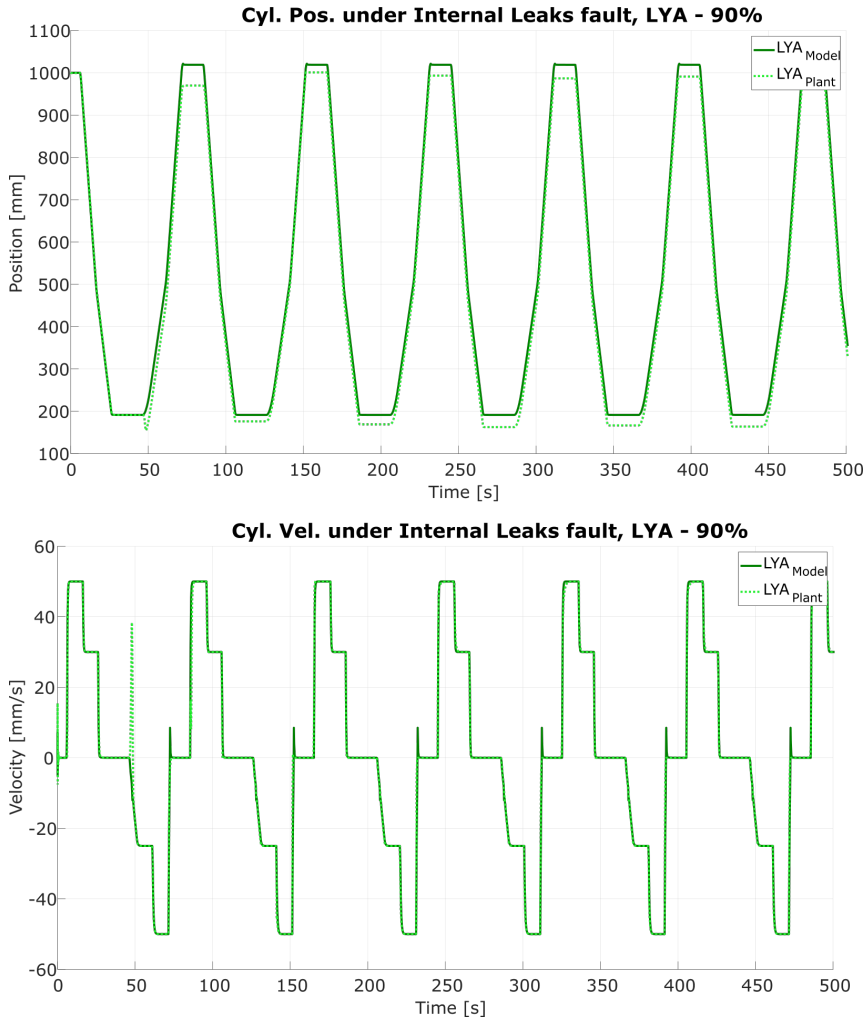


Figure 4.41: Hydraulic-Press Position and Velocity under *Internal Leaks* fault for the Lyapunov rule controller, performance against a 90% fault.

While the fault remains at moderate levels, for instance, at 10%, the controller maintains a similar position as the optimal performance independently of the control architecture (see Fig. 4.42). When the fault spreads at 90%, the cylinder position diverges vastly from the optimal performance. MRACs controllers have closer responses than the PID counterpart; nonetheless, they are still below the nominal range (see Fig. 4.43).

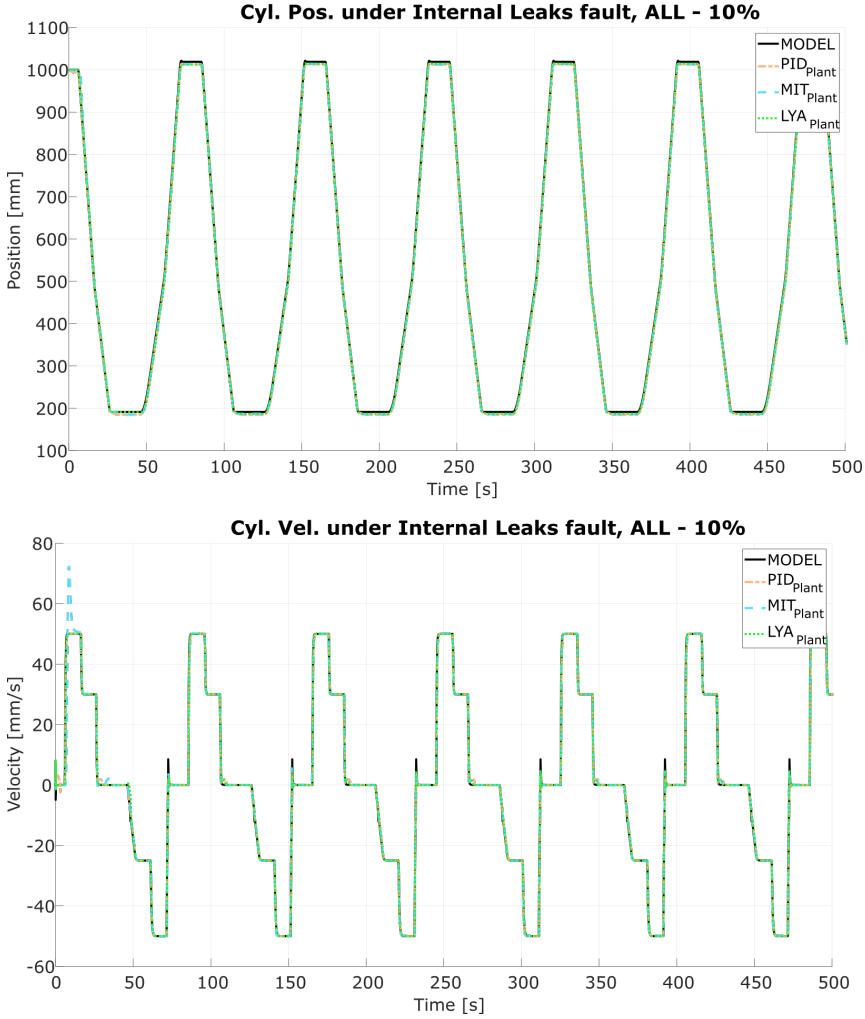


Figure 4.42: Hydraulic-Press Position and Velocity under *Internal Leaks* fault for all the controllers, performance against a 10% fault.

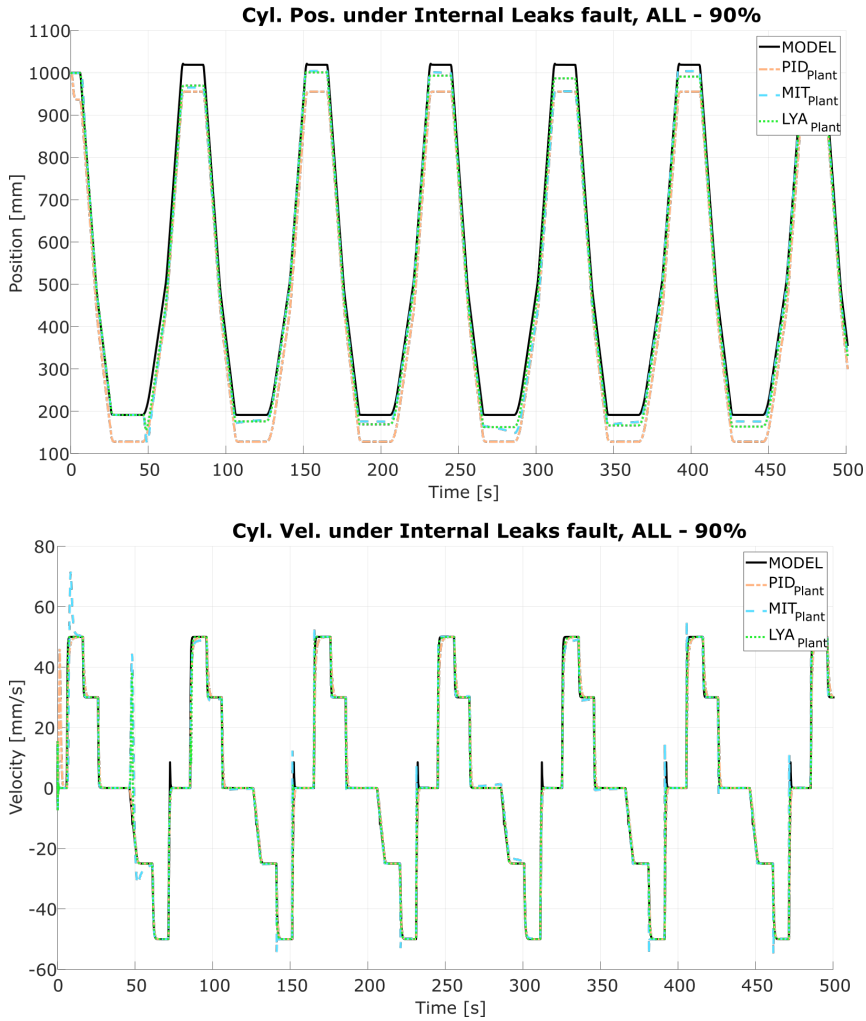


Figure 4.43: Hydraulic-Press Position and Velocity under *Internal Leaks* fault for all the controllers, performance against a 90% fault.

After the fault, Adaptive Controllers maintain the system stable, although their performance is below the nominal range, they are still able to produce as if they are in the optimal region. Nonetheless, this drawback tends to equalise pressure and flow rate in the cylinder chambers, decreasing the controller effectiveness. Even though the control algorithm adapts the signal, the fault neglects the piston rod movement.

4.2.2.2 Analytical Approach

The research continues comparing each control architecture against the optimal performance. This correlation estimates the maximum separation and Root Mean

Square Error when the fault has altered the nominal behaviour around 10% or 90%. The experiment presents ten samples, belonging to the first ten HP cycle iterations after the fault emergence.

When the fault has degraded a 10% the movement, the controllers remain inside the region of required performance, varying their values slightly (see Fig. 4.44). PID controllers perform a similar response regardless of the cycle; however, MIT and Lyapunov architectures reduce the error across each iteration. Highlight the first cycle of MIT rule, as the maximum separation reaches higher values than their Lyapunov counterpart.

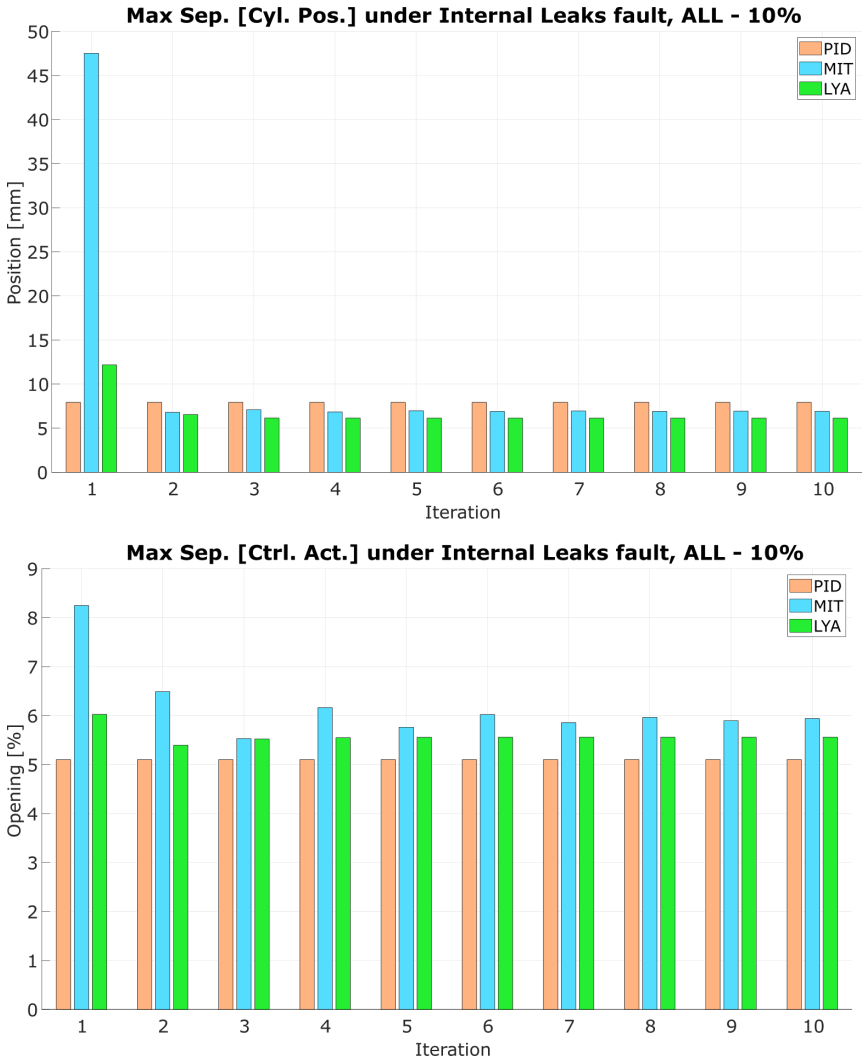


Figure 4.44: Maximum separation between real system and model in a *Internal Leaks* fault for all the controllers, performance against a 10% fault.

The RMSE also reflects this low dispersion between controllers and optimal performance (see Fig. 4.45). While the PID maintains a constant error, MIT and Lyapunov rules minimise the divergence for the cylinder position and velocity. The latest adaptive controller presents a better response against this fault, accumulating lower error.

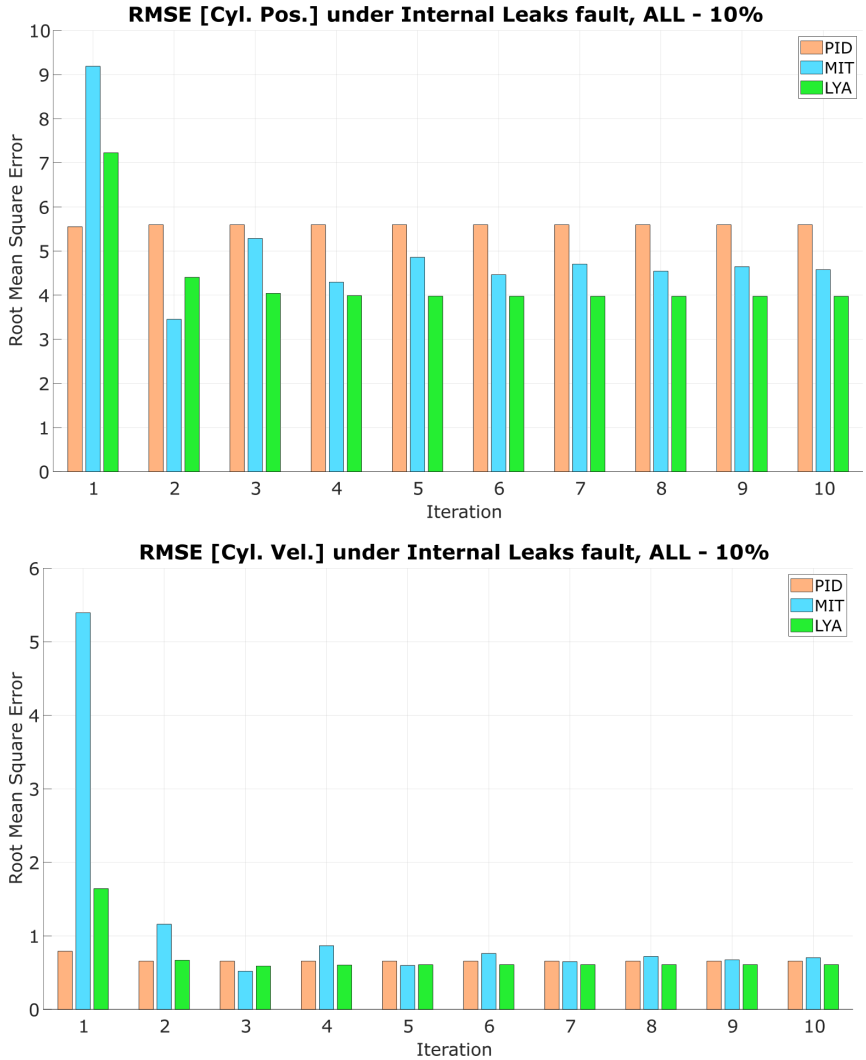


Figure 4.45: Root Mean Square Error between real system and model in a *Internal Leaks* fault for all the controllers, performance against a 10% fault.

When the fault degradation increases at 90%, controllers behaviour become compromised, obtaining a degraded performance (see Fig. 4.46). PID algorithms are unable to recover the nominal behaviour, maintaining a constant maximum error across each iteration. MIT rule controllers attempt the adaptation mechanism, but they fall into a local minimum, destabilising the system. MRACs based on Lyapunov rules sought the global minimum, obtaining after stabilising the signal a 50% max separation reduction in comparison with the PID alternative.

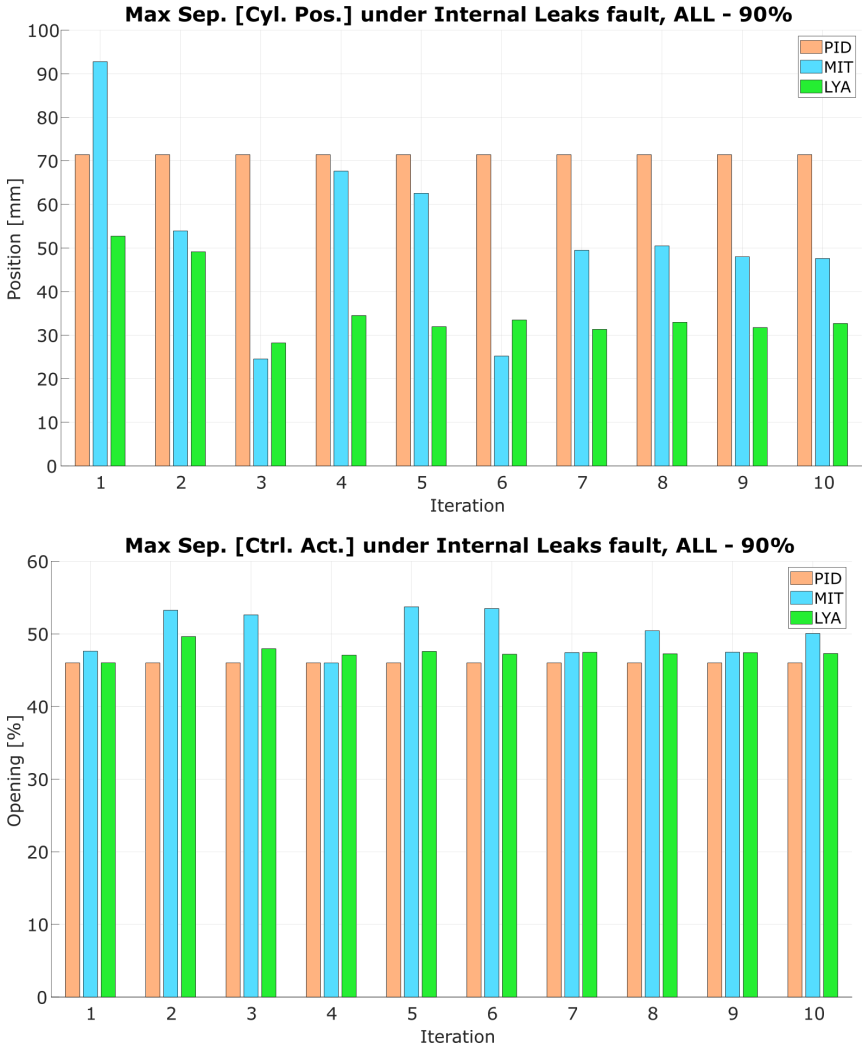


Figure 4.46: Maximum separation between real system and model in a *Internal Leaks* fault for all the controllers, performance against a 90% fault.

RMSE graphs corroborate the early assumptions, presenting the cylinder position and velocity dispersion (see Fig. 4.47). PID algorithms are stable, but their error neglects to maintain the production under the region of desired performance. MIT rule algorithms present a destabilized system, suffering high discrepancies between iterations. Lyapunov controllers minimise the adverse effect, recovering the production partially.

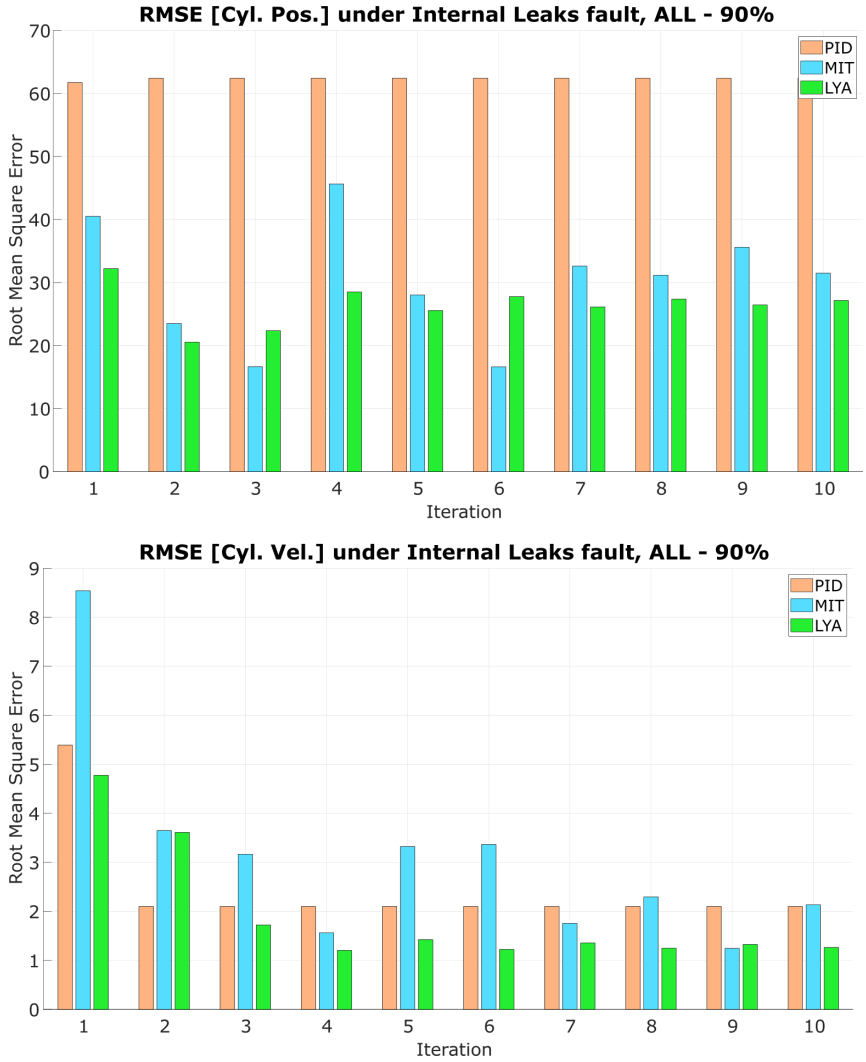


Figure 4.47: Root Mean Square Error between real system and model in a *Internal Leaks* fault for all the controllers, performance against a 90% fault.

As previously stated, the cylinder position executes the cycle below the optimal performance (see Fig. 4.48). The PID controller attempts to reproduce the first cycle, but the fault has displaced the opening into the negative plane, reducing its effectiveness.

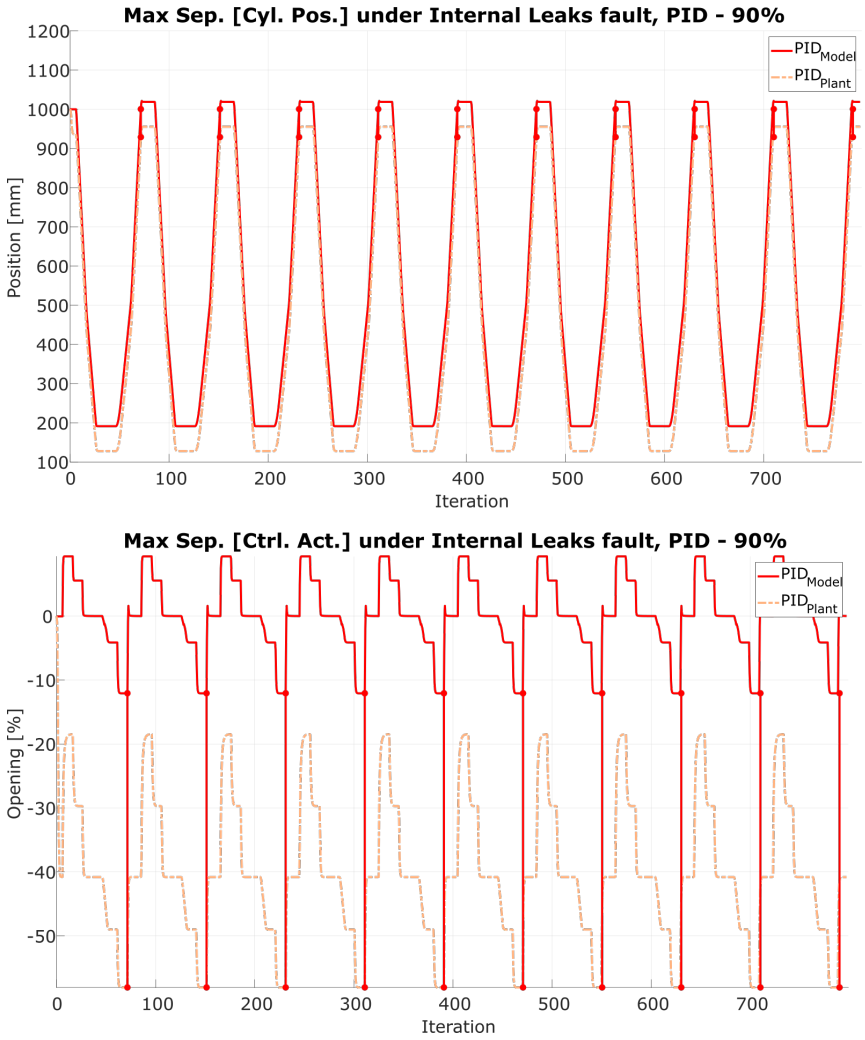


Figure 4.48: Maximum separation between real system and model in a *Internal Leaks* fault for the PID controller, performance against a 90% fault.

This divergence is also present in the Root Mean Square Error, as the cylinder position has a similar failure across all the iterations (see Fig. 4.49). The velocity value decreases considerably after the first cycle, but their rates are still significant to alter the cylinder movement. Although these architectures recovers the machine from soft faults, they are inefficient when it spreads.

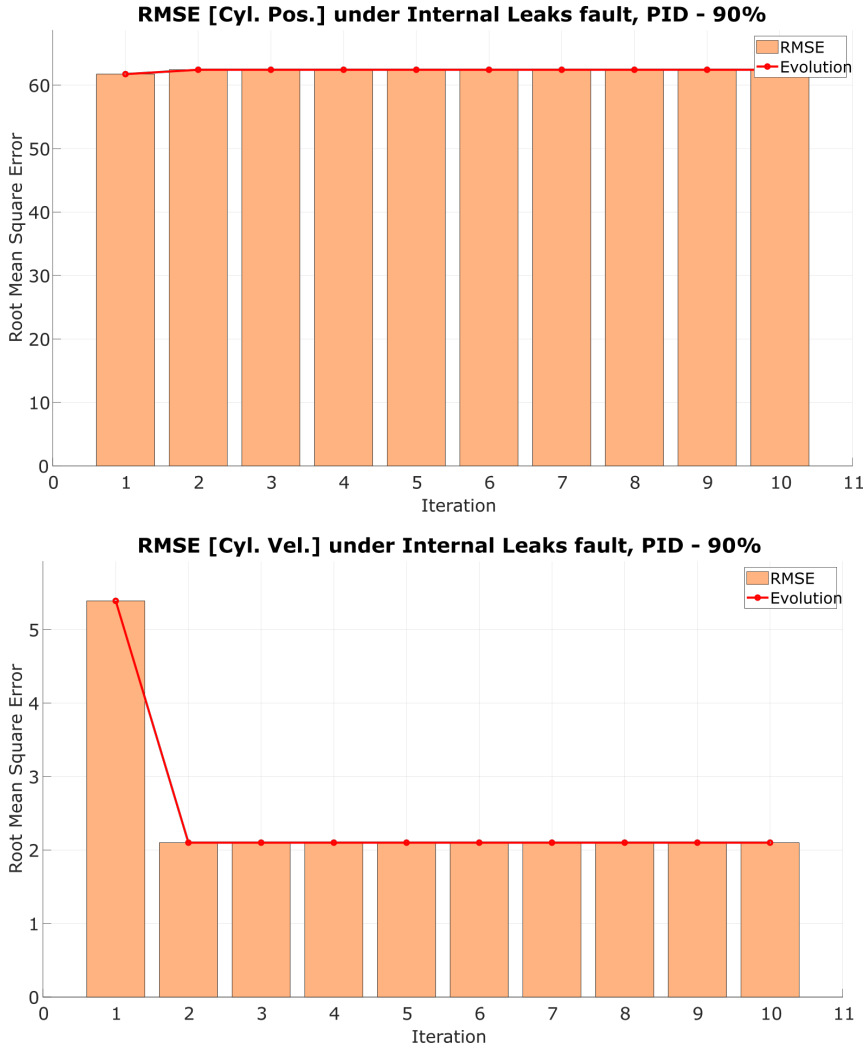


Figure 4.49: Root Mean Square Error between real system and model in a *Internal Leaks* fault for the PID controller, performance against a 90% fault.

MIT controllers present a more unstable behaviour, as the maximum separation in the cylinder position varies each iteration (see Fig. 4.50). The valve opening has also displaced into the negative plane. The controller fluctuates their values attempting to stabilise the signal, but the algorithm reaches local minimums.

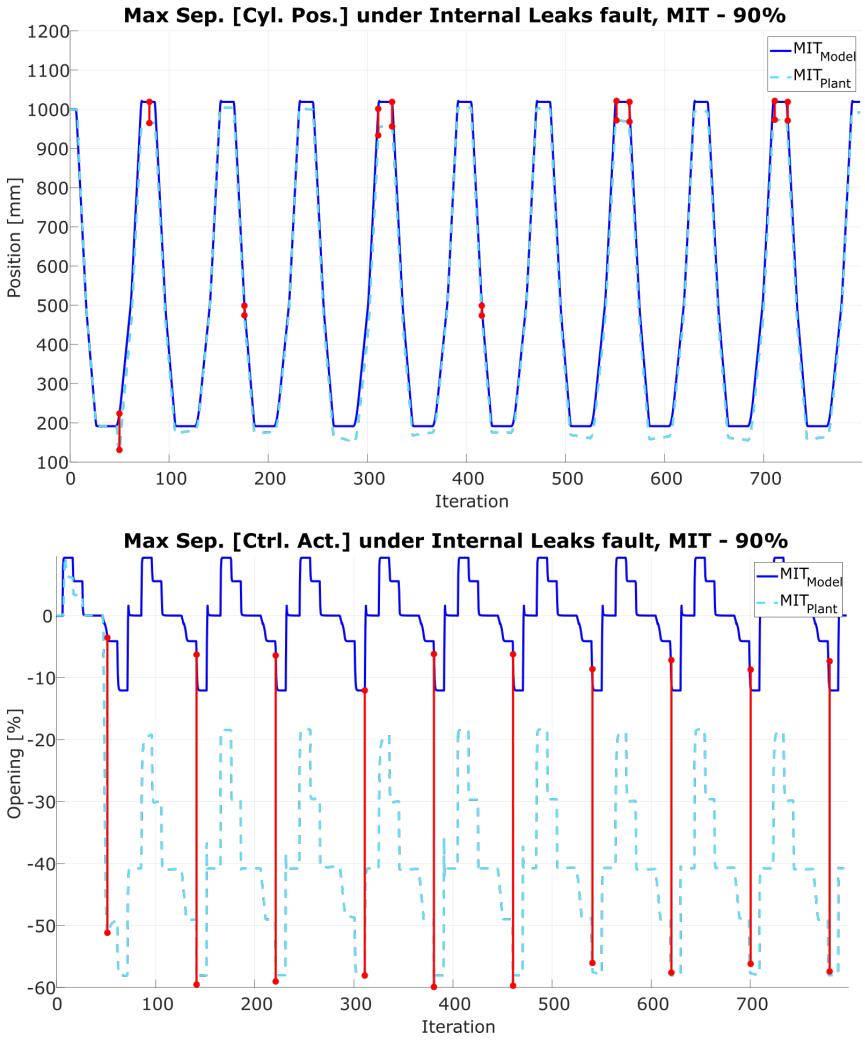


Figure 4.50: Maximum separation between real system and model in a *Internal Leaks* fault for the MIT controller, performance against a 90% fault.

The effect previously described is appreciated through the RMSE (see Fig. 4.51). The controller adapts the signal reducing the error accumulated during the first three iterations. On the fourth iteration, the divergence increases to their maximum value, initiating another recovery process, truncated on the seventh iteration.

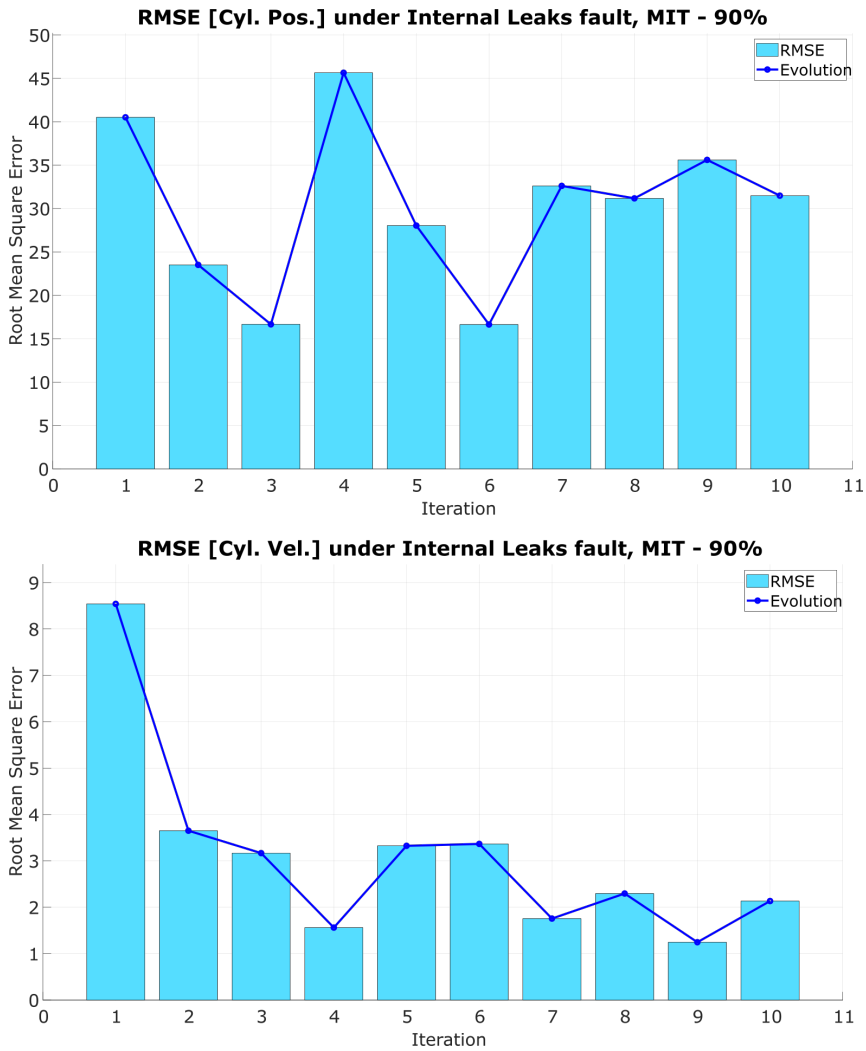


Figure 4.51: Root Mean Square Error between real system and model in a *Internal Leaks* fault for the MIT controller, performance against a 90% fault.

Lyapunov controllers also have a constant discrepancy between model and plant responses; nonetheless, this error has a lower value than the previous cases (see Fig. 4.52). The control action has softer values than their previous counterparts; despite the fault only allows open the valve in the negative plane.

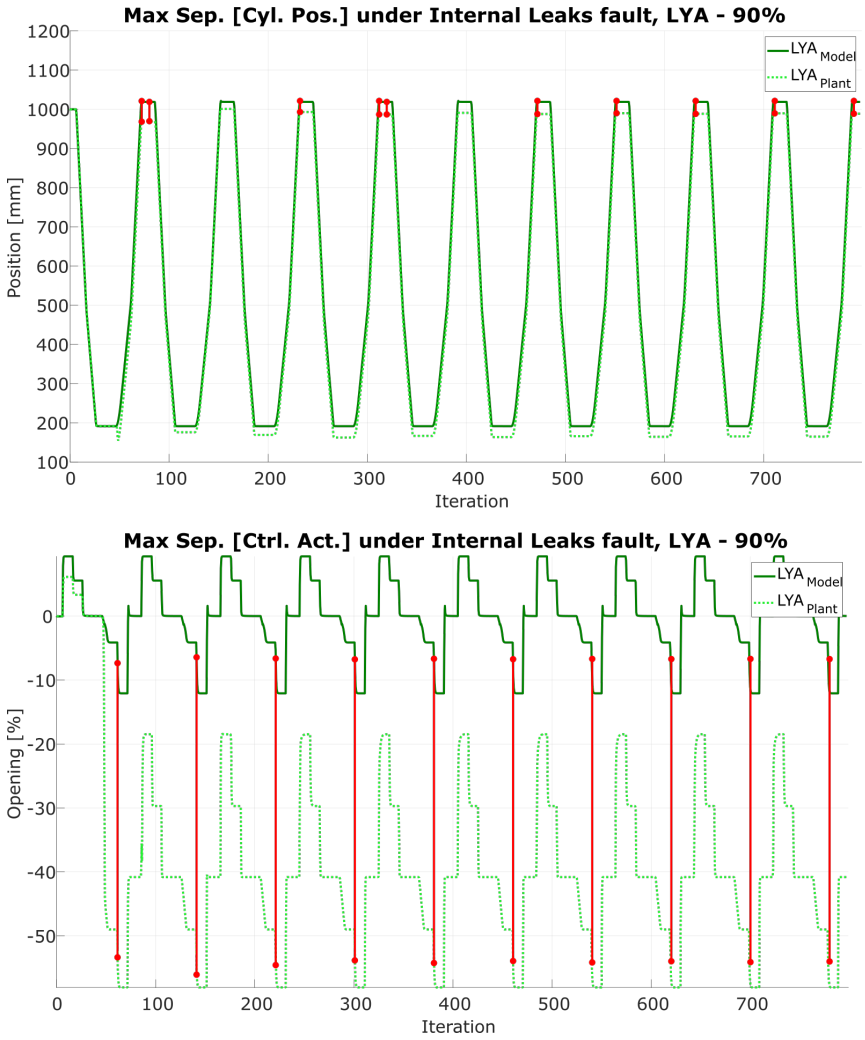


Figure 4.52: Maximum separation between real system and model in a *Internal Leaks* fault for the Lyapunov controller, performance against a 90% fault.

Although the adaptation process minimises the error, the system still suffers from punctual deviances (see Fig. 4.53). They are observable in the cylinder position graph, as the system increases the RMSE. In a contrary situation, the cylinder velocity graph presents descent behaviour.

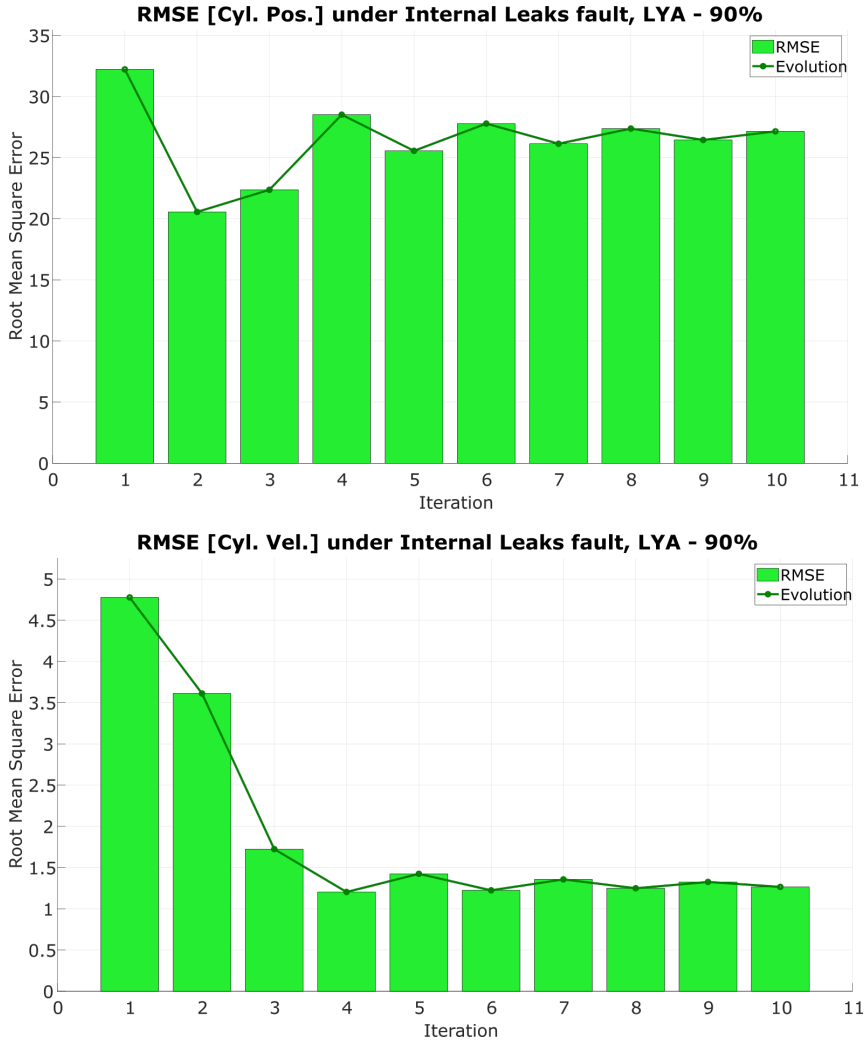


Figure 4.53: Root Mean Square Error between real system and model in a *Internal Leaks* fault for the Lyapunov controller, performance against a 90% fault.

Adaptive Controllers based on Lyapunov rule have a better performance than PIDs; however, they only reduce the fault effect, as there are still some disturbances on the signal. MIT rule algorithms reach local minimums, progressing towards unstable systems without encountering a stable state (see Table 4.2).

This fault generates instability on the system as it neglects actuators com-

$FAULT$		$InternalLeaks$					
$VALUE$		10%					
$CONTROL$		PID		MIT		LYA	
$CYCLE$		1^{st}	10^{th}	1^{st}	10^{th}	1^{st}	10^{th}
CP	MAX	7.92	7.92	47.52	6.91	12.17	6.13
	$RMSE$	5.55	5.60	9.19	4.58	7.23	3.97
CV	MAX	5.10	5.10	8.25	5.94	6.03	5.56
	$RMSE$	0.79	0.66	5.40	0.70	1.64	0.61
$VALUE$		90%					
$CONTROL$		PID		MIT		LYA	
$CYCLE$		1^{st}	10^{th}	1^{st}	10^{th}	1^{st}	10^{th}
CP	MAX	71.43	71.43	92.75	47.59	52.72	32.67
	$RMSE$	61.74	62.42	40.51	31.48	32.22	27.15
CV	MAX	46.00	46.00	47.62	50.06	46.01	47.30
	$RMSE$	5.39	2.10	8.54	2.13	4.78	1.27

Table 4.2: Maximum separation and mean squared error between the desired signal and the AC for *Internal Leaks*.

mands. The hydraulic fluid is decanted from main to annular chamber, equalizing the pressures and neglecting the piston rod movement. Adaptive Controllers recover the optimal behaviour partially, as they increase the proportional valve opening accordingly to the amount of fluid conveyed.

4.2.3 3^{rd} Experiment: Sensor Fault

Sensors are operators gates to understand the machine behaviour, as they inform continuously about their status. Manufacturers offer multiple classes attending to the measured physical property, such as manometers and flow-meters. However, all of them share a standard feature; an electronic circuit transforms a mechanical signal into an electrical one.

Industrial systems have several sensors installed across their circuit, reporting the current machine status to operators. Their electronic design converts with high precision mechanical signals from the physical properties into electrical responses understandable by the controller. Although these measures represent trustworthy the current machine status, they reduce their effectiveness in industrial environments.

The heavy machinery installed on modern industries produces electrostatic radiation that alters sensor measurements. These continuous pulses produce divergences on the estimated signals known as noise disturbances. Manufacturers are well-aware of these adverse conditions, integrating frequency filters on the sensors to mitigate the adverse effect.

Besides the physical solutions, PID control algorithm design also behaves actively against these faults, reducing their impact on the system considerably. The experiments sought to verify if these architectures recovers optimal behaviour when sensors are under a 10% or 90% fault. The test will also prove if MRACs are suitable for these drawbacks.

4.2.3.1 Hydraulic-Press Overview

Modern industries have upgraded machines introducing additional sensors to monitors its behaviour. This increment brings constant feedback about the system status. Nevertheless, it also offers a gate to this fault, as the probability of appearing is directly proportional to the number of sensors installed.

Although modern industrial machines are prone to suffer noise dissonance on their sensors, having multiple devices running in parallel mitigates the time spent detecting the fault. Operators discern the malfunctioning equipment when at least one component brings an incorrect measure while the others maintain their nominal or optimal values.

Whereas operators identify this fault through the measures reflected on the HMI, the detection algorithm compares the current values against the database and the overall performance. For instance, when a sensor describes the altered behaviour, but the others continue operating correctly, the detection algorithm discerns a component malfunction rather than a global fault.

The experiments resemble the previous studies (see Sections 4.2.1 and 4.2.2), where a manometer suffers from noise dissonance at 10% or 90%. The Hydraulic-Press executes its cycle controlled by three different controllers, one based on PID algorithms and two on MRACs architectures. The latest based on two adaptive rules, MIT or Lyapunov.

When a 10% disturbance alters the manometer, the PID controller attempts to recover the nominal behaviour (see Fig. 4.54). Despite there are irregularities across the first iteration, the controller regains the stability and continues performing under optimal conditions on the subsequent cycles.

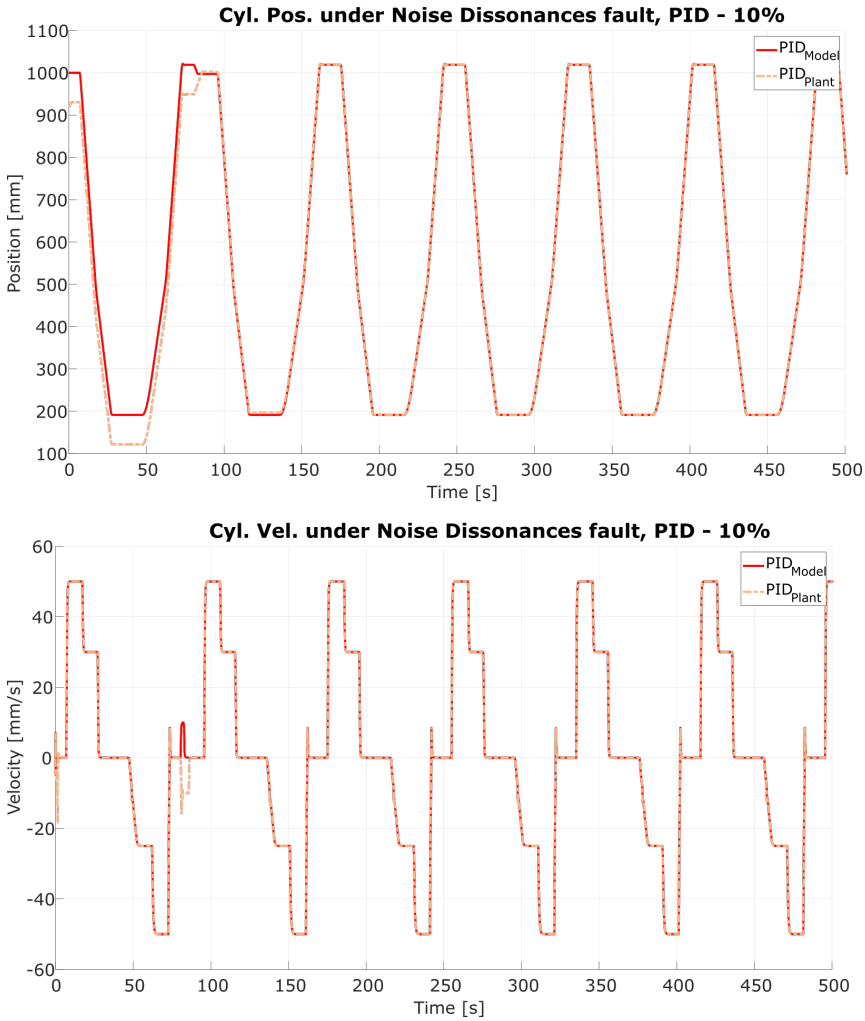


Figure 4.54: Hydraulic-Press Position and Velocity under *Noise Dissonance* fault for the PID controller, performance against a 10% fault.

Similarly, MRACs based on MIT rules have a divergence on the first iteration, corrected during the following cycles (see Fig. 4.55). After the algorithm detects the fault, the adaptation mechanism alters the controller signal accordingly to the noise disturbance.

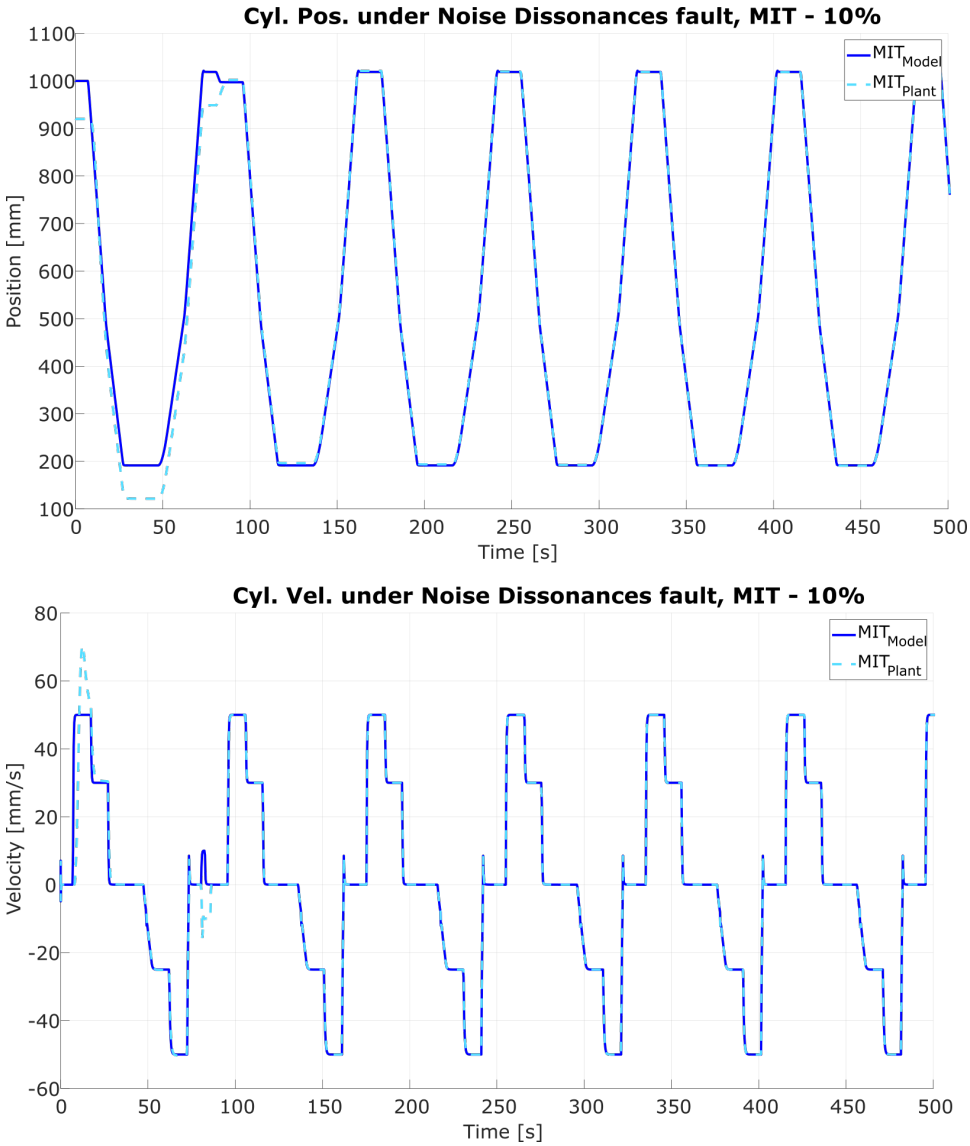


Figure 4.55: Hydraulic-Press Position and Velocity under *Noise Dissonance* fault for the MIT rule controller, performance against a 10% fault.

Lyapunov controllers also diverge during the first iteration and recover the optimal performance on the subsequent cycles (see Fig. 4.56). In comparison with MIT rule algorithms, the adaptation process requires more cycles to attain a global minimum. During this stage, the current machine response differs slightly from the model performance.

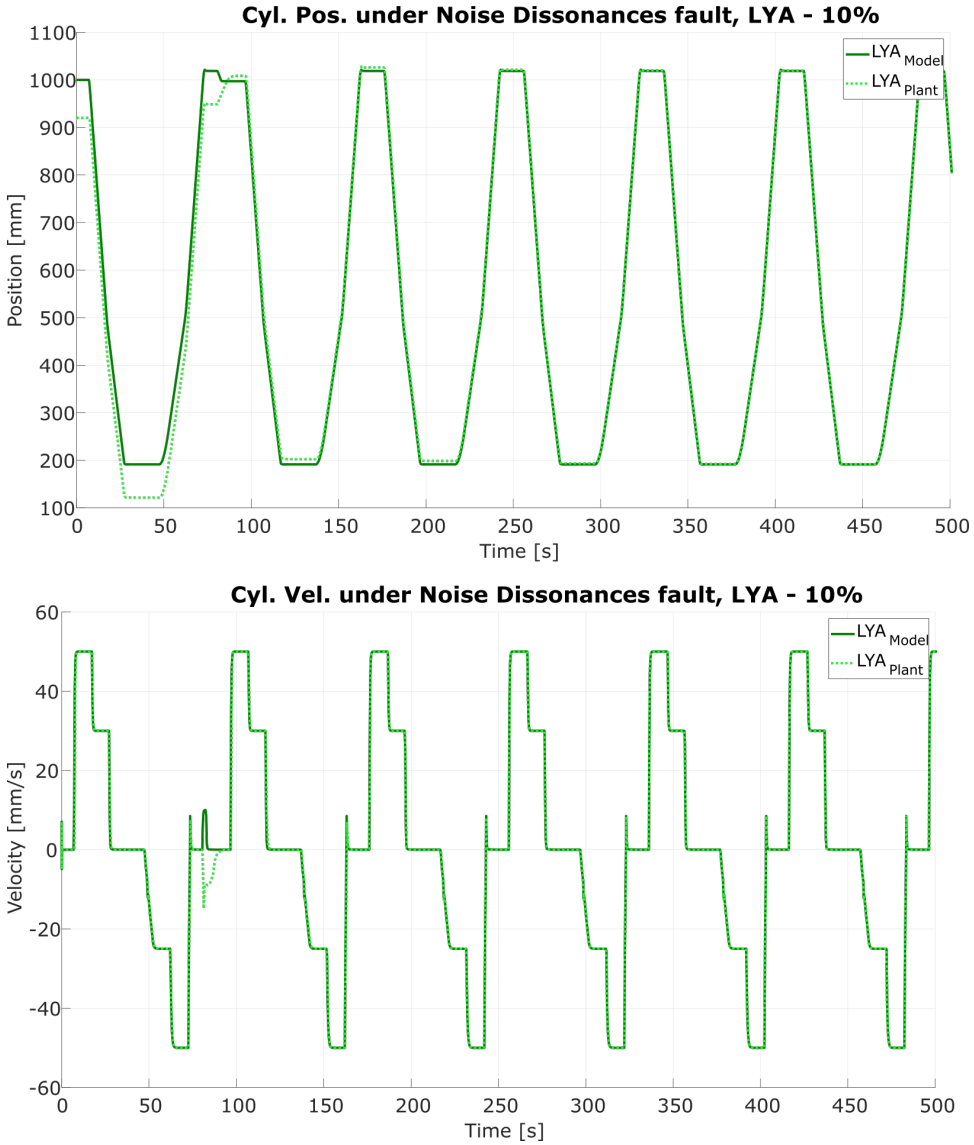


Figure 4.56: Hydraulic-Press Position and Velocity under *Noise Dissonance* fault for the Lyapunov rule controller, performance against a 10% fault.

When the fault spreads to 90%, the system controlled by the PID algorithm describes a similar pattern as the 10% case (see Fig. 4.57). During the first cycle, the fault alters the cylinder position and velocity. On the subsequent iterations, the controller regulates the plant accordingly to surpass the drawback and recover the optimal performance.

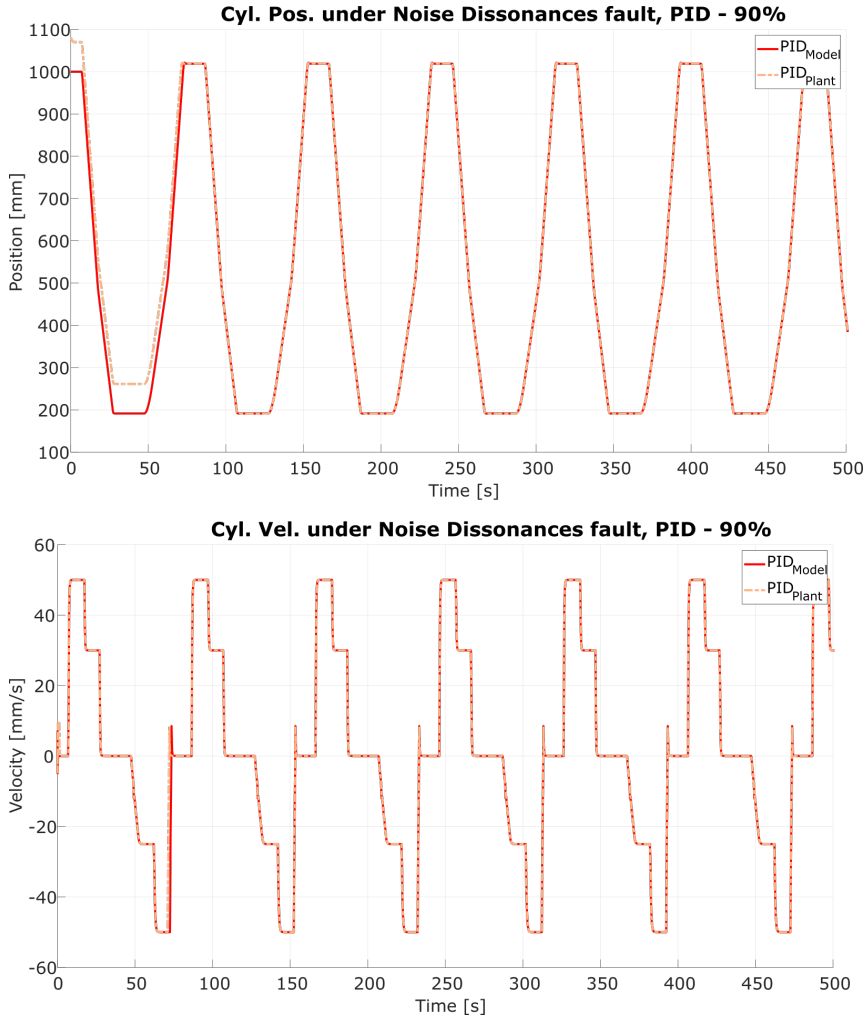


Figure 4.57: Hydraulic-Press Position and Velocity under *Noise Dissonance* fault for the PID controller, performance against a 90% fault.

Similarly, MIT controllers present discordances on the first iteration, corrected across the successive cycles (see Fig. 4.58). On this situation, the adaptation process requires more cycles to attain stability, as the controller updates the signal gradually until nullifying the error.

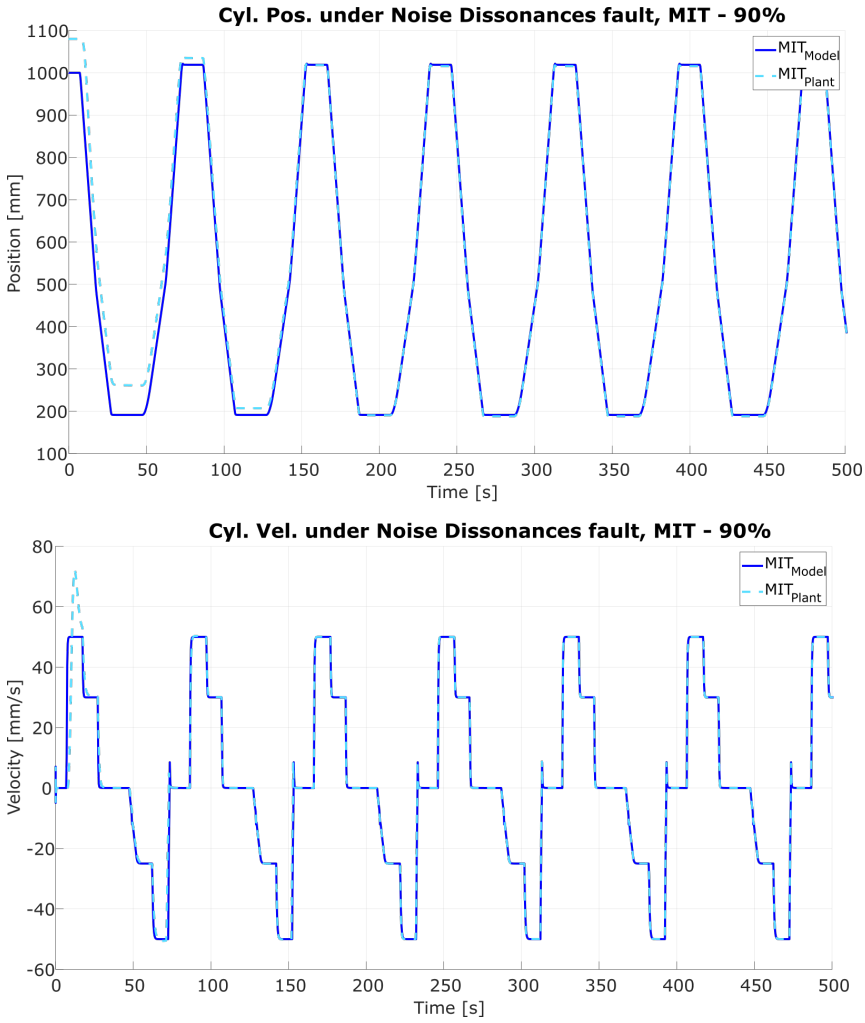


Figure 4.58: Hydraulic-Press Position and Velocity under *Noise Dissonance* fault for the MIT rule controller, performance against a 90% fault.

On their behalf, MRACs based on Lyapunov rules maintain the stability in the cylinder position and velocity across every iteration (see Fig. 4.59). They experiment with an adaptation process that minimises the divergences and recovers the optimal performance after a few iterations.

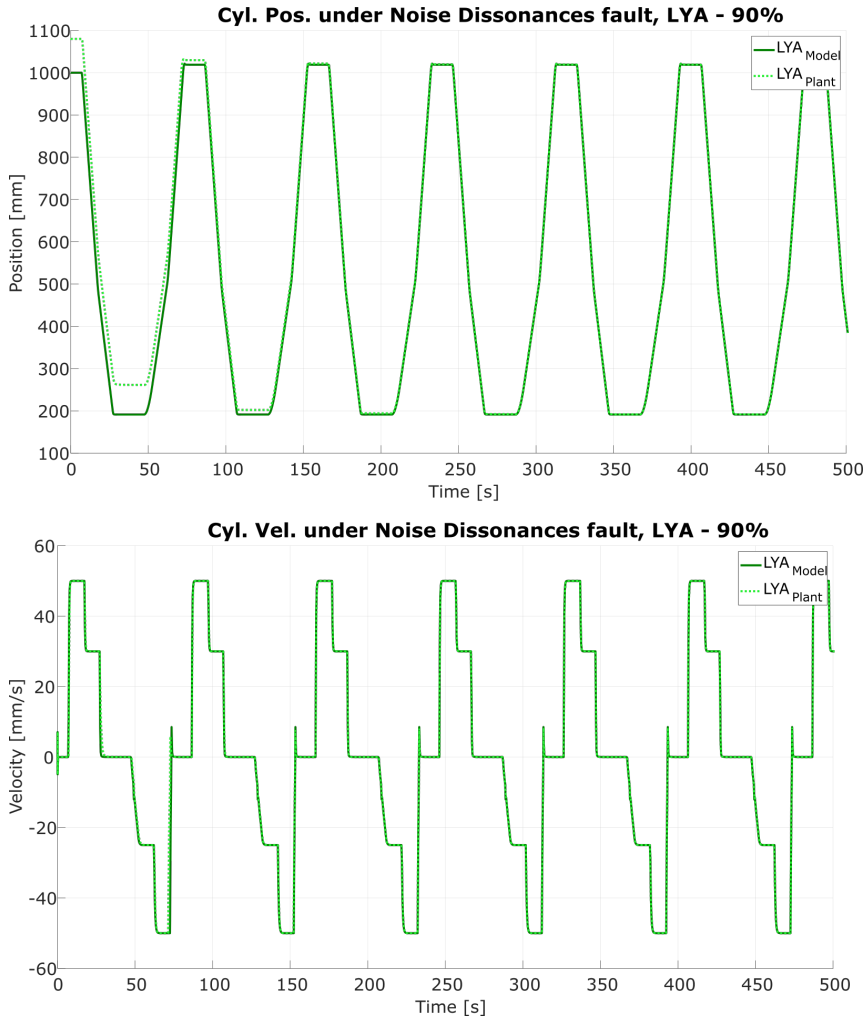


Figure 4.59: Hydraulic-Press Position and Velocity under *Noise Dissonance* fault for the Lyapunov rule controller, performance against a 90% fault.

Noise disturbance faults alter nominal system behaviour; however, after the detection stage, the controller attempts a successful recovery process independently of the algorithm selected (see Fig. 4.60 and 4.61). This drawback produces a disturbance in the measures received from the machine. The controller architectures studied filter the signal, easing the recovery process and attaining faster the optimal performance.

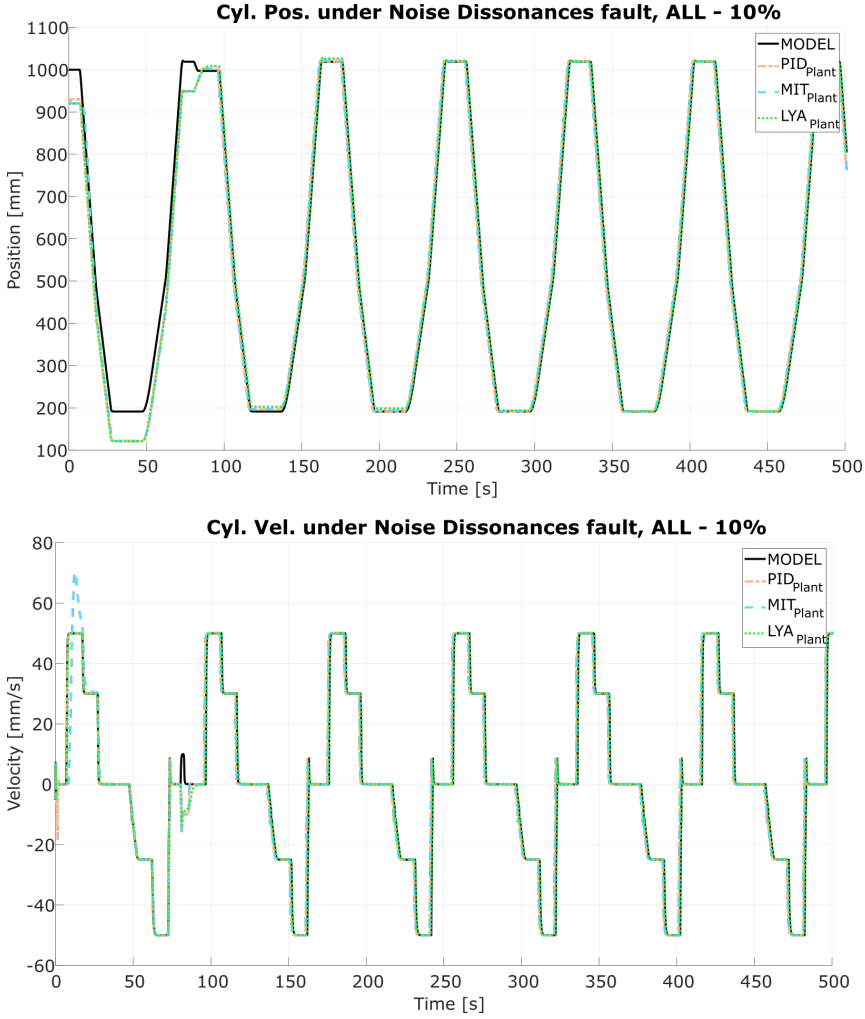


Figure 4.60: Hydraulic-Press Position and Velocity under *Noise Dissonance* fault for all the controllers, performance against a 10% fault.

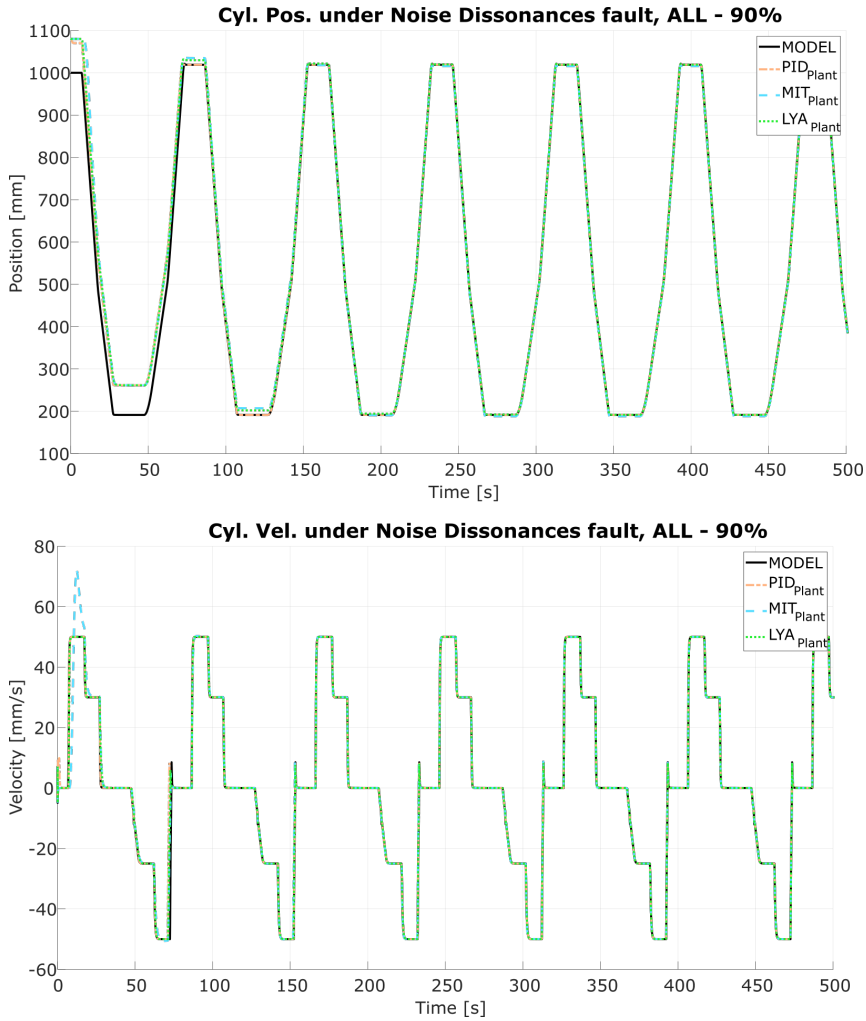


Figure 4.61: Hydraulic-Press Position and Velocity under *Noise Dissonance* fault for all the controllers, performance against a 90% fault.

4.2.3.2 Analytical Approach

After studying the cylinder position and velocity when the Hydraulic-Press suffers from noise disturbances, the maximum separation and the RMSE between the optimal performance and each control architecture has been examined. The measurements presented belong to the first ten iterations of the Hydraulic-Press cycle under both fault effects (10% and 90%).

The maximum separation graph corroborates the previous assumptions (see Fig. 4.62). Each control architecture reduces considerably the rift between model and plant after the first cycles. For instance, when the fault has a 10% harming effect, the PID controller almost nullifies this divergence on the first four cycles. Similarly, ACs bring the current performance closer to the optimal.

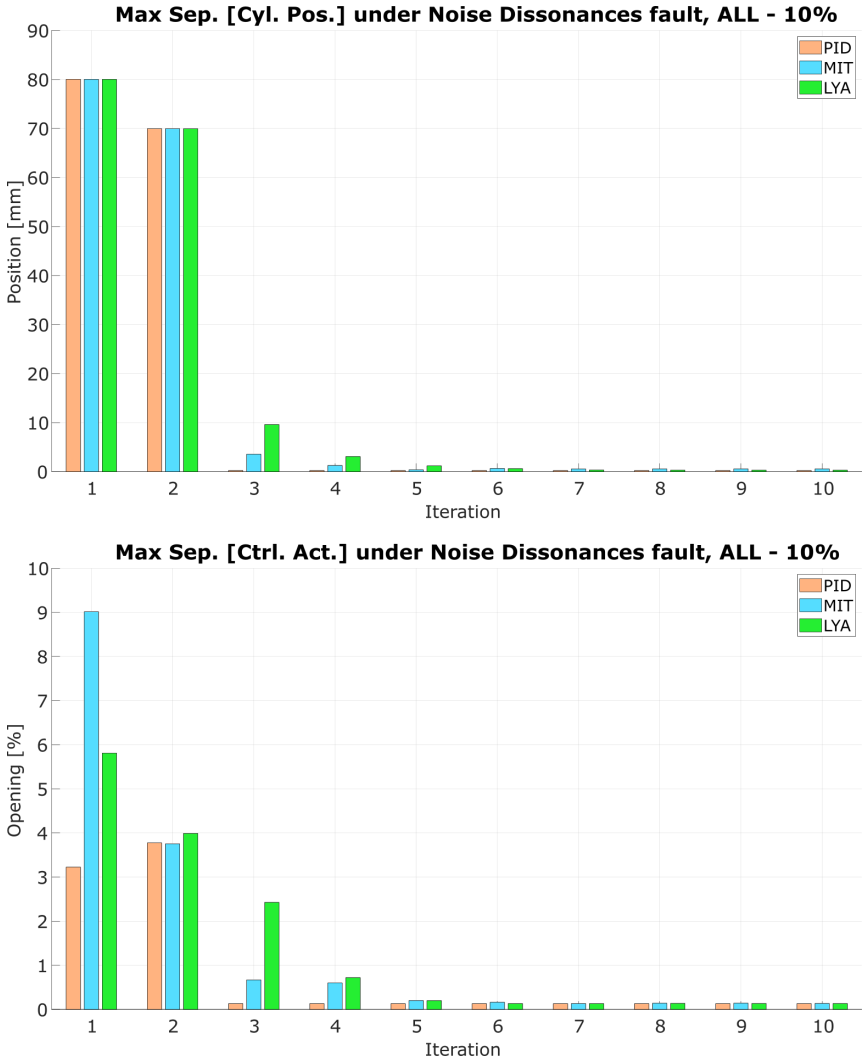


Figure 4.62: Maximum separation between real system and model in a *Noise Dissonance* fault for all the controllers, performance against a 10% fault.

The RMSE between the cylinder position and velocity also has a decreasing tendency (see Fig. 4.63). Initially, all the controllers present a similar behaviour, being PID architectures the ones that minimise earlier the error. Due to their algorithm, adaptive controllers require more cycles to acquire an identical behaviour as the nominal case.

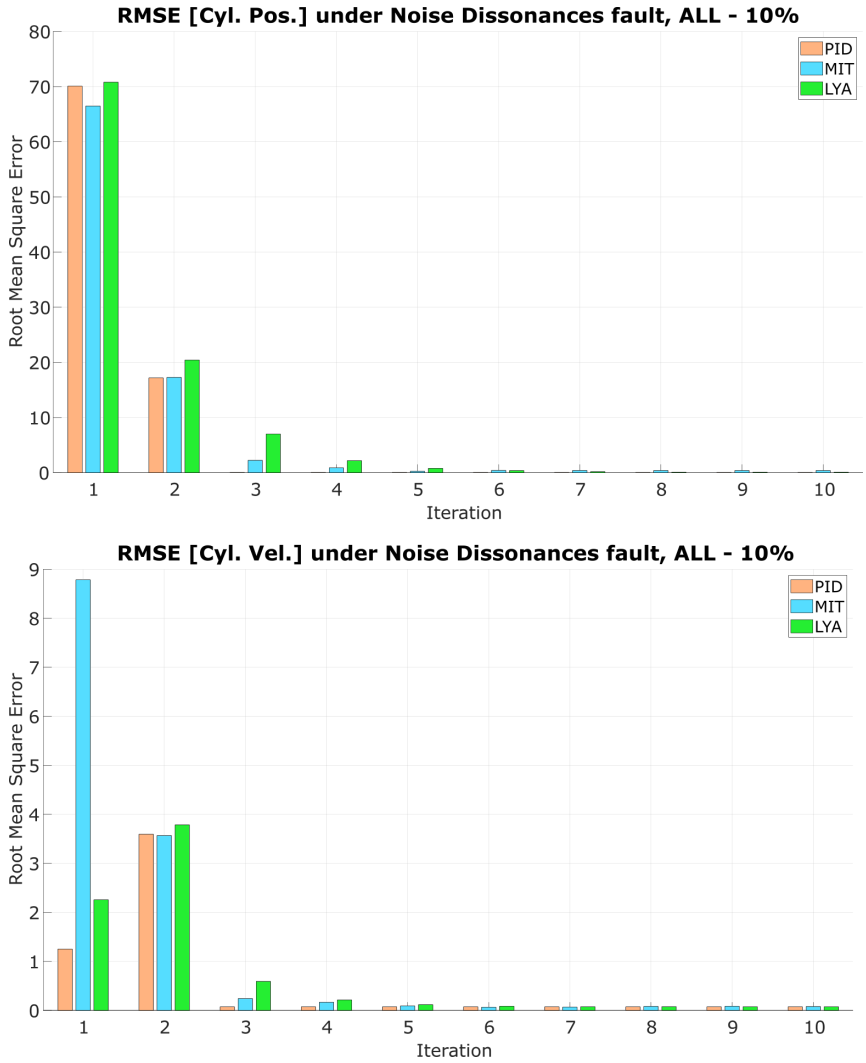


Figure 4.63: Root Mean Square Error between real system and model in a *Noise Dissonance* fault for all the controllers, performance against a 10% fault.

Even when the fault spread to 90% of their theoretical rate, the maximum separation between cylinder position and control action against their nominal performance reaches a null situation (see Fig. 4.64). On this case, MIT rule algorithms require more iterations to attain this status, as their algorithm focuses on local minimums instead of global ones.

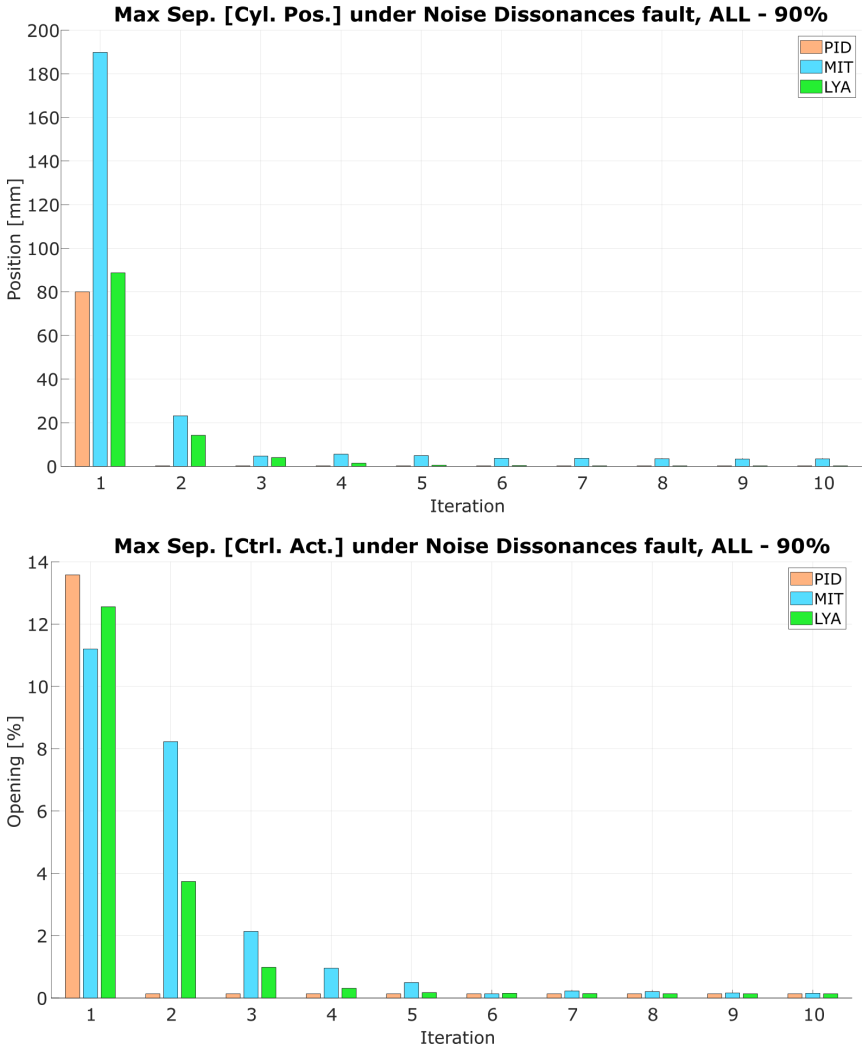


Figure 4.64: Maximum separation between real system and model in a *Noise Dissonance* fault for all the controllers, performance against a 90% fault.

This effect has also a reflection on the cylinder position and velocity (see Fig. 4.65). The HP responses diverge from the nominal performance when the fault emerges, but the controllers recover an optimal behaviour after the first iterations. Notice how MIT rule algorithms, despite maintaining the system stable, their accumulated error is bigger than the other alternatives.

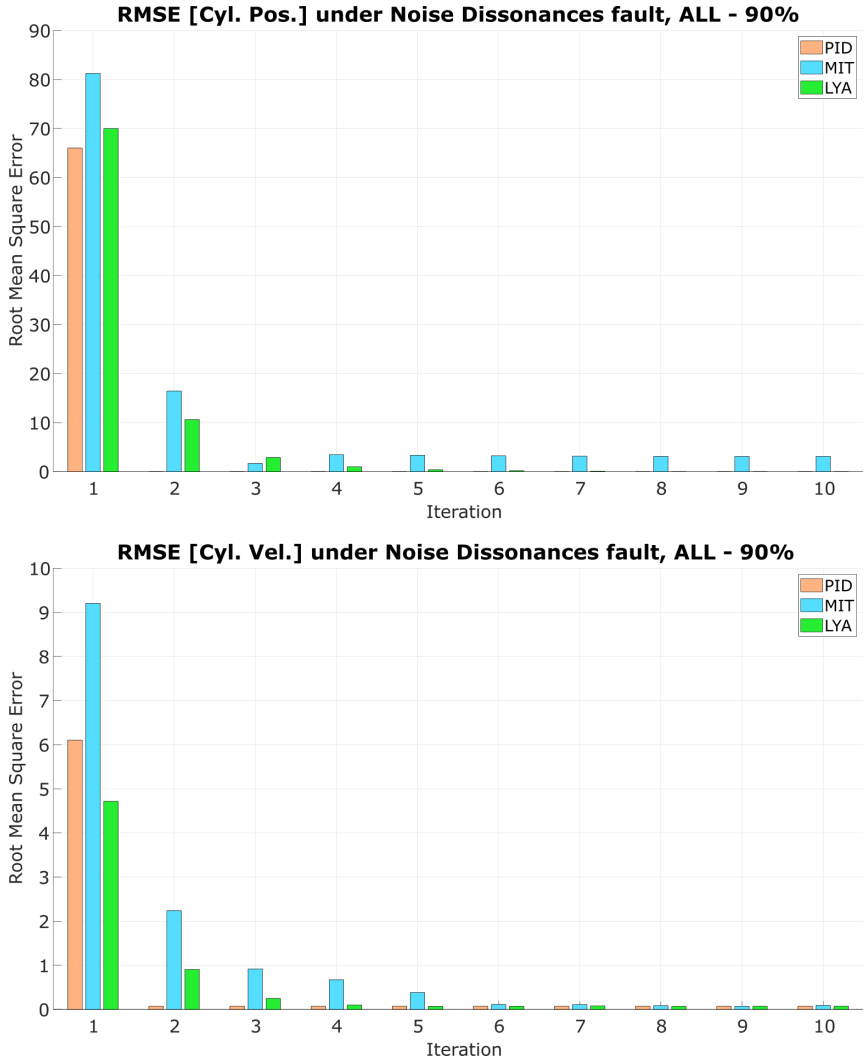


Figure 4.65: Root Mean Square Error between real system and model in a *Noise Dissonance* fault for all the controllers, performance against a 90% fault.

The study continues presenting in detail the cylinder position and the control action when the PID algorithm controls the plant (see Fig. 4.66). After the first cycles, the controller recovers the nominal behaviour, and both signals coincide, maintaining the stability for the successive iterations.

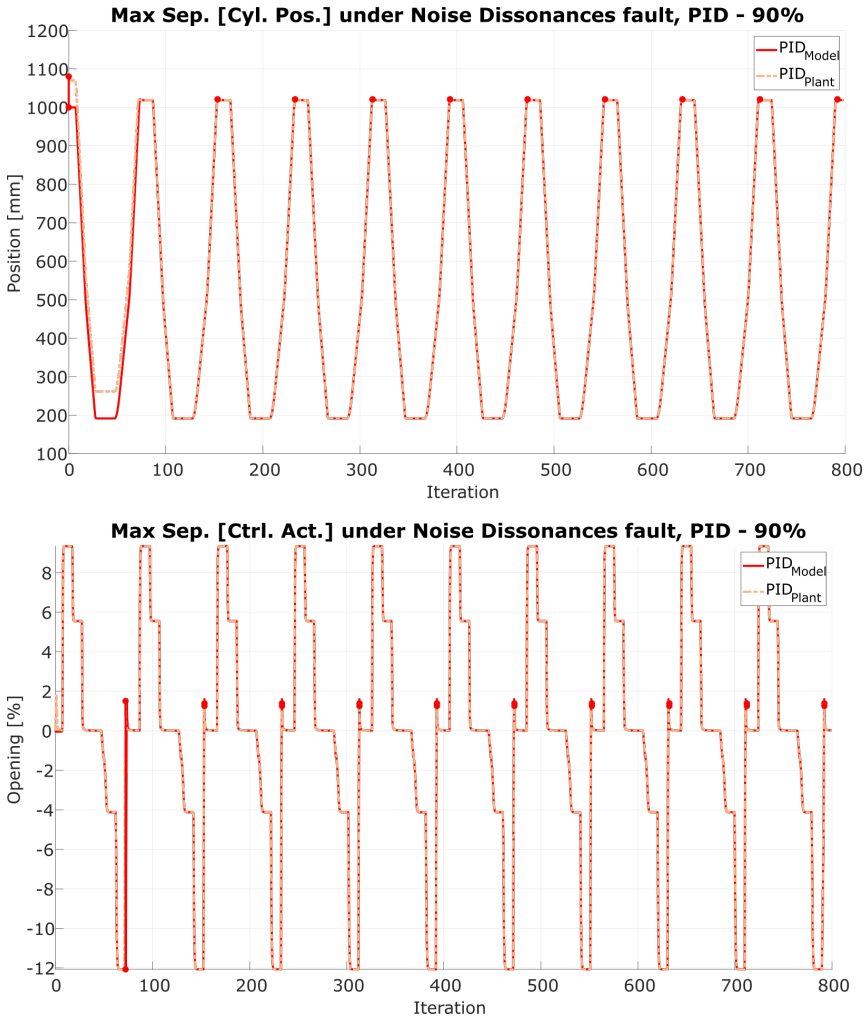


Figure 4.66: Maximum separation between real system and model in a *Noise Dissonance* fault for the PID controller, performance against a 90% fault.

As the controller recovers the nominal behaviour during the first cycles, the RMSE tends to disappear after these stages in the cylinder position, remaining only a residual error due to the fault in the cylinder velocity (see Fig. 4.67). Despite the PID algorithms efforts to filter these last disturbances, the fault still has a persistent effect on the plant signal.

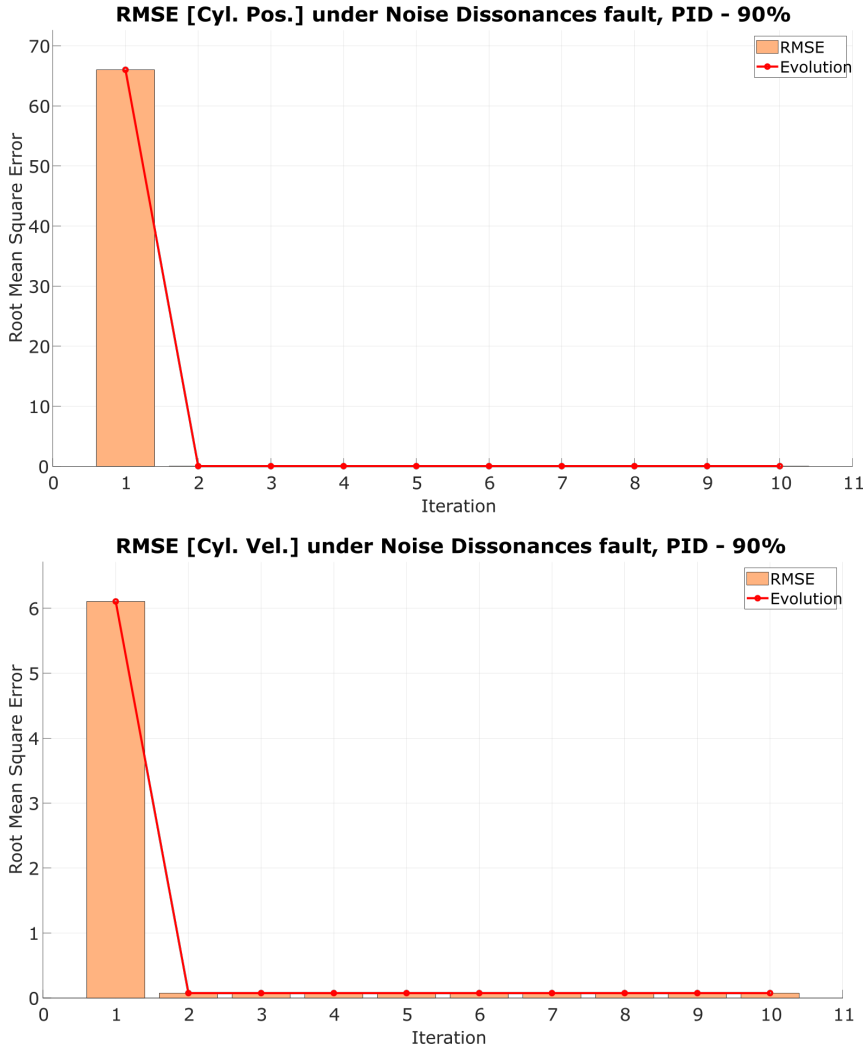


Figure 4.67: Root Mean Square Error between real system and model in a *Noise Dissonance* fault for the PID controller, performance against a 90% fault.

The study continuous analysing the values recorded for the MIT rule controller (see Fig. 4.68). During the fault emergence, the system achieves the maximum separation, in both, cylinder position and control action, minimising this error across the subsequent iterations.

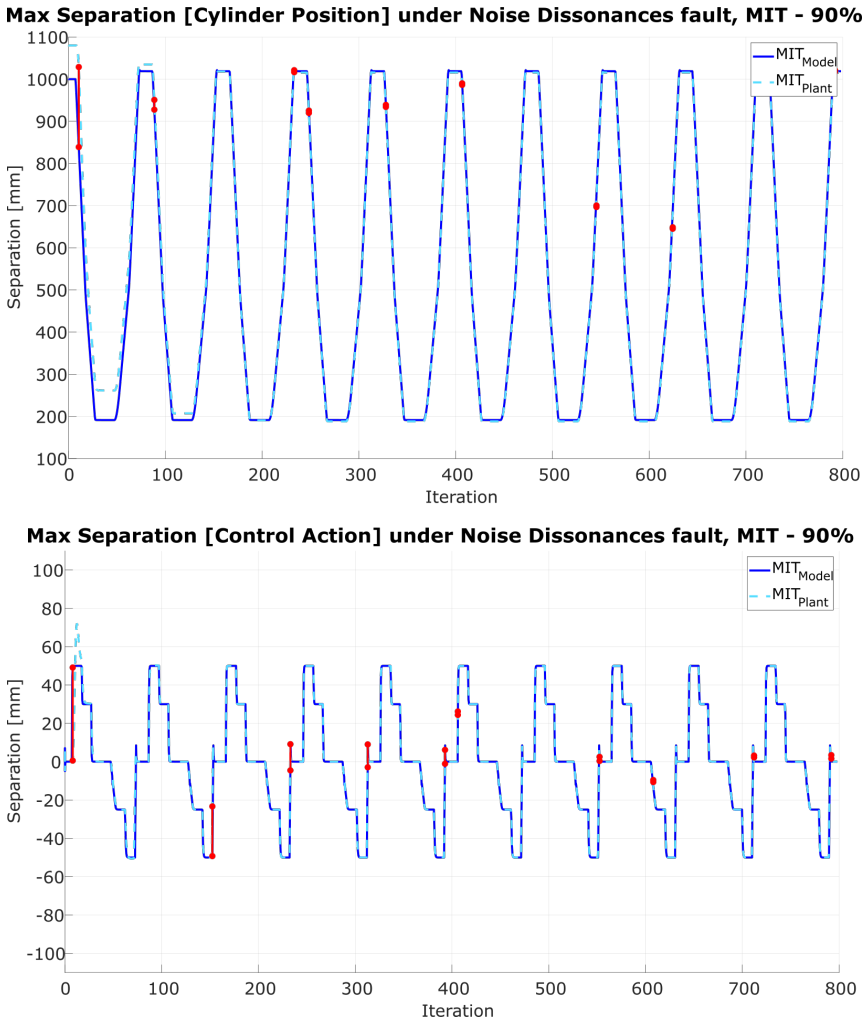


Figure 4.68: Maximum separation between real system and model in a *Noise Dissonance* fault for the MIT controller, performance against a 90% fault.

The RMSE study also presents this decreasing tendency (see Fig. 4.69). During the initial cycle, the cylinder position and velocity present the maximum error, minimising it until reaching a stable state in the subsequent iterations. As MIT rule controllers sought to acquire the minimum error between model and signal through local minimums instead of global ones, the position RMSE responses have a slight increase after the third cycle.

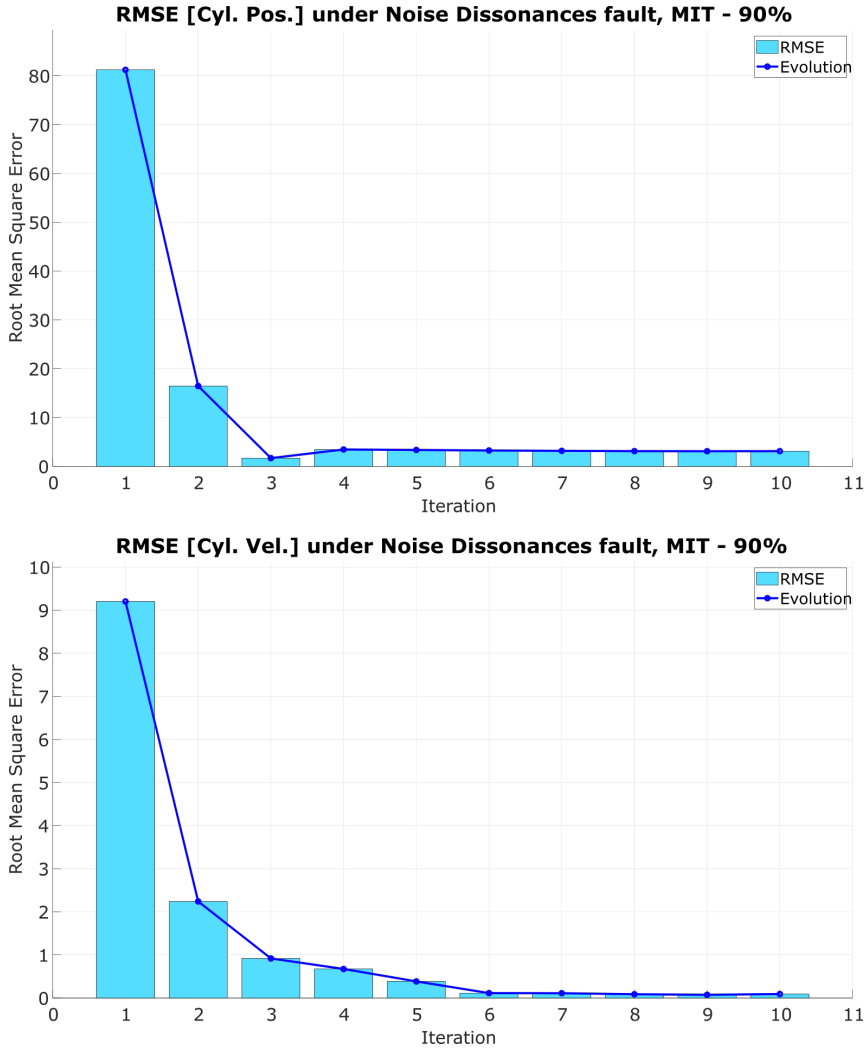


Figure 4.69: Root Mean Square Error between real system and model in a *Noise Dissonance* fault for the MIT controller, performance against a 90% fault.

Lyapunov rule controllers offer the best performance, as they minimise the error constantly (see Fig. 4.70). The cylinder position remains practically identical to the nominal behaviour, minimising the dispersion across each iteration. Similarly, the control action maintains the system stable.

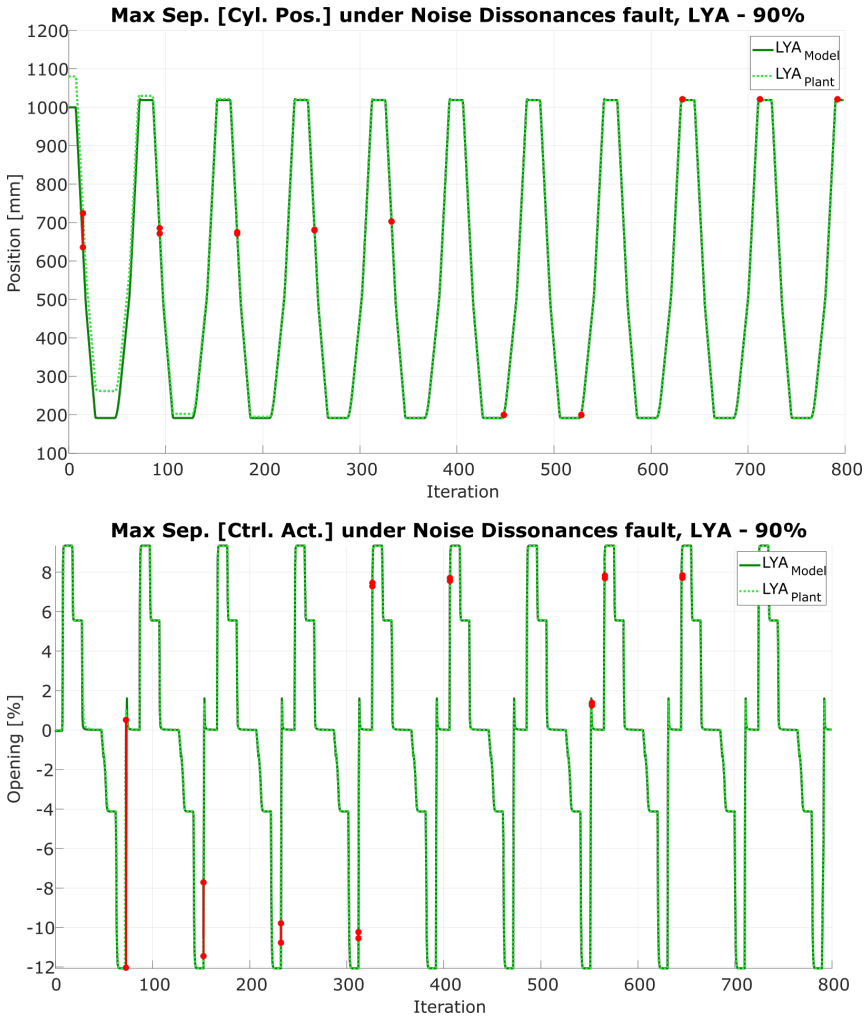


Figure 4.70: Maximum separation between real system and model in a *Noise Dissonance* fault for the Lyapunov controller, performance against a 90% fault.

RMSE study corroborates the early assumptions, as they present the error between the cylinder position and velocity across the first ten iterations (see Fig. 4.71). After the fifth cycle, the position signal presents a null error, while there is a small divergence on the velocity response. Despite this deviation, the controller maintains the system stable across all the iterations, surpassing the fault effectively.

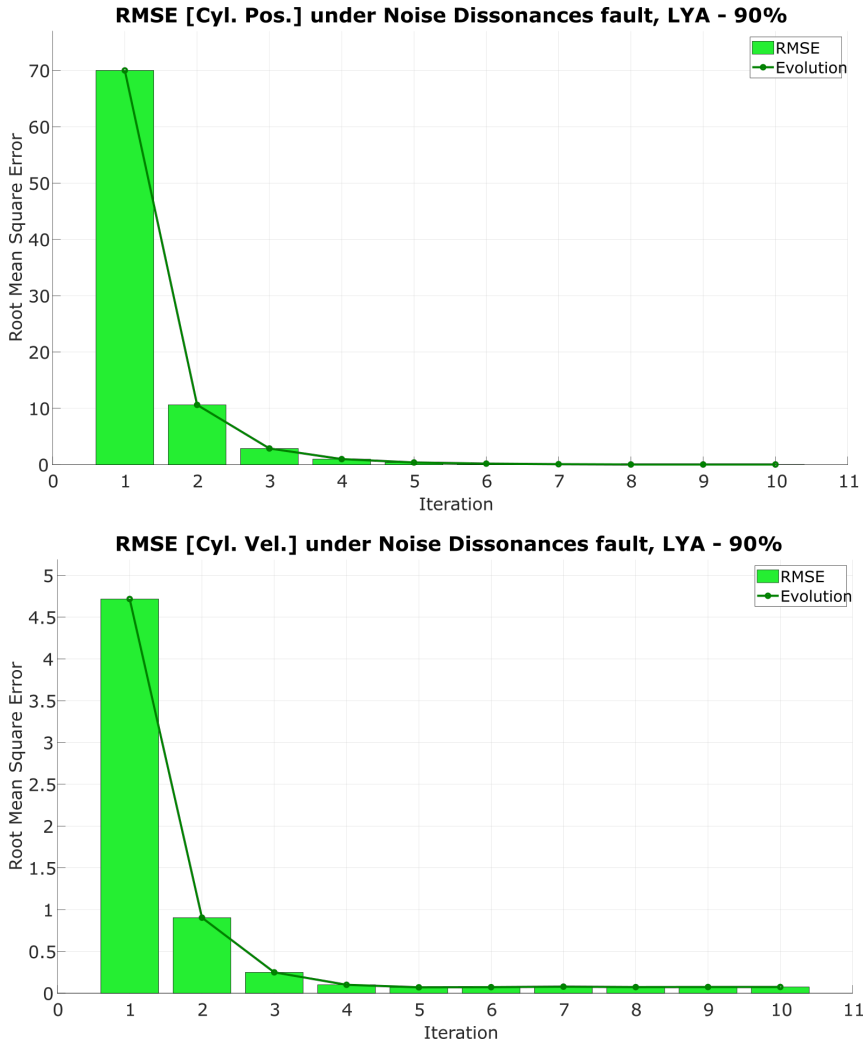


Figure 4.71: Root Mean Square Error between real system and model in a *Noise Dissonance* fault for the Lyapunov controller, performance against a 90% fault.

On noise dissonance faults, the system continues performing the cycle under optimal conditions despite the fault emergence (see Table 4.3). Across their first cycles, the signal has presented a degraded behaviour, an issue addressed by

the controller on the subsequent iterations. After filtering the signal, the control algorithms perform almost an identical position response with a slight divergence on the velocity graph.

$FAULT$		$NoiseDissonance$					
$VALUE$		10%					
$CONTROL$		PID		MIT		LYA	
$CYCLE$		1^{st}	10^{th}	1^{st}	10^{th}	1^{st}	10^{th}
CP	MAX	80.04	0.24	80.01	0.57	80.01	0.30
	$RMSE$	70.10	0.04	66.49	0.38	70.8	0.08
CV	MAX	3.23	0.13	9.01	0.14	5.81	0.14
	$RMSE$	1.25	0.07	8.78	0.08	2.26	0.07
$VALUE$		90%					
$CONTROL$		PID		MIT		LYA	
$CYCLE$		1^{st}	10^{th}	1^{st}	10^{th}	1^{st}	10^{th}
CP	MAX	80.00	0.24	189.83	3.38	88.72	0.24
	$RMSE$	66.00	0.03	81.23	3.11	69.99	0.03
CV	MAX	13.58	0.13	11.20	0.15	12.56	0.14
	$RMSE$	6.10	0.07	9.20	0.09	4.72	0.07

Table 4.3: Maximum separation and RMSE between the desired signal and the AC for *Position Sensor*.

Although noise disturbance faults are common in modern industries, they are easily identify by operators and detection mechanism. Besides, conventional control techniques, such as PID algorithms, have integrated filters to minimise this adverse effect. These premises establish a system that maintains stability despite the emergence of these faults. Even though the performance of PID controllers is stable, MRACs offer a suitable platform for other fault cases with slightly better behaviour against these faults.

4.2.4 Adaptive Model Reference Experiment

The previous studies present the recovery process of three control architectures against actuator, plant and sensor faults (see Section 4.2.1, 4.2.2 and 4.2.3). For faults with moderate harm rates, PID controllers maintain the system under the optimal performance region; however, they are inefficient when their harm spreads. MRACs offer a viable alternative, as they maintain the system stable indepen-

dently of the degradation suffered. Between both rules, MIT and Lyapunov, the latest present a better performance than their counterpart.

Although the system remains stable and controlled, there are instances where the fault has altered the plant saturation limits. Despite MRAC efforts to adapt the signal, these tight restraints narrow the adaptation mechanism, preventing the algorithm from reaching the global minimum. These instances neglect a proper system performance, even though the adaptation mechanism could continue lessening the breach between model and plant.

This thesis has proposed an enhance to increase MRACs reliability, substituting the conventional model reference stage with a Digital-Twin controlled by an adaptive algorithm (see Section 3.2.6). This novel feature improves the overall reliability, as the model control algorithm responds actively to prevent the fault. Instead of only adapting the principal controller, the mechanism also accommodates the Digital-Twin accordingly to the fault.

The capabilities of this novel mechanism has been tested repeating the experiments carried out for actuator faults (see Section 4.2.1). These round of essays has reduced the Hydraulic-Press cycle time, increasing the cylinder speed. This practice increases the fault effect, as the actuator reach earlier its mechanical limits. Similarly to the previous analyses, the study exhibits the system behaviour when the mechanical fatigue has degraded the proportional valve performance a 10% or 90%.

4.2.4.1 Hydraulic-Press Overview

As previously stated, operators detect these faults quickly, as they alter the piston rod displacement directly. On this round of experiments, the cylinder velocity has increased its nominal value to accentuate the harming effect, unbalancing earlier the plant. The three control algorithms attempt to recover the optimal performance when the fault emerges and degraded the behaviour a 10% or 90%.

After the velocity increase, the Hydraulic-Press cycle suffers a re-accommodation stage where the cylinder recovers the initial position (see Fig. 4.72). When the actuator performance suffers for a 10% mechanical fatigue fault, the PID controllers attempt successfully to recovery the optimal performance.

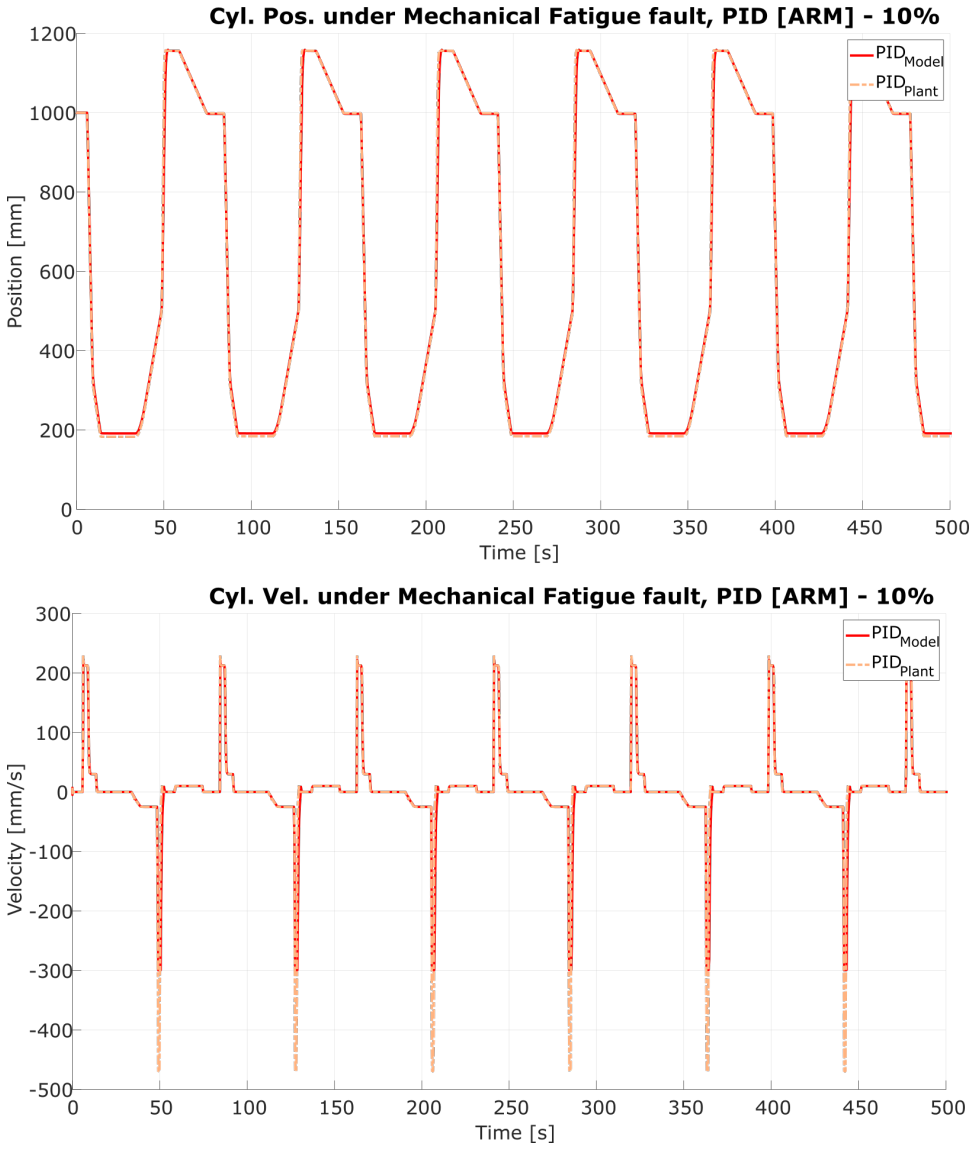


Figure 4.72: Hydraulic-Press Position and Velocity under *Mechanical Fatigue* fault for the PID controller improved with the Adaptive Reference Model, performance against a 10% fault.

Controllers based on MIT rule also recovers the optimal performance, but they stabilise the signal after more cycles (see Fig. 4.73). During the first iteration, after the fault emergence, the cylinder position is below the optimal range, losing the production. This effect appears in the system until the adaptive gains encounter a local minimum, where the plant recovers the optimal performance.

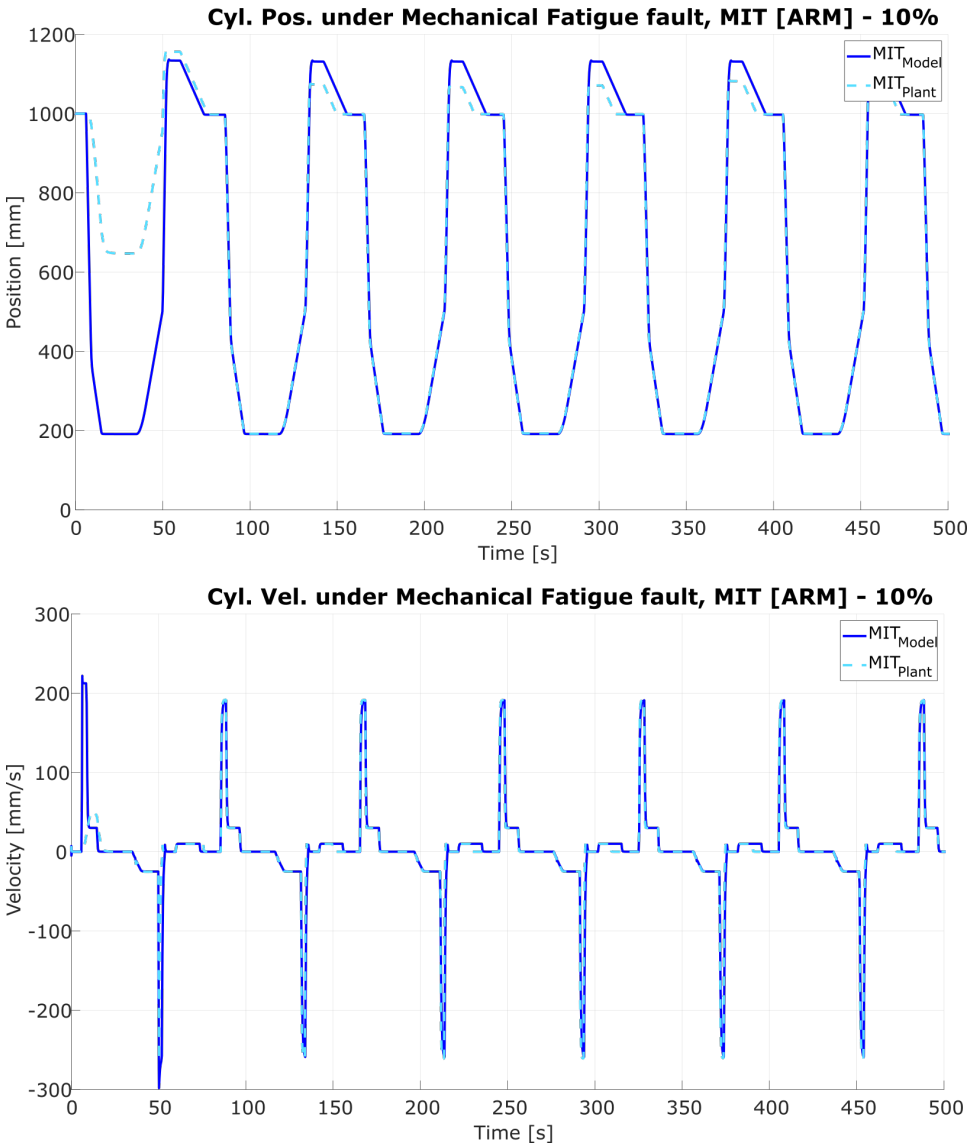


Figure 4.73: Hydraulic-Press Position and Velocity under *Mechanical Fatigue* fault for the MIT rule controller improved with the Adaptive Reference Model, performance against a 10% fault.

Lyapunov controllers also suffer a slight deviation during their first cycle, as the cylinder velocity analysis shows (see Fig. 4.74). However, they encounter the global minimum earlier than their MIT counterpart, recovering the optimal performance without any unproductive cycle. Similarly to the conventional mechanical fatigue study, the three control algorithms recover the production when the fault has degraded the actuator a 10%.

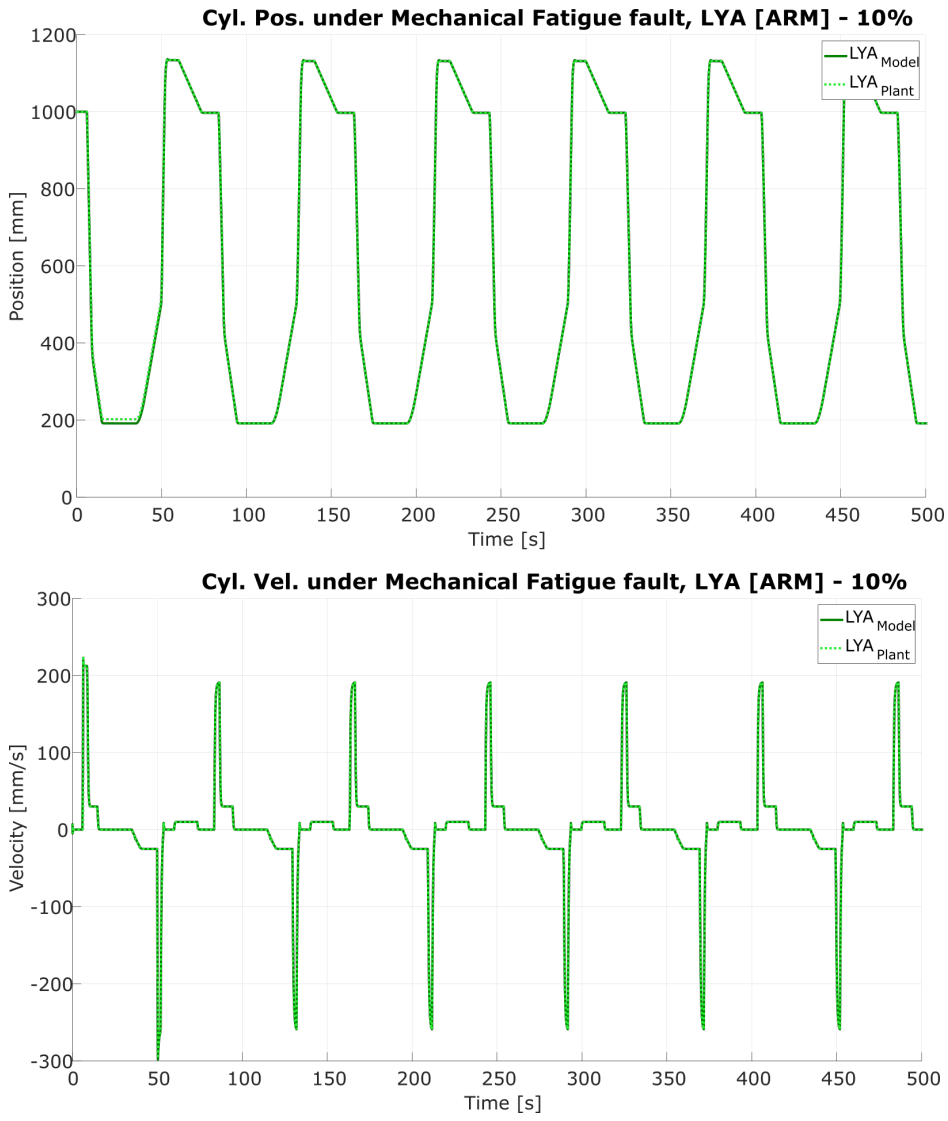


Figure 4.74: Hydraulic-Press Position and Velocity under *Mechanical Fatigue* fault for the Lyapunov rule controller improved with the Adaptive Reference Model, performance against a 10% fault.

The system has the opposite behaviour when the fault spreads to 90% (see Fig. 4.75). The cylinder velocity graph presents a performance below the reference signal, as the proportional valve opening has reached its limits. This effect has also a negative impact on the cylinder position, neglecting the machine production entirely.

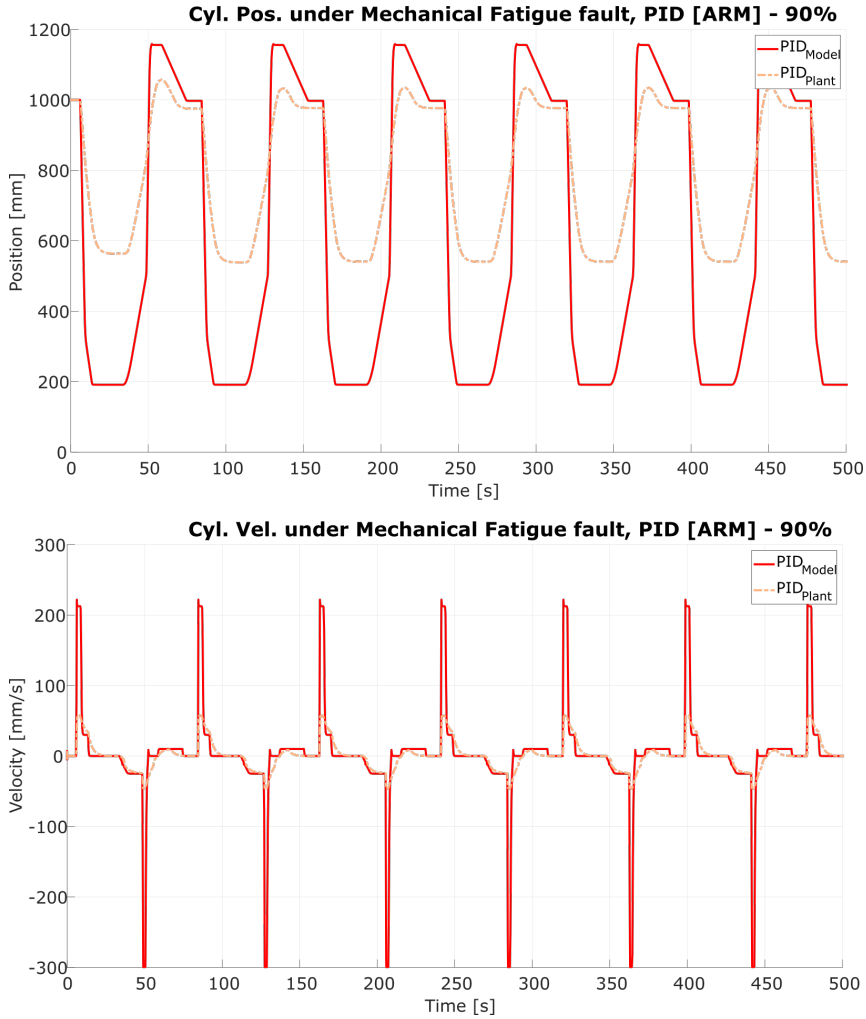


Figure 4.75: Hydraulic-Press Position and Velocity under *Mechanical Fatigue* fault for the PID controller improved with the Adaptive Reference Model, performance against a 90% fault.

MIT rule controllers present an initial disturbance on the signal, as the cylinder position diverges entirely from the optimal behaviour during the first cycles (see Fig. 4.76). However, as the controller maintains the stability, the novel adaptive reference model accommodates the Digital-Twin proportional valve opening to the latest plant behaviour. Afterwards, the principal adaptation mechanism ensures that model and plant signals converge minimising their error.

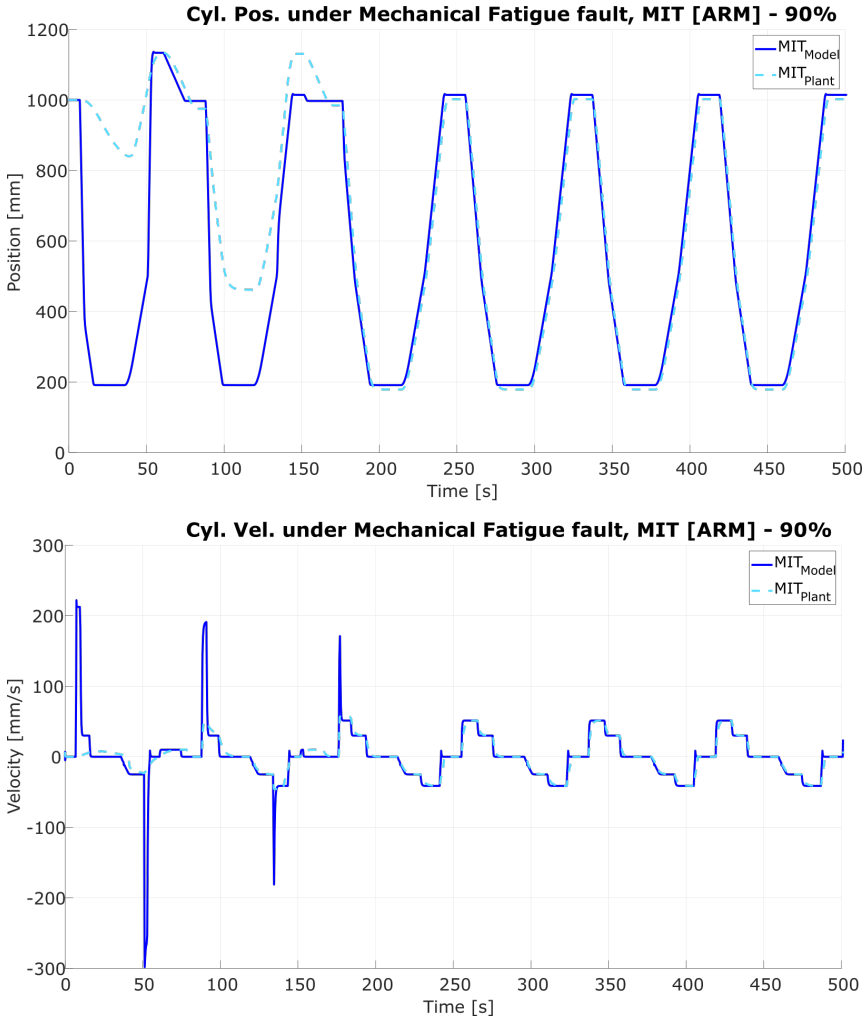


Figure 4.76: Hydraulic-Press Position and Velocity under *Mechanical Fatigue* fault for the MIT rule controller improved with the Adaptive Reference Model, performance against a 90% fault.

Similarly to the previous case, Lyapunov rule controllers also recover the optimal performance after the first iterations (see Fig. 4.77). The model reference adaptive mechanism alters the proportional valve opening accordingly to the fault case. The velocity reduction accommodates the plant signals to the model, minimising the error without losing the stability.

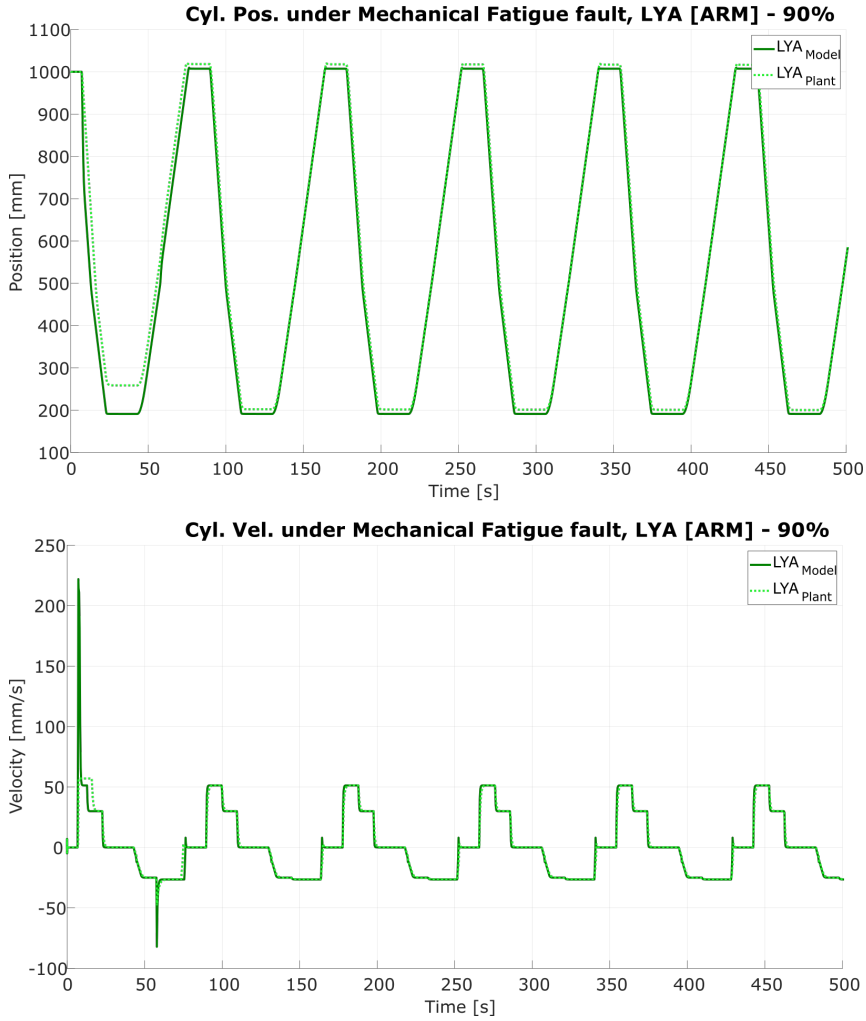


Figure 4.77: Hydraulic-Press Position and Velocity under *Mechanical Fatigue* fault for the Lyapunov rule controller improved with the Adaptive Reference Model, performance against a 90% fault.

During the 10% fault, the controllers respond actively to the fault and maintain an optimal performance with a slight degradation on the cylinder position (see Fig. 4.78). The velocity signal also suffers a small degradation, but it still preserves an optimal performance. Between the three controllers, MIT rule algorithms have the worst performance, as they neglect the production during the first iteration after the fault emergence.

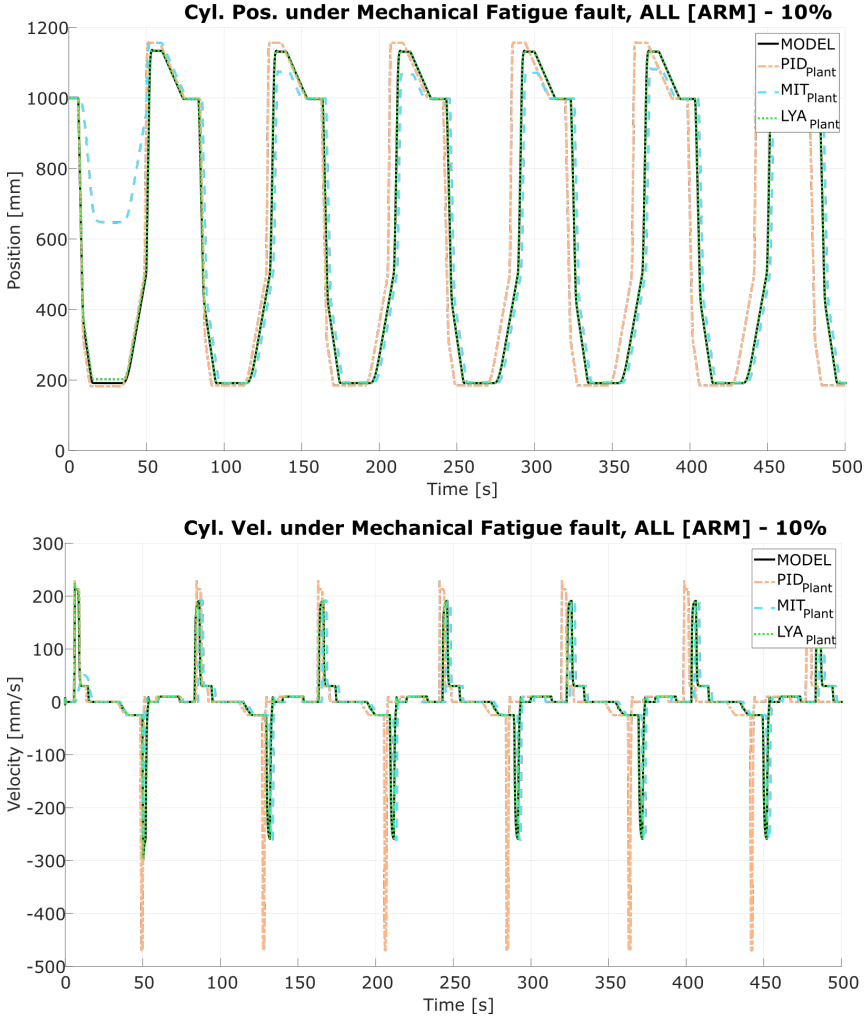


Figure 4.78: Hydraulic-Press Position and Velocity under *Mechanical Fatigue* fault for all the controllers improved with the Adaptive Reference Model, performance against a 10% fault.

When the fault spreads to 90%, the system performance gets completely degraded for PID controllers, while the MRACs recover an optimal performance after some iterations (see Fig. 4.79). Between both rules, Lyapunov ones present better behaviour, as they regain the production earlier than their MIT counterpart.

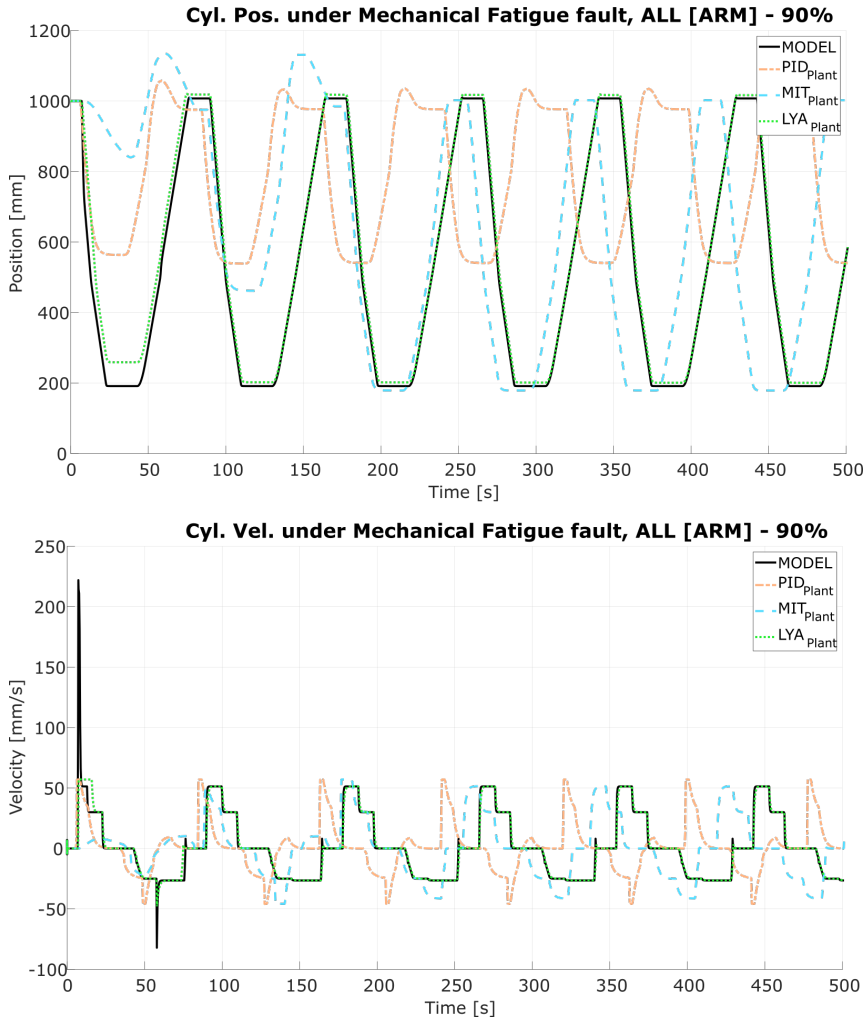


Figure 4.79: Hydraulic-Press Position and Velocity under *Mechanical Fatigue* fault for all the controllers improved with the Adaptive Reference Model, performance against a 90% fault.

4.2.4.2 Analytical Approach

After the position and velocity study, the RMSE between ideal and plant responses have been investigated. The graphs present the Hydraulic-Press behaviour across ten iterations, showing the deviation after the fault and the ACs recovery process. Besides, the cylinder position and control action maximum separation has been presented. Similarly to the previous study, the analysis presents the responses when the mechanical fatigue fault has degraded the proportional valve opening 10% or 90%.

Even when the fault has degraded the opening a 10%, PID controllers present a high deviation with the optimal response. They present a separation around 200 mm in each iteration, as the control action maintains a constant separation during the cycle. The first iteration of MIT controllers has the maximum separation, being an unproductive cycle for the machine. On their behalf, Lyapunov rule controllers consistently reduce the maximum separation, reaching values near zero after ten cycles (see Fig. 4.80).

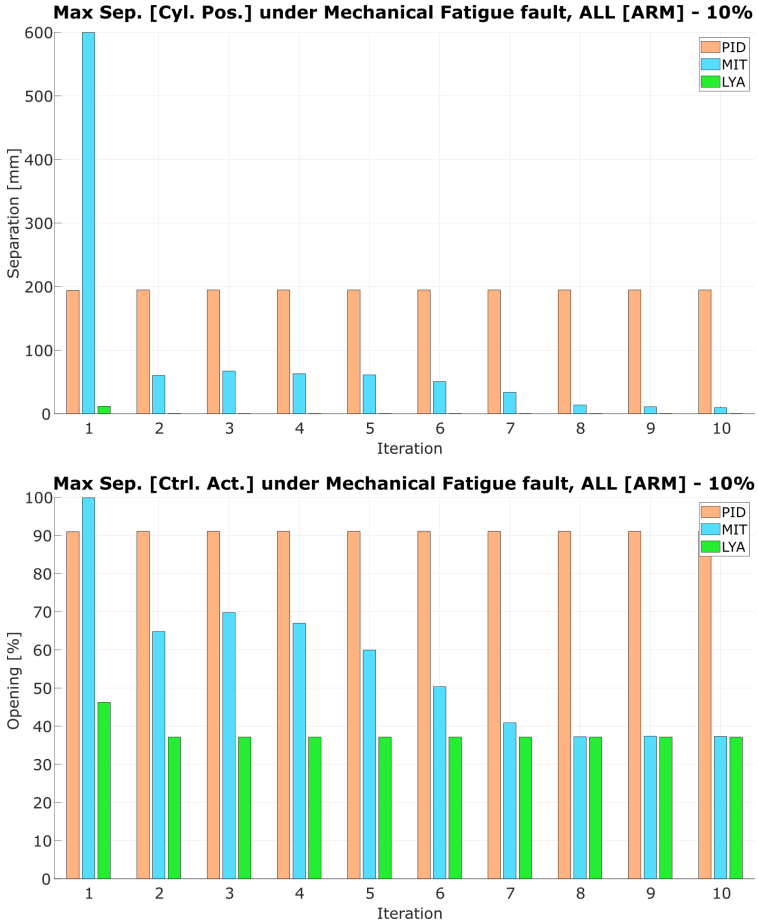


Figure 4.80: Maximum separation between real system and model in a *Mechanical Fatigue* fault for all the controllers improved with the Adaptive Reference Model, performance against a 10% fault.

The RMSE for the cylinder position and velocity corroborates the early assumptions, as the PID controller has a constant error independently of the iteration. MIT controllers reduce the error between the first and second cycle, increasing the RMSE on the third one. The adaptation mechanism attempts to recover the performance, but during these stages reaches a local minimum, increasing the error and starting the adaptation mechanism again. Lyapunov controllers minimise the RMSE satisfactorily across all the iterations (see Fig. 4.81).

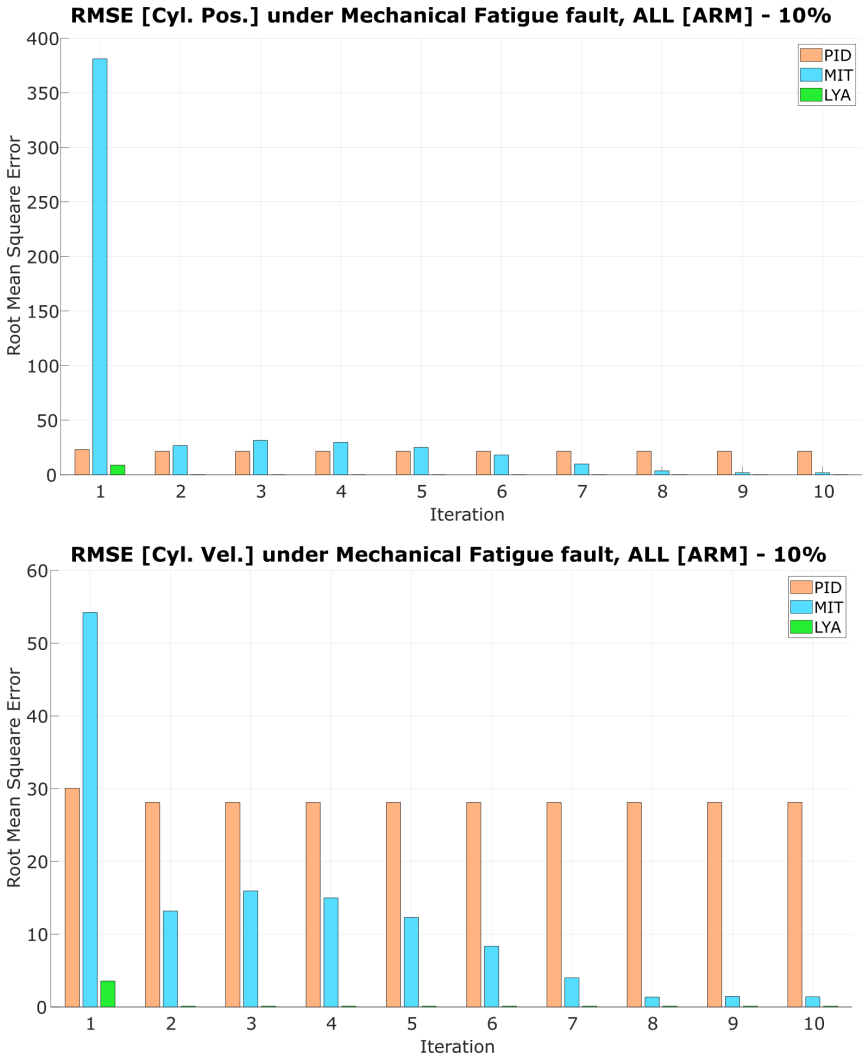


Figure 4.81: Root Mean Square Error between real system and model in a *Mechanical Fatigue* fault for all the controllers improved with the Adaptive Reference Model, performance against a 10% fault.

When the fault spreads to 90%, PID controllers present similar responses as the previous case. When the drawback escalates, the controller attempts to recover the nominal behaviour, but instead of reaching the stability, they maintain a constant separation of 500 mm. Similarly, both ACs rules increase the divergence, but in their case, they reduce the separation, minimising the error across each iteration (see Fig. 4.82).

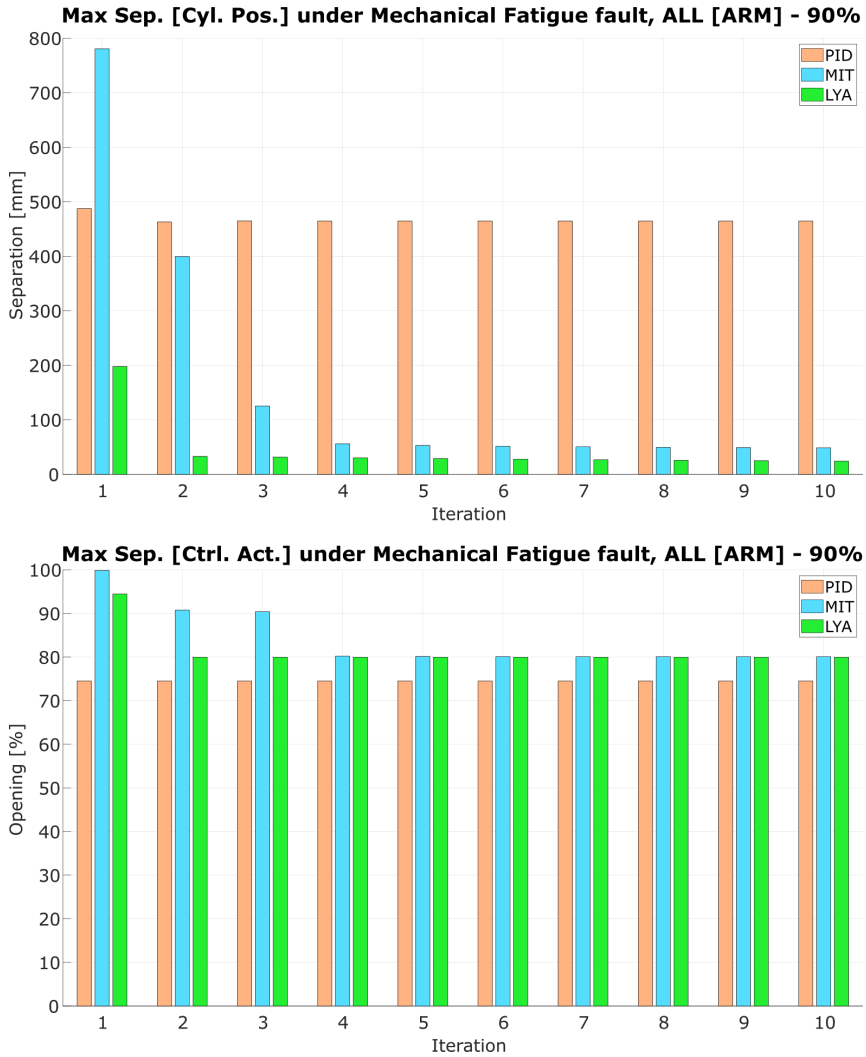


Figure 4.82: Maximum separation between real system and model in a *Mechanical Fatigue* fault for all the controllers improved with the Adaptive Reference Model, performance against a 90% fault.

Similarly to the previous analysis, the RMSE corroborates the early assumptions (see Fig. 4.83). PID controllers have a significant separation in the cylinder position and velocity, performing below the optimal point. The machine is unable to sustain the manufacturing process, ending its operation. MIT rule controllers maintains the system stable, recovering the production partially after the first unproductive cycles. Lyapunov controllers diverge significantly during the first cycle, but, afterwards, the adaptation mechanism minimises this separation without com-

promising the production.

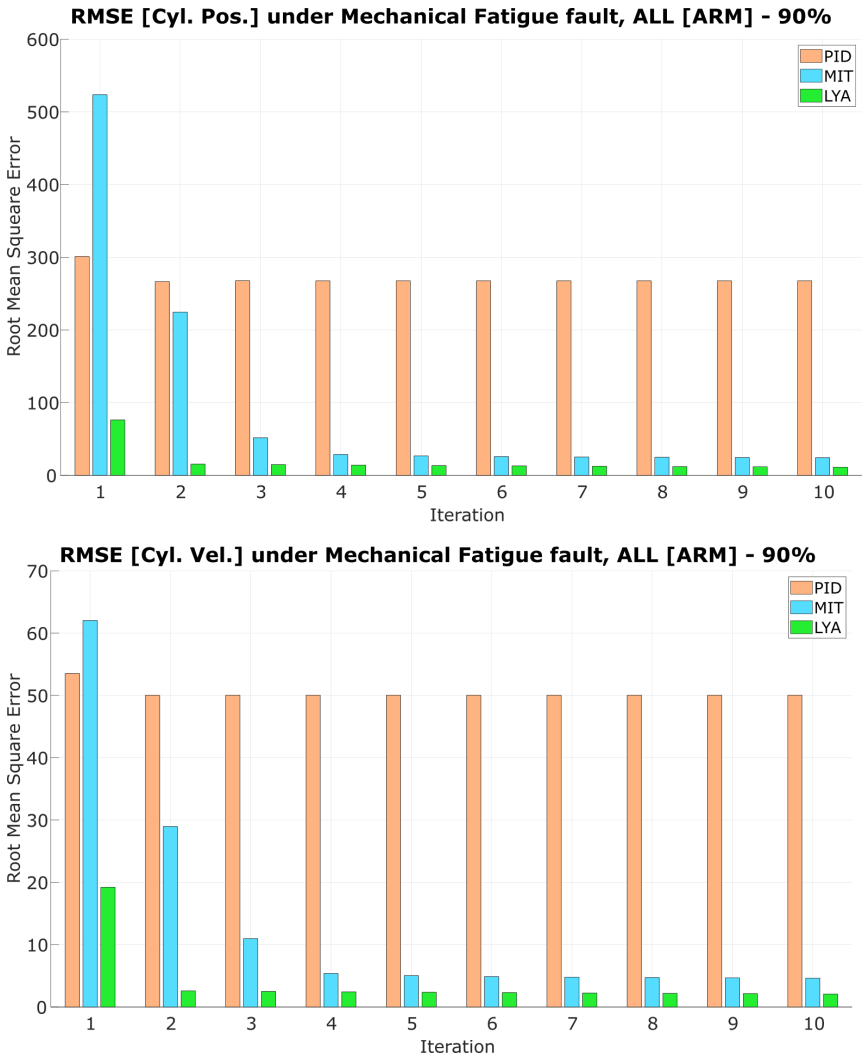


Figure 4.83: Root Mean Square Error between real system and model in a *Mechanical Fatigue* fault for all the controllers, performance against a 90% fault.

PID controllers are unable to maintain the production when the mechanical fatigue has neglected a 90% of the proportional valve opening (see Section 4.84). The controller ceases its service, as the control action performs below the optimal range. Due to the opening reduction, the controller requires more time to accomplish the Hydraulic-Press cycle.

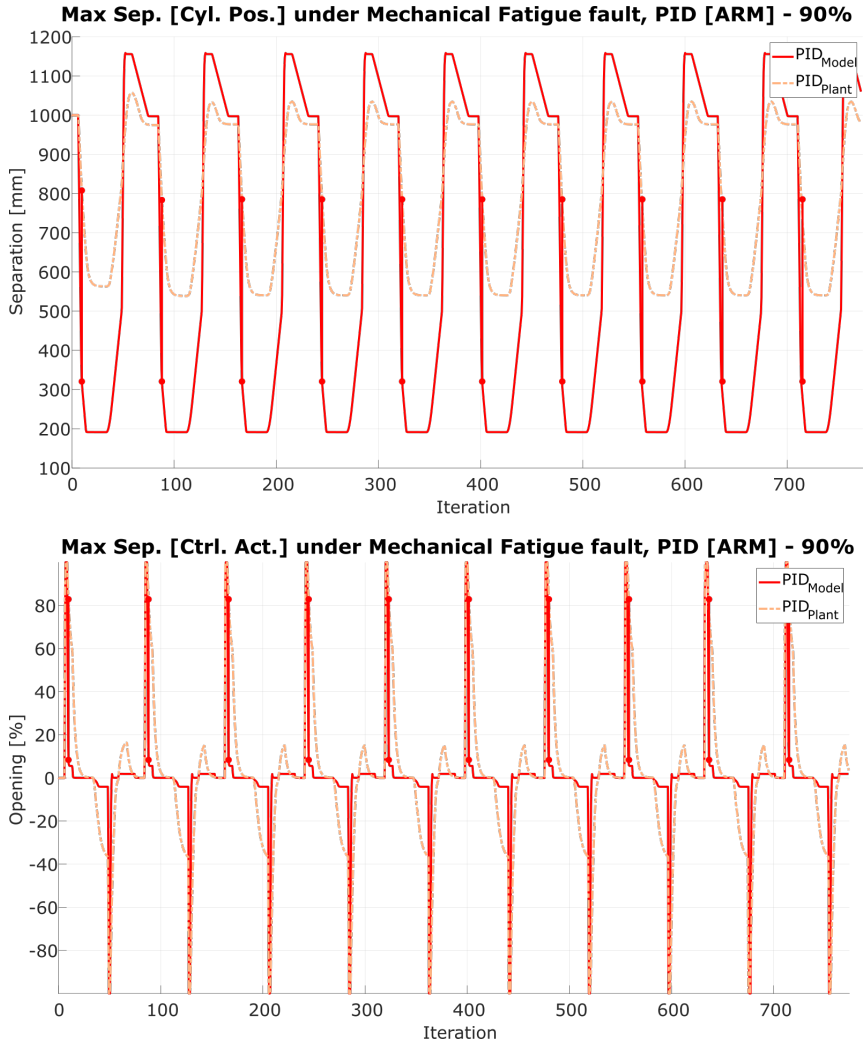


Figure 4.84: Maximum separation between real system and model in a *Mechanical Fatigue* fault for the PID controller improved with the Adaptive Reference Model, performance against a 90% fault.

After the fault emergence, the PID controller attempts to recover the signal, as the error between model and plant responses has a slight reduction (see Fig. 4.85). Although the controller maintains the stability, it is incapable of regaining the production during the subsequent iterations. As the proportional valve has reduced its opening, the piston rod maximum velocity has also diminished.

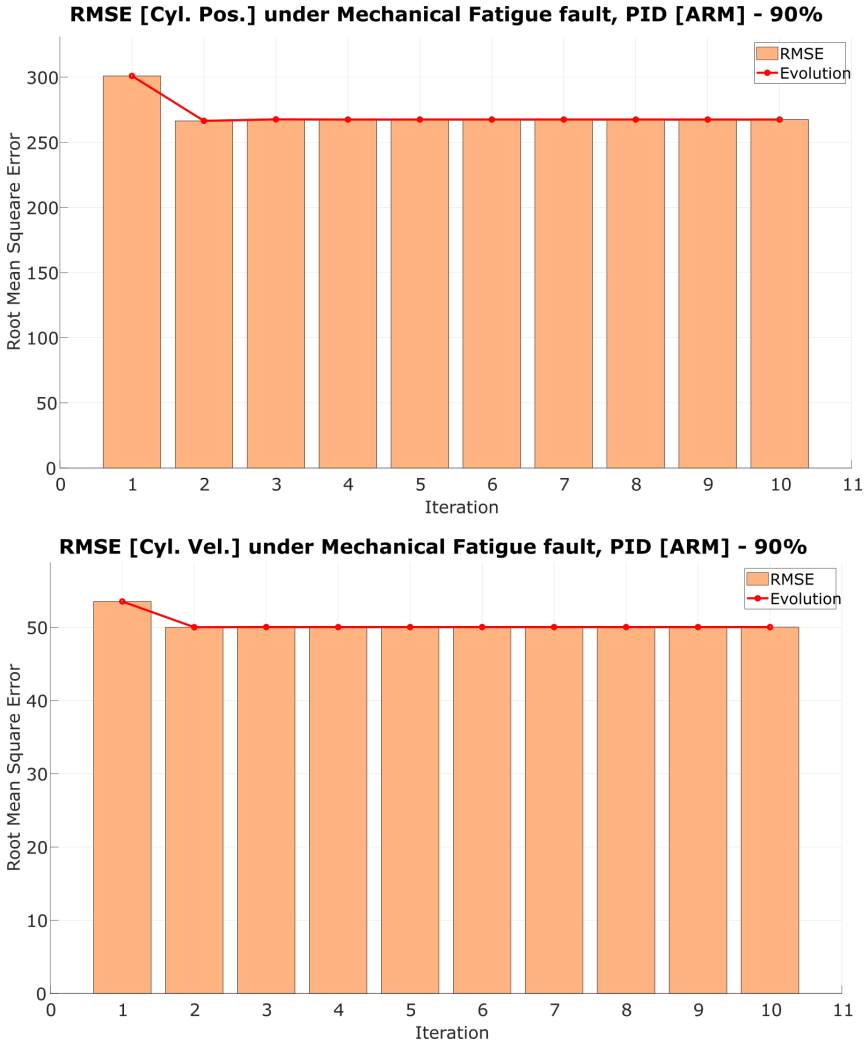


Figure 4.85: Root Mean Square Error between real system and model in a *Mechanical Fatigue* fault for the PID controller improved with the Adaptive Reference Model, performance against a 90% fault.

The novel model reference adaptive technique introduced solves the problems presented in the PID case, as the MIT rule algorithm study shows (see Fig. 4.86). During the first iteration, the controller opens the proportional valve below the optimal performance, neglecting the production. After this cycle, the adaptation mechanism alters the model accordingly to the new fault limits, recovering the nominal behaviour partially.

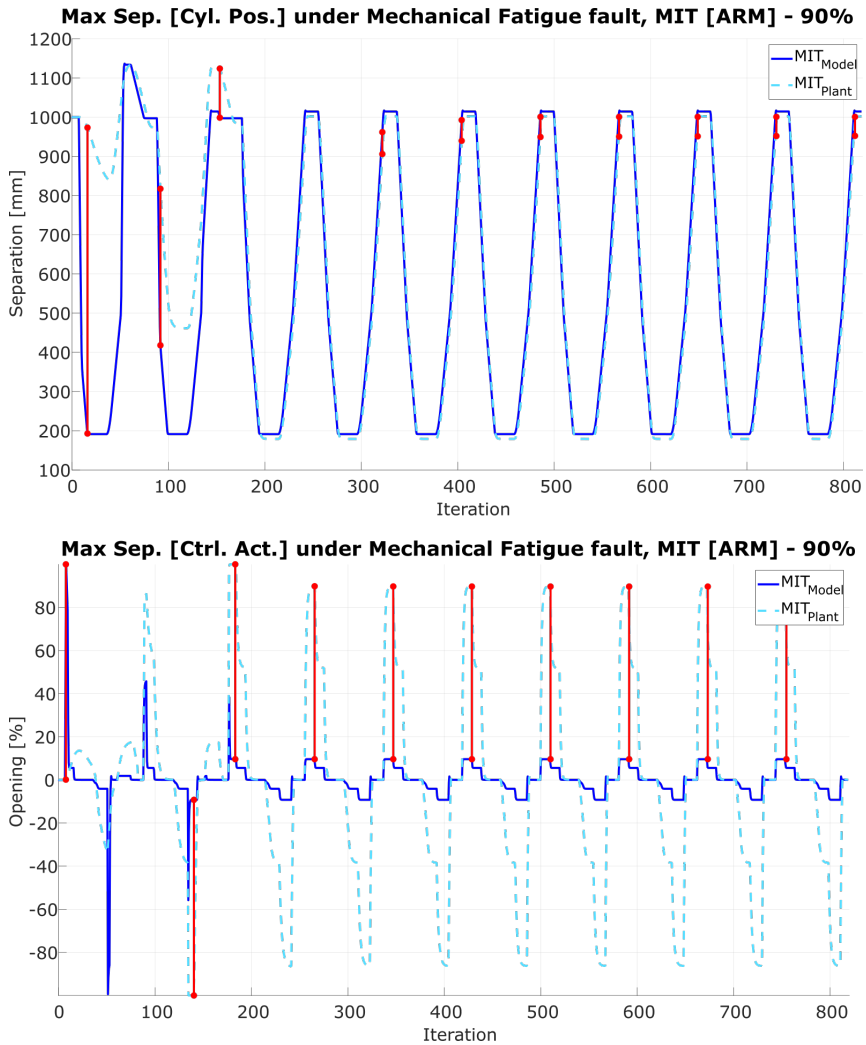


Figure 4.86: Maximum separation between real system and model in a *Mechanical Fatigue* fault for the MIT controller improved with the Adaptive Reference Model, performance against a 90% fault.

During the RMSE study, the controller shows this gradual adaptation process (see Fig. 4.87). The cylinder position and velocity continually reduce the error between model and plant signals. After the sixth cycle, the adaptive gains have obtained the stability, entering into a minimum local region and ending the process.

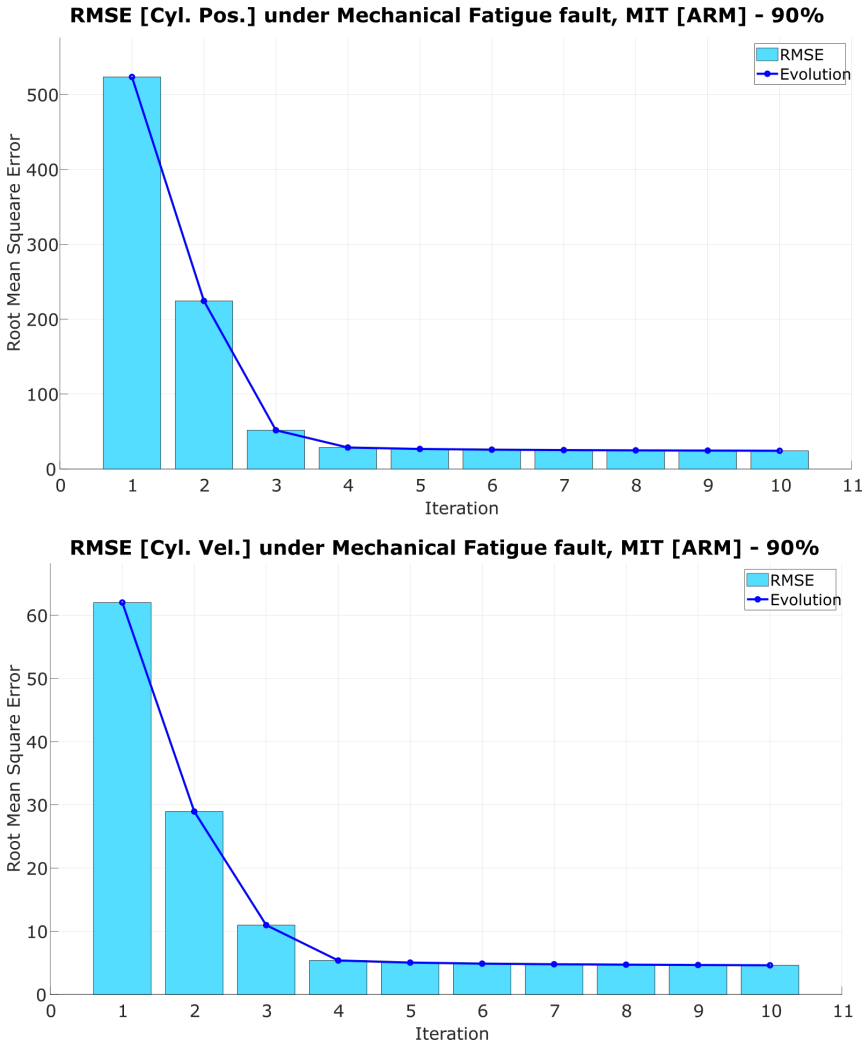


Figure 4.87: Root Mean Square Error between real system and model in a *Mechanical Fatigue* fault for the MIT controller improved with the Adaptive Reference Model, performance against a 90% fault.

Lyapunov rule controllers present an even better response than their MIT counterpart (see Fig. 4.88). Although they also have a significant error during the first iteration, they initiate earlier the adaptation mechanism reaching the global minimum after a few iterations. The control action adjusts their performance accordingly to the fault, altering the position response to ensure the system stability.

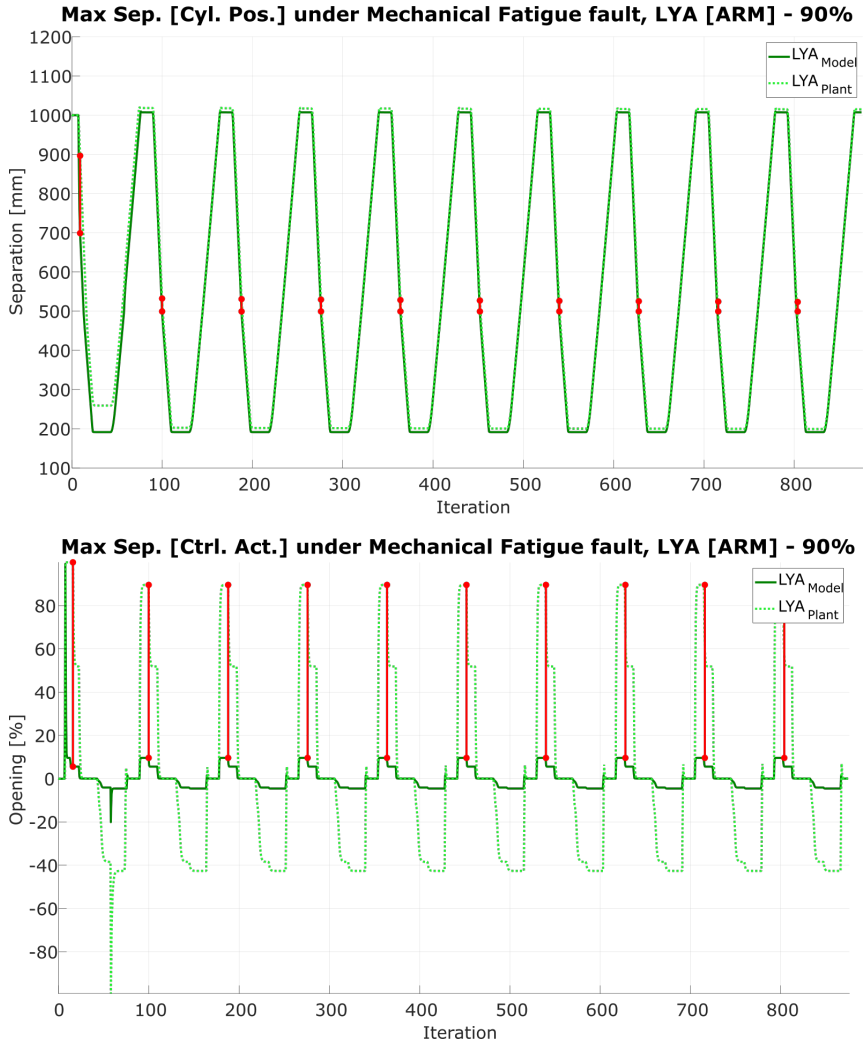


Figure 4.88: Maximum separation between real system and model in a *Mechanical Fatigue* fault for the Lyapunov controller, performance against a 90% fault.

The cylinder position and velocity RMSE graphs show the progressive adaptation process (see Fig. 4.89). They perform better than the MIT counterpart, but after ten cycles, they are still adapting the controller. They require more iterations to reach stability, as there is two adaptation process involved, for the control algorithm and the model reference stage.

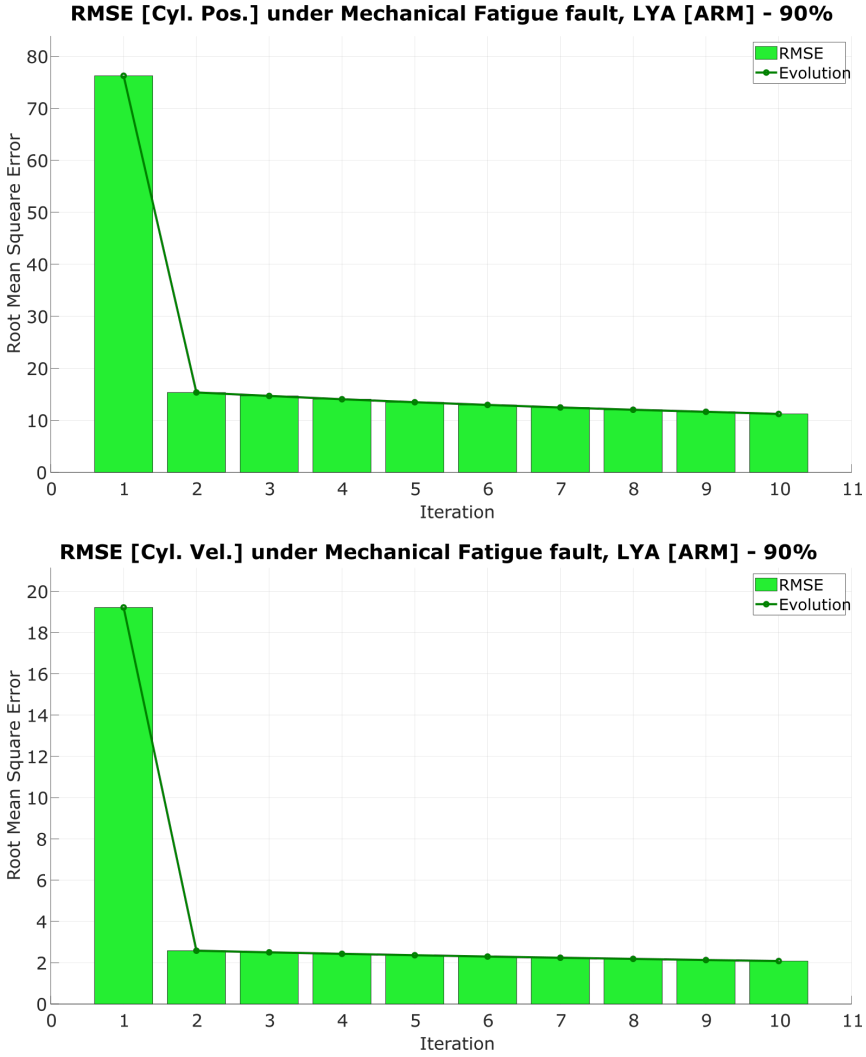


Figure 4.89: Root Mean Square Error between real system and model in a *Mechanical Fatigue* fault for the Lyapunov controller, performance against a 90% fault.

This study corroborates the hypothesis early stated, include adaptive algorithms into the model reference stage improves the system responses. They adapt the Digital-Twin control algorithm accordingly to the plant limits after the fault, ensuring the system stability. These mitigate the error during each iteration without ceasing the production (see Table 4.4).

$FAULT$		$MechanicalFatigue$					
$VALUE$		10%					
$CONTROL$		PID		MIT		LYA	
$CYCLE$		1^{st}	10^{th}	1^{st}	10^{th}	1^{st}	10^{th}
CP	MAX	193.77	194.66	599.47	9.39	11.80	0.24
	$RMSE$	23.10	21.47	381.06	1.73	8.84	0.07
CA	MAX	90.97	91.06	99.87	37.30	46.25	37.12
	$RMSE$	30.06	28.10	54.20	1.38	3.57	0.10
$VALUE$		90%					
$CONTROL$		PID		MIT		LYA	
$CYCLE$		1^{st}	10^{th}	1^{st}	10^{th}	1^{st}	10^{th}
CP	MAX	487.40	464.60	780.46	48.52	197.62	24.14
	$RMSE$	300.98	267.52	523.54	24.24	76.25	11.22
CA	MAX	74.51	74.51	99.87	80.07	94.45	79.97
	$RMSE$	53.53	50.03	62.02	4.62	19.22	2.08

Table 4.4: Maximum separation and RMSE between the desired signal and the AC for *Mechanical Fatigue*.

The methodology ensures the system stability and recovers the production even when the fault harm grade would completely negate it. However, the adaptive algorithm accommodates the reference signal, disabling the controller positions partially. The methodology has proven to avoid production interruptions, but the machine require maintenance to recover completely the optimal performance.

4.3 Control Redesign: Advanced Manufacturing Techniques

This thesis has proposed three enhances over conventional Model Reference Adaptive Controllers to increase their feasibility for the industrial environment. The reliability attained after the improvements ensure that MRACs are suitable for the Control Redesign phase of FTCs. When a fault arises, the controller maintains the system stable while the adaptive gains adjust its signal accordingly to the latest plant dynamics. After the redesign stage, the system surpasses the fault without interrupting the production, maintaining an optimal performance until the maintenance period.

The following pages complement the theoretical research done in the previous chapter (see Chapter 3.3). During this section, the MRAC fully enhanced performance has been presented. The Model Reference stage introduces a Digital-Twin model replicating the plant responses. The Adaptation Mechanism stage based their algorithm on an MRAC, optimising its performance for the industrial environment through a reference adaptation mechanism. The Adjustable Controller substitutes the traditional PID architecture for an **upgradable** platform based on a Bank of Controllers.

Instead of a simplistic Bank of Controllers architecture, this thesis proposes an architecture that comprehends the MRAC (see Fig. 4.90). During the commissioning stage, control designers investigate the faults altering the machine behaviour significantly, storing its performance under this drawback in a fault database. Afterwards, they train the NN through this information, obtaining an FDI mechanism that detects, measures and isolates the fault source. In parallel to this operation, control designers tune the adaptive gains to surpass faults more efficiently.

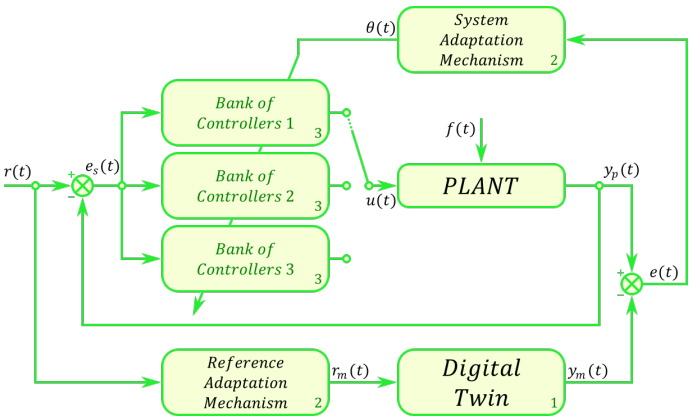


Figure 4.90: Schematic view of the novel MRAC features: Advanced Manufacturing Techniques.

The information gathered during the Fault Detection and Isolation phase discriminates the fault case neglecting the machine behaviour, adjusting the MRAC adaptive gains accordingly to this drawback. The BCs switches between their pre-

stored values attending to the current fault case. Due to this process, the controller manages the plant with an updated version of its algorithm.

Across this section, the Hydraulic-Press performance under several faults has been presented(see Section 4.3.1). The system performs a controlled amount of cycles under the drawbacks studied in this thesis (see Append B), switching between the control architectures studied (see Section 4.2). After validating their behaviour, the HP controller has been updated with the novel algorithm and subjects its performance against two experiments, altering the fault case and modifying the cycle duration, respectively (see Section 4.3.2).

4.3.1 Controller Performance

This round of experiments validates the controller performance simulating the Hydraulic-Press behaviour against several cycles. Similarly to previous studies (see Section 4.2.1, 4.2.2, 4.2.3 and 4.2.4), the responses obtained in the system when its controller varies between four algorithms has been investigated. The initial structure corresponds to a PID architecture identical to the original machine controller. After analysing the responses, the subsequent study presents the improvements brought by the Adaptive Controllers. There are two structures involved in this research, MIT or Lyapunov rule controllers. On the last study, instead of the conventional MRAC based on Lyapunov rules, their enhanced version has been introduced. This controller has upgraded the MRAC model reference stage with an adaptive structure (see Section 3.2.6).

Across these experiments, instead of studying the system responses against a single fault, the machine undergo against a distinct disadvantage every five iterations. As there has been studied four situations, the research reproduces the system performance for twenty cycles, distributing the mistakes equally between the faultless case, the actuator mechanical fatigue, the cylinder chambers internal leaks and the sensor noise dissonance.

The responses represent the current machine behaviour under the forementioned fault cases when the control algorithms belong to a PID, MIT, Lyapunov or Adaptive Reference Model architecture. Each controller performance has been already validated through its experiment, previously presented in this thesis. They reproduce the responses for cylinder position and velocity in conjunction with the control action consumed. In addition to the graphical responses, the study also displays the RMSE value accumulated in each iteration.

4.3.1.1 PID Controller

As previous section has introduced, the first round of experiments belongs to the PID architecture. This controller has similar gains as the original control algorithm designed by operators during the commissioning process, obtaining similar responses as the machine while the system remains faultless. This investigation studies how faults degrade the performance after their emergence.

The Hydraulic-Press position responses confirm the PID controller inefficiency to continue tracking references when faults emerge. During the nominal behaviour, the plant matches the ideal responses, whereas, throughout the actuator faults, the RMSE has an increasing tendency. For cylinder leaks fault, the system preserves a constant RMSE, nullifying it again when the fault switches to a noise dissonance in the sensor (see Section 4.91).

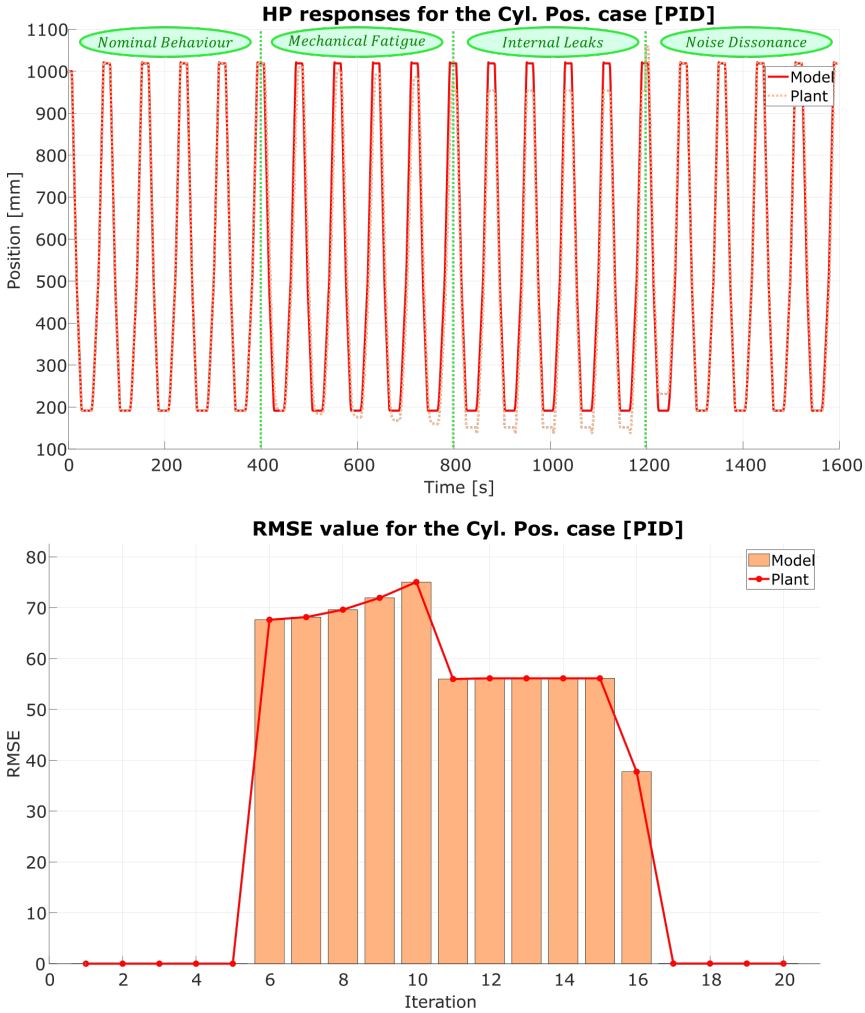


Figure 4.91: Cylinder Position and Root Mean Square Error between model and plant for several faults in the PID controller, performance against a 90% fault.

As position and velocity are variables linked through its derivative, this secondary analysis resembles the previous example. The system responds correctly in the absence of faults or when they emerge on the sensor, avoiding the noise disturbances generated. Actuator faults produce a situation where the system ceases following the reference, a behaviour shared with the cylinder chambers internal joins faults (see Section 4.92).

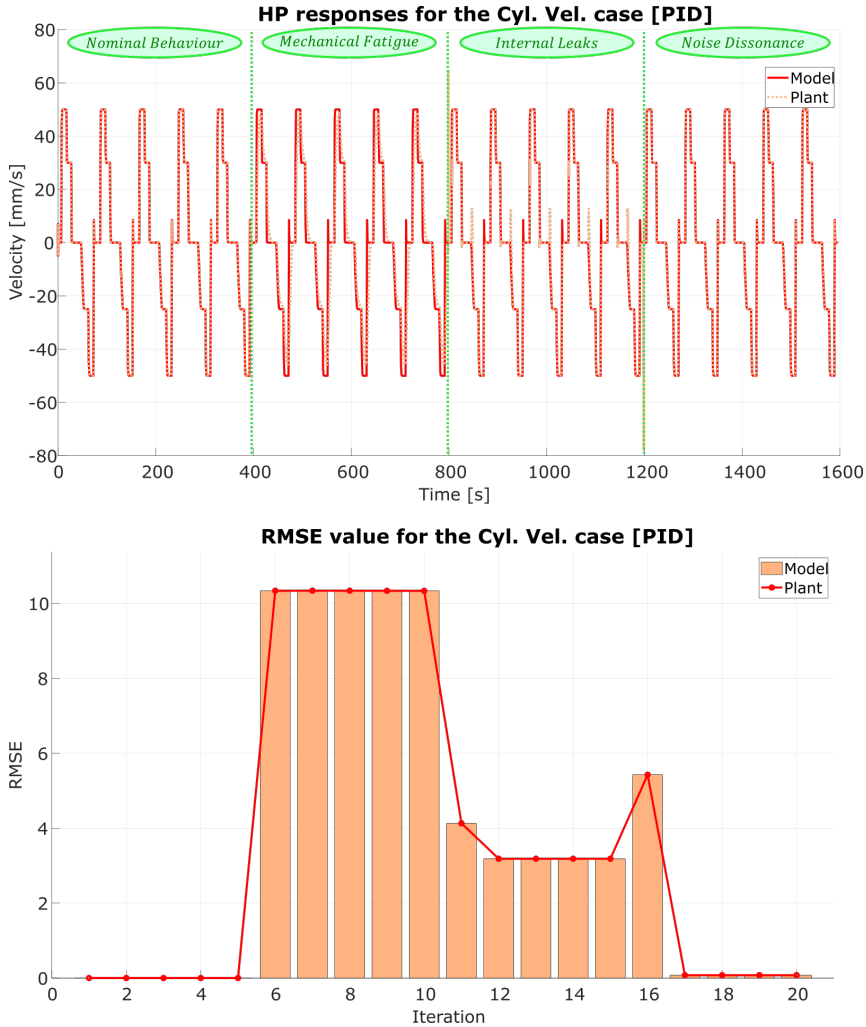


Figure 4.92: Cylinder Velocity and Root Mean Square Error between model and plant for several faults in the PID controller, performance against a 90% fault.

These control action experiments validate the previous assumptions, as the controller is unable to recover an optimal performance when the fault alters the actuator behaviour or the plant dynamics. They have a similar error response against the multiple iterations, that leads to a dissonance on the position and velocity signals (see Fig. 4.93).

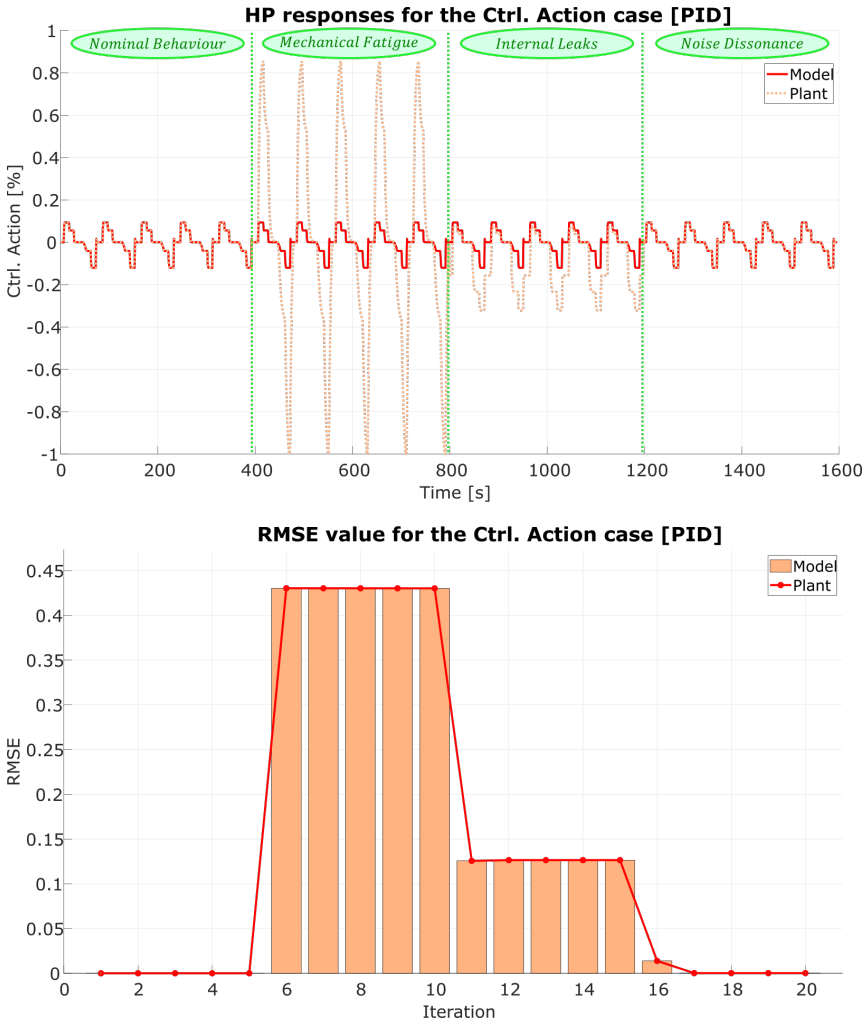


Figure 4.93: Control Action and Root Mean Square Error between model and plant for several faults in the PID controller, performance against a 90% fault.

Although the original controller maintains the system stable for the faultless situation, they only recover an optimal performance for noise dissonance faults. This behaviour corroborates that despite these controllers are widely extended on the industry; they are unable to regain the optimal performance for soft faults.

4.3.1.2 MIT Rule Controller

The research continues through the Adaptive Controllers. This section presents the system performance when the controller has an MRAC based on the MIT rule. They recover an optimal performance in several fault situations; however, they are prone to reach a local minimum instead of the global (see Section 4.2).

As expected, the Hydraulic-Press position response corroborates the local minimum dilemma. During the faultless behaviour, the controller encounters a minimum position on the second iteration, but the RMSE still increases on the subsequent cycles. This adverse influence also neglects the optimal behaviour for internal leaks and noise dissonance faults (see Fig. 4.94).

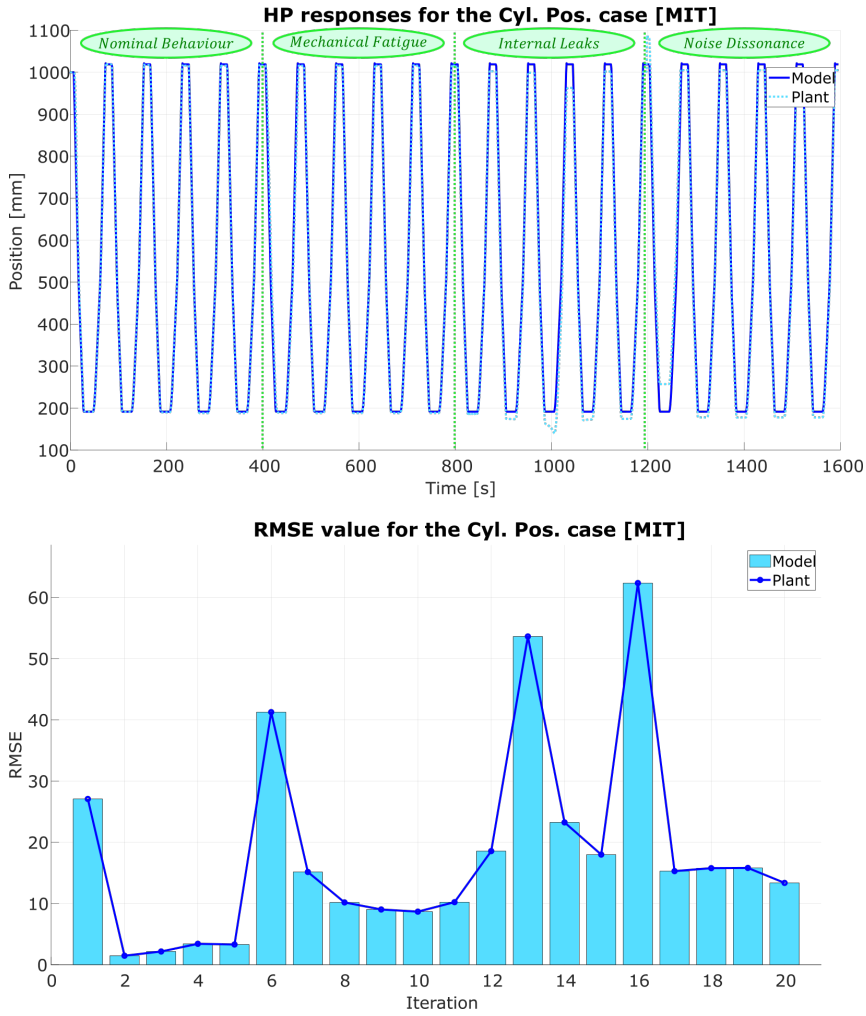


Figure 4.94: Cylinder Position and Root Mean Square Error between model and plant for several faults in the MIT rule controller, performance against a 90% fault.

The cylinder velocity responses show the instability produced by the local minimum (see Fig. 4.95). During the first iteration, the system reaches its maximum RMSE error, but it also presents an oscillatory behaviour. This erratic performance remains for the noise dissonance fault, as the controller attempts to encounter the stability unsuccessfully.

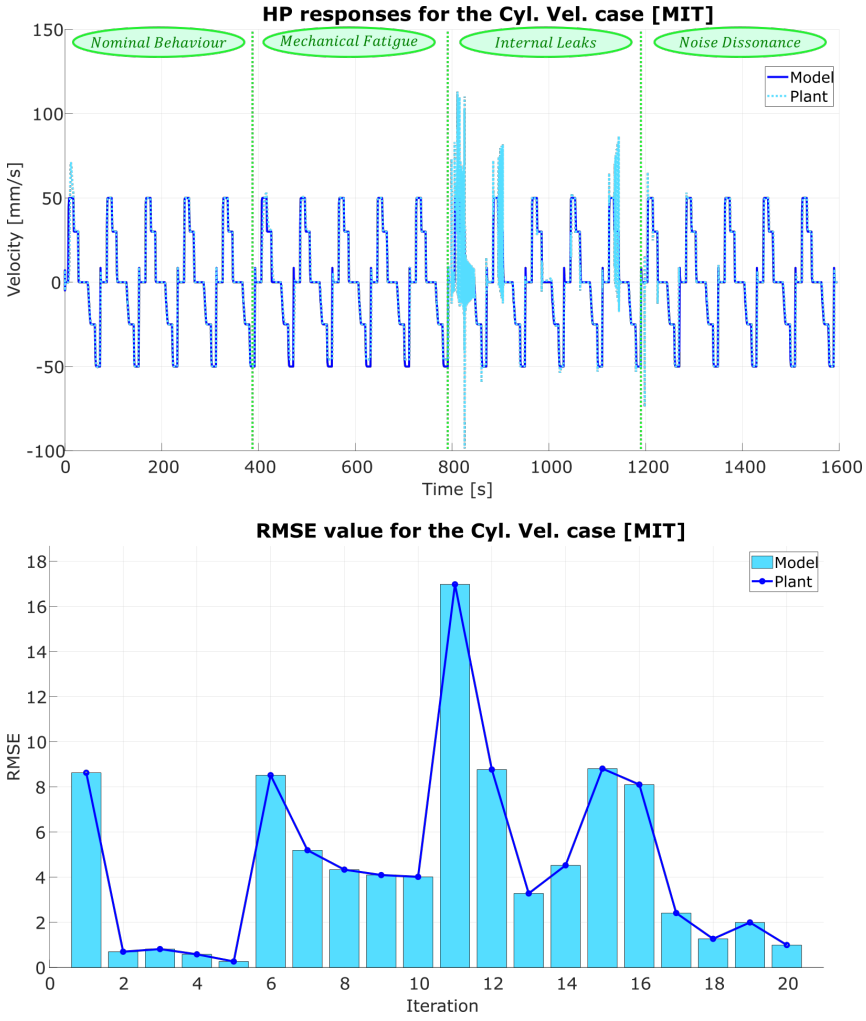


Figure 4.95: Cylinder Velocity and Root Mean Square Error between model and plant for several faults in the MIT rule controller, performance against a 90% fault.

Despite the instabilities, there are faults where MRAC based on MIT rule encounter the global minimum, for instance, in the actuator mechanical fatigue case. The source of this outcome appears in the controller action response, as the signals are identical to the original behaviour, but velocity and position graphs are distant to the model (see Fig. 4.95).

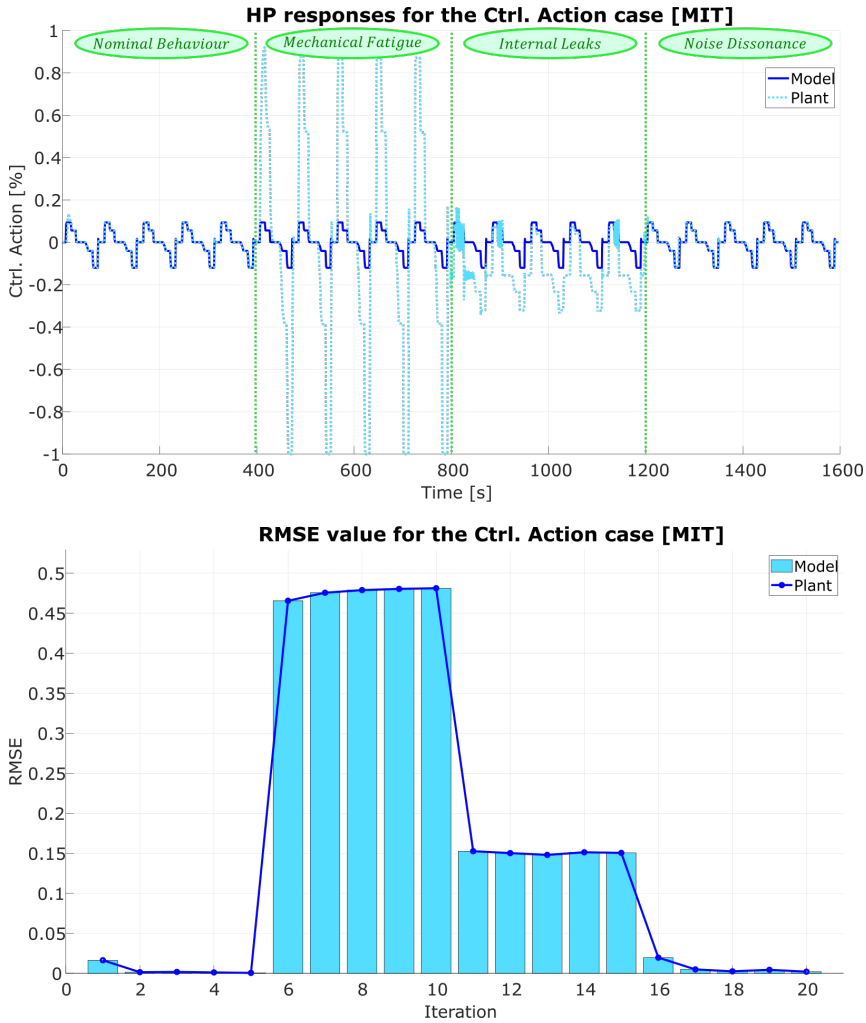


Figure 4.96: Control Action and Root Mean Square Error between model and plant for several faults in the MIT rule controller, performance against a 90% fault.

MRACs based on MIT rule regain the optimal behaviour when they encounter global minimums; however, they are prone to enter a local situation where the system becomes unstable. This lack of robustness reduces their applicability into the industrial environment, as they spread the fault and endanger operators.

4.3.1.3 Lyapunov Rule Controller

After studying the first adaptive controller, the research analyses the performance of MRACs based on Lyapunov rules. These controllers optimise an equation representing the system adaptability potential, reaching global minimums rather than local ones. Nonetheless, due to this behaviour, the controller adapts the signal continuously until nullifying the error between model and plant responses.

The Hydraulic-Press responds to the Lyapunov controller appropriately when there are disturbances due to the mechanical fatigue or the noise dissonance fault, but they have suboptimal performances for the internal leaks incident (see Fig. 4.97).

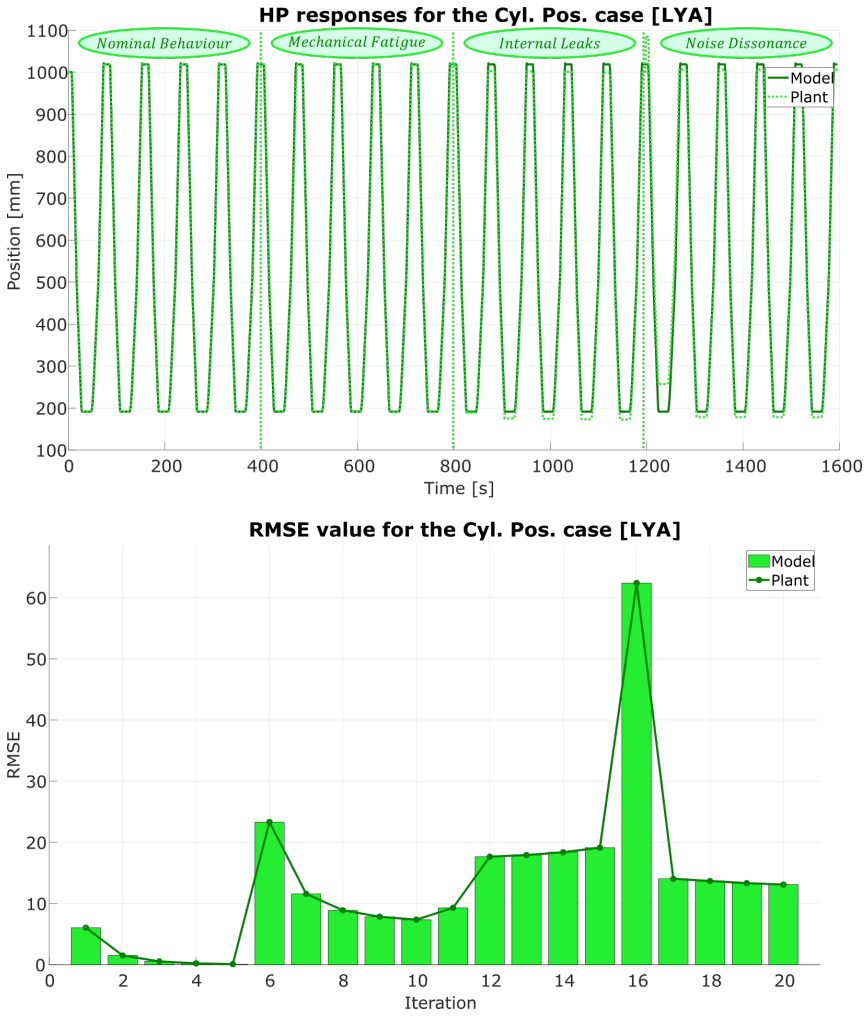


Figure 4.97: Cylinder Position and Root Mean Square Error between model and plant for several faults in the Lyapunov rule controller, performance against a 90% fault.

The cylinder velocity study corroborates this assumption, showing clearly the disturbances produced on the system when the internal leaks fault emerge (see Fig. 4.98). Despite the initial instability, the adaptation mechanism progress until reaching a stable solution. Around the fourth iteration, the MRAC regains the optimal performance, decreasing the error between plant and model signals continuously.

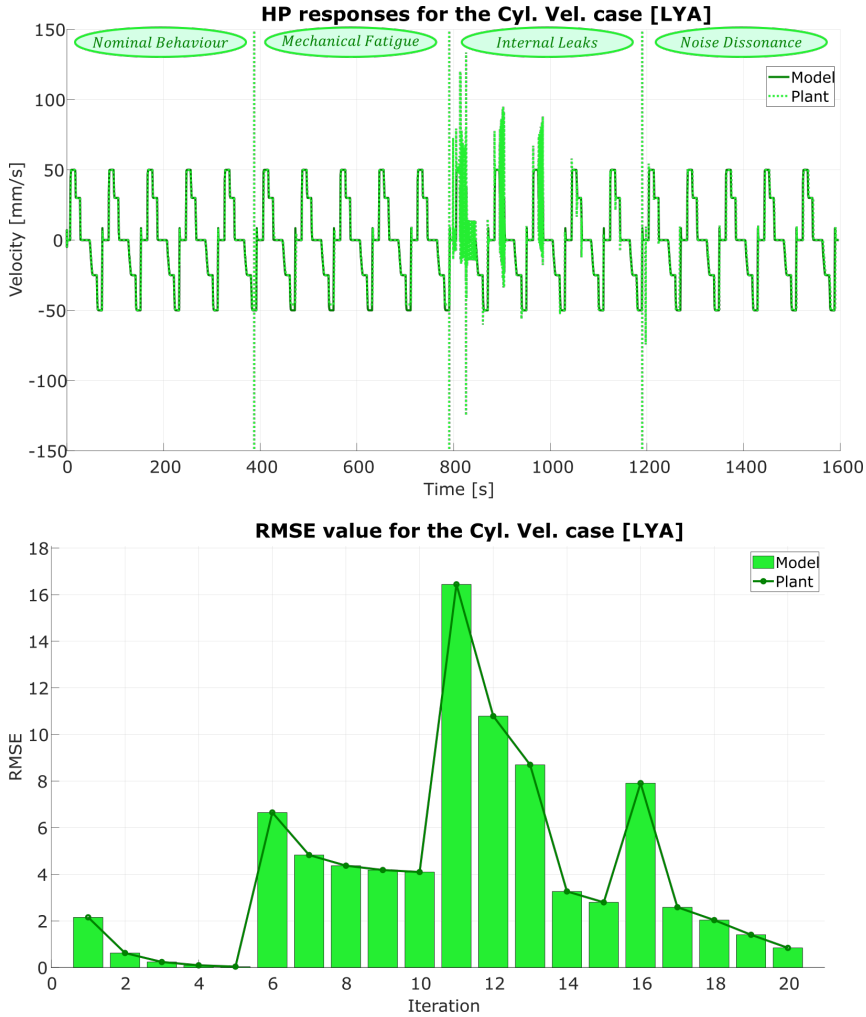


Figure 4.98: Cylinder Velocity and Root Mean Square Error between model and plant for several faults in the Lyapunov rule controller, performance against a 90% fault.

Similarly to the previous diagrams, the controller has an unstable signal when the internal leaks fault emerge (see Fig. 4.99). Due to the effect of the adaptation mechanism, the system regains the stability.

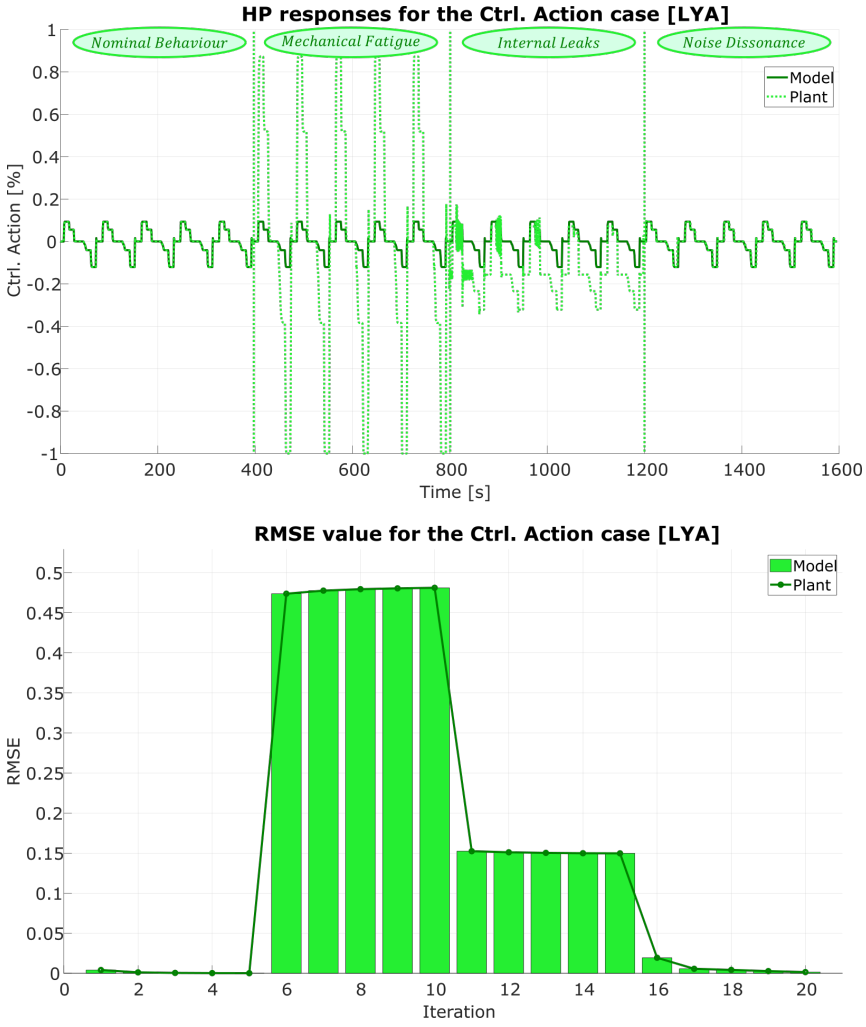


Figure 4.99: Control Action and Root Mean Square Error between model and plant for several faults in the Lyapunov rule controller, performance against a 90% fault.

MRACs based on Lyapunov rule regain the optimal performance adapting the controller signal during each iteration. However, there are instances where the system reaches unstable positions, such as the internal leaks fault. When this disadvantageous situation emerges, the controller requires some cycles to regain the desired behaviour.

4.3.1.4 Adaptive Reference Model Controller

Lastly, the conventional MRAC based on Lyapunov rules structure has been upgraded through an Adaptive Reference Model. This architecture offers better responses than their traditional counterpart, as the controller adapts the Digital-Twin accordingly to the plant mechanical limits under the fault influence.

Similarly to previous cases, the first study introduces the cylinder position across each fault (see Fig. 4.100). For each fault, the system presents a decreasing tendency as the adaptation gains reach their optimal value. Emphasise the point that the system remains stable during the simulation.

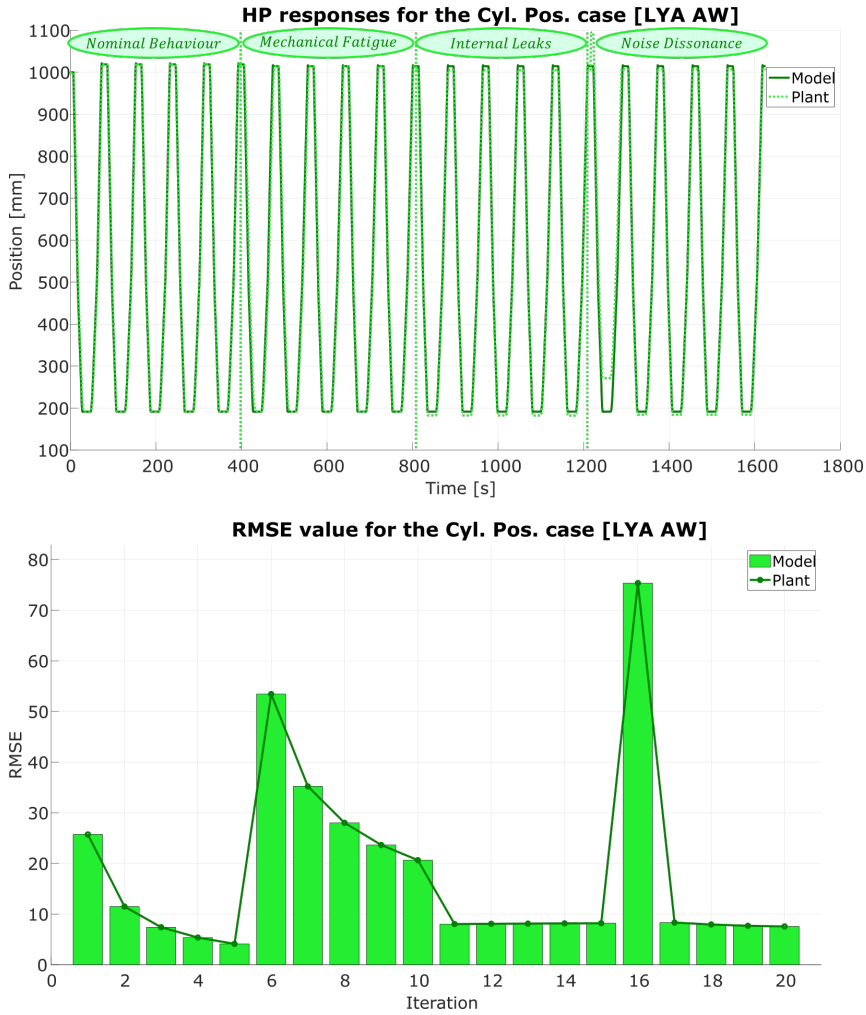


Figure 4.100: Cylinder Position and Root Mean Square Error between model and plant for several faults in the Lyapunov rule controller with Adaptive Model Reference, performance against a 90% fault.

In contrast to the conventional Lyapunov rule structure, the improvements maintain the cylinder velocity stable during the experiment (see Fig. 4.101). MRACs typical decreasing behaviour is appreciated for each fault situation, reducing the error between model and plant responses continuously. During the experiment, the production remains constant, without interrupting it despite multiple faults have emerged.

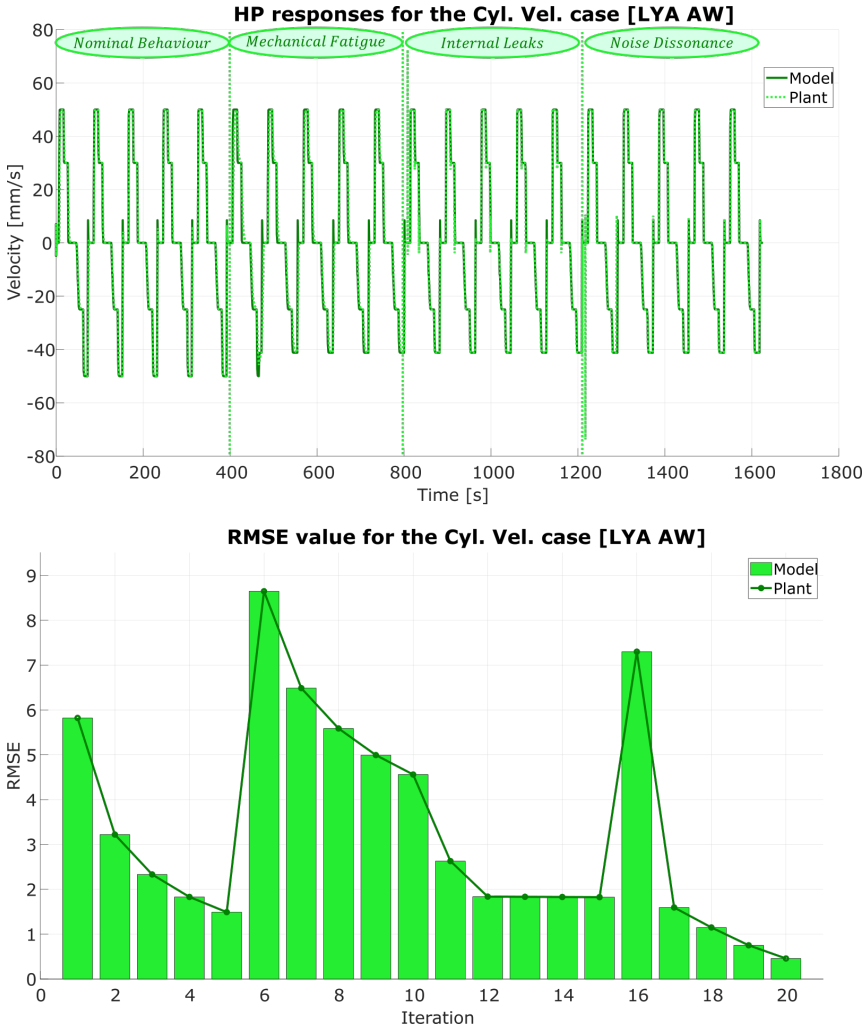


Figure 4.101: Cylinder Velocity and Root Mean Square Error between model and plant for several faults in the Lyapunov rule controller with Adaptive Model Reference, performance against a 90% fault.

The controller action maintains the previously described tendency. During the experiment, the controller adjusts its signal automatically accordingly to the fault case, minimising the adverse impact over the system, recovering the optimal performance (see Fig. 4.102).

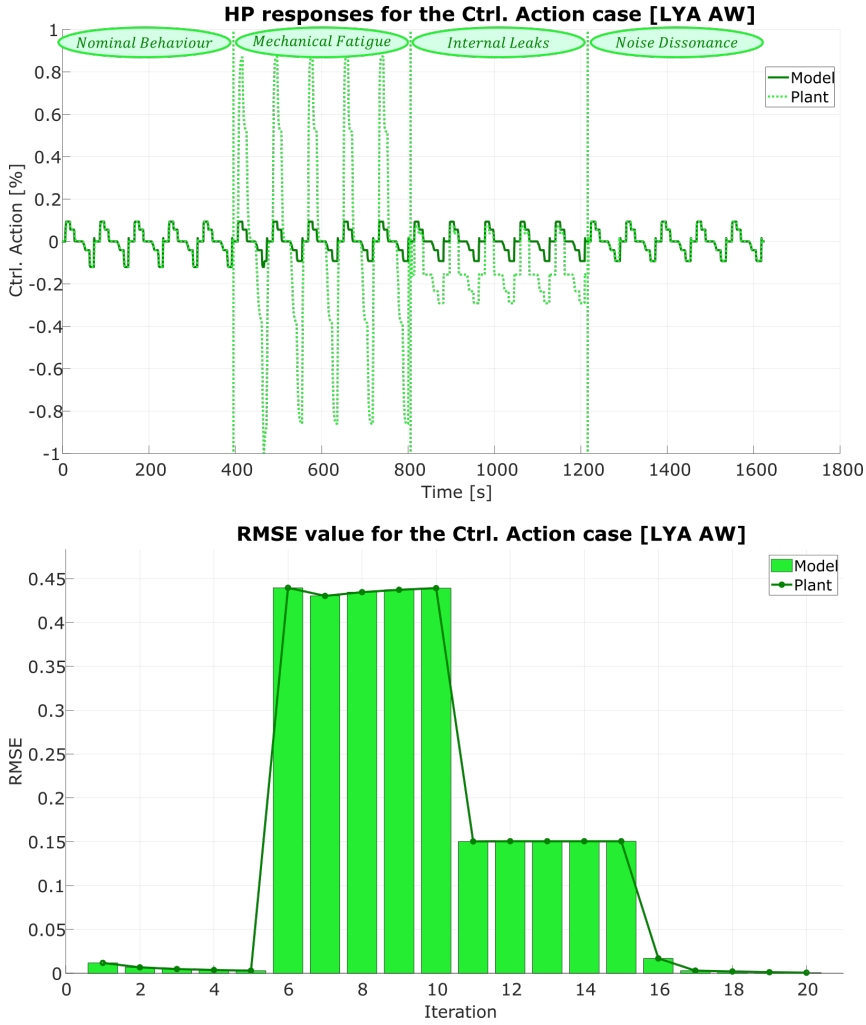


Figure 4.102: Control Action and Root Mean Square Error between model and plant for several faults in the Lyapunov rule controller with Adaptive Model Reference, performance against a 90% fault.

The enhances introduced into the conventional MRAC structure improves the responses against the faults, maintaining the system stable and recovering the optimal performance more efficiently.

4.3.2 Enhanced MRAC Validation

Through the previous experiments, the Hydraulic-Press performance has been studied when several faults emerge at distinct timestamps (see Section 4.3.1). During the individual analysis of each control algorithm, the exclusive architecture that maintains the system stable for internal leaks is the MRAC based on Lyapunov rules enhanced with an Adaptive Reference Model.

While further alternatives lead the system to instability, the novel MRAC architecture proposed in this thesis maintains the system controlled without interrupting the production. This controller has proved to optimise the performance for each circumstance, from nominal behaviour to noise dissonance fault. The controller reproduces a constant adaptation process, altering the signals accordingly to the plant dynamics.

Despite their superior behaviour against internal leaks faults, there are distinct control architectures that maintain the stability and recover the optimal performance more efficiently for different fault situations. For instance, PID controllers regain the nominal behaviour when there are noise dissonances in the sensors. Furthermore, if the system remains in a faultless condition, introducing the MRAC structure increase the control action consumption.

Due to these considerations, the latest enhanced that has been proposed for MRACs is a Bank of Controllers. This platform combines into the same closed-loop structure several control algorithms coexisting during the machine execution. Throughout the FDI phase, the system automatically detect, measure and isolate the fault source, adjusting the CR phase accordingly to this knowledge. The BCs determine the adaptive gains that recover the optimal performance to this fault situation efficiently.

Besides these improvements, the BCs platform also allows switching between control architectures. This feature provides a novel paradigm in the control design for industrial machines, implementing several architectures inside the same hardware. On the case study presented in this thesis, the MRAC based on Lyapunov rules and enhanced through the Adaptive Reference Model share the machine controllability with its original PID structure.

This BCs methodology switches between both techniques automatically attending to the machine status. For instance, if the system remains in the faultless situation, the original PID architecture control the machine. When a fault emerges, the FDI mechanism detects its source, for instance, if the fault has emerged on the actuator (mechanical fatigue) or the plant (internal leaks). Afterwards, the BCs automatically switches between their preloaded MRAC structures, selecting the algorithm that maintains the stability without interrupting the production. Similarly, if the fault belongs to a noise dissonance on the sensors, BCs adopt the best architecture, the original PID.

These hypothesis have been validated through a double experiment. During the first approach, the system maintains an identical performance in each iteration, switching between faults after ten repetitions (see Section 4.3.2.1). During the second approach, in addition to the fault emergence cadence, each window has different loads, reproducing the machine factory behaviour (see Section 4.3.2.2).

4.3.2.1 Multi-Fault Enhanced MRAC Simulation

Despite the emergence of several faults, the control algorithms maintain the production stable across all the iterations (see Fig. 4.103). There are alterations when the experiment enters the noise dissonance region, but the controller fastly recovers the optimal performance.

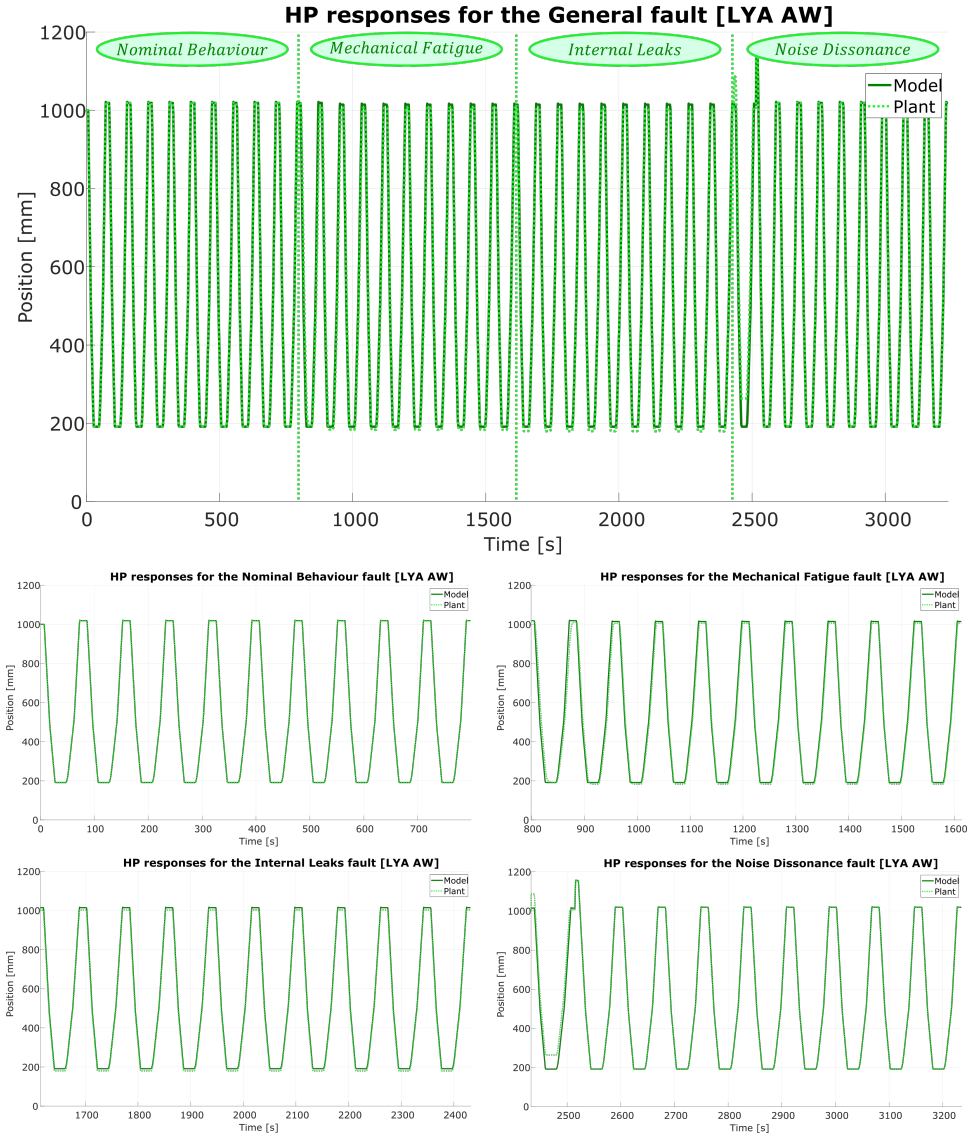


Figure 4.103: Cylinder Position between model and plant for several faults, performance against a 90% fault.

The RMSE diagrams present the decreasing nature of the MRACs (see Fig. 4.104). Across each iteration, the adaptive gains reduce the error between model and plant, nullifying it when the number of iterations tends to the infinity.

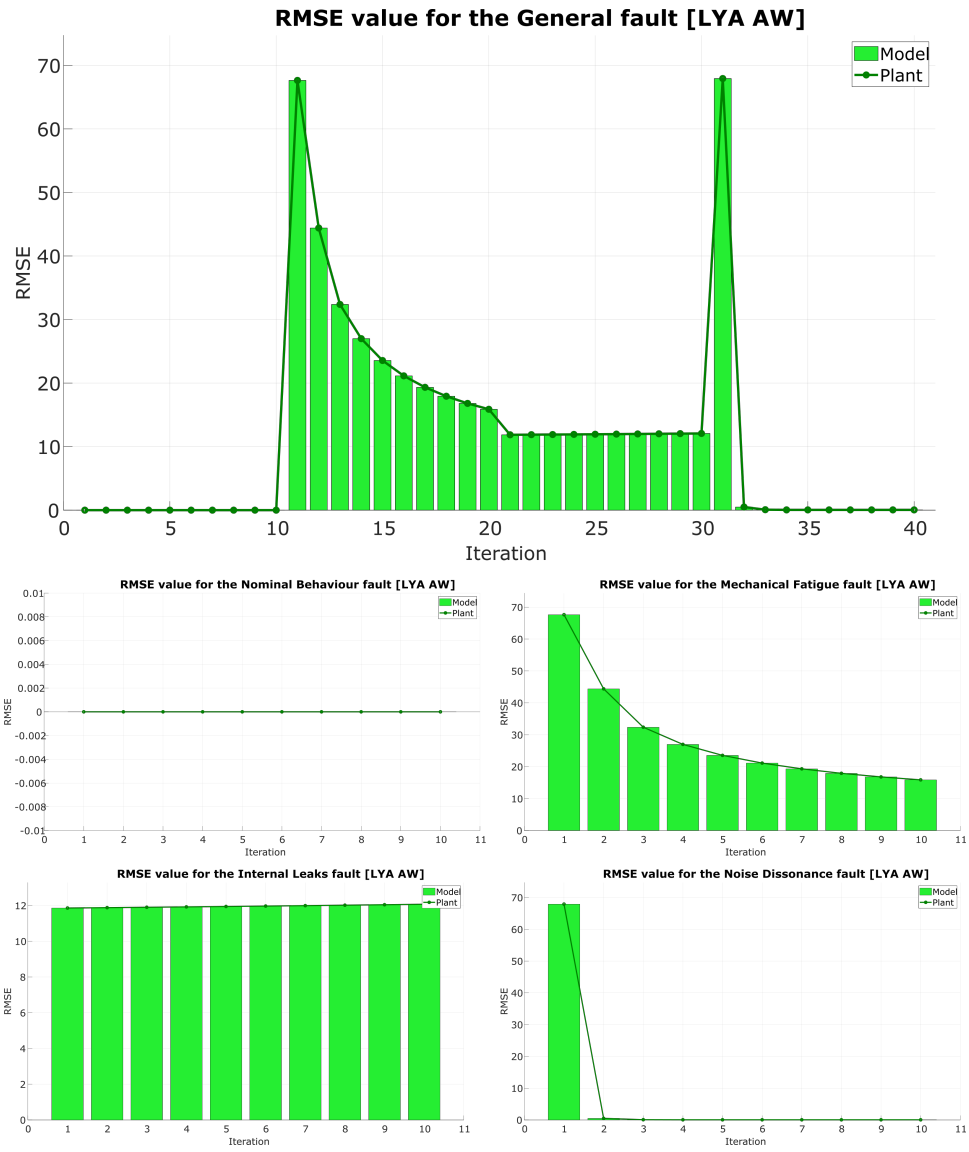


Figure 4.104: Root Mean Square Error between model and plant for several faults, performance against a 90% fault.

4.3.2.2 Multi-Load Enhanced MRAC Simulation

The research continues presenting a full simulation under factory conditions. During the experiment, the cylinder position remains unaffected by the fault emergence, maintaining a constant production (see Fig. 4.105). The Bank of Controllers adjusts the controller automatically to the most suitable solution independently of the load variations.

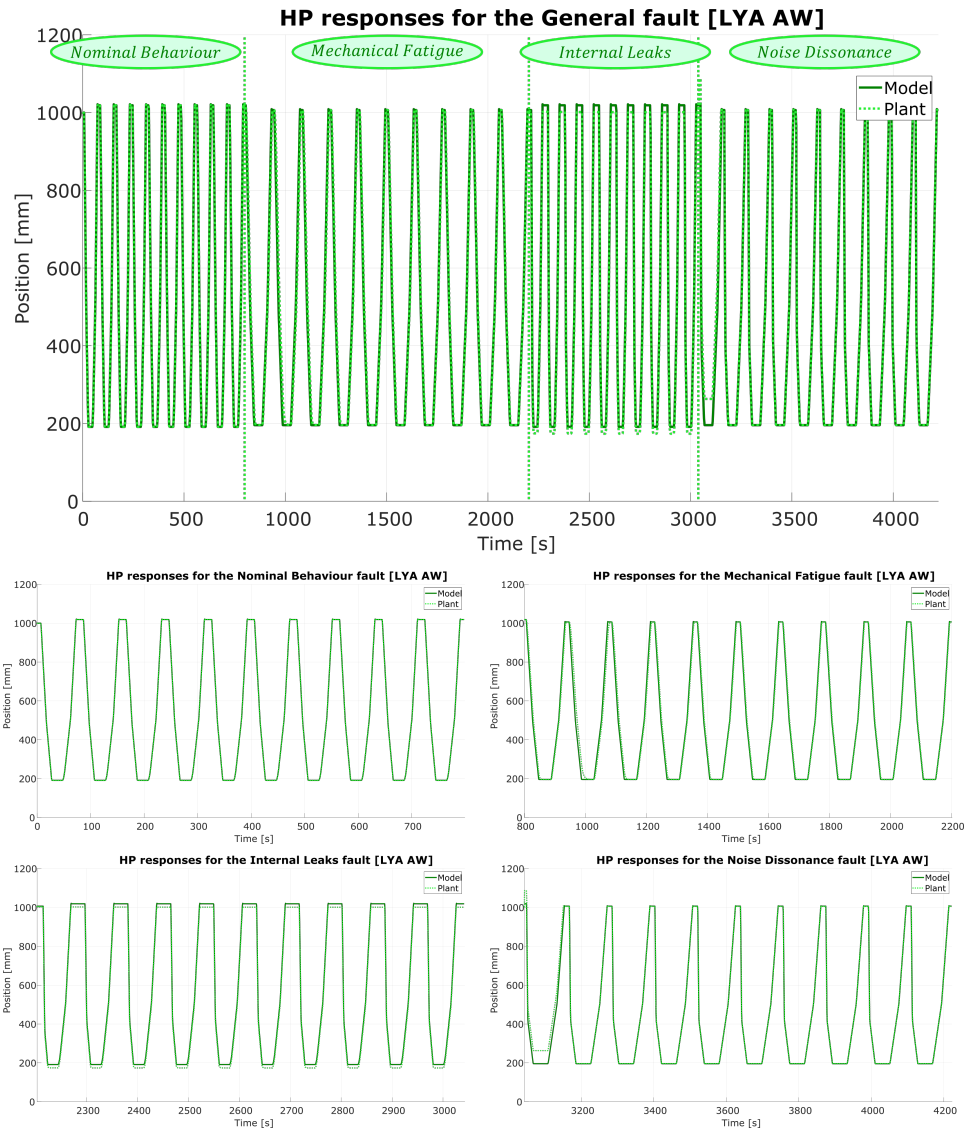


Figure 4.105: Cylinder Position between model and plant for several faults with varying loads, performance against a 90% fault.

Similarly to previous cases, the RMSE decreases as the Hydraulic-Press perform a more extensive number of iterations (see Fig. 4.106). The Bank of Controllers maintains the stability regarding there are variations in the cylinder career or adverse dynamics appear in the system. This experiment corroborates the early assumptions, as the designed platforms present a controller that surpasses faults without interrupting the production and optimised for industrial environment.

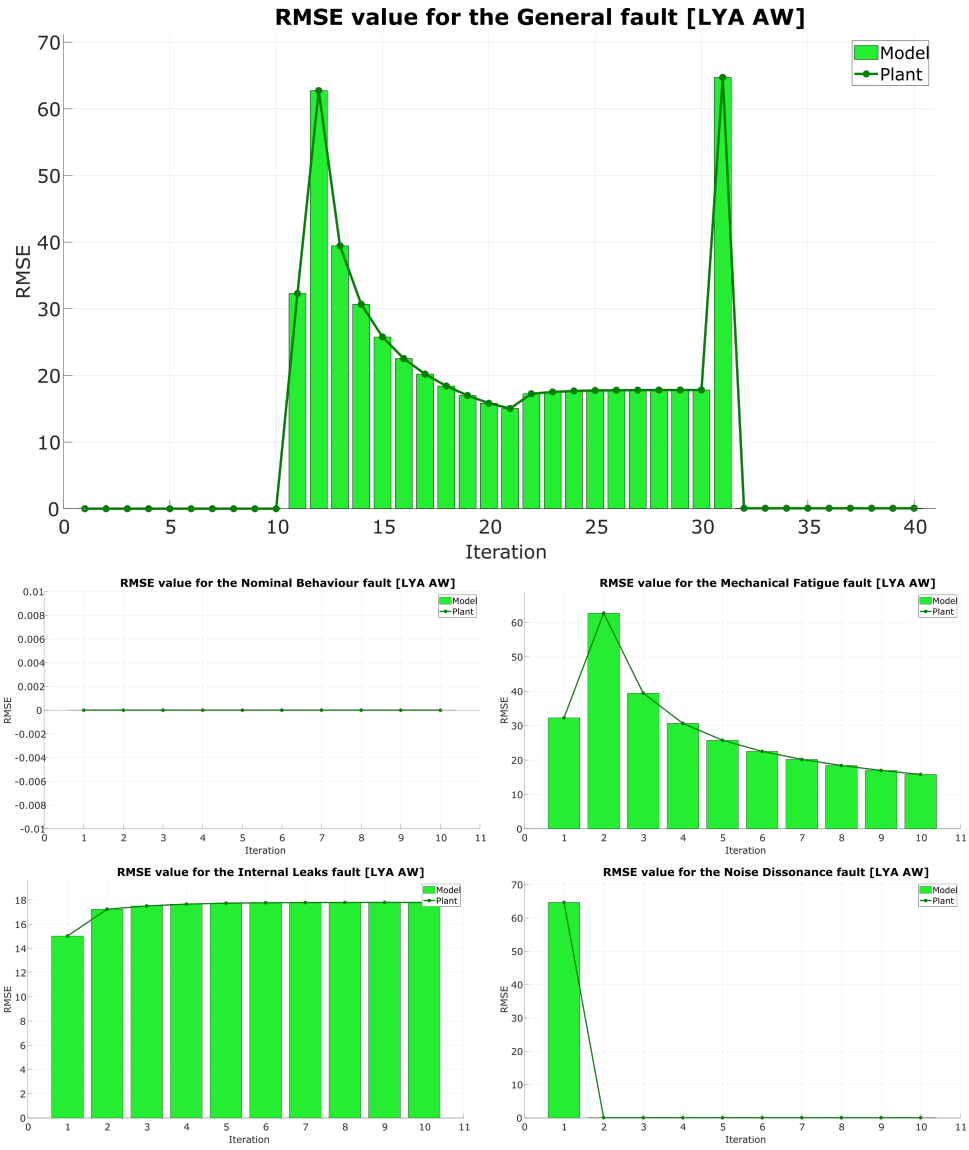
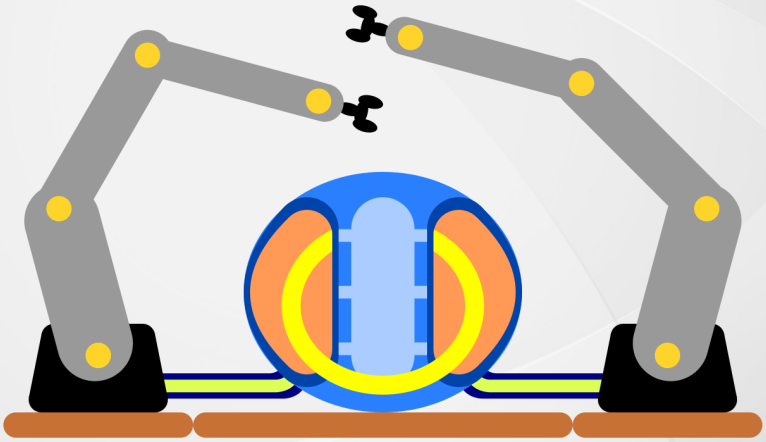


Figure 4.106: Root Mean Square Error between model and plant for several faults with varying loads, performance against a 90% fault.

5 Conclusion and Future Work



Competitions demand athletes their maximum performance. Each step determines its position on the finishing line. Maintaining the concentration in these situations is crucial, as they have to mitigate the physical and mental extenuation to continue performing at their best. Similarly, manufacturing machines are also submitted to several charges that fatigue their behaviour. In this case, the controllers behave as brains that distribute the efforts to avoid disrupting the production.

In control theory, these charges are considered as faults, the source altering the plant dynamics. Manufacturing machines have several points were these drawbacks emerge, reducing the production as they neglect the nominal system behaviour. This thesis proposes a novel Fault-Tolerant Control methodology to deal with these uncertainties, recovering an optimal performance and maintaining the production constant without human interaction.

The FTC methodology presented in this thesis belongs to the active approach, where the recovery process is divided into two phases. During the first one, called as FDI, an algorithm detects, grades and isolates the fault source, while during

the second one, called as CR, the controller actualise its signals to regain the optimal performance. These phases have their analogue in the previous example. Athletes perceive when their muscles are fatigued and adapt their performance accordingly to reduce the pressure without quitting.

This thesis has presented the improvements brought to these phases, focusing the research in the introduction of Adaptive Control techniques into the CR phase. The conclusions drawn from this investigation are presented in this section, explaining initially the contributions made to state of the art (see Section 5.1). The chapter continues showing the publications supporting the study (see Section 5.2) and explaining the future lines for the research (see Section 5.3).

5.1 Contributions

Humanity, as a species, has suffered an astonishing transformation during the last decades. There has been a revolution in mathematics, chemistry, physics, biology, informatics, medicine, engineering and, also, manufacturing. This thesis contributes to the state of art through a novel methodology for manufacturing machines, where the controller maintains a constant production despite the emergence of faults.

The technological step proposed imitates species adaptation mechanism. Individuals have evolved to ensure their survivability, strengthening different features depending on their environment and their group. For instance, wolves promote motor skills to hunt as a pack, while ants construct complex societies based on roles. Despite their diverse talents, they share a collective and intrinsic instinct that adapts its behaviour when they are threatened to improve their physical and mental status. Its source varies depending on the species, for instance, in mammals is a hormone called adrenaline. Still, its finality is identical, mitigate the fatigue and injuries during a short period to escape from the menace.

When wolves perceive a hunter, they segregate adrenaline to boost their muscles and actuate faster. Factories are far from this natural environment; nonetheless, they also present hazard conditions for the manufacturing machines. Similarly to wolves, these manufacturing systems have to adapt their behaviour when they suffer harm. In this case, instead of pumping a miraculous repairing elixir into their veins, through a control algorithm that behaves actively to surpass the fault without interrupting the production.

This Fault-Tolerant Control algorithm detects, grades and isolate the fault source during an initial phase, called as Fault Detection and Isolation; and modifies the controller responses accordingly during a final phase, called as Control Redesign. This thesis has researched a novel methodology for this last phase, introducing Adaptive Control algorithms into them. On that behalf, the Model Reference Adaptive Control architecture has been updated through a flexible, robust and upgradable platform compatible with manufacturing machines.

The FDI phase has also suffered some improvements, as the controller proposed detects, grades and isolates faults through a Neural-Net. These algorithms are designed in a three-stage process. During the first stage, a report with the

primary fault sources is generated, presenting how they modify the system behaviour. During the second stage, a fault database is stored replicating the fault conditions inside a model that has identical responses as the original machine. During the third stage, the NN is trained through the information stockpiled during the previous steps.

The CR phase represents the research chore, initiating the investigation from the MRAC architecture. The current controllers based on this technology maintain systems stable and recover the former performance when new dynamics appear in the plant. Despite these benefits, the controller has to deal with faults that alter the nominal performance of manufacturing machines, requiring high flexibility platforms in the adaptation mechanism to avoid harming the system, high robustness to operate under industrial environment and high upgradability to integrate the novel techniques with the previous control algorithms.

Conventional MRAC architectures adapt the signal minimising the error between the plant responses and its closed-loop behaviour. The poles and zeros defining these models are obtained studying the system responses without faults. Despite the mathematical model accuracy, faults also alter features that have been previously unnoticed, for instance, actuator mechanical limits. This thesis substitutes the reference model stage for a Digital-Twin connected to the machine controller, replicating trustworthy the original system performance (see Sections 3.1 and 4.1).

The industrial environment represents a hostile environment for machines, as they are susceptible to suffer from multiple faults. The research has demonstrated that MRACs based on Lyapunov rules perform better against a broader range of cases than the algorithms based on MIT rule. Besides, MRACs robustness has been improved introducing an adaptive reference model stage, where the Digital-Twin controller adapts automatically its behaviour attending to the mechanical limits imposed in the machine under the fault effect (see Sections 3.2 and 4.2).

Faults affect multiple components in the manufacturing machine, altering its performance in unexpected manners. This heterogeneous pattern troubles encountering a unique adaptive gain to recover the optimal performance. Instead of a general adaptive gain, MRACs have been improved with a Bank of Controllers that switches automatically between these values. This upgradeable platform configures the controller architecture attending to the fault, selecting the configuration that surpasses the failure more efficiently (see Sections 3.3 and 4.3).

Due to these advances, a novel methodology for industrial machines based on FTC techniques is born. The mechanism maintains manufacturing machine operating when a fault emerges until they are repaired without halting the production. These novel techniques imitate natural species mechanism to continue performing despite they suffer from injuries or fatigue, bringing manufacturing machines with an adaptation mechanism that ensures its survivability.

5.2 Relevant Publications

These advancements have been presented to the scientific community in journals, conferences and workshops.

Regarding the journals, there have been published in the IEEE Access (Q1 with 86 H Index and 0.775 SJR):

Rodriguez-Guerra, J., Calleja, C., Elorza, I., Macarulla, A.M., Pujana, A., Azurmendi, I., "A Methodology for Real-Time HiL Validation of Hydraulic-Press Controllers Based on Novel Modeling Techniques" in IEEE Access, Volume 7, Number 1, Pages 110541-110553, August 2019, DOI: 10.1109/ACCESS.2019.2934170

Rodriguez-Guerra, J., Calleja, C., Elorza, I., Macarulla, A.M., Pujana, A., Ramos, M., "On Fault Tolerant Control Systems: A Novel Reconfigurable and Adaptive Solution for Industrial Machines" in IEEE Access, Volume 8, Number 1, Pages 39322-39335, February 2020, DOI: 10.1109/ACCESS.2020.2975543

Regarding the conferences, there have been the following disseminations:

Rodriguez-Guerra, J., Calleja, C., Elorza, I., Pujana, A., Azurmendi, I., "Real-time HiL for hydraulic press control validation" in SIMULTECH 2017 - Proceedings of the 7th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, Pages 126-133, 2017

Rodriguez-Guerra, J., Calleja, C., Elorza, I., Macarulla, A.M., Pujana, A., Azurmendi, I., "Fault-Tolerant Control Study and Classification: Case Study of a Hydraulic-Press Model Simulated in Real-Time" in ICFTCFD 2018: 20th International Conference on Fault-Tolerant Control and Fault Detection, 2018

5.3 Future Work

Knowledge is an enormous mountain climbed by scientists during their careers. This ascension begins with the thesis, where the young researcher selects the route for its investigation. On this journey, they found several obstacles, but they overcome them using their intellect and creativity, discovering new research topics and wisdom. Instead of being part of the slope, these roads behave as caves, where the more you investigate, the deeper you enter. Even when the dissertation has been presented, there are still bends to explore.

After the thesis conclusion, researchers continue exploring the field of expertise, expanding the knowledge. The study carried out on this thesis also experiences this phenomenon. The FTC methodology presented has been optimised for a single industrial machine. The improvements brought to MRACs allows manufacturing machines to continue producing despite faults emerge on its components, without taking into account the other systems compounding the factory.

Despite the positive results obtained with the methodology, there are instances where the productivity is slowed down. For example, the Adaptive Reference Model adjusts the DT mechanical limits to the fault condition, increasing the HP cycle time in the process. Due to this action, the fault is surpassed without interrupting the production, generating a domino effect on the additional manufacturing machines that decrease the product cadence.

Manufacturing lanes have several machines operating in series that are affected by this decreased cadence. On this context, despite the fault is surpassed at the machine level, it's effects are still perceived at the production level. This problem is out of the proposed FTC methodology scope, as the MRAC already forces the optimal performance in the machine. However, this drawback require additional studies to lessen its adverse effect. In this context, instead of improving the machine controller, the focus has switched to optimise the whole manufacturing lane.

Game Theory algorithms open the gate to novel procedures. This area of expertise was discovered by Nash during its thesis, becoming the predominant theory to describe economic transactions, as it studies mathematical models of strategic interaction among rational decision-makers. Initially, its primary application was solving zero-sum games were several participants gains and losses are balanced. After this approach, they have shown impressive results to describe logical decision making in humans, animals and computers.

The FTC methodology has solved the problem to maintain manufacturing machines operating despite the fault emergence, proposing as a future work the introduction of Game Theory algorithms to optimise the whole manufacturing lane. They consider each system as a player, creating a game where they are rewarded when the production is optimal. Due to this technique, after the MRAC recovers the system, an additional controller analyses the production and adjust the health system's behaviour to obtain the maximum benefit.

This approach combines the current FTC methodology benefits, as machines always have the optimal performance, with an external algorithm that maintains the production optimised. This supplementary control analyses energy and material consumption to rearrange the machines to operate in the optimal point. This future approach creates an environment where factories maintain the production constant and optimal despite the emergence of faults.

Append A: Hydraulic-Press

The system under test in this thesis is one of Fagor Arrasate's Hydraulic-Press (see Section 1.2.1). These heavy machines that combine mechanical, electrical and hydraulic components behave as the case study for the methodology presented in this thesis. Multiple control architectures have been part of these experiments, searching for the optimal algorithm that recovers the system from faults faster without spreading its damage (see Chapter 4).

This HP belongs to a process dedicated to stamp metal-sheets for commercial vehicles. The manufacturing machine describes a repetitive cycle to mould the plates after compressing two cylinders (see Fig. A.1). The following points introduce the characteristics of each part briefly:

- **Slide:** HP upper section compounded of one cylinder divided into two chambers, main and annular. The axial piston pumps fed both chambers, regulating the flow rate and pressure inside them through a single proportional valve.
- **Cushion:** Bottom section in the HP set up of six single-acting cylinders (passives) and one double-acting (active). The hydraulic circuit has a proportional valve for each cylinder, commanding the bed movements through the active and exerting the force over the metal-sheet with the passives.

Currently, each proportional valve has its own PID controller regulating the cylinder movement or the force exerted over the metal-sheet. This HP moulds the plates in a continues and repetitive process, considered as the Hydraulic-Press cycle. Across their stages, the slide travels from its rest position until colliding with the cushion, combining both cylinders in the making force stage to mould the piece. Afterwards, they separate their piston rods, recovering the initial position.

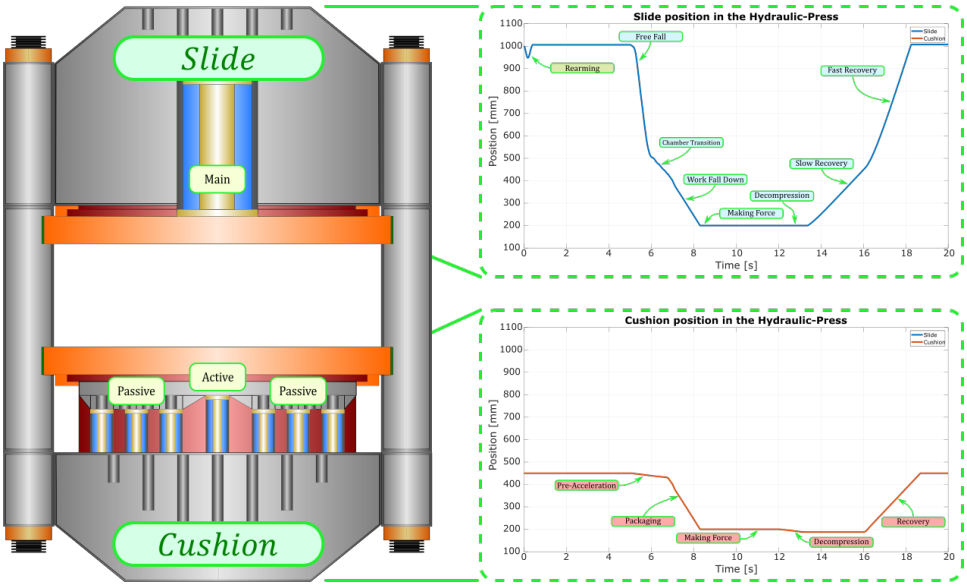


Figure A.1: Hydraulic-Press virtual representation. Slide and cushion visual description of their position and stages.

Append B: Fault Detection and Isolation

There are several harm-grades in industrial machines, from soft faults that are corrected by the controller to more severe ones where the production has to be halted. Between both extremes, there are faults that affect the system negatively but are avoidable modifying the control loop. This dissertation has been focused on this last faults, presenting a methodology to recover the system optimal performance when a drawback emerges on its components.

Faults have been already defined (see Section 2.1.2), clarifying the scope between them and failures. If the system reaches a failure status, the FTC presented on this thesis is useless, as the adaptive algorithms are unable to recover the control and restore the production. Nonetheless, for faults altering the system nominal behaviour, MRACs are able to adapt the controller signal and surpass this drawback efficiently (see Section 3.2.4).

From the terminology, every emerging negative effect in a manufacturing machine is considered a fault. Due to the heterogeneous composition of systems, there are a wide number of possible fault candidates, from sensors and actuators to internal components. Attending to the source, in AFTC approach, faults are classified in one of the following four categories:

- **Component:** Comprehends faults in the physical pieces compounding the machine. They are linked with a degraded performance in actuators or sensors, such us noise, disconnection or offsets in the signals.
- **Plant:** Encompasses the emergence of new dynamics due to internal or external factors, such as mechanical fatigue or leaks. The machine behaviour suffers variations due to mechanical degradation on its components.
- **Communication:** Represents corrupted transmissions between plant and control algorithm. This drawbacks suppose a discordance between controller and machine signals.
- **Controller:** Implies a degradation on the control-loop behaviour, leading the system to instability. They are related with controller malfunctions or erroneous user commands.

Despite the fact that faults are classified attending to their source, due to the unpredictable behaviour of components under their effect its confusing discern this information. The novel methodology presented minimizes this lack of knowledge during FDI phase (see Section B.1), comparing the current machine response against previously analysed fault cases.

B.1 Fault Process

During FDI phase, faults are detected, graded and isolated. This task present several complications due to the lack of *a priori* knowledge about the machine performance under their effect. Traditionally, mathematical methods have simplified this uncertainty identifying the fault source, nonetheless, industrial machines present non-linearities hardly reproducible by these algorithms, neglecting the mathematical approximations to detect the component damaged [147],[134].

This thesis has presented a novel methodology based on Neural-Networks to identify the fault source. This Artificial Intelligence algorithm brings a powerful instrument prepared to detect the fault, discriminate its source and grade its harm even when there are uncertainties about the degradation suffered by the system. The NN is trained through a fault database created replicating the undesired behaviour in the machine Digital-Twin. The simulations are executed under a Hardware in the Loop platform that replicates the manufacturing conditions [97], [50], [51], obtaining the fault databases through a three stages process (see Fig. B.1 and Section B.1.1, B.1.2 and B.1.3).

B.1.1 Fault Study

The main issue control designers encounter during the codification of FDI algorithms through a three stage process is the short amount of *a priori* information about the system behaviour under the effect of faults. They have to estimate the component performance in uncertain situations, predicting the fault magnitude to design a mathematical algorithm capable of grading, detecting and isolating it.

In the methodology presented, this lack of knowledge is diminished realizing an early and offline (without connection to the real machine) **fault study**. For each potential fault component, control designers realize a report presenting:

- **Scenario:** Similar components are prone to fail from multiple sources, for instance, the proportional valve performance is degraded by the mechanical fatigue or by a lack of communication with the controller. Each possible fault source has to be defined in the study, allocating them in a unique scenario.
- **Tolerance:** The components are subjected to multiple faults, nonetheless, they have an intrinsic tolerance to be affected by them, that is to say, there are faults that appear more regularly in the component than others. Control designers detect and identify the most common drawbacks to know the fault emergence likelihood.

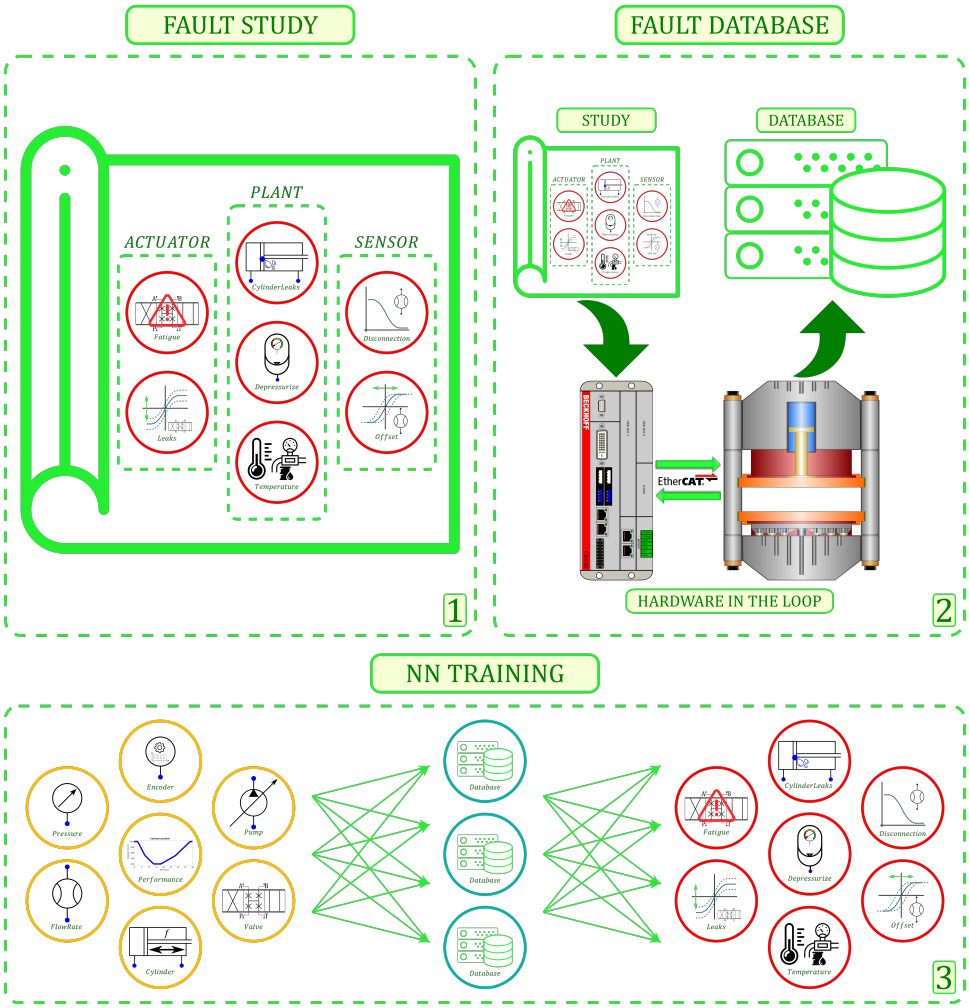


Figure B.1: Process followed during Fault Detection and Isolation phase.

- **Harm Grade:** Faults have a widespread harm grade, as they are affected from soft drawbacks that barely modify the system performance to harmful situations that require to conclude the production to avoid spreading the damage. Each fault scenario has to identify the harm grade discriminating the degradation suffered in the machine performance.

- **Range:** When a fault emerges on the system, they affect the machine performance negatively. However, this drawback continue spreading if it remains undetected by the FDI mechanism. After trespassing the nominal performance threshold, faults modify the system behaviour attending to an increasing scale representing the damage taken. This range of affliction has

been normalised through a percentage.

This study is reproduced in every machine improved with the FTC algorithm. On the Hydraulic-Press, the current system under study, thirty two fault scenarios have been identified affecting negatively the system performance without compromising completely its stability. The following Table shows the information gathered during the research, whose *Tolerance* and *Harm Grade* field have been filled with high, medium or low attending to the fault emergence probability and the damage taken in the component respectively (see Table B.1).

CASE	*COMPONENT*	*SCENARIO*	*TOLERANCE*	*HARM GRADE*
0	Machine	Nominal	High	Low
1	Proportional Valve	Input Signal Offset	High	Low
2	Proportional Valve	Leaks	Medium	Medium
3	Proportional Valve	Mechanical Fatigue [Pressure]	High	Medium
4	Proportional Valve	Mechanical Fatigue [Flow Rate]	High	Medium
5	Pump 1	Viscosity Deviation	Low	High
6	Pump 2	Viscosity Deviation	Low	High
7	Pump 3	Viscosity Deviation	Low	High
8	Cylinder	Internal Leaks	Medium	Medium
9	Cylinder	External Leaks	Medium	Medium
10	Pump 1	Drop [Pressure]	Medium	Low
11	Pump 2	Drop [Pressure]	Medium	Low
12	Pump 3	Drop [Pressure]	Medium	Low
13	Deposit	Drop [Pressure]	Medium	Low
14	Cylinder Main Chamber	Sensor Offset [Pressure]	Medium	Low

Table B.1 – *Continued from previous page*

CASE	*COMPONENT*	*SCENARIO*	*TOLERANCE*	*HARM GRADE*
15	Cylinder Annular Chamber	Drop [Pressure]	Medium	Low
16	Pump 1	Drop [Flow Rate]	Medium	Low
17	Pump 2	Drop [Flow Rate]	Medium	Low
18	Pump 3	Drop [Flow Rate]	Medium	Low
19	Deposit	Drop [Flow Rate]	Medium	Low
20	Cylinder Main Chamber	Drop [Flow Rate]	Medium	Low
21	Cylinder Annular Chamber	Drop [Flow Rate]	Medium	Low
22	Cylinder Stroke	Sensor Offset [Position]	Medium	Low
23	Cylinder Stroke	Sensor Offset [Velocity]	Medium	Low
24	Pump 1	Signal Offset [Power]	Medium	Medium
25	Pump 2	Signal Offset [Power]	Medium	Medium
26	Pump 3	Signal Offset [Power]	Medium	Medium
27	Pump 1	Signal Offset [Pressure]	Low	Medium
28	Pump 2	Signal Offset [Pressure]	Low	Medium
29	Pump 3	Signal Offset [Pressure]	Low	Medium
30	Pump 1	Signal Offset [Flow Rate]	Low	Medium

Table B.1 – *Continued from previous page*

CASE	*COMPONENT*	*SCENARIO*	*TOLERANCE*	*HARM GRADE*
31	Pump 2	Signal Offset [Flow Rate]	Low	Medium
32	Pump 3	Signal Offset [Flow Rate]	Low	Medium

Table B.1: Fault study developed for the system under test in this thesis, the Hydraulic-Press.

The first row in Table B.1 defines a *Nominal* fault scenario (Case 0), referring to the optimal machine behaviour. Despite this situation is not a fault per se, the NN algorithm has to classify every scenario into at least one case, so control designers have to consider also the nominal performance as part of one fault situation, despite there is no harm in the machine. Control designers have to realize this study carefully, as the information brought from it serves as the initial point to generate the database, employed afterwards to train the Neural-Networks.

B.1.2 Fault Database

Despite the accurate mathematical models describing the system behaviour when a fault emerges, they still present uncertainties to reproduce trustworthy the machine performance under its effect. This condition reduces the effectiveness of fault databases, as the mathematical models only present a hypothetical performance. Manufactures also have a limited information of faults from previous cases, so the FDI algorithms have to be designed through partial information obtained from previous fault situations or inaccurate experimental analysis done in the laboratory.

This thesis presents a novel methodology prepared to increase the feasibility of the experiments developed in the laboratory against the system models, generating a fault database. It is generated reproducing the fault behaviour in the DT across a three step process

1. **Study:** Initially, the DT is configured for the fault case under study. This virtual system reproduces the drawbacks through the tolerance, harm grade and range information gathered on the previous study (see Section B.1.1).
2. **Hardware in the Loop:** Afterwards, when the DT has been embedded in this platform and connected to the original controller, the experiment is configured. On these platform, the system behaviour is completely reproduced as it replicates the machine conditions in the manufacturing process (See Section 3.1.4.2).
3. **Database:** Finally, each fault case is reproduced cyclically on the HiL platform, adjusting the range in every iteration. The measures obtained from

the virtual sensors is stored to create a database with information about the system performance under the effect of faults.

When the fault replication process is finished, control designers would have a database with real information about the system performance under the effect of each fault considered during the study. Each case in B.1 has been reproduced using this procedure, generating a fault database (see Table B.2) that classifies the faults attending to their source into plant or component, discerning between sensor or actuator for the last one.

CASE	*SOURCE*	*DESCRIPTION*
0	Plant	Nominal behaviour
1	Component [Actuator]	Offset in the signal read from the controller leading to an erroneous opening
2	Plant	Internal leaks between the proportional valve chambers
3	Plant	Proportional valve characteristic curve modified due to a mechanical fatigue associated with the pressure jump
4	Plant	Proportional valve characteristic curve modified due to a mechanical fatigue associated with looses in the flow rate
5	Plant	Oil density modification due to the external temperature in Pump 1
6	Plant	Oil density modification due to the external temperature in Pump 2
7	Plant	Oil density modification due to the external temperature in Pump 3
8	Plant	Hydraulic looses between the cylinder internal chambers (main and annular)
9	Plant	Hydraulic looses between the annular chamber and the outside
10	Plant	Pressure variation in the hydraulic circuit connecting the Pump 1
11	Plant	Pressure variation in the hydraulic circuit connecting the Pump 2
12	Plant	Pressure variation in the hydraulic circuit connecting the Pump 3
13	Plant	Pressure variation in the hydraulic circuit connecting the Deposits

Table B.2 – *Continued from previous page*

CASE	SOURCE	DESCRIPTION
14	Plant	Pressure variation in the hydraulic circuit connecting the Cylinder Main Chamber
15	Plant	Pressure variation in the hydraulic circuit connecting the Cylinder Annular Chamber
16	Plant	Flow Rate variation in the hydraulic circuit connecting the Pump 1
17	Plant	Flow Rate variation in the hydraulic circuit connecting the Pump 2
18	Plant	Flow Rate variation in the hydraulic circuit connecting the Pump 3
19	Plant	Flow Rate variation in the hydraulic circuit connecting the Deposit
20	Plant	Flow Rate variation in the hydraulic circuit connecting the Cylinder Main Chamber
21	Plant	Flow Rate variation in the hydraulic circuit connecting the Cylinder Annular Chamber
22	Component [Sensor]	Noise or offset variation in the measure received from the position sensor
23	Component [Sensor]	Noise or offset variation in the measure received from the velocity sensor
24	Component [Actuator]	Variation on the input signal of the power in Pump 1
25	Component [Actuator]	Variation on the input signal of the power in Pump 2
26	Component [Actuator]	Variation on the input signal of the power in Pump 3
27	Component [Actuator]	Variation on the input signal of the pressure in Pump 1
28	Component [Actuator]	Variation on the input signal of the pressure in Pump 2
29	Component [Actuator]	Variation on the input signal of the pressure in Pump 3
30	Component [Actuator]	Variation in the input signal of the flow rate in Pump 1

Table B.2 – *Continued from previous page*

CASE	SOURCE	DESCRIPTION
31	Component [Actuator]	Variation in the input signal of the flow rate in Pump 2
32	Component [Actuator]	Variation in the input signal of the flow rate in Pump 3

Table B.2: Classification of faults regarding their source and brief description about the system performance under their effect.

On Table B.2, there has been considered only two fault sources, component and plant. Despite communication and controller faults also affect negatively the system behaviour, the FTC methodology presented is not prepared to surpass them. The first case require to introduce an additional loop to ensure that controller and plant are connected and synchronised, improvement that has to be considered during control design. The second case implies a degradation of the MRAC or PID controller performance, leading to system failure without any possibility to recover the machine.

B.1.3 Neural-Networks

A Neural-Network is an artificial circuit of neurons (nodes), simulating the human brain behaviour to solve Artificial Intelligence algorithms. The NN designed on this thesis is based on the *sigmoid neuron* an evolution from *perceptrons*, that operate taking several binary inputs $x_j = [x_1, x_2, \dots [$ to produce a single binary output, computed through a rule based on real numbers to express the importance between both signals called *weights* $w_j = [w_1, w_2, \dots]$. The neuron's output, 0 or 1, is determined by whether the weighted sum $\sum_j w_j x_j$ is less or greater than some *threshold* value:

$$output = \begin{cases} 0 & if \quad \sum_j w_j x_j \leq threshold \\ 1 & if \quad \sum_j w_j x_j > threshold \end{cases} \tag{B.1}$$

This equation is simplified introducing two changes to $\sum_j w_j x_j > threshold$, substituting the expression $\sum_j w_j x_j$ for a dot product of vectors $w \cdot x$ and moving the threshold to the other side of the inequality, replacing it by the *perceptron's bias* b:

$$output = \begin{cases} 0 & if \quad w \cdot x + b \leq 0 \\ 1 & if \quad w \cdot x + b > 0 \end{cases} \tag{B.2}$$

Despite their benefits, a small change in the weights or bias of *perceptrons* produce high variations in the output, going from a 0 value to a 1. This binary

performance cause the network behaviour to change in a complicated manner, increasing the difficulty to detect gradual modifications that are getting closer to the desired performance. This problem is overcome introducing a new type of artificial neuron called a *sigmoid*. Just like their predecessors, the neuron has inputs $x_j = [x_1, x_2, \dots]$, but instead of being just 0 or 1, they can take any values in that range. Similarly they have weights $w_j = [w_1, w_2, \dots]$, and an overall bias b, whose equation is described by a sigmoid function σ:

$$\sigma(z) \equiv \frac{1}{1 + e^{-z}} \tag{B.3}$$

where the output of a sigmoid neuron with inputs $x_j = [x_1, x_2, \dots]$, weights $w_j = [w_1, w_2, \dots]$ and bias b is:

$$\frac{1}{1 + e^{-\sum_j w_j x_j - b}} \tag{B.4}$$

This thesis presents a FDI algorithm based on a Neural-Networks trained with the information gathered from replicating in a HiL platform the fault situation (See Section B.1.2). Across the Hydraulic-Press cycle, control designers mark critical points where the NN algorithm analyse actuators and sensor responses detecting if a fault has emerge and classifying it into one of the scenarios studied on Section B.1.1. When the HP completes a cycle, an additional NN would ensure that a fault has emerge avoiding false positive, grading the damage taken in the component and isolating it from the system. This information is sent to the CR phase.

Append C: General Design of MRACs

Section 3.2 presents the MRACs mathematical analysis for several generic systems, varying the number of input and output constraints. Despite the accuracy of this study, some intermediate steps have been avoided to simplify the equations, reducing the option for other researchers to repeat the research or corroborate the results. In order to mitigate this situation, this append introduces general concepts regarding the design of Adaptive Controllers, exposing all the intermediate steps and simplifications accounted for in the equations from Section 3.2.

Before explaining the equations, the notation describing the variables will be introduced (later on, this clarification would be properly defined):

- **Symbol $*$:** Denotes an unknown parameter for the user.
- **Symbol $\wedge$:** Denotes an unknown parameter that has been estimated by the user.
- **Symbol $\sim$:** Denotes an unknown parameter obtained from the difference between an estimated and a measured parameter.

Previously to untangle the adaptive procedure, the mathematical analysis defines the system closed-loop response $y_p(t)$:

$$\dot{y}_p(t) = a_p y_p(t) + b_p \left[\Lambda u(t) + f(y_p)\right] \tag{C.1}$$

where $y_p(t)$ is the system output signal, a_p, b_p represent known parameters defining the plant dynamics, $u(t)$ is the output controller signal and $f(y_p)$ is the unknown dynamics or fault in this thesis defined as:

$$f(y_p) = \theta^{*\intercal} \phi(y_p) \tag{C.2}$$

being the variables:

$$\theta^{*^{\intercal}} \triangleq unknown\ matrix\ with\ dimensions \in \mathbb{R}^{p\times m} \tag{C.3}$$

$$\phi(y_p) = \left[\phi_1(y_p)\dots\phi_p(y_p)\right] \triangleq known\ vector\ with\ dimensions \in \mathbb{R}^{m\times p} \tag{C.4}$$

This equations are feasible regarding that $\phi(y_p)$ is a locally Lipschitz function.

Due to the early assumptions, the main equation describing the system behaviour is rewrite as:

$$\dot{y}_p(t) = a_p y_p(t) + b_p \Lambda u(t) + b_p \theta^{*^{\intercal}} \phi(y_p) \tag{C.5}$$

The system presents known and unknown terms, being necessary to encounter a control law that tracks the reference signal and maintains the system stable despite these latest uncertainties. To accomplish this task, control designers follow a tuning process divided into five stages.

C.1 Nominal Control Design

The control signal $u(t)$ from the Equation C.5 has two terms:

$$u(t) = u_n(t) + u_a(t) \tag{C.6}$$

$$Controller = Nominal\ Control\ Design\ +\ Ref.\ Model\ Selection \tag{C.7}$$

where $u_n(t)$ is the nominal control component designed accordingly to the plant dynamics and $u_a(t)$ is the adaptive control component.

Instead of designing in unison both control actions, the study focuses on tuning initially the nominal controller and, afterwards, implementing the adaptive algorithm, that is to say, $u_a(t) = 0$, $\theta^{*^{\intercal}}\phi(y_p) = 0$ and $\Lambda = I$. This situation leads to design a controller for a simplified system:

$$\dot{y}_{p_n}(t) = a_p y_{p_n}(t) + b_p u_n(t) \tag{C.8}$$

where $y_{p_n}(t)$ is the system closed-loop for the plant without external dynamics.

There are several controller topologies, for instance, PI, PD or PID architectures. However, as this append presents a generic case, its variable $u(t)$ is solved selecting generic gains:

$$u(t) = u_n(t) = -k_1 y_{p_n}(t) + k_2 r(t) \tag{C.9}$$

where k_1 and k_2 are matrices of known controller gains with dimensions $k_1 \in \mathbb{R}^{m\times p}$ and $k_2 \in \mathbb{R}^{m\times r}$, respectively and $r(t)$ is the reference signal selected by the user with dimensions $r(t) \in \mathbb{R}^{r\times 1}$.

Substituting this terms in the Equation C.8, the nominal system is:

$$\dot{y}_{p_n}(t) = a_p y_{p_n}(t) + b_p \left[-k_1 y_{p_n}(t) + k_2 r(t)\right] \tag{C.10}$$
$$= \left[a_p - b_p k_1\right] y_{p_n}(t) + \left[b_p k_2\right] r(t) \tag{C.11}$$

extracting the reference model closed-loop parameters a_m and b_m as:

$$a_m = a_p - b_p k_1 \tag{C.12}$$
$$b_m = b_p k_2 \tag{C.13}$$

Assuming that the terms are Hurwitz or the system is dynamically stable, the following Lyapunov equation is satisfied:

$$0 = a_m^{\intercal} P + P a_m + I \tag{C.14}$$

where P is the unknown n by n symmetric matrix in the algebraic Riccati equation with positive sign $P > 0$.

If the conditions previously mentioned are accomplished, the reference model is defined by the equation:

$$\dot{y}_m(t) = a_m y_m(t) + b_m r(t) \tag{C.15}$$

C.2 Adaptive Control Design

The previous study (see Section C.1) obtains the control algorithm for the nominal plant, similarly as how control designers have tuned these algorithms traditionally. Instead of a conventional controller, the study presented has an additional adaptive control algorithm, so combining Equation C.5, C.6 and C.9 is obtained a system that attends this characteristic:

$$\dot{y}_p(t) = a_p y_p(t) + b_p \Lambda \left[-k_1 y_p(t) + k_2 r(t) + u_a\right] + b_p \theta^{*\intercal} \phi(y_p) \tag{C.16}$$

This equation represents the system controlled by an adaptive algorithm, but the expression has reached their maximum simplification form. The study continues adding three terms that are neutral to the equation:

$$Term\ 1 \rightarrow \Lambda\Lambda^{-1} = 1 \tag{C.17}$$
$$Term\ 2 \rightarrow b_p k_1 y_p(t) - b_p k_1 y_p(t) = 0 \tag{C.18}$$
$$Term\ 3 \rightarrow b_p k_2 r(t) - b_p k_2 r(t) \tag{C.19}$$

$$\begin{aligned}\dot{y}_p(t) = a_p y_p(t) + b_p\Lambda\left[-k_1 y_p(t) + k_2 r(t) + u_a\right] + b_p\theta^{*\intercal}\phi(y_p)\Lambda\Lambda^{-1} \\ + b_p k_1 y_p(t)\Lambda\Lambda^{-1} - b_p k_1 y_p(t) + b_p k_2 r(t) - b_p k_2 r(t)\Lambda\Lambda^{-1}\end{aligned} \tag{C.20}$$

This equation is rearranged considering the additional terms:

$$\begin{aligned}\dot{y}_p(t) = \underbrace{(a_p - b_p k_1)}_{a_m} y_p(t) + \underbrace{b_p k_2}_{b_m} r(t) \\ + b_p\Lambda\left[u_n(t) + u_a(t) + \Lambda^{-1}\theta^{*\intercal}\phi(y_p) - \Lambda^{-1}\left[\underbrace{-k_1 y_p(t) + k_2 r(t)}_{u_n(t)}\right]\right]\end{aligned} \tag{C.21}$$

where the adaptive terms θ^* and $\phi(y_p, u_n)$ are defined as:

$$\theta^* \triangleq \left[\Lambda^{-1}\theta^{*\intercal}, (I - \Lambda^{-1})\right]^\intercal \tag{C.22}$$
$$\phi(y_p, u_n) \triangleq \begin{bmatrix}\phi(y_p) \\ u_n\end{bmatrix} \tag{C.23}$$

and the bracket equation (red colour) is solved as:

$$u_n(t) + u_a(t) + \Lambda^{-1}\theta_p^{*\intercal}\phi(y_p) - \Lambda^{-1}u_n(t) \tag{C.24}$$
$$\underbrace{\left[I - \Lambda^{-1}\right]u_n(t) + \Lambda^{-1}\theta_p^{*\intercal}\phi}_{\theta^{*\intercal}\phi(.)} + u_a(t) \tag{C.25}$$

being $\theta^{*\intercal}$ the unknown weight matrix and $\phi(.)$ the known aggregated basis function.

After substituting the assumptions in Equation C.21, the following system model is obtained:

$$\dot{y}_p(t) = a_m y_p(t) + b_m r(t) + b_p\Lambda\left[u_a(t) + \theta^{*\intercal}\phi(.)\right] \tag{C.26}$$

where the brackets terms (red colour) have to be cancelled to design an adaptive controller that tracks references without entering unstable regions.

In a conventional design, the terms are cancelled when the control action is identical to the weights:

$$u_a(t) = -\theta^{*\intercal}\phi(.) \tag{C.27}$$

however, in this case the weight matrix $\theta^{*\intercal}$ is unknown, being necessary to estimate it:

$$u_a(t) = -\hat{\theta}^{*\intercal}\phi(.) \tag{C.28}$$

$$\tilde{\theta} \triangleq \hat{\theta} - \theta^* \tag{C.29}$$

obtaining the expression:

$$u_a(t) + \theta^{*\intercal}\phi(.) \tag{C.30}$$

$$-\hat{\theta}^{\intercal}\phi(.) + \theta^{*\intercal}\phi(.) \tag{C.31}$$

$$-\left(\hat{\theta} - \theta^*\right)^{\intercal}\phi(.) \tag{C.32}$$

$$-\tilde{\theta}^{\intercal}\phi(.) \tag{C.33}$$

Substituting these assumptions in Equation C.26, the final expression defining the system behaviour is:

$$\dot{y}_p(t) = a_m y_p(t) + b_m r(t) - b_p \Lambda \tilde{\theta}\phi(.) \tag{C.34}$$

Despite Equation C.34 represents a controlled system by an adaptive control algorithm, there are still uncertainties. The designer has to select a value for the unknown weight $\tilde{\theta}$. These uncertainties are solved through a parameter adjustment mechanism, that minimise the tracking error:

$$e(t) \triangleq y_p(t) - y_m(t) \tag{C.35}$$

$$\dot{e}(t) = \dot{y}_p(t) - \dot{y}_m(t) \tag{C.36}$$

$$= a_m y_p(t) + b_m r(t) - b_p \Lambda \tilde{\theta}\phi(.) - a_m y_m(t) - b_m r(t) \tag{C.37}$$

$$= a_m e(t) - b_p \Lambda \tilde{\theta}^{\intercal}\phi(.) \tag{C.38}$$

where the values of $\dot{y}_p(t)$ and $\dot{y}_m(t)$ have been obtained from Equations C.34 and C.15, respectively.

C.3 Error Equation Definition

From the previous study (see Section C.2) the first derivative of the tracking error equation has been defined as:

$$\dot{e}(t) = a_m e(t) - b_p \Lambda \tilde{\theta}^\intercal \phi(.) \tag{C.39}$$

where the matrix weight $\tilde{\theta}^\intercal$ is $\tilde{\theta}^\intercal = \hat{\theta}^\intercal - \theta^{*\intercal}$.

The parameter adjustment mechanism is constructed minimizing a potential function, in this case, a Lyapunov equation that stabilizes the non-linear system:

$$V(t) = e(t)^\intercal P e(t) + \gamma^{-1} tr\left(\tilde{\theta}\Lambda^{1/2}\right)^\intercal \left(\tilde{\theta}\Lambda^{1/2}\right) \tag{C.40}$$

where $\gamma > 0$ for each possible value.

The stability margins are studied through the first derivative of Equation C.40:

$$\dot{V}(t) = 2e(t)^\intercal P\left[a_m e(t) - b_m \Lambda \tilde{\theta}^\intercal \phi(.)\right] + 2\gamma^{-1} tr\left(\Lambda \tilde{\theta}^\intercal \dot{\tilde{\theta}}\right) \tag{C.41}$$

$$= 2e(t)^\intercal P a_m e(t) - 2e(t)^\intercal P b_m \Lambda \tilde{\theta}^\intercal \phi(.) + 2\gamma^{-1} tr\left(\Lambda \tilde{\theta}^\intercal \dot{\tilde{\theta}}\right) \tag{C.42}$$

whose first term (green colour) is solved as:

$$2e(t)^\intercal P a_m e(t) \tag{C.43}$$

$$e(t)^\intercal P a_m e(t) + e(t)^\intercal P a_m^\intercal e(t) + e(t)^\intercal P a_m e(t) \tag{C.44}$$

$$e(t)^\intercal \left[P a_m^\intercal + P a_m\right] e(t) \tag{C.45}$$

$$e(t)^\intercal \left[-I\right] e(t) \tag{C.46}$$

$$-e(t)^\intercal e(t) \tag{C.47}$$

$$-\|e\|_2^2 \tag{C.48}$$

whose second term (blue colour) and third term (red colour) are solved through the trace theory $tr\left(a^\intercal b\right) = tr\left(ab^\intercal\right) = a^\intercal b$:

$$2e(t)^\intercal P b_m \Lambda \tilde{\theta}^\intercal \phi(.) \tag{C.49}$$

$$-2tr\left(\underbrace{\Lambda\tilde{\theta}\phi(.)}_{b}\underbrace{e(t)^\intercal P b_m}_{a}\right) + 2\gamma^{-1} tr\left(\Lambda \tilde{\theta}^\intercal \dot{\tilde{\theta}}\right) \tag{C.50}$$

$$-2\gamma^{-1} tr\left(\Lambda \tilde{\theta}^\intercal \left[\gamma \phi(.) e(t)^\intercal P b_m - \dot{\tilde{\theta}}\right]\right) \tag{C.51}$$

simplifying these additional terms (purple colour) selecting the parameter adjustment mechanism as:

$$\dot{\hat{\theta}} = \gamma\phi(.)e(t)^{\intercal}Pb_m \tag{C.52}$$

After cancelling the uncertain terms in Equation C.42, the Lyapunov stability is achieved when:

$$\dot{V}(t) = -\|e\|_2^2 \leqslant 0 \tag{C.53}$$

The structure $\dot{V}(t) \leqslant 0$ gives the condition $(e(t), \tilde{\theta}) \in \mathscr{L}_\infty$, as they belong to the space of $\mathscr{L}$ infinity.

C.4 Ensure Lyapunov Boundedness

The assumptions made on Equation C.53 maintain the system stable while the Lyapunov function remains boundedness. This condition is reached when the second derivative $\ddot{V}(t)$ remains bounded, that is to say, the first derivative $\dot{V}(t) \to 0$ and therefore the error $e(t) \to 0$ also tends to its minimum value.

Due to the early assumptions, the system ensures $\ddot{V}(t)$ boundedness when:

$$b d \ddot{V}(t)\ ;\ \dot{V}(t) \to 0 \tag{C.54}$$

$$\dot{V}(t) = -\|e\|_2^2 = -e(t)^{\intercal}e(t) \tag{C.55}$$

$$\ddot{V}(t) = -2\underbrace{e^{\intercal}(t)}_{bd}\underbrace{\dot{e}(t)}_{?} \tag{C.56}$$

so the assumption is corroborated if $\dot{e}(t)$ is bounded (see Equation C.39):

$$\begin{aligned}\dot{e}(t) &= a_m e(t) - b_m \Lambda \tilde{\theta}^{\intercal}\phi(.) \\ &= a_m \underbrace{e(t)}_{bd} - b_m \Lambda \left(\underbrace{\tilde{\theta}}_{bd} - \underbrace{\theta}_{bd} \right) \phi \left(\underbrace{e(t)}_{bd} + \underbrace{y_m(t)}_{bd}, \underbrace{u_n(t)}_{bd} \right)\end{aligned} \tag{C.57}$$

as all the terms are bounded, so $\dot{e}(t)$ must also be bounded by Barbara's lemma, ergo:

$$\dot{V} = -\|e\|_2^2 \to 0 \; as \; t \to \infty \tag{C.58}$$

$$\lim_{t\to\infty} e(t) = 0 \tag{C.59}$$

C.5 Selection of the Learning Rate

Lastly, when the boundedness of $\ddot{V}(t)$ is ensured, the system is stable and tracks the reference signal correcting the deviations through the adaptive control algorithm. This adaptation process would reach the optimal value faster attending to the learning rate or adaptive gain γ:

$$\dot{V}(t) \leqslant 0 \Rightarrow V\left(e(t), \tilde{\theta}(t)\right) \leqslant V\left(e_0, \tilde{\theta}_0\right) \tag{C.60}$$

assuming the initial error is null $e_0 = 0$ and the first estimation weight is also zero $\tilde{\theta}_0^* = \hat{\theta}_0 - \theta_0 \; where \; \hat{\theta}_0 = 0$, the comparison evolves to:

$$V\left(e(t), \tilde{\theta}(t)\right) \leqslant \underbrace{e_0^\intercal P e_0}_{0} + \underbrace{\gamma^{-1} tr\left(\Lambda \tilde{\theta}_0^\intercal \tilde{\theta}_0\right)}_{\gamma^{-1} tr\left(\Lambda \theta^\intercal \theta\right)} \tag{C.61}$$

the lower bound of the equation is determined by the expression:

$$e(t)^\intercal P e(t) \leqslant \gamma^{-1} tr\left(\Lambda \theta^\intercal \theta\right) \tag{C.62}$$

and as the first term is defined by the minimum eigenvalue $\lambda(p)\left\|e(t)\right\|_2^2 \leqslant e(t)^\intercal P e(t)$, the euclidean norm of the error is upper bounded by:

$$\left\|e(t)\right\|_2 \leqslant \left[\frac{1}{\gamma \lambda(p)} tr\left(\Lambda \theta^\intercal \theta\right)\right]^{1/2} \tag{C.63}$$

Equation C.63 holds for all the time $\forall t \geqslant 0$, so the error converges to zero. This upper bond is refereed as the transient performance, determining the adaptive controller behaviour under the uncertain dynamical system. This assumption implies that the distance between $y_p(t)$ and $y_m(t)$ captured by the error will not be more than the obtained term (green colour). This equation is tuned through the value of gamma γ, that is to say, the bigger of its value, the smaller of the transient performance bound.

Append D: Extended Figures

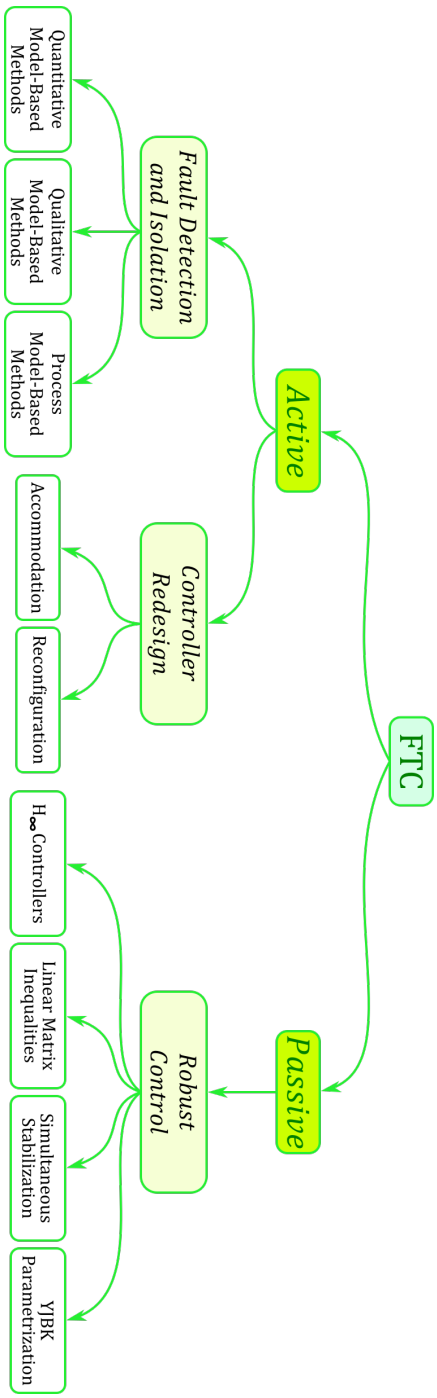


Figure D.1: Expanded classification of the Fault-Tolerant control techniques (text 2.2).

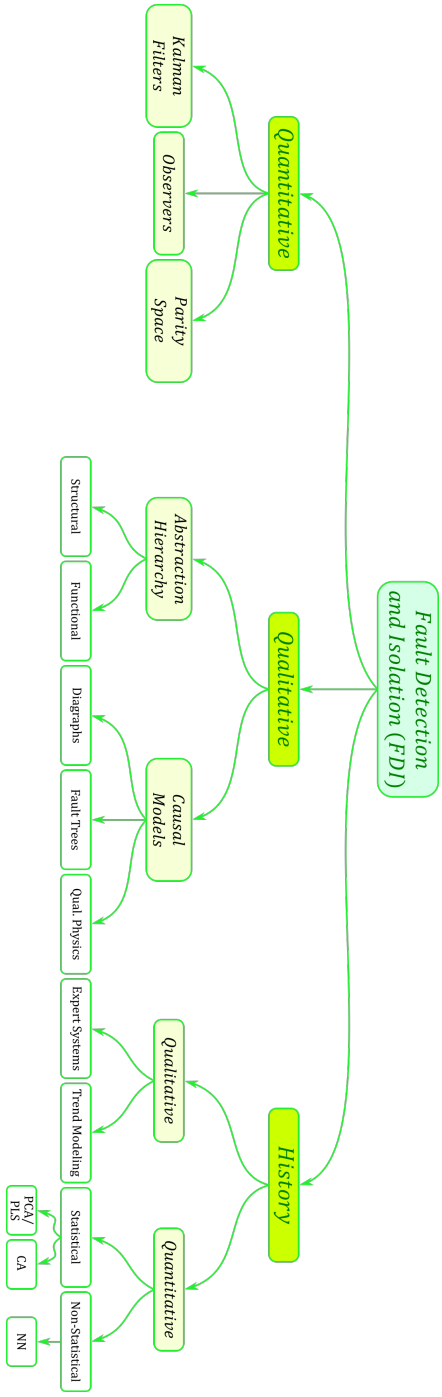


Figure D.2: Expanded Fault Detection and Isolation (FDI) methodologies (text 2.3).

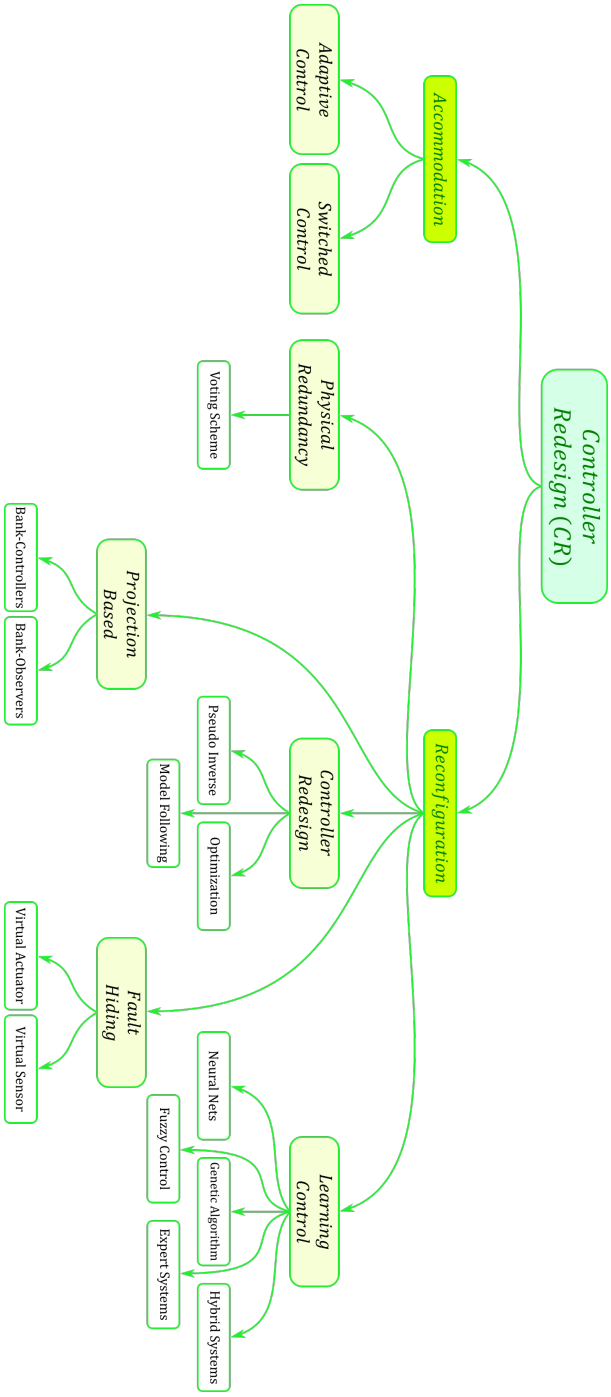


Figure D.3: Expanded controller Redesign (CR) methodologies (text 2.4).

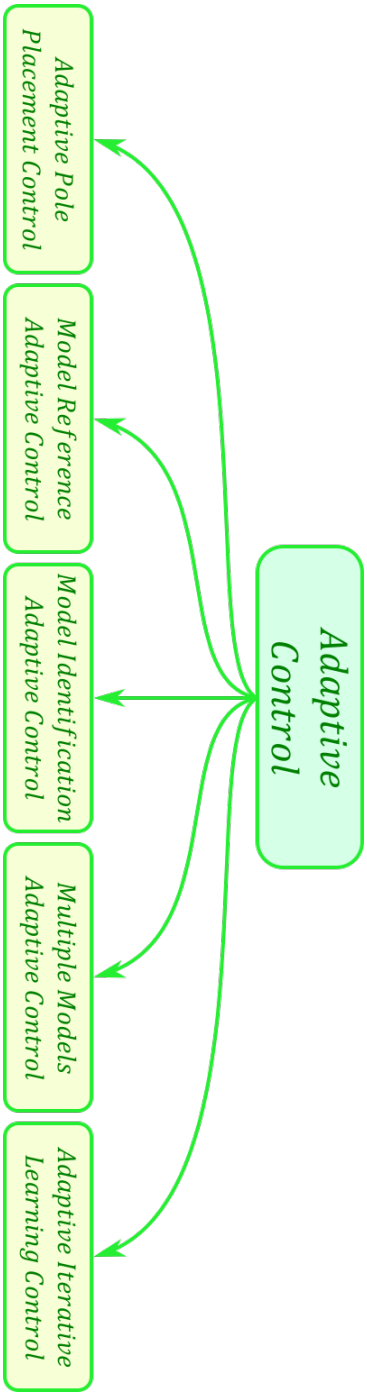


Figure D.4: Expanded AC methodologies classification attending to its topology (text 2.16).

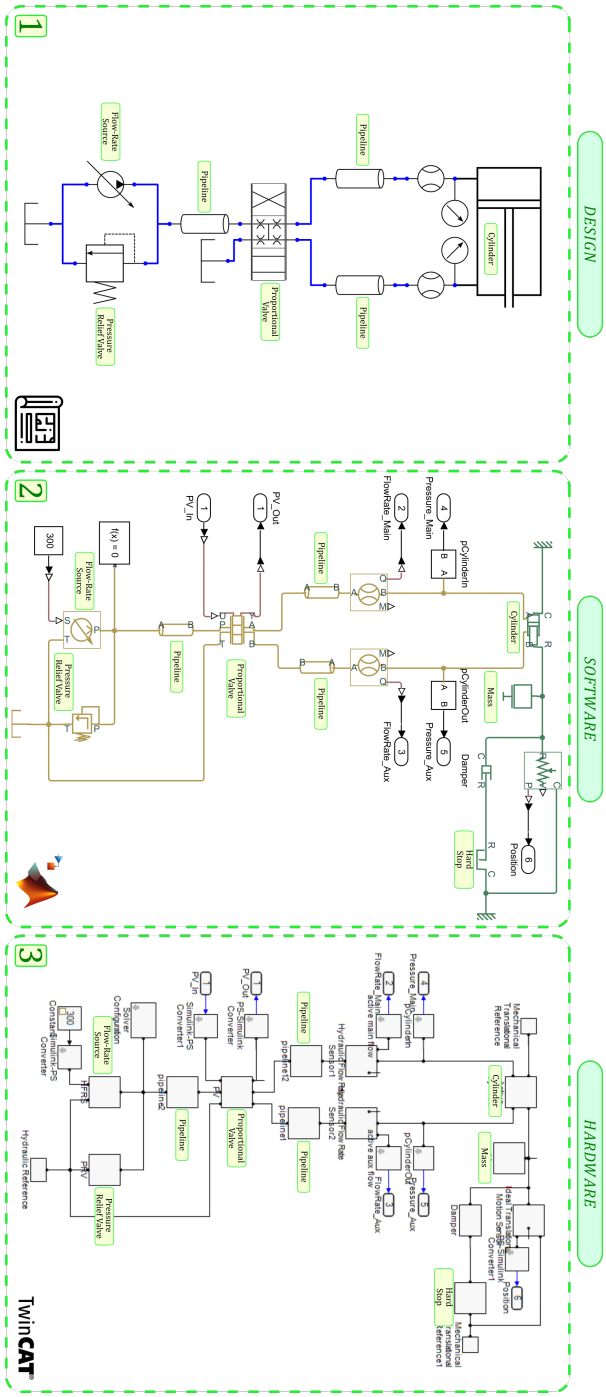


Figure D.5: Expanded virtualization process, from manufacturing machines to industrial Digital-Twins (text 3.4).

Figure D.6: Expanded overview of the MRAC algorithms studied across this section (text 3.7).

List of Tables

List of Figures

Bibliography

[1] B. Abdul Rahim and K. Soundara Rajan. Adaptive dynamic genetic algorithm based node scheduling for time-triggered systems. *Advances in Intelligent Systems and Computing*, 556:705–714, 2017. 23

[2] M. Achtelik, T. Bierling, J. Wang, L. Hõcht, and F. Holzapfel. Adaptive control of a quadcopter in the presence of large/complete parameter uncertainties. 2011. 60

[3] M. Arlcl and T. Kara. Improved Adaptive Fault-Tolerant Control for a Quadruple-Tank Process with Actuator Faults. *Industrial and Engineering Chemistry Research*, 57(29):9537–9553, 2018. 31, 35, 36, 77

[4] F.A. Bender, S. Goltz, T. Braunl, and O. Sawodny. Modeling and Offset-Free Model Predictive Control of a Hydraulic Mini Excavator. *IEEE Transactions on Automation Science and Engineering*, 14(4):1682–1694, 2017. 45

[5] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and fault-tolerant control*. Third Edition. 2016. 2, 16, 17, 18, 19, 20, 22, 25, 29, 43

[6] M. Bodson and J.E. Groszkiewicz. Multivariable adaptive algorithms for reconfigurable flight control. *IEEE Transactions on Control Systems Technology*, 5(2):217–229, 1997. 31, 35, 36

[7] M. Bodson and W.A. Pohlchuck. Command limiting in reconfigurable flight control. *Journal of Guidance, Control, and Dynamics*, 21(4):639–646, 1998. 36

[8] Jovan D. Boskovic and Raman K. Mehra. Stable multiple model adaptive flight control for accommodation of a large class of control effector failures. In *Proceedings of the American Control Conference*, volume 3, pages 1920–1924, 1999. 38

[9] D.A. Bristow, M. Tharayil, and A.G. Alleyne. Survey Of Iterative Learning Control: A Learning-Based Method for High-Performance Tracking Control. *IEEE Control Systems*, 26(3):96–114, 2006. 39

[10] J.L. Cale, B.B. Johnson, E. Dall'Anese, P.M. Young, G. Duggan, P.A. Bedge, D. Zimmerle, and L. Holton. Mitigating communication delays in remotely connected hardware-in-the-loop experiments. *IEEE Transactions on Industrial Electronics*, 65(12):9739–9748, 2018. 46

[11] R. Casado-Vara, P. Novais, A.B. Gil, J. Prieto, and J.M. Corchado. Distributed Continuous-Time Fault Estimation Control for Multiple Devices in IoT Networks. *IEEE Access*, 7:11972–11984, 2019. 43

[12] B. Catino, S. Santini, and M. Di Bernardo. MCS Adaptive Control of Vehicle Dynamics: An Application of Bifurcation Techniques to Control System Design. volume 3, pages 2252–2257, 2003. 34, 36

[13] A. Chakravarty, T. Khan Nizami, I. Kar, and C. Mahanta. Adaptive Compensation of Actuator Failures using Multiple Models. *IFAC-PapersOnLine*, 50(1):10350–10356, 2017. 38, 44

[14] J.-W.J. Cheng and Y.-M. Wang. Direct adaptive pole placement control with input magnitude and rate constraints - An analysis of its input matching property. volume 5, pages 4234–4235, 2001. 31, 34, 35

[15] J.Y. Choi and J.S. Lee. Adaptive iterative learning control of uncertain robotic systems. *IEE Proceedings: Control Theory and Applications*, 147(2):217–222, 2000. 39, 41

[16] A.D.D. Corcuera, A. Pujana-Arrese, J.M. Ezquerra, A. Milo, and J. Landaluze. Linear models-based LPV modelling and control for wind turbines. *Wind Energy*, 18(7):1151–1168, 2015. 21

[17] A.D.D. Corcuera, A. Pujana-Arrese, J.M. Ezquerra, E. Segurola, and J. Landaluze. h_∞ based control for load mitigation in wind turbines. *Energies*, 5(4):938–967, 2012. 21

[18] A.D.D. Corcuera, A. Pujana-Arrese, J.M. Ezquerra, E. Segurola, and J. Landaluze. Linear models based LPV (Linear Parameter Varying) controls for wind turbines. volume 1, pages 311–317, 2013. 21

[19] A.D.D. Corcuera, A. Pujana-Arrese, J.M. Ezquerra, E. Segurola, and J. Landaluze. Wind turbine load mitigation based on multivariable robust control and blade root sensors. volume 555, 2014. 21

[20] F. Das Chagas Da Silva Jr. and Dantas De Araújo Aldayr. A variable structure Adaptive Pole Placement Control applied to the speed control of a three-phase induction motor. volume 7, pages 1052–1057, 2007. Issue: PART 1. 30, 34

[21] S. Dey, P. Pisu, and B. Ayalew. A Comparative Study of Three Fault Diagnosis Schemes for Wind Turbines. *IEEE Transactions on Control Systems Technology*, 23(5):1853–1868, 2015. 21

[22] K. Ding, A. Morozov, and K. Janschek. Classification of hierarchical fault-tolerant design patterns. volume 2018-January, pages 612–619, 2018. 20

[23] J. dos Reis, C. Oliveira Costa, and J. Sá da Costa. Strain gauges debonding fault detection for structural health monitoring. *Structural Control and Health Monitoring*, 25(12), 2018. 24

[24] M.A. Duarte and K.S. Narendra. A new approach to model reference adaptive control. *International Journal of Adaptive Control and Signal Processing*, 3(1):53–73, 1989. 31, 36

[25] C.C.A. Eguti and L.G. Trabasso. The virtual commissioning technology applied in the design process of a flexible automation system. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 40(8), 2018. 47

[26] M.M. Fateh and M.M. Zirkohi. Adaptive impedance control of a hydraulic suspension system using particle swarm optimisation. *Vehicle System Dynamics*, 49(12):1951–1965, 2011. 35

[27] Gang Feng, C. Zhang, and M. Palaniswami. Adaptive pole placement control subject to input amplitude constraints. volume 3, pages 2493–2498, 1991. 31, 34, 35

[28] D. Fennibay, A. Yurdakul, and A. Sen. Introducing hardware-in-loop concept to the hardware/software co-design of real-time embedded systems. pages 1902–1909, 2010. 50

[29] L. Ferranti, Y. Wan, and T. Keviczky. Fault-tolerant reference generation for model predictive control with active diagnosis of elevator jamming faults. 2018. 23

[30] D. Fischer and R. Isermann. Mechatronic semi-active and active vehicle suspensions. *Control Engineering Practice*, 12(11):1353–1367, 2004. 9

[31] M. Fratczak, P. Nowak, T. Klopot, J. Czeczot, S. Bysko, and B. Opilski. Virtual commissioning for the control of the continuous industrial processes - Case study. pages 1032–1037, 2015. 47

[32] W. Fuli and L. Mingzhong. A neural network-based adaptive pole placement controller for nonlinear systems. *International Journal of Systems Science*, 28(4):415–421, 1997. 35

[33] S.K. Ghoshal and A.K. Samantaray. Multiple fault disambiguations through parameter estimation: a bond graph model-based approach. *International Journal of Intelligent Systems Technologies and Applications*, 5(1-2):166–184, 2008. 22

[34] T.E. Gibson, A.M. Annaswamy, and E. Lavretsky. On adaptive control with closed-loop reference models: Transients, oscillations, and peaking. *IEEE Access*, 1:703–717, 2013. 77

[35] A.O. Gizatullin and K.A. Edge. Adaptive control for a multi-axis hydraulic test rig. *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*, 221(2):183–198, 2007. 34, 36

[36] O. Goloubeva, M. Rebaudengo, M. Sonza Reorda, and M. Violante. Improved software-based processor control-flow errors detection technique. pages 583–589, 2005. 16

[37] Graham Goodwin. Model Identification and Adaptive Control: From Windsurfing to Telecommunications. ISBN: 9781447107118. 37

[38] Q. Guo and D. Xie. On-line identification and compensation-based model reference adaptive neural network speed control for LPMSM. pages 322–325, 2002. 31, 35, 36

[39] N. Hadroug, A. Hafaifa, N. Batel, A. Kouzou, and A. Chaibet. Active fault tolerant control based on a neuro fuzzy inference system applied to a two shafts gas turbine. 2018. 23, 27

[40] M.T. Hamayun, C. Edwards, and H. Alwi. An output integral sliding mode FTC scheme using control allocation. *Studies in Systems, Decision and Control*, 61:81–101, 2016. 23, 28

[41] W. He and S.S. Ge. Cooperative control of a nonuniform gantry crane with constrained tension. *Automatica*, 66:146–154, 2016. 33

[42] J.S. Hernandez, R. Rivas-Perez, and J.J.S. Moriano. Model Reference Adaptive Temperature Control of a Rotary Cement Kiln. 2018. 30, 36

[43] J. Hespanha, D. Liberzon, A. Stephen Morse, B.D.O. Anderson, T.S. Brinsmead, and F. De Bruyne. Multiple model adaptive control. Part 2: Switching. *International Journal of Robust and Nonlinear Control*, 11(5):479–496, 2001. 33

[44] P. Hoffmann, R. Schumann, T.M.A. Maksoud, and G.C. Premier. Virtual commissioning of manufacturing systems a review and new approaches for simplification. pages 175–181, 2010. 46

[45] R. Ibarra, S. Florida, W. Rodríguez, G. Romero, D. Lara, and I. Pérez. Attitude control of a quadrocopter using adaptive control technique. *Applied Mechanics and Materials*, 598:551–556, 2014. 60

[46] Petros Ioannou and Simone Baldi. Robust Adaptive Control. pages 1–22. December 2010. 27, 30, 37

[47] I.M. Jaimoukha, Z. Li, and V. Papakos. A matrix factorization solution to the H h_∞ fault detection problem. *Automatica*, 42(11):1907–1912, 2006. 21

[48] S. Kabir, M. Walker, and Y. Papadopoulos. Quantitative evaluation of Pandora temporal fault trees via Petri Nets. *IFAC-PapersOnLine*, 28(21):458–463, 2015. 22

[49] M. Kaddour, M.E.E. Najjar, Z. Naja, N.A. Tmazirte, and N. Moubayed. Fault detection and exclusion for GNSS measurements using observations projection on information space. pages 198–203, 2015. 23, 28

[50] M. Karpenko and N. Sepehri. Hardware-in-the-loop simulator for research on fault tolerant control of electrohydraulic flight control systems. volume 2006, pages 4645–4651, 2006. 16, 238

[51] M. Karpenko and N. Sepehri. Hardware-in-the-loop simulator for research on fault tolerant control of electrohydraulic actuators in a flight control application. *Mechatronics*, 19(7):1067–1077, 2009. 238

[52] Ogata Katsuhiko. *Modern Control Engineering*, volume 5. January 2009. 121

[53] Howard Kaufman, Itzhak Barkana, and Kenneth Sobel. *Direct Adaptive Control Algorithms: Theory and Applications*. Communications and Control Engineering. Springer-Verlag, New York, 2 edition, 1998. 2, 34, 37

[54] A. Kayihan and F.J. Doyle III. Friction compensation for a process control valve. *Control Engineering Practice*, 8(7):799–812, 2000. 9

[55] R. Khan, P. Williams, P. Riseborough, A. Rao, and R. Hill. Fault detection and identification a filter investigation. *International Journal of Robust and Nonlinear Control*, 28(5):1852–1870, 2018. 23

[56] M. Khatibi and M. Haeri. A unified framework for passive active fault-tolerant control systems considering actuator saturation and l_∞ disturbances. *International Journal of Control*, 92(3):653–663, 2019. 5, 20, 44

[57] J. Krystek, S. Alszer, and S. Bysko. Virtual commissioning as the main core of industry 4.0 Case study in the automotive paint shop. *Advances in Intelligent Systems and Computing*, 835:370–379, 2019. 2, 16, 43, 45

[58] T. Kumon, T. Suzuki, M. Iwasaki, M. Matsuzaki, N. Matsui, and S. Okuma. System identification using evolutionary computation and its application to internal adaptive model control. volume 3, pages 363–368, 2001. 37

[59] R. Lara, E. Lalinde, and M.T. Moreno. Phosphorescent platinum(ii) alkynyls end-capped with benzothiazole units. *Dalton Transactions*, 46(14):4628–4641, 2017. 16

[60] E. Lavretsky and N. Hovakimyan. Positive μ-modification for stable adaptation in dynamic inversion based adaptive control with input saturation. volume 5, pages 3373–3378, 2005. 36

[61] C.-H. Lee and C.-C. Teng. Identification and control of dynamic systems using recurrent fuzzy neural networks. *IEEE Transactions on Fuzzy Systems*, 8(4):349–366, 2000. 31

[62] J.I. Leon, S. Kouro, L.G. Franquelo, J. Rodriguez, and B. Wu. The Essential Role and the Continuous Evolution of Modulation Techniques for Voltage-Source Inverters in the Past, Present, and Future Power Electronics. *IEEE Transactions on Industrial Electronics*, 63(5):2688–2701, 2016. 16

[63] U. Leturiondo, O. Salgado, L. Ciani, D. Galar, and M. Catelani. Architecture for hybrid modelling and its application to diagnosis and prognosis with missing data. *Measurement: Journal of the International Measurement Confederation*, 108:152–162, 2017. 2, 43, 46

[64] Y. Liu, H. Ma, and H. Ma. Adaptive Fuzzy Fault-Tolerant Control for Uncertain Nonlinear Switched Stochastic Systems with Time-Varying Output Constraints. 2018. 23, 44

[65] Y. Liu, H. Ma, and H. Ma. Adaptive fuzzy fault-tolerant control for uncertain nonlinear switched stochastic systems with time-varying output constraints. *IEEE Transactions on Fuzzy Systems*, 26(5):2487–2498, 2018. 27

[66] M.T. Long, R.R. Murphy, and L.E. Parker. Distributed Multi-Agent Diagnosis and Recovery from Sensor Failures. volume 3, pages 2506–2513, 2003. 16

[67] C.J. Lopez-Toribio, R.J. Patton, and S. Daley. Supervisory fault tolerant system using fuzzy multiple inference modelling. pages 4381–4386, 2015. 22

[68] J. Lunze and J. Richter. Control Reconfiguration: Survey of Methods and Open Problems. 16, 17, 22, 24

[69] J. Lunze and J.H. Richter. Reconfigurable fault-tolerant control: A tutorial introduction. *European Journal of Control*, 14(5):359–386, 2008. 23

[70] H. Ma, Q. Zhou, L. Bai, and H. Liang. Observer-Based Adaptive Fuzzy Fault-Tolerant Control for Stochastic Nonstrict-Feedback Nonlinear Systems With Input Quantization. 2018. 23, 27, 44

[71] P.S. Maybeck and R.D. Stevens. Reconfigurable Flight Control Via Multiple Model Adaptive Control Methods. *IEEE Transactions on Aerospace and Electronic Systems*, 27(3):470–480, 1991. 38

[72] A.-R. Merheb, F. Bateman, and H. Noura. Passive and active fault tolerant control of octorotor UAV using Second Order Sliding Mode control. pages 1907–1912, 2015. 20

[73] A. Mirzaee and K. Salahshoor. Fault diagnosis and accommodation of nonlinear systems based on multiple-model adaptive unscented Kalman filter and switched MPC and H-infinity loop-shaping controller. *Journal of Process Control*, 22(3):626–634, 2012. 21

[74] Kevin L. Moore. *Iterative Learning Control for Deterministic Systems*. Advances in Industrial Control. Springer-Verlag, London, 1993. 39

[75] M.J. Morshed and A. Fekih. A Fault-Tolerant Control Paradigm for Microgrid-Connected Wind Energy Systems. *IEEE Systems Journal*, 12(1):360–372, 2018. 16

[76] K.S. Narendra and J. Balakrishnan. Adaptive control using multiple models. *IEEE Transactions on Automatic Control*, 42(2):171–187, 1997. 30, 38

[77] K.S. Narendra and C. Xiang. Adaptive control of discrete-time systems using multiple models. *IEEE Transactions on Automatic Control*, 45(9):1669–1686, 2000. 30, 38

[78] R. Nasiri and A. Radan. Adaptive robust pole-placement control of 4-leg voltage-source inverters for standalone photovoltaic systems: Considering digital delays. *Energy Conversion and Management*, 52(2):1314–1324, 2011. 34

[79] K. Nassiri-Toussi and W. Ren. Indirect adaptive pole-rarement control of mimo stochastic systems: Self-tuning results. *IEEE Transactions on Automatic Control*, 42(1):38–52, 1997. 30, 31, 34, 35

[80] M. Nazzal and H. Ozkaramanli. Directionally-structured dictionary learning and sparse representation based on subspace projections. pages 1606–1610, 2015. 23

[81] N.T. Nguyen. Model-reference adaptive control. *Advanced Textbooks in Control and Signal Processing*, (9783319563923):83–123, 2018. 60

[82] H. Niemann. A model-based approach for fault-tolerant control. pages 481–492, 2010. 21

[83] H. Niemann and N.K. Poulsen. Fault tolerant control - A residual based set-up. pages 8470–8475, 2009. 21

[84] H. Niemann and N.K. Poulsen. Control switching in high performance and fault tolerant control. pages 6205–6209, 2010. 21

[85] J. Niguez, S. Amari, and J.-M. Faure. Fault-Tolerant Control of Discrete Event Systems: Comparison of two approaches on the same case study. volume 2015-October, 2015. 23

[86] M. Norrlöf. An adaptive iterative learning control algorithm with experiments on an industrial robot. *IEEE Transactions on Robotics and Automation*, 18(2):245–251, 2002. 40

[87] M. Oppelt and L. Urbas. Integrated virtual commissioning an essential activity in the automation engineering process: From virtual commissioning to simulation supported engineering. pages 2564–2570, 2014. 47

[88] P. Pisu and G. Rizzoni. A comparative study of supervisory control strategies for hybrid electric vehicles. *IEEE Transactions on Control Systems Technology*, 15(3):506–518, 2007. 33

[89] J. Qi, Z. Wang, and Y. Shen. Fault-tolerant control and optimal fault hiding for discrete-time linear systems. pages 1368–1373, 2015. 23, 28

[90] R. Qi, G. Tao, B. Jiang, and C. Tan. Adaptive control schemes for discrete-time T-S fuzzy systems with unknown parameters and actuator failures. *IEEE Transactions on Fuzzy Systems*, 20(3):471–486, 2012. 44

[91] W. Ren, H. Yang, B. Jiang, and M. Staroswiecki. Fault recoverability analysis of switched nonlinear systems. *International Journal of Systems Science*, 48(3):471–484, 2017. 23

[92] J. Richter and J. Lunze. Reconfigurable control of Hammerstein systems after actuator faults. volume 17, 2008. 23

[93] J.H. Richter and J. Lunze. Reconfigurable control of Hammerstein systems after actuator failures: Stability, tracking, and performance. *International Journal of Control*, 83(8):1612–1630, 2010. 23

[94] Martin Riedmiller and Heinrich Braun. Direct adaptive method for faster backpropagation learning: The RPROP algorithm. pages 586–591, 1993. 31

[95] M. Rodrigues, M. Adam-Medina, D. Theilliol, and D. Sauter. Fault diagnosis on industrial systems based on a multiple model approach. volume 37, pages 107–112, 2004. Issue: 15. 38

[96] M. Rodrigues, D. Theilliol, and D. Sauter. Design of an active fault tolerant control and polytopic unknown input observer for systems described by a multi-model representation. volume 2005, pages 3815–3820, 2005. 5, 20, 38, 44

[97] J. Rodríguez, C. Calleja, I. Elorza, A. M. Macarulla, A. Pujana, and I. Azurmendi. A Methodology for Real-Time HiL Validation of Hydraulic-Press Controllers Based on Novel Modelling Techniques. *IEEE Access*, 7(1):110541–110553, December 2019. 54, 238

[98] J. Rodríguez, C. Calleja, I. Elorza, A. M. Macarulla, A. Pujana, and I. Azurmendi. On Fault Tolerant Control Systems: A Novel Reconfigurable and Adaptive Solutions for Industrial Machines. *IEEE Access*, 8(1):110541–110553, December 2020. 28

[99] Jorge Rodriguez-Guerra, Carlos Calleja, Aron Pujana, Iker Elorza, and Ana M. Macarulla. Fault-Tolerant Control Study and Classification: Case Study of a Hydraulic-Press Model Simulated in Real-Time. Turkey, 2018. 28

[100] D. Rotondo, F. Nejjari, and V. Puig. Passive and active FTC comparison for polytopic LPV systems. pages 2951–2956, 2013. 20, 44

[101] I. Sadeghzadeh, A. Mehta, Y. Zhang, and C.-A. Rabbath. Fault-tolerant trajectory tracking control of a quadrotor helicopter using gain-scheduled PID and model reference adaptive control. pages 247–256, 2014. 30, 36, 77

[102] D. I. Sagias, E. N. Sarafis, C. I. Siettos, and G. V. Bafas. Design of a model identification fuzzy adaptive controller and stability analysis of nonlinear processes. 37

[103] R. Sakthivel, C.K. Ahn, and M. Joby. Fault-Tolerant Resilient Control For Fuzzy Fractional Order Systems. 2018. 21

[104] R. Sakthivel, M. Joby, C. Wang, and B. Kaviarasan. Finite-time fault-tolerant control of neutral systems against actuator saturation and nonlinear actuator faults. *Applied Mathematics and Computation*, 332:425–436, 2018. 21

[105] M. Salimifard and H.A. Talebi. Robust output feedback fault-tolerant control of non-linear multi-agent systems based on wavelet neural networks. *IET Control Theory and Applications*, 11(17):3004–3015, 2017. 23, 28

[106] R. Salminen, A. Marttinen, and J. Virkkunen. Adaptive pole placement control of a pilot crane. volume 2, pages 313–318, 1991. 34

[107] A. Salvi, S. Santini, D. Biel, J.M. Olm, and M. Di Bernardol. Model reference adaptive control of a full-bridge buck inverter with minimal controller synthesis. pages 3469–3474, 2013. 34, 36

[108] Y. Salwa, B. Saida, and A. Kamel. Estimation and compensation of sensor fault for perturbed PWA systems. pages 214–216, 2017. 23

[109] P. Sarhadi, A.R. Noei, and A. Khosravi. Model reference adaptive PID control with anti-windup compensator for an autonomous underwater vehicle. *Robotics and Autonomous Systems*, 83:87–93, 2016. 31, 36

[110] M. Schluse and J. Rossmann. From simulation to experimentable digital twins: Simulation-based development and operation of complex technical systems. 2016. 50

[111] M. Schreier. Modeling and adaptive control of a quadrotor. pages 383–390, 2012. 37

[112] M. Schulte. Model-based integration of reusable component-based avionics systems - A case study. volume 2005, pages 62–71, 2005. 22

[113] D. Schwung, T. Kempe, A. Schwung, and S.X. Ding. Self-optimization of energy consumption in complex bulk good processes using reinforcement learning. pages 231–236, 2017. 44

[114] S. Sharifi, S.M. Rezaei, A. Tivay, F. Soleymani, and M. Zareinejad. Multi-class fault detection in electro-hydraulic servo systems using support vector machines. pages 252–257, 2017. 9

[115] J. Shin, S. Kim, and A. Tsourdos. Neural-networks-based Adaptive Control for an Uncertain Nonlinear System with Asymptotic Stability. 2018. 23

[116] S. Simani, S. Alvisi, and M. Venturini. Fault tolerant model predictive control applied to a simulated hydroelectric system. volume 2016-November, pages 251–256, 2016. 5, 20, 44

[117] J.-J.E. Slotine and W. Li. Adaptive Manipulator Control: A Case Study. *IEEE Transactions on Automatic Control*, 33(11):995–1003, 1988. 31

[118] M. Staroswiecki and D. Berdjag. Passive/active fault tolerant control for LTI systems with actuator outages. pages 2506–2511, 2014. 20

[119] M. Staroswiecki and Amani Moradi. Fault tolerance of distributed systems by information pattern reconfiguration in the publisher/subscriber communication scheme. pages 1975–1980, 2014. 23, 28

[120] J. Stoustrup and V.D. Blondel. Fault Tolerant Control: A Simultaneous Stabilization Result. *IEEE Transactions on Automatic Control*, 49(2):305–310, 2004. 21

[121] J. Stoustrup and V.D. Blondel. A simultaneous stabilization approach to (Passive) fault tolerant control. volume 2, pages 1817–1822, 2004. 21

[122] K. Sun, S. Sui, and S. Tong. Optimal adaptive fuzzy FTC design for strict-feedback nonlinear uncertain systems with actuator faults. *Fuzzy Sets and Systems*, 316:20–34, 2017. 23

[123] P. Swarnkar, S. Jain, and R.K. Nema. Effect of adaptation gain on system performance for model reference adaptive control scheme using MIT rule. *World Academy of Science, Engineering and Technology*, 46:620–625, 2010. 30

[124] X. Tan and J.S. Baras. Adaptive identification and control of hysteresis in smart materials. *IEEE Transactions on Automatic Control*, 50(6):827–839, 2005. 37

[125] J. Tang, D. Wang, Y. Polyanskiy, and G.W. Wornell. Defect Tolerance: Fundamental Limits and Examples. *IEEE Transactions on Information Theory*, 64(7):5240–5260, 2018. 23

[126] A. Tayebi. Adaptive iterative learning control for robot manipulators. *Automatica*, 40(7):1195–1203, 2004. 39, 40, 41

[127] A. Tayebi and C.-J. Chien. A unified adaptive iterative learning control framework for uncertain nonlinear systems. *IEEE Transactions on Automatic Control*, 52(10):1907–1913, 2007. 41

[128] M. Thiel, D. Schwarzmann, M. Schultalbers, and T. Jeinsch. Adaptive Model Recovery Anti-Windup for Output-Feedback Plants. *IFAC-PapersOnLine*, 50(1):11523–11528, 2017. 31, 35, 36

[129] S. Tong, T. Wang, and Y. Li. Fuzzy adaptive actuator failure compensation control of uncertain stochastic nonlinear systems with unmodeled dynamics. *IEEE Transactions on Fuzzy Systems*, 22(3):563–574, 2014. 31

[130] E. Tuci, M.H.M. Alkilabi, and O. Akanyeti. Cooperative object transport in multi-robot systems: A review of the state-of-the-art. *Frontiers Robotics AI*, 5(MAY), 2018. 16

[131] N. Vafamand, M.H. Khooban, T. Dragičević, and F. Blaabjerg. Networked Fuzzy Predictive Control of Power Buffers for Dynamic Stabilization of DC Microgrids. *IEEE Transactions on Industrial Electronics*, 66(2):1356–1362, 2019. 46

[132] M. Van, S.S. Ge, and H. Ren. Finite Time Fault Tolerant Control for Robot Manipulators Using Time Delay Estimation and Continuous Nonsingular Fast Terminal Sliding Mode Control. *IEEE Transactions on Cybernetics*, 47(7):1681–1693, 2017. 16

[133] M. Van, S.S. Ge, and H. Ren. Robust Fault-Tolerant Control for a Class of Second-Order Nonlinear Systems Using an Adaptive Third-Order Sliding Mode Control. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(2):221–228, 2017. 44

[134] A. Vargas-Martínez and L.E. Garza-CastaÃśón. Combining artificial intelligence and advanced techniques in fault-tolerant control. *Journal of Applied Research and Technology*, 9(2):202–226, 2011. 238

[135] Adriana Vargas Martínez and Luis E. Garza CastaÃśón. Artificial Intelligence Methods in Fault tolerant Control. 16, 22

[136] P. Vuilleumier. How brains beware: Neural mechanisms of emotional attention. *Trends in Cognitive Sciences*, 9(12):585–594, 2005. 31

[137] Wangguang, Z. Wang, D. Wang, Y. Li, and M. Li. A review on fault-tolerant control of PMSM. volume 2017-January, pages 3854–3859, 2017. 16

[138] J. Wei, Y. Zhang, M. Sun, and B. Geng. Adaptive iterative learning control of a class of nonlinear time-delay systems with unknown backlash-like hysteresis input and control direction. *ISA Transactions*, 70:79–92, 2017. 40, 41

[139] F. Westbrink and A. Schwung. Virtual Commissioning Approach based on the Discrete Element Method. pages 424–429, 2018. 45

[140] F. Xiao, W. Liu, Z. Li, L. Chen, and R. Wang. Noise-Tolerant Wireless Sensor Networks Localization via Multinorms Regularized Matrix Completion. *IEEE Transactions on Vehicular Technology*, 67(3):2409–2419, 2018. 16

[141] J.-X. Xu and B. Viswanathan. Adaptive robust iterative learning control with dead zone scheme. *Automatica*, 36(1):91–99, 2000. 39, 40

[142] H. Yang, B. Jiang, and M. Staroswiecki. Fault recoverability analysis of switched systems. *International Journal of Systems Science*, 43(3):535–542, 2012. 27, 44

[143] H. Yang, H. Li, B. Jiang, and V. Cocquempot. Fault Tolerant Control of Switched Systems: A Generalized Separation Principle. 2018. 23, 27, 44

[144] Y. Yang, S. Cheng, Y. Yin, and D.C. Wunsch. Containment control of heterogeneous systems with non-autonomous leaders: A distributed optimal model reference approach. *IEEE Access*, 6:60689–60703, 2018. 77

[145] X. Ye. Nonlinear adaptive control using multiple identification models. pages 488–491, 2008. 38

[146] H.A. Yousef, K. Al-Kharusi, M.H. Albadi, and N. Hosseinzadeh. Load frequency control of a multi-area power system: An adaptive fuzzy logic approach. *IEEE Transactions on Power Systems*, 29(4):1822–1830, 2014. 31

[147] M. Yu, H. Li, W. Jiang, H. Wang, and C. Jiang. Fault diagnosis and RUL prediction of nonlinear mechatronic system via adaptive genetic algorithm-particle filter. *IEEE Access*, 7:11140–11151, 2019. 238

[148] Q. Yu, Z. Hou, and R. Chi. Adaptive iterative learning control for nonlinear uncertain systems with both state and input constraints. *Journal of the Franklin Institute*, 353(15):3920–3943, 2016. 41

[149] D. Zhai, C. Xi, J. Dong, and Q. Zhang. Adaptive Fuzzy Fault-Tolerant Tracking Control of Uncertain Nonlinear Time-Varying Delay Systems. 2018. 23, 27, 44

[150] D. Zhang, Z. Wang, and S. Hu. Robust satisfactory fault-tolerant control of uncertain linear discrete-time systems: An LMI approach. *International Journal of Systems Science*, 38(2):151–165, 2007. 20, 21, 44

[151] J.-X. Zhang and G.-H. Yang. Prescribed performance fault-tolerant control of uncertain nonlinear systems with unknown control directions. *IEEE Transactions on Automatic Control*, 62(12):6529–6535, 2017. 16

[152] W. Zhang, Q. Zhao, H. Zhao, G. Zhou, and W. Feng. Diagnosing a strong-fault model by conflict and consistency. *Sensors (Switzerland)*, 18(4), 2018. 24

[153] D. Zumoffen and M. Basualdo. From large chemical plant data to fault diagnosis integrated to decentralized fault-tolerant control: Pulp mill process application. *Industrial and Engineering Chemistry Research*, 47(4):1201–1220, 2008. 22, 23

[154] D. Zumoffen and D. Feroldi. Analyzing plant-wide control structures for industrial processes. In *Process Control: Theory, Applications and Challenges*, pages 27–68. 2014. 22, 23