mathematics

MDPI

Article

A Novel Control Hardware Architecture for Implementation of Fractional-Order Identification and Control Algorithms Applied to a Temperature Prototype

Juan J. Gude [1,*] and Pablo García Bringas [2]

[1] Department of Computing, Electronics and Communication Technologies, Faculty of Engineering, University of Deusto, 48007 Bilbao, Spain
[2] Department of Mechanics, Design and Industrial Management, Faculty of Engineering, University of Deusto, 48007 Bilbao, Spain
* Correspondence: jgude@deusto.es

Abstract: In this paper, the conceptualization of a control hardware architecture aimed to the implementation of integer- and fractional-order identification and control algorithms is presented. The proposed hardware architecture combines the capability of implementing PC-based control applications with embedded applications on microprocessor- and FPGA-based real-time targets. In this work, the potential advantages of this hardware architecture over other available alternatives are discussed from different perspectives. The experimental prototype that has been designed and built to evaluate the control hardware architecture proposed in this work is also described in detail. The thermal-based process taking place in the prototype is characterized for being reconfigurable and exhibiting fractional behaviour, which results in a suitable equipment for the purpose of fractional-order identification and control. In order to demonstrate the applicability and effectiveness of the proposed control hardware architecture, integer- and fractional-order identification and control algorithms implemented in various control technologies have been applied to the temperature-based experimental prototype described before. Detailed discussion about results and identification and control issues are provided. The main contribution of this work is to provide an efficient and practical hardware architecture for implementing fractional-order identification and control algorithms in different control technologies, helping to bridge the gap between real-time hardware solutions and software-based simulations of fractional-order systems and controllers. Finally, some conclusions and concluding remarks are offered in the industrial context.

Keywords: control hardware; fractional-order controllers; fractional-order modelling; experimental equipment

MSC: 93C83; 93A30; 26A33

check for updates

Citation: Gude, J.J.; García Bringas, P. A Novel Control Hardware Architecture for Implementation of Fractional-Order Identification and Control Algorithms Applied to a Temperature Prototype. *Mathematics* **2023**, *11*, 143. https://doi.org/10.3390/math11010143

Academic Editors: Pengyu Chen, Haiyong Qin and Haoyu Niu

Received: 21 November 2022
Revised: 21 December 2022
Accepted: 21 December 2022
Published: 28 December 2022

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Enormous advances in technology have marked a new stage in the field of process control in the process industries, causing the need for universities and educational institutions to continually update themselves [1]. Alongside technological advances, laboratory equipment plays a significant role in control engineering research and education. The lack of adequate equipment to bring students closer to the reality in industry is one of the main problems encountered by a Faculty of Engineering in the practical education in the field of control engineering [2]. Due to this situation, there is a gap between the technical-practical and the theoretical education of the university graduates, which increases the deep-rooted distrust that companies have in academia because generally the practical training in these institutions does not meet industrial demands [3].

The past three decades have witnessed remarkable development in the use of fractional calculus in various fields, such as process control and modelling. It is now an important tool for the international industrial and scientific communities. In fact, the development of fractional calculus has enabled a major industrial and academic effort centered on the transition from conventional modelling and control to those described by fractional-order differential equations [4]. Another relevant monograph is [5], which covers the design and implementation of different kinds of fractional-order controllers.

In particular, in these last two references, experimental equipment and practical experience play an important role in demonstrating the usefulness and applicability of fractional calculus in the areas of modelling and control. However, there are not many process rigs that clearly and comprehensibly illustrate the advantages of fractional systems and their control and are therefore suitable for fractional-order modelling and control. As a review of such platforms, the following references can be found in the literature:

Malti and co-workers developed an aluminum heat transfer platform in [6] to investigate parameter and order estimation of fractional-order models. A similar platform has recently been presented in [7] to introduce fractional systems and fractional calculus as an effective tool for the study of both fractional-order modelling and control. Sierociuk and co-workers developed a metal beam with Peltier elements with a heating-cooling experimental platform in [8] to study heat transfer in heterogeneous media. Macias and Sierociuk studied fractional-order PID control on the same platform in [9]. Malek et al. used the Quanser-based heat flow experimental (HFE) platform to explore the modelling and control of fractional-order heat processes in [10]. The identification and control algorithms are implemented on a PC and the process is accessed through a data acquisition card. Reference [11] presents a low-cost hardware platform based on a Peltier cell for the modelling and control of multi-input multi-output (MIMO) fractional-order dynamic systems. In [12] the same platform is used as a single-input single-output (SISO) system to introduce students to fractional systems, and to demonstrate that such systems can be very effective in the modelling task. In both cases, an Arduino-type microcontroller is used as the control hardware.

From the above references, one can see that most of them use experimental platforms related to temperature or heat transfer to deal with fractional-order processes. However, the hardware used is generally not emphasized, but taken as ancillary.

Furthermore, it has also been shown that fractional calculus in the design of control systems results in controllers that are more efficient in comparison with traditional integer-order controllers [4]. However, the use of fractional-order control algorithms in industry is currently low, despite the fact that the fractional-order proportional integral derivative (PID) controller, as a generalization of the standard PID controller, provides significant benefits over the integer-order controller. Although the fractional-order PID controller was introduced in 1999, see [13], and since then there has been considerable development in design methods, tuning methods, and available software tools [14], the well-known implementation issues have contributed to the difficulties in conveying the advantages of fractional-order controllers in industry [15].

Moreover, the following references can be considered as a sample of recent research progress on fractional-order dynamical systems [16–18].

Therefore, considering the above, there is a need for laboratory equipment for training and researching in applied fractional calculus. To bridge the gap existing between theoretical fractional-order identification and control algorithms and their practical implementation in a control hardware device, this paper proposes a control hardware architecture that simplifies their real implementation on a real-time target. Training in implementation and tuning methods for fractional-order controllers is of major interest for making them more convenient and attractive for industry, thus facilitating their transition from state-of-the-art to state-of-use, see [19]. To the authors' knowledge, there is currently no work in the literature on hardware architectures such as the one presented in this paper.

Several integer- and fractional-order identification and control algorithms have been implemented in different control technologies in order to test the performance of the proposed control hardware architecture, which has been applied to a temperature-based experimental prototype that has recently been designed and built at the University of Deusto.

This paper is organized as follows. In Section 2, a description of the temperature experimental prototype is made, explaining the process that takes place and emphasizing its characteristics as a reconfigurable controlled process. The proposed control hardware architecture is presented in Section 3, detailing the different control technologies available and their potential capabilities for the practical implementation of fractional-order identification and control algorithms in each of them. In order to demonstrate the effectiveness of the control hardware architecture, Sections 4 and 5 present some results related to fractional-order identification and control, respectively, applied to the experimental setup. Finally, this paper is concluded in Section 6.

2. Temperature Experimental Prototype

This section presents the temperature-based experimental setup that is to be used to demonstrate the effectiveness and applicability of the control hardware architecture proposed in this work.

The section is divided into the following parts: First, the temperature prototype to be used in this paper is described. Then, a detailed description of the thermal process that takes place in the prototype is provided. Finally, as the thermal process can be configured with three different settings, its possibilities as a reconfigurable controlled process are considered in detail.

2.1. Description of the Prototype

The prototype can be divided into two clearly different parts, as shown in Figure 1.

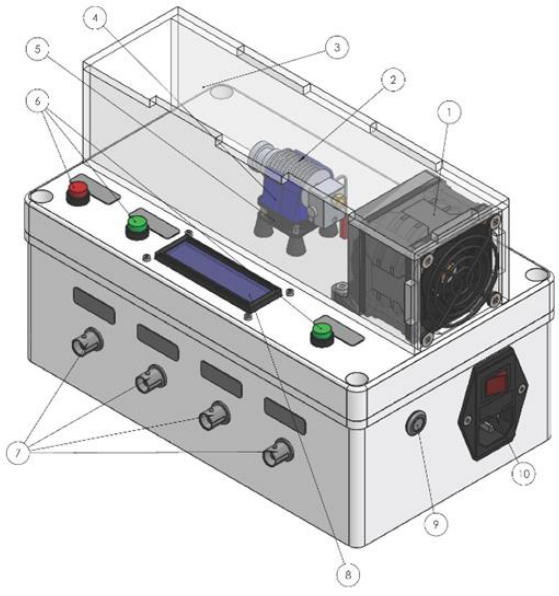


Figure 1. 3D-model layout of the Deusto Heater Experimental Setup.

- This prototype has a 3D-printer extruder head, which is inside a methacrylate duct, and an air fan installed in front of the hot end. Outside the enclosure there is an LCD display, LEDs, BNC output connectors for displaying process variables on an external device, such as an oscilloscope, and a user button. The list of the main external components of the prototype is shown in Table 1.

Table 1. Deusto Heater Experimental Setup—List of the main external components of the prototype.

#	Component
1	Air Fan
2	3D-printer extruder head
3	Acrylic duct
4	Plastic duct covering heat sink
5	Forced convection fan
6	User LEDs
7	BNC output connectors
8	LCD display
9	User button
10	IEC Power connector

- The power source and all electrical and hardware components required for the proper functioning of the prototype are located in the inner part of the enclosure. A 34-way standard IDC connector, located on one side of the enclosure, is also used to connect input and output signals to the proposed control hardware.

Deusto Heater Experimental Setup (Deusto HES) is the name of this equipment and has been built by using simple components, which makes this prototype affordable. Its technical characteristics, its size, and weight make it portable and suitable to work with it at home, which is a very important characteristic for a Faculty of Engineering [1].

This lab equipment has been designed and built in the Laboratory of Measuring Systems and Control, and in Deusto Fablab, both belonging to the Faculty of Engineering, University of Deusto. The total cost of the Deusto HES platform, without including the hardware device, is estimated to be around EUR 250.

2.2. Thermal-Based Process

Figure 2 details the various components of the extruder head, which is the area of the apparatus where the thermal process of interest takes place. The figure also illustrates the different thermodynamic phenomena taking place that are the origin of the fractional-order behaviour exhibited by the thermal-based process, as will be discussed later.

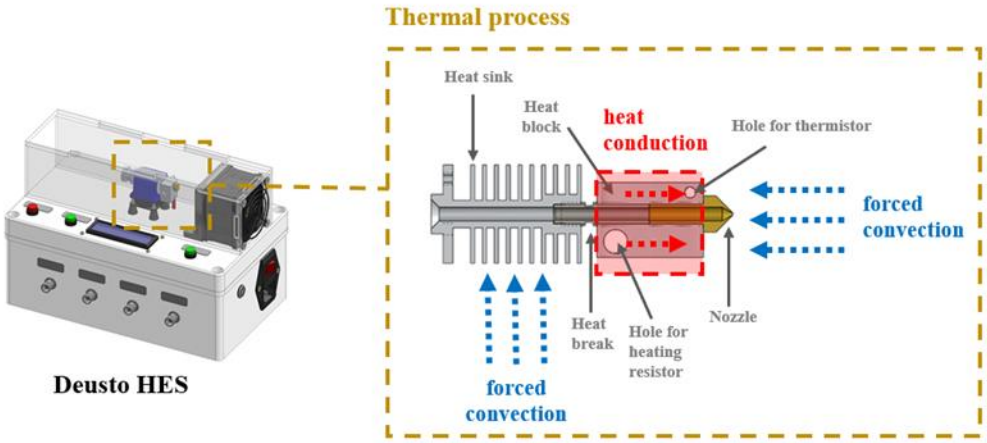


Figure 2. Deusto Heater Experimental Setup—Detail of the thermal process taking place in the extruder head.

It is important to note that no extrusion process takes place in the 3D-printer head, since it is only used as a heating element.

The four parts of the extruder head, which is shown in Figure 2, are described in the following way:

- Heat block. It is constituted by a high-conductivity aluminum. A heating resistor is embedded inside it, which provides the main heat inflow into the system, increasing the nozzle temperature. A thermistor-type temperature sensor is embedded in the block inside a second hole. Its main function is to monitor the evolution of the temperature in the extruder head.
- Nozzle. This part is heated by conduction from the heat block. In a real 3D-printer the hot extrusion is directly developed in it.
- Heat break. This is the first of the protection measures provided by the extruder head. This component minimizes heat conduction towards the heat sink, since it is constituted by a low-conductivity stainless steel. This is why it is directly attached to the heat block. Nevertheless, dissipation by forced convection is needed, since this barrier is generally not enough. For that purpose, the heat sink is covered using a plastic duct with a fan running at a constant speed.
- Heat sink. It is a finned surface that promotes heat dissipation by convection. This part, together with the aforementioned heat break, contributes to decrease the temperature along the extruder head body.

Figure 2 also shows the thermodynamic phenomena of heat conduction in the heat block due to the action of the heating resistor and forced convection due to the action of the air fan. Both, the airflow in the air fan and the heating power can be controlled by modifying the duty cycle of both pulse width modulation (PWM) signals sent by the control hardware.

2.3. Reconfigurable Controlled Process

The control objective is to bring the heat block temperature $T(t)$ to the desired value, T_{SP}, in spite of any disturbances that could affect the controlled process.

Considering that there is no change in the physical properties, only the influence of the air fan and the heating power of the resistance on the controlled variable $T(t)$ need to be taken into account.

Consequently, from a control point of view, the unique heating source in the process is the heat conduction generated by the heating resistance. Moreover, the forced convection of the air flow caused by the fan over the heat block is the other phenomenon involved in the thermal process [20,21].

Since both of these phenomena can be controlled using the control hardware, the configuration of the controlled process can be easily established by using software.

Therefore, taking into account all the above considerations, the controlled process can be configured to control the temperature in the heat block under the following three settings:

1. Configuration #1: The heating resistor acts as the final control element and the air fan speed is kept constant.
2. Configuration #2: The air fan acts as the final control element and the heating power of the resistance is kept constant.
3. Configuration #3: Using both, the heating element and the air fan as final control elements.

The main process variables including their units in each of the different configurations are those shown in Table 2.

Table 2. Main components and process variables depending on the controlled process configuration.

Components or Process Variables	Controlled Process Configuration #1	Controlled Process Configuration #2	Controlled Process Configuration #3
Controlled variable	Temperature in the heat block T(t) [°C]		
Manipulated variable(s)	Power delivered to the heat block by the heating resistance P(t) [W]	Airflow F(t) [m^3/min]	Power delivered to the heat block by the heating resistance P(t) [W] and airflow F(t) [m^3/min]
Measured variable(s)	Temperature measured by the thermistor T_m(t) [V] and Rotational speed of fan ω_F(t) [V]		
Control signal	Output of the controller u_H(t) [V]	Output of the controller u_F(t) [V]	Output of the controller u_H(t) [V] or u_F(t) [V]
Final control element(s)	Heating resistance	Air fan	Heating resistance and air fan
Measurement devices	Temperature transmitter (TT) and Frequency transmitter (ST)		
Disturbance(s)	Ambient temperature T_a(t) [°C], command signal to air fan u_F(t) [%]	Ambient temperature T_a(t) [°C], command signal to heating resistance u_H(t) [%]	Ambient temperature T_a(t) [°C], command signal to air fan u_F(t) [%] (when u_H(t) is applied) or command signal to heating resistance u_H(t) [%] (when u_F(t) is applied)

In configurations #1 and #2, a simple feedback control loop can be considered as the control structure, while in configuration #3 a split-range control can be used.

3. Control Hardware Architecture

The main objective of this section is to present a novel control hardware architecture that facilitates the implementation of integer- and fractional-order identification and control algorithms on a real-time target.

For that purpose, this section is divided into the following parts: First, a review of the different control hardware usually applied in industry is described. Then, an overview of the real-time target used in this paper and its technical capabilities and features are provided. Finally, the proposed control hardware architecture is described in detail, showing the interaction between hardware components and software.

3.1. Review of Control Hardware Technologies

Nowadays, as computers have become even more popular, control technology has progressed to be combined with computer technology for more accurate and faster methods of computation. Hence, new control methods for real-time systems are progressively being introduced in industry and taught in institutions. Generally speaking, one can find many different approaches to process control technology in the literature. These usually vary based on the software and hardware architecture used. The most frequent types can be divided into the following major categories, as shown in Figure 3. In addition to the specific references that will be included below in each section devoted to modelling and implementation of fractional-order dynamic systems, respectively, some recent and significant references for each hardware technology have also been included in each category in the context of this paper.

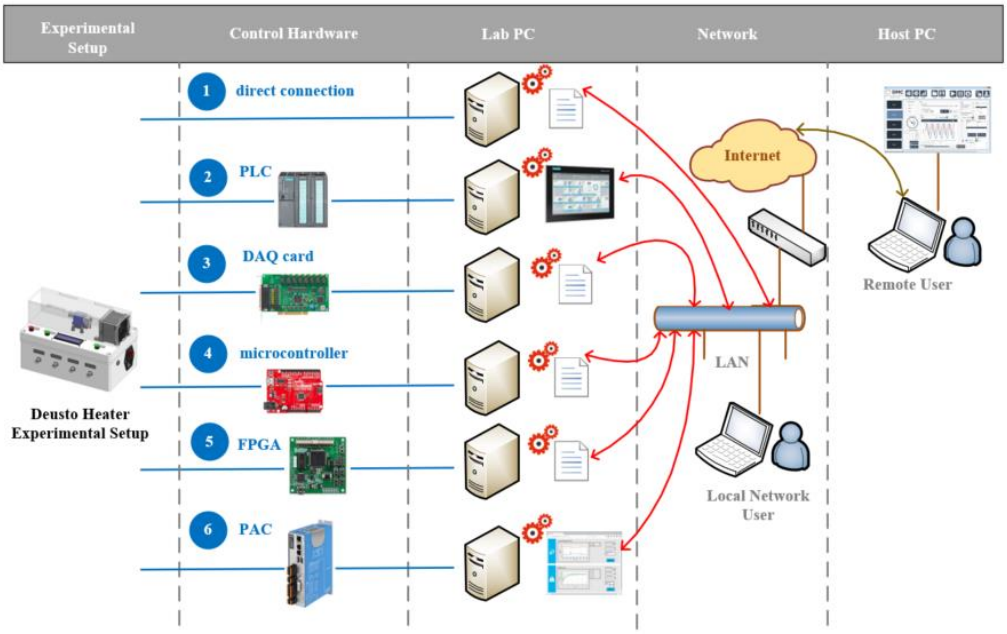


Figure 3. Review of the most frequently approaches to control hardware applied in industry.

1. Direct connection (Figure 3–Case 1). This type of architecture is commonly found when a direct connection between the experimental setup and the lab PC (e.g., by serial or USB port) is used. In this case, the experimental apparatus incorporates some form of data acquisition (DAQ) equipment and the appropriate software for controlling this device must be provided to the lab PC.
2. Programmable Logic Controller (PLC) (Figure 3–Case 2). A Supervisory Control And Data Acquisition (SCADA) system with a control hardware, generally a PLC or another hardware controller, is one of the most widely used industrial approaches. Most SCADA systems provide variable control capabilities and connectivity, data acquisition, and Human Machine Interfaces (HMI) as their main and common features. [22] describes the implementation of a fractional-order PI controller in a PLC, which is used for positioning of an electrohydraulic drive with servo valve. [23] describes the design, parameter tuning, and experimental evaluation of a PLC-based fractional-order PID temperature control in a pipeline with induced air-flow in a laboratory test bench. Regarding the development and implementation of tuning algorithms, the implementation of an auto-tuning method for fractional-order PID controllers using a PLC and a position servo motor has been carried out in [24].
3. DAQ device (Figure 3–Case 3). This architecture is often used in a situation when it is not possible to directly interconnect the experimental setup and the PC. In this case, the acquisition device, which is usually a DAQ card, is a separate part of the architecture. Proper software for data acquisition and control of the experimental setup must be provided to the lab PC. The following two references to works that fall into this category are given below: In [10], it is verified that a FFOPDT model represents a more accurate model than the standard FOPDT model for a Heat Flow Equipment (HFE). In this work, an integer order PID controller and two fractional order PI controllers have been designed for FFOPDT systems. Both fractional controllers guarantee robustness against loop gain variations. [25] develops a synthesis method of fractional-order PID controllers to meet five different design specifications for the closed-loop system, guaranteeing a robust behaviour of the controlled system against gain and noise changes. In addition, a self-tuning method for the fractional-order PID controller using the relay test has been proposed.

In recent years, industrial processes are controlled using other very popular approaches that are based on programmable targets, such as microcontrollers, cheap al-

ternatives to standard computers, and Field Programmable Gate Array (FPGA). Another common approach in industry is the use of certain Programmable Automation Controllers (PAC), which is an industrial technology focused on automated control, prototyping and measurement.

4. Microcontroller (Figure 3–Case 4). A standard microcontroller can be interfaced to the experimental setup and programmed to operate a control system or collect and perform data operations. In this case, the microcontroller usually has the appropriate hardware for DAQ and a lab PC with specific software is typically required. One of the first monographs devoted to the implementation of fractional-order controllers in various hardware devices is [26]. This reference includes not only this category, but also the following ones. Another more recent book where fractional-order PID control algorithms are implemented on a microprocessor-based real-time target is [4]. Ref [27] presents the time, frequency and real-time properties of a fractional-order PID implemented on a microprocessor-based real-time platform. The results obtained can be used in building embedded fractional control systems implemented on resource-constrained platforms.
5. FPGA (Figure 3–Case 5). The implementation of control systems on FPGA devices by using optimal hardware resources is one of the challenging research areas in control engineering. In this approach, a standard FPGA is used for operating and implementing real-time control systems for the experimental device. In this case, the FPGA provides interface and quick access to real-world I/O at several levels. Although a stand-alone application can be programmed in the FPGA, a lab PC with specific software is typically required for HMI operations. A recent monograph that provides a simple, step-by-step procedure for the implementation of fractional-order PID controllers on an FPGA-based real-time target is [28]. Other significant examples are the following: In [29], a fractional-order PI controller for DC motor speed control on an FPGA target is proposed and implemented. FPGA implementation issues and the advantages of using FPGA devices to implement robust fractional-order PI controllers are addressed. The same authors present in [30] the design of two advanced control algorithms are implemented on FPGA and applied to the speed control of a DC motor and evaluated in terms of closed-loop robustness, power consumption, execution times and resource minimization. In [31], a fractional-order PI controller is designed to control the speed of the permanent magnet synchronous motor (PMSM), and the accuracy of the numerical implementation of the fractional-order PI controller is evaluated. The accuracy of the numerical implementation of the fractional-order operator is investigated. In this paper, three common methods of numerical implementation of the fractional order operator are studied. In this paper, three common methods of numerical implementation of the fractional-order operator are studied, showing that the high-precision numerical implementation method of the designed fractional controller has better performance than the ordinary-precision fractional operation implementation method and the traditional integer-order PI controller. traditional integer-order PI controller.
6. PAC (Figure 3–Case 6). PAC devices combine the control reliability of a PLC with the monitoring and computational flexibility of a PC. Certain PACs often include technologies covered in categories 4 and 5. Although any other PAC could be used, a NI myRIO device has been selected in this paper. The approach followed in this paper to conceive and implement the proposed control hardware architecture exhibits enhanced flexibility, since the hardware architecture incorporates categories 3, 4, and 5 explained above. The main references for this category are provided in Section 5.

In general, in any of the considered approaches the lab PC can be connected to the Internet so that the control hardware can be easily accessed from a local or a host PC. In this way, any of these architectures allow the integration of the experimental device in a remote laboratory, as indicated, e.g., in [32].

3.2. Control Hardware Overview

In this paper, a NI myRIO device has been selected as a control hardware for operation of the temperature-based experimental setup and implementation of integer- and fractional-order identification and control algorithms in various control technologies, as will be explained later.

The NI myRIO 1900 is a reconfigurable and reusable device that features analog inputs (AI), analog outputs (AO), digital input and output lines (DIO), power output, and secondary digital functions (such as SPI, I2C, UART, quadrature encoder input, and PWM) in a compact embedded device, see [33].

It also features a 667 MHz dual core ARM Cortex-A9 programmable processor and custom Xilinx FPGA I/O which can be used by users to start system development and solve design problems faster.

The NI myRIO device allows final user to interface with the Zynq-7010 System on a Fully Programmable Chip (SoC) to develop real-time (RT) and FPGA-level applications.

The flexibility in the configuration and access to the available hardware that has been addressed in the design of the control hardware architecture allows to exploit the RT and FPGA capabilities of the device to implement identification and control algorithms, as will be discussed in the following sections.

Figure 4 represents a general scheme of the myRIO device used as a control hardware for operating and controlling the prototype. This figure also depicts the different control technologies and specific software that has been configured to be used as a part of the proposed hardware architecture.

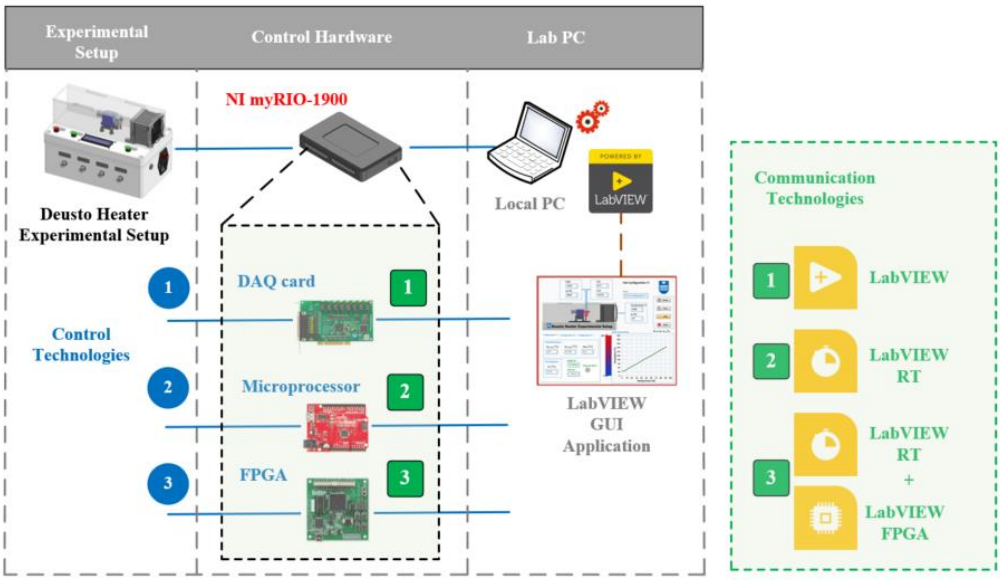


Figure 4. General scheme of the prototype using myRIO as a control hardware device. The different hardware technologies contained in the device and the software used are also depicted.

3.3. Proposed Control Hardware Architecture

The prototype is designed to naturally integrate a control hardware unit through a standard 34-pin IDC connector. Although any other control hardware or microprocessor could have been easily incorporated, a NI myRIO-1900 device has been used as a control hardware equipment.

Figure 5 shows in detail the scheme of the hardware components available in myRIO devices, showing the input and output (I/O) connections and the different interactions between hardware components and software.

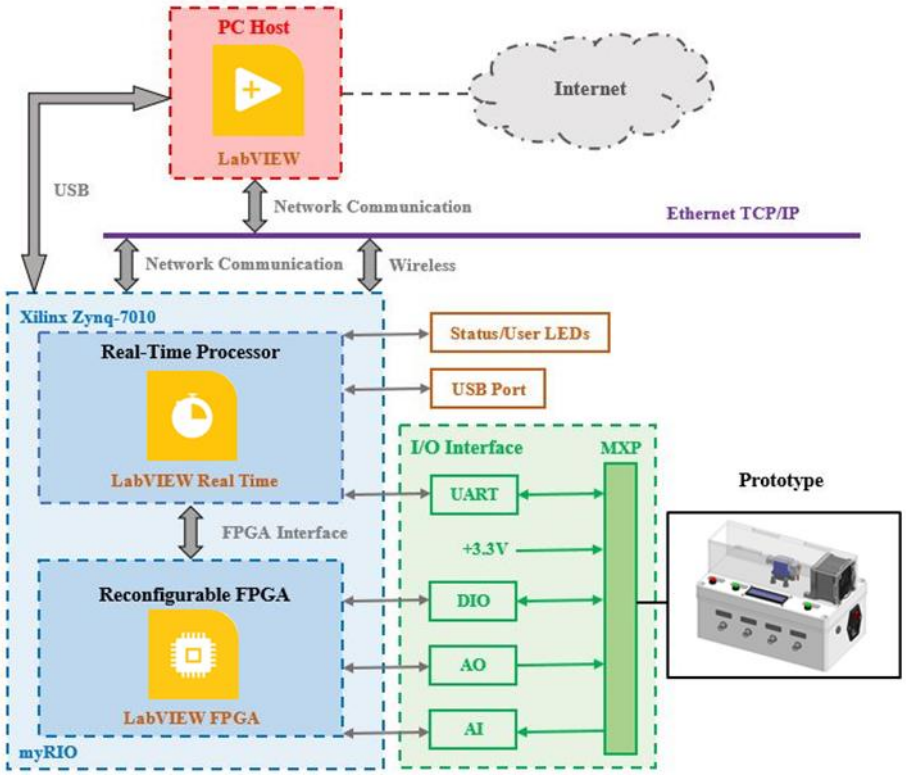


Figure 5. Proposal of hardware architecture applied to the laboratory prototype. The interaction between hardware and software is also illustrated.

The flexibility and transparency provided by this control hardware offer the ability to enable the following control hardware architecture for the practical implementation of advanced control and identification algorithms. Thus, hardware components are accessed using LabVIEW software to constitute the control hardware architecture that enables the following control modes or technologies [34]:

1. PC-based mode. In this mode, the identification or control algorithms can be implemented using a non-deterministic LabVIEW application on the user PC. Special functions that use the I/O functions of the myRIO toolkit have been implemented to access transparently the I/O signals of the myRIO device. This means that the control device is used as a DAQ card in this control mode.
2. Real-time (RT) microprocessor-based mode. This mode enables the RT processor of the myRIO device as a RT target to implement microprocessor-based deterministic applications. Identification or control algorithms are implemented by using LabVIEW RT and executed on the RT controller. LabVIEW RT also accesses the I/O ports through the FPGA interface and is responsible for communication with the host application, which is implemented in LabVIEW. A user can access or operate the control system from a local or remote PC.
3. FPGA-based mode. This mode enables the reconfigurable FPGA of the myRIO device as a RT target to implement FPGA-based deterministic applications. Identification or control algorithms are implemented using LabVIEW FPGA and executed on the reconfigurable FPGA. LabVIEW FPGA also is responsible to access the I/O ports through the FPGA interface. Communication between the FPGA and the local or remote PC is programmed in LabVIEW RT and executed in the real-time controller. A user can access or operate the control system from a local or remote PC.

Table 3 shows the signals from the output BNC connectors available for viewing on an oscilloscope. Tables 4 and 5 show the analog and digital input signals, and digital outputs, respectively, connected through the I/O Interface, as shown in Figure 5.

Table 3. Deusto Heater Experimental Setup—Signals in BNC output connectors in the prototype.

BCN Output Connector	Signal
BCN1	command signal to air fan u_F(t) [V]
BCN2	Rotational speed of Fan, ω_F(t) [V]
BCN3	command signal to heating resistance u_H(t) [V]
BCN4	Temperature measured by the thermistor in the heat block T_m(t) [V]

Table 4. Deusto Heater Experimental Setup—Analog and digital signals connected to the control hardware through the I/O interface.

Analog or Digital Inputs	Signal
AI0	command signal to air fan u_F(t) [V]
ENC.A	Rotational speed of Fan, ω_F(t) [V]
DIO13	User button

Table 5. Deusto Heater Experimental Setup—Digital output signals connected to the control hardware through the I/O interface.

Digital Outputs	Signal
PWM0	PWM signal to heating resistance u_H(t) [V]
PWM1	PWM signal to air fan u_F(t) [V]
DIO1	User LED 1
DIO2	User LED 2
DIO3	User LED 3

This section has presented the conceptualization of a hardware control architecture that combines the capability of implementing PC-based control applications with embedded applications on microprocessor- and FPGA-based real-time targets.

In what follows, the advantages of this hardware architecture over other available alternatives will be discussed.

Although there are several works in the technical literature that employ myRIO or compactRIO platforms to implement fractional-order PID controllers; see, e.g., [29,35,36], none of the works have defined and utilized their hardware in the manner presented in this paper. They are generally restricted to the implementation of the fractional-order PID algorithm on an FPGA-based RT target.

Different hardware platforms, generally those based on microprocessor or FPGA, have also been used in the literature to implement fractional-order PID control algorithms. See, e.g., [4,26,27]. While each hardware architecture has its unique characteristics, the one proposed in this paper provides some properties that are different from any other, such as:

a. It is a novel hardware control architecture that uniquely and efficiently exploits the different hardware available in myRIO device enabling their use indistinctly.
b. From the control hardware point of view, it features great flexibility since it can be configured to use the following technologies or control modes: PC-based, microprocessor-based, or FPGA-based control.
c. From the software point of view, it offers simplicity and transparency since the same programming language is used, regardless of the control hardware. Thus, instead of spending time developing user interfaces or debugging code syntax, end users can apply the LabVIEW graphical programming paradigm to focus on implementing their conventional or advanced identification and control algorithms without the added pressure of a complicated programming language for each hardware platform.
d. From a practical training perspective, the engineer has three hardware technologies on the same device for their use, without having to perform an experimental setup for each hardware platform to be used.

Hence, discussion on these directions involves novel interests.

The next two sections will evaluate the effectiveness and applicability of this hardware control architecture for implementing fractional-order identification and control algorithms.

4. Modelling Issues and Results

Fractional calculus has been considered for a long time as a pure mathematical field without many real applications. However, such a state of affairs has been changed in recent decades. In fact, it has been proven that fractional calculus can be useful and even powerful, especially in the area of modelling [4].

There is increasing evidence that certain dynamic processes can be described more accurately by using fractional-order models, see the applications in [37,38].

If the process reaction curve exhibits fractional behaviour as described in [39], as in the case of processes involving thermal conduction, the use of a FFOPDT model has clear advantages over integer-order models [40,41].

In this section, issues related to modelling of the thermal-based controlled process are described. Consequently, for the purposes of this paper related to modelling considerations, it is sufficient to consider the dynamic and static characteristics of the controlled process and estimating integer- and fractional-order models at a certain operating point by using process information collected from the reaction curve.

This section is divided into the following parts: First, the Fractional First-Order Plus Dead-Time (FFOPDT) model identification method to be applied to the laboratory prototype is briefly presented. Then, the experimental results obtained by implementing the aforementioned identification method on a microprocessor-based hardware and applied to the temperature-based controlled process under configurations #1 and #2, respectively, are presented. These results are compared with those obtained using well-recognized identification methods for integer-order models. Next, the different applications implemented in LabVIEW to characterize the process and implement the fractional-order model identification methods under an industrial hardware platform and in the laboratory prototype are presented. Finally, some final remarks and discussion about modelling issues in this context are provided.

4.1. FFOPDT Model Estimation

This section deals with the FFOPDT model identification procedure, which is summarized below, to be used with the thermal-based laboratory setup.

The transfer function of a standard FFOPDT model is:

$$P(s) = \frac{Ke^{-Ls}}{1 + Ts^{\alpha}} \tag{1}$$

where K is the static gain, $T > 0$ is the time constant, $L \geq 0$ is the apparent dead-time, and α is the model non-integer order.

Therefore, the FFOPDT model parameters to be determined are $\theta_P = \{K, T, L, \alpha\}$.

In the context of a fractional-order model, the average residence time T_{ar} can be defined as:

$$T_{ar} = \frac{\int_0^{\infty} tg(t)dt}{\int_0^{\infty} g(t)dt} = L + T^{\frac{1}{\alpha}} \tag{2}$$

where g(t) is the impulse response of the system, and T_{ar} is essentially a rough measure of how long it takes the input to have a significant influence on the output [42].

The FFOPDT dynamics can be fully characterized considering two dimensionless parameters: the normalized fractional dead-time τ_α and the fractional order α, which is defined as:

$$\tau_\alpha = \frac{L}{T_{ar}} = \frac{L}{L + T^{\frac{1}{\alpha}}} \tag{3}$$

Note that processes with small τ_α are easy to control, and the difficulty in controlling the system increases as τ_α increases.

The FFOPDT model (1) is considered as a generalization of the conventional FOPDT, which has been widely applied in practice to describe approximately the dynamics of many industrial processes for the purpose of control design [43]. Although with a FFOPDT model it is possible to characterize the overdamped and underdamped behaviour of the process, the focus in this paper will only be on processes characterized by an S-shaped response. These types of processes exhibiting such essentially monotone step responses are very common in process control [43].

If a step-input signal u(t) is applied to a FFOPDT model (1), an output signal $y_\alpha(t)$ is obtained, as shown in Figure 6. The FFOPDT model response to a Δu step-input change is:

$$y_\alpha(t) = K\left\{1 - E_{\alpha,1}\left[-\frac{1}{T}(t-L)^\alpha\right]\right\}\Delta u \tag{4}$$

where $E_{\alpha,\beta}$ is the two-parameter Mittag–Leffler function, which is defined in [40], $t \geq L$, and the output signal variation is $\Delta y = K \cdot \Delta u$.

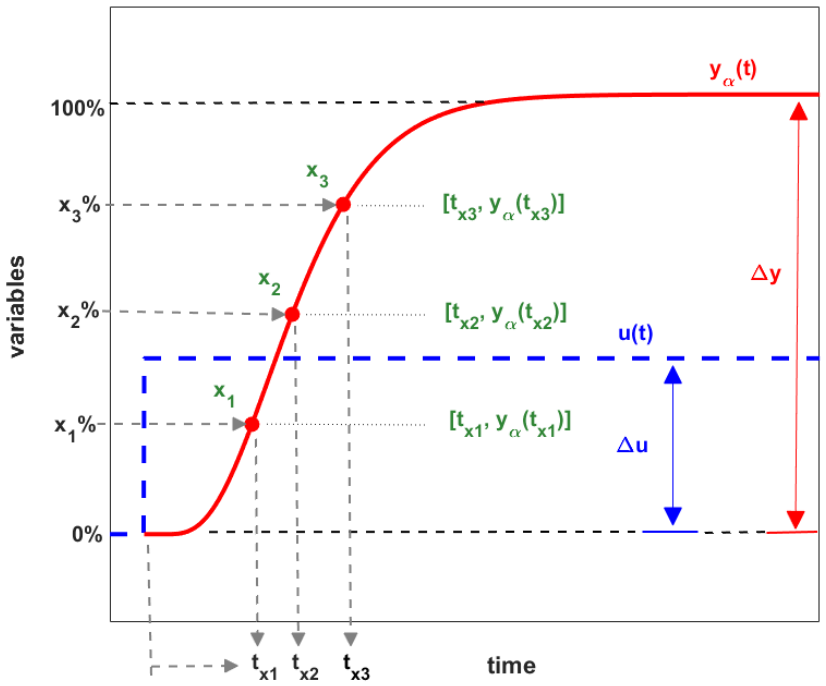


Figure 6. Step-input signal u(t), process reaction curve $y_\alpha(t)$, and arbitrary representative points $\{x_1, x_2, \text{and } x_3\}$ on the process reaction curve. $\{\Delta y, \Delta u, t_{x1}, t_{x2}, t_{x3}\}$ are the parameters taken from the process reaction curve required for the identification.

The following expression is obtained by normalizing the process output $y_\alpha(t)$ with respect to its final value $\Delta y = K \cdot \Delta u$ and making use of the normalized and shifted time $\tau = \frac{1}{T}(t-L)^\alpha$:

$$\tilde{y}_\alpha(\tau) = 1 - E_{\alpha,1}(-\tau), \quad \tau \geq 0 \tag{5}$$

where $0 \leq \tilde{y}_\alpha(\tau) \leq 1$.

Parameter τ_x is the normalized time required for the normalized process output $\tilde{y}_\alpha(\tau)$ to reach x% of the total output change and can be calculated from (5). Thus, the time t_x required for the process output (4) to reach such x-point is defined as follows:

$$t_x = L + (\tau_x T)^{\frac{1}{\alpha}} \tag{6}$$

A general procedure for identifying a FFOPDT model has recently been presented in [42]. This procedure is based on fitting three arbitrary points $\{x_1\text{-}x_2\text{-}x_3\%\}$ on the process reaction curve, where the process information is obtained from a simple open-loop test. A simplification of the general identification procedure has also been considered in [44], particularizing for the case where only symmetrically located points $\{x\text{-}50\text{-}(100-x)\%\}$ on the reaction curve are selected.

According to [45], the accuracy of the identified fractional-order model is also sensitive to the position of the central point x_2 within the set of symmetrical points on the process reaction curve. It has been discussed how a more accurate identified model can be obtained and new insights have also been offered on this selection of the central point x_2 in the context of the symmetrical procedure proposed in [44].

The following set of equations is used to determine the FFOPDT model parameters, $\theta_P = \{K, T, L, \alpha\}$, by using data information from the process reaction curve $\{\Delta y, \Delta u, t_{x1}, t_{x2}, t_{x3}\}$:

$$\begin{cases} K = \frac{\Delta y}{\Delta u} \\ \alpha = f_1(\Delta) \\ T = f_2(\alpha)(t_{x3} - t_{x1})^{\alpha} \\ L = \max\left[t_{x3} - f_3(\alpha)T^{1/\alpha}, 0\right] \end{cases} \tag{7}$$

where Δu is the step-input amplitude, Δy is the total process output change, and set of times $\{t_{x1}, t_{x2}, t_{x3}\}$ are times required to reach $x_1\%$ $(y_\alpha(t_{x1}))$, $x_2\%$ $(y_\alpha(t_{x2}))$, and $x_3\%$ $(y_\alpha(t_{x3}))$ of the total process output change on the process reaction curve, respectively, as shown in Figure 6.

Note that $f_1(\Delta)$, $f_2(\alpha)$ and $f_3(\alpha)$ are functions whose expressions are experimentally determined from normalized times $\{\tau_{x1}, \tau_{x2}, \tau_{x3}\}$ for $0.50 \leq \alpha \leq 1.10$. More specifically, function f_1 depends on the ratio index Δ, which is a function of normalized times τ_{x1}, τ_{x2}, and τ_{x3}; and functions f_2 and f_3 depend on α and times τ_{x1} and τ_{x3}, and τ_{x3}, respectively.

Figure 7 illustrates the scheme of the procedure to identify the parameters of a FFOPDT model, $\theta_P = \{K, T, L, \alpha\}$, using three arbitrary points $(x_1\text{-}x_2\text{-}x_3\%)$ on the process reaction curve. Note that this is a general procedure and accepts that points x_1, x_2 and x_3 to be arbitrary, as illustrated in Figure 6.

This procedure admits both symmetrical and asymmetrical position of the points on the reaction curve, as explained in [42].

This scheme is divided into two different parts, as depicted in Figure 7.

Part A is the general procedure to determine rational expressions for functions $f_1(\Delta)$, $f_2(\alpha)$ and $f_3(\alpha)$, which depend on the location of the set of points $(x_1\text{-}x_2\text{-}x_3\%)$. It consists of the following steps:

1. The values of the normalized times $\{\tau_{x1}, \tau_{x2}, \tau_{x3}\}$ for the considered three arbitrary points are obtained from the normalized process output (5), $\widetilde{y}_\alpha(\tau)$, for the different values of α, with $0.50 \leq \alpha \leq 1.10$.
2. Different data sets $\{\Delta, \alpha\}$, $\{\alpha, f_2(\alpha)\}$, and $\{\alpha, f_3(\alpha)\}$ are obtained using the values of the corresponding normalized times for the considered set of points $(x_1\text{-}x_2\text{-}x_3\%)$.
3. Using a curve-fitting procedure, rational functions are obtained for $f_1(\Delta)$, $f_2(\alpha)$ and $f_3(\alpha)$, respectively.
4. Considering the different rational functions $f_1(\Delta)$, $f_2(\alpha)$, and $f_3(\alpha)$, which have been determined in the previous step, the set of Equation (7) are completed.

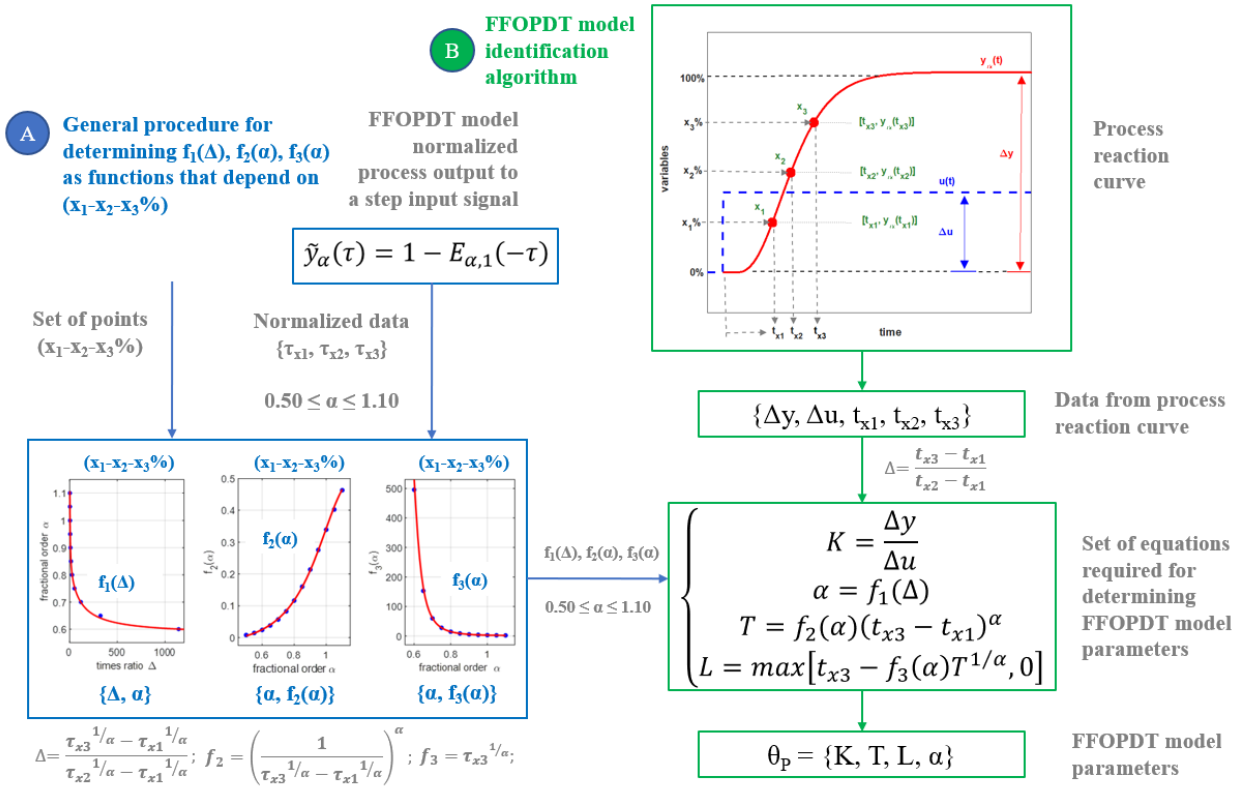


Figure 7. Scheme of the complete procedure for identifying a FFOPDT model considering three arbitrary points (x_1-x_2-x_3%) on the process reaction curve. Note that the scheme is divided into two parts: Part A represents the general procedure to obtain the set of equations for determining FFOPDT model parameters. These equations depend on the set of points (x_1-x_2-x_3%). Part B of the scheme illustrates the FFOPDT model identification algorithm for estimating model parameters θ_P = {K, T, L, α} with the information collected from the process reaction curve {Δy, Δu, t_{x1}, t_{x2}, t_{x3}}.

Part B consists of the algorithm for identifying the FFOPDT model from the process reaction curve. It consists of the following steps:

1. Data required for the identification procedure are obtained from the process reaction curve {Δy, Δu, t_{x1}, t_{x2}, t_{x3}}.
2. The value of the ratio index Δ is calculated from the set of times {t_{x1}, t_{x2}, and t_{x3}}.
3. The value of α and the numerical values of f_2 and f_3 are estimated.
4. After determining the numerical values of α, f_2, and f_3, the FFOPDT model parameters, θ_P = {K, T, L, α}, are calculated by using the set of equations (7) and the process data taken from the process reaction curve {Δy, Δu, t_{x1}, t_{x2}, t_{x3}}.

For a more detailed information about the identification method used in this paper, we refer the reader to [42], where one can find the complete development of the set of Equation (7), an analysis of the influence of the location of the set of points (x_1-x_2-x_3%) on the accuracy of the identified model, and some rules of thumb for the selection of symmetrical and asymmetrical sets of points. The effectiveness and applicability of this identification method for both symmetrical and asymmetrical sets of points on the reaction curve are also discussed in this paper.

4.2. Experimental Results

In this section, the results obtained by applying the identification method that has been summarized in the previous section on the thermal-based process under configurations #1 and #2 are presented.

Without loss of generality, an application has been programmed in LabVIEW using the microprocessor-based hardware for the results in this section. Since the considered identification algorithm is analytical, its implementation on any of the control technologies available in the hardware architecture presented in Section 3 is straightforward.

The experimental procedure followed in this section is as follows: An open-loop experiment has been applied to the input for both controlled process configurations and the obtained process reaction curve has been recorded. After that, the parameters of the FFOPDT models have been determined using the process output responses and the proposed identification method for sets of points (5-50-95%) and (20-60-95%), which are the symmetrical and asymmetrical sets, respectively, recommended in [42].

Finally, the accuracy of the identified model parameters and the effectiveness of the adopted model structure are evaluated in comparison to other integer-order models by using the mean absolute error (MAE):

$$E(\bar{\theta}) = \frac{1}{N_S}\sum_{k=1}^{N_S}\left|e(kT_S,\bar{\theta})\right| = \frac{1}{N_S}\sum_{k=1}^{N_S}\left|y(kT_S) - y_m(kT_S,\bar{\theta})\right| \quad (8)$$

where $e(kT_S,\bar{\theta})$ is the difference between the process reaction curve and the identified model step response, $y(kT_S)$ and $y_m(kT_S,\bar{\theta})$, respectively, T_S is the sampling period, N_S is the number of collected samples, $N_S \cdot T_S$ is the time length of the dynamic response, and $\bar{\theta}$ is the vector of process model parameters. The sampling period that has been used in all experiments is T_S = 100 ms.

Note that all the identification methods used in this section are analytical and based on the reaction curve, which means that they are simple to apply and the computational effort to implement them on a hardware device is low in all cases.

4.2.1. Controlled Process Configuration #1

In this part, the thermal-based laboratory setup considering controlled process configuration #1 is used. The block diagram of the controlled process for this configuration including variables, units, and the different components are illustrated in Figure 8.

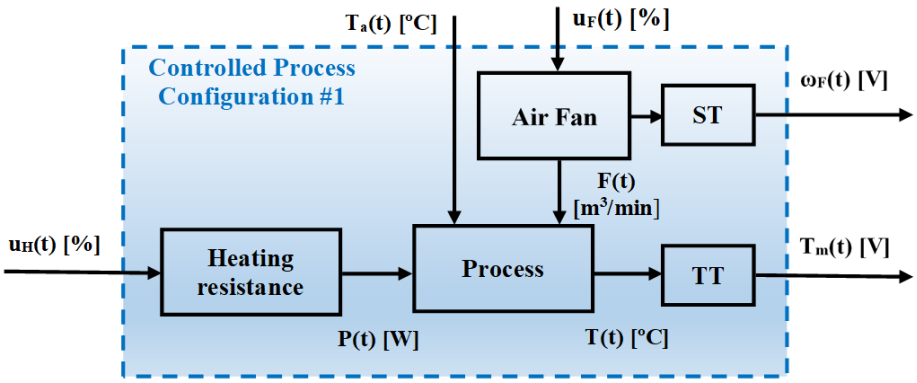
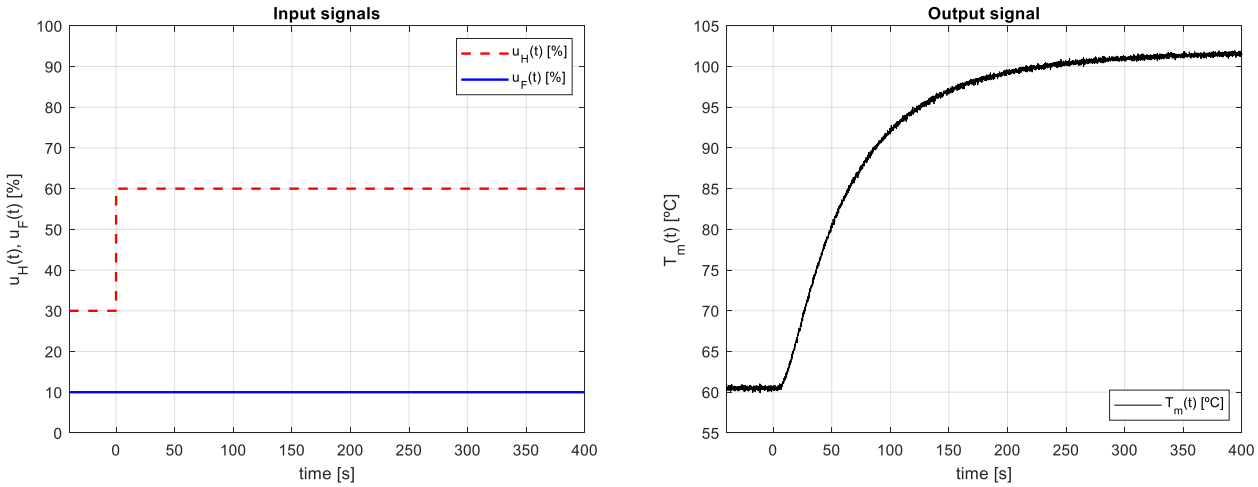


Figure 8. Block diagram for the controlled process configuration #1.

The open-loop step-test procedure is applied to the controlled process in this configuration as follows:

Initially, the control signal to the heating element and the command signal to fan are u_H = 30% and u_F = 10%, respectively. A step input with amplitude Δu_H = 30% is applied at t = 0 s, while u_F is maintained constant, as illustrated in Figure 9a. The measured variable T_m (measured temperature in the heat block), which is recorded in Figure 9b, increases from 60.5 to 102.5 °C (ΔT_m = 42 °C).



(**a**) (**b**)

Figure 9. Experimental data obtained from a step test for identification of the process model in the controlled process configuration #1: (**a**) Control signal $u_H(t)$ [%] and command signal to air fan $u_F(t)$ [%]; (**b**) Process reaction curve $T_m(t)$ [°C] at a certain operating point.

Table 6 shows the process information required for applying the fractional-order identification method for the proposed set of points (5-50-95%) and (20-60-95%) and collected from the process reaction curve.

Table 6. Process information for Configuration #1 for fractional-order model identification with sets of points (5-50-95%) and (20-60-95%).

FFOPDT		
Symmetrical (5-50-95%)		**Asymmetrical (20-60-95%)**
	$\Delta y = \Delta T_m = 42$ °C	
	$\Delta u = \Delta u_H = 30\%$	
$t_5 = 12.40$ s		$t_{20} = 24.70$ s
$t_{50} = 53.30$ s		$t_{60} = 67.30$ s
	$t_{95} = 251.60$ s	

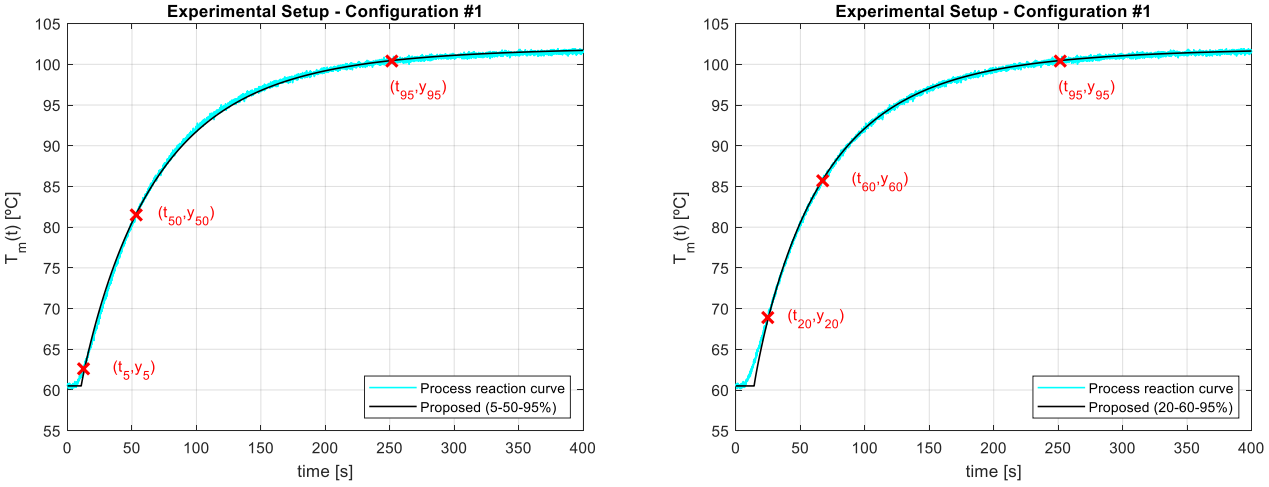
Table 7 shows the FOPDT and FFOPDT model parameters for the thermal-based process obtained by using identification methods proposed by Alfaro in [46] and Vitecková et al. in [47], respectively, and the ones obtained with the identification method proposed in [42] for (5-50-95%) and (20-60-95%), respectively.

Table 7. Model parameters obtained using the considered identification method for FFOPDT models using (5-50-95%) and (20-60-95%), and the ones using identification methods proposed by Alfaro and Vitecková for FOPDT models.

FFOPDT (5-50-95%)	**FFOPDT (20-60-95%)**	**FOPDT [46] (25–75%)**	**FOPDT [47] (33–70%)**
$K_{1,1} = 1.4$ °C/%	$K_{1,2} = 1.4$ °C/%	$K_{1,3} = 1.4$ °C/%	$K_{1,4} = 1.4$ °C/%
$T_{1,1} = 49.02$ s	$T_{1,2} = 42.56$ s	$T_{1,3} = 64.26$ s	$T_{1,4} = 63.37$ s
$L_{1,1} = 10.55$ s	$L_{1,2} = 14.06$ s	$L_{1,3} = 10.20$ s	$L_{1,4} = 10.25$ s
$\alpha_{1,1} = 0.9418$	$\alpha_{1,2} = 0.9270$	-	-

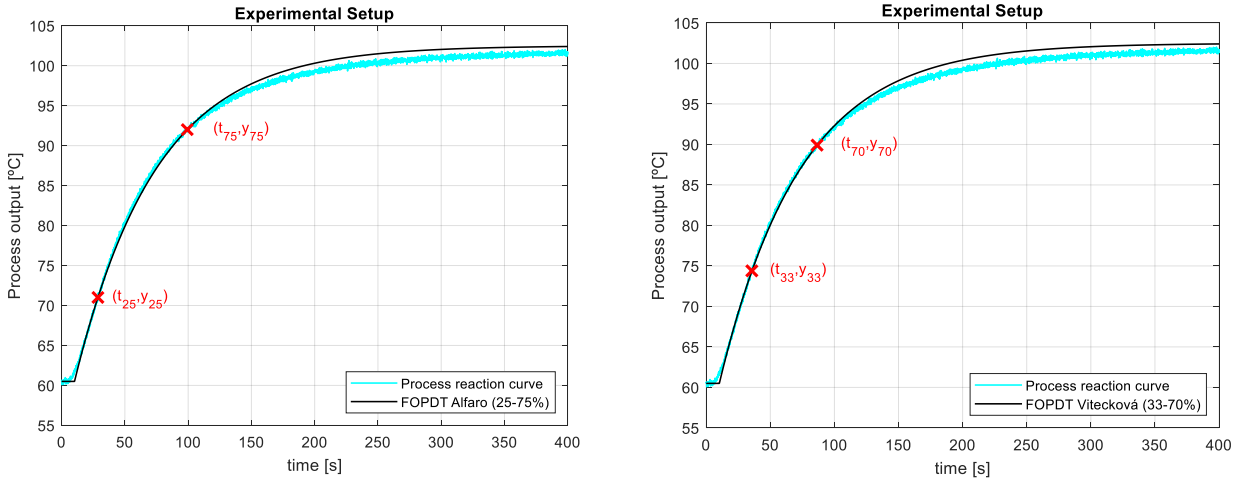
The step responses of the estimated models for FFOPDT models obtained with the symmetrical (5-50-95%) and asymmetrical (20-60-95%) case and the ones for FOPDT models obtained applying the considered classical methods, are then compared with the process

reaction curve and plotted in Figures 10 and 11, respectively. These figures also show the corresponding representative points.



(**a**) (**b**)

Figure 10. Process reaction curve in a certain operating point (Configuration #1) and FFOPDT model step response using the identification method proposed in [42] for the experimental setup: (**a**) Symmetrical set of points (5–50–95%); (**b**) Asymmetrical set of points (20–60–95%).



(**a**) (**b**)

Figure 11. Process reaction curve in a certain operating point (Configuration #1) and FOPDT model step response using two-point identification methods for the experimental setup: (**a**) Method proposed by Alfaro in [46]; (**b**) Method proposed by Vitecková in [47].

Figure 10 illustrates that both FFOPDT models obtained using the method proposed by Gude and García Bringas fit the process reaction curve for the thermal process better than the results obtained for FOPDT models, which are illustrated in Figure 11.

Table 8 shows the values of the time-domain performance index $E(\bar{\theta}_{1,i})$ for the estimated models obtained with this configuration. Considering this table, one can observe that the result obtained with the asymmetrical set of points is better than the one obtained with the symmetrical case, more precisely, the value of E for the asymmetrical case is 1.6 times lower than the one obtained for the symmetrical case.

Table 8. Model performance indexes obtained for the considered identification methods for FFOPDT and FOPDT models.

i	Identification Method	Set of Points	$\bar{E}(\theta_{1,i})$
1	FFOPDT Gude and G. Bringas	(5–50–95%)	6.60×10^{-3}
2	FFOPDT Gude and G. Bringas	(20–60–95%)	4.10×10^{-3}
3	FOPDT Alfaro	(25–75%)	2.50×10^{-2}
4	FOPDT Vitecková	(33–70%)	2.59×10^{-2}
	Number of samples:		$N_S = 4001$

On the other hand, the results obtained using the method for FFOPDT models proposed by Gude and García Bringas for the symmetrical and asymmetrical case provide better results in terms of E than the ones obtained for FOPDT models. More specifically, the FFOPDT model estimated for (5–50–95%) and for (20–60–95%) give E values 3.8 and 3.9, and 6.1 and 6.3 times lower than those obtained for FOPDT models proposed by Alfaro and Vitecková, respectively.

4.2.2. Controlled Process Configuration #2

The controlled process configuration #2, which is the one selected in this part, is illustrated as a block diagram in Figure 12.

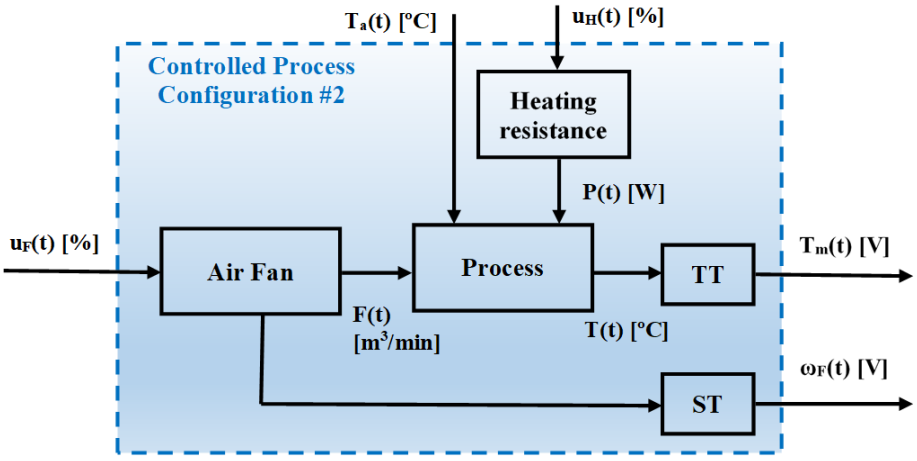
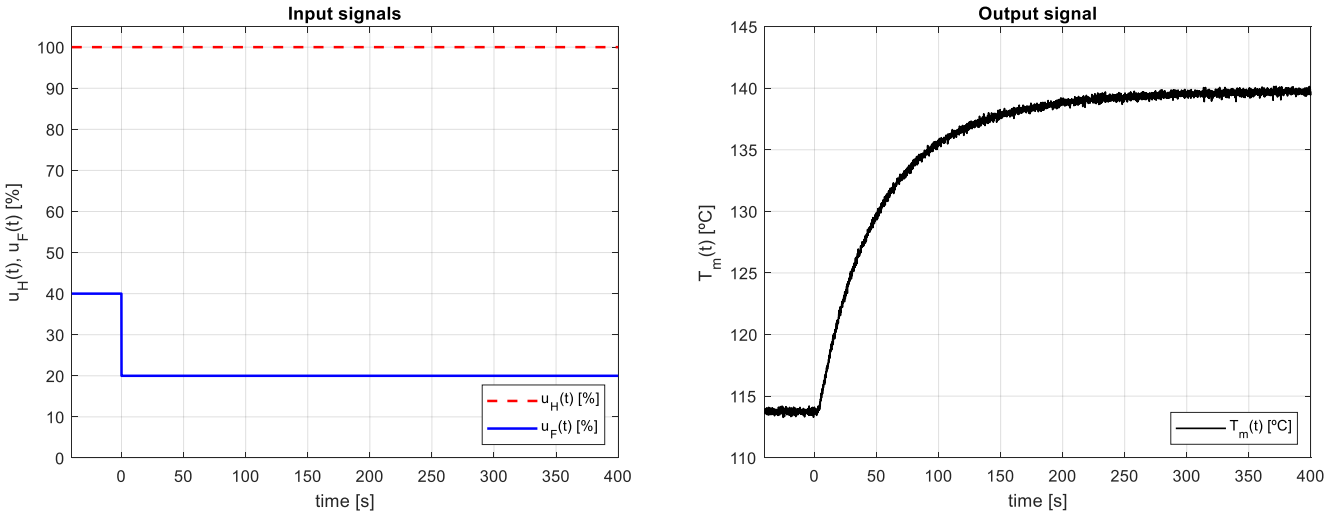


Figure 12. Block diagram for the controlled process configuration #2.

Note that the controlled process is reversed-action in this configuration and its dynamic characteristics are different from the previous case. The procedure followed in this case is similar to the one followed in the previous case. Initially, the control signal to the air fan is at $u_F = 40\%$ and the command signal to the heating element is at $u_H = 100\%$. A step input with amplitude $\Delta u_F = -20\%$ is applied at t = 0 s, while u_H is maintained constant, as depicted in Figure 13a. The measured variable T_m (measured temperature in the heat block), which is recorded in Figure 13b, changes from 113.75 to 139.75 °C ($\Delta T_m = 26$ °C).

Table 9 contains the process information required for applying the fractional-order identification method for the proposed set of points (5-50-95%) and (20-60-95%) and collected from the process reaction curve.

Table 10 shows the FOPDT and FFOPDT model parameters for the thermal-based process in configuration #2 by using the identification method proposed by Gude and García Bringas in [42] for (5-50-95%) and (20-60-95%), respectively, and the ones obtained with methods proposed by Alfaro in [46] and Vitecková et al. in [47], respectively.



(**a**) (**b**)

Figure 13. Experimental data obtained from a step test for identification of the process model in the controlled process configuration #2: (**a**) Control signal $u_F(t)$ [%] and command signal to heating resistance $u_H(t)$ [%]; (**b**) Process reaction curve $T_m(t)$ [°C] at a certain operating point.

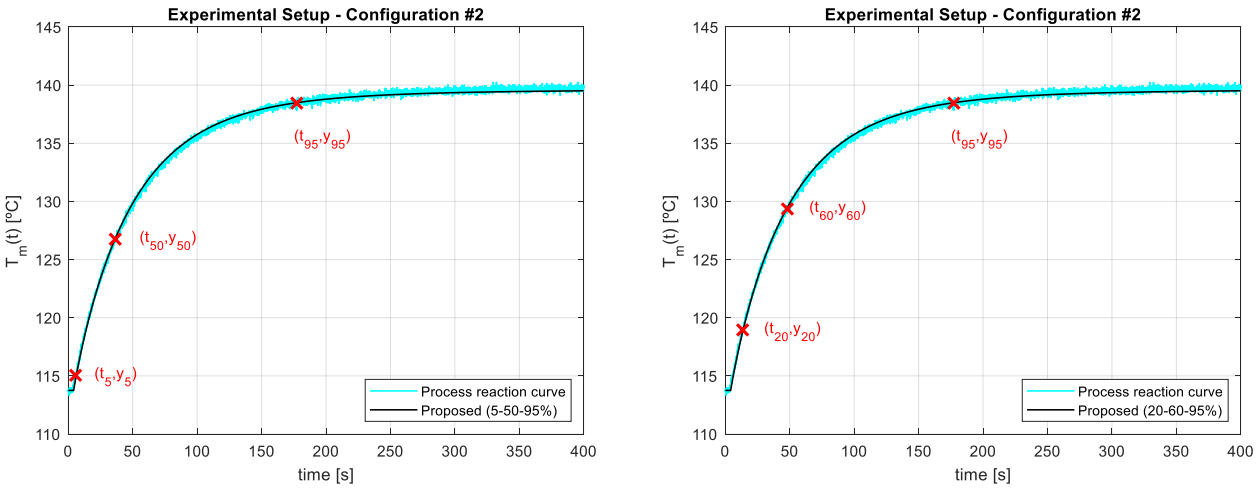
Table 9. Process information for Configuration #2 for fractional-order model identification with sets of points (5–50-95%) and (20-60-95%).

	FFOPDT	
Symmetrical (5-50-95%)		**Asymmetrical (20-60-95%)**
	$\Delta y = \Delta T_m = 26$ °C	
	$\Delta u = \Delta u_F = -20\%$	
$t_5 = 5.69$ s		$t_{20} = 13.51$ s
$t_{50} = 36.49$ s		$t_{60} = 48.20$ s
	$t_{95} = 177.34$ s	

Table 10. Model parameters obtained using the considered identification method for FFOPDT models using (5-50-95%) and (20-60-95%), and the ones using identification methods proposed by Alfaro and Vitecková for FOPDT models.

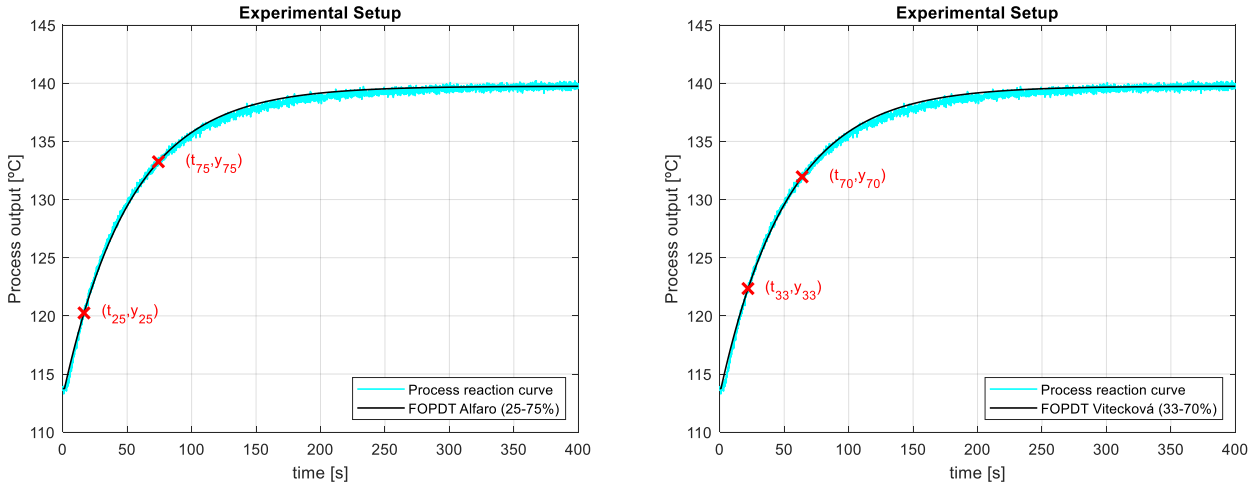
FFOPDT (5-50-95%)	**FFOPDT (20-60-95%)**	**FOPDT [46] (25-75%)**	**FOPDT [47] (33-70%)**
$K_{2,1} = -1.3$ °C/%	$K_{2,2} = -1.3$ °C/%	$K_{2,3} = -1.3$ °C/%	$K_{2,4} = -1.3$ °C/%
$T_{2,1} = 39.20$ s	$T_{2,2} = 40.20$ s	$T_{2,3} = 52.54$ s	$T_{2,4} = 52.34$ s
$L_{2,1} = 4.10$ s	$L_{2,2} = 3.76$ s	$L_{2,3} = 1.37$ s	$L_{2,4} = 0.93$ s
$\alpha_{2,1} = 0.9523$	$\alpha_{2,2} = 0.9551$	-	-

In this configuration, the step responses of the estimated FFOPDT models obtained using both the symmetrical (5-50-95%) and asymmetrical (20-60-95%) case and the ones for FOPDT models are then compared with the process reaction curve and plotted in Figures 14 and 15, respectively. These figures also show the corresponding representative points.



(**a**) (**b**)

Figure 14. Process reaction curve in a certain operating point (Configuration #2) and FFOPDT model step response using the identification method proposed in [42] for the experimental setup: (**a**) Symmetrical set of points (5-50-95%); (**b**) Asymmetrical set of points (20-60-95%).



(**a**) (**b**)

Figure 15. Process reaction curve in a certain operating point (Configuration #2) and FOPDT model step response using two-point identification methods for the experimental setup: (**a**) Method proposed by Alfaro in [46]; (**b**) Method proposed by Vitecková in [47].

Figure 14 shows that both FFOPDT models obtained with the method proposed by Gude and García Bringas give a good fit to the process reaction curve compared with the results for FOPDT models, as illustrated in Figure 15.

Table 11 shows the values of the time-domain performance index $E(\bar{\theta}_{2,i})$ for the estimated models obtained with this configuration. Considering this table, one can observe that the result obtained with the asymmetrical set of points is slightly better than the one obtained with the symmetrical set of points. The results obtained using the method for FFOPDT models proposed by Gude and García Bringas for the symmetrical and asymmetrical case provide better results in terms of E than the ones obtained for FOPDT models. More specifically, the FFOPDT model estimated for (5-50-95%) and for (20-60-95%) give E values 1.5 and 1.6, and 1.8 and 1.9 times lower than those obtained for FOPDT models proposed by Alfaro and Vitecková, respectively.

Table 11. Model performance indexes obtained for the considered identification method for FFOPDT and FOPDT models.

i	Identification Method	Set of Points	$\bar{E}(\theta_{2,i})$
1	FFOPDT Gude and G. Bringas	(5-50-95%)	7.20×10^{-3}
2	FFOPDT Gude and G. Bringas	(20-60-95%)	6.10×10^{-3}
3	FOPDT Alfaro	(25-75%)	1.10×10^{-2}
4	FOPDT Vitecková	(33-70%)	1.15×10^{-2}
	Number of samples:		$N_S = 4001$

4.3. LabVIEW-Based Application

In this work, the programming language LabVIEW has been selected as a tool to operate the experimental setup, estimate integer- or fractional-order model parameters, and implement integer- and fractional-order control algorithms on different hardware technologies, as discussed previously.

Nowadays, the graphical programming language LabVIEW can be considered as an industry standard. LabVIEW provides transparent access to hardware devices and, with the most advanced programming techniques, facilitates the application of various modeling approaches, as well as the practical implementation of both advanced and conventional control algorithms.

Figure 16 shows the main menu of the LabVIEW-based application that has been implemented to configure and operate with Deusto HES prototype. Two different parts can be observed in this figure, one corresponding to process modelling, and another one corresponding to control with different hardware technologies.

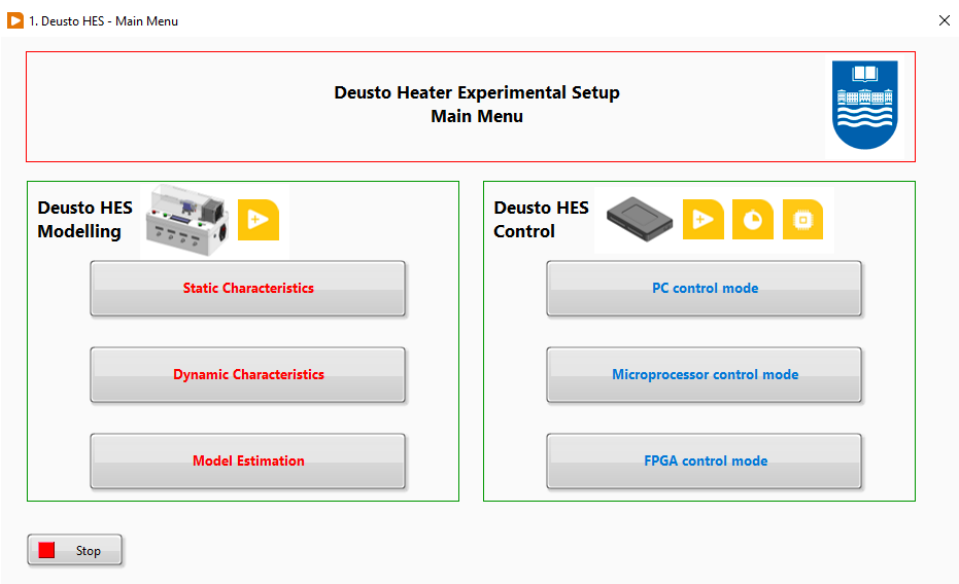


Figure 16. Main menu of the LabVIEW-based application to operate with Deusto HES prototype.

Deusto HES modelling

Three applications have been developed to characterize the controlled process in each one of the configurations:

1. Static characteristics. This application allows determining the static characteristics of the controlled process in the different available configurations, as shown in Figure 17.
2. Dynamic characteristics. It allows obtaining the evolution of the controlled variable when there is a change in the manipulated variable or disturbance, respectively. Figure 18 illustrates variation of $T_m(t)$ when the value of $u_F(t)$ is modified as a staircase

while u_H(t) is maintained constant. A nonlinear behaviour of the temperature in the heat block can be emphasized in this figure.

3. Model estimation. The values of the parameters for a simple-structure fractional-order model can be estimated numerically with this application.

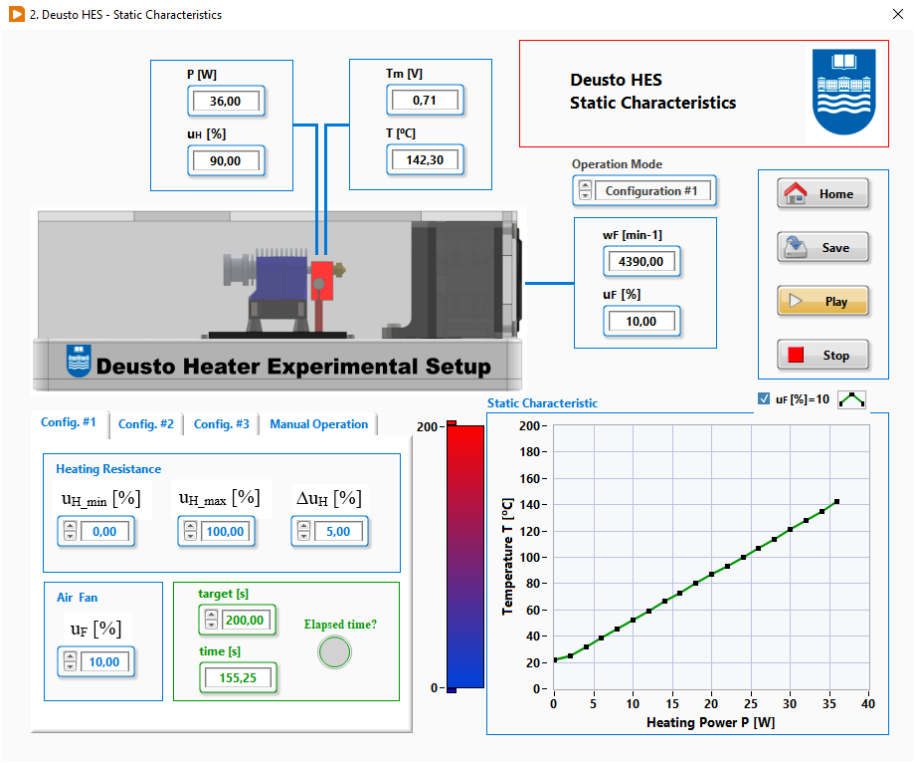


Figure 17. LabVIEW-based application for obtaining static characteristics for different configurations of the controlled process and different operating points.

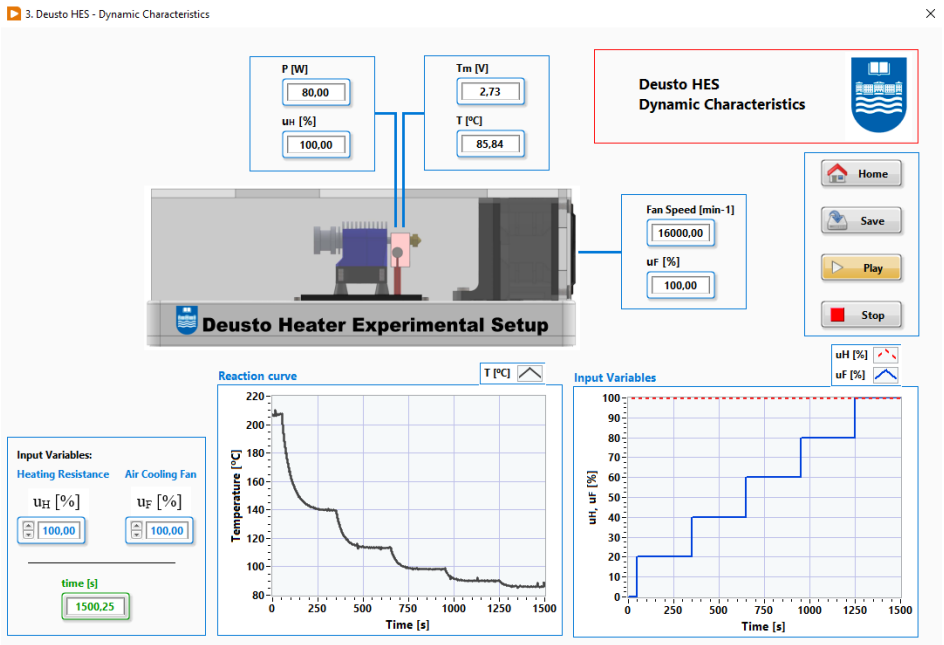


Figure 18. LabVIEW-based application for obtaining dynamic characteristics for different configurations of the controlled process and different operating points.

Figure 19 illustrates the appearance of the LabVIEW-based application for estimation of fractional models by using the process reaction curve-based identification method proposed in [42] for different symmetrical and asymmetrical sets of points.

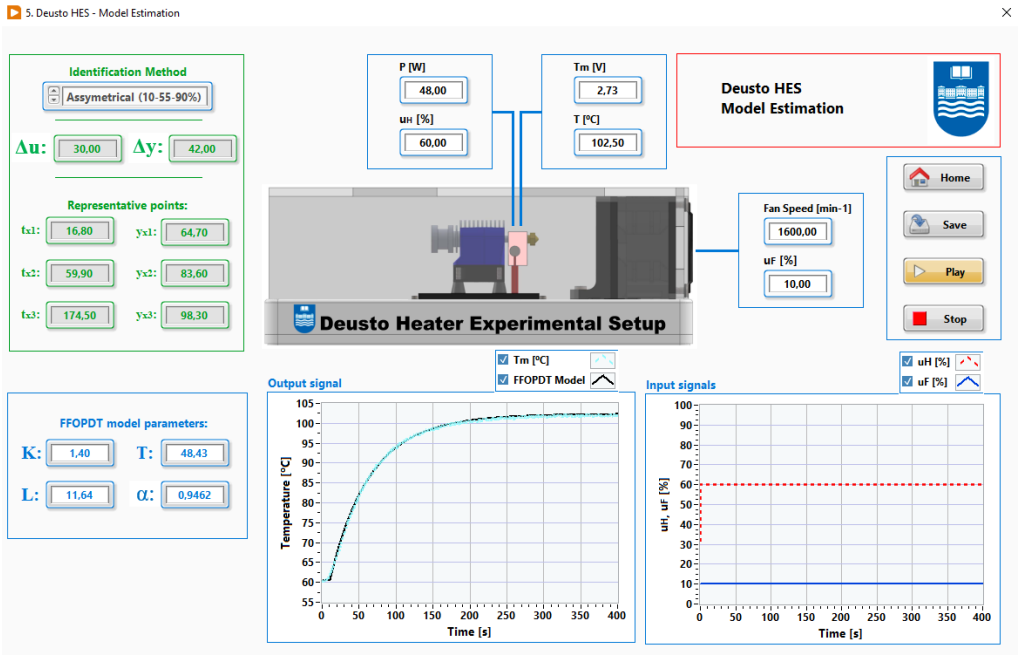


Figure 19. Application programmed with LabVIEW for the implementation of fractional-order identification algorithms.

This application presents the following features:

- Selection of the fractional-order identification method and set of points:
  - Asymmetrical case (x_1-x_2-x_3%).
  - Symmetrical case (x-50-(100−x)%).
- Determination of process data {Δu, Δy, t_{x1}, t_{x2}, t_{x3}} taken from the reaction curve.
- Identification of the fractional-order model parameters: θ_P = {K, T, L, α}. Note that FOPDT model parameters can be determined by using the same identification procedure, since FOPDT model can be considered as a particular case with α = 1.
- Graphs for registering the process reaction curve $T_m(t)$ [°C], the step response of the identified model, representative points of the process reaction curve {(t_{x1}, $T_m(t_{x1})$), (t_{x2}, $T_m(t_{x2})$), (t_{x3}, $T_m(t_{x3})$)}, the command signal to air fan $u_F(t)$ [%], and the command signal to heating resistance $u_H(t)$ [%].
- Export the experimental data in text- or excel-format.

4.4. Discussion

As mentioned above, one of the objectives of this paper is to present a novel control hardware architecture for the practical training in the implementation of integer- and fractional-order identification and control algorithms in a real-time target.

In this section, the hardware architecture proposed in the previous section is used to validate its effectiveness for the implementation of fractional-order simple-structure model identification algorithms.

In this regard, the temperature-based process, which involves thermal conduction, exhibits fractional behaviour in both configurations #1 and #2. This makes this laboratory setup suitable for training in aspects related to modeling and implementation of integer- and fractional-order model identification methods. Since fractional behaviour is ubiquitous in the industrial context, the existence of methods for fractional-order model identification and the ability to be implemented on real-time targets is of major interest.

In the specific scope of this section, the implemented identification procedure has been compared with various well-recognized identification procedures for integer-order systems that use process data taken from the process reaction curve. It has been obtained that the estimated fractional-order models significantly outperform integer-order models in terms of accuracy. This confirms the well-known benefit obtained when using fractional calculus and justified in the technical literature in terms of more accurate modeling; see, e.g., [4,48,49]. In a recent paper [42], there is an interesting discussion on whether the additional effort of considering fractional calculus is worth it to obtain a more accurate model and, eventually, a better control performance.

Without loss of generality, microprocessor-based hardware has been used for the implementation of fractional-order identification algorithms, which has made possible to verify their applicability in an industrial environment. Although in this section only analytical identification methods have been implemented, whose implementation is very simple and straightforward in any of the real-time targets available in the proposed architecture, it is also possible to implement more complicated algorithms.

In this regard, it is important to note that a typical process plant involves hundreds or thousands of control loops and for an identification method of this type to have a potential impact in industry, a key requirement is simplicity. Additionally, having laboratory setups such as the one proposed in this paper supports training in the implementation of more complex identification algorithms, usually based on optimization.

In the authors' opinion, the industry needs such identification procedures, especially when the emphasis is on simplicity, and that will help to bridge the existing gap between the practical implementation in industry and theoretical research on fractional modelling.

5. Control Issues and Results

In recent decades, it is widely recognized in both academia and industry that the new computational techniques and the development of fractional calculus have made possible significant advances in fractional-order modeling and control.

However, the existing problems with the implementation of fractional-order control algorithms in real-time targets has caused to become more challenging to convey the practical benefits of fractional calculus in the industrial field [15]. Therefore, the use of fractional control algorithms in the industry is currently relatively low, despite the potential advantages offered by fractional-order PID controllers compared to integer-order ones.

In this section, various examples using different technologies have been presented in order to show the applicability and effectiveness of the proposed control hardware architecture in the implementation of fractional-order control algorithms. Some hardware implementation issues are also discussed.

Consequently, for the purposes of this paper related to control considerations, it is enough to design and implement fractional-order PI controllers on two different real-time targets, i.e., on microprocessor- and FPGA-based hardware, respectively, thus verifying the applicability of the proposed hardware architecture and dealing with the specific implementation issues of each hardware platform.

One of the contributions of this paper lies in providing an efficient and practical hardware architecture for implementing fractional-order controllers in different control technologies. This work's main objective is to help in filling the gap between real-time hardware solutions and software-based simulations of fractional controllers.

This section is divided into the following parts: First, issues about hardware implementation are discussed. More specifically, aspects about microprocessor- and FPGA-based implementation are addressed. Then, the results of two examples are discussed for illustrative purposes, both considering the implementation of integer- and fractional-order PI controllers on microprocessor- and on FPGA-based mode, respectively. After that, the application that has been implemented in LabVIEW in order to operate the experimental setup and for the implementation of fractional-order control algorithms in different control

technologies is described. Finally, some final remarks and discussion about control issues in this context are provided.

5.1. Hardware Implementation Issues

As was mentioned previously, fractional-order systems and controllers have been widely studied during the last two decades, and considerable advancements have been achieved in the analysis and theory of these systems. Moreover, a great deal of effort has been made in the hardware realization of such systems. As a general rule, the implementation of fractional-order control algorithms or systems typically requires the approximation of such systems as high-order rational systems. Consequently, in general it is difficult to translate them into hardware.

Because of the particular practical issues in the approximation and implementation of fractional-order controllers, it is necessary to consider certain aspects, such as hardware cost and speed, and system performance, when choosing the hardware device [26].

Broadly speaking, implementation of fractional-order integrators and differentiators can be realized by considering analog or digital approximations.

In the analog realization, the considered fractional operator is then approximated as a rational higher-order system that keeps a constant phase within a selected frequency band [50–52]. The most commonly used analog approximations are based on Oustaloup's Recursive Approximation method and the Modified Oustaloup Filter [5,53], Continued Fraction Expansion (CFE) methods [54], or Charef's method [55]. These CFE methods can be subdivided into low- and high-frequency methods, as well as Matsuda's and Carlson's methods [56].

Otherwise, digital realizations of fractional operators, which are more suitable for a hardware implementation, can be performed by considering direct or indirect discretization methods.

Considering an indirect approach, the procedure consists of first doing the frequency-domain fitting in the continuous-time domain and then discretizing the fitted continuous-time transfer function.

Considering a direct approach, these can be based on continued fraction expansion (CFE), power series expansion (PSE), and MacLaurin series expansion, in combination with an appropriate generating function that converts the continuous-time element into the equivalent discrete-time element.

The general formula for the generating function is as follows

$$w\left(z^{-1}\right) = \left(\frac{1}{\beta T_S}\frac{1-z^{-1}}{\gamma+(1-\gamma)z^{-1}}\right) \tag{9}$$

where $w(z^{-1})$ is the discrete equivalent element, Ts is the sampling period, and β and γ are the gain and phase tuning parameters of the generating function.

Several different types of such generating functions have been proposed in technical literature, such as Al-Alaoui, Simpson, recursive Tustin, mixed Tustin-Simpson, mixed Euler-Tustin-Simpson, impulse response-based, and other higher-order generating functions; see the following references [54,56–60].

The expansion formula together with the generating function defines the form of the approximation and the coefficients [5]. The expansion of the generating function can generally be performed through the PSE or CFE methods, which lead to approximations in the form of polynomial or rational functions, respectively. In case of using the PSE method, an approximation with only zeros is obtained for the fractional-order element:

$$s^{\alpha} \cong \mathrm{PSE}\left[w\left(z^{-1}\right)^{\alpha}\right] = P_p\left(z^{-1}\right) \tag{10}$$

where α is the non-integer order of the fractional operator and P_p is the corresponding polynomial with degree p of variable z^{-1}.

In case of using the CFE method, an approximation with zeros and poles are obtained for the fractional-order element:

$$s^{\alpha} \cong \mathrm{CFE}\left[w\left(z^{-1}\right)^{\alpha}\right] = \frac{P_p\left(z^{-1}\right)}{Q_q\left(z^{-1}\right)} \tag{11}$$

where α is the non-integer order of the fractional operator, P_p and Q_q are the corresponding polynomials with degrees p and q, respectively, of variable z^{-1}.

The CFE method is usually preferred over the PSE method because rational functions have the ability to model functions with zeros and poles, which converge faster and have a wider domain of convergence in the complex plane.

Moreover, direct discretization is in general the preferred approach for digital realization over its indirect counterpart [61]. Note that the final choice for the most appropriate approximation should be connected to various aspects, among the most relevant are the accuracy required in the frequency or the time domain, the large the integer-order transfer functions may be, the maximum order of the transfer function, etc.

In the following, some practical aspects are discussed in particular for the cases where the implementation is performed on microprocessor- and FPGA-based hardware, respectively.

5.1.1. Microprocessor-Based Implementation

A survey of several fractional-order controller implementations based on microprocessor-based devices is presented in [26]. In this monograph, the fundamentals of the discrete approximations of a fractional-order operator are described, as well as the control algorithms for the implementation of the controllers. Three examples using discrete fractional-order controllers implemented on a PLC, on a PC with data acquisition card, and on a PIC-type microcontroller, respectively, are also presented. For each of the cases considered, the results obtained are illustrated.

It is important to mention that controller realization generally is not equivalent to simulation or numerical evaluation of the fractional-order differential and integral operators.

In the controller realization case, it is worth taking into account some significant considerations:

1. Depending on the type of microprocessor, each system has its specific minimum value for the sampling period.
2. It is required to perform all the calculations required by the control law between two samples.

Because of this last point, obtaining good approximations with a minimum set of parameters is very important because as the number of parameters in the approximation increases, so does the speed and the amount of the required memory.

As a conclusion, the key point in the digital implementation of a fractional-order controller on a microprocessor-based device is the discretization of the fractional-order operators [26].

5.1.2. FPGA-Based Implementation

One of the first monographs devoting to implementation of fractional-order systems in hardware devices is [26]. There, the basic operator s^{α}, where α is considered to be a real number, is approximated by the binomial expansion of the backward difference and then a hardware implementation of the differintegral operator is proposed using an FPGA. In this context, this building block can be considered as a basic element used to implement fractional-order control systems.

There are some recent references in the technical literature where the implementation of fractional-order operators in the LabVIEW environment is discussed. A systematic procedure for hardware implementation of the basic operators, i.e., fractional-order integrator and derivative, using the Grünwald–Letnikov definition, on FPGA in LabVIEW environment is presented in [62]. In [63], the authors proposed the implementation of the

fractional-order PI controller on an FPGA for a DC motor speed control. The advantages of using FPGAs for implementing fractional-order PI controllers are addressed and the FPGA implementation issues of robust fractional-order PI controllers are discussed. However, several implementation-related issues were not resolved. Such implementation issues that may be encountered are mainly related to the data representation. On the one hand, calculations using floating-point data are generally very accurate, but are hard to be applied and implemented in hardware applications. In contrast, operations using fixed-point data offer the benefit of increased computation speed and are quite easy to be implemented. The main inconvenience of fixed-point data resides in the loss of accuracy. The comparative results obtained in the implementation of fractional-order controllers on two different FPGA real-time targets using various data representations have been presented in [29]. The experimental results obtained in the application of speed control of a DC motor show that integer, double, and fixed-point data representation can be applied efficiently for control purposes.

It is widely acknowledged that FPGA implementations are up to one hundred times faster than microprocessor implementations; this additional speed may be further exploited to provide a higher performance in terms of digital approximations of fractional control algorithms. Overall, the implementation technique exploits versatility and the parallel structure of FPGA-based devices in order to obtain a low-cost implementation providing a higher performance.

5.2. Experimental Results

Some recently published studies have pointed to fractional-order controllers, more specifically of PID type, as an emerging trend in the process industry (see [15,64]). The main explanation for their popularity is the inherent robustness they provide by having a higher degree of freedom for operating and tuning controllers.

For the purposes of this paper in relation to control considerations, it is sufficient to design and implement integer- and fractional-order PI controllers on the technologies offered by the proposed hardware architecture and applied for the temperature control of the Deusto HES prototype.

The transfer function of the fractional-order PI controller is:

$$G_{FO-PI}(s) = \frac{U(s)}{E(s)} = K_P\left(1 + \frac{1}{T_i s^{\lambda}}\right) = K_P + \frac{K_i}{s^{\lambda}} \quad (12)$$

where E and U are the Laplace transforms of the error and control signals, respectively, K_P is the proportional gain, K_i is the integral gain, T_i is the integral time, and λ is the non-integer order of the integrator. For the particular case where $\lambda = 1$, the fractional-order PI controller becomes the traditional integer-order PI controller.

This section is divided into two parts: In the first part, the considered integer- and fractional-order PI controllers are implemented in the microprocessor mode and applied to the controlled process in its configuration #1. Tuning rules based on an open-loop test are used to tune both controllers. While in the second part, the controllers are implemented in the FPGA mode and applied to the controlled process in its configuration #2. In this case, tuning methods based on the frequency response have been used.

Note that the objective of this section is not to perform an evaluation of any particular control algorithm or tuning method, but rather to test the applicability and effectiveness of the proposed hardware architecture for the implementation of integer- and fractional-order controllers.

5.2.1. Microprocessor-Based Implementation

In this subsection, results obtained with integer- and fractional-order PI controllers implemented in the microprocessor-based real-time target proposed in the hardware architecture are presented.

As the configuration #1 of the controlled process is used in this example, the power supplied to the heat block by the heating resistor P(t) is the manipulated variable. This variable can be modified by varying the PWM signal in u_H(t), which is the control signal in this configuration. The PWM signal to the air fan u_F(t) can be used as a disturbance. See Table 2 for a complete list of the main process variables for this configuration.

Figure 20 shows the scheme of the different software and hardware components used for the microprocessor-based closed-loop implementation of temperature control in this example.

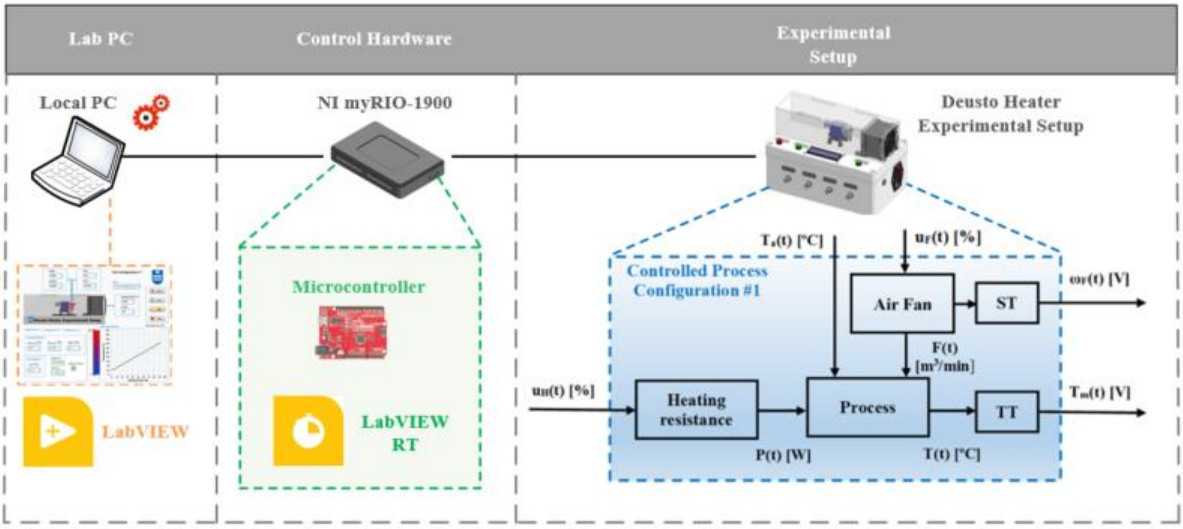


Figure 20. Scheme of microprocessor-based closed-loop implementation of temperature control with Controlled Process Configuration #1.

As discussed in Section 4, the controlled process in configuration #1 exhibits fractional behaviour. Therefore, considering the process reaction curve in Figure 9, which has been obtained for a certain operating point by using a typical open-loop step-test experiment, the controlled process can be modeled by using an FFOPDT or an FOPDT model. Note that FFOPDT and FOPDT model identification methods have been implemented on the microprocessor-based hardware in Section 4.

Although there are a large number of tuning methods for such PI and fractional-order (FO) PI controllers, in this example the AMIGO and F-MIGO methods proposed by Hägglund and Åström in [65] and by Bhaskaran et al. in [66] will be used for PI and FO-PI controllers, respectively.

According to both tuning methods, the FOPDT model parameters and the normalized dead time τ are used to design both an integer- and a fractional-order PI controller. AMIGO and F-MIGO methods provide the tuning rules for the PI and FO-PI controllers, respectively, as long as the controlled process step response has an S-shaped form that could be approximated by an FOPDT model. The transfer function of an FOPDT model is as follows:

$$\mathrm{P(s)} = \frac{\mathrm{Ke}^{-\mathrm{Ls}}}{1 + \mathrm{Ts}} \tag{13}$$

where K is the process gain, T > 0 is the time constant, and L $\geq$ 0 is the apparent dead-time.

The identified parameters of the FOPDT model at this operating point, θ_P = {K, T, L}, which are necessary to apply the considered tuning methods, are: K = 1.40 °C/%, T = 64.26 s., and L = 10.20 s.

On the other hand, the normalized dead-time is defined as:

$$\tau = \frac{\mathrm{L}}{\mathrm{T_{ar}}} = \frac{\mathrm{L}}{\mathrm{L + T}} \tag{14}$$

has the property $0 \leq \tau \leq 1$ and is a parameter that can be used to characterize the difficulty of controlling a process. In the technical literature, some authors use the controllability index L/T to characterize the process, similar to the τ parameter; See, e.g., [43]. A process is lag-dominated if τ is small (T >> L), delay-dominated if τ is large (L >> T), and balanced if τ is around 0.5.

For the identified model, the normalized dead time is $\tau = 0.1370$, and the controllability index is L/T = 0.1587.

Both AMIGO and F-MIGO are robust tuning methods that determine the optimum parameters for PI and FO-PI controllers, respectively, in such a way that the load disturbance rejection is optimized, with a constraint on the maximum sensitivity M_S.

Fractional MIGO (F-MIGO) is a popular tuning method, which is suitable for fractional-order PI controllers, and was developed as an extension of the MIGO (M_S constrained integral gain optimization) method proposed in [67].

According to [66], the fractional order λ of the controller is practically independent of L, but is dependent on the normalized dead-time τ. For some practical situations, when $0.4 \leq \tau < 0.6$, it is determined that an integer-order PI controller is more convenient for controlling the process. Proportional gain and integral time are defined as a function on the normalized dead-time τ.

The tuning rules for $M_S = 1.4$ can be summarized as follows:

$$\begin{aligned} K_P K &= \frac{0.2978}{\tau + 0.00037} \\ \frac{T_i}{T} &= \frac{0.8578}{\tau^2 - 3.402\tau + 2.405} \\ \lambda &= \begin{cases} 1.1, & \tau \geq 0.6 \\ 1, & 0.4 \leq \tau < 0.6 \\ 0.9, & 0.1 \leq \tau < 0.4 \\ 0.7, & \tau < 0.1 \end{cases} \end{aligned} \tag{15}$$

Considering the estimated parameters of the FOPDT model and the value of the normalized dead-time for the controlled process, the controller parameters obtained with these tuning rules are $\theta_C = \{K_P, K_i, \lambda\}$, with $K_P = 1.5486$, $K_i = 0.055$, and $\lambda = 0.9$. In this case, the Oustaloup Recursive Approximation method [5] with the following parameters: N = 6, $\omega_b = 10^{-2}$, $\omega_h = 10^2$ has been used to approximate the fractional-order operator. The continuous-time controller is then discretized.

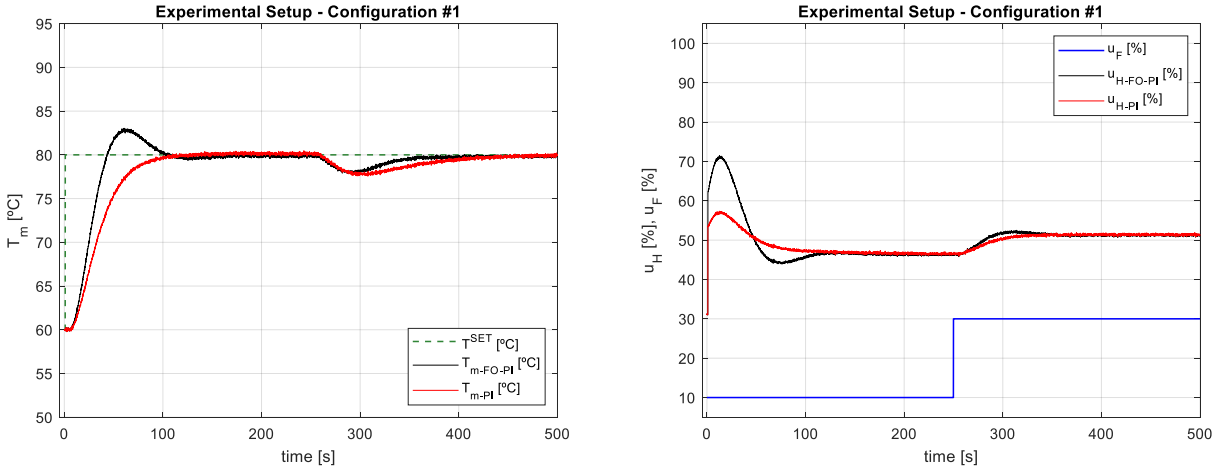
The considered method for PI controllers is based on the MIGO method [67]. The tuning rules proposed in [65] are obtained by finding simple relations between process parameters and controller parameters. The resulting design method is called AMIGO (Approximate MIGO based on step response data).

Most processes where the dynamics are the limiting factor for control design have a relation between dead-time L and time constant T that satisfy $0.15 < L/T < 1$. For these processes, the proposed tuning rules for $M_S = 1.4$ become particularly simple:

$$\begin{aligned} K_P K &= \frac{0.25T}{L} \\ \frac{T_i}{T} &= 0.8 \end{aligned} \tag{16}$$

Considering the estimated parameters of the FOPDT model for the controlled process, the controller parameters obtained with these tuning rules are $\theta_C = \{K_P, K_i\}$, with $K_P = 1.1250$ and $K_i = 0.0219$.

The closed-loop experimental results, obtained considering the operating point at which both the integer- and the fractional-order PI controllers have been designed, have been presented in Figure 21a for servo and regulatory control. In this example, the closed-loop test for servo control consists of a setpoint change T_{SP} from 60 to 80 °C at time $t = 0$ s, while the closed-loop test for regulatory control consists of a change in the signal u_F from 10 to 30% at time $t = 250$ s. The corresponding control signals for both controllers, $u_{H\text{-}PI}$ and $u_{H\text{-}FO\text{-}PI}$, and disturbance signal u_F are given in Figure 21b.



(**a**) (**b**)

Figure 21. Closed-loop microprocessor-based experimental results for servo and regulatory control, considering a PI and a fractional-order PI controller: (**a**) Temperature in the heat block; (**b**) control signal u_H and disturbance u_F.

Figure 21a shows that both microprocessor-based implemented controllers give good performance, especially for regulatory control. Note that both tuning methods are robust approaches that seek to optimize load disturbance rejection.

5.2.2. FPGA-Based Control

The design and FPGA-based implementation of an integer- and a fractional-order PI controller for the temperature control in Deusto HES is exemplified in this section. Therefore, the FPGA-based real-time target in the proposed hardware architecture is used.

As the configuration #2 of the controlled process is used in this example, the airflow generated by the air fan F(t) is the manipulated variable. This variable can be modified by varying the PWM signal in u_F(t), which is the control signal in this configuration. The PWM signal to the heating resistance u_H(t) can be used as a disturbance. See Table 2 for a complete reference of the main process variables for this configuration.

Figure 22 illustrates the scheme of the different hardware and software components used for the FPGA-based closed-loop implementation of temperature control in this example.

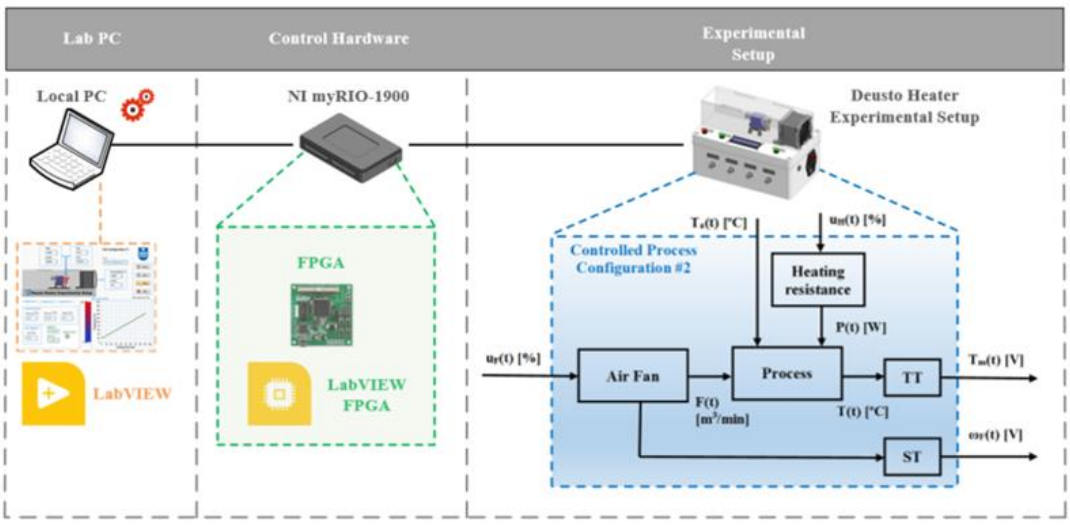


Figure 22. Scheme of FPGA-based closed-loop implementation of temperature control with Controlled Process Configuration #2.

Similar to the previous example, the controlled process in this configuration also exhibits a fractional behaviour. Considering the process reaction curve in Figure 13 obtained for a certain operating point by using a typical open-loop step-test experiment, the controlled process can be modeled by an FFOPDT or a FOPDT model, as detailed in Section 4.

In [68,69], tuning methods for PI and fractional-order PI controllers in the spirit of the kappa-tau tuning rules were presented. An extension of the well-known modified Ziegler-Nichols tuning rules for fractional-order PI controllers is presented in [70], where an interpretation of these tuning rules as methods where one point of the Nyquist curve is positioned in a desired point is also given.

Without loss of generality, the tuning rules for PI and fractional-order PI controllers in the frequency domain proposed by Gude and Kahoraho in [69] will be used in this example. When considering this frequency domain method, the process dynamics is characterized by three simple parameters, i.e., the static gain K, and the gain K_{180} and the frequency ω_{180} where the process lag is 180°.

For the time-domain methods in the previous section, the normalized dead-time τ has been used as a parameter to characterize the process. The corresponding frequency-domain parameter is the gain ratio κ, which is defined as follows:

$$\kappa = \frac{K_{180}}{K} = \frac{|G(\omega_{180})|}{G(0)} \tag{17}$$

where G(s) is the open-loop transfer function of the controlled process. This parameter gives useful information about the process. Processes with small κ can be considered easy to control and the difficulty in controlling the system increases as κ increases.

These tuning rules have been devised in order to minimize the performance criterion, which is mathematically expressed as a measure of the system ability to handle low-frequency load disturbances, with a robustness constraint on the maximum sensitivity function M_S.

According to the considered tuning method, the identified frequency domain parameters of the controlled process at this operating point, $\theta_P = \{K, K_{180}, T_{180}\}$, which are necessary to apply the considered tuning methods, are: $K = -1.30$ °C/%, $K_{180} = 0.053$, and $T_{180} = 13.93$ s. Therefore, the gain ratio is $\kappa = 0.041$.

The fractional order λ of the controller is practically constant for a certain range of κ, as indicated in [69], and the tuning rules for a fractional-order PI controller and $M_S = 1.4$ can be summarized as follows:

$$\begin{gathered} K_P K_{180} = 0.29 - \frac{0.085}{\kappa + 0.34} \\ \frac{T_i}{T_{180}} = 0.086 + \frac{0.15}{\kappa + 0.058} \\ \lambda = 1.12 \end{gathered} \tag{18}$$

Considering the estimated process parameters and the value of the gain ratio for the controlled process, the controller parameters obtained with these tuning rules are $\theta_C = \{K_P, K_i, \lambda\}$, with $K_P = 1.2$, $K_i = 0.056$, and $\lambda = 1.12$.

On the other hand, the tuning rules for a PI controller and $M_S = 1.4$ can be summarized as follows:

$$\begin{gathered} K_P K_{180} = 0.19 - \frac{0.038}{\kappa + 0.25} \\ \frac{T_i}{T_{180}} = 0.11 + \frac{0.073}{\kappa + 0.022} \end{gathered} \tag{19}$$

Considering the estimated process parameters and the value of the gain ratio κ for the controlled process, the controller parameters obtained with these tuning rules are $\theta_C = \{K_P, K_i\}$, with $K_P = 1.12$ and $K_i = 0.063$.

In order to implement this fractional-order PI controller on the FPGA-based real-time target, its discrete form is required. The discretization approach that has been used consists of the ninth-order recursive Tustin method [5], with a sampling period T = 0.01 s.

The discretization is based on the approximation of the fractional operator s^{α} with the Tustin generating function, which is given by:

$$s^{\alpha} = \left(\frac{2}{T}\right)^{\alpha} \left(\frac{1 - z^{-1}}{1 + z^{-1}}\right)^{\alpha} \tag{20}$$

By applying the Muir-recursion method [5], the following approximation of the fractional-order operator is obtained:

$$s^{\alpha} = \left(\frac{2}{T}\right)^{\alpha} \frac{A\left(z^{-1},\ \alpha\right)}{A(z^{-1}, -\alpha)} \tag{21}$$

where the polynomial A can be computed recursively according to the Muir-recursion method.

The discrete transfer function for the fractional-order PI controller will have the following structure:

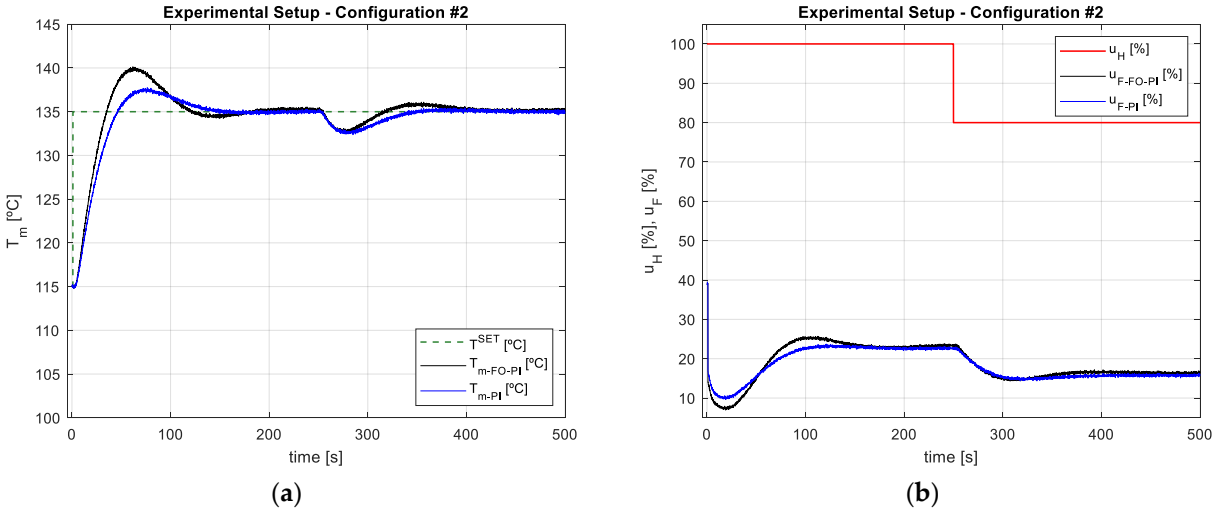
$$G_{FO-PI}\left(z^{-1}\right) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_{10} z^{-10}}{1 + b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_{10} z^{-10}} \tag{22}$$

For the implementation of the fractional-order PI control algorithm on the FPGA target, a recursive relation for the control signal is derived [26]:

$$u(k) = a_0 e(k) + a_1 e(k-1) + \ldots + a_{10} e(k-10) - -b_1 u(k-1) - b_2 u(k-2) - \ldots - b_{10} u(k-10) \tag{23}$$

where u(k) and e(k) are the control and error signals, respectively.

The closed-loop experimental results, obtained considering the operating point at which both the integer- and the fractional-order PI controllers have been designed, have been presented in Figure 23a for servo and regulatory control. In this example, the closed-loop test for servo control consists of a setpoint change T_{SP} from 115 to 135 °C at time t = 0 s, while the closed-loop test for regulation consists of a change in the u_H signal from 100 to 80% at time t = 250 s. The corresponding control signals for both controllers, $u_{F\text{-}PI}$ and $u_{F\text{-}FO\text{-}PI}$, and disturbance signal u_H are given in Figure 23b.



(**a**) (**b**)

Figure 23. Closed-loop FPGA-based experimental results for servo and regulatory control, considering a PI and a fractional-order PI controller: (**a**) Temperature in the heat block; (**b**) control signal u_F and disturbance u_H.

Figure 23a shows that both FPGA-based implemented controllers offer good performance, especially for regulatory control since both tuning methods are robust approaches that seek to optimize load disturbance rejection.

5.2.3. Communication between Windows HMI, RT Processor, and FPGA

As previously mentioned, in this section FPGA-based integer- and fractional-order controllers have been implemented using the proposed hardware architecture. However, for the complete operation and temperature control of the Deusto HES, different functionalities must be implemented at the FPGA, RT processor, and computer level, using the most advanced programming techniques of LabVIEW.

The objective of this part is to provide an overview of the communication between the different hardware resources, i.e., windows HMI, RT processor, and FPGA, in the context of the proposed control hardware architecture.

Figure 24 shows the complete scheme of functionalities implemented at software level using the hardware control architecture that has been proposed in this paper.

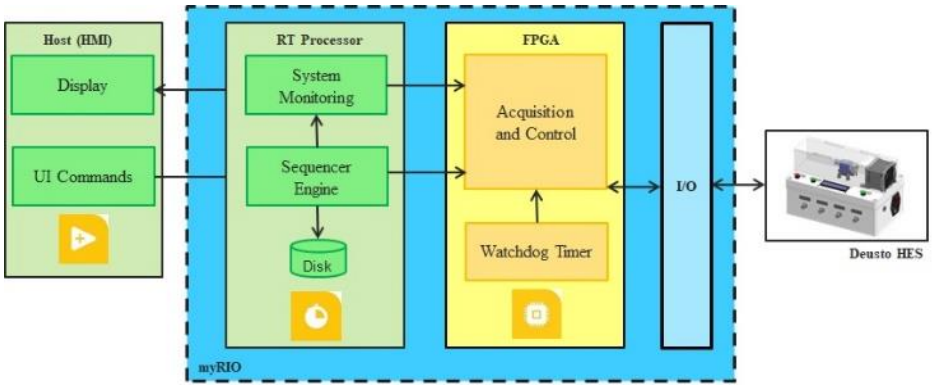


Figure 24. Scheme of the hardware architecture used for operation and implementation of control algorithms applied to the laboratory.

5.2.4. FPGA

The code on the FPGA is implemented using LabVIEW FPGA. The functions that are executed at the FPGA level include the deterministic data acquisition and control loops, which use low-level functions, i.e., implementation of the integer- and fractional-order control algorithm, and data acquisition through the FPGA I/O interface. Another function running on the FPGA is the watchdog timer, which provides a manner of setting the hardware device outputs into known safe-states in case of a system failure. This fail-safe feature provided by the watchdog timer safeguards the hardware connected to the experimental setup in case the control system stops operating as expected.

5.2.5. RT Processor

Code running on the RT processor is implemented with LabVIEW RT. This code executes functions associated with system monitoring, sequencer engine, and data disk storage. A sequencer is a design pattern used for executing LabVIEW code sequentially. This software structure is a FIFO RT design pattern that manages the communication between host and FPGA.

5.2.6. Host (HMI)

The code that runs on the computer is implemented with LabVIEW. This code executes functions of the graphical user interface (GUI), i.e., those related to the visualization of variables, states and graphs, and those related to the commands to be executed.

In this context, the following section deals with the GUI and shows the appearance of the FPGA-based application with the different functionalities and possibilities. All the functions available in the FPGA-based control application are also detailed. Communication between the host and the hardware device is managed by the RT processor and is bidirectional:

- Data that is sent from the control hardware to the host for all those display tasks.

- UI commands sent from the GUI interface to the control hardware to be processed properly.

5.3. LabVIEW-based Application

In this section, the control-related applications implemented in LabVIEW for the different control technologies available in the control hardware architecture are described. The main features presented by each of the implemented applications are also illustrated.

Deusto HES control

Three different applications have been developed to control the controlled process with the considered laboratory setup in each one of the configurations, as shown in the right part of the main menu in Figure 16:

1. PC control mode.
2. Microprocessor control mode.
3. FPGA control mode.

The appearance of the graphical user interface for the FPGA-based control application is shown in Figure 25 for illustrative purposes. However, the following functionalities are common for any of the available control modes, i.e., computer-, microprocessor-, and FPGA-based control:

- Select Controlled Process Configuration #1, #2, or #3 to operate the equipment.
- Set the instantaneous values of u_H [%] or u_F [%], which, depending on the configuration being used, can be considered as a disturbance to the process.
- Visualize time responses for temperature T(t) [°C] and set point T_{SP}(t) [°C] in a graph.
- Visualize time responses for signals u_H(t) [%] and u_F(t) [%] in another graph. In Figure 25, as configuration #1 is selected, u_H and u_F will be manipulated variable and disturbance, respectively
- Select the control operation mode (manual or auto).
- Display the instantaneous value of the various process variables: P [W], u_H [%], T_m [V], T [°C], ω_F [min^{-1}], and u_F [%].
- Export data in graphs to a text- or an excel-format file.

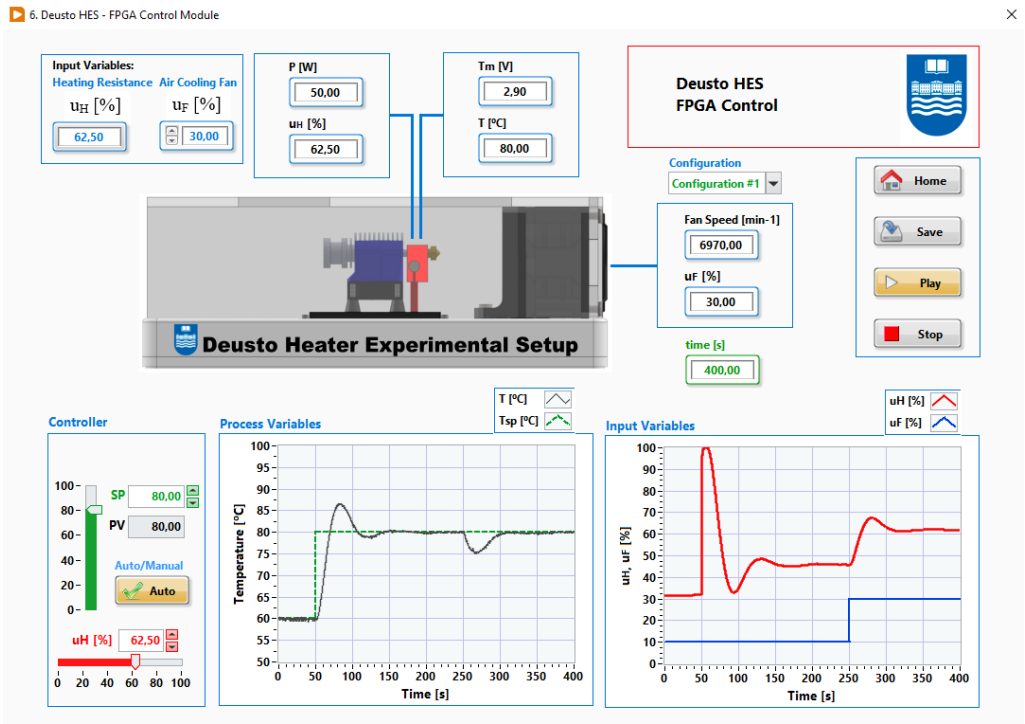


Figure 25. LabVIEW-based application for FPGA control.

5.4. Discussion

As mentioned above, one of the objectives of this paper is to present a novel control hardware architecture for the practical implementation of integer- and fractional-order control algorithms on a real-time target.

In this section, the proposed hardware control architecture has been used to validate its effectiveness for the implementation of integer- and fractional-order control algorithms. More specifically, integer- and fractional-order PI controllers have been implemented on microprocessor- and FPGA-based real-time targets, which constitute part of the proposed hardware architecture, and have been applied to the temperature-based experimental setup under two different configurations.

Note that the thermal process exhibits fractional-order behaviour in both configurations, whose dynamics in both cases are different and nonlinear since the process dynamics vary as a function of the operating point.

Although there is a wide variety of tuning methods for integer- and fractional-order PI controllers in the literature that use different approaches, without loss of generality, in order to design the PI controllers the tuning rules that have been used are characterized by their simplicity, and they are based on process information that can be determined using a simple step-input experiment for the time-domain method or using a relay-feedback experiment for the frequency-domain method. In both cases, the used tuning rules consider the well-known trade-off between performance and robustness.

In this section, the authors have opted to focus on the implementation issues of integer- and fractional-order PI controllers on real-time targets, since the use of fractional-order PID controllers in industry is currently low mainly due to the aforementioned implementation issues. Some final remarks about the implementation of fractional-order controllers on real-time targets are discussed below.

In this work, both integer- and fractional-order PI controllers have been implemented in microprocessor- and FPGA-based real-time targets, validating the effectiveness of the proposed hardware architecture for this purpose. In addition, a straightforward procedure for the development of a fractional-order PI controller has been provided and, at the same time, the implementation issues on both real-time platforms have been addressed.

The main advantage of this hardware architecture over other alternatives is that it provides three control technologies in the same hardware equipment, which increases significantly its possibilities for the training with the aforementioned technologies in the development of identification and control algorithms. In addition, all of these three control technologies use the same programming language: LabVIEW. In this way the engineer can concentrate on the implementation of control algorithms instead of being concerned with learning other programming languages for each hardware platform.

Fractional-order controllers present the fractional $s^{\pm\alpha}$ operators that involve infinite memory, i.e., an infinite number of terms are necessary to approximate their dynamic behaviour, either in the time or in the frequency domain. However, practical applications require realizing such operators with an algorithm of finite duration, which implies that, compared to the original, these realizations are valid only in a certain frequency range. If the case of real orders is considered, the main methods for realizing the fractional derivative and integral operators have been reviewed in this paper. Under this point of view, the discretization of the fractional-order controller takes on a key point in development of the final implementation, whatever the device on which it is to be implemented.

In the technical literature, one can verify that the implementation of control algorithms on microprocessor-based platforms is a widespread option and has become dominant in low-rate applications, which are characterized by not having major constraints in terms of power consumption or space saving. For some time now, FPGA devices have become more and more common as these constraints are becoming more and more common.

The choice of a microprocessor or a FPGA as an implementation device is generally based on the requirements of the application. FPGA provides important benefits, such as: reduced dimension of the device, fast execution time, high computational speed, low power

consumption, which makes possible further development of portable equipment, and simpler hardware implementation compared to other devices such as DSPs, which require dedicated peripherals and possibly even an operating system, or external memory modules.

6. Conclusions

This paper proposes a novel control hardware architecture aimed to the implementation of fractional-order identification and control algorithms.

Although there are several hardware architectures available and each of them has its own characteristics, the one proposed in this paper has some features that make it unique. A discussion about the advantages of this hardware architecture over other available alternatives is provide.

In order to demonstrate the applicability and effectiveness of the proposed control hardware architecture, the design and experimental validation of integer- and fractional-order identification and control algorithms implemented in various control technologies, which are available in the proposed hardware architecture, have been applied to a temperature-based experimental setup.

More specifically, FOPDT and FFOPDT model identification algorithms based on the process reaction curve have been implemented on microprocessor-based hardware, thus verifying their applicability in an industrial environment.

On the other hand, integer- and fractional-order PI controllers have been implemented on the microprocessor- and FPGA-based hardware, respectively. In both cases, the way to implement the corresponding controllers in the different real-time targets and the specific implementation issues in each hardware platform have been discussed.

The main contribution of this work stands in providing a practical and efficient hardware architecture to implement fractional-order identification and control algorithms in different control technologies, helping to bridge the gap between software-based fractional-order identification and control simulations and real-time hardware solutions.

In the authors' opinion, this type of control hardware prepares engineers in the use of control technologies and the realization of low-cost embedded systems of fractional-order controllers that will encourage their industrial use.

Finally, some conclusions and final remarks have been offered in the industrial context.

Author Contributions: Conceptualization, J.J.G. and P.G.B.; methodology, J.J.G. and P.G.B.; software, J.J.G.; investigation, J.J.G.; writing—original draft preparation, J.J.G.; writing—review and editing, J.J.G. and P.G.B.; supervision, J.J.G. and P.G.B.; project administration, P.G.B.; funding acquisition, P.G.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Basque Government, through the Consolidated Research Group D4K—Deusto for Knowledge (IT1528-22) and the ELKARTEK program (REMEDY, grant no. KK-2021/00091).

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to acknowledge the support of Unai Conejo as Lab Assistant in the construction of the Deusto HES prototype.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rossiter, J.A.; Pasik-Duncan, B.; Dormido, S.; Vlacic, L.; Jones, B.; Murray, R. A survey of good practice in control education. *Eur. J. Eng. Educ.* **2018**, *43*, 801–823. [CrossRef]
2. Gude, J.J.; Kahoraho, E. An innovative equipment for practical teaching in process control. *IFAC Proc. Vol.* **2010**, *42*, 233–239. [CrossRef]
3. Sánchez-Peña, R.S.; Quevedo, J.; Puig, V. *Identification and Control: The Gap between Theory and Practice*; Springer: London, UK, 2007.
4. Tepljakov, A. *Fractional-Order Modeling and Control of Dynamic Systems*; Springer: Cham, Switzerland, 2017.
5. Monje, C.A.; Chen, Y.Q.; Vinagre, B.M.; Xue, D.; Feliu, V. *Fractional-Order Systems and Controls*; Springer: London, UK, 2010.
6. Victor, S.; Malti, R.; Garnier, H.; Oustaloup, A. Parameter and differentiation order estimation in fractional models. *Automatica* **2013**, *49*, 926–935. [CrossRef]

7. Thomson, D.; Padula, F. Introduction to Fractional-Order Control: A Practical Laboratory Approach. *IFAC-Pap.* **2022**, *55*, 126–131. [CrossRef]
8. Sierociuk, D.; Dzieliński, A.; Sarwas, G.; Petras, I.; Podlubny, I.; Skovranek, T. Modelling heat transfer in heterogeneous media using fractional calculus. *Phil. Trans. R. Soc. A* **2013**, *371*, 20120146. [CrossRef]
9. Macias, M.; Sierociuk, D. Fractional order calculus for modeling and fractional PID control of the heating process. In Proceedings of the 13th International Carpathian Control Conference (ICCC 2012), High Tatras, Slovakia, 28–31 May 2012. [CrossRef]
10. Malek, H.; Luo, Y.; Chen, Y.Q. Identification and tuning fractional order proportional integral controllers for time delayed systems with a fractional pole. *Mechatronics* **2013**, *23*, 746–754. [CrossRef]
11. Li, Z.; Zhao, T.; Chen, Y.Q. A low cost research platform for modeling and control of multi-input multi-output fractional order dynamic systems. In Proceedings of the International Conference on Fractional Differentiation and Its Applications (ICFDA 2014), Catania, Italy, 23–25 June 2014. [CrossRef]
12. Radici, A.; Chen, Y.Q.; Padula, F.; Visioli, A. A Laboratory Setup for an Introduction to Fractional Order Systems. *IFAC-Pap.* **2019**, *52*, 62–67. [CrossRef]
13. Podlubny, I. Fractional-order systems and $PI^{\lambda}D^{\mu}$ controllers. *IEEE Trans. Autom. Control* **1999**, *44*, 208–214. [CrossRef]
14. Shah, P.; Agashe, S. Review of fractional PID controller. *Mechatronics* **2016**, *38*, 29–41. [CrossRef]
15. Tepljakov, A.; Hosseinnia, S.H.; Gonzalez, E.; Alagoz, B.B.; Yeroglu, C.; Petlenkov, E. FOPID controllers and their industrial applications: A survey of recent results. *IFAC-Pap.* **2018**, *51*, 25–30. [CrossRef]
16. Yusuf, A.; Qureshi, S.; Feroz Shah, S. Mathematical analysis for an autonomous financial dynamical system via classical and modern fractional operators. *Chaos Solitons Fractals* **2020**, *132*, 109552. [CrossRef]
17. Xu, C.; Liao, M.; Li, P.; Guo, Y.; Liu, Z. Bifurcation Properties for Fractional Order Delayed BAM Neural Networks. *Cogn. Comput.* **2021**, *13*, 322–356. [CrossRef]
18. Li, H.L.; Zhang, L.; Hu, C.; Jiang, Y.L.; Teng, Z. Dynamical analysis of a fractional-order predator-prey model incorporating a prey refuge. *J. Appl. Math. Comput.* **2017**, *54*, 435–449. [CrossRef]
19. Chevalier, A.; Francis, C.; Copot, C.; Ionescu, C.M.; De Keyser, R. Fractional-order PID design: Towards transition from state-of-art to state-of-use. *ISA Trans.* **2019**, *84*, 178–186. [CrossRef]
20. Bergman, T.L.; Lavine, A.S.; Incropera, F.P.; DeWitt, D.P. *Fundamentals of Heat and Mass Transfer*, 8th ed.; Wiley: Hoboken, NJ, USA, 2017.
21. Skogestad, S. *Chemical and Energy Process Engineering*, 1st ed.; CRC Press: Boca Raton, FL, USA, 2009.
22. Możaryn, J.; Petryszyn, J.; Ozana, S. PLC based fractional-order PID temperature control in pipeline: Design procedure and experimental evaluation. *Meccanica* **2021**, *56*, 855–871. [CrossRef]
23. Monje, C.A.; Vinagre, B.M.; Santamaría, G.E.; Tejado, I. Auto-tuning of fractional order PIλDμ·controllers using a PLC. In Proceedings of the IEEE Conference on Emerging Technologies & Factory Automation (ETFA 2009), Palma de Mallorca, Spain, 22–25 September 2009. [CrossRef]
24. Rybarczyk, D. PLC implementation of fractional PI controller in positioning of electrohydraulic servodrive. *Control Cybern.* **2016**, *45*, 301–316.
25. Monje, C.A.; Vinagre, B.M.; Feliu, V.; Chen, Y.Q. Tuning and auto-tuning of fractional order controllers for industry applications. *Control Eng. Pract.* **2008**, *16*, 798–812. [CrossRef]
26. Caponetto, R.; Dongola, G.; Fortuna, L.; Petrás, I. *Fractional Order Systems: Modeling and Control Applications*; World Scientific Publishing Company: Singapore, 2010.
27. Oprzędkiewicz, K.; Rosół, M.; Żegleń-Włodarczyk, J. The Frequency and real-time Properties of the Microcontroller Implementation of fractional-order PID Controller. *Electronics* **2021**, *10*, 524. [CrossRef]
28. Copot, C.; Ionescu, C.M.; Muresan, C.I. *Image-Based and Fractional-Order Control for Mechatronic Systems*; Springer: Cham, Switzerland, 2020.
29. Muresan, C.I.; Folea, S.; Mois, G.; Dulf, E.H. Development and implementation of an FPGA based fractional order controller for a DC motor. *Mechatronics* **2013**, *23*, 798–804. [CrossRef]
30. Muresan, C.I.; Folea, S.; Mois, G. Optimal implementation of advanced control methods on FPGA targets. In Proceedings of the 2013 IEEE RIVF International Conference on Computing & Communication Technologies-Research, Innovation, and Vision for Future (RIVF 2013), Hanoi, Vietnam, 10–13 November 2013. [CrossRef]
31. Wang, B.; Wang, S.; Peng, Y.; Pi, Y.; Luo, Y. Design and High-Order Precision Numerical Implementation of Fractional-Order PI Controller for PMSM Speed System Based on FPGA. *Fractal Fract.* **2022**, *6*, 218. [CrossRef]
32. Kalúz, M.; García-Zubía, J.; Fikar, M.; Čirka, Ľ. A Flexible and Configurable Architecture for Automatic Control Remote Laboratories. *IEEE Trans. Learn. Technol.* **2015**, *8*, 299–310. [CrossRef]
33. NI myRIO-1900 User Guide and Specifications. Available online: http://www.ni.com/pdf/manuals/376047c.pdf (accessed on 10 November 2022).
34. Gude, J.J.; Bringas, P.G. Proposal of a control hardware architecture for implementation of fractional-order controllers. In Proceedings of the 16th International Conference Dynamical Systems Theory and Applications (DSTA 2021), Lodz, Poland, 6–9 December 2021.
35. Koszewnik, A.; Pawłuszewicz, E.; Ostaszewski, M. Experimental Studies of the Fractional PID and TID Controllers for Industrial Process. *Int. J. Control Autom. Syst.* **2021**, *19*, 1847–1862. [CrossRef]

36. Dastjerdi, A.A.; Saikumar, N.; HosseinNia, S.H. Tuning guidelines for fractional order PID controllers: Rules of thumb. *Mechatronics* **2018**, *56*, 26–36. [CrossRef]
37. Dzieliński, A.; Sierociuk, D.; Sarwas, G. Some applications of fractional order calculus. *Bull. Pol. Acad. Sci. Techn. Sci.* **2010**, *58*, 583–592. [CrossRef]
38. Machado, J.A.T.; Silva, M.F.; Barbosa, R.S.; Jesus, I.S.; Reis, C.M.; Marcos, M.G.; Galhano, A.F. Some Applications of Fractional Calculus in Engineering. *Math. Probl. Eng.* **2010**, 639801. [CrossRef]
39. Sabatier, J. Modelling Fractional Behaviours Without Fractional Models. *Front. Control Eng.* **2021**, *2*, 7. [CrossRef]
40. Podlubny, I. *Fractional Differential Equations*; Academic Press: San Diego, CA, USA, 1999.
41. Yuan, J.; Ding, Y.; Fei, S.; Chen, Y.Q. Identification and parameter sensitivity analyses of time-delay with single-fractional-pole systems under actuator rate limit effect. *Mech. Syst. Signal Process.* **2022**, *163*, 108111. [CrossRef]
42. Gude, J.J.; García Bringas, P. Proposal of a General Identification Method for Fractional-Order Processes Based on the Process Reaction Curve. *Fractal Fract.* **2022**, *6*, 526. [CrossRef]
43. Åström, K.J.; Hägglund, T. *Advanced PID Control*; The Instrumentation, Systems, and Automation Society; ISA: Research Triangle Park, NC, USA, 2006.
44. Gude, J.J.; García Bringas, P. Influence of the Selection of Reaction Curve's Representative Points on the Accuracy of the Identified Fractional-Order Model. *J. Math.* **2022**, 7185131. [CrossRef]
45. Gude, J.J.; García Bringas, P. Effect of the central point on the accuracy of a fractional-order model identification method based on the process reaction curve. In Proceedings of the 1st International Conference on Mathematical Modelling in Mechanics and Engineering (ICME 2022), Belgrade, Serbia, 8–10 September 2022.
46. Alfaro, V.M. Low-order models' identification from the process reaction curve. *Cienc. Y Tecnol.* **2006**, *24*, 197–216.
47. Vitecková, M.; Vitecek, A.; Smutny, L. Simple PI and PID controllers tuning for monotone self-regulation plants. *IFAC Proc. Vol.* **2000**, *33*, 259–264. [CrossRef]
48. Guevara, E.; Meneses, H.; Arrieta, O.; Vilanova, R.; Visioli, A.; Padula, F. Fractional order model identification: Computational optimization, In Proceedings of IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA 2015), Luxembourg city, Luxembourg, 8–11 September 2015. [CrossRef]
49. Alagoz, B.B.; Tepljakov, A.; Ates, A.; Petlenkov, E.; Yeroglu, C. Time-domain identification of one noninteger order plus time delay models from step response measurements. *Int. J. Model. Simul. Sci. Comput.* **2019**, *10*, 1941011. [CrossRef]
50. Krishna, B.T. Studies on fractional order differentiators and integrators: A survey. *Signal Process* **2011**, *91*, 386–426. [CrossRef]
51. Elwakil, A.S. Fractional-order circuits and systems: An emerging interdisciplinary research area. *IEEE Circuits Syst. Mag.* **2010**, *10*, 40–50. [CrossRef]
52. Maundy, B.; Elwakil, A.S.; Freeborn, T.J. On the practical realization of higher-order filters with fractional stepping. *Signal Process* **2011**, *91*, 484–491. [CrossRef]
53. Oustaloup, A.; Levron, F.; Mathieu, B.; Nanot, F.M. Frequency-band complex noninteger differentiator: Characterization and synthesis. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **2000**, *47*, 25–39. [CrossRef]
54. Chen, Y.Q.; Vinagre, B.M.; Podlubny, I. Continued Fraction Expansion Approaches to Discretizing Fractional Order Derivatives—An Expository Review. *Nonlinear Dyn.* **2004**, *38*, 155–170. [CrossRef]
55. Charef, A.; Sun, H.H.; Tsao, Y.Y.; Onaral, B. Fractal system as represented by singularity function. *IEEE Trans. Autom. Control* **1992**, *37*, 1465–1470. [CrossRef]
56. Vinagre, B.M.; Podlubny, I.; Hernández, A.; Feliu, V. Some Approximations of fractional order operators used in control theory and applications. *Fract. Calc. Appl. Anal.* **2000**, *3*, 231–248.
57. Tenreiro Machado, J.A.; Galhano, A.M.; Oliveira, A.M.; Tar, J.K. Approximating fractional derivatives through the generalized mean. *Commun. Nonlinear Sci. Numer. Simul.* **2009**, *14*, 3723–3730. [CrossRef]
58. Vinagre, B.M.; Chen, Y.Q.; Petras, I. Two direct Tustin discretization methods for fractional-order differentiator/integrator. *J. Frankl. Inst. Eng. Appl. Math.* **2003**, *340*, 349–362. [CrossRef]
59. Chen, Y.Q.; Moore, K.L. Discretization schemes for fractional-order differentiators and integrators. *IEEE Trans. Circuits Syst. I: Fundam. Theory Appl.* **2002**, *49*, 363–367. [CrossRef]
60. Gupta, M.; Varshney, P.; Visweswaran, G.S. Digital fractional-order differentiator and integrator models based on first-order and higher order operators. *Int. J. Circuit Theory Appl* **2011**, *39*, 461–474. [CrossRef]
61. Das, S.; Pan, I. *Fractional Order Signal Processing: Introductory Concepts and Applications*; Springer Briefs in Applied Sciences and Technology; Springer: Berlin/Heidelberg, Germany, 2012.
62. Rana, K.P.S.; Kumar, V.; Mittra, N.; Pramanik, N. Implementation of fractional order integrator/differentiator on field programmable gate array. *Alex. Eng. J.* **2016**, *55*, 1765–1773. [CrossRef]
63. Muresan, C.I.; Mois, G.; Folea, S.; Ionescu, C.M. Alternative implementations of a fractional order control algorithm on FPGAs, in Proceedings of the 2013 International Conference on Reconfigurable Computing and FPGAs (ReConFig), Mexico, 9–11 December 2013.
64. Birs, I.; Muresan, C.; Nascu, I.; Ionescu, C. A Survey of Recent Advances in Fractional Order Control for Time Delay Systems. *IEEE Access* **2019**, *7*, 30951–30965. [CrossRef]
65. Hägglund, T.; Åström, K.J. Revisiting the Ziegler-Nichols Tuning Rules for PI Control. *Asian J. Control* **2002**, *4*, 364–380. [CrossRef]

66. Bhaskaran, T.; Chen, Y.Q.; Xue, D. Practical Tuning of Fractional Order Proportional and Integral Controller (I): Tuning Rule Development. In Proceedings of the ASME 2007 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Las Vegas, NV, USA, 4–7 September 2007; Volume 5, pp. 1245–1258.
67. Panagopoulos, H.; Åström, K.J.; Hägglund, T. Design of PID controllers based on constrained optimization. *IEE Proc. Control Theory Appl.* **2002**, *149*, 32–40. [CrossRef]
68. Gude, J.J.; Kahoraho, E. Comparison between Ziegler-Nichols type tuning rules for PI and fractional PI controllers. In Proceedings of the 3rd Seminar for Advanced Industrial Control Applications-SAICA 2009, Madrid, Spain, 16–17 November 2009.
69. Gude, J.J.; Kahoraho, E. Comparison between Ziegler-Nichols type tuning rules for PI and fractional PI controllers—Part 2: The frequency method. In Proceedings of the 3rd Seminar for Advanced Industrial Control Applications-SAICA 2009, Madrid, Spain, 16–17 November 2009.
70. Gude, J.J.; Kahoraho, E. Modified Ziegler-Nichols method for fractional PI controllers. In Proceedings of the IEEE 15th Conference on Emerging Technologies & Factory Automation, Bilbao, Spain, 13–16 September 2010. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.