International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:13, No:2, 2019

Fault-Tolerant Control Study and Classification: Case Study of a Hydraulic-Press Model Simulated in Real-Time

Jorge Rodriguez-Guerra, Carlos Calleja, Aron Pujana, Iker Elorza, Ana Maria Macarulla

***Abstract*—Society demands more reliable manufacturing processes capable of producing high quality products in shorter production cycles. New control algorithms have been studied to satisfy this paradigm, in which Fault-Tolerant Control (FTC) plays a significant role. It is suitable to detect, isolate and adapt a system when a harmful or faulty situation appears. In this paper, a general overview about FTC characteristics are exposed; highlighting the properties a system must ensure to be considered faultless. In addition, a research to identify which are the main FTC techniques and a classification based on their characteristics is presented in two main groups: Active Fault-Tolerant Controllers (AFTCs) and Passive Fault-Tolerant Controllers (PFTCs). AFTC encompasses the techniques capable of re-configuring the process control algorithm after the fault has been detected, while PFTC comprehends the algorithms robust enough to bypass the fault without further modifications. The mentioned re-configuration requires two stages, one focused on detection, isolation and identification of the fault source and the other one in charge of re-designing the control algorithm by two approaches: fault accommodation and control re-design. From the algorithms studied, one has been selected and applied to a case study based on an industrial hydraulic-press. The developed model has been embedded under a real-time validation platform, which allows testing the FTC algorithms and analyse how the system will respond when a fault arises in similar conditions as a machine will have on factory. One AFTC approach has been picked up as the methodology the system will follow in the fault recovery process. In a first instance, the fault will be detected, isolated and identified by means of a neural network. In a second instance, the control algorithm will be re-configured to overcome the fault and continue working without human interaction.

***Keywords*—Fault-tolerant control, electro-hydraulic actuator, fault detection and isolation, control re-design, real-time.

I. INTRODUCTION

NOWADAYS, consumer tendency is towards increasing demand on high quality products manufactured in less time. Scientist and engineers look into more reliable systems capable of producing even though a failure appears on the manufacturing process. Furthermore, machines will be able to adapt their production cycle avoiding or correcting imperfections on the products without stopping. A new methodology has overcome focused on detect, isolate and adapt the manufacturing process when a harmful situation appears [1]- [3].

This methodology brings plenty of benefits. Plant complete shutdown will become a past issue, production losses will be minimized and primary services, such as power grids, transportation systems, water supplies and communications will not be interrupted [4], [5]. Benefits are not limited to economic refund. Machines are considered as one of the most risky components for humans in the industry because a malfunction endangers their life. With the methodology presented, this problem will be reduced and, eventually, it will disappear [6].

This last point will be critical in some manufacturing processes, for instance, nuclear and chemical plants. They set restrictive security measures with redundancy controllers as backups systems for faulty situations. Would not be interesting if they, instead, adapt to this faulty situation without human interaction? There are multiple approaches on modern control theories that deal with this situation, where Fault-Tolerant Control (FTC) could play a significant role [7].

In the previous paragraphs some examples about how FTC could improve system reliability have been mentioned. As they have shown, the most important approach of FTC is to keep the plant operating in spite of a fault, even if the process performance has suffered degradation. Other additional benefits are [1]:

- Plant availability and system reliability when a fault occurs.
- Prevent a single fault from turning into system failure.
- Reduce hardware redundancy in favour of using information redundancy to detect faults.
- System components reconfiguration with the aid of fault accommodation.
- FTC admits degraded performance keeping intact system availability.
- Reduce hardware investment replacing mechanical elements with virtual ones.

Due to these benefits, FTC has become common in flight control, aerospace systems, automotive engine systems and industrial processes [8]- [10], [13]. On these cases, the employed FTC utilizes redundancy to create an intelligent system that supervises the behaviour of the system components, increasing their reliability. Despite its multiple benefits, introducing this methodology increases the overall cost. Controllers need to be designed with more complex algorithms and some components need to be replicated [11] and [12]. In order to reduce the inversion in these fault-tolerant systems, a study must be done answering the following questions:

- How critical is the component?
- How likely is the component to fail?

Jorge Rodrguez-Guerra is with the Faculty of Engineering, University of Deusto, Spain (e-mail: jorge.rodriguez@ikerlan.es).

- How expensive is to make the component fault tolerant?

With this study, the components with more failure potential will be detected and measured. For example, if a component is likely to fail, but it is harmless for the industrial process, it is not necessary to duplicate it [2]. In these cases, for instance, the fault is avoided by reconfiguring the controller. Before starting designing the FTC, each fault must be classified. With this information, the programmer selects the most suitable fault-tolerant approach, Passive Fault-Tolerant Control (PFTC) or Active Fault-Tolerant Control (AFTC). The study developed on Section II-C reveals how AFTC is divided into two phases, one focused on detect, isolate and recover the fault and other on redesign the controllers.

The paper has been divided into three sections. In II Fault-Tolerant Control techniques are classified into active or passive. Afterwards, in Section III, these techniques are in-depth analysed. Section IV shows a case study about a FTC techniques applied to a hydraulic-press. Finally, Section V highlights the results acquired from the research.

II. CLASSIFICATION

This section introduces concepts and ideas from the field of Fault-Tolerant Control (FTC). On II-A, basic concepts and notions of FTC field are given. II-B will bring general information about FTC properties and requirements, while II-C is a classification of FTC techniques.

A. Terminology

Fault-Tolerant Control techniques are not new in the literature. Their first approach comes from the 1970s. They have been studied by control designers since that date, although they have not appealed the scientific community until recent years. With its aid, multiple approaches to these techniques have been studied and implemented. Nonetheless, they shared a common terminology [6], which is studied on the following points:

- **Faults:** Are considered the base element to analyse in FTC, because they are the main source of the malfunction in technical systems. Faults may be triggered by inherit machine elements, such as a pipeline leakage, or by inherit machine properties, such as mechanical fatigue in the cylinder. There are three locations in the system were faults can appear: actuator, plant and sensor. Each one belongs to one of the parts in which technical systems could be divided: **Plant faults:** Modify the dynamical I/O properties of the system. **Sensor faults:** Sensor readings are given wrong information. **Actuator faults:** Actuator behaviour on the plant is interrupted or modified.
- **Failures:** If after a fault the system performance is reduced, after a failure the system stops providing service, cancelling its completely availability. Faults and failures can occur both at the component level and at the aggregated system level. Preventing component faults, component failures or subsystem faults from becoming system failures is the aim of FTC.
- **Fault-Tolerance:** Two terms can be extrapolated, Fault-Tolerant Systems (FTS) and Fault-Tolerant Control (FTC). The first one denotes a controlled system which continues serving its purpose in spite of the appearance of a fault until it is repaired. The time and fault degree supported under this condition vary. The second one denotes a framework prepared to turn control loops into FTS, that is to say, search the design of automatic control laws with fault-tolerance in mind.
- **Tasks in fault-tolerant control:** Fault diagnosis and controller adjustment have been considered the two conceptual steps required to achieve FTC. Diagnosis is a technique prepared to detect, isolate and identify faults occurring while the system is running. Controller adjustment is defined as techniques for attenuating the effects of faults by adjusting the control loop in a suitable manner.

B. Requirements and Properties

As it has been previously exposed, faults may cause substantial damage on machinery and can be a potential risk for human life. In [2], Blanke et al. describes four notions related with fault-tolerant control defining the properties that must be followed by faultless systems:

- **Safety:** This term is linked with the absence of danger and it includes the control equipment prepared to protect a technological system from permanent damage. A system is called fail-safe when, in response to a critical signal, triggers a controlled shut-down to protect the machine.
- **Reliability:** This term refers to the probability that a system accomplishes its intended function for a specified period of time under normal conditions. Reliability studies do not bring information about the current fault, they focus on analysing the frequency in which the system is faulty. Fault-Tolerant Control does not affect directly to component reliability, but it improves system overall reliability.
- **Availability:** This term focus on the probability of a system to be operational when it was required. In this case, system availability could be influenced by maintenance policies.
- **Dependability:** It lumps together the previous properties: safety, reliability and availability, that is to say, a dependable system could be defined as a fail-safe system with high reliability and availability.

Attending to these notions, fault-tolerant systems are the ones with the property of avoiding the spread of a fault into a failure. Faults reduce system performance in different degrees, which brings the two views on faulty systems:

- **Fail-operational:** In the strict form, the performance remains the same.
- **Fail-graceful:** In a reduced form, the system remains in operation after faults have occurred.

These views divide systems into regions, as it is shown in Fig. 1. On the example, faults are divided into regions of performance. This graphical representation gives information about the four regions in which systems are divided. One of them defines a region where system performance is optimal, that is to say, without faults. When the fault appears, two approaches are possible. On the first approach the recovery is possible and the system returns to the region of required performance (fail-operational). On the second approach the system was unable to recover from the fault and it stays in the region of degraded performance (fail-graceful).

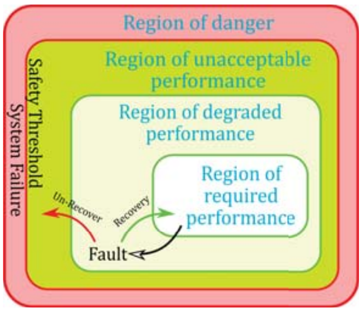

Fig. 1 Regions of performance: required and degraded

If the fault continues to spread across the system, it will reach the region of unacceptable performance. In this situation, the system stability is compromised and task fulfilment is below the optimal range. Even if it still works, the system has been degraded beyond the reliability and safety it should ensure. Fault-Tolerant Control should avoid this region.

When the safety criterion threshold is reached, the system enters the region of danger. This stage implies system failure, which means the system has been damaged. This region should never be reached. Due to the risk and endanger for humans when this region is reached, a safety system must be designed to interact on the threshold between the region of unacceptable performance and the region of danger independently to the fault-tolerant controller. It must be capable of avoiding a harmful situation for the user and it must satisfy security standards and policies independently of the FTC, as this controllers are designed with less restrictive safety criteria because they actuate in a less harmful region.

C. Classification

On the literature, Fault-Tolerant Control Systems can be classified into two techniques: active and passive (Fig. 2). There are multiple approaches defining these techniques, such as [2], [28], [15]- [20]. The following points summarize the differences between active and passive FTC (Fig. 3):

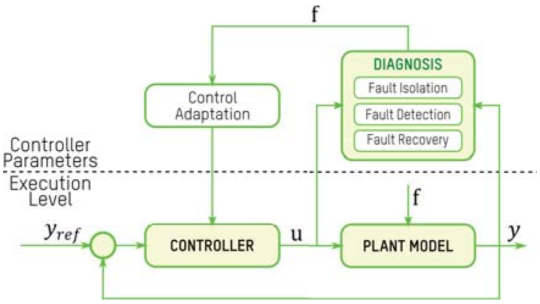

Fig. 2 Architecture for Active Fault-Tolerant Control Systems

- **Passive Fault-Tolerant Control (PFTC):** The controller is designed to tolerate changes in the plant and to continue satisfying its goals under all faulty conditions. In this case, FTC does not need to redesign control parameters when the fault appears. Indeed, this point is also their main problem. Their lack of adaptation makes this kind of controllers only suitable for specific changes in plant behaviour. In addition, even if the controller can detect and control the fault when it appears, their performance becomes reduced. It is also called *Robust Control*.
- **Active Fault-Tolerant Control (AFTC):** The system is capable of detecting and isolating the fault, and with this information the controller reconfigures their gains to the new situation. This principle is particularly efficient when the plant shows linear behaviour and slowly varying parameters. In addition, this kind of control is suitable for systems with multiple subsystems, because it isolates the fault and avoids its spreading to other ones. It is also called *Adaptive Control*.

As it has been explained, PFTCs are based on robust control and prepared to continue working when the fault appears without further modifications. The controllers are designed to guarantee stability and performance against a specific fault. Robust controllers include the following methodologies:

- **H_∞ Controller:** These controllers are prepared to minimize the H-infinity-norm in order to optimize the worst case of performance specifications [21], [22]. They are designed to be robust and stable dynamical compensators, as it has been exposed by [23], or with a Kalman filter to develop a robust controller with fault detection and isolation [24], [25].
- **Linear Matrix Inequalities (LMIs):** In this case, FTC is prepared to achieve robustness against actuator and sensor faults [26]- [29]. These controllers are prepared to solve convex problems with precise matrix constraints and are used with fuzzy logic, as it has been exposed in [30].
- **Simultaneous Stabilization:** Optimal solution when multiple plants controlled with only one controller needs to achieve stability in the presence of faults [31], [32].
- **Youla-Jabr-Bongiorno-Kucera (YJBK) parameterization:** This methodology has been presented in and mainly published by one author [33]- [35] and has been focused in stabilize controllers picking a feedback loop.

In contra-position to PFTC, AFTC can reconfigure the control system when a fault appears, keeping some properties of the original system. Active fault tolerance occurs in two stages: fault detection and accommodation will be the first one and controller reconfiguration will be the second one [1], [2].

FDI can be related with early detection, diagnosis, isolation, identification, classification and explanation of single and multiple faults. As it is shown in Fig. 4, FDI has been divided in the following methodologies:

- **Quantitative Model-Based:** Fault detection is based on the process model and dynamics analysed using mathematical equations. The unknown parameters are

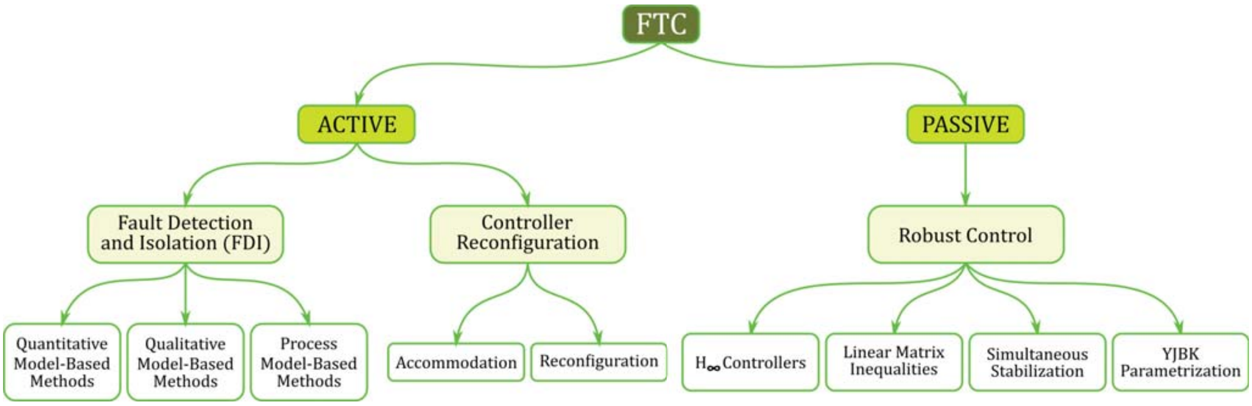

Fig. 3 Overview about the classification of Fault-Tolerant Control Techniques

calculated by estimative methods based on measures from input and outputs signals. This approach could be obtained with Kalman filters, observers and parity space [36], [37].

- **Qualitative Model-Based:** The fault is identified comparing physical and chemical properties of the system with empirical information. The methodology can be divided in abstraction hierarchies and causal models. The first ones are based in models showing how the component suffers degradation of its physical properties when a fault occurs. The second ones take the causal system structure and they present process relationships and at the same time they classified them in diagraphs, fault trees and qualitative physics [38], [39].
- **Process History-Based:** Compares the system behaviour with information that has been previously storaged. FTC needs a priori information about the system dynamics to identify and isolate the fault. In this case, quantitative and qualitative approaches are used. The first ones are divided in expert systems and trend modelling. The second ones in statistical methods prepared to recognize and classify the problem [40], [41].

When the fault has been detected and isolated, the controller needs to be re-designed [1], [2], [6]. There are two approaches (Fig. 5): fault accommodation and fault reconfiguration.

In fault accommodation, the input and output signals, which are manipulated by the controller, remains unchanged. Nonetheless, its gains may change, even with a modification in its dynamic order. The techniques used to configure this methodology are explained in the following points:

- **Adaptive Control:** In this case, the change on the adaptation phase only affects the control law. It will interact with system parameters that are dependent of time variables. In addition, it is capable of minimizing the error between the fault behaviour and the desirable one without pre-information about the parameters limits. Despite they are optimal for linear plants with slow parameter variations; their performance becomes degraded when the fault arises abruptly [42], [43].
- **Switched Control:** Multiple controllers (normal operation or fault) are designed with the goal to switch between them when the fault appears. Each fault situation must have its unique controller, which makes the necessity to design multiple control loops that can be unused [44], [45], [61], [47]. Despite the internal controller structure differs, they must refer to the same I/O signals.

In fault reconfiguration both the controller and its input and output signals may change. This active approach has been prepared to reconfigure the control loop structure in response

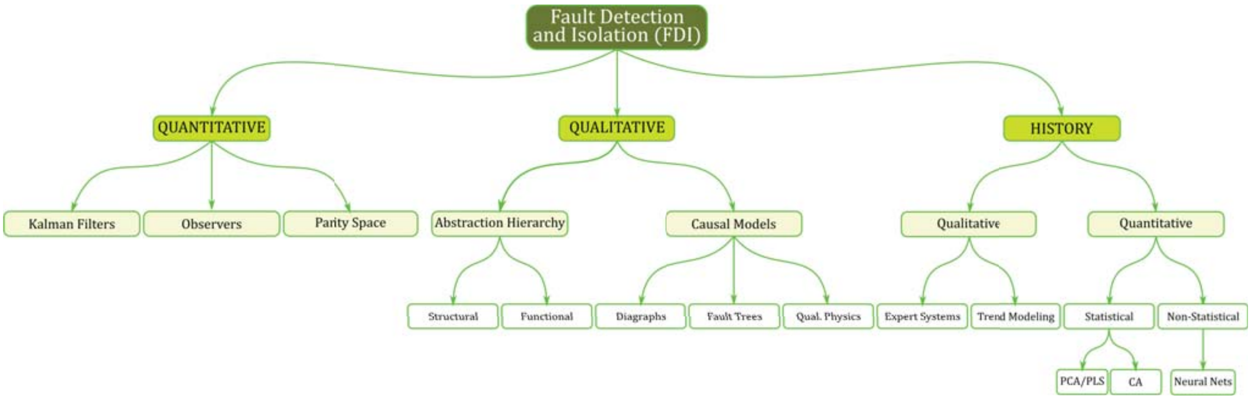

Fig. 4 Fault Detection and Identification (FDI) methodologies

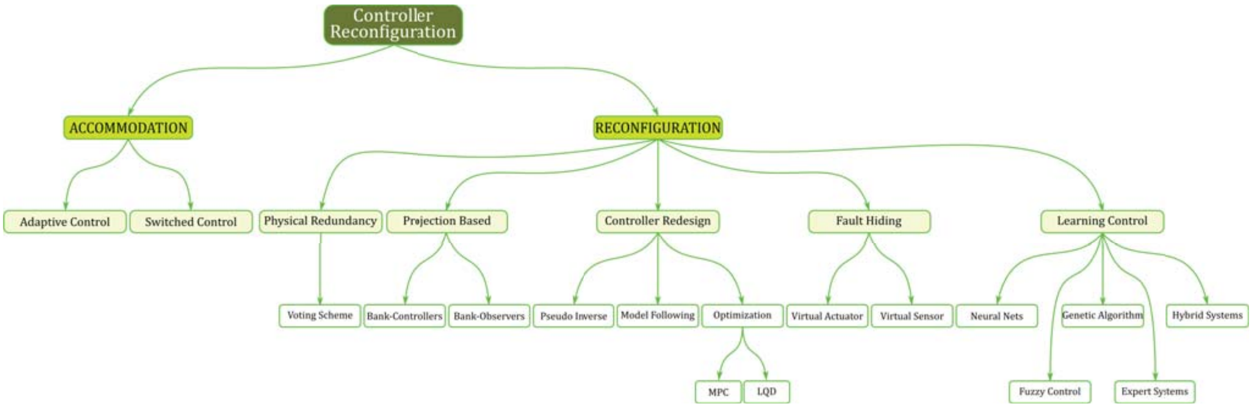

Fig. 5 Controller Reconfiguration methodologies

to faults. The techniques used to configure this methodology are explained in the following points:

- **Physical Redundancy:** The logic decision process is simplified because the fault tolerant behaviour is achieved with hardware redundancy (sensor and actuators). The control loop switches from the faulty component to a new one with identical characteristics [66], [67].
- **Projection Based Methods:** The control designers define a limited number of fault scenarios and, after, they design a control loop for each one. This methodology is based in creating a bank of controllers and a bank of observers. When the fault occurs, the closest predefined scenario is selected [52]- [54]. In this case, each bank of controllers has their own set of I/O signals
- **Controller Redesign:** When a fault occurs, controller switches their gains to continue achieving its objective. This technique has several approaches: pseudo inverse methods, model following and optimization [48]- [51].
- **Fault Hiding Methods:** The fault is hidden from the control loop by a reconfiguration block placed between the plant and the controller. This block masks the faulty signal, which allows the controller to continue working as if the fault does not exist. This method is developed using virtual actuators or sensors [55]- [58].
- **Learning Control:** The objective is to deal with faults before they appear. This methodology is based in Artificial Intelligence (AI) techniques, for instance neural networks [59], [60], fuzzy logic [42], [61], [62], genetic algorithms [63], [41], expert systems and hybrid systems. The advantage of these techniques relies on its ability to detect, identify and accommodate the fault.

III. DESIGNING A FAULT-TOLERANT CONTROL

As an example, we are going to study the steps required to develop a fault-tolerant control based on the previously explained active technique, afterwards, this methodology will be applied to a case study of a hydraulic-press. Active Fault-Tolerant Control seems as an optimal solution to overcome faults in these systems as control designers will attempt in vain to fulfil a controller robust enough for each faulty situation as a result of these systems combination of hydraulic, mechanic and electric components, which are prone to fail in unexpected manner.

This active technique requires a two step process to recover system nominal behaviour when the fault arises. Due to this fact, design a controller with this methodology requires accomplish both stages, that is to say, the control algorithm will be prepared to detect the fault source and, afterwards, recover from it. The following paragraphs explain the steps necessary to design a Fault-Tolerant Control based on this active technique. On Section III-A, the characteristics of fault-detection and isolation are studied, while Section III-B analyse control-redesign approaches.

A. *Fault Detection and Isolation*

The first task of fault-tolerant control concerns the identification and detection of existing faults. On Fig. 6 a graphical description about this problem is shown, which could be called the diagnostic problem. This is defined by means of the following idea: for a given I/O pair (U, Y), find the fault f [6], [64], [65].

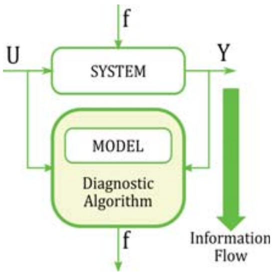

Fig. 6 Graphical description about the fault diagnosis

Diagnostic algorithm procedure has three steps to quantify the magnitude and locate the fault system:

- **Fault detection:** Discern if a fault has occurred. This step is crucial to diagnose the time at which the system is subject to some fault.
- **Fault isolation:** Detect which component has suffered the fault. This step focuses on determining the fault location.

- **Fault identification and fault estimation:** Identify and estimate the fault magnitude. This step is in charge of determining the kind of fault and its severity.

This procedure has been analysed and solved through multiple approaches, but they share a common principle, the consistency-based diagnosis. Even thought this is an abstract concept, it is easily explained with system behaviour theory. A general overview of this concept is shown in Fig. 7. System behaviour theory put forwards two necessaries concepts to detect a fault: the measurement information from the (U, Y) signals (input and output from the system, Fig. 6) and a set of equations modelling the system, which describes the relation between input and output sequences. System behaviour (β) is represented with this model and it will serve as a reference to discriminate if a fault has arisen.

This principle is based on analysing and comparing I/O measures of the system (U, Y) signals, with the nominal behaviour. The fault is detected when the I/O pair is not consistent with it, that is to say, it could be detected if $(U, Y) \notin \beta$. In this situation a fault has emerged on the system (f) and a new I/O pair appears (U_f, Y_f), which is consistent with the system faulty behaviour (β_f). This fault f is called a fault candidate.

In Fig. 7, three faults (f_0, f_1 and f_2) have been identified, each one with its corresponding behaviour (β_{f_0}, β_{f_1} and β_{f_2}). If the I/O pair is getting the results marked in points A, C or D the faults are easily determined as they belong to f_0, f_1 or f_2, respectively. In point B, the system is subjected to faults f_0 or f_1. The diagnostic algorithm cannot distinguish between faults leading to an ambiguity.

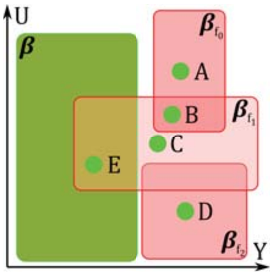

Fig. 7 Graphical representation of the system behaviour

As this ambiguity could not be discerned with the measured information, a question emerges about how a certain fault is detected. In the example (Fig. 7), point E belongs to a faultless behaviour(β) or a faulty one (β_{f_0}). These situation is deep studied by the field of *system diagnosability* or *fault detectability*. They define the diagnostic principle, which is named as the **consistency-based diagnosis**: when the system subject to the fault f and describes the behaviour β_f, the I/O pair satisfies the relation $(U_f, Y_f) \in \beta_f$. The system will determine what category the faults belong with further tests based on the following methodologies:

- **Fault detection:** When the I/O pair is inconsistent with the behaviour of the faultless system, $(U, Y) \notin \beta$, then a fault has occurred.
- **Fault isolation and identification:** When the I/O pair is consistent with the behaviour $(U_f, Y_f) \in \beta_f$, then a fault f may have occurred. In this cases, the fault must be studied to determine their source and measure how it affects the system.

This principle takes several assumptions:

- It is possible to detect the fault without information about its behaviour, because fault detection algorithms are based on nominal plant models.
- A fault model is required to identify the fault. This model brings information about how the fault spread through the system, allowing fault isolation and identification.
- If it is not possible to prove that a certain fault is present in the fault set, it must be excluded as a fault candidate to preserve consistency-based diagnosis.
- Not all faults can be measured and distinguished. Other approaches must be used to identify these faults.

The information recorded with fault diagnostic will be crucial in control reconfiguration phase. Active Fault-Tolerant Control benefits from the information brought by fault isolation and fault identification, because they make possible to set up a model of the faulty system and to facilitate control re-design phase.

B. Control Re-Design

On AFTC, after the fault has been detected and isolated, the controller is re-designed to adapt itself to the new situation. During this phase, the control structure is modified with the aim to satisfy the closed loop requirements despite the existence of a fault. This has been explained on Fig. 8, where the system behaviour is used to make an example about how the closed-loop must interact when a fault appears.

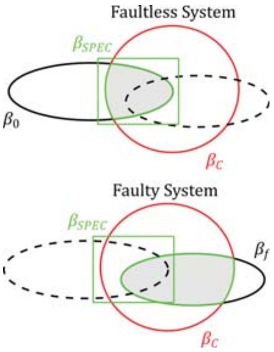

Fig. 8 System behaviour of the faultless (upper) and faulty (bottom) control closed-loop

The faultless system has the plant behaviour specified by β_0 and the controller behaviour specified by β_C. Both behaviours represent the multiple I/O pairs achieved in a conventional control loop. The I/O pairs suitable to achieve the system optimal performance are denoted as the behaviour β_{SPEC}. When the system is working without a failure, the consistent I/O pair that satisfies the control laws (the ones where β_0, β_C and β_{SPEC} collide) are described by the grey zone. This represents the nominal behaviour of the system, that is to say, the outputs achieved by the system when the plant satisfies the control law without a fault. This sentence is summarized with the following equation:

$$\beta_0 \cap \beta_C \subset \beta_{SPEC}$$

When the fault appears, the plant behaviour changes from the initial situation, to a new one, whose behaviour has been described in β_f. It substitutes the plant with a new model conditioned by the fault, changing the closed-loop system behaviour. In this new case, part of $\beta_f \cap \beta_C$ does not belong to the set β_{SPEC}, as it is demonstrated in the left picture of Fig. 8. Practically, most of the system behaviour (marked in grey) is out of β_{SPEC}. On this situation, a control re-design is required to deal with the fault, one accomplishing with:

$$\beta_f \cap \beta_C \subset \beta_{SPEC}$$

There are two approaches prepared to accomplish with the previous statement, which would be described in the following points:

1) Fault Accommodation: On this approach, the controller re-designed to find a new one matching with β_{SPEC} behaviour when a failure appears. On this scenario, the control parameters are adjusted to match the faulty situation, as it has been shown in Fig. 9. The controller rules must be kept identical, so no arbitrary gains can be used.

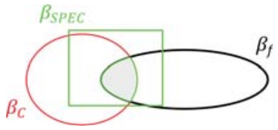

Fig. 9 System behaviour for fault accommodation

On *fault accommodation*, the controller parameters are adapted to the dynamical properties of the faulty system. As it is possible to see in Fig. 10, the input and output of the controller remains intact. After the diagnosis, the fault is detected and accommodated via an external loop.

One of the common approaches to work with this technique consists in pre-designing the control loop off-line for each possible fault. When a fault appears, the FTC swaps between controllers to the one designed for that case. This option is suitable for real-time applications, because the switch is done easily and fast. Despite these benefits, every FTC needs to be implemented during design or commissioning phase and they must be stored on disk memory, consuming a lot of computational resources.

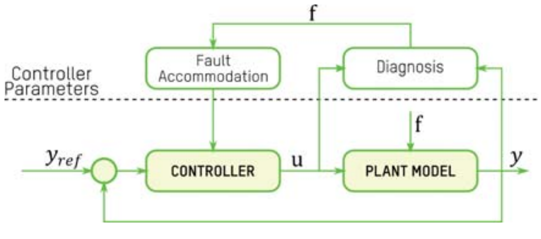

Fig. 10 Closed-loop schematic for fault accommodation

2) Control Reconfiguration: In contraposition with *fault accommodation*, in *control reconfiguration* there are no gains available to satisfy control rules and match β_{SPEC} behaviour (Fig. 11). In this situation, a new control configuration must be chosen. The signals under consideration will be adapted and, hence, the behaviour of the plant.

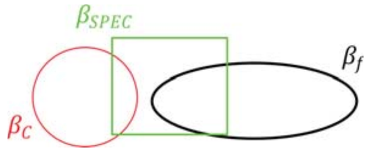

Fig. 11 System behaviour for control reconfiguration

When no accommodation is possible, the system needs a *control reconfiguration*. This technique requires adapt a new controller with a new pair of input and outputs. While the system is running, these signals must be chosen and the control law adapted, without shutting down the machine (Fig. 12).

This controller is suitable for sensor, actuator or plant failures. The first two cases are obvious, if the sensor or actuator fails and no replacement equipment is found, the controller needs to find alternative ones and keep the system under identical characteristics creating a new control law.

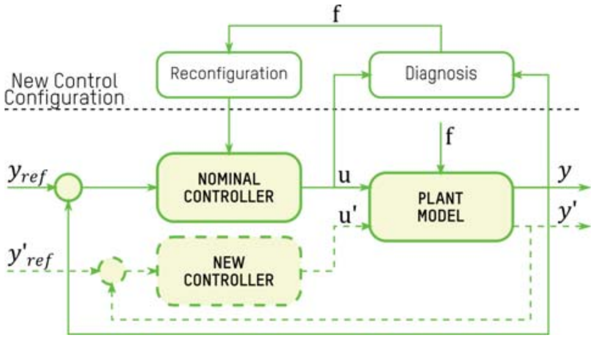

Fig. 12 Closed-loop schematic for control reconfiguration

However, applying these two techniques, *fault accommodation* and *control reconfiguration*, does not ensure fault avoidance. There are situations where neither the first technique nor the second technique is capable of finding a closed-loop controller that can avoid the instability. For instance, a plant that becomes unstable and unobservable due to faults. On these cases, the solution will pass for shutting down and restoring the machine.

IV. HYDRAULIC-PRESS

Nowadays, it is hard to find a manufacturing process without at least one electro-hydraulic actuator. Faults are as common as these machines are in industrial factories due to the combination of mechanical, hydraulic and electrical actuators and sensors, which are prone to fail. The fault arises without previous notification and in multiple shapes: a pipeline leakage, axial piston pumps working behind nominal conditions or proportional valves not responding to signals. On this context, a new research field has been identified in the scope of control for hydraulic-presses. A new methodology has born focused on detect, identify and isolate the fault in order to re-design the control algorithm embedded on the hydraulic-press.

Why has this methodology arisen? Hydraulic-actuators are widely used on manufacturing lines, due to their versatility and multiple tasks, such as stamp, displace loads or cut pieces. Despite their multiple benefits, they are prone to fail as they combine electric, hydraulic and mechanical parts. Even though most of these drawbacks are avoided with FTC techniques, as controllers are designed at the end of hydraulic-press commissioning process, study the faults and implement these solutions consumes a high quantity of man-hours.

On a previous work, a library prepared to simulate the hydraulic, mechanical and electrical behaviour of these actuators have been developed [68]. The library is capable of replicating the physical properties of hydraulic-presses at data sheet-level fidelity while they are simulated in real-time. Programmers brought off the task of design controllers and test them without the necessity of a real machine, with the aid of a hydraulic-press model embedded under a real-time validation platform.

Improvements in the field of control design are not restricted to just conventional controllers. With this methodology, control algorithms for hydraulic-presses are easily designed in laboratory conditions and tested early in the commissioning process. These benefits reduce the overall cost of the process, which leads to introduce new improvements in the design of hydraulic-press controllers, for instance, FTC techniques.

In the case studied on this paper, with the aid of a hydraulic-press model based on an existent machine (Appendix A), four fault situations have been simulated:

- **Pipeline Leakage:** In this case, hydraulic fed circuit has been compromised as the pipeline is loosing flow rate and pressure due to a crack. The simulations carried out represent from a small load lose produced by a scratch to more complex situations, for example, the complete breakdown of the hydraulic line.
- **Sensor Fault:** These components are prone to fail in two ways. Sensor collect a wrong measure (in extreme conditions, these represents its disconnection) or the signal receives external noise. Both situations have been contemplated in the experiment.
- **Proportional Valve Erroneous Opening:** With their opening, they regulate the flow rate through the cylinder, opening the valve proportionally to the control electric signal. When the component fails, its behaviour is compromised and the opening stops reacting as the manufacturer has specified in data-sheet characteristic curve.
- **Pump Failure:** As part of the fed circuit, their fault varies the flow rate contributed to the circuit, sets an erroneous pressure point or brings an abnormal power consumption.

Even though, these techniques have been experimented independently, they share a common root. The research has started from the same initial point, an existent model of a real hydraulic-press which has been adapted to simulate the mentioned faults [68]. This model has been picked up as the fautless controllers are already developed, that is to say, the hydraulic-press was fully controlled and the control algorithm is well-known. In normal behaviour situation (without fault), the model achieves a similar response as the real machine. From this initial point, the model is modified to become a test-bench to analyse the four faults previously mentioned.

Each fault requires their own adaptation in the hydraulic-press model, for instance, pipeline leakage has been simulated connecting into the hydraulic circuit an external branch regulated by a throttle and a cartridge valve. The first one controls the flow rate and pressure lost in the hydraulic circuit, while the second one enables the fault. This changes are appreciated in Fig. 13, which compares the initial hydraulic line against the modifications introduced to simulate the fault. With this technique, the controller designer has fully control of the fault, enabling and grading it depending on the experiment to be performed.

In each other fault case, the configuration methodology followed is similar to the one described above. The component has been adapted to simulate a fault situation, whose harmful degree is configurable and it is enabled by a command signal. This practice allows to simulate the hydraulic-press model in a faultless situation or a faulty one with the same model. When the hydraulic-press model is completely adapted, fault experiments will be carried out to identify, detect and isolate the failure source (as described in Section IV-A) in order to re-design the controllers in the following stage (as described in Section IV-B), accomplishing an Active Fault-Tolerant Control.

A. Neural Net

As it has been studied in Section III, AFTC technique first step corresponds with fault detection, isolation and identification process (Section III-A). In this case study, a FDI technique of *process history-based* kind was selected to analyse the fault and their source. This technique synergies with the methodology described in the previous paragraphs, as they identify the fault comparing system instant behaviour with information previously stored.

Hydraulic-press behaviour remains practically identical in each cycle, as the cylinder describes similar movements each time. Even though the properties remains the same, measures acquire from sensors vary slightly. In addition, faults do not follow a regular distribution, which difficulties the creation of a statistical algorithm containing each case. Due to this variations, a direct comparison between the historical data is hard and imprecise, so discriminate the fault requires a more complex algorithm, one based on artificial intelligent. Neural nets have been selected to seem suitable for this situation, as they compare the actual system behaviour and discern when the fault has arisen even if the signal has suffered some degradation. With this artificial intelligent algorithm, a qualitative analysis is developed, accomplishing each step in FDI process:

- **Fault detection:** As the hydraulic-press describes a non-discrete cyclic process, analyse each step time and measure in the simulation with the neural net requires

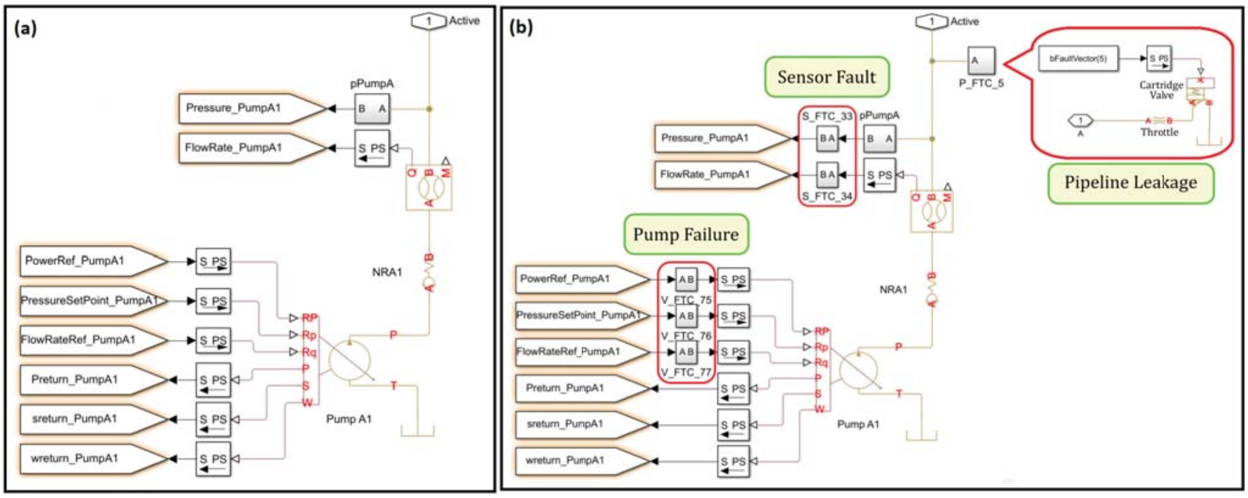

Fig. 13 Representation of the active pump hydraulic line in the model without fault (a) and with fault (b)

high computational cost. Instead, and in order to ensure the real-time capabilities, it has been selected some fixed points in the cycle where the fault will be studied. In this case study, it has been selected one point every five hundred milliseconds.

- **Fault isolation:** There are four types of faults well defined. The neural net has been prepared to recognize patterns, that is to say, it localizes the fault and indicates in which type of component the fault has arise.
- **Fault identification:** In a similar way as the previous point, the magnitude of each fault case is pre-configured by the user. Each type has been pre-defined with a severity, so when the net isolate the fault, it assign a grade of hardness to that type of fault.

With the methodology described in Section IV, create a fault historic database is accomplished activating one of the possible faults in each simulation. Each hydraulic-press cycle has been considered a simulation. Each iteration stores the hydraulic-press position and velocity, in addition to the value measure from every sensor installed on the machine and the control signals from the proportional valves and pumps. For nominal behaviour, pipeline leakage, proportional valve erroneous opening and pump failure two hundred and fifty experiments have been recorded for each one. In the case of sensor fault, it has been five hundred, distinguishing between a wrong measure on account of high noise or an inaccurate disconnection of the sensor.

Fig. 14 represents how the hydraulic-press cycle varies when the fault arises in each one of the previous experiments. In nominal behaviour, hydraulic-press accomplishes the cycle without any problem, nonetheless, in each fault case, the slide and cushion behaviour was modified. For instance, in situation (b), when one of slide pipelines suffers a leakage, it loses the pressure and flow rate, which makes the cylinder fall down. In situation (c), the controller is receiving a wrong measure from the sensor, so it gets stacked in a position that do not correspond with the desired value. Situation (d) shows how the controller is robust enough to perform the cycle, but the noise introduced in the sensor destabilizes the position, leading to vibrations in the cylinder position. In the proportional valve experiment, as it is shown in situation (e), the opening is below their nominal value, which reduces the flow rate from the pumps to the cylinder avoiding the slide downfall movement. Finally, situation (f) represents an over-flow in the pump, that is to say, it is giving more flow rate that the one demanded to perform the cycle. In this last situation, the cylinder remains in the upper position, its mechanical limit.

B. Control Re-Design

After the fault has been detected and isolated with the neural net, the controller is re-designed to avoid the faulty situation. In Section II-C, AFTC re-design techniques have been classified into fault accommodation or fault reconfiguration. Despite their benefits, Fig. 14 shows how the fault modifies system input and output signals in a way that the controller is not capable of tracking the reference signal, which led to introduce an active approach prepared to modify these signals in addition with the controller, that is to say, a fault reconfiguration technique which adapts the control loop structure in response to faults.

The study from Sections II-C and III-B2 revels *Projection Based Methods* as the control reconfiguration technique suitable for this case study. There are four fault scenarios well defined and identified by a neural net algorithm (bank of observers). Each scenario requires their own controller prepared to avoid the fault without modifying the reference (bank of controllers). For instance, in Fig. 14 case (d), when the fault arises, the control loop incorporates a filter to clean the noise signal, restoring a situation where the controller accomplishes hydraulic-press cycle. Fig. 15 shows how the system restores from a fault of the bank of experiments studied in case (e). In this case, the fault source is one of the proportional valves from the cushion, whose opening differs from the command signal requested by the controller. After the control reconfiguration phase, gains are adapted to the new

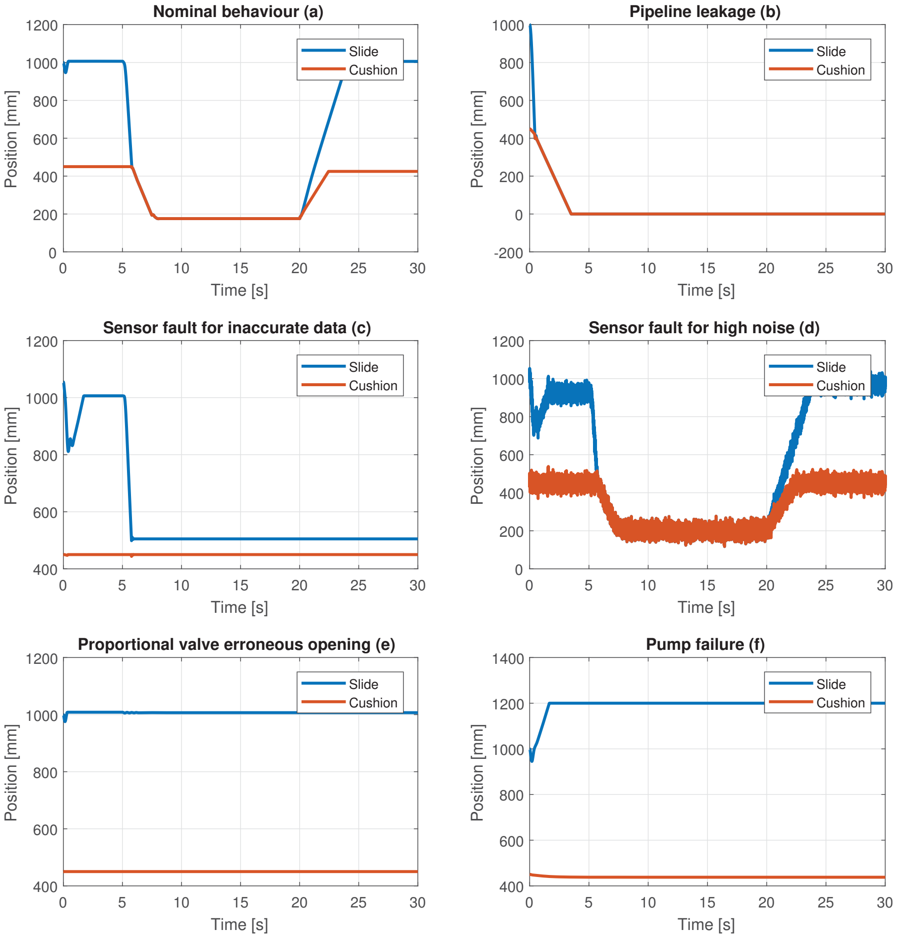

Fig. 14 Press cycle (position) in multiple situations: nominal behaviour (a), pipeline leakage (b), sensor fault for inaccurate data (c), sensor fault high noise (d), proportional valve erroneous opening (e) and pump failure (f)

situation and they are capable of reproducing hydraulic-press nominal behaviour.

The AFTC technique presented in the case study detects, isolates and identifies the fault source with the aid of a neural net trained by a historic of faults. With a hydraulic-press model, the simulations prepared to stored the fault database has been obtained harmless for the real machine. After the fault is detected, a Projection Based Method fault reconfiguration technique avoided it and the machine continues achieving their nominal behaviour.

V. CONCLUSION

Fault-Tolerant Control techniques increase manufacturing processes reliability, producing high quality products in shorter production cycles. Attending to this factors, it is expected an increase in the techniques dedicated to FTC and the number of manufacturing processes in which they will be applied.

Nowadays, there are plenty of approaches prepared to deal with the fault problem. These are classified into two main techniques, Passive Fault-Tolerant Control or Active Fault-Tolerant Control. The first one focused on design controllers robust enough to avoid the fault when it arises, while the second one is suitable when the application requires a more detailed study of the fault source and how to accommodate them. On this paper, both techniques have been classified seeking for their main common approaches.

Active Fault-Tolerant Control, has been deeply analysed due to their more complex fault identification and recovery process. It has been divided into two stages, fault detection and isolation and control redesign. On one side, FDI analysed the fault source, detect the faulty component, isolate that component and identify why the fault has arisen. On the other side, control redesign allows to accommodate the fault or reconfigure the control loop to restore the system to its nominal behaviour.

Finally, on this paper a case study about a hydraulic-press has been presented. It sought to prove a new methodology in which the fault is analysed by a real-time simulated hydraulic-press model based on a real machine. The machine accomplishes a cyclical process, which has been compromised with four fault cases. The results from each experiment have been processed and stored into a fault database. Afterwards, a neural net pattern recognition algorithm has been used to discriminate the fault, detect the source and isolate the

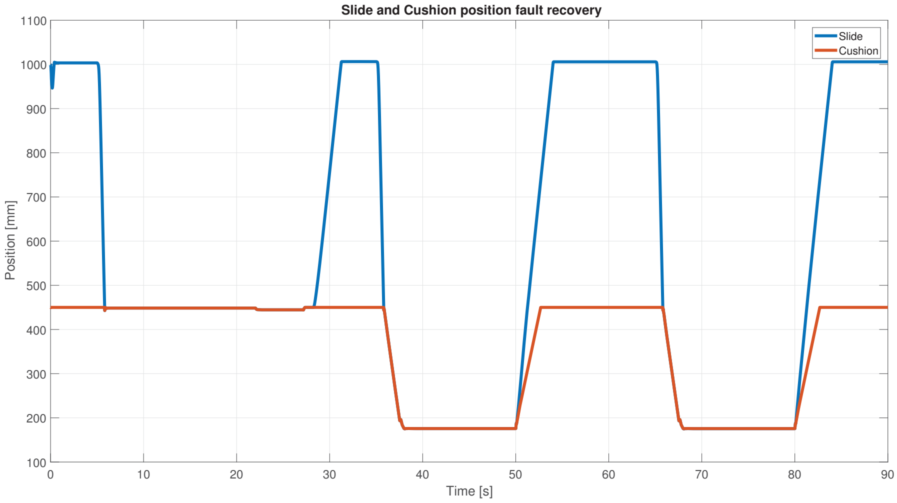

Fig. 15 Recovery process, the fault arises and automatically the hydraulic-press recover their normal behaviour

component. After this phase, the system overcomes the fault with an AFTC technique based on projection methods, which is capable of switching the controller between a bank of pre-defined control algorithms.

As a recapitulation, this paper classifies the most common approaches in FTC techniques, deeply analysing AFTC technique and introduces a new methodology based on neural net FDI with control re-design applied to a case study of a hydraulic-press.

APPENDIX A
HYDRAULIC-PRESS MODEL

The research starts from an existent hydraulic-press model based on a real machine. It has been installed in a manufacturing dedicated to metal sheet stamping. Their function consist in shaping the metal sheet into the desired piece applying force for an amount of time pre-configured by the user. The hydraulic-press has been build up by two main components:

- **Slide:** It is positioned in the top part and describes a downfall movement. It is made of one hydraulic cylinder feed with three axial piston pumps and the flow rate across each chamber is controlled by an independent proportional valve. In addition, there are multiple protection elements, such as relief valves and non-return valves.
- **Cushion:** It is positioned in the bottom part and describes an ascendant movement. The cushion is made of six passive cylinders (simple acting) and one active cylinder (double acting). Each one is controlled by an independent proportional valve and feed with four axial piston pumps. In a similar way as the slide, there are additional protection components.

The hydraulic-press describes a repetitive cycle in intervals of thirty seconds. During each interval, the slide moves from the rest position to cushion one, moment when both collide and become hitch. When they are hitch, they describe a brief and slow down movement and start making force against the metal sheet. When a timer triggers, both elements release and start recovering to their initial position. When both cylinders rest in this position, the controller waits for a short interval until initiating the next cycle.

REFERENCES

[1] A. Vargas Martinez and L. E. Garza Castaon, Artificial Intelligence Methods in Fault tolerant Control. 2014.

[2] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, Diagnosis and fault-tolerant control, third edition. 2016.

[3] 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, vol. 19, no. 7, pp. 10671077, 2009.

[4] M. J. Morshed and A. Fekih, A Fault-Tolerant Control Paradigm for Microgrid-Connected Wind Energy Systems, IEEE Systems Journal, vol. 12, no. 1, pp. 360372, 2018.

[5] 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, vol. 63, no. 5, pp. 26882701, 2016.

[6] J. Lunze and J. Richter, Control Reconfiguration: Survey of Methods and Open Problems, Research report of Institute of Automation and Computer Control Ruhr-Universit/at Bochum, Germany, 2006

[7] S. X. Ding, Model-based fault diagnosis techniques: Design schemes, algorithms, and tools. 2008.

[8] 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, vol. 5, no. MAY, 2018.

[9] Wangguang, Z. Wang, D. Wang, Y. Li, and M. Li, A review on fault-tolerant control of PMSM, presented at the Proceedings - 2017 Chinese Automation Congress, CAC 2017, vol. 2017-January, pp. 38543859, 2017.

[10] 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, vol. 62, no. 12, pp. 65296535, 2017.

[11] 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, vol. 47, no. 7, pp. 16811693, 2017.

[12] 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, vol. 67, no. 3, pp. 24092419, 2018.

[13] O. Goloubeva, M. Rebaudengo, M. Sonza Reorda, and M. Violante, Improved software-based processor control-flow errors detection technique, presented at the Proceedings - Annual Reliability and Maintainability Symposium, pp. 583589, 2005.

[14] 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, vol. 38, no. 2, pp. 151165, 2007.

[15] D. Rotondo, F. Nejjari, and V. Puig, Passive and active FTC comparison for polytopic LPV systems, presented at the 2013 European Control Conference, ECC 2013, pp. 29512956, 2013.

[16] M. Staroswiecki and D. Berdjag, Passive/active fault tolerant control for LTI systems with actuator outages, presented at the 2009 European Control Conference, ECC 2009, pp. 25062511, 2014.

[17] A.-R. Merheb, F. Bateman, and H. Noura, Passive and active fault tolerant control of octorotor UAV using Second Order Sliding Mode control, presented at the 2015 IEEE Conference on Control and Applications, CCA 2015 - Proceedings, pp. 19071912, 2015.

[18] S. Simani, S. Alvisi, and M. Venturini, Fault tolerant model predictive control applied to a simulated hydroelectric system, presented at the Conference on Control and Fault-Tolerant Systems, SysTol, vol. 2016-November, pp. 251256, 2016.

[19] M. Khatibi and M. Haeri, A unified framework for passiveactive fault-tolerant control systems considering actuator saturation and L disturbances, 2017.

[20] K. Ding, A. Morozov, and K. Janschek, Classification of hierarchical fault-tolerant design patterns, presented at the Proceedings - 2017 IEEE 15th International Conference on Dependable, Autonomic and Secure Computing, 2017 IEEE 15th International Conference on Pervasive Intelligence and Computing, 2017 IEEE 3rd International Conference on Big Data Intelligence and Computing and 2017 IEEE Cyber Science and Technology Congress, DASC-PICom-DataCom-CyberSciTec 2017, vol. 2018-January, pp. 612619, 2018.

[21] 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, presented at the Journal of Physics: Conference Series, vol. 555, 2014.

[22] 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, vol. 5, no. 4, pp. 938967, 2012.

[23] I. M. Jaimoukha, Z. Li, and V. Papakos, A matrix factorization solution to the H- / H fault detection problem, Automatica, vol. 42, no. 11, pp. 19071912, 2006.

[24] S. Dey, P. Pisu, and B. Ayalew, A Comparative Study of Three Fault Diagnosis Schemes for Wind Turbines, IEEE Transactions on Control Systems Technology, vol. 23, no. 5, pp. 18531868, 2015.

[25] 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, vol. 22, no. 3, pp. 626634, 2012.

[26] 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, vol. 18, no. 7, pp. 11511168, 2015.

[27] 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, presented at the European Wind Energy Conference and Exhibition, EWEC 2013, vol. 1, pp. 311317, 2013.

[28] 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, vol. 38, no. 2, pp. 151165, 2007.

[29] 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, vol. 332, pp. 425436, 2018.

[30] R. Sakthivel, C. K. Ahn, and M. Joby, Fault-Tolerant Resilient Control For Fuzzy Fractional Order Systems, 2018.

[31] J. Stoustrup and V. D. Blondel, A simultaneous stabilization approach to (Passive) fault tolerant control, presented at the Proceedings of the American Control Conference, vol. 2, pp. 18171822, 2004.

[32] J. Stoustrup and V. D. Blondel, Fault Tolerant Control: A Simultaneous Stabilization Result, IEEE Transactions on Automatic Control, vol. 49, no. 2, pp. 305310, 2004.

[33] H. Niemann, A model-based approach for fault-tolerant control, presented at the Conference on Control and Fault-Tolerant Systems, SysTol10 - Final Program and Book of Abstracts, pp. 481492, 2010.

[34] H. Niemann and N. K. Poulsen, Control switching in high performance and fault tolerant control, presented at the Proceedings of the 2010 American Control Conference, ACC 2010, pp. 62056209, 2010.

[35] H. Niemann and N. K. Poulsen, Fault tolerant control - A residual based set-up, presented at the Proceedings of the IEEE Conference on Decision and Control, pp. 84708475, 2009.

[36] C. J. Lopez-Toribio, R. J. Patton, and S. Daley, Supervisory fault tolerant system using fuzzy multiple inference modelling, presented at the European Control Conference, ECC 1999 - Conference Proceedings, pp. 43814386, 2015.

[37] S. Kabir, M. Walker, and Y. Papadopoulos, Quantitative evaluation of Pandora temporal fault trees via Petri Nets, IFAC-PapersOnLine, vol. 28, no. 21, pp. 458463, 2015.

[38] 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, vol. 5, no. 12, pp. 166184, 2008.

[39] M. Schulte, Model-based integration of reusable component-based avionics systems - A case study, presented at the Proceedings - Eighth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing, ISORC 2005, vol. 2005, pp. 6271, 2005.

[40] 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, vol. 47, no. 4, pp. 12011220, 2008.

[41] D. Zumoffen and D. Feroldi, Analyzing plant-wide control structures for industrial processes, in Process Control: Theory, Applications and Challenges, pp. 2768, 2014.

[42] 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.

[43] 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.

[44] W. Ren, H. Yang, B. Jiang, and M. Staroswiecki, Fault recoverability analysis of switched nonlinear systems, International Journal of Systems Science, vol. 48, no. 3, pp. 471484, 2017.

[45] H. Yang, H. Li, B. Jiang, and V. Cocquempot, Fault Tolerant Control of Switched Systems: A Generalized Separation Principle, 2018.

[46] Y. Liu, H. Ma, and H. Ma, Adaptive Fuzzy Fault-Tolerant Control for Uncertain Nonlinear Switched Stochastic Systems with Time-Varying Output Constraints, 2018.

[47] D. Zhai, C. Xi, J. Dong, and Q. Zhang, Adaptive Fuzzy Fault-Tolerant Tracking Control of Uncertain Nonlinear Time-Varying Delay Systems, 2018.

[48] R. Khan, P. Williams, P. Riseborough, A. Rao, and R. Hill, Fault detection and identificationA filter investigation, International Journal of Robust and Nonlinear Control, vol. 28, no. 5, pp. 18521870, 2018.

[49] L. Ferranti, Y. Wan, and T. Keviczky, Fault-tolerant reference generation for model predictive control with active diagnosis of elevator jamming faults, 2018.

[50] J. Lunze and J. H. Richter, Reconfigurable fault-tolerant control: A tutorial introduction, European Journal of Control, vol. 14, no. 5, pp. 359386, 2008.

[51] M. Staroswiecki and A. Moradi, Fault tolerance of distributed systems by information pattern reconfiguration in the publisher/subscriber communication scheme, presented at the 2014 European Control Conference, ECC 2014, pp. 19751980, 2014.

[52] Y. Salwa, B. Saida, and A. Kamel, Estimation and compensation of sensor fault for perturbed PWA systems, presented at the 2016 17th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering, STA 2016 - Proceedings, pp. 214216, 2017.

[53] M. Nazzal and H. Ozkaramanli, Directionally-structured dictionary learning and sparse representation based on subspace projections, presented at the 2015 23rd Signal Processing and Communications Applications Conference, SIU 2015 - Proceedings, pp. 16061610, 2015.

[54] 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, presented at the 2015 5th International Conference on Digital Information and Communication Technology and Its Applications, DICTAP 2015, pp. 198203, 2015.

[55] J. Qi, Z. Wang, and Y. Shen, Fault-tolerant control and optimal fault hiding for discrete-time linear systems, presented at the 2015 IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, IEEE-CYBER 2015, pp. 13681373, 2015.

[56] J. Niguez, S. Amari, and J.-M. Faure, Fault-Tolerant Control of Discrete Event Systems: Comparison of two approaches on the same case study, presented at the IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, vol. 2015-October, 2015.

[57] J. H. Richter and J. Lunze, Reconfigurable control of Hammerstein systems after actuator failures: Stability, tracking, and performance, International Journal of Control, vol. 83, no. 8, pp. 16121630, 2010.
[58] J. Richter and J. Lunze, Reconfigurable control of Hammerstein systems after actuator faults, presented at the IFAC Proceedings Volumes (IFAC-PapersOnline), vol. 17, 2008.
[59] J. Shin, S. Kim, and A. Tsourdos, Neural-networks-based Adaptive Control for an Uncertain Nonlinear System with Asymptotic Stability, 2018.
[60] 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, vol. 11, no. 17, pp. 30043015, 2017.
[61] Y. Liu, H. Ma, and H. Ma, Adaptive Fuzzy Fault-Tolerant Control for Uncertain Nonlinear Switched Stochastic Systems with Time-Varying Output Constraints, 2018.
[62] 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, vol. 316, pp. 2034, 2017.
[63] R. Abdul and R. Soundara, Adaptive dynamic genetic algorithm based node scheduling for time-triggered systems, Advances in Intelligent Systems and Computing, vol. 556, pp. 705714, 2017.
[64] J. dos Reis, C. Oliveira Costa, and J. S da Costa, Strain gauges debonding fault detection for structural health monitoring, Structural Control and Health Monitoring, vol. 25, no. 12, 2018.
[65] W. Zhang, Q. Zhao, H. Zhao, G. Zhou, and W. Feng, Diagnosing a strong-fault model by conflict and consistency, Sensors (Switzerland), vol. 18, no. 4, 2018.
[66] J. Tang, D. Wang, Y. Polyanskiy, and G. Wornell, Defect tolerance: fundamental limits and examples, 2017.
[67] M. T. Hamayun, C. Edwards, and H. Alwi, An output integral sliding mode FTC scheme using control allocation, Studies in Systems, Decision and Control, vol. 61, pp. 81101, 2016.
[68] J. Rodr/guez, C. Calleja, A. Pujana, I. Elorza, and I. Azurmendi, Real-time HiL for hydraulic press control validation, presented at the SIMULTECH 2017 - Proceedings of the 7th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, pp. 126133, 2017.