Universidade de Aveiro Departamento de Engenharia Mecânica
2019

Jonathan Daniel Oliveira

Development of a virtual bench for simulation and monitoring of water heating devices

Desenvolvimento de bancada virtual para simulação e monitorização de dispositivos de aquecimento de água

Esta dissertação teve o apoio dos projetos
UID/EMS/00481/2019-FCT - FCT - Fundação para a Ciência e a Tecnologia; e
CENTRO-01-0145-FEDER-022083 - Programa Operacional Regional do Centro (Centro2020), através do Portugal 2020 e do Fundo Europeu de Desenvolvimento Regional

Universidade de Aveiro Departamento de Engenharia Mecânica
2019

Jonathan Daniel Oliveira

Development of a virtual bench for simulation and monitoring of water heating devices

Desenvolvimento de bancada virtual para simulação e monitorização de dispositivos de aquecimento de água

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Mecânica, realizada sob orientação científica de Jorge Augusto Fernandes Ferreira, Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro, e de Vítor António Ferreira da Costa, Professor Catedrático do Departamento de Engenharia Mecânica da Universidade de Aveiro.

The present study was developed in the scope of the **Smart Green Homes** Project [POCI-01-0247-FEDER-007678], a co-promotion between **Bosch Termotecnologia S.A.** and the **University of Aveiro**. It is financed by Portugal 2020 under the Competitiveness and Internationalization Operational Program, and by the European Regional Development Fund.

o júri / the jury

Presidente / President

Prof. Doutor Nelson Amadeu Dias Martins
Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro

Vogais / Committee

Doutor Manuel José Cabral dos Santos Reis
Professor Associado com Agregação da Universidade de Trás-Os-Montes e Alto Douro

Prof. Doutor Jorge Augusto Fernandes Ferreira
Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro

agradecimentos / acknowledgements

Countless people supported my effort on this thesis.

Professor Jorge Ferreira provided guidance and constant support during the accomplishment of this work and was always available in times of need.

To Smart Green Homes project for financing all of the equipment in order to develop the virtual test bench.

To Luis Rodrigues for giving advice and a second opinion in the different subjects of this work.

To Rui Heitor for helping to build the experimental platform used in the experiences.

A very special thanks to André Quintã for the unconditional support throughout the development of this dissertation.

To my brother who has the ability to bring me joy even in bad days.

A special thanks to my close friend André Garcia, who I also consider a brother, has the strength and emotion to motivate me in many different ways.

A special thanks to my girlfriend Patrícia Simões who was always present for me in these last five years, and is a person that I know I can rely on.

And for last I want to thank my mother for the wide support and help during all these years.

keywords Thermal systems; dynamic models, control strategies; hardware-in-the-loop; water heating devices; virtual test bench; real time simulation

abstract

A controller for thermal systems is normally equipped with many facilities to make it flexible and the heating systems more cost-efficient. This results in a number of input parameters to be given by the user. It is not obvious how to choose appropriate values for these parameters unless the user has a large experience in this field.

Water heating is a very important part of a household's energy use, and tankless gas water heaters (TGWH) are widely used. There are design and engineering challenges to develop more efficient devices, with lower emissions of pollutant gases and providing comfort improvements from the user point of view.

Mathematical and semi-empirical models of the thermal systems were developed in order to simulate the dynamic models of water heating devices. A simulated environment is a less expensive and fastest way of evaluating the relative merits of different control schemes for a given thermal system. A technique to accelerate the process for developing controllers was implemented. Hardware-in-the-loop simulation (HILS) has proved to be very useful to test hardware controllers in virtual environments simulated in real-time.

In the scope of the Smart Green Homes Project, a virtual test bench with a TGWH was proposed to support the multiple phases of controller's development, whether it is to control a real or a virtual system.

The experimental platform was developed to test the implemented hybrid models performance in hardware-in-the-loop simulation experiences. The platform is composed by a TGWH with a group of sensors, by real-time hardware and by a package of software tools for data acquisition and control.

In the final stage of this work, two case studies were carried out, in which the first was dedicated to the validation of the virtual bench concept and the second was to control and monitor a water heating device. Very satisfactory results, from a set of HILS experiences performed in real-time simulations, were obtained for the semi-empirical models proposed.

palavras-chave

Sistemas térmicos; modelos dinâmicos; estratégias de controlo; hardware-in-the-loop; dispositivos de aquecimento de água; bancada de ensaios virtual; simulação em tempo real

resumo

Um controlador para sistemas térmicos está normalmente equipado com muitas instalações para o tornar flexível e os sistemas de aquecimento mais económicos. Isto resulta numa série de parâmetros de entrada a serem dados pelo utilizador. Não é óbvio como escolher valores apropriados para estes parâmetros, a menos que o utilizador tenha uma grande experiência neste campo.

O aquecimento de água é uma parte muito importante do consumo de energia de um agregado familiar, e os aquecedores de água a gás sem tanque (TGWH) são amplamente utilizados. Há desafios de projeto e engenharia para desenvolver dispositivos mais eficientes, com menores emissões de gases poluentes e proporcionando melhorias de conforto do ponto de vista do utilizador.

Foram desenvolvidos modelos matemáticos e semi-empíricos dos sistemas térmicos para simular os modelos dinâmicos dos dispositivos de aquecimento de água.

Um ambiente simulado é uma forma menos dispendiosa e mais rápida de avaliar os méritos relativos de diferentes esquemas de controle para um determinado sistema térmico. Foi implementada uma técnica para acelerar o processo de desenvolvimento de controladores. A simulação Hardware-in-the-loop (HILS) provou ser muito útil para testar controladores de hardware em ambientes virtuais simulados em tempo real.

No âmbito do projecto Smart Green Homes, foi proposta uma bancada de ensaios virtual com um TGWH para apoiar as múltiplas fases de desenvolvimento do controlador, seja para controlar um sistema real ou virtual.

A plataforma experimental foi desenvolvida para testar o desempenho dos modelos híbridos implementados em experiências de simulação hardware-in-the-loop. A plataforma é composta por um TGWH com um grupo de sensores, por hardware em tempo real e por um pacote de ferramentas de software para aquisição e controlo de dados.

Na fase final deste trabalho, foram realizados dois estudos de caso, em que o primeiro foi dedicado à validação do conceito da bancada virtual e o segundo foi para controlar e monitorizar um dispositivo de aquecimento de água. Foram obtidos resultados muito satisfatórios, a partir de um conjunto de experiências HILS realizadas em simulações em tempo real, para os modelos semi-empíricos propostos.

Contents

List of Tables

List of Figures

List of Acronyms

ADC Analog to Digital Conversion

AWHP Air-to-Water-Heat-Pumps

BNC Bayonet Neill-Concelman

DAC Digital to Analog Conversion

DAQ Data Acquisition

DHW Domestic Hot Water

DSP Digital Signal Processing

ECU Electronic Control Unit

ECU Engine Control Unit

GUI Graphical User Interface

HIL Hardware-In-The-Loop

HiMAT Highly Maneuverable Aircraft Technology

IDE Integrated Development Environment

I/O Input/Output

PCI Peripheral Component Interconnect

PD Proportional-Derivative

PI Proportional-Integral

PID Proportional-Integral-Derivative

PWM Pulse Width Modulation

RC Resistor-Capacitor

RCP Rapid Control Prototyping

RTI Real-Time Interface

RTD Resistance Temperature Detector

RTW Real-Time Workshop

R&D Research and Development

SH Space Heating

TGWH Tankless Gas Water Heater

VAC Voltage in Alternating Current

VDC Voltage in Direct Current

UART Universal Asynchronous Receiver-Transmitter

Chapter 1

Introduction

1.1 Motivation

In recent years, we have also seen tremendous growth in the development and use of thermal systems in which fluid flow and transport of energy play a dominant role. These systems arise in many diverse engineering fields such as those related to manufacturing, power generation, air conditioning, and aerospace and automobile engineering. Energy is present in multiple domains, such as mechanical, electrical and chemical. The transfer of energy from one domain to another usually involves a thermal component in some portion of the process. The understanding of thermal energy transfer and conversion is critical to maintain and increase efficiency measures of all aspects of the energy supply chain, consisting of energy conversion, transmission, conservation, and scavenging because it allows for better control of available energy resources. The research communities in the fields of thermodynamics or heat and mass transport number in the thousands, and a great deal of excellent work is being performed across the world [1].

The present development in high-tech industry causes an exponential growth of systems' simulation performance requirements that cannot be covered by the rising simulation computer performance. The current method is to use different levels of abstraction, since a higher level of abstraction increases simulation speed. Modeling is a trade-off between high speed, high accuracy, and low effort. It is easy to create a model covering two of these three attributes but it is nearly impossible to build high speed and high accuracy models spending only a small amount of effort. This trade-off is especially critical if systems' software development is supposed to begin on the simulated system hardware. Because of this, software developers need highly accurate and fast models. The use of dynamic models has been widely used in the development of new control strategies, the ease with which it becomes possible to optimize the actual controllers in different methods of operation for each equipment, including the which hardly appear in practice. This way, it is possible to save some time and expenses, foreseeing the behavior of a given system without the need of experimentation. This type of test is called Hardware-In-The-Loop (HIL) simulation.

Despite all attempts to improve and speed up the development process, platforms are needed to adapt to the different tests. Here is where the test benches are crucial, as they support development and enable the testing of equipment and controllers with the objective of detecting design errors and avoid further damage to the same [2].

1.2 Identifying the poblem

At Bosch TT, there is a great need to test new controllers and equipment at the design stage. This company is well known for its high responsiveness to the needs of the market for the production of water heaters and among other water heating devices. For this, engineers need to develop new products more efficiently and in a short time, which translates into a reduction in development time and the need for new testing tools.

The outlet temperature dynamics of a TGWH depends on several factors as the inlet water temperature, flow rate, thermal power released to the heat cell, thermal inertia of the system and ducts' lengths, between many others which have smaller contributions to the outlet temperature changes. One critical problem of TGWH appears when sudden hot water flow rate changes occur, which can be essentially due to changes in the number of users demanding hot water, to changes on the hot water flow rate demand of one user, and during the starting periods of hot water production. Even TGWH with feedback control of the outlet hot water temperature are not able to reduce the changes in the outlet temperature to acceptable values when fast hot water flow rate changes occur. This happens due to the relatively slow dynamics of the heat cell, and to the time delays on the outlet hot water temperature changes which strongly depend on the hot water flow rate and on the length of the ducts where it is flowing [3], [4].

Test benches play a key role in supporting researchers in the development stage of testing new equipment and controllers to ensure their proper functioning. However, these present some limitations, with emphasis on the fact that they only enable hardware testing at an advanced stage of the project. The general procedure in testing a controller requires to develop all equipment and program the microcontroller to optimize the operation of the same. All this costly process of concept generation, development, construction, programming, and optimization takes time and requires a great deal of collaboration between the various development sectors of a company.

Proposed by Bosch TT collaborators, there was interest in creating a test bench that would be more flexible than the existing ones and could be more useful at an even earlier stage of the project.

1.3 Objectives

In its essence, this work proposes a methodology to develop and implement dynamic models of thermal systems and test controllers with the purpose of executing hardware-in-the-loop experiences.

Next, a concept of the test bench is proposed and, not only it allows to apply the HIL simulation, but also becomes flexible for testing different forms of simulation that may be beneficial in the process of developing and testing controllers.

It is commonly accepted that higher complex models require higher computational performances and larger simulation times. So the effort can be focused on the search for low complexity models, which may be semi-empirical, but have to effectively translate the behavior of the components, especially the most relevant to test controllers. It is then proposed a group of models with different complexities for the components of the thermal systems. In order to adjust the complexity of the global model to the available real-time simulation hardware, experiments with HIL simulation are carried out in order to test new controllers and/or algorithms without the need of implementing external hardware.

On a second note, the controllers possess complex hardware/software systems, which are a combination of existing parts like μCs, memory subsystems, interconnect structures, input/output (I/O) modules and hardware accelerators. The possibility to embed a real μC into a system simulation of a larger technical context would be a great benefit. Compared to a model, the real μC will run with high speed and full accuracy. This is an advantage, especially for early software development. One of the challenges is to embed the μC into the simulation environment so that it is transparent to the system's simulation whether it is a simulated or a real μC.

Among some objectives, those that stand out are:

- Modeling dynamic thermal systems in MATLAB/Simulink and customizing them to the different operating modes;
- Implementing the software and hardware required to perform HIL simulations dedicated to the test of controllers;
- Construction of an experimental platform which enables different types of experiences.

For this purpose, the first case study developed served to prove the proper functionality, validate the concept of the virtual test bench and operating methods through the measurement and control of the temperature inside an oven. The second case study applies to a tankless gas water heater from Bosh TT and has the objective of implementing the techniques previously discussed.

1.4 Dissertation organization

The present dissertation is organized in 7 chapters constituting this introduction in the first chapter. The organization of this document is shown in Figure 1.1.

The following Chapter 2 includes the fundamental concepts and state of art, related work and the basis of the dissertation. Five main topics are identified: thermal systems, modeling thermal systems, Proportional-Integral-Derivative (PID) controllers, hardware-in-the-loop simulation and test benches. These topics are examined in the following five sections. Chapter 3 represents a proposal for the layout of the test bench, as well as a description of the different operating modes. Chapter 4 describes the hardware and software implemented in the experimental platform. Chapters 5 and 6 cover two distinct case studies, but with similar structures. Both explain the general procedure and methods applied to obtain experimental data. At the end of each case study, an analysis of the experimental results is made. Finally, the conclusions and suggestions for future work can be found in Chapter 7.

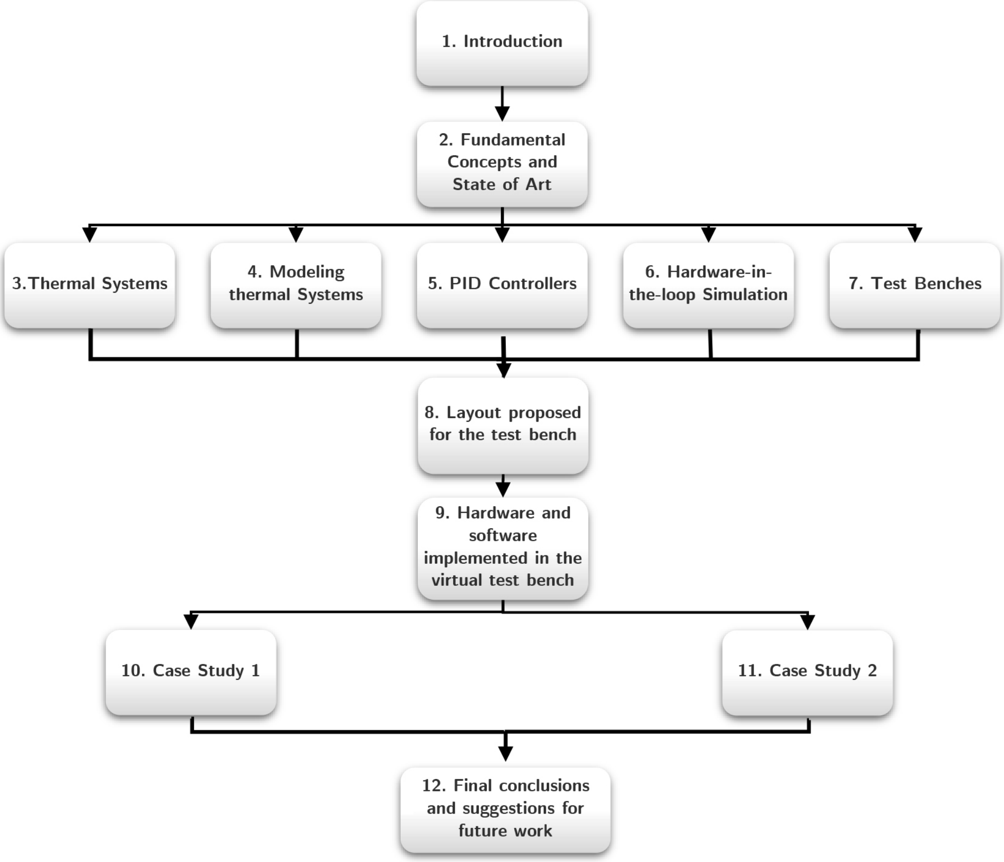

1. Introduction
2. Fundamental Concepts and State of Art
3.Thermal Systems
4. Modeling thermal Systems
5. PID Controllers
6. Hardware-in-the-loop Simulation
7. Test Benches
8. Layout proposed for the test bench
9. Hardware and software implemented in the virtual test bench
10. Case Study 1
11. Case Study 2
12. Final conclusions and suggestions for future work

Figure 1.1: Document Structure

Chapter 2

Fundamental concepts and state of the art

The objective of this chapter is to present some of the fundamental concepts for a better understanding of the work developed. It also shows some brief descriptions of the main concepts which define the sequential development of this dissertation. Sections 2.1 and 2.2 define the starting point for this work and represent the basis and foundation of the entire structure and activities developed. At the end of Sections 2.3, 2.4 and 2.5 related works will be presented.

2.1 Thermal systems

A thermal system consists of multiple units or items that interact with each other. Thus, the system can be used to represent a piece of equipment, such as a heat exchanger or a pump or a larger arrangement with many types of equipment, such as a blast furnace or a cooling tower. It also may represent a complete establishment, such as a power plant, steel plant, or manufacturing assembly line. The two main distinguishing features of a system are constituents that interact with each other and the consideration of the whole entity for analysis and design.

It is important to recognize that thermal systems arise in many and diverse fields of engineering, such as aerospace engineering, manufacturing, power generation, and air conditioning. Consequently, a study of thermal systems usually brings in additional mechanisms and considerations, making the problem much more complex than what might be expected from a thermal sciences study alone.

The analysis of thermal systems is often complex because of the nature of fluid flow and of heat and mass transfer mechanisms that govern these systems. As a result, typical thermal systems have to be approximated, simplified, and idealized in order to make it possible to analyze them and thus obtain the inputs needed for design. Following are some of the characteristics that are commonly encountered in thermal systems and processes: time-dependent, energy losses and irreversibilities, turbulent flow, influence of ambient conditions, complex geometries, multidimensional, change of phase and material structure, variety of energy sources, coupled transport phenomena, variable material properties, nonlinear mechanisms, and complex boundary conditions.

Because of the time-dependent, multidimensional nature of typical systems, the governing equations are generally a set of partial differential equations, with nonlinearity arising due to convection of momentum in the flow, variable properties, and radiative transport. However, approximations and idealizations are used to simplify these equations, resulting in algebraic and ordinary differential equations for many practical situations and relatively simpler partial differential equations for others.

2.2 Modeling thermal systems

Modeling is one of the most crucial elements in the design and optimization of thermal systems. Practical processes and systems are generally very complicated and must be simplified through idealizations and approximations to make a problem amenable to a solution. Modeling also refers to the process of simplifying a given problem so that it may be represented in terms of a system of equations, for analysis, or a physical arrangement for experimentation, it is needed for understanding and predicting the behavior and characteristics of thermal systems. Once a model is obtained, it is subjected to a variety of operating conditions and design variations. If the model is a good representation of the actual system under consideration, the outputs obtained from the model characterize the behavior of the system [1], [5].

The model may be *descriptive* or *predictive*. To explain the function of a device, a working model of an engineering system, such as a robot, an internal combustion engine, a heat exchanger, is used. Frequently, the model may be made of clear plastic or may have a cutaway section to show the internal mechanisms. Such models are known as *descriptive* [1].

Predictive models are of particular engineering interest because they can be used to predict the performance of a given system. The equation governing the cooling of a hot metal sphere immersed in an extensive cold-water environment represents a predictive model because it allows us to obtain the temperature variation with time and determine the dependence of the cooling curve on physical variables such as the initial temperature of the sphere, water temperature, and material properties.

One of the most important considerations in modeling is whether the system can be assumed to be at steady-state, involving no variations with time, or if the time-dependent changes must be taken into account. Since time brings in an additional independent variable, which increases the complexity of the problem, it is important to determine whether these effects can be neglected or not. Most thermal processes are time-dependent, but for several practical circumstances, they may be approximated as steady [1].

2.2.1 Basic mechanisms of heat transfer

The three basic mechanisms of heat transfer (conduction, convection and radiation) will be presented in a rather condensed form because it is assumed that the general concepts of these mechanisms are well known.

Conduction

It is the mechanism of transferring heat within an element or between two due to the molecules vibrating at their mean positions through their thermal level. For this to occur the elements must be in contact with one another. Matter can move or not while transferring heat lay conduction. As temperature rises the conduction heat transfer is higher due to the fact that the interaction between molecules also rises. Molecular density in solids is higher than in liquids and gases, and as a consequence, the heat transfer will also be higher in solids than in liquids or gases [6].

Convection

Convection heat transfer occurs mostly in liquids and gases. This mechanism takes place in the actual motion of matter on, around or through the element. The results in the motion of these states are momentum, energy and mass transfer. This movement may also occur as a result of conduction at the surface causing a fluctuation in the fluid density [6].

Radiation

This process of heat transfer occurs due to electromagnetic waves. There is no requirement needed for the transmitter and receiver to be in contact, radiation may occur even in a vacuum environment. If two elements are visible to each other, both will emit and receive radiation. The heat transfer rate between the two elements will be the algebraic sum of these quantities [6].

2.2.2 Lumped models of thermal systems

Mathematical models of thermal systems are usually derived from the basic energy balance equation that follows the general form

$$\begin{aligned}\begin{pmatrix}\text{rate of energy}\\ \text{stored}\\ \text{within system}\end{pmatrix} &= \begin{pmatrix}\text{heat flow}\\ \text{rate}\\ \text{into system}\end{pmatrix} - \begin{pmatrix}\text{heat flow}\\ \text{rate}\\ \text{out of system}\end{pmatrix}\\ &\quad + \begin{pmatrix}\text{rate of heat}\\ \text{generated}\\ \text{within system}\end{pmatrix} + \begin{pmatrix}\text{rate of work}\\ \text{done}\\ \text{on system}\end{pmatrix} \end{aligned} \tag{2.1}$$

For a stationary system composed of a material of density ρ, specific heat c_p, and constant volume $\mathcal{V}$, the energy-balance equation takes the form

$$\rho c_p \mathcal{V} \frac{dT}{dt} = \dot{Q}_{in}(t) - \dot{Q}_{out}(t) + \dot{Q}_{gen}(t) + \frac{dW}{dt} \tag{2.2}$$

where $\dot{Q}_{in}$ is the inlet rate of heat transfer, $\dot{Q}_{out}$ the outlet rate of heat transfer, $\dot{Q}_{gen}$ the generated rate of heat transfer, T the temperature and W the rate of energy transferred by as work.

Equation 2.2 can be used only if the temperature distribution in the system, or in a part of the system, is uniform that is when the temperature is independent of spatial coordinates, $T(x, y, z, t) =$ T. The assumption about the uniformity of the temperature distribution also implies that the system physical properties, such as density and specific heat, are constant within the system boundaries. Two basic parameters used in lumped models of thermal systems are thermal capacitance and thermal resistance. The thermal capacitance of a thermal system of density ρ, specific heat c_p, and volume $\mathcal{V}$ is

$$C_h = \rho c_p \mathcal{V} = m c_P \tag{2.3}$$

The rate of energy storage in a system of thermal capacitance C_h is

$$\dot{Q}_{stored} = C_h \frac{dT}{dt} \tag{2.4}$$

Physically, thermal capacitance represents a system's ability to store thermal energy. It also provides a measure of the effect of energy storage on the system's temperature. If the system thermal capacitance is large, the rate of temperature change owing to heat influx is relatively low. On the other hand, when the system's thermal capacitance is small, the temperature increases more rapidly with the amount of energy stored in the system [7].

2.3 PID controllers

Since the concept of a PID controller is very familiar in the industrial world, there is no point in explaining the details of it. Therefore, the main features of this control method will be explained in the remainder of this section. This section will conclude with some related work, which will later help in understanding the features and applications of this control method.

2.3.1 Structure

A common definition given to a three-term control is PID controller (P-Proportional, I-Integral, D-Derivative). These are widely used in industrial controllers. Simple and complex industrial systems may include a mainframe that is controlled by a PID model through the main control building block. In the process industries, this controller has been used extensively for the high-volume market.

The introduction of the Laplace transform to study the performance of feedback control systems supported its technological success in the engineering community. The theoretical basis for analyzing the performance of a PID controller is considerably aided by the simple representation of an Integrator by the Laplace transform function $[\frac{1}{s}]$, and a Differentiator using $[s]$. Conceptually, the PID controller is sophisticated and can be represented in three different ways. First, there is a symbolic representation (Figure 2.1(a)), where each of the three terms can be selected to achieve different control actions. Secondly, there is a time domain operator form (Figure 2.1(b)), and finally, there is a Laplace transform version of the PID controller (Figure 2.1(c)). This gives the controller an s-domain operator interpretation and allows the link between the time domain and the frequency domain which defines the PID controller performance [8].

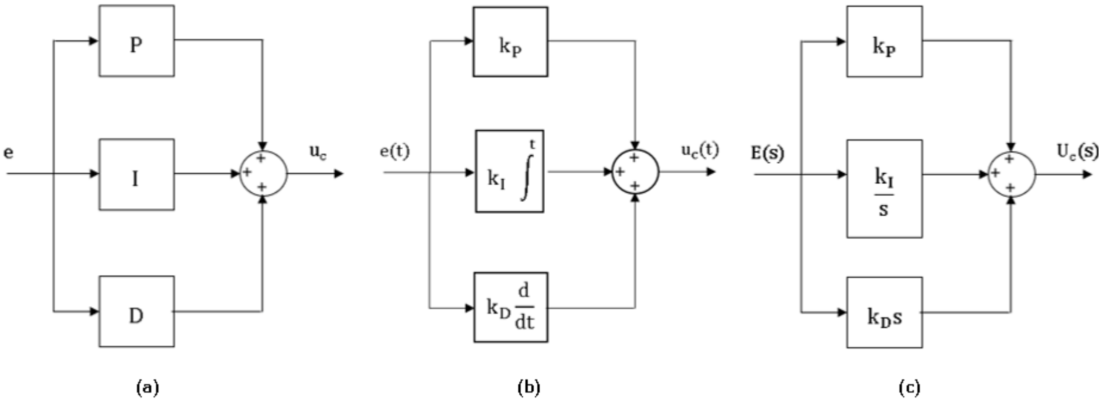


Figure 2.1: PID controller representations (Adapted from [8])

The typical industrial control loop structure can be expressed through a process of loops with more than four engineering components, as can be seen in Figure 2.2.

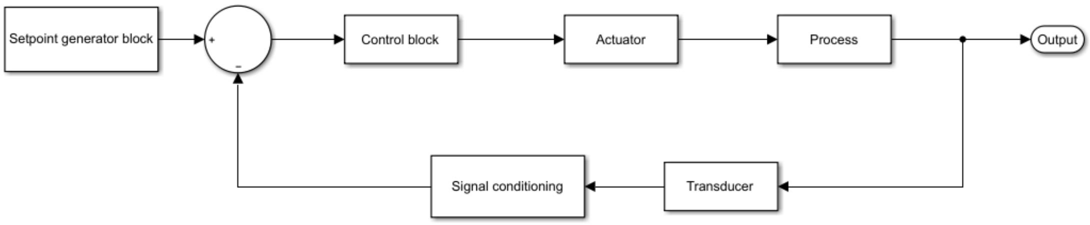


Figure 2.2: Industrial control loop components (Adapted from [8])

The majority of the components can be organized according to the following loop operations [8]:

- Process: represents the actual system for which some specific physical variables are to be controlled or regulated. Typical process industry examples are boilers, furnaces and distillation towers.
- Actuation: the actuator is described by the process unit that supplies material or power input to the process. The amplification is an example of how the actuator can be used. For example, the control signal could be a small movement on a valve stem controlling a large flow of natural gas into a gas-fired industrial boiler.
- Measurement: It is very common to witness that without measurement there is no control. Typically, the measurement process incorporates a transducer and is associated to signal processing components. The transducer will comprise a sensor to detect a specific physical property (such as temperature), and will output a representation of the property in a different physical form (such as voltage). It is possible that the measured output will be a noisy signal and that some of that noise will still manage to pass through the signal-conditioning component of the measurement device into the control loop.
- Control: This expresses the unit designed to create a stable closed-loop system and also achieve some pre-specified dynamic and static process performance requirements. The input to the controller unit is usually an error signal based

on the difference between a desired setpoint or reference signal and the actual measured output.

– Communications: the above units and components in the control loop are all linked together. In small local loops, the control system is usually hardwired, but in spatially distributed processes with distant operational control rooms, computer communication components (networks, transmitters and receivers) will possibly be needed. This aspect of control engineering is not often discussed. However, the presence of communication delays in the loop may be an important obstacle to a better control of the systems' performance.

To specify the controller, the performance objectives of the loop must be considered carefully, but in many situations it is only after the loop, and when it is used for a longer period of production activity that new and unforeseen process problems are identified. Consequently, industrial control engineering often has two stages of activity: (i) control design and commissioning and (ii) post-commissioning control redesign.

Control design and commissioning

The construction of a PID controller design helps to understand its functionality and process. Almost all physical processes are nonlinear in operation, but fortunately, many industrial processes run in conditions of steady operation with the purpose to maximize the system's efficiency. This makes linearisation of nonlinear process dynamics about steady-state system operation, a feasible analysis route for many processes. Subsequently, nonlinearity at different operating conditions can be overcome by gain scheduling or adaptation techniques [8]. Although linearity is the route to the construction of a simple model, even straightforward loops can lead to complicated models if all the components are considered. This will result in a diagram model and can be seen in Figure 2.3, with the respective description of the components and signals presented in Table 2.1.

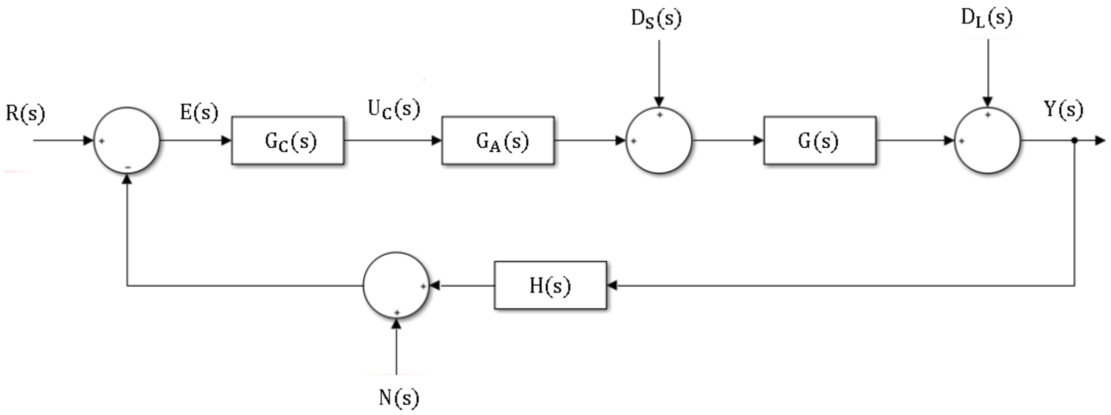


Figure 2.3: Model diagram using Laplace transformations in a loop. (Adapted from [8])

In most cases, this finely detailed block diagram is usually only a step towards the standard control engineering block diagram. It is normal to manipulate, refine, approximate and reduce the models to reach the standard control block diagram of Figure 2.4, where $G_P(s)$ represents the composite process model and $D(s)$ the disturbance signal.

Table 2.1: Description of the loop model diagram from Figure 2.3. (Adapted from [8])

Key	**Components**	**Loop signals**	**Designation**
G(s)	Process model	Y(s)	Process output
		D_L(s)	Load disturbance signal
H(s)	Measurement process model	N(s)	Measurement noise
G_C(s)	Controller unit	R(s)	Setpoint or reference signal
		E(s)	Process error input to controller
		U_C(s)	Controller output
G_A(s)	Actuator unit model	U(s)	Actuator output to process
		D_S(s)	Supply disturbance signal

Using the standard block diagram, the performance objectives will include [8]:

1. The selection of a controller to make the closed-loop system stable.
2. Achieving a reference-tracking objective and making the output follow the reference or setpoint signal.
3. If a process disturbance is present, the controller may have disturbance rejection objectives to attain.
4. Some noise filtering properties may be required in the controller to attenuate any measurement noise associated with the measurement process.
5. A degree of robustness in the controller design to model uncertainty may be required. In this case, the construction of a model and the subsequent model reduction may provide information about the probable level of uncertainty in the model. This information can be used when seeking control loop robust stability and robust control loop performance.

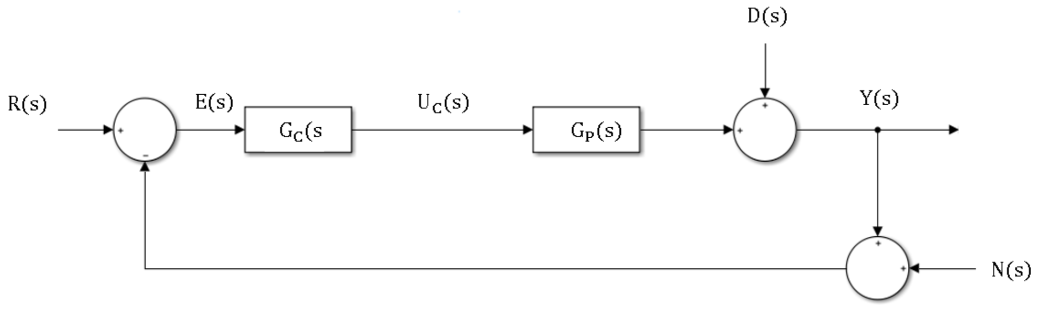


Figure 2.4: Simplified standard model diagram. (Adapted from [8])

2.3.2 The three-term control and functionality

The architecture of input and output signals for the three-term controller can be seen in Figure 2.5. With this, it is possible to examine and study each term of the PID controller.

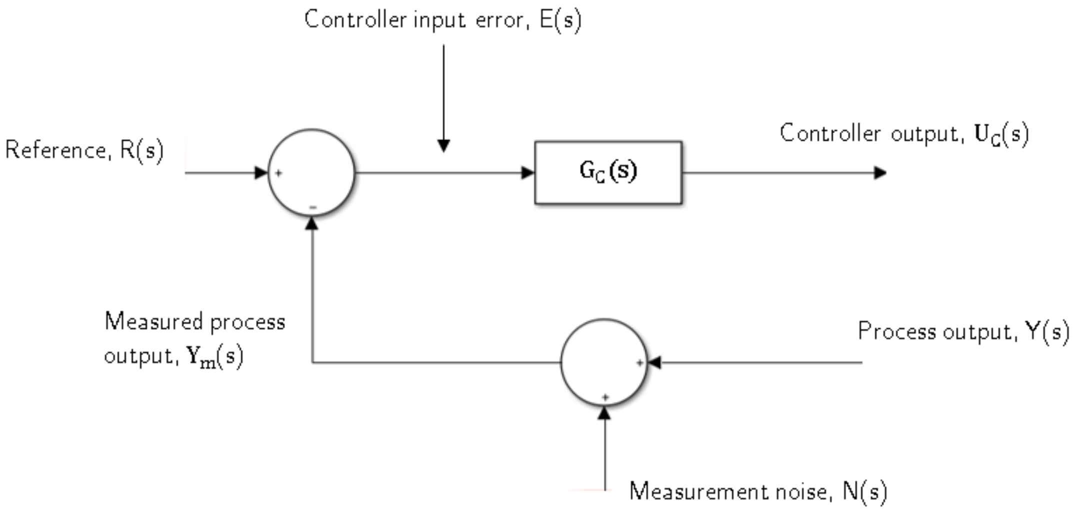

Controller input error, E(s)
Reference, R(s)
+
-
G_C(s)
Controller output, U_C(s)
Measured process output, Y_m(s)
+
+
Process output, Y(s)
Measurement noise, N(s)

Figure 2.5: Inputs and outputs from the controller. (Adapted from [8])

Proportional action

The proportional control mode changes the controller output in proportion to the error. If the error increases, the control action increases proportionally. This is commonly used when the controller action is to be proportional to the size of the process error signal.

The adjustable setting for proportional control is called the Controller Gain (k_P). A higher controller gain increases the amount of proportional control action for a given error. If the controller gain is set too high, the control loop will begin oscillating and become unstable. If set too low, the control loop will not respond adequately to disturbances or setpoint changes.

The time and Laplace domains representations of the proportional controller are represented by the following mathematical equations, respectively [8]:

$$u_C(t) = k_p e(t) \tag{2.5}$$

$$U_C(s) = k_p E(s) \tag{2.6}$$

where the proportional gain is denoted k_p.

Proportional controllers are simple to understand and easy to tune: the controller output is simply the control error times the controller gain, plus a bias. The bias is needed so the controller can maintain a non-zero output while the error is zero (process variable at setpoint). The drawback is offset, which is a sustained error that cannot be eliminated by proportional control alone. Under proportional-only control, the offset will remain present until the operator manually changes the bias on the controller's output to remove the offset. This is known as a manual reset of the controller [9].

Integral action

The need for manual reset led to the development of automatic reset, known as the integral control mode. The function of the integral control mode is to increment or decrement the controller's output over time to reduce the error, as long as there is an error present (process variable not at setpoint). Given enough time, the integral action will drive the controller output until the error is zero.

If the error is large, the integral mode will increment/decrement the controller output at a fast rate; if the error is small, the changes will be slow. For a given error, the speed of the integral action is set by the controller's integral time setting. If the integral time is set too long, the controller will be sluggish. If it is set too short, the control loop will oscillate and becomes unstable.

Integral controllers overcome the shortcoming of proportional control by eliminating offset without the use of excessively large controller gain. Equations 2.7 and 2.8 describe the integral controller time and Laplace domains, respectively [8], [9]:

$$u_c(t) = k_i \int^{t} e(\tau) d\tau \tag{2.7}$$

$$U_c(s) = \left[\frac{k_i}{s}\right] E(s) \tag{2.8}$$

where k_i represents the integral controller gain.

Derivative action

Derivative control is often used in motion control. It is very sensitive to measurement noise, it makes trial-and-error tuning more difficult, and it is not absolutely required for process control. However, using a controller's derivative mode can make certain types of control loops - temperature control, for example - respond faster than with PI control alone.

The derivative control mode produces an output based on the rate of change of the error. It produces more control action if the error changes at a faster rate. If there is no change on the error, the derivative action is null. This mode has an adjustable setting called Derivative Time (Td). The larger the derivative time setting, the higher will be the derivative action produced. If the derivative time is set too long, oscillations will occur and the control loop will be unstable. A Td setting of zero effectively turns off the derivative mode. Two units of measure are used for the derivative setting of a controller: minutes and seconds.

If a controller can use the rate of change of an error signal as an input, then this introduces an element of prediction into the control action. Derivative control uses the rate of change of an error signal. The following mathematical expressions describe the time and Laplace domains of the derivative controller respectively [8], [9]:

$$u_c(t) = k_d \frac{de}{dt} \tag{2.9}$$

$$U_c(s) = [k_d s] E(s) \tag{2.10}$$

where the derivative control gain is expressed as k_d.

Proportional and integral control mode

The PI controller is a variation of the PID controller and it is mainly used to eliminate the steady-state error resulting from the P controller. However, in terms of the speed of the response and overall stability of the system, it has a negative impact. This controller is mostly used in areas where the speed of the system is not an issue. Since a PI controller has no ability to predict the future errors of the system, it cannot decrease the rise time and eliminate the oscillations [10].

With the combination of both controllers, after a disturbance, the integral mode continues to increment the controller's output until it has eliminated all offset and brought the equipment outlet signal back to its setpoint. The time and Laplace domains formulas for PI controllers are, accordingly, given as [8]:

$$u_c(t) = k_p e(t) + k_i \int^t e(\tau) d\tau \tag{2.11}$$

$$U_c(s) = \left[k_p + \left(\frac{k_i}{s}\right)\right] E(s) \tag{2.12}$$

where k_p and k_i are variables that correspond to the proportional controller gain and integral controller gain, respectively.

Proportional and derivative control mode

The objective when using a PD controller is to increase the stability of the system by improving control since it has an ability to predict the future error of the system response. In order to avoid the effects of the sudden change on the value of the error signal, the derivative is taken from the output response of the system variable instead of the error signal. Therefore, the derivative mode is designed to be proportional to the change of the output variable to prevent the sudden changes occurring on the control output resulting from sudden changes on the error signal [10].

A property of derivative control that should be noted arises when the controller input error signal becomes constant but not necessarily zero, as might occur in steady-state process conditions. Under these circumstances, the derivative of the constant error signal is zero and the derivative controller produces no control signal. Consequently, the controller is taking no action and is unable to correct for steady-state offsets. To avoid the controller settling into a dormant state, the derivative control term is always used in combination with a proportional term. The expressions for time and Laplace domains of PD controllers are, respectively [8]:

$$u_c(t) = k_p e(t) + k_d \frac{de}{dt} \tag{2.13}$$

$$U_c(s) = [k_p + k_d s] E(s) \tag{2.14}$$

where k_p and k_d are variables that correspond to the proportional controller gain and derivative controller gain, respectively.

2.3.3 Mobile robot path tracking

The project reported in [11] presents a simple and effective solution for the path tracking problem of a mobile robot using a PID controller. The proposed method uses a simple linearized model of a mobile robot composed of an integrator and a delay. The synthesis procedure is simple and allows the PID controller to be tuned considering the nominal performance and the robustness as control specifications.

In that work, a classical PID approach is proposed for the path-tracking controller, which represents the control strategy most frequently used in the industry. A very simple model of the mobile robot kinematics was used and thus a robust PID tuning was necessary. That paper is organized in four stages. In stage one, the model considered for the mobile robot kinematics is presented. The description and robustness of the proposed tuning for the PID controller are discussed in stage two. The way in which the PID controller is applied to the path-tracking problem is shown in stage three. Finally, stage four expresses the conclusions of this work based on experimental results.

Beginning with the initial stage, which represents the path tracking problem to be analyzed using the Nomad 200 mobile robot, this robot has a synchro-drive type locomotion system that consists of three drive wheels whose turning speed and orientations alter simultaneously. The robot provides a position estimation system based on odometry. This technique has an accumulative error which implies the need to update the estimation from environmental data provided from sensor systems with a predetermined frequency. The next stage expresses the robust stability of the closed loop to be analyzed considering an additive unstructured description of the uncertainties. In stage three, it was proposed a PID controller for the control structure. Finally, the last stage reached the conclusion that the main advantage of the proposed methodology is that it used a very simple model for the mobile robot which allowed the tuning of a simple PID controller. The new method for the robust tuning of the PID controller was based on classical concepts and could be applied to the integrative process plus a delay. The proposed PID controller was tested using a synchro-drive mobile robot, in which some experimental results showed a better performance obtained in spite of delay estimation uncertainties.

2.3.4 Temperature control of water bath system

In the work reported in [12] the control of temperature in a water bath system used a PID controller. This project involved two stages, and the first stage was to design the transfer function for the water bath system. The second stage was the implementation of the PID controller for the water bath system.

That project was designed with the possibility of the user defining data. In other words, the user sets up the required temperature limit in the PID controller as the controller set point. A heating element is then used for heating the water in the water bath, and the resistance temperature detector (RTD) sensor placed in the water bath was responsible for measuring the water temperature. Based on the controller constant values and the temperature of the water, the temperature is controlled by the water's flow rate to the bath. The Ziegler Nichols tuning method was used to tune the PID controller.

The water bath system consisted of various hardware/software, such as a water tank, sensors, a data acquisition (DAQ) system, a controller and a heater. As mentioned before, the RTD is used as a temperature sensor, and the DAQ was used for the connection between the sensor and controller, as well as the controller and driver circuit.

The experimental procedure was described by the following process: when the temperature was measured by the RTD, it was converted into the voltage, which then became the input for the controller through DAQ. The difference between the setpoint and actual value was applied to the controller.

As it was referenced before, the temperature process and the control strategy adopted was the PID controller. The temperature controller was identified based on the simulation output curve of PI, PD and PID responses respectively. With this rise time, settling time, delay time and maximum peak overshoot values were obtained. From the experimental results, it was concluded that the PID controller has less settling time, delay time, rise time and maximum peak overshoot when compared to the other controllers using the transfer function model. This controller controls the water's temperature while maintaining it, as it compares with different controllers.

2.3.5 Autonomous car tracking in urban traffic

In paper [13] an online self-tuned PID controller was proposed to control a car with the objective of following each other, at distances and speeds typical of urban traffic. The tuning mechanism implemented in that work was the MIT rule, due to its ease of implementation. However, this method does not guarantee the stability of the system, providing good results only for constant or slowly varying reference signals and in the absence of noise, which are unrealistic conditions. When the reference input varies at an appreciable rate or in the presence of noise, eventually it could result in system's instability. In that paper, an alternative method was proposed that significantly improved the robustness of the system for varying inputs or in the presence of noise.

The control system used sensors, which provided the actual speed of the controlled car and the relative position of the car ahead. This allowed calculating the actions on the steering wheel, the accelerator, and the brake. To control the direction, a proportional controller with a constant gain was used, while for controlling the accelerator-brake a PID self-tuning controller was employed.

To tune the parameters of the PID controller of the acceleration, two different algorithms were used: the MIT rule, and the proposed rule of that work.

This alternative method tries to avoid the continuous increasing of the parameters, using new rules that allow their decreasing. Two new signals are calculated, representing a filtered version of the error (a simple first order low-pass filter was used in that work), and its derivative.

In this way, the three parameters can increase and decrease along the time, so this method will be, at least initially, more robust than the MIT rule when referring to the system's stability.

This method was successfully applied to control the car which tries to follow another vehicle ahead. The robustness of the method against noisy measurement signals was validated by means of several simulations. In that work, a conclusion was reached: the proposed method showed better performance than the MIT rule above.

2.3.6 Vehicle control strategies analysis

PID control theories and fuzzy logic control were the fundamental concepts of the work reported in [14]. In that work, it was also established two degrees of freedom vehicle dynamic models.

The controller of vehicle stability was designed by using the method of direct yaw moment control and the different control strategies. By comparing and analyzing the control effect of PID control and fuzzy logic control, the results show as follows: slip angle and yaw rate combined control was better than slip angle and yaw rate controlled individually by comparing control effect; fuzzy logic control has a better robustness and PID control is simple and practical by analyzing control theories. Different control methods can be used in the same control systems according to the needs for practical application. The results can improve and enhance passengers car maneuverability and stability control.

A yaw rate sensor was used to transmit the difference between the vehicle yaw rate and the nominal yaw rate to the controller. When input variables changed, the yaw moment will be adjusted by the controller. Afterward, brake force was distributed to every wheel. Increasing PID control algorithm was used.

Medium vehicles were selected as a simulation test subject. The vehicle speed was about 70 km/h and the steering wheel angle of 70 deg in the angle step input was set.

The conclusions were drawn from the steering wheel angle step input simulation. There were no control theories or control strategies that could run all the conditions of the vehicle and achieve a good effect. Although, for general operation conditions, it was necessary to study a more effective theoretical method and control strategy. For example, when the vehicle is sliding, it is necessary to impose yaw moment control. It can control the size of the slip angle and yaw rate effectively. In this situation, lateral acceleration will not exceed the limit of later surface attachment. If the conditions above are met, the control improves. The slip angle is relatively larger when the vehicle is in the middle and high velocity, and the vehicle shows mainly dynamic characteristics. The main purpose of control is for stability, so yaw rate control can achieve a better control preference. With the increase of the angle, the result will become worse when to control separately yaw rate. In the high adhesion coefficient road, combined control can control the slip angle and yaw rate efficiently and satisfy the dynamic characteristics.

2.4 Hardware-in-the-loop simulation

Hardware-in-the-loop simulation is most often used in the development and testing of embedded systems, when those systems cannot be tested easily, thoroughly, and repeatedly in their operational environments. It requires the development of real-time simulation, covering some or all parts of the real system. This simulation technique enables to associate both virtual and real elements, thus allowing to approximate the experience to a more realistic scenario [15].

The remainder of this section will explain in detail the origin of this technique, its evolution with time, its structure and some related works.

2.4.1 Historical development

The first approach to hardware-in-the-loop simulation was probably realized for (real-time) *flight simulation*, where the early objectives were to simulate the instruments with a secure and stable cockpit, the "Link Trainer", also known as, "Blue Box" and "Pilot Trainer" (1936) (Figure 2.6), [16], [17], [18], and later on in addition to move a cockpit according to aircraft motions, e.g. for the training of pilots. In these simulations, the cockpit and the pilot were real, and the motions were generated by electrical and hydraulic actuators. The first generations used analog tube controllers and analog motion simulations, which were subsequently replaced by computers processors [16], [17], [19].

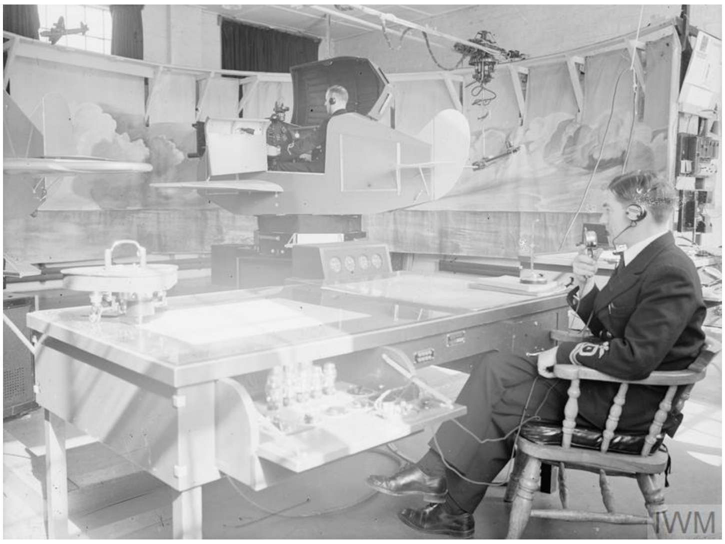


Figure 2.6: Link trainer in use at a British Fleet Air Arm station (Adapted from [20])

HIL simulation proved to be very resourceful when testing missile guidance systems [21], [22], [23], [24], [25]. The Sidewinder program used HIL simulation back in 1972 [21], [25]. Soon after, in parallel with missile simulations, NASA was working on the development of highly maneuverable aircraft technology (HiMAT) [21], [26]. The purpose of the HiMAT program was to investigate the use of advanced concepts such as reduced static stability and fly-by-wire. NASA developed a range of high-fidelity HIL simulations to enable this research. More recently, HIL simulation has gained popularity in the automotive industry.

HIL motion simulators were built for the dynamic testing of *vehicle components* (e.g., suspensions, bodies) with hydraulic or electrical actuators (testing machines). In this case, the excitation of wheels by a road surface was simulated. Alternatively, a different type of HIL motion simulation is the *vehicle driving simulator* (Figure 2.7) [16], [27], [28]. *Dynamic motor teststands*, where the engines are real and the vehicles and gears are simulated by different hardware (an electrical DC or AC motor) together with a digital computer processor, is a special kind of HIL simulator [16], [29], [30].

With the development of electronic technology in the 1990s, Fennel *et al.* [31] developed a real-time simulator using transputers, which are smaller than VAX stations and Michales [32] developed the real-time simulation system implementing a 17 degree-of-freedom vehicle model. Sailer and Essers [33] developed a real-time simulator using transputers with three-dimensional and non-linear vehicle model [34].

With the remarkable growth of the personal computer, it became possible to create a real-time simulator through the use of a low-cost personal computer. Bach [35] developed the real-time simulation system, 'Roadrunner', which has the ability to compute three-dimensional vehicle behavior. Although this system consisted of a personal computer, it was complicated and costly by using additional microprocessors and extension cards [34].

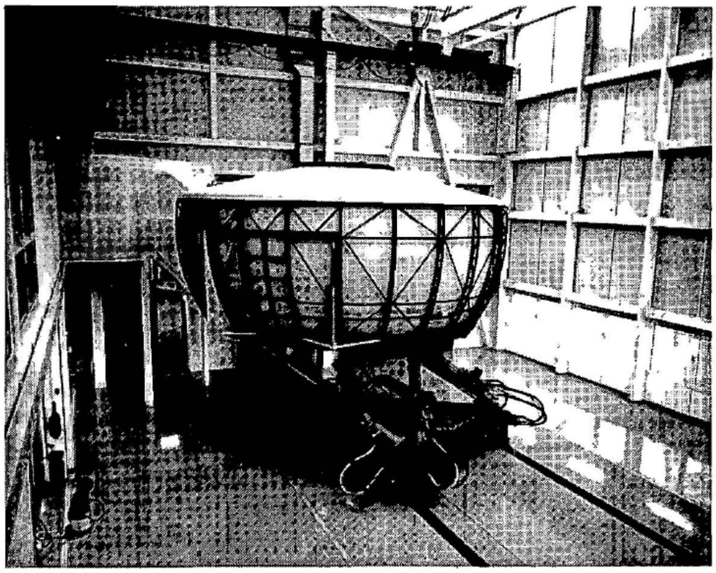


Figure 2.7: Daimler-Benz driving simulator (Adapted from [36])

With the advancement of digital electronic control systems for vehicles, such as, the ABS (anti-lock breaking systems), and TCS (traction control system) for drive chains and automatic gears, the HIL simulators followed through various stages of development [16], [37], [38]. First versions used high-performance workstations and computers processors. However, the amount of real time simulation was very limited [16], [39], [40]. The availability of *parallel computers*, in the form of transputers (RISC processors with on-chip RAM and high-speed communication links) and digital signal processors (DSP: microprocessors with efficient operation of additions and multiplications by parallel processing with different memories for the program and data). This opened the way for real-time simulations of complete hydraulic systems, sensors, actuators and suspension systems [16], [31], [41]. Further research then showed how the mechanical systems can be simulated in real-time through parallel computers. It is possible to find these systems implemented in *multibody systems* [16], [42], [43], [44], *brake systems* [16], [45], [46] and *combustion engines* [16], [47], [48], [49], [50], [51].

2.4.2 Architecture

HIL simulation is characterized by the operation of real components in connection with real-time simulated components [16]. It is considered the safest and least cost-effective technique for components in virtual environments. Most of the actual components are replaced by mathematical models and elements are to be tested and inserted in a closed loop. One of the reasons to include components in a simulation is often due to the inexistence of knowledge about their characteristics, or attributes are very complex, or then there is the need to test the actual components themselves, as is the case with testing controllers [52].

Generally, the control-system and real system relate to the hardware and software, respectively, and are used for series production. The controlled process, consisting of actuators, physical processes, and sensors, can then be either fully or partially simulated [16]. However, there are some paths in the mesh that are not possible. In other words, it is impossible for an actual process to be monitored through simulated sensors or simulated actuators in real processes [52]. Frequently, some actuators are real, and the process, as well as the sensors, are simulated. The reason is that actuators and the control hardware often form one integrated subsystem. Furthermore, actuators are difficult to model accurately and to simulate in real-time. The use of real sensors with a simulated process may require a considerable effort because no physical sensor input exists and it must be generated artificially. In order to change or redesign some functions of the control hardware or software, a bypass unit can be connected to the basic control hardware. Hence, hardware-in-the-loop simulators may also consist of a partially simulated (emulated) control functions [16].

It is possible to find different interactions of real components with simulated versions of different elements in a typical control system in Figure 2.8. Table 2.2 shows some distinct combinations between integrated real and simulated process components for a HIL simulation.

The third configuration expresses a real working system, since no simulation is involved, whereas the fourth combination is a fully simulated system. Such a configuration is called *Full Simulation*. When a real component is in the simulation loop, it is called *Partial Simulation*.

Note that during the development cycle it is possible to shift from configuration four to three replacing real actuators, real sensors and eventually real processes. However, these transitions are very difficult to implement since each step implies a new physical interface, to be implemented between the simulation and the real actuator/sensor [53].

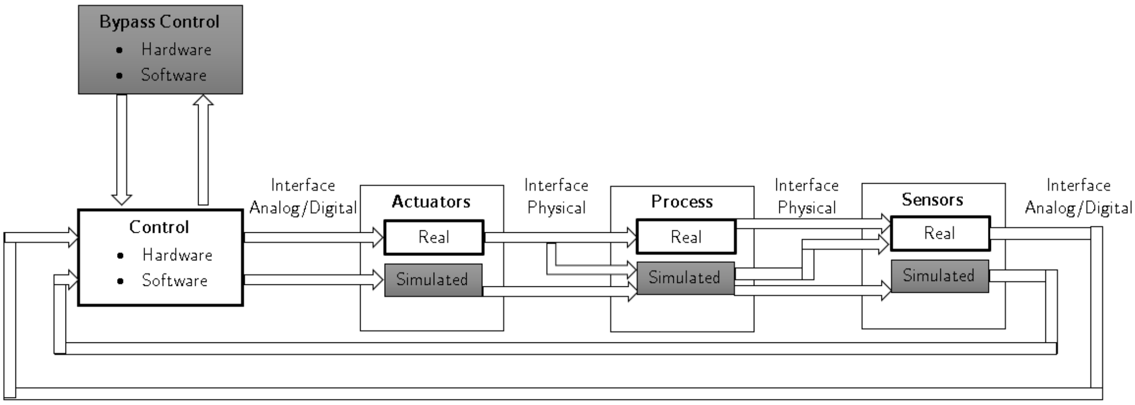


Figure 2.8: Hardware-in-the-loop simulation possibilities in control systems (Adapted from [16], [52], [53])

Table 2.2: Different cases for hardware-in-the-loop simulation of hybrid real and simulated process components.

Case	**Actuators**		**Process**		**Sensors**		
	Real	Simulated	Real	Simulated	Real	Simulated	
1	x	-	-	x	-	x	Partial Simulation
2	x	-	-	x	x	-	Partial Simulation
3	x	-	x	-	x	-	Real System
4	-	x	-	x	-	x	Full Simulation
5	-	x	-	x	x	-	Partial Simulation

With the growing prevalence of HIL simulation, various advantages have made this technique fundamental to the industrial development, such as [54]:

- It requires significantly less hardware than physical prototyping, thereby costing less.
- It often demand less hardware than fully physical prototypes; HIL simulators can also be considerably quicker to build. Controller prototypes, for instance, can be built rapidly and evaluated in the loop utilizing the appropriate software platform.
- The prototyping in hardware with components whose dynamics or other attributes (e.g., transient emissions formation in diesel engines) are not fully understood, HIL simulators often achieve fidelity levels unattainable through purely virtual simulation.
- When simulating complex physical systems runs faster than purely virtual simulations of the same type of system (e.g. IC engine simulations based on Computational Fluid Dynamics).
- Systems that operate in highly variable environments (e.g., off-road vehicle suspension systems) can often be tested in controlled lab settings through HIL simulation, which may significantly increase repeatability.
- It enables the simulation of destructive events without risk. Sometimes it is necessary to test the behavior of a system; in extreme cases, for example the destruction of the system, with HIL simulation it is possible to do so in a safe environment and without components expenses.
- It can often be used to train human operators of safety-critical systems in safer environments (e.g., flight simulators).
- It allows different teams to develop distinct parts of a system in hardware without losing sight of integration issues, thereby enabling concurrent systems engineering.
- The embodiment of tests for controlling hardware and software under extreme environmental conditions in the laboratory (e.g., high/low temperatures,

high accelerations and mechanical shocks, aggressive media, electromagnetic compatibility).

In summary, HIL simulation includes a setup that prototypes parts of a given system in hardware. Additionally, this virtually emulates the rest while maintaining *bidirectional* information flow between these physical and virtual subsystems. The traditional application of a HIL simulation concept is controller design and testing. For experiments, one of the essential conditions is the ability to simulate mathematical models of physical components in real-time.

2.4.3 Modeling hydraulic systems for simulation

In the work [52] a methodology was proposed with the aim of modeling libraries and setting semi-empirical hybrid models for hydraulic components. The objective was to test the hybrid performance models in platforms for hardware-in-the-loop simulation.

In order to carry out these experiments, the experimental platform required minimum instrumentation and safety requirements. To build the experimental platform it was necessary to implement a hydraulically actuated manipulator with a set of sensors, real-time hardware and a package of software tools for data acquisition and control. The sensors used for experimentation were pressure sensors, and the manipulator's actuator was fully hydraulic. It was an actuator of a single pallet. Some of the actuators were controlled by a solenoid operated spool valves with the nominal flow. For data acquisition and control, the hardware utilized was a National Instruments card (PCI 1200) and a dSPACE card (DS1102). The software was developed for the experimentation of different control algorithms and to acquire the necessary data for the parameterization of hydraulic components models. The manipulator control software was developed in Simulink, and later on, translated into C language to be installed in the DS1102 card through dedicated dSPACE software.

In order to develop a real-time monitoring and operation software of the manipulator a graphical interface was created, using the dSPACE ControlDesk platform. One of the examples to run real-time operation was the variation of the controller parameters or the manipulator operation's manual.

Different software was used for monitoring some types of equipments, such as the pressure transducers, so an application was developed in the platform LabView with the possibility to synchronize with the control software, through digital channels of the data acquisition card and from the control card.

2.4.4 Test platform for planetary rovers

In a dissertation for the University of Waterloo [55], a test platform for planetary rovers was designed, fabricated and tested. The ability of planetary rover designers and mission planners to estimate the rovers performance through software simulation is crucial. HIL testing can take the benefits of software simulations by allowing designers to incorporate hardware components within traditionally pure software simulations. This provides more accurate performance results without having access to all hardware components, as would be required for a full prototype testing.

The test platform was designed with complete modularity such that different types of tests could be performed for distinct types of planetary rovers and in different environments. To demonstrate the concept, a test bench consisting of powertrain components loosely based on NASA's Mars Exploration Rovers was developed. The system consisted of solar panels, a solar charge controller, a battery, a AC/DC converter, a DC motor, and a flywheel. In addition, a lighting system was designed to simulate the solar radiation conditions solar panels would experience throughout a typical day. On the software side, a library of component models was developed within MapleSim and model parameters were tuned to match the hardware on the test bench. A Labview program was developed for real-time simulations within allowing communication between hardware components and software models. This program consists of all the component models, hardware controls and data acquisition. The graphical user interface (GUI) of this program enables the user to select which component is to be tested and which component is to be simulated. It is also possible to change the model's parameters as well as to see real-time sensor measurements for each component. A signal scaling technique based on non-dimensionalization is also presented, which can be used in a HIL application to obtain scaling factors to ensure dynamic similarity between two systems.

A demonstration of power estimation was performed using the pure software model simulations as well as pure hardware testing. Hardware components were then added into the software simulation progressively which resulted in a better accuracy with the addition of hardware. The rover's power flow was also estimated under different load conditions and seasonal variation. These simulations demonstrated the effectiveness of a HIL platform for testing the rover's hardware performance.

2.4.5 High speed servo system

To demonstrate the possibilities of implementing HIL simulation, the work in [56] reports experiments with servo systems. The Y500 solder jet printer was the equipment used and it has high accuracy and speed requirements, resulting in a short control loop, which puts hard constraints on the model calculation time. The simulation is implemented on a dSPACE MicroAutoBox, using the MATLAB/Simulink and SimMechanics toolboxes. It communicates with the control system via SPI and implements an existing SimMechanics model and a newly developed Simulink model.

The Simulink model developed was found to be a sufficiently realistic simplification of the SimMechanics model. Furthermore, the SimMechanics model cannot be executed in a closed loop on the chosen hardware due to its extensive complexity. This work also shows that when using the internal clock, even the shortest model step time allowed by the hardware is insufficient to perform a stable HIL simulation. Lastly, the SPI implementation tested introduced computational delays into the control loop of the MY500, which led the system to be able to calculate only one axis.

2.4.6 A platform to test and improve the control of heating systems using Modelica

To test the concept of HIL simulation in an area other than mechanics, paper [57] demonstrates the control of water heating devices for space heating (SH) and domestic hot water (DHW) generation. Among these technologies, this article focuses on double service air-to-water-heat-pumps (AWHP). The control of AWHP strongly impacts their performances, so that manufacturers of AWHP need to optimize the control of their systems to increase their performances. That paper presents the implementation of a HIL real-time simulation test bench for AWHP. The new test bench proposed used real weather conditions and simulated building's and occupant's response thanks to a virtual model. This test bench allowed R&D departments and manufacturers to test several heating systems in parallel as well as different control options. It thus has the potential to reduce the cost and time required for AWHP development-to-market process.

As mentioned before a real AWHP is tested under real conditions. Thus, the thermal load was calculated through the connection between the heat pump and the virtual building via HIL real-time simulation. The building and the hydraulics network were modeled using the Modelica-based library BuildSysPro. The domestic hot water generation is tested due to the connection of the heat pump to a real DHW tank with a real tapping scenario. The test bench was composed of several elements, namely a heating system (AWHP) to be tested (both the indoor unit and the outdoor unit), and a DHW tank, equipped with an immersed coil connected to the AWHP via a three-way valve. A hot tap water bench is connected to this DHW tank in order to emulate the DHW tapping in the dwelling. Data from the tapping scenario were synchronized with a real-time simulator. Connected to the heat pump, there was a cooling water bench, whose purpose was to keep the AWHP inlet water temperature at the setpoint calculated by a simulated system. In order to monitor the heat pump behavior, there was an acquisition board, actuators, and HP communication interface, equipped with temperature, solar radiation, and mass flow rate sensors. In order to take into account the real-time simulated indoor air temperature, an indoor air thermostat was emulated. The analysis of the impact of the heat pumps control on the system dynamics and on the indoor thermal comfort enabled to highlight control issues related to the heating curve, the DHW management and the defrost process of AWHP. Optimized control parameters throughout time were proposed in view of reaching lower energy consumption and better comfort performances.

The building and the hydraulics heating circuit were simulated in a computer using the Modelica environment. Finally, in order to evaluate different control parameters settings as a function of time, there was a control management interface.

2.5 Test benches

As already mentioned, one of the objectives of this work was to develop a test bench that could perform several experiments, including HIL simulation. In addition, it is intended to develop an innovative concept that allows different forms of testing and performing experiments. It is therefore important to support this study to know what software/hardware an experimental platform would require. The following are some examples of projects carried out that are related to test benches, but focused mainly on thermal systems.

2.5.1 Development of test benches project

The work reported in [58] was developed in the context of a curricular internship, where it was intended to carry out a project of a total renovation of the systems testing at Vulcano Thermotechnology, currently Bosch TT.

The main objective of this assignment was to develop an integrated system of acquisition and data analysis to test the operation and technical features of water heaters.

This development included the following aspects, starting with the specification of new test stands, through the choice of transducers to be applied, defining the physical structure of the test bench, as well as defining the architecture of the data acquisition system and the development of the software responsible for the interface with the operator and also the data analysis/storage.

The developed test bench enabled a robust system for data acquisition with FieldPoint solution from National Instruments. The implemented module runs an application created in Labview real-time software for interface and monitoring. In addition, extra temperature and pressure transducers were implemented, in comparison with the previous benches. One more advantage was the possibility of carrying out the tests without the need to have the computer on, using independent displays.

With this work, recent test benches became safer and with more robust and modern software, with a great capacity of signal acquisition, to generate files with information and all of this with a user-friendly interface, increasing the tests efficiency and decreasing their costs.

2.5.2 Study on technical features of domestic gas water heaters

A study of the action with technical features for water heaters from different brands was made in the dissertation [59]. The fundamental goal was to evaluate the thermal behavior of domestic gas burning water heaters. To do so, a physical model was developed to describe this behavior and the relative importance of radiative and convective heat transfer could then be analyzed.

The model was zero-dimensional, where the furnace was a stirred reactor. The combustion gases were considered grey and the external surface of the gas/water heater exchanger was also considered grey and at a constant temperature.

The parameters measured throughout the test were: the inlet and outlet water temperatures, the combustion gas temperature at the heat exchanger outlet, propane flow rate, water flow rate and the combustion gas composition at the heat exchanger outlet. The equipment used for the measurement of these elements was: infrared and paramagnetic meters, type T thermocouples and flow and differential pressure transducers.

Regaring the interface and monitoring of the tests, the program was developed in Basic. This was responsible for the interpretation of the analog and digital signals received from the data acquisition card, as well as the generation of the files with the measurements data.

With that work, new test benches started to have greater security and robust software, with a great ability to acquire signals. Additionally, they could generate files with information. All of this data could be controlled by the user through a user-friendly interface which would benefit the efficiency of the experimental tests.

2.5.3 Control design for a thermal hardware-in-the-loop test bench for automobile thermal management systems

It is important to notice in paper [60] the techniques, types of equipment and software utilized for the development of a HIL test bench. It was designed and built from scratch as a means of real-time testing for new hardware components dedicated to thermal management of electrified vehicles in a realistic environment.

The task of the test bench was to simulate the thermal load of the tested device which, afterwards, was defined by the volume flow rate and the inlet temperature of the fluid. For that purpose, the test bench features a flow heater, a fluid pump and a process cooling that was to be actuated by an appropriate control algorithm in order to control the desired thermal load. The experimental platform was connected to a real-time framework on which the control algorithm was implemented. In figure 2.9 it is possible to view the schematic diagram, consisting of a circuit with several thermal-hydraulic components that were connected by pipes.

The control structure consists of a cascade model-based predictive controller with underlying conventional PID-control.

The equipment installed in the test bench was: temperature sensors to measure the fluid temperature downstream and upstream of each component, a flow meter, and pressure sensors.

In order to process the information obtained from the data acquisition board, the software used in that paper was Matlab. With the help of measurement data from Matlab it was possible to optimize the acquired parameters.

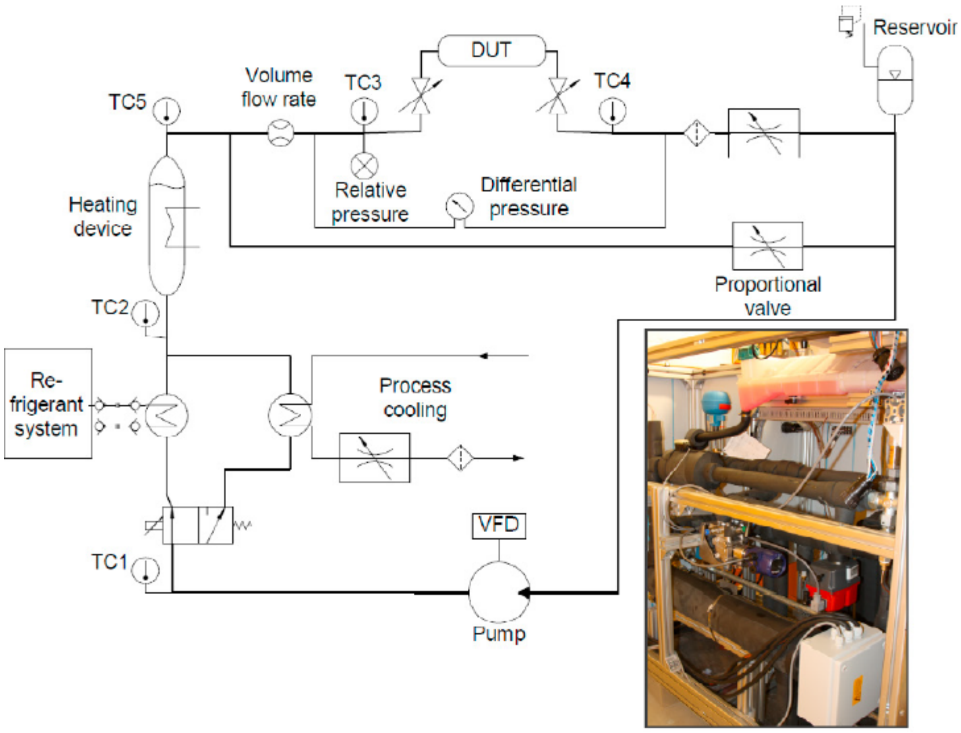


Figure 2.9: Representation and picture of the thermal-hydraulic test bench (Adapted from [60])

2.5.4 Development of a platform to test the equipment control strategies for domestic water heaters

In dissertation [61] the techniques and methods to test controllers and perform HIL simulation were very similar. The main objective was to develop and implement a methodology for testing of control strategies in domestic water heating systems. In this work, a study was carried out for the different possibilities of hardware and software that allow to achieve HIL simulation, and identify the key limitations of existing test benches. It was also proposed a test bench concept that allowed to apply the HIL simulation technique, and turn it flexible to the point of allowing other forms of testing and simulation that may be useful in the process of developing and testing controllers.

Temperature sensors, electrical resistances, a National Instruments card (PCI1200), a DS1102 board and microcontrollers (Arduino boards) were some of the types of equipment implemented in the bench for measurement and control. The software used to monitor, data logging and develop dynamic models were, ControlDesk, Arduino IDE, MATLAB/Simulink, among others.

Two case studies were carried out. The first one was to demonstrate the test bench features. That case study developed was simple and related to water heating. This case study was executed because of the deficit of HIL applications in thermal systems. The second was to apply a Bosch TT water heater, with the objective of reporting the essential steps to apply the techniques used in the experiment of that dissertation.

Chapter 3

Layout proposed for the test bench

Based on the research done in Section 2.5 it was possible to draw some conclusions and think of an alternative method, in order to address some of the limitations of the existing test benches.

Most of the test benches have yet to test the installation of the system, with the respective sensors for instrumentation and have the software of acquisition and interface, in order to visualize the state of the variable. Basically, these enable to test physical systems and to monitor them.

With the research made it is possible to conclude that there are three types of test benches, which are: (i) those that allow testing the actual model and its controller and monitor the behavior; (ii) those that allow testing the actual model using a simulated controller and perform the monitoring; (iii) and those that allow to manually create some instabilities to see how they respond to the prototypes.

In order to overcome some of the limitations, this chapter will present not only a concept of the virtual test bench but will also be the starting point to its development. It also explains some of the functionalities and main features of the test bench by describing the different operating modes.

3.1 Concept of the experimental platform

As mentioned before, one of the fundamental objectives of this virtual test bench is to be innovative and perform HIL simulations. Additionally, the proposed test platform allows for the incorporation of a tankless gas water heater (TGWH) hardware within a software simulation. The platform is completely modular such that it can be used for a variety of testing. For example, one can use HIL testing to estimate the flow rate or the power applied to the heat cell before all the hardware components are available, by using the associated software models to substitute the components which are not yet available. Lastly, component testing can be performed on this platform.

To design such a testing platform, a library of mathematical models of TGWH components is required and a physical test bench with the hardware components must be designed. Users should be able to select which components are to be tested from the library and which are to be simulated. Sensors are required to evaluate hardware performance, and a method must be established to ensure real-time simulation. Also, a graphical user interface is needed to allow users to modify model parameters, select test hardware and view real-time response of each component.

Based on all the limitations detected, and with the objective of implementing a concept of a virtual test bench that is distinct from the actual ones, Figure 3.1 shows the proposed layout of the virtual test bench. This includes the real system, in this case a TGWH, by the real controller, typically a μC, and finally by the virtual simulation in real-time. The monitoring and the data acquisition systems are integrated into the computer, which contains the software for modeling and simulating the systems and also the acquisition and control hardware.

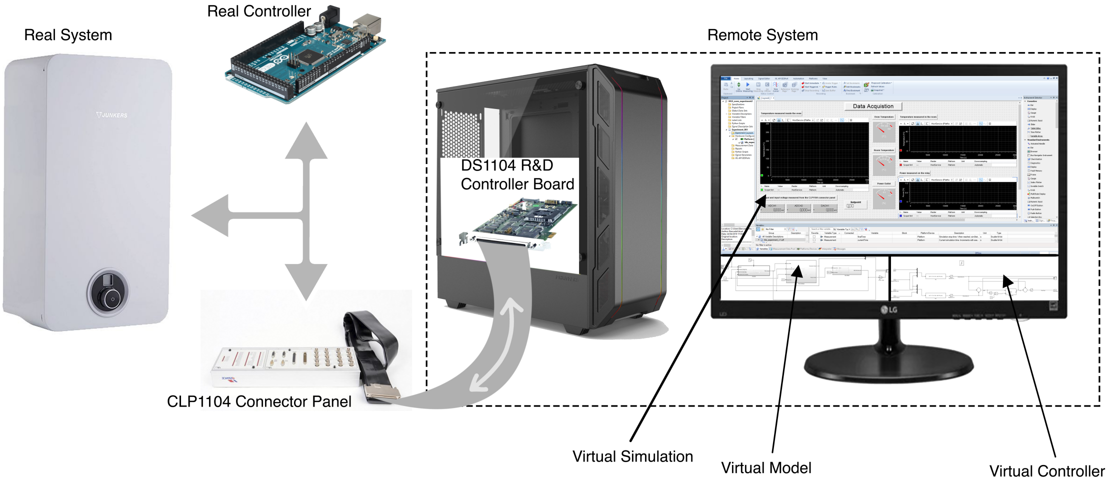


Figure 3.1: Layout proposed for the virtual test bench.

With the proposed concept, a 3D illustrative model of the virtual test bench was made, resorting to the Solidworks software. In Figure 3.2 it is possible to see the placement of each equipment by comparing the 3D model and real experimental platform.

The remainder of this chapter describes the different operating modes that can be applied to the experimental platform.

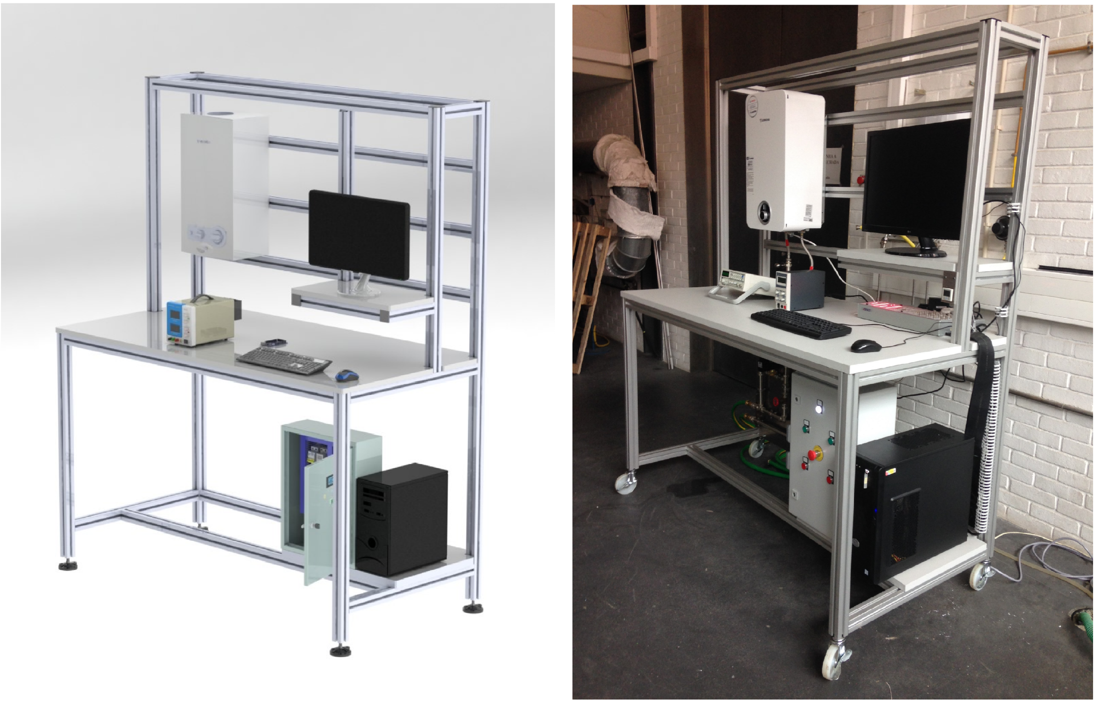


Figure 3.2: Comparison between the 3D model and the real virtual test bench.

3.2 Description of the operating modes

To evaluate the concept and its features it must be tested through different experiments and controllers. Usually, the design of a controller starts with the modeling of either in a real or simulated environment. Afterwards, the controller is implemented in a control device and finally, it is tested with physical prototypes of the system.

Switching from a virtual controller to a μC is very beneficial because the behavior is recognized immediately in the modeled system. On the contrary, in the μC only after programming it, this still needs to be tested on a real system in order to evaluate its action. There is a considered loss in time and resources in this process that could be avoided if one could predict the behavior of the controller and its optimization in the real system even before programming the μC. Therefore, simulating virtually the models in MATLAB/Simulink plays a fundamental role to considerably reduce the simulation time.

3.2.1 Data acquisition

The first option of the test bench enables the ability to acquire and record data from the equipment with a data acquisition software and a very simple model. The equipment and the virtual model don't have any type of controller, as this mode is only intended for real-time data acquisition. However, one of the benefits of this mode is to discover unknown parameters, that is, variables of the dynamic model that need to be determined. By executing different experiments it is possible to achieve data that can, afterward, be processed or analyzed and, finally, achieve the desired parameters. Figure 3.3 shows the simple version of the discussed mode.

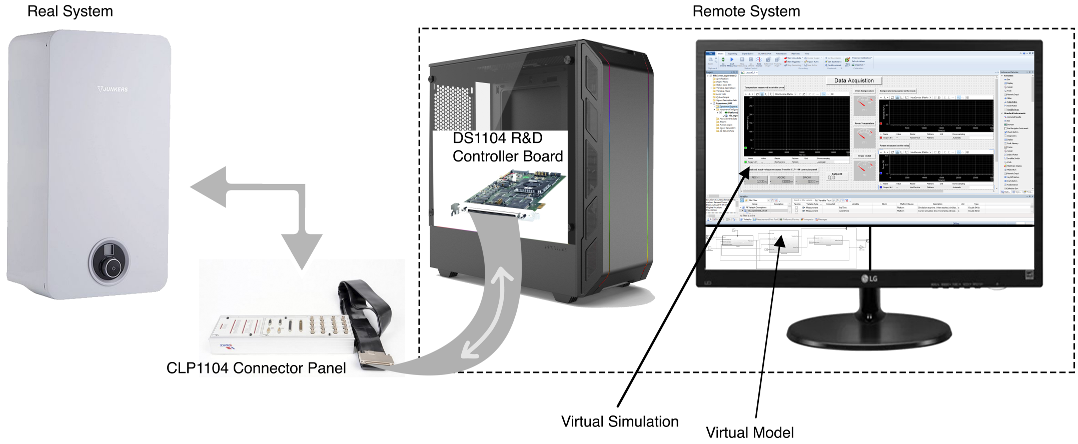


Figure 3.3: Data acquisition mode of the virtual test bench.

3.2.2 Real-time virtual simulation

The second operating mode is the capability of performing a virtual simulation in real-time without any hardware or prototype integrated. The dynamic model of the thermal system is controlled by the virtual controller while the simulation is executed in real-time. At an early stage of development, this mode proves to be useful since it is possible to test the controllers and tune their parameters. In Figure 3.4 it is possible to see the schematic module of this configuration that has the specific designation of full simulation.

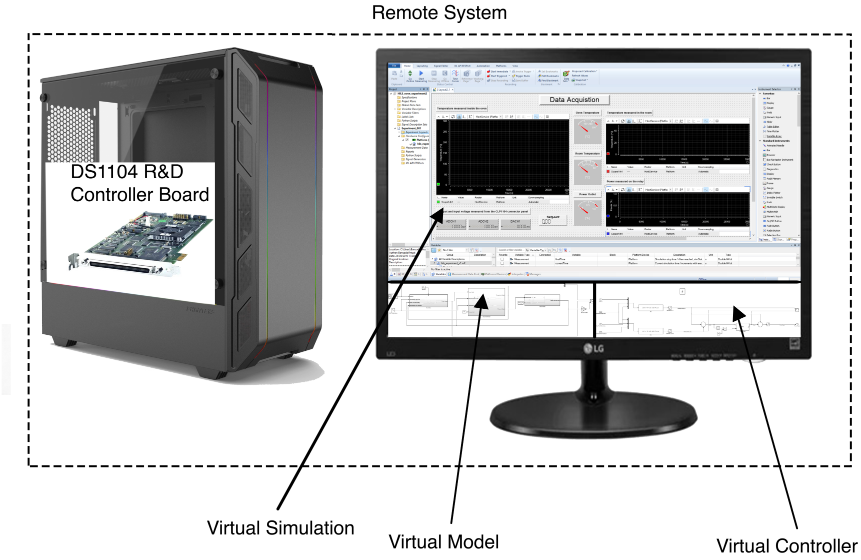


Figure 3.4: Real-time virtual simulation mode of the virtual test bench.

3.2.3 Hardware-in-the-loop simulation

The third option the test bench can perform is HIL simulations, that can be either a real system to be controlled by a virtual controller, or a μC controlling the virtual model. Both modes will be explained in the next two sections.

Virtual controller

In this situation the only real component is the equipment. The real model is then controlled by the virtual controller (Figure 3.5).

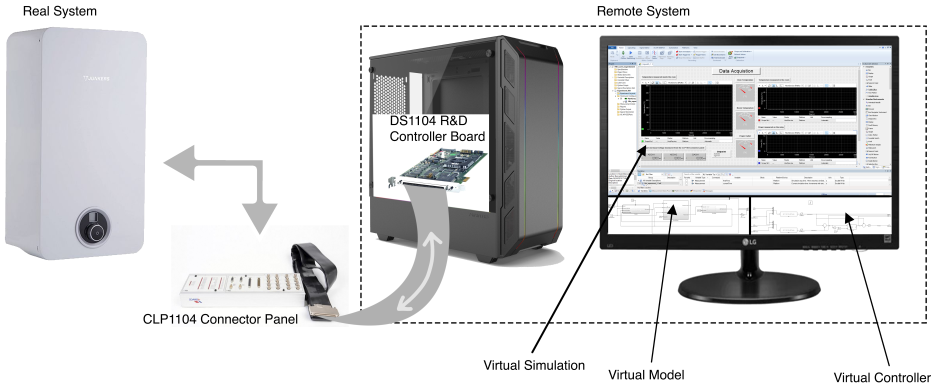


Figure 3.5: Hardware-in-the-loop simulation with a virtual controller mode of the virtual test bench.

Real controller

In this version of the HIL simulation the real element is the μC which controls the virtual model as if it were a real system (Figure 3.6).

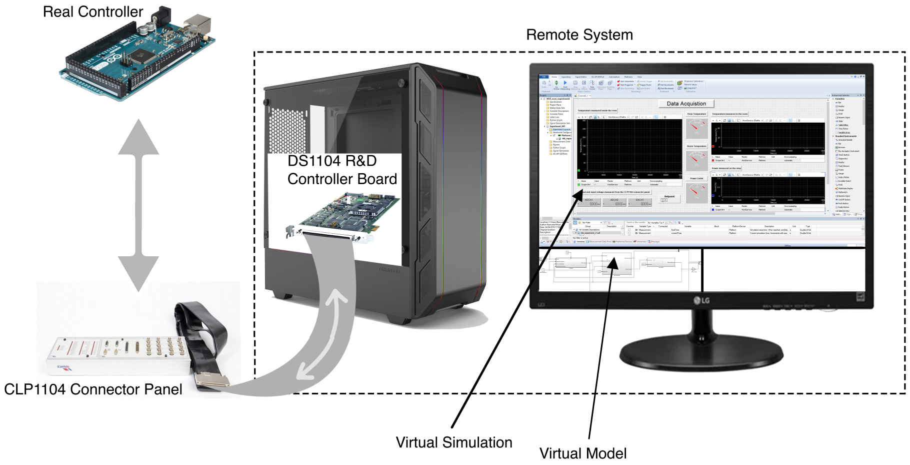


Figure 3.6: Hardware-in-the-loop simulation with a real controller mode of the virtual test bench.

3.2.4 Real system controlled by the microcontroller

Finally, it is possible to test the actual controller in the real system. This feature is beneficial in the final phase of the testing where all components are real. This is a common practice for the most of the existing test benches in the different areas of engineering. Figure 3.7 shows a simplified schematic version of the layout.

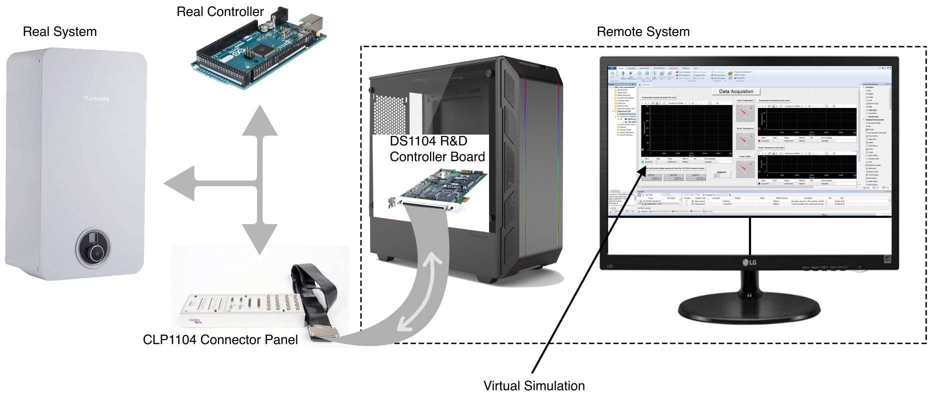


Figure 3.7: Real system controlled by the microcontroller mode of the virtual test bench.

Now that the different operating modes have been explained, the next chapter will explain the procedure to build the virtual text bench as well as the hardware and software used to perform the various experiences.

Chapter 4

Hardware and software implemented in the virtual test bench

This chapter introduces the required hardware and software in order to exploit every feature of the virtual test bench.

Through the MATLAB/Simulink software, it was possible to model the dynamic models of the thermal systems in a virtual environment. In order to be able to communicate between the control and acquisition board and the dynamic models, an extension of the Simulink, RTI, was used. This has a library with all the necessary blocks to communicate between the different systems. Using a second extension of Simulink, RTW, it is possible to generate and compile C and C++ code from the dynamic models. With the generated file it can be downloaded to the acquisition and control board, which then allows to perform data acquisition, hardware control and also a virtual simulation in real-time. An Arduino is used as a μC to be able to control both virtual and real systems, where the programming is done through a specific software of this μC. Finally, a graphical interface is used to monitor and acquire data in real-time.

These tools were used because, in addition to being cost-effective, they allow to perform every type of experiment necessary for the virtual bench of this work.

The following definitions are introduced to measure the transparency of the experimental platform.

4.1 RTW

This software is an extension of Simulink and enables MATLAB to automatically generate, package and compile source code from Simulink models to create real-time software applications on different systems. It generates and executes C and C++ code from Simulink models, Stateflow charts and MATLAB functions. The generated source code can be used for real-time and nonreal-time applications, including simulation acceleration, rapid prototyping, and hardware-in-the-loop testing. It is possible to tune and monitor the generated code using Simulink or run and interact with the code outside MATLAB and Simulink. Real-Time Workshop (RTW) is the foundation for producing code generation capabilities. Additionally, this software supports various features, such as [62]: (i) Automatic code generation tailored for a variety of target platforms; (ii) A simple graphical user interface; (iii) A rapid and direct path from system design to implementation; (iv) An open architecture and extensible make process; (v) Seamless integration with MATLAB and Simulink.

4.2 RTI

This program handles any kind of continuous-time, discrete-time, and multirate system. Depending on the I/O hardware, different channels of the same I/O board can be used with different sample rates. Real-Time Interface (RTI) supports asynchronous events and enables them to set task priorities and overrun strategies for executing the interrupt-driven subsystems. It also supports time-triggered tasks and timetables, which allow for implementing tasks and groups of tasks with variable or predefined delay times in relation to an associated trigger event. This makes task handling in the model very flexible. RTI also possesses hardware configuration and provides automatic consistency checks to avoid parameterization errors.

When working with RTI to connect the model to a dSPACE I/O board, there is I/O module available from the RTI block library that can be simply selected and dragged to the model, which afterward enable to connect it to the Simulink blocks. All settings, such as parameterization, are available by clicking the appropriate blocks. RTW generates the model code while RTI provides blocks that implement the I/O capabilities of dSPACE systems in the Simulink models, thus preparing the model for the real-time application. The real-time model is then compiled, downloaded, and started automatically on real-time hardware, without the need to write a single line of code. RTI provides consistency checks, so potential errors can be identified and corrected before or during the build process [63].

Figure 4.1 shows some blocks from the RTI library in a Simulink environment.

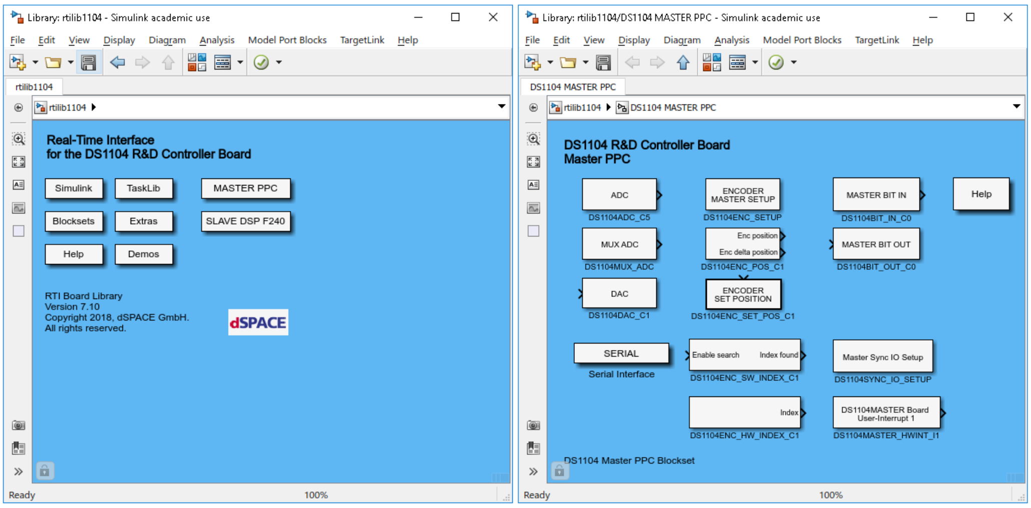


Figure 4.1: Examples of RTI library blocks.

4.3 dSPACE software

The dSPACE software is frequently used in systems for research and development applications in industry and universities where fast time-to-market and robust results are key requirements. Typical applications of dSPACE are rapid control prototyping, automatic production code generation, and HIL simulation. dSPACE Simulator enables engine control units (ECU) tests in real-time and in a realistic test environment that does not need real components, through HIL simulation. Testing with this technique is a viable alternative because it allows new ECU's and software to be tested largely in a virtual environment, without real vehicles or prototypes.

The modularity involved in dSPACE's simulators allows it to run simulators in a network or separately as single simulators. dSPACE has already shown the capability to test a whole electronic car system, even if it is a network of 40 or more ECU's. In an ECU network test with dSPACE Simulator, ECU's that are not yet available can be simulated and replaced by their real counterparts later on.

The modeling of dynamic systems can be developed in MATLAB/Simulink while the data acquisition and control and generation of code are under the responsibility of the dSPACE software. It provides a set of data acquisition cards guided through real-time simulation of systems and for controllers, ensuring the execution of control algorithms in real-time. Real-Time Interface (RTI) links MATLAB / Simulink and the hardware. As mentioned before, RTW has the ability to generate very simple code for simulation and without great work for the user. While RTW generates the model code, RTI provides blocks that implement the I/O capabilities of dSPACE systems in Simulink models, that prepare later on for the real-time application. The model in time real-time hardware is compiled, loaded, and started automatically on real-time hardware without the need to write intermediate code [64].

4.4 DS1104 R&D Controller board

The controller board that will be used in further experiments will be the DS1104 model. This single-board system with real-time hardware based on PowerPC technology and its set of I/O interfaces make the controller board an ideal solution for developing controllers in various fields, such as drives, robotics, and aerospace.

This board is a cost-effective entry-level system with I/O interfaces and a real-time processor on a single board that can be plugged directly into a PC. The board can be installed in virtually any PC with a free PCI (Peripheral Component Interconnect) or PCIe slot. The I/O resources of the DS1104 board are split between the two processors on the board, the Master PPC (Power PC) and the Slave DSP F240.

The Real-Time Interface is implemented in this board and can be accessed through the RTI1104 Board Library in Simulink [64].

Figure 4.2 shows the internal structure of the DS1104 board as well as the connection to the connector panel.

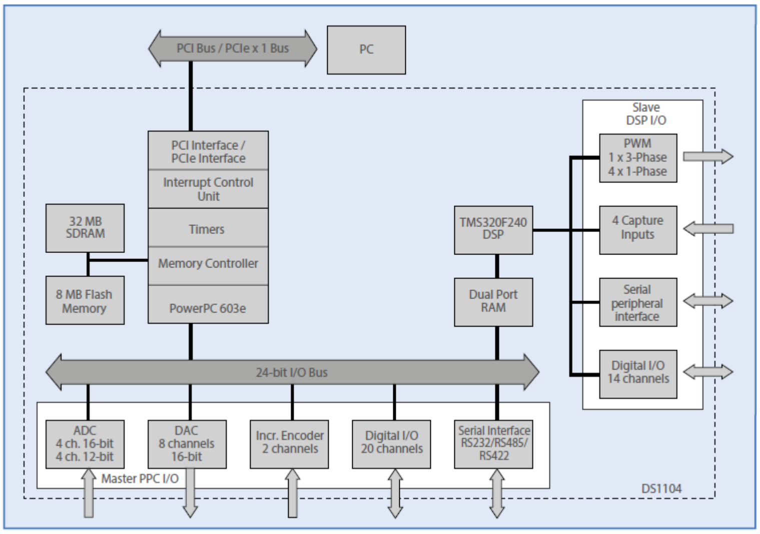


Figure 4.2: Block diagram of the DS1104 board linked to the connector panel (Adapted from [64]).

4.5 Connector panel CLP1104

The equipment provides easy-to-use connections between the DS1104 R&D Controller Board and devices to be connected to it. Devices can be individually connected, disconnected or interchanged without soldering via BNC connectors and Sub-D connectors. This simplifies system's construction, testing and troubleshooting. In addition, this panel provides an array of LEDs indicating the states of the digital signals [64].

Fig 4.3 shows the equipment described previously.

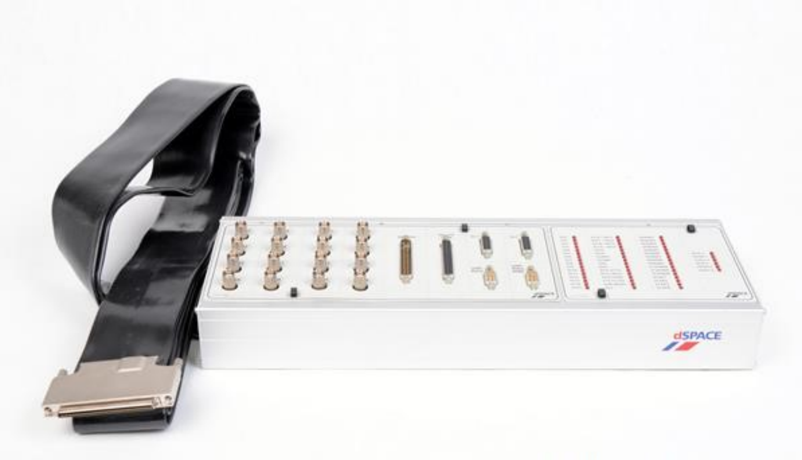


Figure 4.3: CLP1104 Connector and LED Panel (Adapted from [64]).

The specifications of this device are exhibited in Appendix A.

4.6 ControlDesk

This program is the dSPACE experiment software for seamless ECU development. It performs all the necessary tasks and gives the user a single working environment, from the start of experimentation right to the end. It can perform various tasks such as Rapid Control Prototyping (RCP; fullpass, bypass), Access to vehicle bus systems (CAN, CAN FD, LIN, Ethernet), ECU measurement, calibration, and diagnostics, Virtual validation with VEOS and SCALEXIO and hardware-in-the-loop simulation.

This software unites functionalities that often require several specialized tools. It provides access to simulation platforms as well as to connected bus systems, and can perform the measurement, calibration, and diagnostics on ECUs, e.g., via standardized ASAM interfaces. Its flexible modular structure provides high scalability to meet the requirements of specific application cases. This gives the user clear advantages in terms of handling, the amount of training needed, the required computing power, and costs [64].

In Figure 4.4 it is possible to see examples of hardware and software used to connect to some of the application areas when working with Controldesk software.

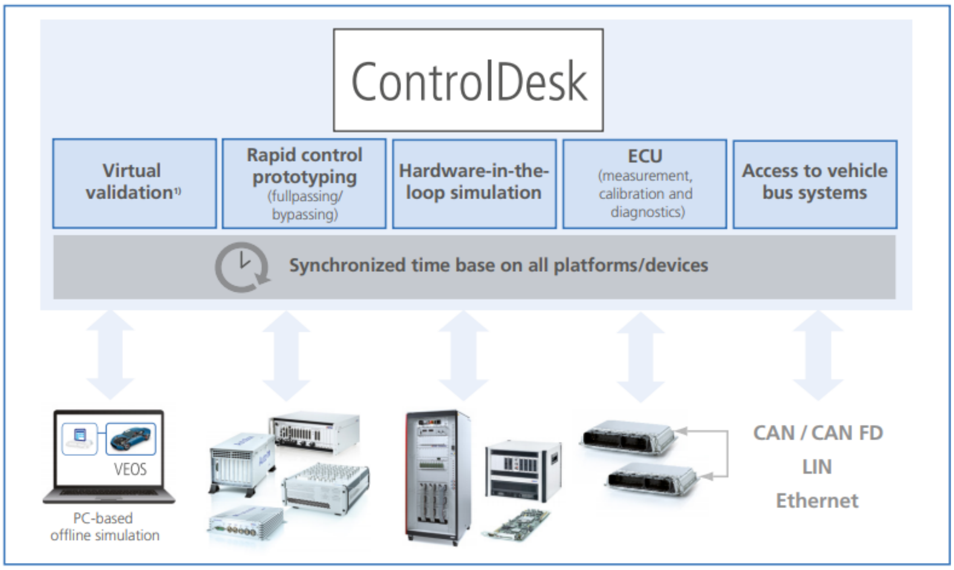

ControlDesk
Virtual validation[1]
Rapid control prototyping (fullpassing/ bypassing)
Hardware-in-the-loop simulation
ECU (measurement, calibration and diagnostics)
Access to vehicle bus systems
Synchronized time base on all platforms/devices
VEOS
PC-based offline simulation
CAN / CAN FD
LIN
Ethernet

Figure 4.4: Connection of software and hardware to some of the application areas of Controldesk (Adapted from [64]).

4.7 MATLAB/Simulink

MATLAB/Simulink is a well-known solution for technical computing and is the most referenced software for modeling dynamic systems. Developed by MathWorks, MATLAB enables large numerical calculation, which together with Simulink enables the development and simulation of dynamic models with complex systems. These can be designed on a high level by modeling the interactions between functional blocks. The MATLAB engine is used to evaluate the equations of the Simulink blocks. Simulink also provides a graphical editor, customizable block libraries, and solvers for modeling and simulating dynamic systems [2].

4.8 Arduino IDE

This software is an open-source physical computing platform based on a simple I/O board and a development environment that implements the Processing/Wiring language. It can be used to develop stand-alone interactive objects or can be connected to software on the computer (e.g. Flash, Processing and MaxMSP). With Arduino Integrated Development Environment (IDE) it is possible to edit, compile and upload Arduino sketches to the Arduino boards. This software supports different operating systems such as Windows, macOS, and Linux, and is accommodated to C and C++ languages [65].

Chapter 5

Case study 1

In this present chapter, a case study is presented to assess the proposed layout and its functionalities, and also to test the control strategies on the virtual bench in five distinct operating modes. It also presents the first approach to validate the concept of the experimental platform. First, a general description, an experimental oven setup, the oven's dynamics, and its thermal components are introduced. Second, an assessment of the virtual models is made in MATLAB/Simulink environment in order to simulate the dynamic models and to parameterize the fundamental parameters. Third, the instruments are calibrated, what is indispensable for the operation of this concept applied to the test bench. There are a series of signals, that will be interconnected and require the necessary adaptation to avoid any hardware damage. Fourth, the experimental data are used to validate the numerical results and to be processed with the objective of determining the ideal parameters. Fifth, a PI controller is implemented to test the behavior of the dynamic model. Sixth, the tests were carried out and, as already mentioned, these are: the virtual system is controlled by the virtual controller, the real system is controlled by the real controller, the real controller controls the virtual system, and the real controller controls the real system. Last, an analysis of the experimental data is carried out.

5.1 General description

The oven concerned is very simple and small, but it allows performing every type of simulation including HIL. This case study not only is related to air heating but can also relate to a thermal system.

The experimental platform is composed of the oven which has two orifices, one to place the electrical heating resistor and the other is dedicated to place the temperature sensor. The electrical heating resistor is controlled by a relay module with a fixed duty cycle.

After the selected equipment, in order to elaborate the modeling and control plan, an operating system was established. Thus, the initial objective is to define the temperature at a steady-state, transfer the power released by the resistor to heat the air in order to maintain the indoor temperature close to that value, and subsequently cut the power to the electrical heater resistor until the chamber reaches room temperature. Four tests were carried out with different duty cycles generated by the PWM signal.

The next group of experiments was dedicated to test controllers and to exploit the features of the virtual test bench.

5.1.1 Experimental setup

The oven opens the possibility of comparing simultaneously both the impact of control parameters settings on the duty cycle and the indoor air temperature. This is possible thanks to the use of the same virtual building model to simulate the power applied to the systems.

In order to complete this case study, as already mentioned, the oven is equipped with sensors and actuators required to control the indoor air temperature and the electrical heating resistor. This last equipment converts electrical energy into heat to raise the temperature inside the oven. The heating element inside the electric heater is an electrical resistor, and its function is to dissipate heat into the oven.

To control the real system with a μC, an Arduino Uno was used. According to the operating methods mentioned in Section 3.2, the Arduino plays a fundamental role in controlling both the real and the virtual systems.

The virtual systems are developed and modeled in MATLAB/Simulink software, the graphical user interface dedicated to monitoring is created in ControlDesk from the dSPACE software, the signals generated by the measuring equipment are acquired and processed through the CLP1104 connector panel, and finally, the simulation in real-time can be executed with the DS1104 R&D Controller board.

The systems mentioned before need to be connected, and since there are many types of signals, signal conditioning modules need to be developed and some types of equipment require calibration. These will be discussed later on.

Figure 5.1 shows the real and virtual systems with the oven actuation and sensor systems installed.

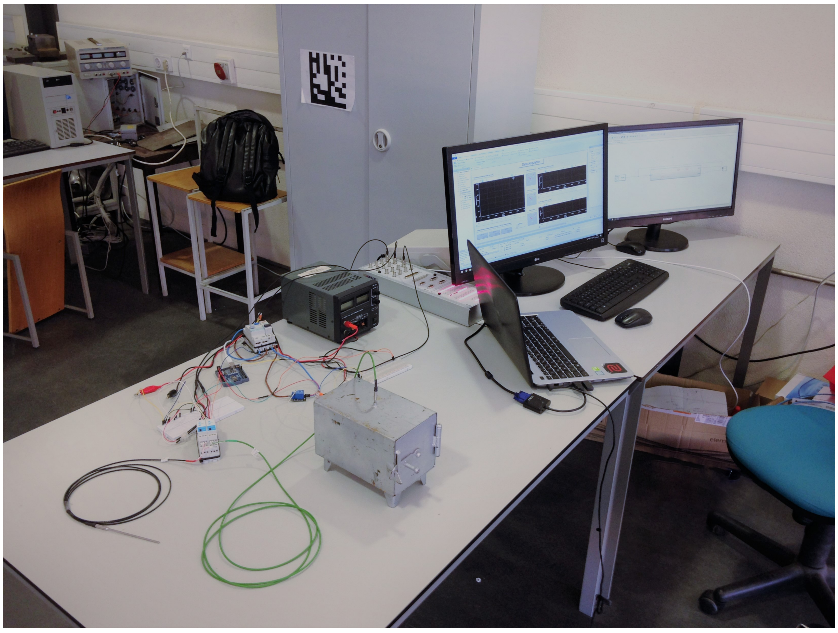


Figure 5.1: Experimental setup of the coupled system.

The remainder of this chapter describes the process and methodology adopted to model the virtual and real systems and also the controllers implemented in these systems.

5.2 Modeling the oven's dynamics

In this Section it is introduced the methodology used to develop the dynamic models in order to describe the dynamic behavior for the thermal component relevant to study the equipments behaviour.

Heat transfer by convection and conduction between the indoor air and the environment air is expressed by means of an overall coefficient. Energy balance is described globally in a module, using average temperature and power generated. Therefore, the heating resistor generates heat and allows the temperature to raise inside the oven.

5.2.1 Methodology adopted for the thermal component

A heat transfer model was built to describe the heating generated by the electrical resistor. Since the main objective of this case study is to validate the concept of the virtual test bench, the dynamic model of the oven will not include heat transfer by convection and radiation. The temperature of the air inside the oven is modeled, and the physical parameters are taken into account by correlations or physical laws.

The model is based on a linear lumped capacitance structure. Although this modeling method greatly simplifies the heat transfer differential equations, it provides a good enough approach to estimate the main temperature of the system. Besides, if model identifiability is demonstrated, which means that there only exists a global optimum parameter set, it can even be used to estimate the main heat fluxes and energies and some extra parameter-based analyses can be made. The connection between lumped capacitance models and heat transfer phenomena will be explained later on.

To implement a lumped capacitance structure, a system is modeled as a set of thermal capacitors interconnected with each other by means of thermal resistances. Additionally, external temperatures or heat fluxes may exist as boundary conditions. Considering a general system whose only external temperature is the environment temperature, T_{env} of the model and has an equation of the form

$$C\frac{dT}{dt} = \dot{Q}_g - UA(T - T_{env}) \tag{5.1}$$

where C is the thermal capacitance, $\dot{Q}$ is the boundary heat flux that directly flows into the capacitance C, T is the temperature, T_{env} is the environment temperature and UA is the loss coefficient of the oven.

Since the element inside the chamber of the oven is air itself and there is no air movement, and therefore no velocity, it is permitted to assume that there is no mass flow rate entering or leaving the oven, and Equation 5.1 can be reorganized in order to isolate the temperature variation with time, as

$$\frac{dT}{dt} = \frac{1}{C}(\dot{Q}_g - UA(T - T_{env})) \tag{5.2}$$

The next step is to model and implement this equation in a Simulink environment in order to simulate it and further test controllers.

5.2.2 Methodology for the implementation of the model in Matlab/Simulink

To build the thermal models presented previously in Matlab/Simulink, the following aspects were taken into account:

- the controller block inputs to be considered are the steady-state setpoint/temperature and the temperature feedback from the oven model. The output will be solely the heating power.
- the oven block will have a single input and output, where the output power of the controller will be admitted by the input of the oven model, and the output of this one will be the internal air temperature.

5.3 Definition of the thermal model's parameters

Some of the parameters of the lumped model could be identified, such as power heating ($\dot{Q}_g$), room temperature (T_{env}) and internal temperature (T). Both the loss coefficient and the thermal capacitance cannot be determined but instead set. The loss coefficient can be determined as the product between the overall heat transfer coefficient and the surface area through which heat transfer takes place since there is no possible method to measure the surface area of the oven, so the approach to determine this parameter needs to be different. The thermal capacitance is simply the product of the mass of an object and its specific heat, and since it was not possible to obtain the mass of air in the oven this parameter becomes unrecognized to the system.

The first step was to acquire data from experimental results. This was possible to achieve with the first feature of the virtual test bench. The procedure, as well as the models used, will be explained in detail in Section 5.9. The second step consisted in applying two distinct methods of processing/optimizing the experimental results. The last step was to simulate the experiment with the parameters determined with each method.

5.3.1 Analytical method

Based on the data gathered from the experiment, this method initially consisted of extracting the information where there were some temperature oscillations due to the error of the measuring equipment, and thus the average temperature in a steady-state was calculated. Thus, the overall heat transfer coefficient parameter was calculated with the average of the steady-state temperature, the power heating and the room temperature, considering Equation 5.2 with null dT/dt,

$$UA = \frac{\dot{Q}_g}{T_{ss} - T_{env}} \tag{5.3}$$

Afterwards, the time interval where it was linear was extracted; in other words, this instant is when part of the data obtained has a constant slope. The thermal capacitance was determined with the following expression, based on Equation 5.2,

$$C = \frac{t_2 - t_1}{T_2 - T_1}(\dot{Q}_g - UA(T_2 - T_{env})) \tag{5.4}$$

where t_2 and t_1 are the extracted time instances, and T_2 and T_1 the extracted temperature instances that is equivalent to the range of time extracted.

Several experiences were carried out, which means several parameters were obtained. To virtually simulate the thermal system it is incorrect to use different values of experiments, so the simulation was executed based on the average of each parameter. Table 5.1 shows the discussed parameters.

Table 5.1: Parameters determined by the analytical method.

Duty cycle [%]	**UA [W/ °C]**	**C [J/°C]**	T_{env} **[°C]**
25	1.12	2261.76	21.20
50	1.22	2460.14	19.50
75	1.29	2559.38	18.70
100	1.23	2552.75	20.83
Average	1.23	2552.75	20.06

This method proved to be very useful in obtaining the desired parameters, although it is not reliable as there was still a significant difference in the values obtained by the experiments, and nothing guarantees that the temperature range extracted is actually linear over time. For this, a second method was used, and it will be explained next. It is important to mention that the processing time was about 30 s.

The simulation results according to this method and the experimental results are presented in Figure 5.2.

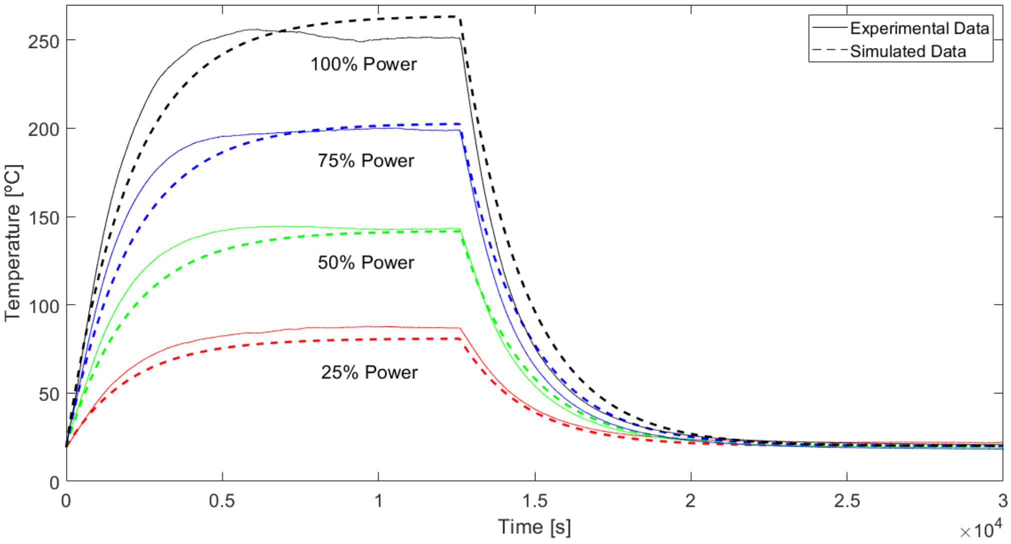


Figure 5.2: Comparison between the experimental results and the several simulations through the analytic method.

5.3.2 Optimized method

The second method is based on the MATLAB function 'fminsearch', which finds the minimum of a scalar function of several variables, starting from an initial estimate. Since the main objective of these methods is to find the ideal parameters, the values previously obtained can be used as the initial estimates for the 'fminsearch' function. These will then be used in the cost function 'J'. After that, virtual simulations will be carried out with the parameters in order to obtain and save data. Based on the result of the mean square error expression, a value is returned and it is used for the next iteration until the optimum parameters are successfully obtained.

After processing and calculating the data from the cost function, it was possible to obtain the values of parameters UA and C as 1.24 W/°C and 2068.22 J/°C, respectively. The processing time took about 655 s. With the increase of time, it further becomes a disadvantage in determining the desired parameters, since one of the features of this virtual test bench is to perform experiences in the lowest time.

Figure 5.3 shows the Simulink model developed with the purpose of simulating the desired parameters, each time the MATLAB script is executed.

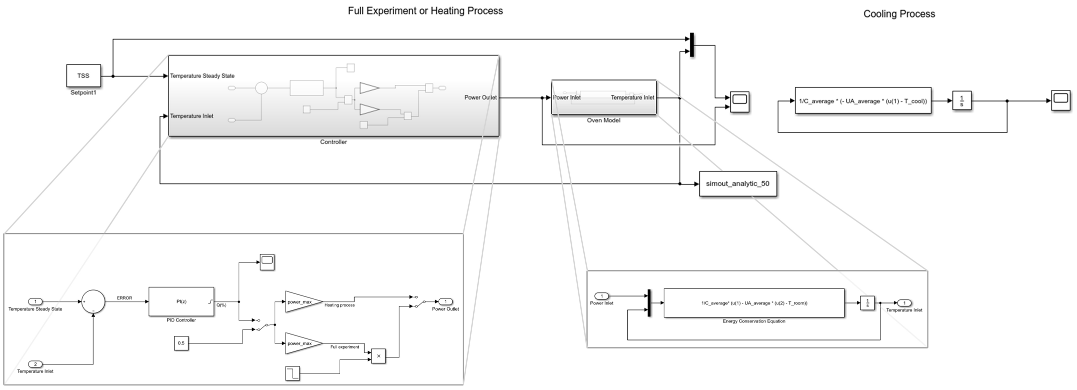


Figure 5.3: Simulibk block diagram of the data acquisition model.

Figure 5.4 shows the correlation between the parameters calculated with each iteration and the value returned from the cost function.

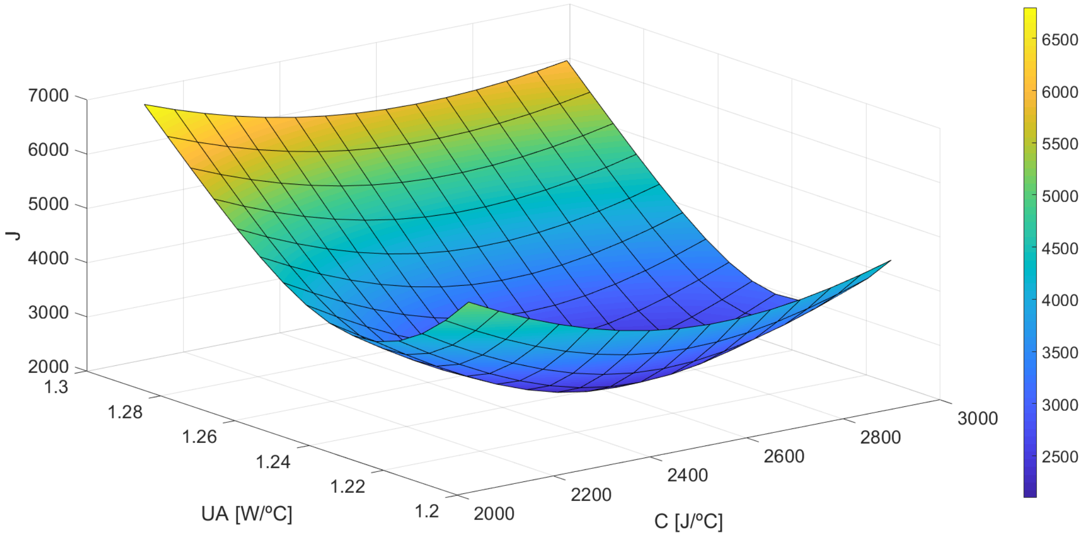


Figure 5.4: A plot of cost function J as a function of thermal capacitance C and loss coefficient UA.

Finally, it is possible to visualize a comparison between the experimental results and the simulated results by using the optimized method in Figure 5.5.

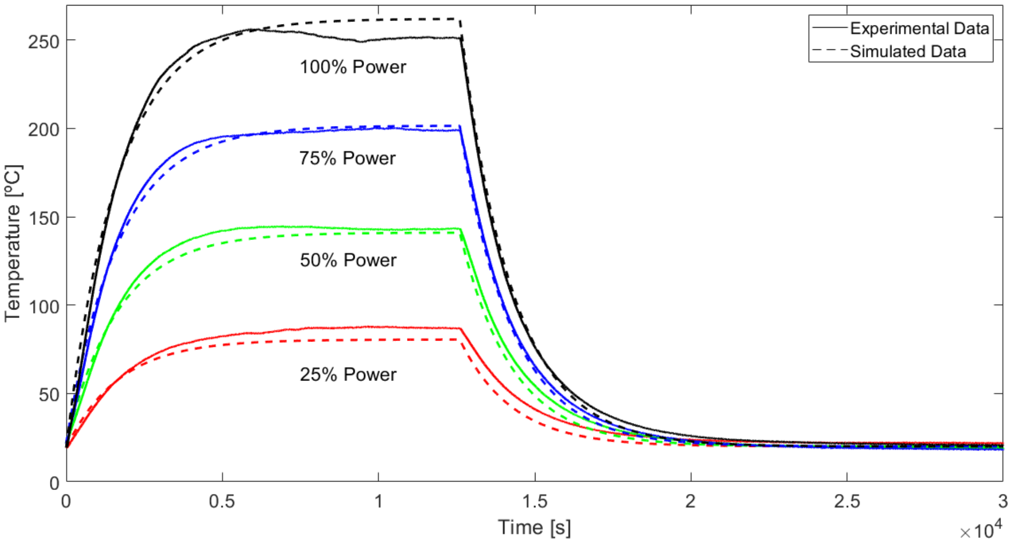


Figure 5.5: Results of the simulations carried out using the optimized method.

5.4 Control strategy

Now that the models have been implemented in a MATLAB/Simulink environment, it is time to evaluate the control of the thermal system. A simple PI controller was chosen without much detail concern at this stage, as its efficiency is only improved after testing it in a real system.

In most controllers, there is a reference or a setpoint, that exits the thermal system. Through, for example, a PI controller, this calculates the error between the output and the reference, and the controller responds in a way to decrease this error, which is referred to as *feedback control*.

It should be mentioned that the control is being carried out based on the inlet temperature delivered to the system.

The methodology proposed to design the controller starts by considering Equation 5.2, where the room temperature is constant in order to isolate the fundamental parameters. The result of this operation will form a simplified version of the main equation.

$$\frac{dT}{dt} = \frac{\dot{Q}_g}{C} - \frac{UA}{C}(T - T_{env}) \tag{5.5}$$

Next is to apply the Laplace transform to obtain the transfer function based on Equation 5.5 and isolate the unknown parameters, which will result in the following expression

$$\frac{T(s)}{\dot{Q}(s)} = \frac{1/C}{s + UA/C} \tag{5.6}$$

The transfer function for the controller is based on Equation 5.7, for it is a PI control mode

$$PI(s) = Kp\frac{s + Ki/Kp}{s} \tag{5.7}$$

It is import to mention that the 's' in the denominator is a pole located at the origin. This has the purpose of eliminating the steady-state temperature error.

Now it is possible to advance to the next step, which consists of developing a MATLAB script and establish the K_p and K_i terms. Afterwards, a transfer function is created for the controller and the plant model. These functions will be multiplied, and with the MATLAB function 'pzmap' it is possible to locate the poles and zeros of the system. This has two poles and one zero, which are located at s = 0, s = -(UA/C) and s = -(K_i/K_p), respectively. The system is then simulated in the Simulink model 5.6 in order to verify the response of the PI parameters.

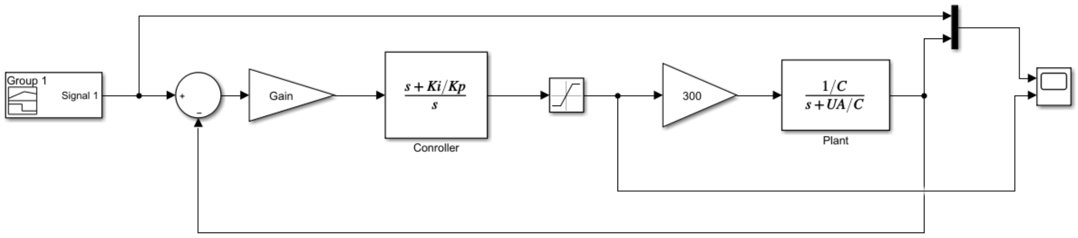


Figure 5.6: Simulink model with the transfer functions.

To adjust the parameters of the controller, random values for the K_i/K_p ratio must be established, then the MATLAB script is executed and with the results originated from the 'pzmap' function a gain value is selected from the imaginary axes. This is related to the velocity of the system's response, the longer this distance the faster the temperature will reach the setpoint. The last step will be to run a virtual simulation with the parameters. This process is repeated until the desired K_p and K_i parameters are reached.

The first set of values is presented in Figure 5.7, which are $K_p = 1$ and $K_i = 0$.

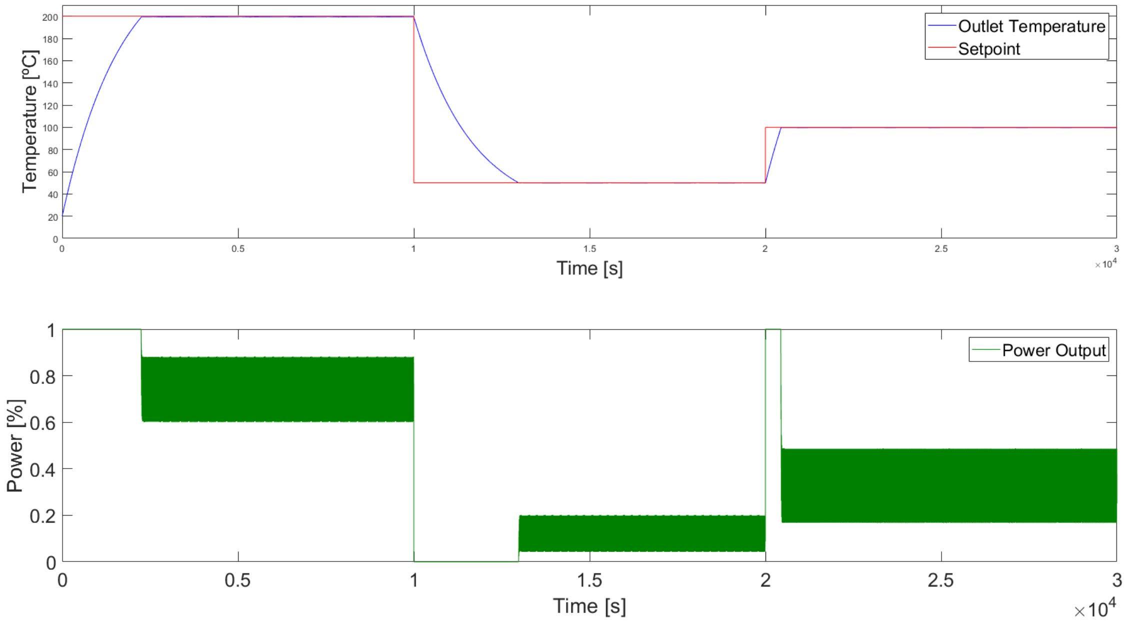


Figure 5.7: PI controller response to a proportional term.

The controller's response is fast, although there is some oscillations at the steady-state region, so the integral term will be inserted in order to vanish that consequence (Figure 5.8).

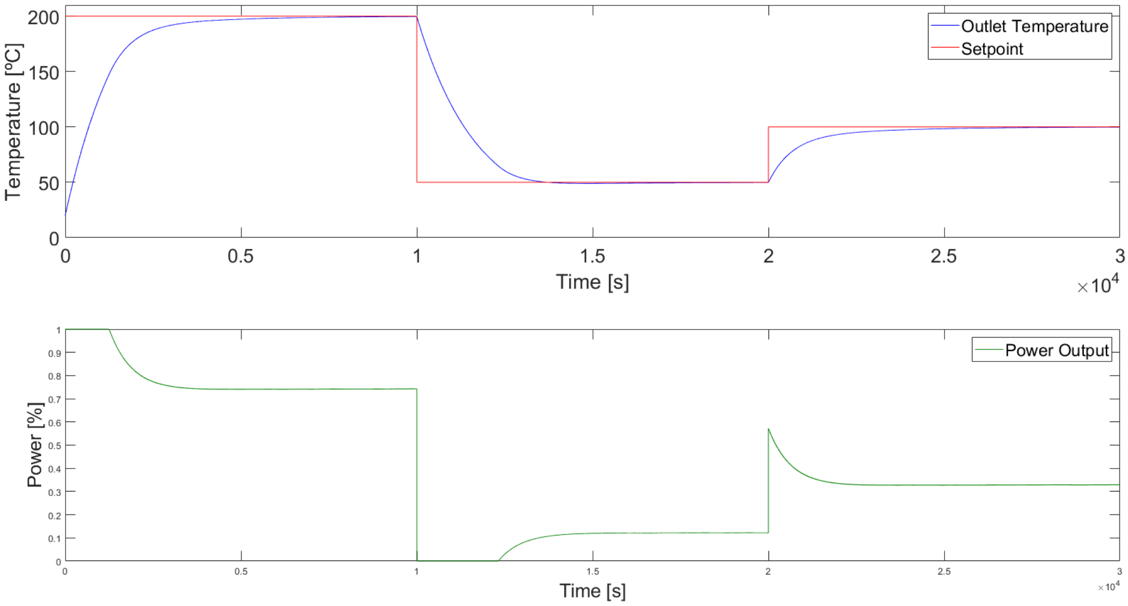


Figure 5.8: PI controller response with a proportional and integral term.

This controller presents a better response than the first one. It reaches the desired temperature more smoothly and the oscillations are imperceptible. Therefore, in future experiences the proportional and integral parameters to be used are 0.009 and 3.9x10^{-6}, respectively.

5.5 Signal regulation

For the interaction between the different systems of this case study it is necessary to develop the signal conditioning/regulation, which allows the real system, the μC, and the CLP1104 connector panel to be interconnected. There is a wide variety of signals, as can be seen in Table 5.2, so it is important to carry out the detailed electrical design in order to avoid corrupted information received from the signals and circuit construction. Regarding the signal conditioning modules, these will be designed only to establish the connection between the real and control systems.

Table 5.2: Input/output signals range of the components.

Equipment	**Input/Output voltage**	
Electrical heating resistor	230 VAC	
Transmitter	0 - 5 VDC	
Relay module	Maximum AC (Common) Supply	10 - 250 VAC 5 VDC
Arduino	Operating Voltage Input Voltage Limits Analog Pins Digital Pins	5 VDC 6 - 20 VDC 0 - 5 VDC 5 VDC
CLP1104 Connector panel	Input Voltage Range Output Voltage Range	±10 VDC ±10 VDC

To implement the idea of flexibility in the different types of tests, the Simulink model itself needs to be adapted to run real-time simulations and switch between simulations. It also needs to be capable of sending signals to the exterior as well as receiving them. Next, several main changes in the Simulink models will be presented.

5.5.1 Adaptation of the analog and digital signals

As previously mentioned, the analog signals to be sent and received are associated with the thermocouple probes. The type K thermocouple has a full working range between -200 and 1370 °C, and the PT100 sensor has a range from -200 to 850 °C. After programming the transmitters, not only the full working range of the sensors is reduced, but also the signals arrive in a range of 0 to 5 V since the CLP1104 connector panel sends and receives digital and analog signals from -10 to 10 V.

As mentioned in Section 4.2, Real-Time Interface has its own library with the appropriate Simulink blocks to send and receive signals to the connector panel.

These blocks proved to be useful, for example, when the virtual controller was controlling the real system. To send the digital signals from the temperature sensor to the connector panel, an ADC (Analog to Digital Conversion) block was used and needs to be adjusted by the maximum value that the sensor allows. To be more accurate, if the temperature read from the thermocouple is around 150 °C, the output of the transmitter will be about 2.5 V, which is half of the voltage range. Therefore, the ADC block needs to be multiplied because this has an output range of ±1.

To receive the signals emitted from the connector panel DAC (Digital to Analog Conversion) blocks were used. This turned out to be very helpful when the virtual system is controlled by the Arduino. The signal is sent to the μC in the form of temperature, and the output will be the controller output. Between these steps, some conversions need to be implemented. The first conversion consists of multiplying 0.25/300, this is translated in the division between half of the voltage that the Arduino is capable of administrating and the maximum range of the sensor temperature. Afterwards, this value is added by an offset which always ensures a positive result. This allows the range to go from [-300, 300] °C to [0, 600] °C. An additional conversion is used in order to obtain the best possible resolution of the Arduino.

It is important to mention that immediately before the DAC block, a saturation needs to implemented in order to avoid any damage of the μC since the input range of the pins is from -0.5 to +5.5 V. Figure 5.9 shows the conversions made as well as the ADC and DAC blocks in this last scenario.

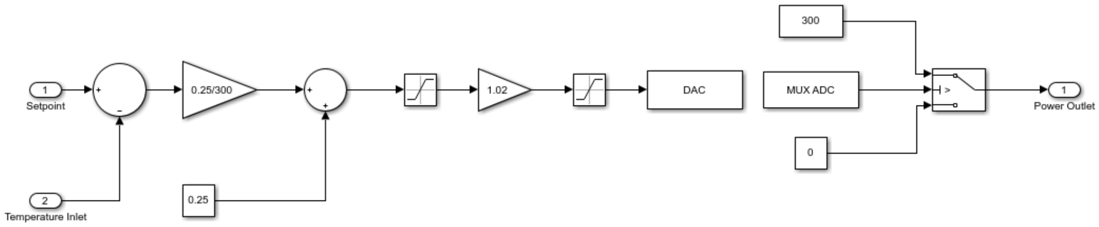


Figure 5.9: Set of blocks associated to receiving and sending signals, and their conversions.

5.6 Graphical interface developed for monitoring

As mentioned in Section 2.1, to monitor and control the simulation in real-time it is necessary a dedicated software for that purpose which in the case of dSPACE is ControlDesk.

With ControlDesk software it is possible to monitor and acquire the state of the signals or variables, register data and perform simulation in real-time. The variables such as temperature and power were implemented in the graphical interface as well as the tension measured incoming and outgoing from the connector panel.

Regarding the data acquisition, the feature 'Start Immediate Recording' from this software was used. It allows the user to save the data from the experiment in a file with a '.mat' extension, parameters that were used, which in the case were the temperature inside the oven, the environment temperature and the PWM signal. After the file has been saved, it can be changed and processed, or it can be used to construct graphs.

Appendix B shows the graphical user interface used in this case study.

5.7 Arduino programming

The μC used to control both virtual and real systems was the Arduino Uno. This board has an ATmega328P processor with a 16 MHz crystal oscillator and a 5-Volt linear regulator. This board also supports six analog inputs (pins) and fourteen digital I/O, in which six pins provide an 8-bit PWM signal [66]. The temperature sensors were implemented in analog pins 'A0' and 'A1'. The relay module received a signal from digital pin number nine.

The principle behind the controller implemented in the Arduino is very similar to the virtual controller referenced in Section 5.4. Based on the inlet and outlet temperature, the power required to give to the system is calculated consecutively along time, which then sends a signal through the output of the controller.

The structure of the source code implemented in the Arduino is divided by the three parts. The first part is dedicated to declare the program variables. The second part is defined by the setup, which is where some of the necessary configurations, initial setups or even to start the communication to the serial monitor are made. The last part is the main loop, where the fundamental code of the program is inserted. This is continuously run in an infinite cycle, and contains the signal acquisition as well as the calculations of the output of the controller as well as the power output, and the ability to send the signal from the controller. The measurement of the analog input signal was made with the temperature sensor, and the code was implemented in the main loop, where each cycle is called and then it will be used to calculate, the controller output, and consequently the power output.

There are two different Arduino programs for this case study: the first one is to control the virtual system and, the second one is to control the real system. They are very identical, the main difference being that the second program enables to establish two steps of setpoint.

The fundamental aspect of this program is its ability to run the code inside the main loop every 100 ms. This translates in the cycle time of the controller, and it means that within every 100 ms the Arduino sends the controller output.

5.8 Experimental procedure

After construction of the experimental platform, the next step will be to carry out the tests. To accomplish the practical component, a set of procedures had to be followed in order to maintain the system in an appropriate and functioning manner. In this section the general view of the test will be explained, taking into account how the model was developed.

The next step, after obtaining the desired parameters, is to simulate the oven model controller board in real-time. The model needs to be developed and only then it is uploaded to the dSPACE software through Real-Time Workshop in order to be simulated in real-time.

To test the virtual controller in the real system, initially the signals need to be adapted for this type of test so that the controller receives the temperature at the analog inputs from the connector panel and, afterward, it sends the PWM signal to the digital output. The Simulink model needs to be compiled, which then enables to monitor the responses of the system through ControlDesk interface. Data are also registered in '.mat' file every 0.01 s.

As the virtual controller was controlled by the μC, the procedure was slightly different from the experience previously mentioned. The Arduino receives temperature measured by the sensor and through the program calculates the resulting controller output, and then it sends a PWM signal over the digital pins to the virtual model. The process of data acquisition is similar to the earlier experience.

Lastly, the Arduino controls the real system, where all responsibility falls in the program implemented in the μC. The data acquisition and monitoring are also performed under ControlDesk.

Within each experience the fundamental parameters were carefully adjusted. Each variable needed to be measured with accuracy so that they would be used in the virtual model and the controllers to carry out tests under the same conditions.

For a greater chronological logic of the tests, the ambition was to keep the same procedure in the product development process and try to approximate the order of the tests to what could happen in an industrial environment. Thus, the first step is to validate the model by comparing the tests with the real system. Then, through testing both real and virtual models, design the controller to be more efficient. With the controller designed and approved, the next step will be to implement it in the μC and test it in the virtual system, in order to verify if both virtual and real controllers have the same response. And for last, perform experiences in a real environment to certify if the behaviour is as expected.

It is important to mention, that these experiences were long, to be more precise, about eight hours and thirty minutes. This is due to the fact that the modeled system is very slow and the heating resistor does not have much power when supplying heat to the system.

The following section describes the distinct methods that the virtual test bench enables, the respective experimental results, and a brief results' analysis.

5.9 Experiments with the different operating modes

With the circuits built, the equipment assembled and the entire system modeled, it is time to prove that the proposed concept is indeed valid and plausible. Temperature control tests were performed with the variants of the model, in order to obtain results from each experiment. Therefore, the first type of test will be to register data in a real-time simulation with a simple virtual model. Then, with the optimum parameters, different type tests were carried out, also in a simulated environment, and the objective was to analyze the behavior of the virtual controller in the virtual model. The third group of tests was to control the equipment with a virtual controller designed in

Simulink, with the purpose of confirming if temperature controlled was near the desired temperature, by tuning the parameters of the PI controller. After further adjustments and optimization of the virtual controller, a program is developed in Arduino and the tests are carried out in a simulated environment. For the last group of tests, the μC will control the real system.

The following section will describe each step performed with the respective results and adjustments to the controller.

Data logging

The virtual systems are intended to translate into a simulator what is actually happening in real conditions. Obviously, these systems only behave according to the variants imposed by those who develop them. However, there are always complicated parameters to model and determine, which makes the virtual system different from a real one. As mentioned before, two possible solutions were proposed in order to determine the unknown parameters. Additionally, the Simulink model needs to be adapted for this type of simulation in this case, the connector panel receives the signals generated by the temperature sensors, and the simulated PWM signal, sent from the connector panel that is implemented in Simulink, controls the real system. Figure 5.10 shows this situation, where the models of the four experiences are similar but only the duty cycle changes.

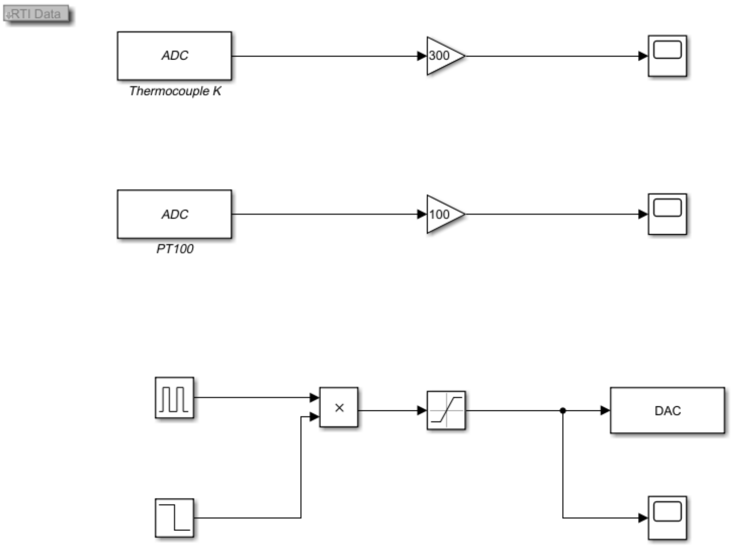


Figure 5.10: Block diagram of the simulated PWM signal.

Applying the methods described in Section 5.3, it was possible to make a comparison between the analytical method and the experimental data. Figure 5.11 shows the difference between both data.

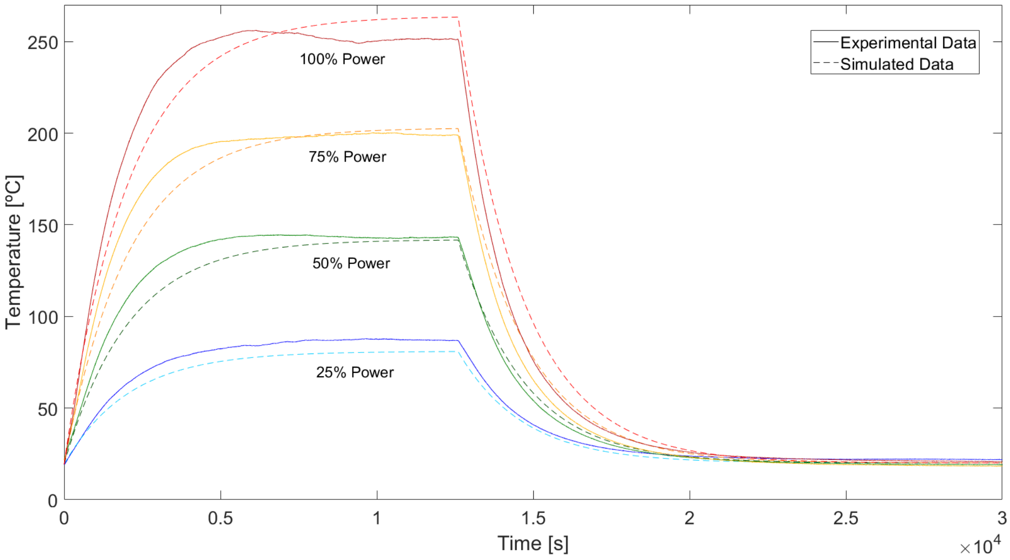


Figure 5.11: Comparison between the experimental and simulated data.

The parameters obtained by the optimized method slightly improved the behavior of the virtual model. This can be examined in Figure 5.12, which compares the simulated data, where the parameters were obtained by using the optimized method.

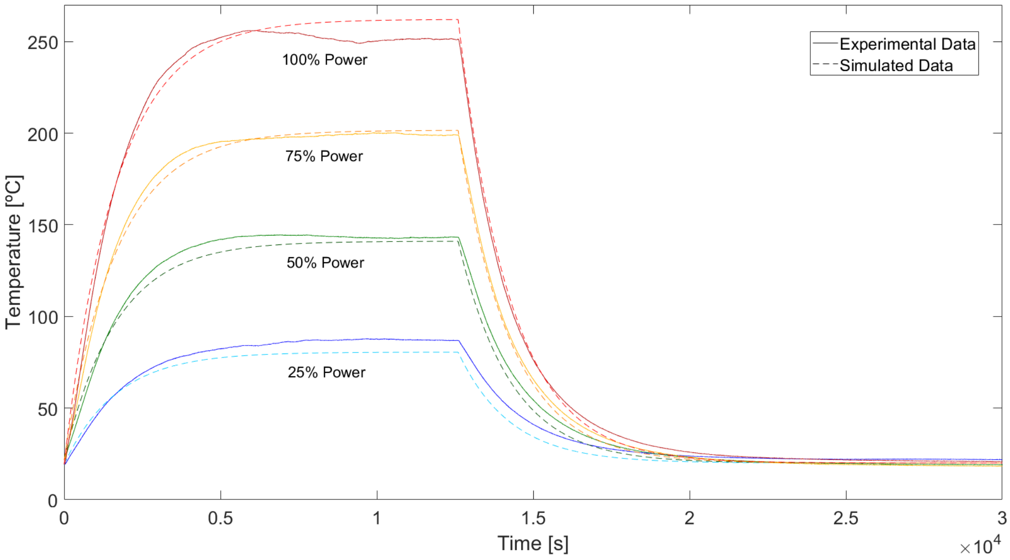


Figure 5.12: Comparison between the experimental and simulated data parameters obtained using the optimized method.

For future experiences, the parameters acquired through the optimized method will be used since they present better results than the analytical method.

Appendix C shows a schematic of the circuit built to perform this type of experiences.

Real-time simulation with the virtual model

As mentioned previously, after modeling, defining the parameters and validating the entire system, it is required to demonstrate that the designed controller has the appropriate behavior. To do so, with the method mentioned in Section 5.4, and the parameters determined in the same section the controller can be tested.

The overall heat conductance and the thermal capacitance have already been determined. It was also necessary to parameterize all common external variables such as the room temperature and the heat generated by the electrical resistor, which are measured experimentally. Table 5.3 summarizes the parameters used in the model and in Appendix D it is presented the Simulink model used to simulate the dynamic system.

Table 5.3: Parameters of the virtual system.

Parameters of the virtual system	**Value [SI]**
Overall heat conductance	1.24 $[W/°C]$
Thermal capacitance	2068.22 $[J/°C]$
Maximum heating power	300 $[W]$
Room temperature	20 $[°C]$
Proportional term	0.009
Integral term	3.9e-6

Now that the parameters are inserted into the model, the virtual simulation in real-time was carried as if it were in reality. Three steps of setpoints were established with the purpose of testing the controller as well as the behaviour of the thermal system. Figure 5.13 shows the experience in real-time simulated environment, with the setpoint, outlet temperature and the inlet power to the virtual model.

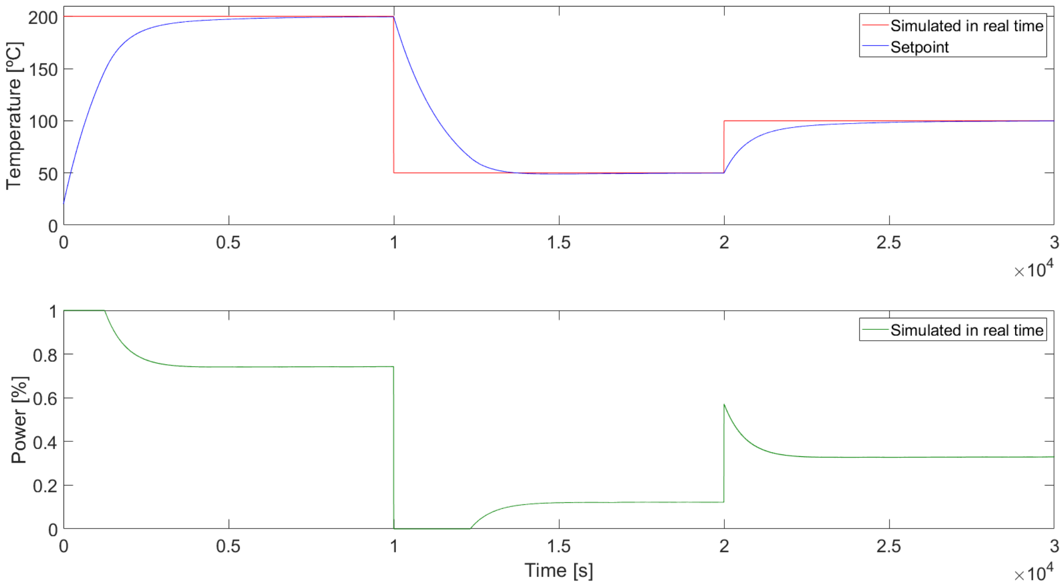


Figure 5.13: Virtual simulation in real-time experience.

Hardware-in-the-loop simulation between the real controller and the virtual system

With the virtual controller successfully implemented in the virtual system, the next step was to apply the μC. Thus, the Arduino was programmed with the same conditions as the virtual controller and the PI parameters. The best way to test whether the program is functioning properly is to test it in a virtual environment before it is implemented in the real model, avoiding possible errors and equipment damage.

With the appropriate signal adaptations and interconnection between systems, it was possible to obtain the experimental data resulting from the experiment, which are exhibited in Figure 5.14.

The signal conditioning and the virtual model developed in MATLAB/Simulink is exhibited in Appendix D. The schematics of the connections made between the μC and the CLP1104 connector panel is shown in Appendix C.

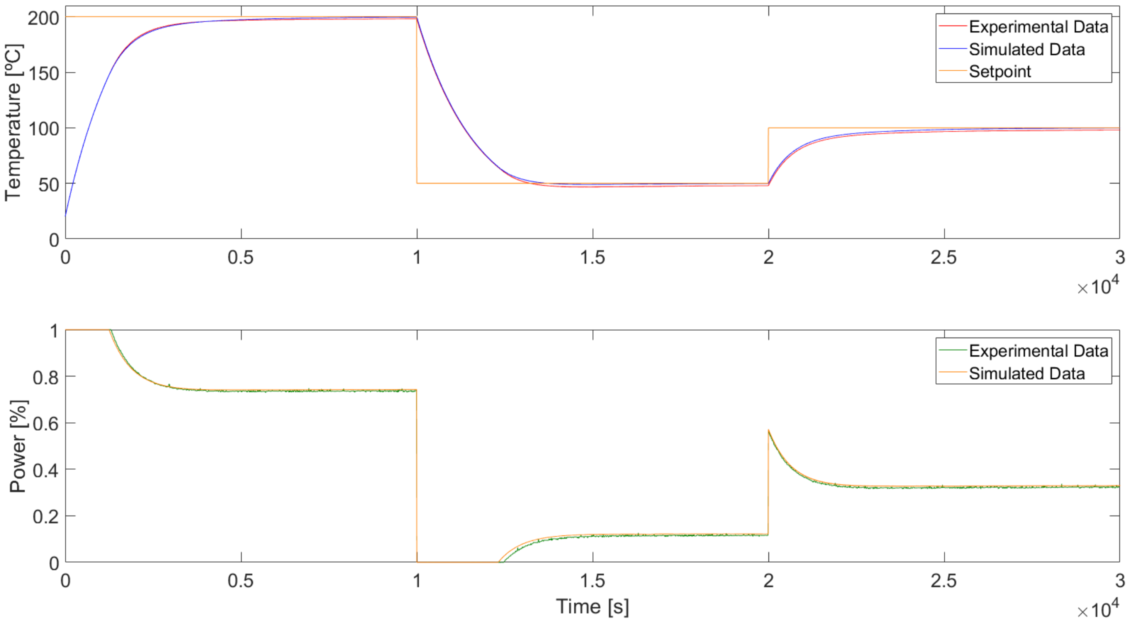


Figure 5.14: Real controller and virtual model experience.

Hardware-in-the-loop simulation between the virtual controller and the real system

The tests in the simulated environment were suitable enough to advance to the next step consisting in controlling the real system with the proposed virtual controller. This step is very useful because it is most likely that the equipment may not have the same reaction as the virtual model. In this experience, the parameters inserted in the model were the same, and the remaining data were measured using sensors and a relay, such as room temperature, outlet temperature, and inlet power.

The conditions are similar to these of the previous experience, although some modifications were necessary. The virtual controller receives the signals from both temperature sensors, and a signal is sent to power the electrical resistor through the relay module. Figure 5.15 shows the experience in real-time, comparing the virtual simulation with the results obtained from this one.

The signal regulation, the virtual controller and the necessary adjustments developed in MATLAB/Simulink are presented in Appendix D. The schematics of the connections made between the CLP1104 connector panel and the real system is identical to the first operating mode, and it is illustrated in Appendix C.

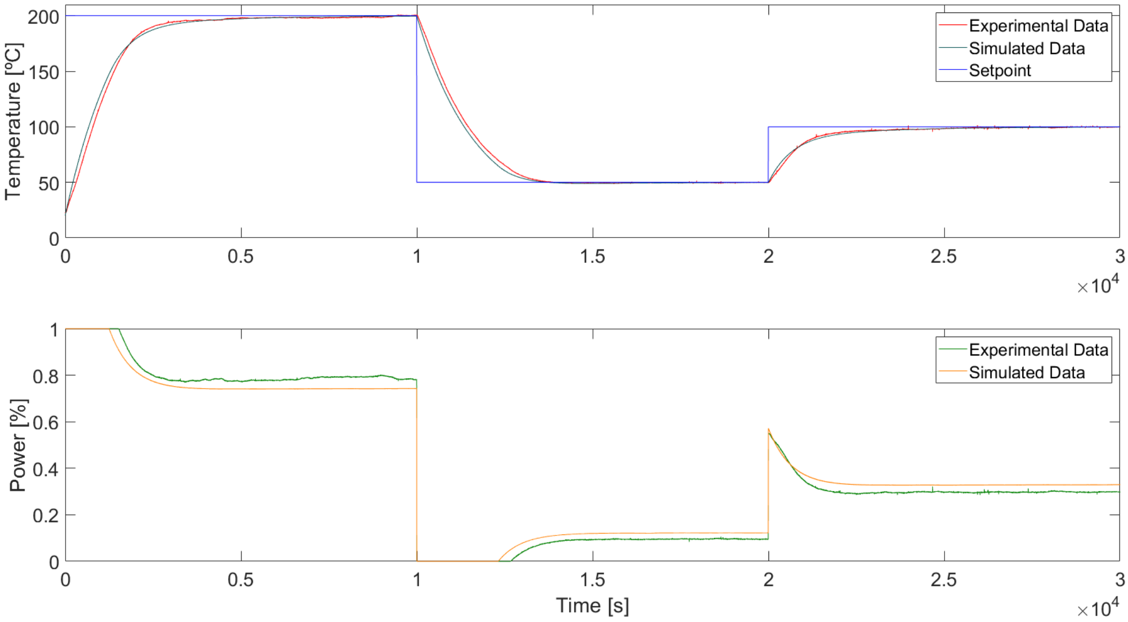


Figure 5.15: Virtual controller and real system experience.

Interaction between the real system and the microcontroller

After ensuring that the Arduino has the expected response when controlling the virtual system, all that's left is to test it on the physical system. This represents the final phase of the case study, where all the types of equipment are real and there is no simulated model or controller.

It was possible to register the data from the experience through the Arduino. Although through the dSPACE software, it acquires data in real-time. With the right connections to the connector panel, all the information received from the signals emitted by the sensors was easily obtained with a simple graphical interface developed in ControlDesk and a Simulink model.

Figure 5.16 shows the results of the experience recently discussed.

The Simulink developed based on the acquisition of data to the controller board is presented in Appendix D. The schematics of the electrical connections between systems is illustrated in Appendix C.

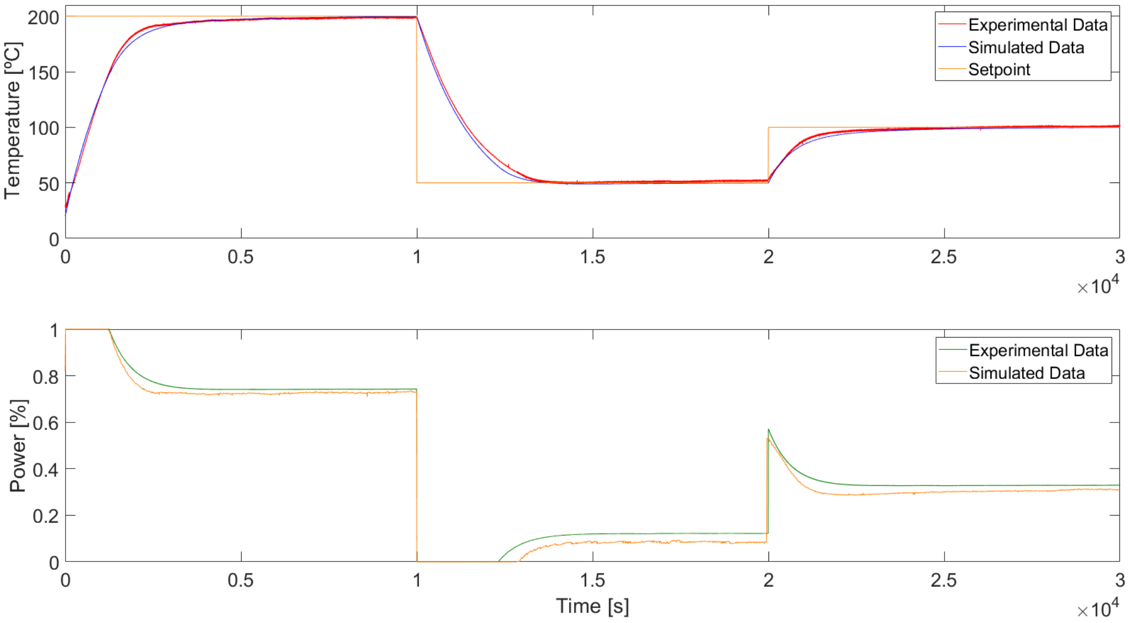


Figure 5.16: Real system controlled by the real controller experience.

5.10 Analysis of the experimental data

Analysis starts with the open-loop experiences and data acquisition. By interpreting Figure 5.12, in the experience with a duty cycle of 25% it is noticeable that the experimental and simulated data are separated by almost 8 °C at the region where the temperature is steady. Additionally, both experiences at a duty cycle of 50% and 100% show some oscillations when reaching a steady-state temperature. These may be explained due to the fact that the complexity of the model is so low that it does not take into account the heat transfers by convection and radiation.

Next, the controller's performance will be analyzed where the optimum parameters are used. When applying the proposed methodology, a controller with a good response capacity was achieved to the system under consideration. It proved to be efficient, can reach the setpoints smoothly with no oscillations, undershoots, overshoots and small error at the steady-state region.

Regarding the hardware-in-the-loop simulation where the real controller controls the virtual model, there were also positive results. When comparing the experimental data with the virtual simulation in real-time (Figure 5.14) there is almost no difference in the temperature registered. Although in the power output some disturbances can be observed, this may be due to the noise when the Arduino acquires the signal.

For the test of the virtual controller in the real system, despite the oscillations encountered it responded well, reached the setpoints and when compared to the simulated data (Figure 5.15) it is in good agreement. But there is a noticeable difference from the controller's power output; one of the reasons that can justify this difference is that the tests were carried out on different days, so the room temperature was also different.

Regarding the last group of tests, where the real system is controlled by the Arduino, the results were quite satisfactory. This can be visualized in Figure 5.16, that the PI controller has been implemented effectively since it almost overlaps with the values corresponding to the virtual system controlled by the virtual controller. In the case of the controller's power output, the observed differences may have been caused due to noise when acquiring the data, since the software ControlDesk was used in order to register the data from the experience, and the connections between the types of equipment were not the best.

Chapter 6

Case study 2

With case study 1, it was possible to successfully perform the experiences on the virtual test bench and validate the concept behind it. Without further delay, the second case study was carried out. The objective of Case Study 2 is to operate an actual water heating device, is this case a tankless gas water heater provided by Bosch Thermotechnology.

The structure of this case study is very similar to that of the first one, the main differences being the equipments used to carry out the experiences and the sensors to measure and monitor the behaviour of the water heating device.

6.1 Case study framework

Case study 1 served to support the fundamental concept behind the virtual test bench. With careful consideration to Bosch TT by having an interest on this virtual test bench, and for providing a sophisticated tankless gas water heater, it was possible to carry out experiences with a higher complexity.

The Hydro 4600 F WTD10-4KME 23 JU is a residential and commercial TGWH condensation model with a thermal power of 22 kW, 86% thermal efficiency, a maximum water flow rate of 8 L/min, with a forced ventilation sytem, which means that a fan is integrated in the chimney to improve, and in some installations conditions to ensure, the exhaustion of combustion gases. This device also includes a digital display to exhibit the water temperature outlet and possible errors of malfunction, and works with natural gas or G.P.L. It has safety devices, such as a ionization electrode to prevent accidental extinction of the flame ignited by the burner, a burner flame condition monitoring device which turns off the equipment if the flame state is significantly low, and a temperature limiter to prevent the overheat of the combustion's chamber.

The European Union defined energy policies to meet the challenges related to climate changes, security of supply, and competitiveness with new energy labelling and ecodesign requirements for heaters and water heaters. Ecodesign requirements are mandatory for all heater manufacturers and suppliers wishing to sell their products in the EU. The regulations set requirements on energy efficiency, nitrogen oxide emission levels, volume for storage water heaters, and heat losses from hot water storage tanks [67], bringing new challenges to the manufactures [68].

Besides the water heating device, several types of equipment were used to measure and control the desired parameters of the system. In addition to the manual valve, a proportional solenoid valve was one of the types of equipment used to control the hot water flow rate by varying the size of the flow passage via a restrictor. Table 6.1 summarizes the main equipments used in the experimental platform, besides the ones already mentioned in Chapter 4.

Table 6.1: List of components associated with control and sensing.

Qty.	**Description**
1	Carbon monoxide detector
2	K-type thermocouple probes
1	RTD PT100 probe
2	Pressure sensor with display
1	Water flow meter
1	Solenoid water flow control valve
1	Manual water flow control valve

Regarding the dynamic models and parametrization, these will be very different due to the complexity of this system. Although, the software for simulation, data acquisition and control will still be dSPACE and the modeling software will also be with MATLAB/ Simulink.

The principle behind the real controller and its functioning will be the same as those used by Bosch TT when applied to the water heaters.

In terms of modeling systems and virtual controllers, they will also be different, as well as other adaptations will be necessary to be implemented in the models to receive and send signals to the outside.

To connect the output and input tubes from the tankless water heating equipment, a stainless steel tube structure was built. This material was chosen because of its high resistance to corrosion since several experiences with water will be carried out. The equipments mentioned in Table 6.1 were assembled in this structure with the appropriate fittings and attachments. To connect the cool water from the line to the pipe structure, and the outlet water from the heating device a plastic PVC water tube was used.

As can be seen, it is intended that this case study is similar to case study 1, although some adjustments and improvements to the equipment were made, but the signal conditioning techniques and the distribution of the elements on the virtual bench are the same.

For safety reasons a carbon monoxide detector was used, preventing carbon monoxide poisoning.

6.2 Modeling the dynamics of the heat cell

This section presents the methodology adopted in the development of the dynamic models to express the dynamic action of the thermal, fluidic and mechanical components.

6.2.1 Thermal component modeling methodology

The systems are modelled by using concentrated parameters.

A lumped space approach was used to model individual components. The lumped system analysis was preferred over distributed analysis, considered through a finite element or finite difference methods, in order to meet the requirements for implementation of predictive control algorithms in computationally limited embedded systems.

The mathematical models result from the application of physical laws that describe, with small but acceptable deviations, the dynamics of the system. For the heat cell, a semi-empirical model was used.

The individual components considered for modeling are a valve and a heat cell. Each component is modelled considering a control volume, for which mass and energy conservation equations are established. The thermal model is detached from the fluidic model.

The fluidic model describes pressure and fluid flow dynamics, starting from the mass conservation law applied to a control volume

$$\frac{dm_{CV}}{dt} = \sum_{in} \dot{m} - \sum_{out} \dot{m} \tag{6.1}$$

where m_{CV} is the mass contained in each instant within the system and $\dot{m}$ is the mass flow rate.

The thermal dynamics is based on the energy conservation equation for the control volume,

$$\frac{dE_{CV}}{dt} = \dot{Q} + \dot{W}_{CV} + \sum_{in} \dot{m}\left(h + \frac{1}{2}V^2 + gz\right) - \sum_{out} \dot{m}\left(h + \frac{1}{2}V^2 + gz\right) \tag{6.2}$$

where $\dot{W}_{CV}$ is the rate of energy transferred by as work, $\dot{Q}$ is the energy transferred to (or withdrawn from) the system as heat, $\dot{m}$ is the mass flow rate, h is the internal energy per unit mass of the fluid, $(1/2)V^2$ is the kinetic energy per unit mass of the fluid, gz is the gravitational potential energy per unit mass of the fluid, E_{CV} is the energy contained in each instant within the system, determined by the sum of the different forms of energy,

$$E_{CV} = U + \frac{1}{2}mV^2 + mgz + m(...) \tag{6.3}$$

where U is the internal energy of the system and m is an intrinsic characteristic of mass. In the cases under analysis the mass contained in each module is constant, and only the time variations of the internal energy of the system are important, which are calculated by $mc(dT/dt)$. In terms of energy conservation, the parameters relative to the changes on kinetic and potential energy are negligible, the specific enthalpy depends only on temperature and the mass inside each control volume is considered to be constant and there is no energy transferred as work. Thus, the simplified forms of Equations 6.1 and 6.2 to consider are

$$0 = \sum_{in} \dot{m} - \sum_{out} \dot{m} \tag{6.4}$$

$$mc\frac{dT}{dt} = \dot{Q} + \sum_{in} \dot{m}cT - \sum_{out} \dot{m}cT \tag{6.5}$$

where $\dot{Q}$ is the energy transferred to (or withdrawn from) the system as heat, c is the specific heat and T is the temperature.

The water circuit configuration of the TGWH is simple and does not include any type of integrated circuit, its schematics being presented in Figure 6.1.

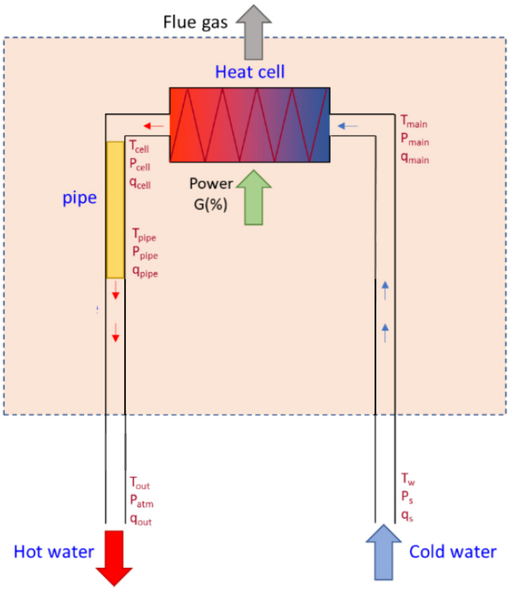


Figure 6.1: Schematic of the TGWH (Adapted from [68]).

The heat cell embraces the gas combustion burner and the transfer of energy to the water in the heat exchanger with condensation of water in the flue gases.

For the heat cell, assuming that the water and metal (copper alloy) are at a similar temperature and the fluid density variations are negligible, the water temperature at the exit can be expressed by setting the energy conservation equation for a control volume as

$$(m_w c_w + m_m c_m)\frac{dT}{dt} = \dot{Q} + \dot{m}c_w(T_{in} - T) \tag{6.6}$$

The heating power $\dot{Q}$ is defined by the flow of air and gas mixture, controlled by a gas valve and atmospheric or fan forced air inlet, considering a heating efficiency η.

6.2.2 Fluidic component modeling methodology

From the mass conservation law, Equation 6.1, for a constant volume and assuming the linearized equation of a fluid,

$$\rho(P,T) = \rho_0\left[1 + \frac{1}{\beta e}(P - P_0) - \alpha(T - T_0)\right] \tag{6.7}$$

and

$$m = \rho V \tag{6.8}$$

where α is the thermal expansion coefficient, β_e the bulk modulus which is defined by Equation 6.9, and V the volume.

$$\beta_e^{-1} = -\frac{1}{V}\left(\frac{\partial V}{\partial P}\right)_T \tag{6.9}$$

It is also assumed that the effects of compressibility are more significant than the effects of thermal expansion and Equation 6.10 comes

$$\rho(P) = \rho_0\left[1 + \frac{1}{\beta e}(P - P_0)\right] \tag{6.10}$$

Thus, the mass conservation Equation 6.1 can be rewritten to obtain a generalized continuity expression

$$\sum_{in}\dot{q} - \sum_{out}\dot{q} = \frac{dV}{dt} + \frac{V}{\beta e}\frac{dP}{dt} \tag{6.11}$$

where $\dot{q}$ corresponds to the volumetric flow rate, V is the volume of the control volume and P the internal pressure of the control volume.

With this, Equation 6.11 can be simplified if the volume of the control volume remains unchanged

$$\sum_{in}\dot{q} - \sum_{out}\dot{q} = \frac{V}{\beta e}\frac{dP}{dt} \tag{6.12}$$

The Reynolds number is used to determine the regime of water flow across each component. For turbulent flow, the water volumetric flow rate through an orifice is expressed by

$$\dot{q} = sign(\Delta P)C_d A_0\sqrt{\frac{2}{\rho}|\Delta P|} \tag{6.13}$$

where ΔP is the difference between the inlet and outlet pressure of the orifice, C_d is the discharge coefficient, A_0 is the orifice area, $sign(\Delta P)$ is the signal of ΔP and ρ is the fluid density.

For laminar flow, and considering a hydraulic conductance, the water volumetric flow rate can be expressed through the following equation

$$\dot{q} = g\Delta P \tag{6.14}$$

where g is the conductance of the orifice, which depends on the orifice's section.

Therefore, to calculate the evolution of the pressure in a certain volume control, and also to determine the inlet and outlet of the volumetric flow rate, the Equations 6.12 - 6.14 are used.

The heat cell involves processes of greater complexity associated with combustion and gases condensation. As a semi-empirical model has been designed, it requires a parametrization using experimental results to reasonably reproduce the static and dynamic relationship between the water flow rate, the temperature and the thermal power delivered to the heating cell, and the temperature of the water at the exit of the heat cell.

When modeling the thermal plant, the pressure drops are considered to be located in the orifices and the evolution of the pressure inside the control volume is determined by the continuity Equation 6.12.

6.3 Methodology proposed to implement the models in Matlab/Simulink

Regarding the implementation of the different models, the following approach was considered:

- A standard block with three inputs ($P_{in}, T_{in}, \dot{q}_{amount}$) and three outputs where: (i) $\dot{q}_{amount}$ refers to the inlet flow rate of the block that models the upstream component; (ii) $\dot{q}_{in}$ considers the inlet flow rate of the component, and transports it to the output block in order to connect to the $\dot{q}_{amount}$ input of the downstream component; (iii) P_{in} and T_{in} are the inlet pressure and temperature of the component; (iv) P_{out} and T_{out} are the outlet pressure and temperature (these are calculated from the mass and energy conservation equations, and they represent the pressure and temperature in the control volume).

- A pressure drop located at the input of the modelled component (it may be an orifice or a different function that allows to calculate the volumetric flow rate dependent of the pressure difference, $P_{in} - P$, where P is the fluid pressure in the control volume.

6.3.1 Implementation of the proposed models in a Matlab/Simulink platform

In this context, several solutions were found, modelled and simulated, such as:

- A controller model with a feedforward and PID control combined;
- A heat cell model with the different equations that build this system;
- A simple valve model that is based on the mass and energy conservation equations.

Figure 6.2 shows the global model developed, which represents the final version of the solution and the model that implements a control strategy (controller, heat cell and outlet valve).

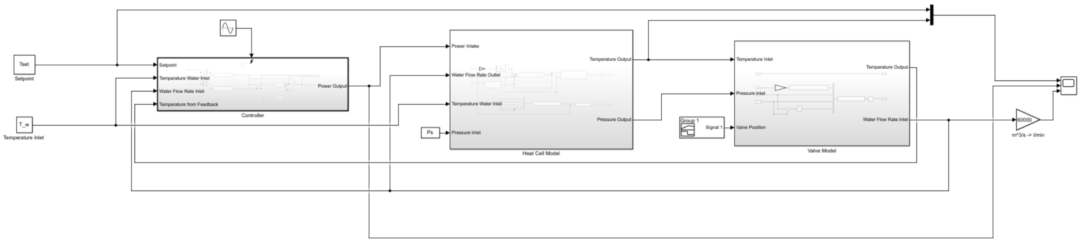


Figure 6.2: Block diagram of the global model.

Heat cell model

A semi-empirical model was developed in which some considerations are assumed in order to describe the dynamics observed in burners/real heat exchangers.

Thus, a model with concentrated parameters of a burner/heat exchanger is considered, where temporary delays for the evolution of the temperature at the exit of the heat cell are predicted. This enables to deliver thermal power (also considering an efficiency) and also increment the inlet and outlet temperature of the heat exchanger. The fundamental equations that describe this system are:

- $\dot{Q}_{in} = \eta \dot{Q}(t - \Delta t)$ where $\eta \leq 1$ is the efficiency, Δt is the delay between inlet power to the burner and its effect (ventilator delay, gas circuit, and others) and $\dot{Q}$ is the thermal power applied to the heat cell;
- $T_{out} = T(t - \Delta t_c) + \Delta T_{out}$, where $\Delta t_c = \frac{\pi R_i^2 L_t}{|\dot{q}_{in}|}$ (temporal delay due to the water flow rate), L_t is the length of the burner's pipe, R_i is the internal radius of the pipe and ΔT_{out} refers to the temperature increment at the outlet of the burner;
- $T_{in} = T_{env} + \Delta T_{in}$, where ΔT_{in} is the temperature increment at the inlet of the burner.

In order to clarify an issue, the metal (copper) of the pipe was considered to be at the same temperature of the water (T_w), in other words, $T_w = T_m = T$.

Figure 6.3 shows the proposed model of the heat cell.

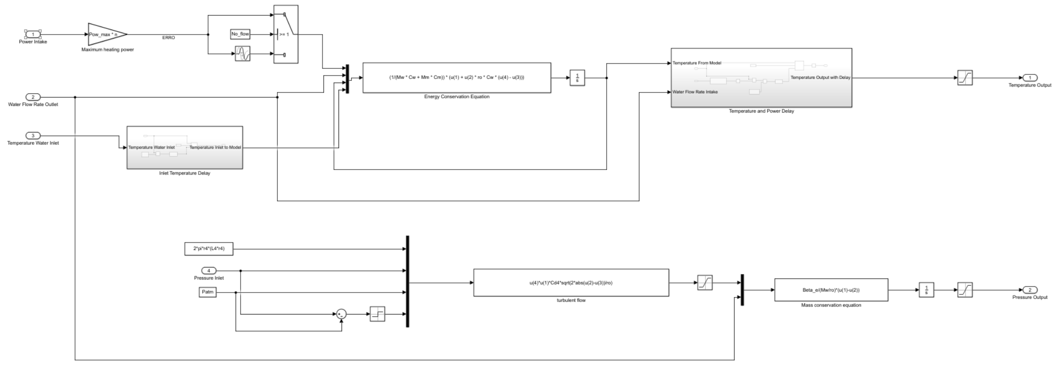


Figure 6.3: Block diagram of the heat cell model.

Valve model

For the proposed model valve, the continuity and energy conservation equations are used to describe the fluidic and thermal behavior, and the orifice equation to describe the pressure drop. The block diagram which implements the valve model is shown in Figure 6.4.

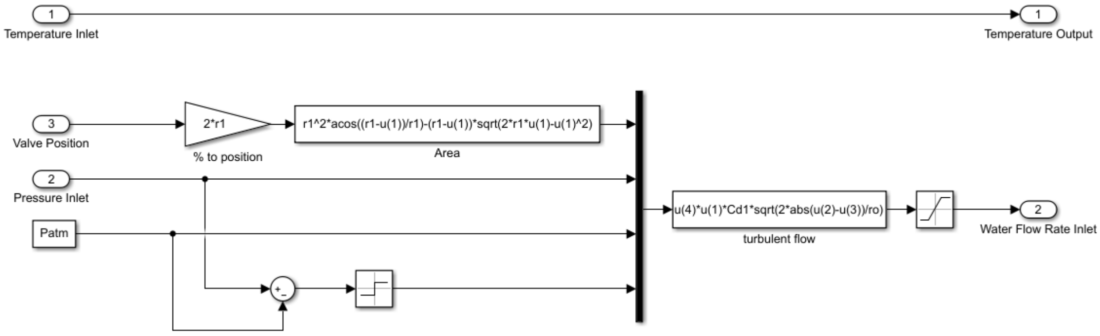


Figure 6.4: Block diagram of the valve model.

For a two-way proportional flow control valve, the orifice area A is determined by the spool position (x) and the radius (R) of the circular water inlet, which is defined as

$$A = R^2 cos^{-1}\left(\frac{R-x}{R}\right) - (R-x)\sqrt{2Rx - x^2} \tag{6.15}$$

The flow regime of the valve orifice is assumed to be turbulent and modeled by Equation 6.13.

The thermal component is modeled by the energy conservation equation and the fluidic component by the mass conservation equation in the form of Equation 6.12 for the control volume of a valve with one inlet and one outlet.

The proposed TGWH individual mathematical and semi-empirical models enable the simulation of several components, such as the heat cell and valve model.

6.4 Control strategy proposed for the model

In order to assess whether the developed models are suitable for use in the development and testing of innovative control strategies, some closed-loop simulations with a combined feedforward and PID control were performed. A model was developed for implementing thermal power control as schematically represented in Figure 6.5.

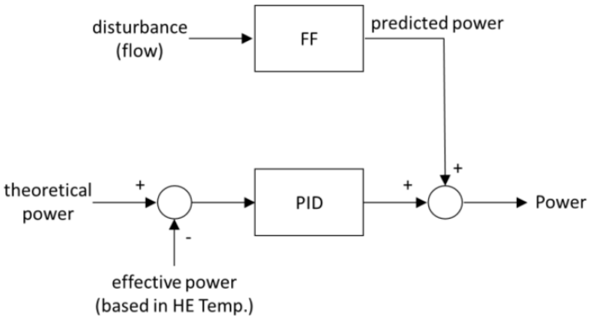


Figure 6.5: Schematics of the combined feed-forward feedback controller (Adapted from [68]).

6.4.1 Heat cell power control

The following strategy was implemented for the heat cell power control in a normal operating mode:

- First, the required theoretical power is calculated in order to supply the burner and reach the desired outlet temperature from the heat cell (initial estimate to the output of the controller). It is important to mention that the theoretical power depends on the inlet temperature and water flow rate.
- Second, the effective power is calculated in function of various parameters, such as the inlet temperature, the water flow rate and the temperature at the outlet of the heat cell;
- Third, the power error is determined and it is controlled, for example, with a PID controller;
- Fourth, the offset determined by the controller's action is added to the estimated power (theoretical power).

To run the entire controller block every 100 ms, the Simulink block 'Trigger Port' was implemented, which then allowed connecting, from the exterior of the subsystem, to a 'Sine Wave' block. This means that each time the sinusoidal wave changes signal, with a frequency of $2 \cdot \pi \cdot 10$ (rad/s), the controller activates every 100 ms.

6.5 Identification of the thermal model's parameters

To advance to the virtual simulation in real-time the parameters need to be defined, but since several variables could not be determined, the initial approach was to use the parameters from a Smart Green Homes Project, which also used a tankless gas water heater, to parameterize the developed models. A Bosch Greentherm T9800 SE 199 was used with a thermal power of 58.3 kW and thermal efficiency of 99%. Is also has a modulating gas valve, water valves, a water bypass circuit, burner power segmentation and forced air ventilation. This information and the data provided from the project lead to Table 6.2, which summarizes the parameters used in the initial virtual simulations, and were useful to design the controller.

Table 6.2: Initial parameters used for virtual simulations.

Section	**Parameters**	**Value [SI]**
Water circuit and environment	Inlet water temperature	16 [°C]
	Water temperature setpoint	45 [°C]
	Output relative pressure (atmospheric)	0 [Pa]
	Water mainline pressure	3x10^5 [Pa]
	Maximum water pressure	10x10^5 [Pa]
	Effective bulk modulus (air-water-mixture)	5x10^8 [Pa]
Output valve	Orifice radius	4x10^{-3} [m]
	Discharge coefficient	0.7
Heat exchanger	Maximum output power	58300 [W]
	Power delay	2.5 [s]
	Temperature increment at input	1.25 [°C]
	Temperature increment at output	1.25 [°C]
	Water mass	2 [kg]
	Cooper mass	2 [kg]
	Orifice radius	8x10^{-3} [m]
	Water circuit length	2.2 [m]
	Discharge coefficient	0.7

Only after performing experiences in the open-loop of the system, it is possible to measure and quantify the parameters needed from the system. This is due to the fact that the virtual test bench is not yet fully built, and the measuring equipments are not assembled as well. Although, it is enough to virtually simulate in real-time the dynamic models of the system, adjust the terms of the controller and to control the virtual model with the μC.

The method adopted to adjust the PID controller terms, based on these parameters, will be discussed in the next section.

6.6 Adjusting the parameters of the PID controller

In order to study the response of the virtual system, two combinations of tests were carried out. The first is to define three different temperature setpoints while the water flow rate is constant. The second combination of tests is to change the water flow rate while the temperature setpoint is consistent. This is more realistic because as, mentioned before, the water flow rate control simulates the user's output valve. Tests with 600 s were performed, and several parameters were monitored such as temperature outlet, output power from the controller and water flow rate.

The methodology adopted in order to adjust the parameters of the PID controller is different from the first case study. While in the first case study there was a logical methodology to obtain the gains terms, since the equations were simple, in this case the process used was trial and error. Although this method is not ideal to project the PID controller and may take a considerable time due to the complexity of the system, it is a simple way to parameterize the PID terms. Therefore, one method to manually determine the PID terms can be determined by following these steps [69], [70], [71]:

1. Set all gains to zero;
2. Increase the Proportional term until the response to a disturbance results in a steady oscillation;
3. Increase the Derivative term until the oscillations disappear;
4. Increase both P and D terms until the increasing of the D term does not interrupt the oscillations;
5. Set both P and D terms to their last stable values;
6. Increase the Integral term until it reaches the setpoint.

Now that the methods have been established, all that's left is to proceed with simulations. Therefore, the initial step is to set all the terms of the PID controller to zero as if there was no action from this.

Because the control is based on the combination of the feedforward and the PID controller, when the terms of the PID controller are null only the feedforward control is applied. The results can be seen in Figure 6.6.

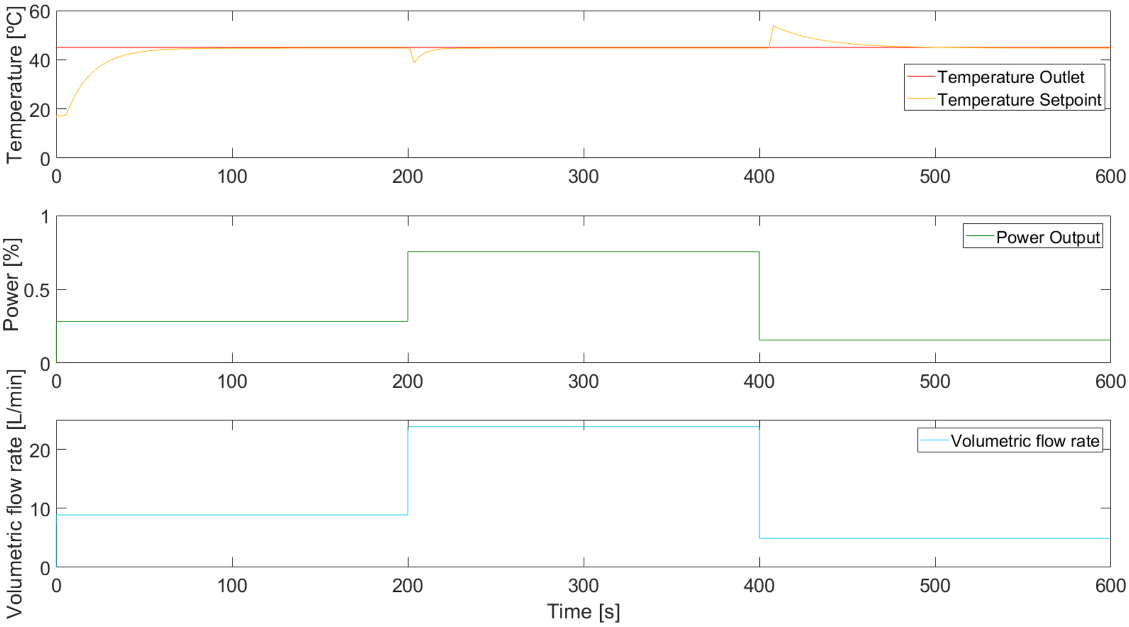


Figure 6.6: PID controller response with all terms set to zero.

Although the outcome of the simulation is quit satisfactory with only the feedforward control, this is not enough since there is a 5 °C difference from the temperature feedback and the setpoint. This is where the PID controller plays a fundamental role with the appropriate terms. Thus, comes the next step, which will be to set a Proportional gain until the response of the controller results in a steady oscillation (Figure 6.7).

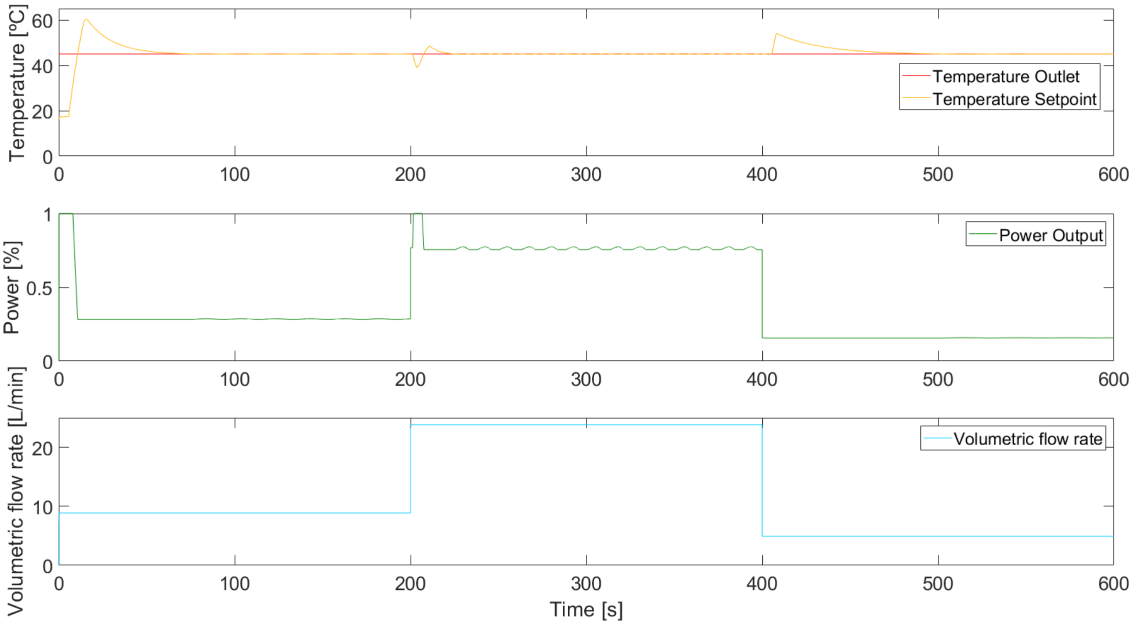


Figure 6.7: PID controller response with a proportional term.

As mentioned in the steps to tune the controller's terms, there is an oscillation in the steady-state region after the feedback temperature reaches the setpoint, so a derivative term is applied. The results have already improved significantly by the adding this term.

This means that all that's left is to include the integral term in order to increment or decrement the controller's output over time and reduce the error. Thus, through a long period of time, the integral action will drive the controller output until the error is zero.

The results acquired with the final parameters of the PID controller are presented in Figure 6.8.

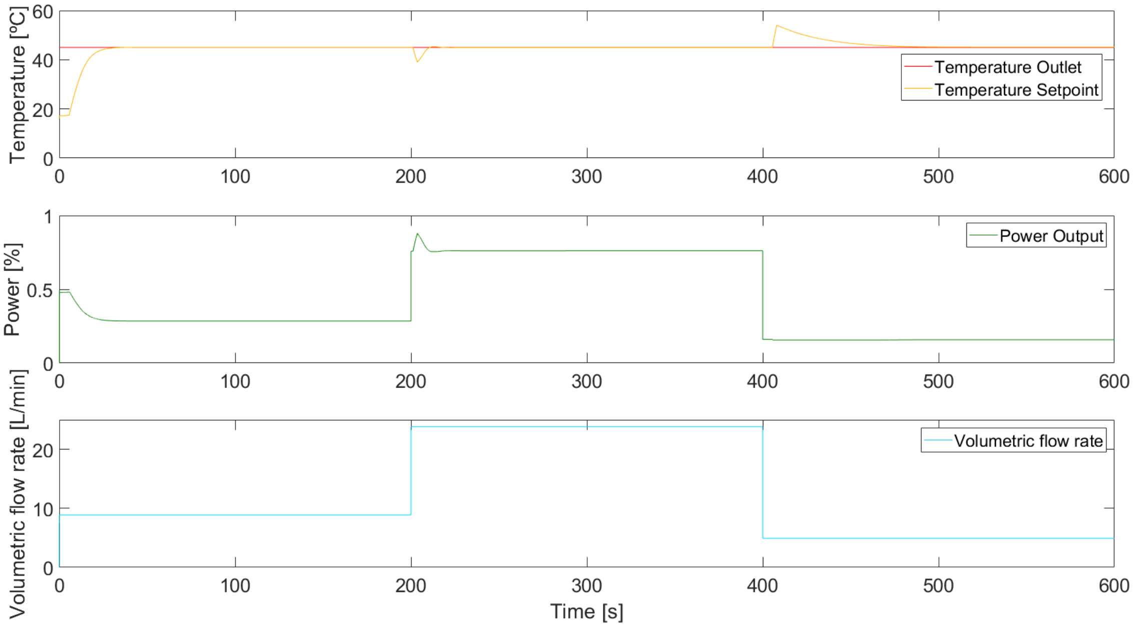


Figure 6.8: PID controller response with the final tuned parameters.

In further experiences, the proportional, integral and derivative terms of this controller to be used are 0.7, 0.001 and 0.005, respectively. These will be used in several tests such as when the virtual controller controls the real system and when it controls both real and virtual models, because these terms will be implemented in the Arduino's source code.

6.7 Adapting the equipment signals

As already discussed in the first case study, to provide a good and fast operation for this type of test bench it is necessary to perform the signal conditioning required to connect the signals of the connector panel, the μC and the measuring equipments. Therefore, a description of the equipment signals is illustrated in Appendix E.

The models dedicated to receive and send analog or digital signals, along with interface packages associated to dSPACE, are incorporated in Simulink environment. This process has already been mentioned in Section 5.5, so there is no need to explain their functionality again. Although, an example of a signal regulation to a component and the opposite, will be illustrated in Figure 6.9, knowing that the remaining signals share the same idea.

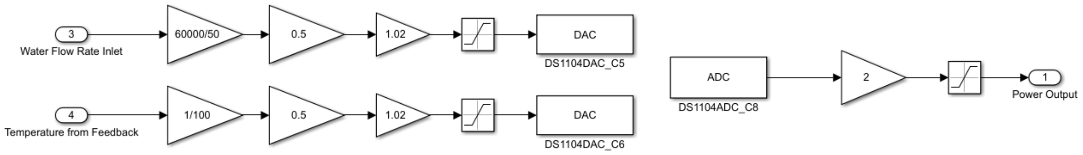
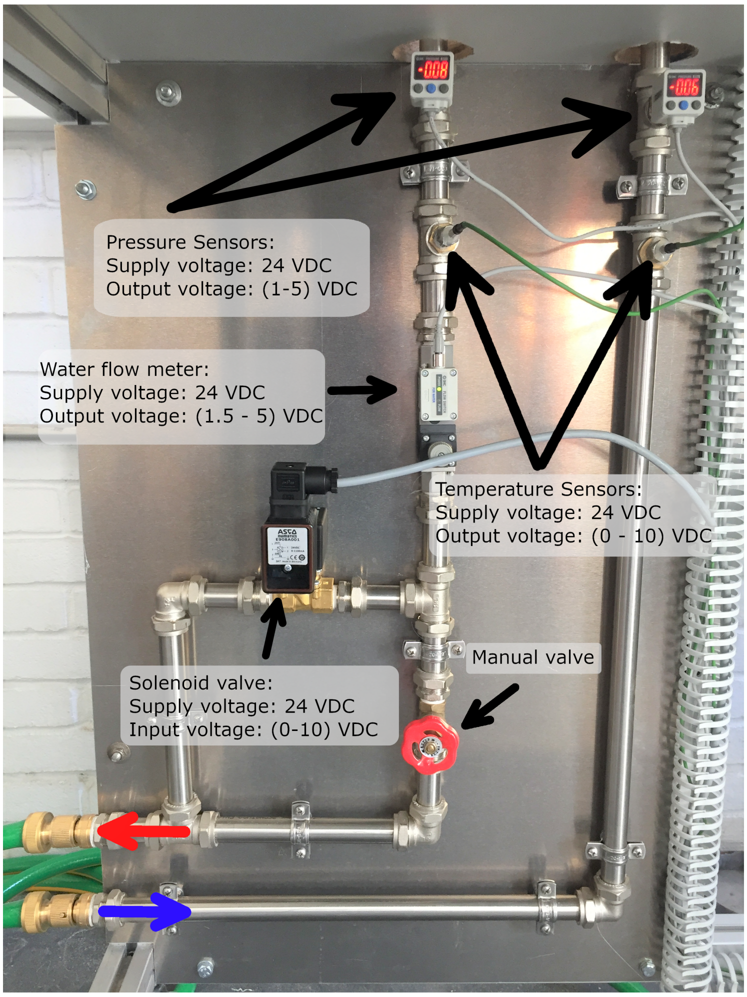


Figure 6.9: Block diagram of the interconnection with Arduino and virtual model.

In order to compare the adjustments made in the Simulink model described in Figure 6.9, the supply and output voltage of the equipment can be visualized in the Figure 6.10.


Pressure Sensors:
Supply voltage: 24 VDC
Output voltage: (1-5) VDC

Water flow meter:
Supply voltage: 24 VDC
Output voltage: (1.5 - 5) VDC

Temperature Sensors:
Supply voltage: 24 VDC
Output voltage: (0 - 10) VDC

Manual valve

Solenoid valve:
Supply voltage: 24 VDC
Input voltage: (0-10) VDC

Figure 6.10: Water circuit with associated equipment and technical specifications.

The following section will present the methodology proposed to develop the program implemented in the μC.

6.8 Program implemented in Arduino IDE

In order to perform hardware-in-the-loop simulations with a μC and to control the real system, an Arduino Mega 2560 was used. Integrated to this controller board is the ATmega2560 processor which has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator and a USB connection [66]. There is no much difference between Arduino Uno and Arduino Mega except the size and number of I/O pins. Although, this may prove to be very helpful since there are several types of equipment that need to be connected to this board, which make the connections less complicated.

The methodology adopted to attach the measuring equipments with the μC is exactly the same as in case study 1. The temperature sensors, pressure sensors, and the water flow meter will all be connected to analog pins of the Arduino. The water flow rate will be controlled by the solenoid valve through the signals sent from the digital pins of the μC.

The structure of the source code is identical to the program implemented in case study 1. The first section is where the program variables are declared, the second section is defined by the initial setups and communication. Finally, the third section is dedicated to the main loop where the fundamental code of the program's execution is placed.

The temperature and water flow rate sent from the virtual model are interpreted by the Arduino through analog inputs, and with function 'analogRead' the signals are then processed and converted to be used later on. To determine the output power of the controller several calculations need to be implemented. First, temperature and water flow rate read from the analog pins need to be converted to an appropriate scale for this purpose; the result obtained from the previous function is multiplied by the maximum range and then divided by 1023, because an analog input of the Arduino is an 8bit analog to digital converter, the "8bit" portion allows to know the actual resolution, and with the equation 2^n where "n" is the number of slots, $2^{10} = 1024$. It is important to mention that 1024 represents the number of bits, but in calculations it is only used 1023 because zero is counted as the first number. Second, mathematical expressions are applied to achieve theoretical and effective power with the implementation of saturation to limit the desired power. Third, an error is determined which consists of the difference between theoretical and effective power, that corresponds to the PID controller input. Fourth consists of calculating the PID terms, wherein the end these are added and saturation is implemented. Finally, the PID output is added with the result obtained from the theoretical power, this is also saturated and multiplied by 255. This number refers to the same reason as mentioned before, except the number of bits of the output pins is 256 (2^8).

It is important to state that before sending the signal from the Arduino to the dSPACE connector panel, an RC filter was implemented. It is a simple circuit element composed by two resistors and a capacitor used to convert a pulsed signal to a flatter signal, in which the essential concept is that the capacitor shunts the fluctuating voltage to ground, producing a DC voltage.

To approximate the virtual controller developed in Simulink, the output power is sent each 100 ms.

6.9 General experimental procedure

Based on case study 1, the steps to the experimental procedure will be fulfilled with the same principle.

The first step is to parameterize the system's model with open-loop experiences. Second, it will be to simulate the system in real-time with a commercial digital signal processing (DSP) controller board dSPACE DS1104, mentioned in Section 4.4. The next stages are to perform experiences mentioned Section 3.2 with the features of the virtual test bench.

The following procedures can be used to any of these methods, the virtual system is controlled by the μC, the virtual controller controls the real system and for last is the purely real environment, where the real system is controlled by the real controller.

To control the virtual model of the system, a real-time simulation platform will be created. As mentioned before, the controller board is supported by MathWorks Real-Time Workshop, and compiles the Simulink in C code, which optimizes a real-time execution on the board.

The dSPACE Real-Time Interface connects the Matlab, Simulink and Real-Time Workshop with dSPACE real-time system, to form an integrated ready-to-use development environment for real-time applications.

As mentioned, the hardware used to implement the controller proposed in this case study is the Arduino Mega board, which specifications are referenced in the previous section.

The hardware and software HIL platform schematics is presented in Figure 6.11. The dynamic model system implement in Matlab/Simulink is compiled and downloaded to the dSPACE DS1104 board.

The dSPACE ControlDesk software, Appendix F, is used as an interface with real-time simulation on the DS1104 board, allowing data acquisition, graphics visualization and interaction with model parameters.

The diagnostic of Arduino program is implemented with serial communication to the PC.

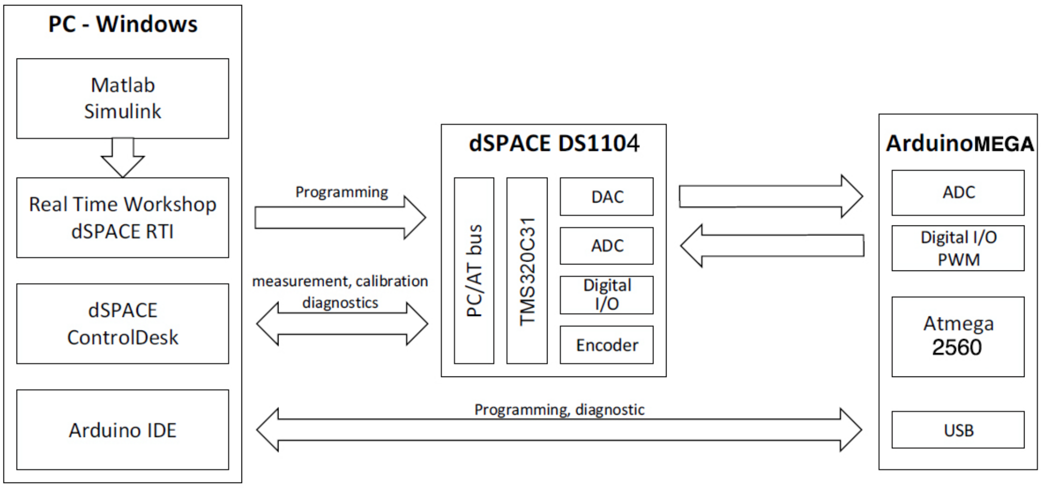


Figure 6.11: Hardware and software HIL simulation framework (Adapted from [68]).

Based on the two case studies developed, it is possible to define a sequence of events to a general experimental procedure that can be applied to any water heating device coupled to the virtual test bench. Therefore, the chronological course of events can be:

1. Data acquisition (open-loop experiences);
2. Parametrization of the system;
3. Virtual simulation in real-time;
4. Virtual model controlled by the μC;
5. Virtual controller controls the real system;
6. Real controller controls the real system.

In order to demonstrate the validity of these events, the several experiments will be presented in the next two sections.

6.10 Application of the operating modes

As mentioned before the experimental platform was not fully built, so only some experiences were performed, which implies that a few parameters of the mathematical models have yet to be discovered, so the unknown values are based on work reported in [68] in order to advance in the experiences. This allowed validating the dynamic model developed in MATLAB/Simulink and to test the program implemented in the μC.

Virtual simulation in real-time

The first stage is to model the virtual system in Simulink (Appendix E). The dynamic models are developed by applying the thermal dynamic equations, and afterward design the controllers. Through the virtual system, the behavior of the controller can be monitored, and the parameters can be tuned. With the optimum parameters of the controller, the virtual simulation in real-time is performed and it is shown in Figure 6.12.

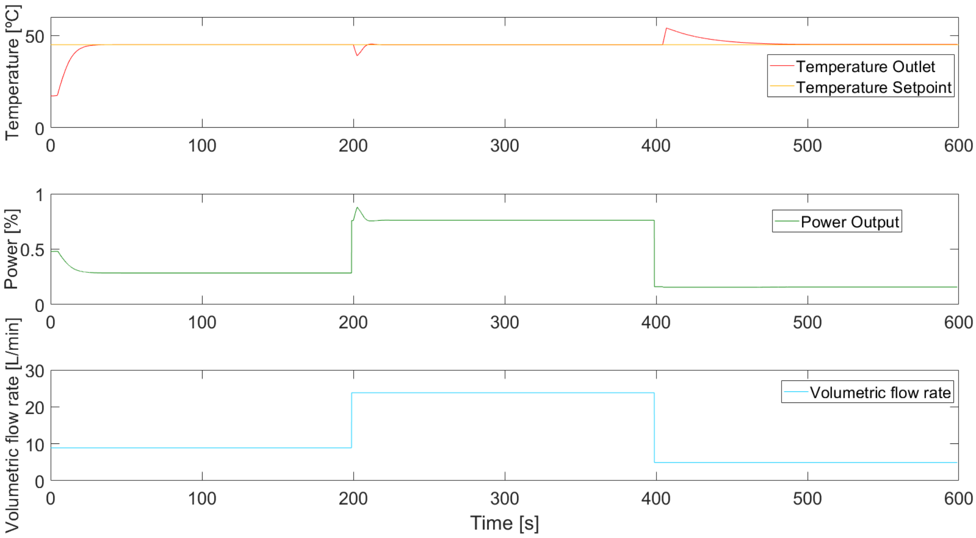


Figure 6.12: Virtual simulation in real-time.

Hardware-in-the-loop simulation test with the microcontroller

As mentioned before, there is no need to parameterize mathematical models of the thermal devices, because these have already been determined; all that is left is to proceed with the experiment, includes programming the Arduino. It is intended that the principle behind the controller implemented in Simulink to be very identical to the controller board.

With the controller programmed, it is possible to control the virtual model using a real controller; in other words, to perform HIL simulations.

In Figure 6.13 it is possible to examine the response of the controller towards the virtual model and in Appendix E is shown the Simulink model with the appropriate block diagrams.

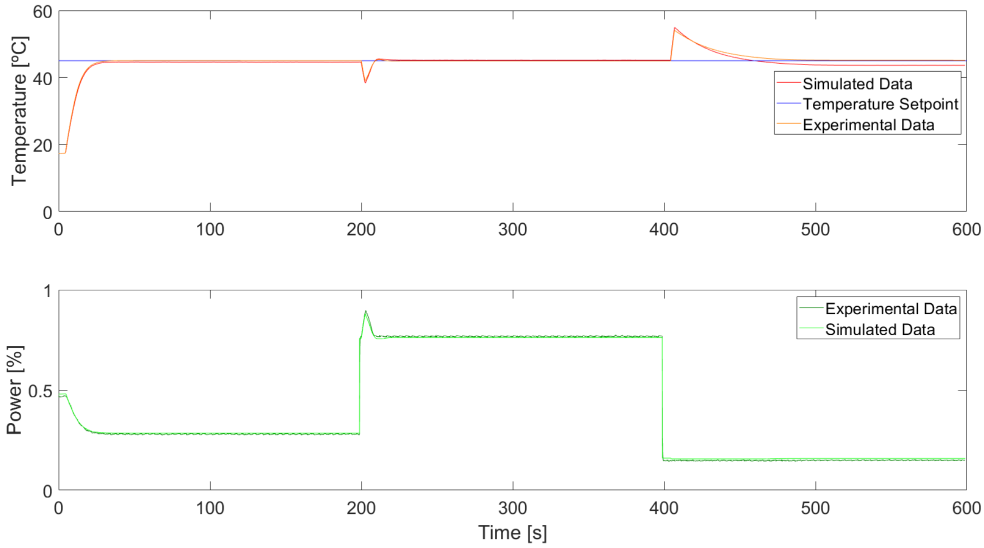


Figure 6.13: Hardware-in-the-loop simulation with real controller.

Real system controlled by the virtual controller

The next step will be to control a real system with the virtual controller. This leads to several advantages, such as enabling the parameterization of the water heating dynamic models, bringing them closer to the performance of a physical system in order to further improve the controller's design.

Additionally, when the virtual controller is able to control the water heating device, it can be optimized by constantly examining the real system behavior and testing several control parameters. The expected result is that the real system has a response when controlled by the virtual controller similar to that in a completely virtual environment.

Since it is not possible to access the TGWH controller inputs, or to replace it with the Arduino due to the system's complexity with all the integrated sensors, this type of experiences will not be carried out.

In the following section are some modes that were not possible to perform relative to the present section due to lack of information and resources.

6.11 Second group of tests

Similar to the first case study and based on the general experimental procedure, the experiences with the several features of the virtual test bench will be carried out, except for the new parameters that will be determined and used to perform the experiences.

Data acquisition and model parameterization

As a result of what was stated in Section 6.5, regarding the model parameterization, this method intends to perform open-loop experiences in order to parameterize the missing parameters. Based on the energy conservation Equation 6.6 and focusing on the constant terms ($m_wc_w + m_mc_m$), it is correct to assume that these variables are constant, so for simplification these parameters are considered as one term. The second and third unknown variables are the temperature increment at the inlet and outlet of the heat exchanger and the power delay that the heat cell delivers. To simplify further processing, these temperature increments were considered to have the same value.

In this method, it is also necessary to adapt the measuring device signals in a simulated environment to monitor the correct value of the output signal of the equipment. To interact with the electronic control unit of the TGWH, a program was provided by Bosch TT.

Every equipment is set, and the software is ready to be initiated, all that is left is to carry out the experiment. The first stage is to set several values of power (32%, 50% and 100%) with a constant hot water flow rate (Figure 6.14). The second stage consists of defining the minimum power of the heat exchanger and set four different values of water flow rate (2.94, 4.34, 5.81 and 6.89) through the manual valve (Figure 6.15). It was possible to perform both experiences with the help of the program provided by Bosch TT mentioned earlier.

According to the technical specifications of this TGWH, the thermal flow rate is 22 kW and an 86% thermal efficiency, this means that the maximum effective power of the heat exchanger is 18.9 kW. Although, trough open loop experiences the maximum effective power of the heat exchanger does not correspond to this value. By Equation 6.6 it is possible to determine the effective power at a steady-state regime, since the slope is null which means that the temperature variation is null and the equation can be simplified to $\dot{Q} = -\dot{m}c_w(T_{in} - T)$. At 100% of the maximum effective power, the average temperature is 56.5 °C and as the other parameters are already known it is possible to obtain an effective power of 18.115 kW. Following the same methodology at 50% an effective power of 11.037 kW was obtained, and for 32% a 7.323 kW effective power was obtained. It is important to note that during the several experiences the only parameter to be changed was the power output from the heat exchanger, the hot water flow rate remaining unchanged.

Using the very same methodology as in Section 5.3 and using the optimized method it was possible to obtain the following new parameters of the heat cell model. With this, it was possible to obtain the following parameters, C1 = 2218.1, ΔT = 0 s and Δt = 2.9188 s, where C1 is the couple of constants mentioned previously, Δt is the temperature increment and P_{delay} the power delay. The inlet temperature is also different based on the values read from the sensor placed at the inlet of the TGWH, and it has the value of 19.6 °C.

It is important to mention that the optimized method was applied in both experimental results since it is not sufficient only to test different stages of power because it does not directly apply in reality. Therefore, a second simulation was carried out to examine if the obtained parameters were satisfactory. There is also a difference between the two simulations, wherein the first one the cost function is determined by comparing the experimental data with all the simulated values, whereas in the second simulation the cost function is calculated from the average flow rates in a steady-state situation.

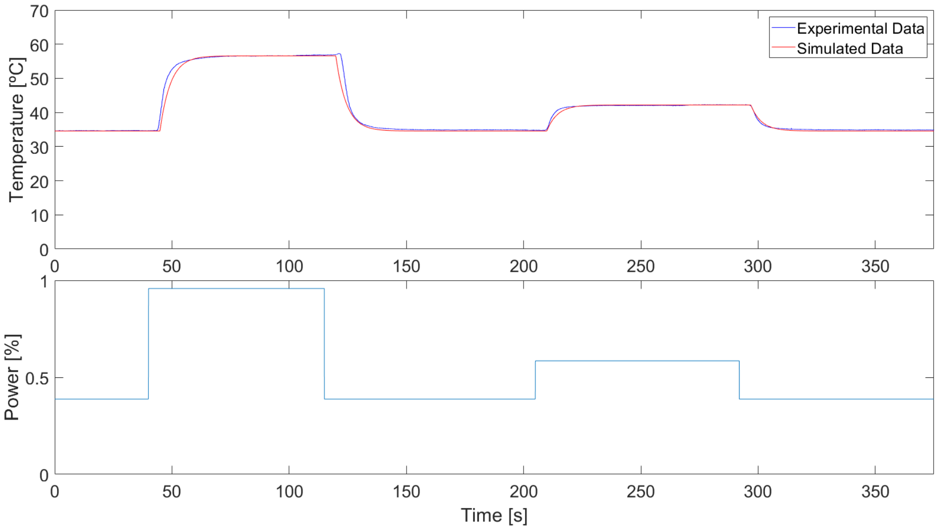


Figure 6.14: Experimental and simulated data of the hot water outlet temperature and flow rate with the TGWH minimum power.

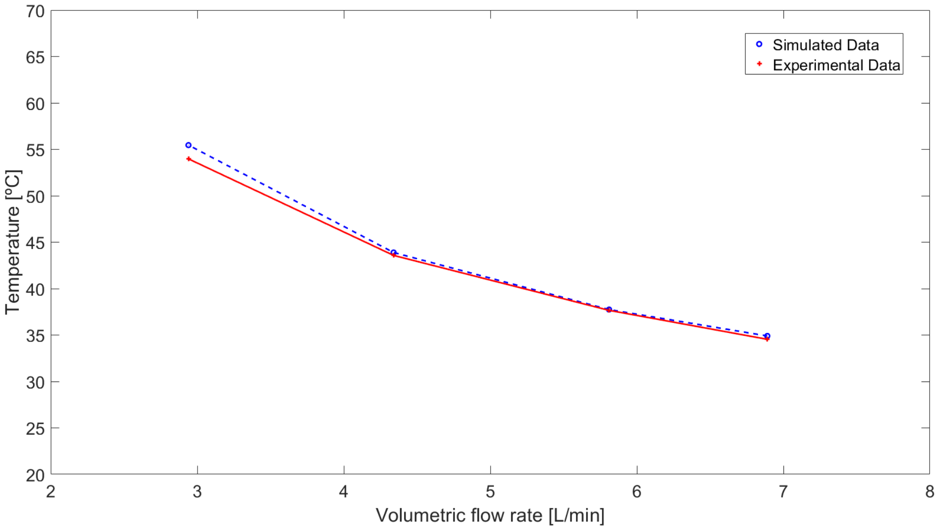


Figure 6.15: Experimental and simulated data of the hot water outlet temperature and water flow rate with different powers.

The correlation between the outlet temperature, volumetric water flow rate and power can be visualized in Figure 6.16.

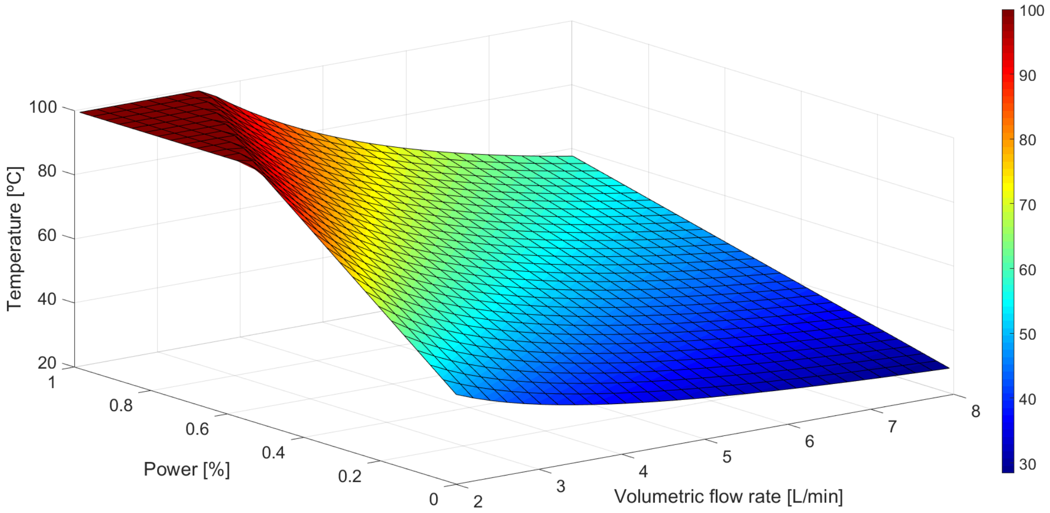


Figure 6.16: Experimental and simulated data of the hot water outlet temperature and power with different values of water flow rate.

Virtual simulation in real-time

Similar to the first group of tests, the virtual simulation in real-time was carried out, with the exception of using new parameters indicated to this water heating device and PID terms. Because the mathematical model values have been changed, this means that the controller terms will also need to be adjusted. By following the same process described in Section 6.6 it was possible to obtain the corresponding proportional, integral and derivative terms of 0.02, 0.012 and 0.01.

With the optimum values applied to the mathematical models and the corrected terms of the controller, the experience is performed which, is shown in Figure 6.17.

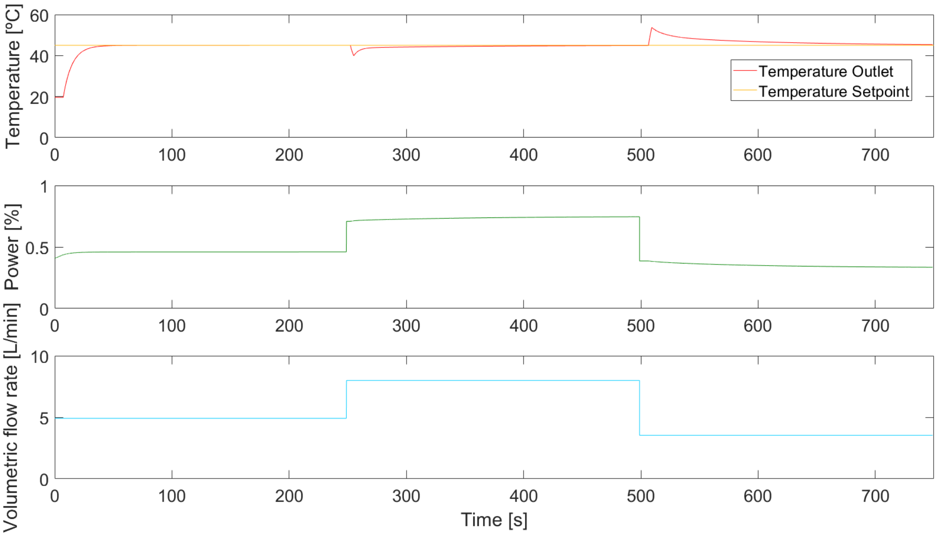


Figure 6.17: Virtual simulation in real-time with the desired parameters.

Hardware-in-the-loop simulation test with the microcontroller

After parameterizing the virtual model and optimizing the water heater controller, it is possible to advance to another phase of development, which includes programming the control board. The source code implemented is the same as in the first group of tests.

With the parameters set, it is possible to perform the hardware-in-the-loop simulation. Now that the model has been parameterized correctly, and already emulates the performance of a real water heater device, it is possible to test if the real controller is well programmed and if the response of the system agrees with the previous simulations.

Figure 6.18 represents the controller's performance towards the virtual model with the desired parameters of the water heater device used in this work. The block diagram implemented in Simulink is the same as that of the first group of tests and can be viewed in Appendix E.

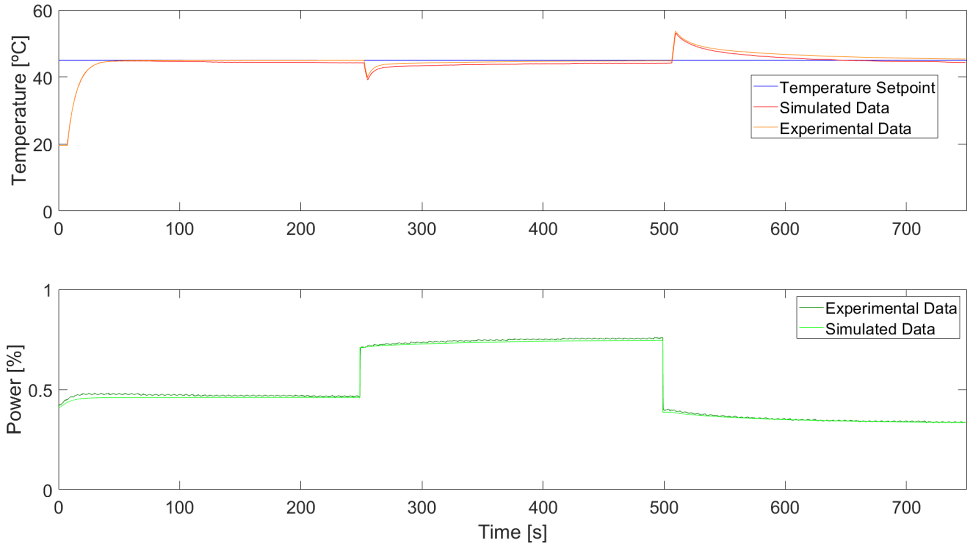


Figure 6.18: Hardware-in-the-loop simulation with real controller and proper parameters.

It is important to note that the experimental results acquired in this experiment relate to the controller's response, in this case, the Arduino. Thus, the experimental data were obtained through the acquisition of signals from the dSPACE connector panel.

TGWH controlled by the electronic control unit

With the approved control board, through the data of the virtual model, it is possible to move on to the last test phase where only real components are involved. As in case study 1, in this last part it is possible to verify the performance of the TGWH when controlled by the electronic control unit, and verify if the whole process is in proper operation (Figure 6.19). This experience is dedicated solely to test the response of the TGWH controller board faced to sudden changes in water flow.

The model developed in Simulink (Appendix E) is the same one that was used in data acquisition because this experience also has this purpose.

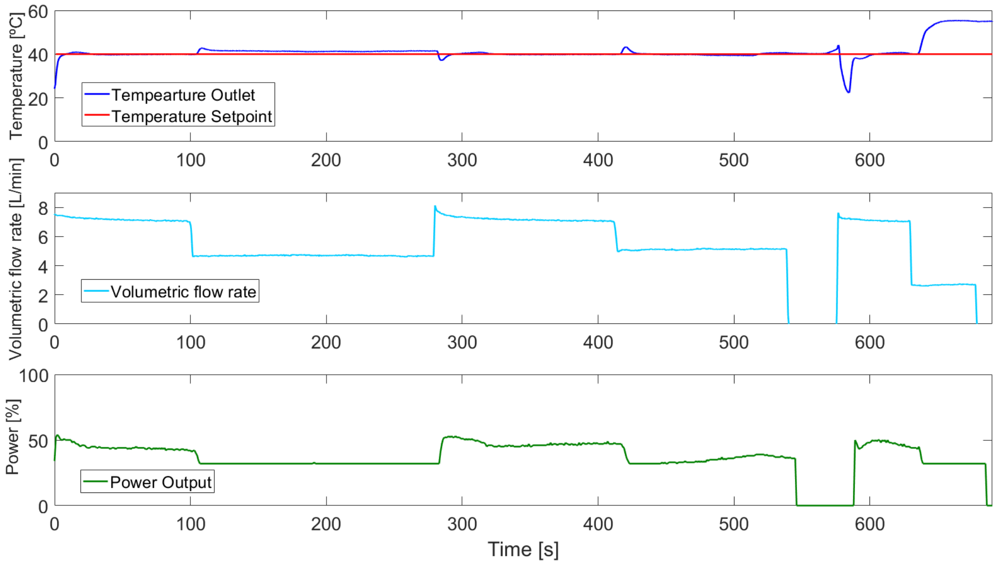


Figure 6.19: TGWH controlled by the ECU implemented in the device experience.

6.12 Analysis of the experimental results

As previously mentioned, the parameters of the TGWH model of this work have not yet been parameterized by the experiments in open loop, and parameterization of the model is skipped and the virtual simulation in real-time is carried out. By interpreting Figure 6.12, the virtual controller's performance is considered to be very satisfactory since there are no oscillations and has a fast response to the desired setpoint. It can also be visualized that the outlet steady-state temperature is almost overlapping the setpoint, which means that the error is very low.

Next is the HIL simulation, where the virtual model is controlled by the μC, which also presented very positive responses. When the experimental data are compared with the virtual simulation in real-time (Figure 6.13) there is a slight discrepancy in the outlet temperature registered. Although in the power output some disturbances can be visualized, this may be due to the electromagnetic noise when the Arduino acquires the dSPACE connector panel signal and to the RC filter, which can be further improved. These disturbances were also verified in the same type of experiences related to the first case study, but in this case better results were obtained since the program cycle can be visualized through the oscillations; with a better filter the output signal will also improve.

The second group of tests starts with open loop experiences and data acquisition in order to determine the TGWH desired parameters. After, simulations are carried out to examine the system's response. By analyzing Figure 6.14, it presents very positive results through the optimized method since both experimental and simulated slopes are very close. Interpreting Figure 6.15, where the same methodology of optimization is used, it was possible to obtain very satisfactory results with the new parameters. By executing the Simulink model with different values of hot water flow and the achieved parameters, when compared to the experimental data the outlet temperature is very close.

Similar to the first group of tests, where the hardware-in-the-loop simulation is carried out, positive outcomes were achieved. By comparing the experimental data with the virtual simulation in real-time (Figure 6.18) there is almost no difference in the outlet temperature registered. Although in the power output some disturbances can be visualized, this may be due to the noise when the Arduino acquires the signal from the connector panel and the RC filter.

Lastly, the TGWH is controlled by the electronic control unit. This experience is devoted to test the performance of the controller board integrated into the water heating device when faced with hot water flow rate changes. This may be due primarily to changes in the number of users requiring hot water, or changes in a user's demand for hot water flow. Even the TGWH with hot water outlet temperature feedback control is not able to reduce variations on outlet temperature to acceptable values when fast transitions on the hot water flow rate occur. This is due to the relatively slow dynamics of the heat exchanger, and the time delays on temperature changes in the outlet hot water, depending heavily on the hot water flow and on the length of the pipes where it is flowing. The above-described problem can be visualized in Figure 6.19, which shows the temperature overshoots and undershoots for unexpected changes on the hot water flow rate. The controller implemented in this equipment intends to stabilize the outlet temperature as 40 °C but, with the unpredictable changes on the water flow rate, the temperature overshoots and undershoots achieve magnitudes which are not acceptable and comfortable for the users.

Chapter 7

Final conclusions

The work presented in this dissertation had the objective of developing a concept of a test bench, where it allows not only to apply HIL simulations, but also becomes flexible to test different forms of simulation that may be beneficial in the process of developing and testing controllers. Additionally, it was also intended to model dynamic thermal systems in MATLAB/Simulink and customize them to the different modes of operation, implementing software and hardware required to perform hardware-in-the-loop simulations dedicated to the test of controllers, and build an experimental platform which enables several types of experiences. Therefore, two case studies were carried out: the first one was to validate the concept of the virtual test bench, and the second one applies to a TGWH.

The use of dynamic models is extremely important because they allow predicting the equipment performance, which is an advantage in terms of safety, savings, and investment. Despite the initial setup used to model the system effectively, this offers a long-term advantage for the reasons mentioned before.

The proposed simulation platform allows the development of advanced control strategies in a simulated environment, taking advantage of the hardware-in-the-loop simulation techniques, without the need for a physical model and avoiding the associated safety risks.

Looking at the test benches of the market, especially those dedicated to testing thermal systems, there is a great limitation regarding the flexibility and adaptation to several types of simulation. Their majority allows testing only at the final phase of development, which increases the uncertainties of the product at an early stage of development. There are some benches that allow hardware-in-the-loop simulations, but very few references tp their application to thermal systems.

The proposed virtual test bench concept, allows assisting the development process at an early stage, where it is possible to test the controllers directly on the real system. This offers a tremendous advantage since even before building and programming the controller board, a model control can already be tested and assessed. Additionally, the testing of μCs in the virtual system delivers some safety, which makes it possible to detect any anomalies in the program without risks. Finally, the concept also allows testing the prototypes in the final phase, with the controller already programmed and with the real system assembled.

Through case study 1 it was possible to demonstrate that the proposed layout concept works as long as the Simulink adaptations are made, and the signal conditioning modules necessary for the systems to interact with each other are considered. The experimental results were very satisfactory even with the disturbances and oscillations caused by the equipments, which is normal since they are not easy to control.

Regarding the experiences performed in case study 1, which was to validate the bench concept, this proved to approximate the virtual controller to the μC as long as the control parameters are similar. The evidence is on the tests performed by both controllers in the virtual system (Section 5.9 second and third operating modes), where a good agreement was reached, not only from the virtual simulation but also with each controller test.

Observing the test performed on the real system, both with the virtual controller and Arduino (Section 5.9 fourth and fifth operating modes), this was further to support the idea that it is possible, in the design phase of the controller in Simulink, to predict the behavior that the system will have when operating through the real controller. Very satisfactory results were also obtained in both methods.

Case study 2 was intended to explore almost all the possibilities that the virtual bench enables when applied to a TGWH. The objective was to support the concept that creating the suitable interface modules it is possible to implement the concept presented to any system, whether it is simple or complex. Obviously, the complexity of the system also increases the work of adapting the models and the signals. As the complexity of the system increases, so does increase the mathematical models to be developed. With a higher number of types of equipment, the need to adapt the output voltage also increases in complexity, since there is no guarantee that the different equipments operates with the same output voltage range.

The results attained in Sections 6.10 and 6.11 demonstrate the concept of the virtual test bench in practice to an actual water heating device. Among the several experiences carried out, it was possible to achieve very satisfactory results.

The main considerations to take into account is that it is possible to perform the different modes of operation, including hardware-in-the-loop simulation, in this virtual test bench as long as the necessary interface modules between the different signals of the components are developed. Furthermore, there must be a concern for the models to be adjusted in order to reproduce the signals that come from various equipments and interpret them. The virtual controller should also be brought closer to the real controller in the optimal form, to achieve better results.

This case study 2 enabled to develop a virtual test bench concept that overcomes some of the limitations of the existing test benches, which also provides a larger selection of test types, including hardware-in-the-loop simulations in thermal systems.

In both case studies, the optimized method proved to be very advantageous and practical after the open loop experiences were performed, so that it was possible to determine the model's parameters. With the proper conditions, an estimate of the initial parameters, and experimental data and Simulink models, any unknown parameters can be discovered.

The objective was achieved by developing a test bench concept with the requirements mentioned, validating it through two case studies and consecutive laboratory experiences.

7.1 Future work

The virtual test bench concept was proven through a simple case study, and there was the possibility of advancing to a large scale project. In this sense, a proposal for future work would be to carry out the experiments that were not possible to be performed, such as controlling the real system with the virtual controller and to control the real system with a μC.

Additionally, it would be interesting to develop new mathematical and semi-empirical models that enable the study of different configurations for water circuits, in order to improve user comfort. Also, develop different models of new components of a water heating device, with the objective of saving water when starting the equipment (water loses during the waiting time).

Finally, there is the possibility of developing and testing new controllers that allow using new control algorithms, including predictive and adaptive control.

References

[1] Y. Jaluria, *Design and Optimization of Thermal Systems*, 2nd ed., L. Faulkner, Ed. Boca Raton: CRC Press/Taylor & Francis Group, 2008.

[2] C. Köhler, "Enhancing Embedded Systems Simulation," Ph.D. dissertation, Technische Universität München, 2010.

[3] K. Mahapatra and L. Gustavsson, "Innovative approaches to domestic heating: homeowners' perceptions and factors influencing their choice of heating system," *International Journal of Consumer Studies*, vol. 32, no. 1, pp. 75–87, 2008.

[4] V. Costa, J. Ferreira, and D. Guilherme, "Modeling and simulation of tankless gas water heaters to reduce temperature overshoots and undershoots," *12th International Conference on Heat Transfer, Fluid Mechanics and Thermodynamics (HEFAT 2016)*, pp. 1404–1409, 2016.

[5] Y. Cengel and M. Boles, *Thermodynamics: An Engineering Approach*, 5th ed. Boston: McGraw-Hill Science/Engineering/Math, 2004.

[6] D. Bacon, *Basic Heat Transfer.* London: Butterworth & Co., 1989.

[7] B. T. Kulakowski, J. F. Gardner, and J. L. Shearer, *Dynamic Modeling and Control of Engineering Systems*, 3rd ed. New York: Cambridge University Press, 2007.

[8] M. A. Johnson and M. H., *PID Control: New Identification and Design Methods*, M. A. Johnson and M. H. Moradi, Eds. Springer-Verlag London Limited, 2005.

[9] L. Payne, "The Modern Industrial Workhorse: PID Controllers," 2014. [Online]. Available: https://www.techbriefs.com/component/content/article/tb/features/articles/20013

[10] S. Temel, S. Yagli, and S. Gören, "P, PD, PI, PID Controllers," Middle East Technical University, Tech. Rep., 2012.

[11] J. E. Normey-Rico, I. Alcalá, J. Gómez-Ortega, and E. F. Camacho, "Mobile robot path tracking using a robust PID controller," *Control Engineering Practice*, vol. 9, no. 11, pp. 1209–1214, 2001.

[12] T. Kavita, B. Parvathi, M. Lavanya, and M. Arivalagan, "Temperature Control Water bath system using PID Controller," *International Journal of Applied Engineering Research*, vol. 10, no. 4, pp. 3443–3446, 2015.

[13] L. Alonso, J. Perez-Oria, B. M. Al-Hadithi, and A. Jimenez, “Self-tuning PID controller for autonomous car tracking in urban traffic,” *2013 17th International Conference on System Theory, Control and Computing, ICSTCC 2013; Joint Conference of SINTES 2013, SACCS 2013, SIMSIS 2013 - Proceedings*, pp. 15–20, 2013.

[14] H. M. Li, X. B. Wang, S. B. Song, and H. Li, “Vehicle control strategies analysis based on PID and fuzzy logic control,” *Procedia Engineering*, vol. 137, pp. 234–243, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S187770581600268X

[15] J. A. Ledin, “Hardware-in-the-Loop Simulation,” *Embedded Systems Programming*, pp. 42–60, 1999.

[16] R. Isermann, J. Schaffnit, and S. Sinsel, “Hardware-in-the-loop simulation for the design and testing of engine-control systems,” *Control Engineering Practice*, vol. 7, no. 5, pp. 643–653, 1999.

[17] N.N., “Schulung und Praxis mit Flugsimulator,” vol. 19, pp. 1104–1107, 1953.

[18] H. Marienfeld, “Flugsimulation,” Ph.D. dissertation, TU Darmstadt, 1965.

[19] K. Anderson, C. V. Corp, and C. V. Corp, “A Design Tool (Research and Development) Simulator Survey Report A Design Tool (Research and Development) Simulator Survey Report,” *Society of Automotive Engineers, Inc*, pp. 17–20, 1962.

[20] A. Beevor, “Imperial war museums,” 2015. [Online]. Available: https://www.iwm.org.uk/

[21] G. A. Munoz-Hernandez, S. P. Mansoor, and D. I. Jones, “Hardware-in-the-loop simulation,” *Advances in Industrial Control*, pp. 3194–3198, 2005.

[22] J. Cole and A. Jolly, “Hardware-in-the-loop simulation at the US Army Missile Command,” *In Technologies For Synthetic Environments: Hardware-In-The-Loop Testing*, vol. 2741, pp. 14–19, 1996.

[23] M. E. Sisle and E. D. Mccarthy, “Hardware-in-the-loop simulation for an active missile,” *Simulation*, vol. 39, no. 5, pp. 159–167, 1982.

[24] H. Eguchi and T. Yamashita, “Benefits of HWIL simulation to develop guidance and control systems for missiles,” *In Technologies For Synthetic Environments: Hardware-In-The-Loop Testing V*, vol. 4027, pp. 66–73, 2000.

[25] M. Bailey and J. Doerr, “Contributions of hardware-in-the-loop simulations to Navy test and evaluation,” *Technologies For Synthetic Environments: Hardware-In-The-Loop Testing*, vol. 2741, pp. 33–43, 1996.

[26] M. B. Evans and L. J. Schilling, *The Role of Simulation in the Development and Flight Test of the HiMAT Vehicle*. California: NASA Ames Research Center, 1984. [Online]. Available: https://ntrs.nasa.gov/search.jsp?R=19840013469

[27] D. Johannes and P. Ferdinand, “The Daimler-Benz Driving Simulator A Tool for Vehicle Development,” *SAE Transactions*, vol. 94, pp. 981–997, 1985.

[28] T. Suetomi, A. Horiguchi, Y. Okamoto, and S. Hata, “The driving simulator with large amplitude motion system,” *SAE International*, vol. 100, pp. 94–101, 1991.

[29] H.-J. v. Thun, “Dynamischer verbrennungsmotorenprüfstand mit echtzeitsimulation des kraftfahrzeug-antriebsstranges,” *Automobiltech- nische ZeitschriJt - ATZ*, vol. 89, 1987.

[30] K. Pfeiffer, “Fahrsimulation eines Kraftfahrzeugs mit einem dynamischen Motorprüfstand,” Ph.D. dissertation, TU-Darmstadt, 1997.

[31] H. Fennel, S. Mahr, and R. Schleysing, “Transputer-Based real-time simulator - A high preformance tool for ABS and TCS development,” *SAE International*, pp. 55–73, 1992.

[32] L. Michales, “The use of a graphical modeling environment for real-time hardware-in-the-loop simulation of automotive ABS systems,” *SAE International*, pp. 29–32, 1993.

[33] U. Sailer and U. Essers, “Real-Time Simulation of Trucks for Hardware-in-the-Loop Applications,” *SAE Technical Paper Series*, vol. 1, no. 412, pp. 1–11, 1994.

[34] M. Suh, J. Chung, C. Seok, and Y. Kim, “Hardware-in-the-loop simulation for ABS based on PC,” *Int. J. of Vehicle Design*, vol. 24, pp. 157–170, 2000.

[35] T. Bach, H. Schmitt, W. Schwanke, and H.-J. Tumbrink, “Roadrunner - Real-time simulation in anti-lock brake system development,” *SAE Technical Paper Series*, pp. 1–8, 1995.

[36] J. Drosdol, W. Kading, and F. Panik, “The Daimler-Benz driving simulator, vehicle system dynamics: International Journal of Vehicle Mechanics and Mobility,” vol. 14, no. 1-3, pp. 86–90, 1985.

[37] D. J. Kempf, L. S. Bonderson, and L. I. Slafer, “Real time simulation for application to ABS development,” *SAE Technical Paper Series*, pp. 1–24, 1987.

[38] W. Huber, W.-D. Jonner, and H. Demel, “Simulation, performance and quality evaluation of ABS and ASR,” *SAE Technical Paper Series*, pp. 33–38, 1988.

[39] L. A. dos Santos and N. M. F. de Oliveira, “Sistema hardware-in-the-loop para teste de determinação de atitude de um piloto automático,” *XVII Simpósio de Pesquisa Operacional e Logística da Marinha*, vol. 1, no. 1, pp. 1–9, 2014.

[40] J. R. Wagner and J. Furry, “A real time simulation environment for the verification of automotive electronic controller software,” *International Journal of Vehicle Design*, vol. 13, no. 4, pp. 365–377, 1992.

[41] H. Hanselmann, “Hardware-in-the loop simulation as a standard approach for development, customization, and production test of ECU’s,” *SAE Technical Paper Series*, pp. 1–11, 1993.

[42] P. Schaefer, “Echtzeitsimulation aktiver Mehrkörpersysteme auf Transputernetzen,” Ph.D. dissertation, University of Stuttgart, 1994.

[43] R. K. and S. W., “Echtzeitsimulation eines Fahrzeugmodells mit aktiver Federung,” *VDI-Bericht*, vol. 1189, pp. 17–34, 1995.

[44] W. Kortüm, R. Sharp, and A. Departer, “Review of multibody computer, codes for vehicle system dynamics,” *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility*, vol. 22:S1, pp. 3–31, 1993. [Online]. Available: http://dx.doi.org/10.1080/00423119308969463

[45] S. U. and E. U., “Parallelverarbeitung als weg zur echtzeitsimulation in der kfz-steuergeräteentwicklung,” *VDI-Bericht*, no. 1283, pp. 99–116, 1996.

[46] S. U., *Nutzfahrzeug-echtzeitsimulation auf parallelrechnern mit hardware-in-the-loop*. Renningen-Malmsheim, Germany: Expert-Verlag, 1996.

[47] S. W. and D. W., “Ermittlung des stationären und des instationären Betriebsverhaltens von kleinen schnellaufenden Dieselmotoren,” *Forschungsberichte Verbrennungskraftmaschinen*, 1990.

[48] C. Savaglio, “Hardware-in-the-loop simulation - An engine controller implementation,” *SAE Technical Paper Series*, pp. 1–10, 1993.

[49] W. R., “Ein Beitrag zur Echtzeitsimulation technischer Systeme hoher Dynamik mit diskreten Modellen,” Ph.D. dissertation, Universität Gesamthochschule Kassel, 1994.

[50] A. Kimura and I. Maeda, “Development of Engine Control System using Real Time Simulator,” pp. 157–163, 1996.

[51] R. Isermann, “Mechatronic systems - A challenge for control engineering,” vol. 1, pp. 2617–2632, 1997.

[52] J. Ferreira, “Modelação de Sistemas Hidráulicos para Simulação com Hardware-in-the-loop,” Ph.D. dissertation, Universidade de Aveiro, 2003.

[53] M. A. A. Sanvido, “Hardware-in-the-loop simulation framework,” Ph.D. dissertation, Swiss Federal Institute of Technology (ETH), 2002.

[54] H. K. Fathy, Z. S. Filipi, J. Hagena, and J. L. Stein, “Review of hardware-in-the-loop simulation and its prospects in the automotive area,” *Modeling and Simulation for Military Applications*, vol. 6228, pp. 1–20, 2006.

[55] A. M., “PID control,” Ph.D. dissertation, Kyoto University, 2011.

[56] E. ÁGÚSTSSON, “Hardware-in-The-Loop Simulation of a High Speed Servo System,” Ph.D. dissertation, Stockholm University, 2013.

[57] A. T. D. L. Cruz, P. Riviere, D. Marchio, O. Cauret, and A. Milu, “Hardware in the loop test bench using Modelica : A platform to test and improve the control of heating systems,” *Applied Energy*, vol. 188, pp. 107–120, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.apenergy.2016.11.092

[58] P. M. P. Semblano, “Desenvolvimento de Bancas de Ensaio,” Relatório de Estágio, Faculdade de Engenharia da Universidade do Porto, Tech. Rep., 2006.

[59] R. M. B. R. Pilão, “Estudo do Comportamento Térmico de Esquentadores Domésticos a Gás,” Ph.D. dissertation, Faculdade de Engenharia da Universidade do Porto, 1996.

[60] T. Lichius, C. Gross-Weege, D. Abel, and S. Baltzer, “Control design for thermal hardware-in-the-loop test bench for automobile thermal managment systems,” *IFAC-PapersOnLine*, vol. 48, no. 15, pp. 441–447, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.ifacol.2015.10.063

[61] M. A. S. Pêgo, “Desenvolvimento de Plataforma para Teste de Estratégias de Controlo de Equipamento para Aquecimento de Águas Domésticas,” Ph.D. dissertation, Universidade de Aveiro, 2016.

[62] Mathworks, “Simulink Coder.” [Online]. Available: https://www.mathworks.com/products/simulink-coder.html

[63] D. GmbH, “Real-Time Interface (RTI),” 2019. [Online]. Available: https://www.dspace.com/en/inc/home/products/sw/impsw/realtimeinterf.cfm

[64] S. Gmbh, *dSPACE Catalog 2019*, 2019.

[65] J. Purdum, *Beginning C for Arduino*, M. Lowman, M. Moodie, K. Sullivan, and M. Behr, Eds., 2012.

[66] “Arduino,” 2019. [Online]. Available: https://www.arduino.cc/

[67] “Commission regulation No. 814/2013 of 2 August 2013 implementing Directive 2009/125/EC of the European Parliament and of the Council with regard to ecodesign requirements for water heaters and hot water storage tanks,” 2013. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32013R0814

[68] A. F. Quintã, J. A. F. Ferreira, A. Ramos, N. A. D. Martins, and V. A. F. Costa, “Simulation models for tankless gas water heaters,” *Applied Thermal Engineering*, vol. 148, no. November 2018, pp. 944–952, 2019. [Online]. Available: https://doi.org/10.1016/j.applthermaleng.2018.11.095

[69] W. K. Wojsznis and T. Blevins, “Evolving PID tuning rules,” 2013. [Online]. Available: https://www.controleng.com/articles/evolving-pid-tuning-rules/

[70] J. Cahill, “PID Control History and Advancements,” 2013. [Online]. Available: https://www.emersonautomationexperts.com/2013/process-optimization/pid-control-history-and-advancements/

[71] H. A. Hazza, M. My, and M. C. Mahdi, “Performance of manual and auto-tuning PID controller for unstable plant - Nano satellite attitude control system,” *2018 6th International Conference on Cyber and IT Service Management (CITSM 2018)*, pp. 1–5, 2018.

Appendices

Appendix A

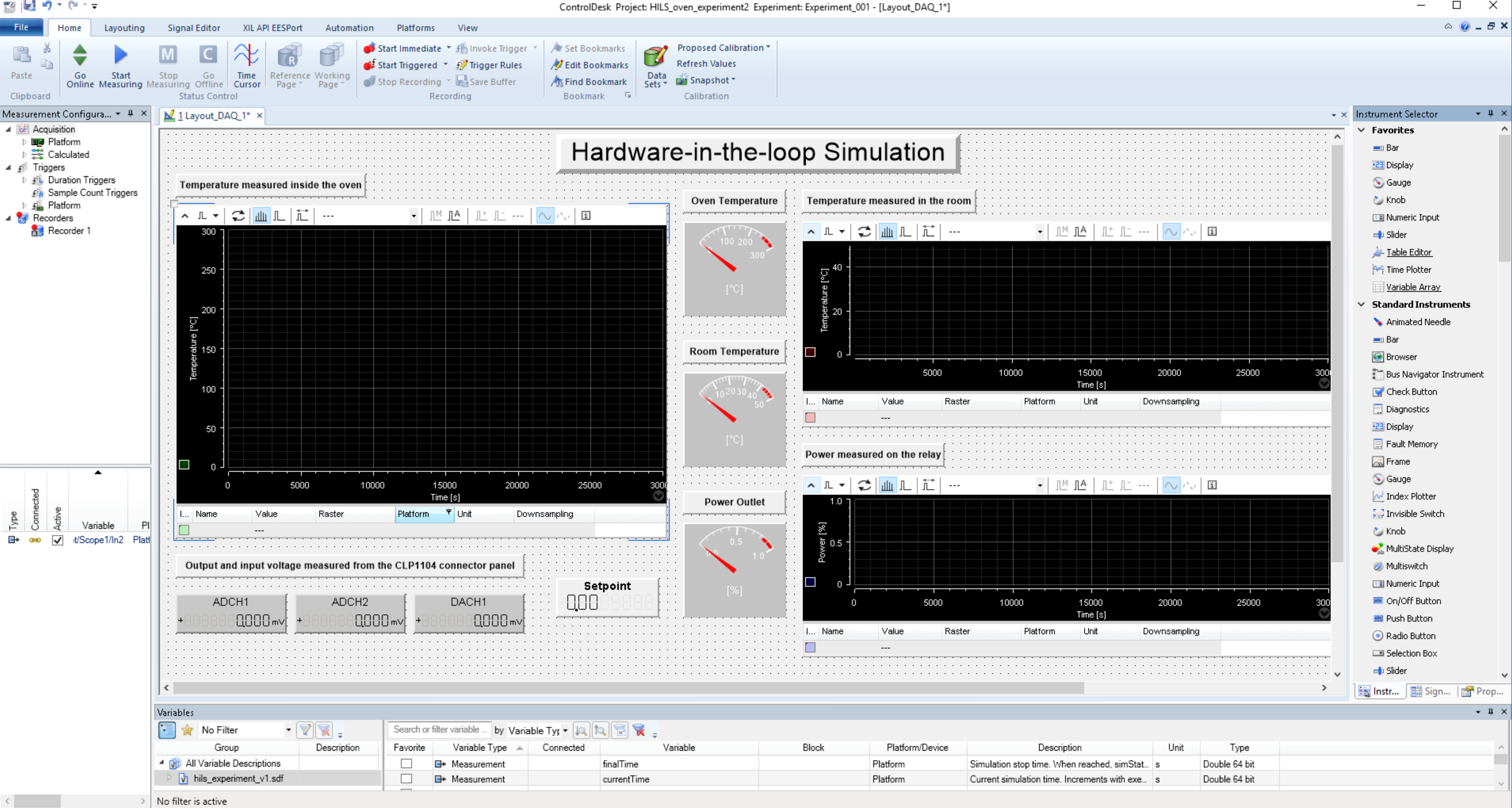
CLP1104 Connector and LED Panel specifications

Table A.1: CLP1104 Connector and LED Panel specifications (Adapted from [64]).

Board	DS1104	
BNC Connectors	- 8 ADC inputs - 8 DAC outputs	
Sub-D Connectors	- Digital I/O - Slave DSP I/O - Incremental encoder interfaces - Serial interfaces	
A/D Converter	Channels	- 4 multiplexed channels equipped with one sample & hold A/D converter (1x16-bit) - 4 parallel channels each equipped with one sample & hold A/D converter (4x12-bit) - 5 A/D converter channels (1x16-bit and 4x12-bit) can be sampled simultaneously
	Resolution	- Multiplexed channels: 16 bit - Parallel channels: 12 bit
	Input voltage range	- ±10V
D/A Converter	Channels	- 8 Channels
	Resolution	- 16-bit
	Output range	- ±10V

Appendix B

ControlDesk graphical interface implemented in case study 1



ControlDesk Project: HILS_oven_experiment2 Experiment: Experiment_001 - [Layout_DAQ_1*]

Type	Connected	Active	Variable	Pl
		☑	t/Scope1/In2	Platf

I...	Name	Value	Raster	Platform	Unit	Downsampling
🟩		---				

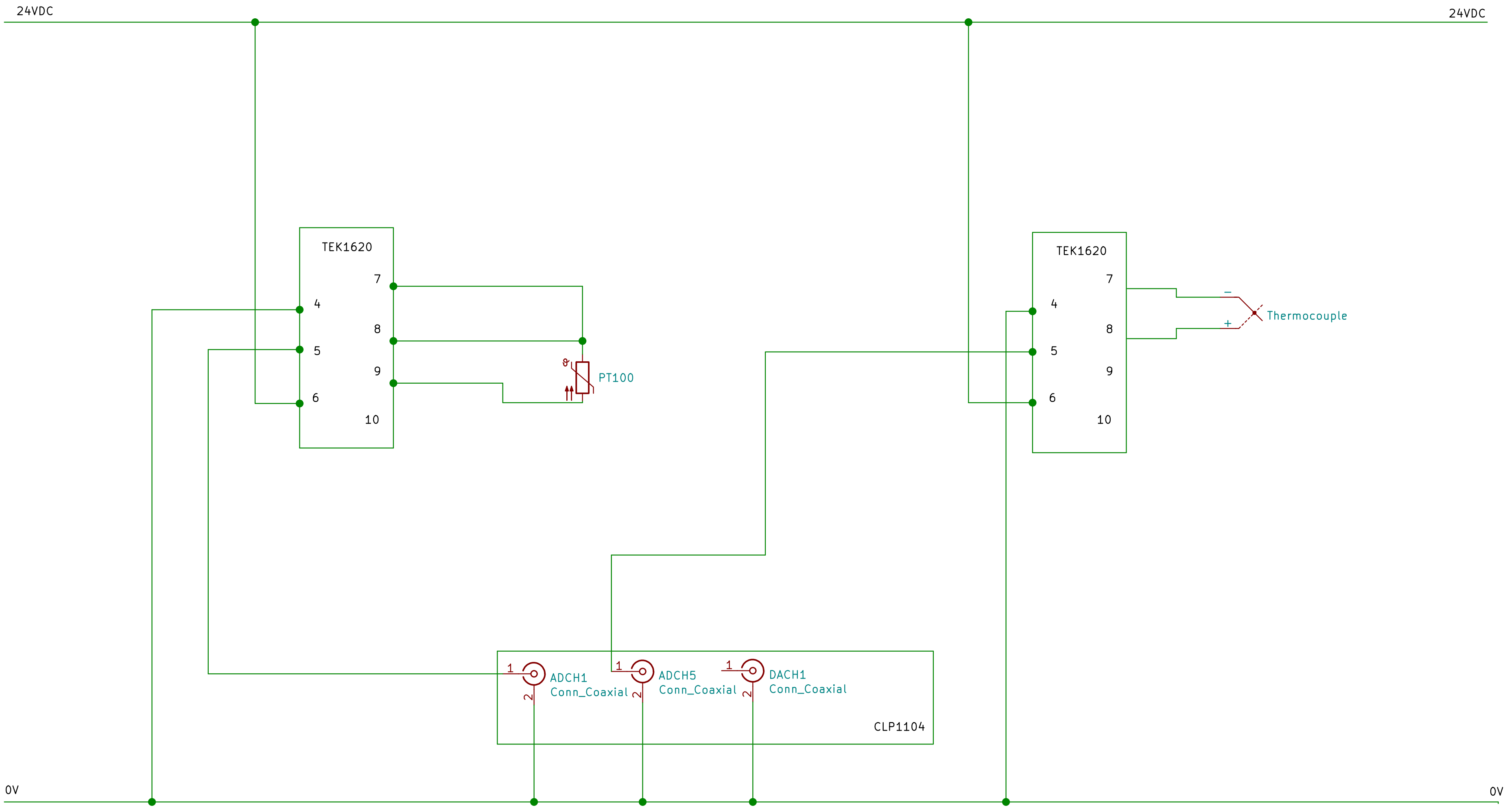
I...	Name	Value	Raster	Platform	Unit	Downsampling
🟥		---				

I...	Name	Value	Raster	Platform	Unit	Downsampling
🟪		---				

Favorite	Variable Type	Connected	Variable	Block	Platform/Device	Description	Unit	Type
□	Measurement		finalTime		Platform	Simulation stop time. When reached, simStat...	s	Double 64 bit
□	Measurement		currentTime		Platform	Current simulation time. Increments with exe...	s	Double 64 bit

Appendix C

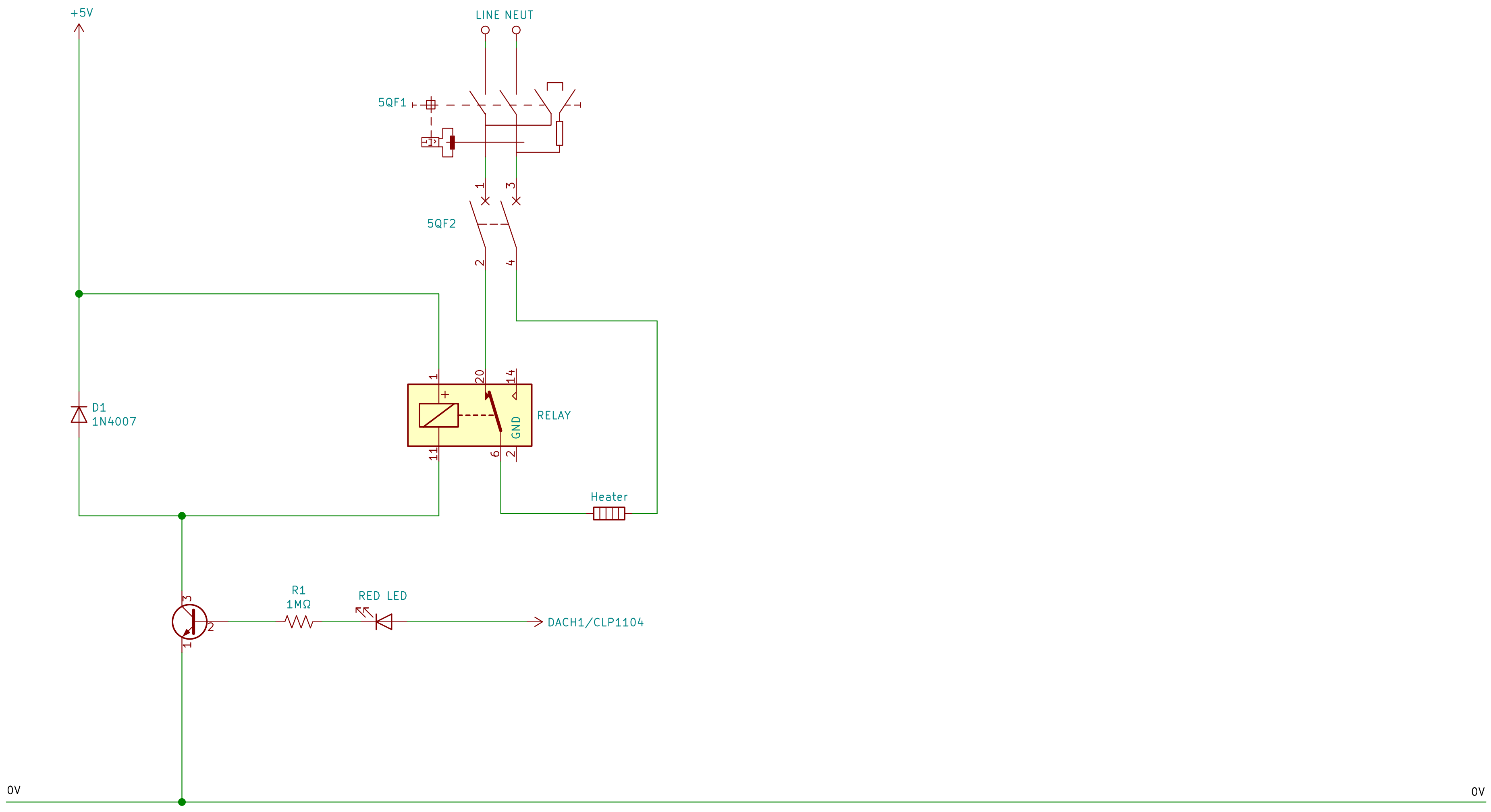
Representation of the electrical connections in case study 1



Sheet: /
File: Virtual controller and real model.sch

Title: DAQ/Rmdl and Vctr schematics (VDC/Measurement)

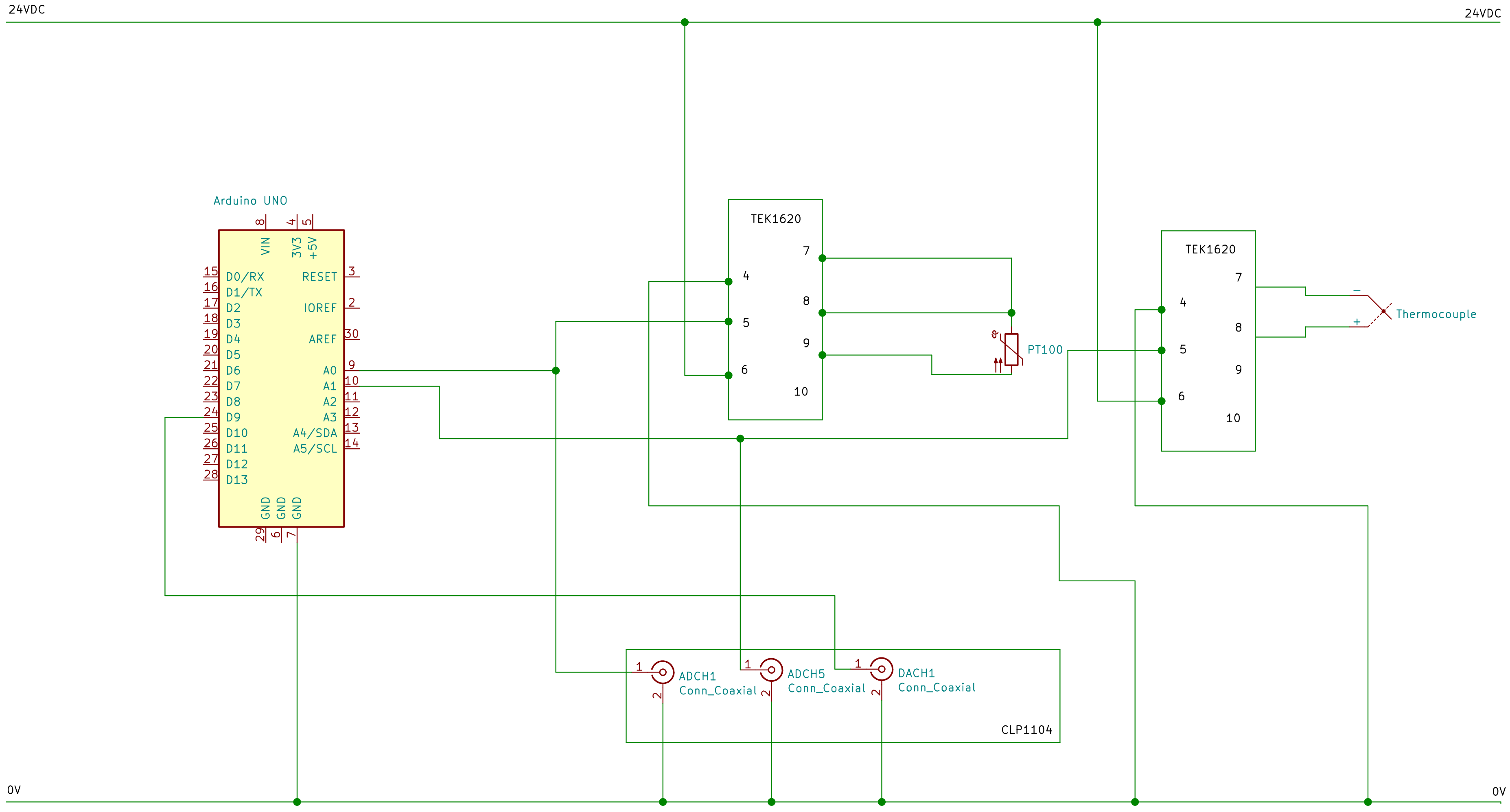
Size: A4 Date: 2019-05-22 **Rev: A4**
KiCad E.D.A. kicad (5.1.2)-1 Id: 1/1



Sheet: /
File: Virtual controller and real model_v2.sch

Title: DAQ/Rmdl and Vctr schematics (VAC/Power)

Size: A4	Date: 2019-05-22	**Rev: A4**
KiCad E.D.A. eeschema (5.1.2)-1		Id: 1/1

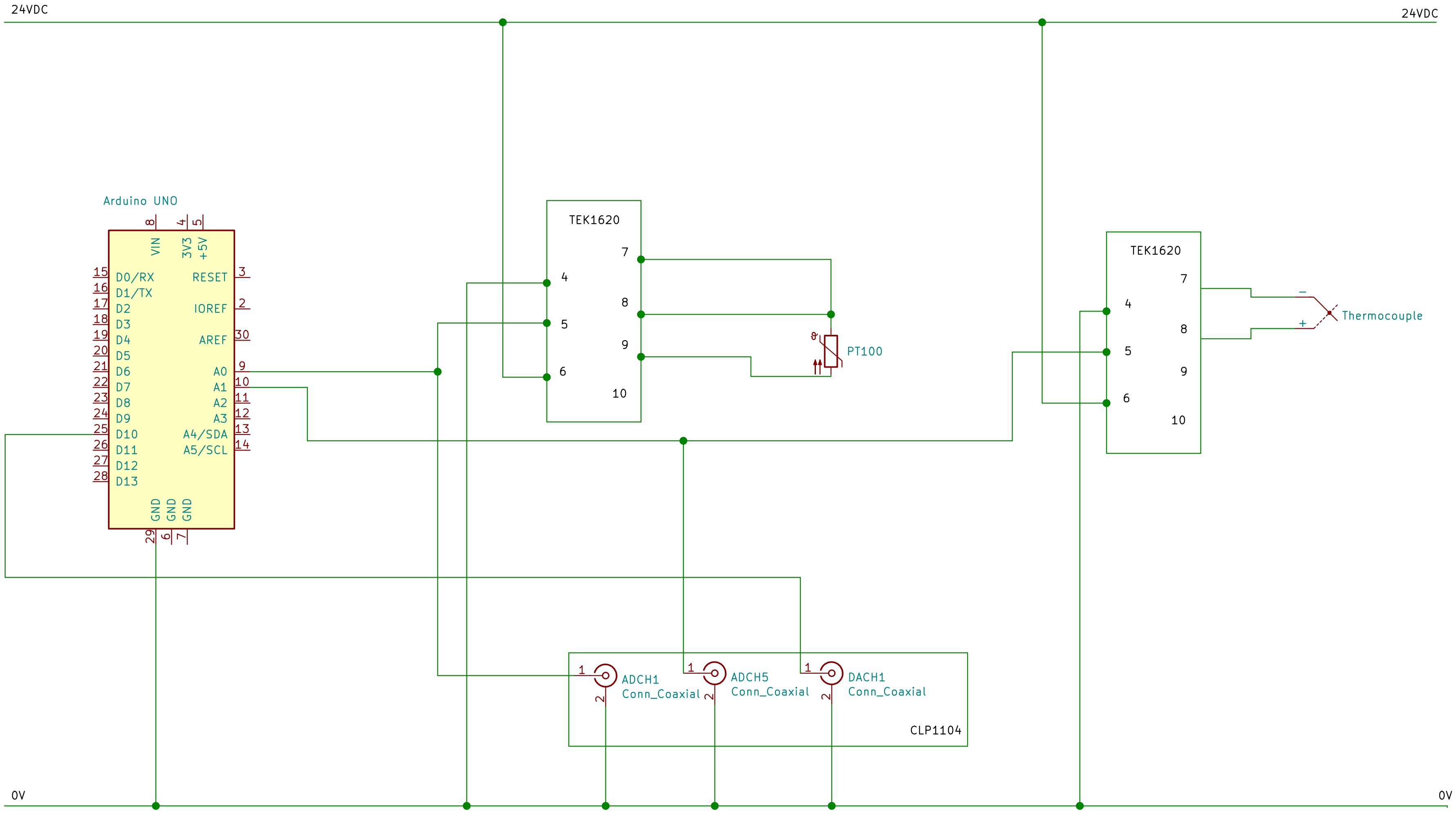


Sheet: /
File: Microcontroller and virtual model.sch

Title: Virtual model and microcontroller schematics (VDC/Measurement)

Size: A4 Date: 2019-05-22 **Rev: A4**

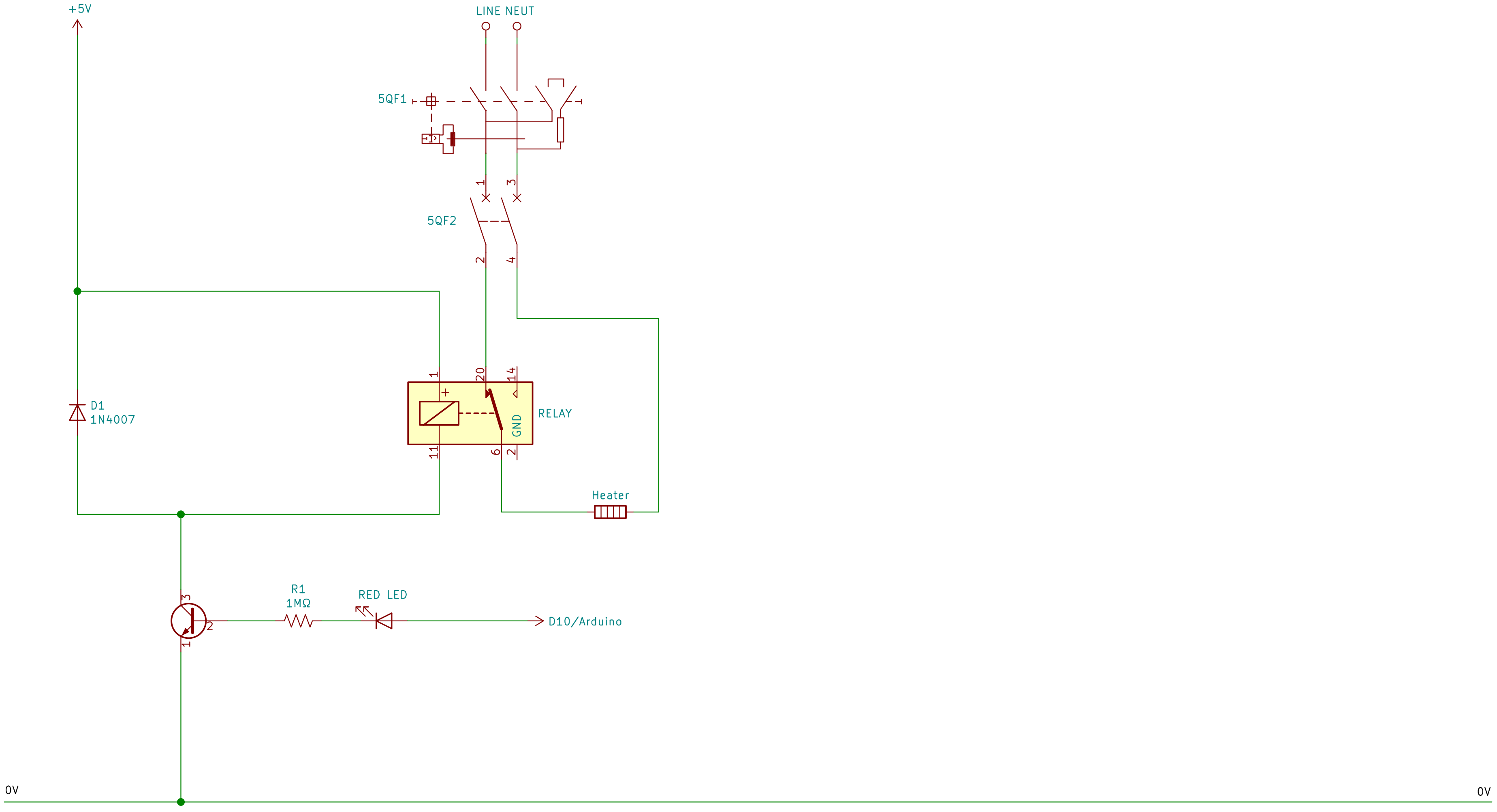
KiCad E.D.A. kicad (5.1.2)-1 Id: 1/1



Sheet: /
File: Microcontroller and real model.sch

Title: Real model and microcontroller schematics (VDC/Measurement)

Size: A4	Date: 2019-05-22	**Rev: A4**
KiCad E.D.A. kicad (5.1.2)-1		Id: 1/1



Sheet: /
File: Microcontroller and real model_v2.sch

Title: Real model and microcontroller schematics (VAC/Power)

Size: A4 Date: 2019-05-22 **Rev: A4**
KiCad E.D.A. eeschema (5.1.2)-1 Id: 1/1

Appendix D

Simulink models associated to case study 1

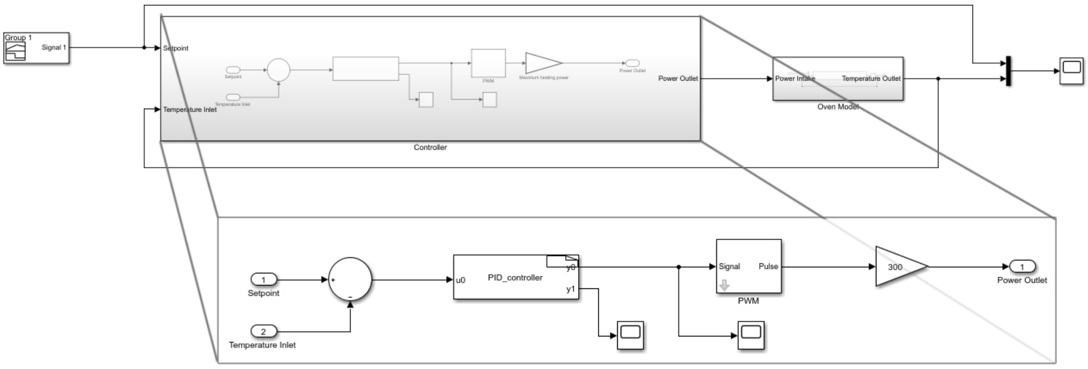


Figure D.1: Virtual controller and virtual model implemented in Simulink.

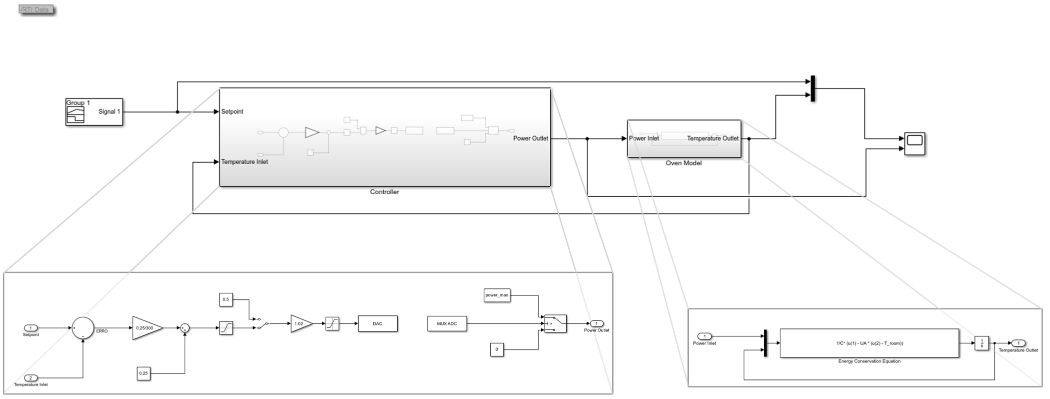


Figure D.2: Real controller and virtual model implemented in Simulink.

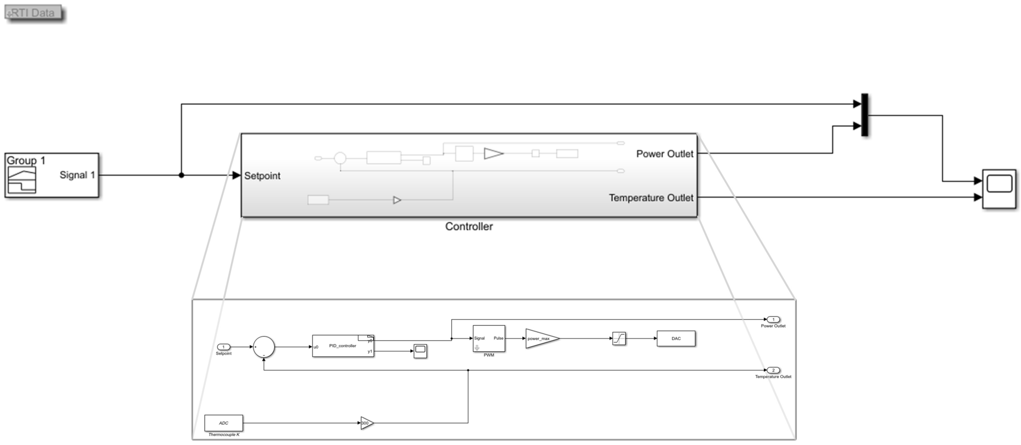


Figure D.3: Virtual controller and real system implemented in Simulink.

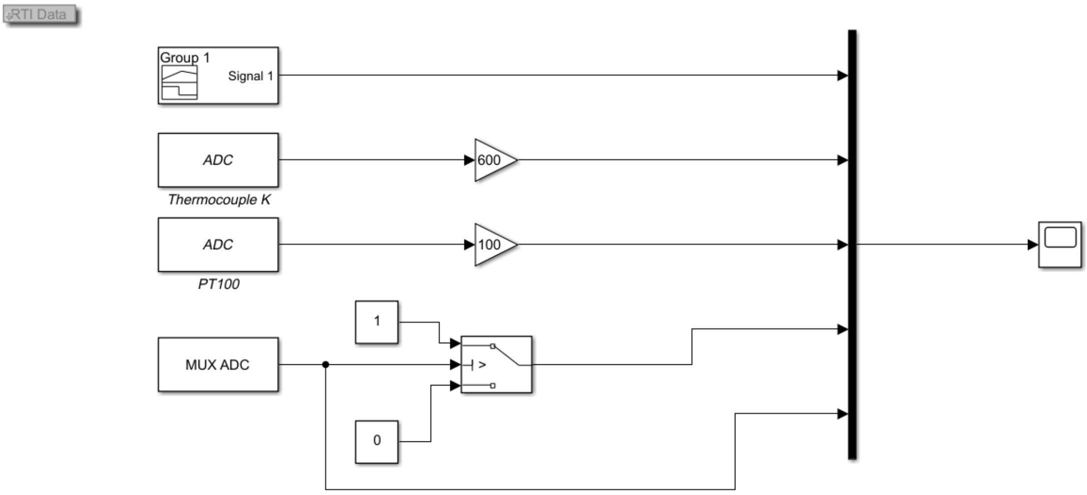


Figure D.4: Real controller and real system, with data acquisiton from the dSPACE controller board model, implemented in Simulink.

Appendix E

Simulink models implemented in case study 2

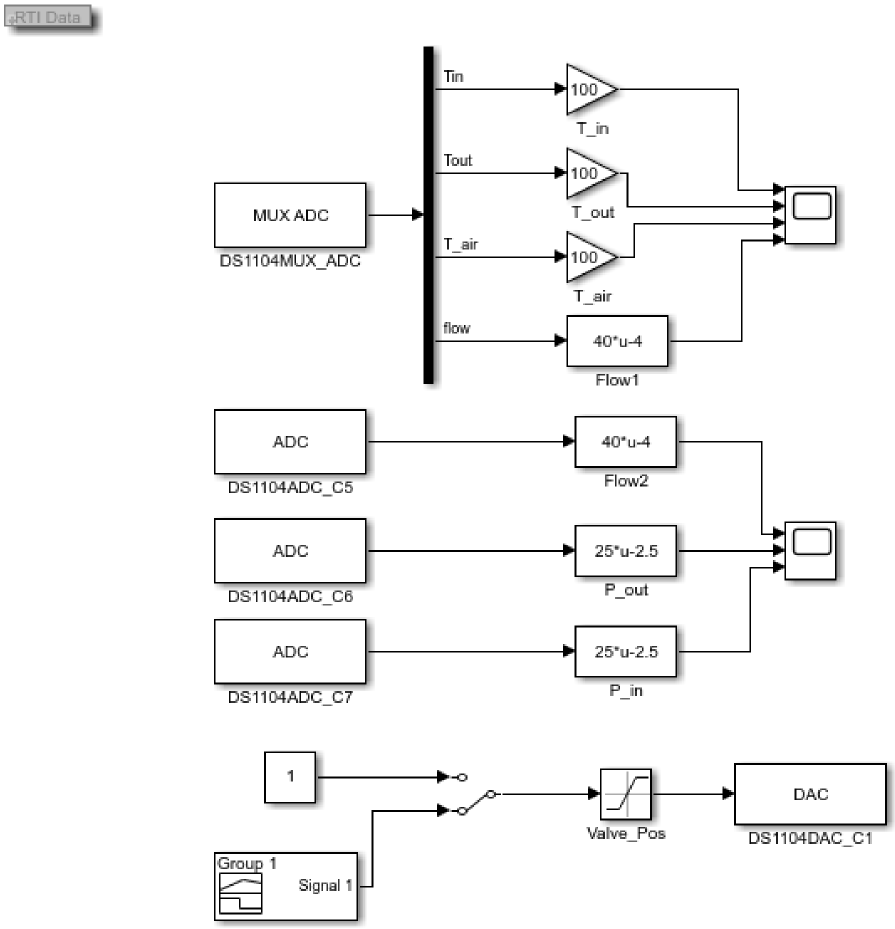


Figure E.1: Block diagram associated with receiving and sending signals, and the appropriate conversions.

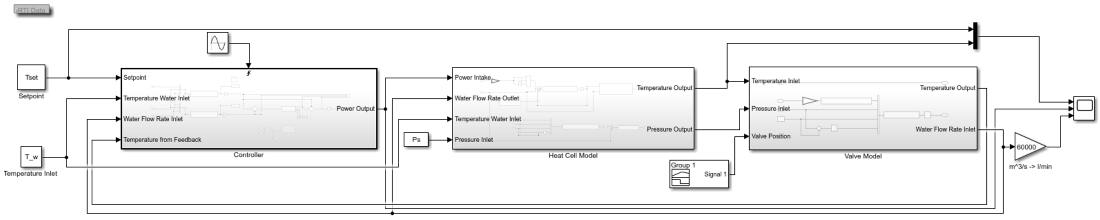


Figure E.2: Block diagram associated with virtual simulation in real time.

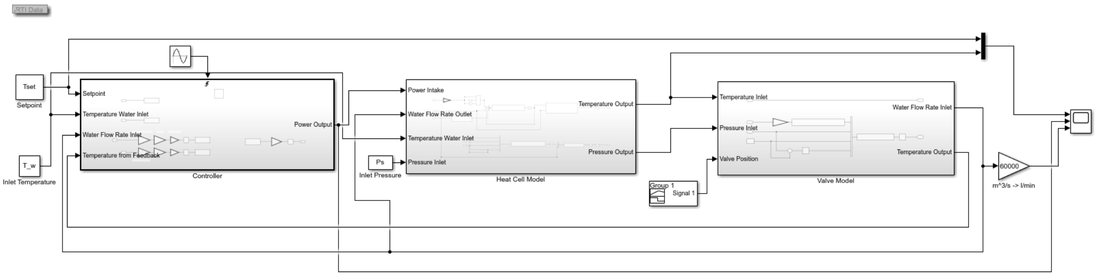
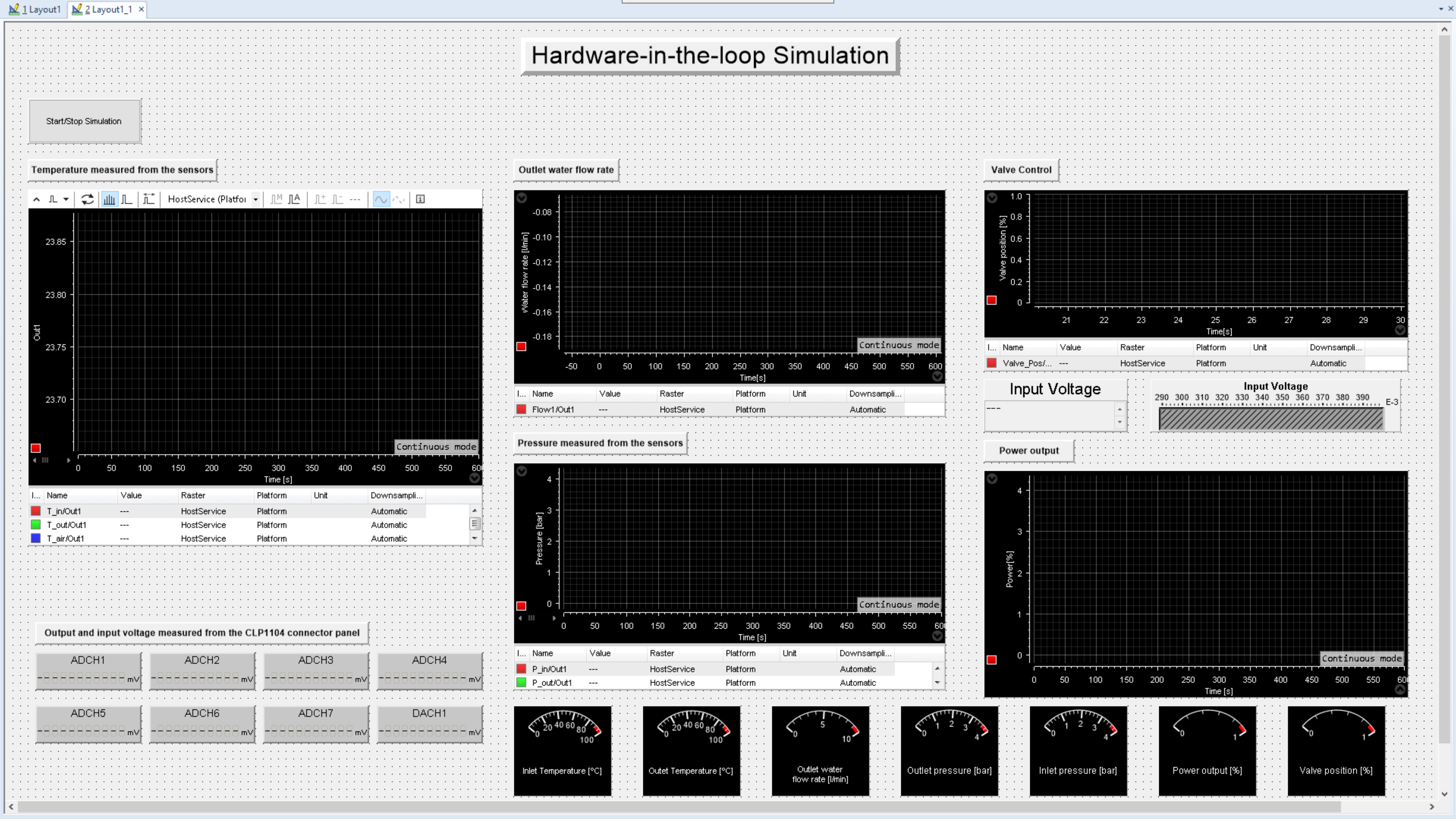


Figure E.3: Block diagram associated to the hardware-in-the-loop simulation with microcontroller.

Appendix F

ControlDesk graphical interface implemented in case study 2



I...	Name	Value	Raster	Platform	Unit	Downsampli...
🟥	T_in/Out1	---	HostService	Platform		Automatic
🟩	T_out/Out1	---	HostService	Platform		Automatic
🟦	T_air/Out1	---	HostService	Platform		Automatic

I...	Name	Value	Raster	Platform	Unit	Downsampli...
🟥	Flow1/Out1	---	HostService	Platform		Automatic

I...	Name	Value	Raster	Platform	Unit	Downsampli...
🟥	Valve_Pos/...	---	HostService	Platform		Automatic

I...	Name	Value	Raster	Platform	Unit	Downsampli...
🟥	P_in/Out1	---	HostService	Platform		Automatic
🟩	P_out/Out1	---	HostService	Platform		Automatic