POLITECNICO DI TORINO

Master's Degree in Mechanical Engineering

AUTOMATION AND DATA VISUALIZATION IMPROVEMENTS OF AN INTERNAL COMBUSTION ENGINE TEST BENCH

Supervisor:
Stefano D'Ambrosio
Politecnico di Torino
DENERG – Energy Department

Co-supervisor:
Andrés Sebastián Herrera
Universidad Politécnica de Madrid
Energetic Engineering Department

Thesis student:
Paolo Dogliotti
n. 291445

Academic year 2022/2023

Abstract

This master's thesis describes the improvement and automation of a test bench for an internal combustion engine at the Universidad Politécnica de Madrid. Installed in the late 70s and initially used for engine testing, the test bench was recently adapted for educational purposes. However, outdated instrumentation presented issues regarding mostly the lack of automatization and poor data visualization that hindered the students' learning experience during laboratory experiments. The objective of the thesis was to modernize and automate the test bench while making it more suitable for university use.

The improvements made to the test bench can be broken down into three categories, the first one being hardware improvements, which include the installation of a new atmospheric pressure sensor, reorganization of thermocouples' cables and recalibration of sensors. Second, changes regarding data visualization and computer interface were applied, adding the visualizations of various engine parameters over time, of the engine's KPIs and of the energy balance. Lastly, upgrades were made to the software program, which both receives input data from the combustion engine sensors, processes and presents them on the screen to the user and sends output signals to control the engine. A system was implemented to control the engine operating point based on the engine load rather than only on the throttle percentage and a function was developed to program the engine's operating cycles.

In particular, a WHSC test, a standardized test cycle used to evaluate pollutant emissions from internal combustion engines, was carried out due to the fact that it has strict preparation requirements and test validations. By being able to meet all of these requirements and validations, a direct verification of the effectiveness of some of the above-mentioned improvements was given.

As a result of this thesis project, the test bench was severely innovated both globally and in the details. A new dimension in data visualization was given, enabling the display of a great amount of data while ensuring clarity, thus maximizing students' learning during laboratory experiences. A severe step forward in automation was made, giving the possibility to carrying out tests as the WHSC test and reducing the need for human intervention on the test bench.

Table of contents

1. Introduction

1.2. Background and motivation

In the Thermal Engines Laboratory of the *Escuela Técnica Superior de Ingenieros Industriales* at the *Universidad Politécnica de Madrid*, there is a test bench for an internal combustion engine. The test bench was installed in the 1970s and was used for engine testing until a few years ago. It was later readapted for educational exercises for university students. In recent years, the university has decided to invest in renovating the test bench, as very few modifications had been made to the laboratory since its inauguration.

The reason for the university's decision to invest in improving the test bench is that the equipment used for it was outdated, and the hardware system presented three main problems: firstly, the equipment was old, and the risk of something breaking or malfunctioning was significant. Secondly, the old manual operating system was extremely complicated and challenging in terms of managing the entire equipment, engine control, and data display, as shown in Figure 1. Lastly, all this hardware, dating back more than 50 years, is voluminous and bulky, so reducing it would bring a considerable advantage in terms of space. Concurrently with this, there was a desire to use the test bench for university exercises, which emphasized these problems that limit students' learning opportunities during these formative moments.

Figure 1: Old hardware instruments

For these three main reasons, in the last two years, two students participated in the innovation project and carried out two thesis works on the test bench to update it and transition from a mainly manual system to an almost entirely digitalised system for data acquisition and engine control. They have worked to eliminate most of the old hardware instruments and replace it with more reliable and compact equipment, ensuring that data passes through a computer located in the test bench control room for data managing, processing and visualization. At this point, having digitalised the entire test bench, it is possible to manage the acquisition and transmission of signals to and from the engine much more easily, thus providing a much greater possibility of customizing the test bench to meet the necessary requirements.

1.2. Practical laboratory lessons

During my Erasmus study period at the *Universidad Politécnica de Madrid*, I attended the Thermal Engines course and participated in a practical laboratory session. The laboratory exercise represents a crucial moment for students, allowing them to apply the theoretical knowledge acquired during the course. In this two-hours experience, students have the opportunity to work directly on a test bench for a thermal engine, applying theoretical concepts and measurement methods. The main activity of the practical exercise is to bring the engine at various operating points to examine its performance under variable conditions. Once the engine is set on an equilibrium point, students record the values of all parameters measured by sensors on the engine, such as temperature, pressure, and flow rates. This recording phase is essential for the subsequent independent analysis that each student must perform on their own to assess some of the key engine performance indices studied during the course.

In this context, students refine their skills in managing experimental data and applying theoretical concepts in the field of thermal engines. The experience not only allows them to consolidate technical skills but also contributes to the development of critical and analytical thinking, essential in the engineering field. Specifically, by the end of the experience, students will have:

- Acquired the ability to identify and understand the different components of the test bench used in the exercise. This competence is crucial to have a comprehensive view of the experimental process and the test environment.
- Understood the operation of sensors for each variable to be measured on the engine, also developing a critical view of the importance of such devices in the context of engine measurements and tests.
- Understood the risks associated with experimental activities on internal combustion engines. Students will have learned to recognize and evaluate potential hazards, while acquiring the necessary knowledge to take appropriate precautions to ensure personal safety and a safe working environment.
- Developed the ability to perform appropriate calculations to obtain accurate results. Students must demonstrate their ability to apply mathematical and analytical methods for processing the collected data, allowing them to draw meaningful conclusions from the conducted tests.

In summary, this exercise goes beyond a simple practical activity, serving as a crucial link between theory and practical application in the field of thermal engines, enabling students to develop a wide range of essential skills for their training as engineers.

1.3. Scope and objectives

We have seen in Chapter 1.1 how, until now, the project's focus has been more on the digitization of data acquisition, visualization, and control systems. In itself, with this transition, the test bench could do almost nothing that could not have been done before with an analogic system, but obviously, the digital system was much more innovative, simple, and clear. Continuing the recently completed work, the objective is to take advantage of the digitalisation of the entire system and innovate and automate the test bench as a whole. We aim to make substantial changes that have a clear impact on the test bench, together with small modifications that may not necessarily revolutionize the test bench but improve specific and minor issues. More specifically, at the beginning of this project, 8 issues were identified concerning the test bench, as described below:

1. The engine is equipped with a large number of sensors and instruments to measure some of its key parameters. In particular, there are pressure sensors and thermocouples at various points in the operation cycle of the internal combustion engine, as well as flow meters for engine fluids. Currently, the instantaneous values of pressure, temperature, and flow are displayed without providing information on their temporal development. This limitation is particularly restrictive during exercises, where students have difficulty understanding the variation of these parameters in response to changes in the engine's operating conditions.
2. Some raw parameters measured on the engine, such as fuel or air flow, can be challenging to interpret. Therefore, there is a need to calculate and display some Key Performance Indices (KPIs) that can provide more information than the single measurement of a parameter.
3. Another important function missing from the user interface panel is the ability to visualise the energy balance of the engine to observe how the portion of useful energy provided by the fuel is distributed between the power delivered by the engine and other dissipative effects.
4. The test bench requires some modifications regarding engine control. The operating point of the engine, characterized by the delivered power and engine speed, is controlled in two different ways: the dynamometer is used to control the engine speed, while the delivered power is controlled by the percentage of the pedal pressed. The non-linearity between load percentage and the percentage of the pressed pedal and its variability with engine’s speed does not allow the user to control the engine by choosing the load percentage.
5. The test bench is devoid of automation and always requires human intervention during exercises and engine tests.
6. An atmospheric pressure sensor is missing. Such parameter needs to be estimated and inserted manually, causing a decrease in accuracy on pressure measurements.
7. Some of the thermocouple cables are old, and due to the high number of sensors, all cables are very disorganized.
8. The main user interface screen needs to be changed to satisfy new requirements and improved data presentation.

1.4. Structure of the manuscript

In this paper, a comprehensive description will be provided of the work carried out for the automation and improvement of the test bench to address each of the 8 key issues just described. In the next chapter, a detailed description of the test bench will be given, aiming to explain the functioning of the system and its individual components. Special attention will be given to the components and parts that will be discussed more extensively throughout this thesis, while other parts which are important but less relevant to this specific work will be described more superficially. In the yet following chapter, all 8 issues and the related modifications made to resolve them will be exhaustively described, in some cases along with the thought process behind them, and the path that led to the final result. In the last chapter, a standardized cycle test will be conducted to test some of the improvements made and verify both qualitatively and quantitatively the effectiveness of these enhancements.

2. Description of the engine test cell

A test bench is a system designed to conduct a series of tests in a controlled environment to evaluate various characteristics of the engine, measure performance, identify potential issues, and optimize its operation. The structure of our test bench is divided into two distinctive rooms: the first hosts the engine and is designed to allow its safe operation, featuring specific functional characteristics. This room is acoustically isolated from the external environment and it is equipped with a support system for the engine, which serves the dual function of providing structural support for coupling with the dynamometer and absorbing vibrations generated by the forces produced by the engine. Finally, an air recirculation system is present, ensuring a constant presence of oxygen for combustion and partially contributing to the cooling function, emphasizing that, under road driving conditions, this function is performed by the airflow through the engine during the vehicle's movement.

Figure 2: Engine room

The second room is occupied by all the hardware instruments necessary for the data acquisition, processing, and visualization system, electric power supply systems, and the control system. Personnel are located in this room during test execution, and for this purpose, there is a physical separation from

the engine room, achieved through a wall with a glass window that facilitates supervision and monitoring of any issues in the engine room.

In this chapter, we will provide a description of the test bench before the interventions carried out during this thesis project, both regarding general operation and individual components. Special attention will be given to those components most relevant to this thesis and that will be discussed further in the following chapters.

2.1. The engine system

As far as the engine is concerned, it is a turbocharged diesel engine with four inline cylinders, model YD25DDTI, manufactured in 2005 and originally designed for the Nissan Pathfinder vehicle. It was donated to the university in 2012 by the manufacturer for educational purposes.

Figure 3: Engine on the bench

In the following table, the main specifications of the engine are shown:

Engine type	Compression-ignition engine
Number and configuration of cylinders	Inline 4
Displacement	2488 cm^3
Maximum power	128 kW @ 3400 rpm
Maximum torque	403 Nm @ 2000 rpm

Table 1: Engine's specifications

The engine is equipped with a turbocharging system that allows the compression of the intake air drawn to increase its density, hence the volumetric efficiency of the engine, using the residual energy from the hot exhaust gases. However, the air heats up during the compression process, reaching a higher temperature than the environment, thereby slightly reducing its density. To address this, an *intercooler* is often employed, a system that cools air before entering the combustion chamber to increase the gas' density once again. For this purpose, the engine is equipped with a fan that cools the gas passing through the ducts between the compressor and the combustion chamber.

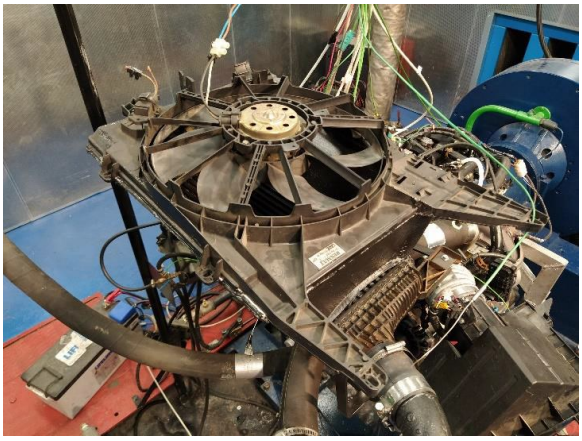

Figure 4: Intercooler fan

Additionally, there is a water circuit responsible for cooling the engine coolant, which heats up due to the energy exchanged with the hot components in and around the combustion chamber.

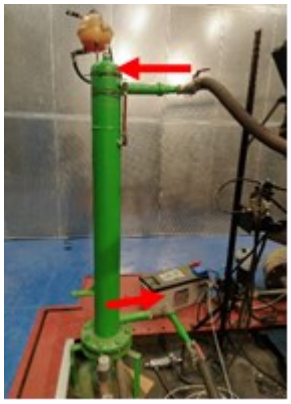

Figure 5: Cooling system

The last main component inside the engine room is the eddy current dynamometer. This is a device connected to the engine's output shaft and is used to measure the torque and power generated by the engine while simultaneously dissipating it. The dynamometer operates on the principle of Eddy

currents, where the resistance force produced on the engine shaft changes with the variation of voltage and thus the intensity of the magnetic field. These currents generate heat, so such types of brakes must include a cooling system to maintain their temperature below the limit.

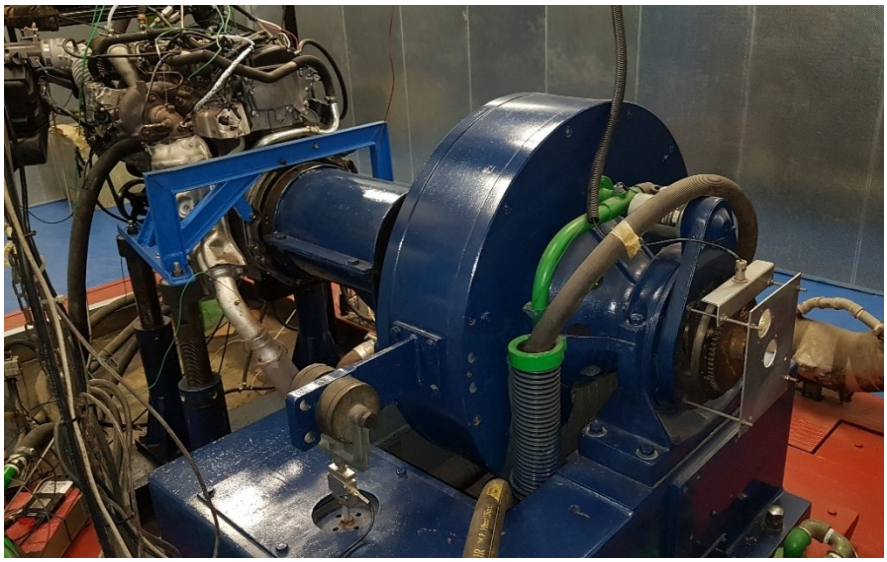

Figure 6: Eddy current dynamometer

2.2. The control room

Next to the engine room of the test bench, we have the control room. In here, there is all the instrumentation that is not needed for running the engine itself but is specific to the test bench for conducting tests and engine analysis. In particular, in the control room, we need:

- The instrumentation necessary for the general operation of the engine, required for engine start and stop, and for the activation of cooling systems.
- A data management system, which is responsible for supplying power to the sensors, receiving and transmitting data from the sensors to the computer, and sending signals to the engine control systems.
- Devices for data processing and visualization.

Originally, there was a rack that performed all three functions mentioned above, divided into various compartments, each with a different sub-function:

Figure 7: Old hardware instruments

We recall that the main reasons for deciding to renovate the test bench were precisely because this type of instrumentation was obsolete, unreliable, and unsuitable for educational exercises. Much work has been carried out in the last two years by two thesis students who worked on digitalising the control and data acquisition system, using a computer for this purpose. At the end of their work, the room (Figure 9) is as follows:

- The control cabin is basically unused except for a few compartments.
- The entire data acquisition system has been replaced with a more advanced system that can be integrated with software. This system will be discussed in Chapter 2.3.
- A small control panel, made more compact, retaining only a few essential functions that did not require digitalisation, such as engine ignition and refrigeration system activation.

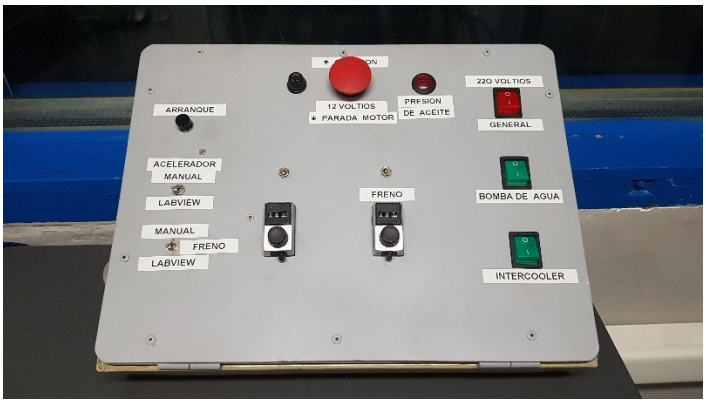

Figure 8: Main control panel with switches

- The computer which, using LabVIEW software, receives signals from the sensors on the engine, performs data processing and visualization, and sends signals for power and rotation speed control. This aspect will be discussed more extensively in Chapter 2.4.

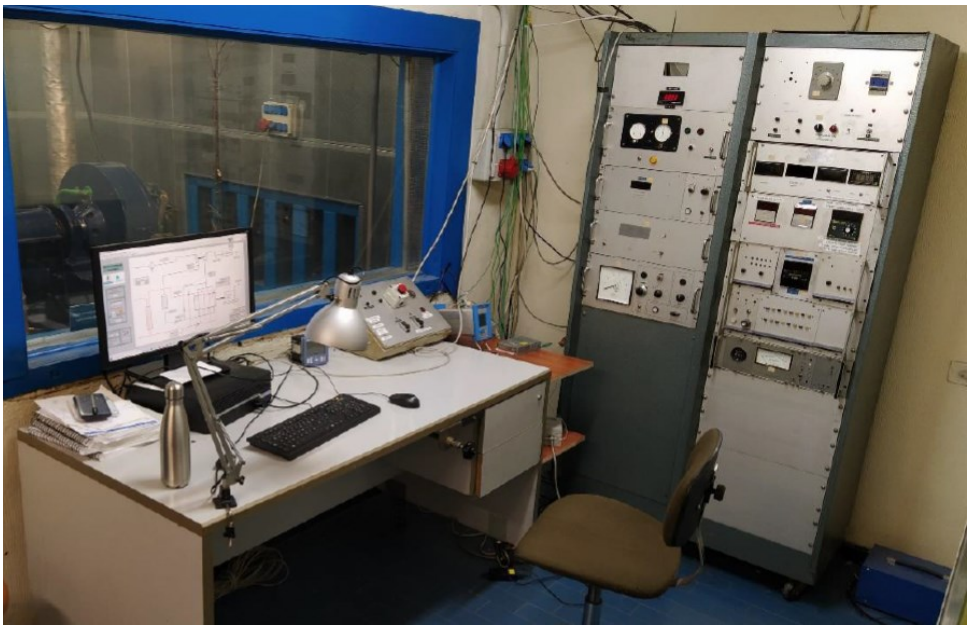

Figure 9: Old aspect of the room

2.3. The data acquisition system

The engine installed on the test bench is equipped with a high number of sensors and devices at various points to measure the relevant quantities. In particular, there are three types of sensors:

- Four pressure sensors, located after the supercharging group compressor, after the intercooler, at the exit of the combustion chambers, and after the supercharging turbine.
- Eleven thermocouples to measure the temperature of fluids before entering and after passing through almost every component of the engine. This includes measuring the temperature:
    - At the intake which is also the compressor's inlet.
    - At the compressor outlet or the intercooler inlet.
    - At the intercooler outlet or the combustion chamber inlet.
    - At the combustion chamber outlet or the turbine inlet.
    - At the turbine outlet or the catalytic converter inlet.
    - At the catalytic converter outlet or the engine exhaust.

    - Of the coolant water before and after passing through the engine.
    - Of the coolant liquid inside the engine and of the fuel.
    - Of the coolant water outlet at the dynamometer.
- Three flow meters to measure the flows of the three main fluid streams: of the air aspirated by the engine, of the fuel, and of the water for cooling the engine coolant.

These sensors measure the physical quantity and convert it into an electrical signal, which is then sent to modules. The test bench is equipped with two input modules, one specific for thermocouples and one for voltages used for pressure and flow meters. These two devices are responsible for converting the electrical signal from the sensor into a digital signal transferred to the computer via a USB port. A third and final voltage output module is used for the opposite function, which is transforming a digital signal sent by the computer into an electrical signal to be sent to the engine for throttle pedal control and to dynamometer for speed control. In the following image, you can see the three data acquisition modules, the one on the bottom for thermocouple inputs, the one in the centre for inputs from all other sensors, and the one on the top for outputting signals to the engine.

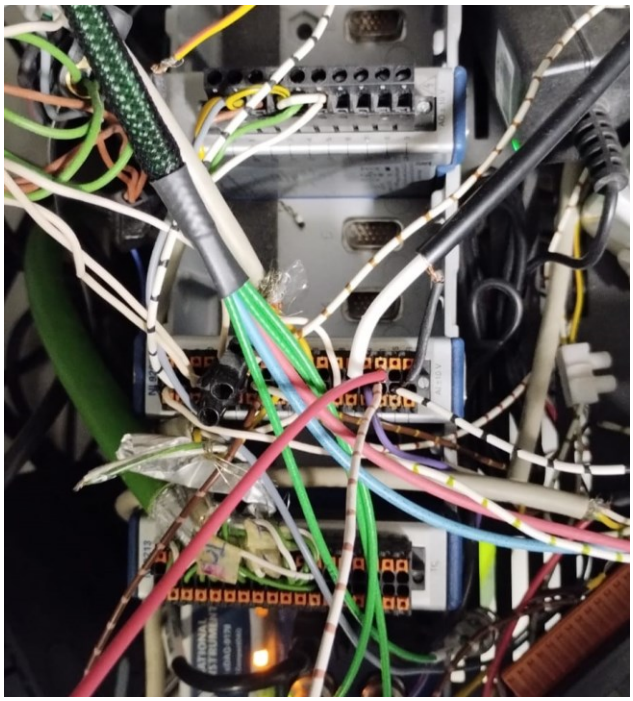

Figure 10: Input and output modules

Finally, the computer is appropriately integrated with the DAQ system to receive and process data from the sensors, which will be discussed in the next chapter.

2.4. The LabVIEW Program

Regarding the data processing and visualization part, we use LabVIEW, a programming environment for creating applications for data acquisition and analysis, industrial automation, testing and measurement, and more. The main feature of LabVIEW is its graphical development environment, where you can create applications by connecting icons and graphical symbols to form a flowchart. LabVIEW has two main advantages: first, the wide and easy hardware integration between a computer and a data acquisition board, useful in this case to obtain data from various sensors on the engine, and second, the graphical programming that allows for rapid changes and improvements to the program.

The four main functions required from the LabVIEW program are the following:

- Integration with the input module to receive signals from the sensors installed on the engine.
- Integration with the output module to send signals for engine control.
- Data processing and analysis.
- Visualization of raw and processed data.

LabVIEW is divided into two panels with profoundly different functions. The first one is the *front panel* and is used during the engine's operation. This panel allows interaction between the user and the engine during its operation. All desired variables are displayed here, and it is possible to change some parameters to send signals to control the engine. A photo of the current panel is shown in Figure 11 on the following page.

From the panel image, we can appreciate two distinct parts. On the right part, there is an engine diagram along with all its components, including the dynamometer, the turbocharger, the catalytic converter, and the three different cooling systems. Real-time values of temperature, pressure, and flow rate at each point in the system where sensors are positioned are displayed. On the bottom right, we also have values for speed, torque, and power produced by the engine. This part is dedicated solely to displaying data acquired and received from the acquisition system.

On the left there is the control section, where users can interact with the engine and the data acquisition system. Below the indicator for manually entering atmospheric pressure, there is the

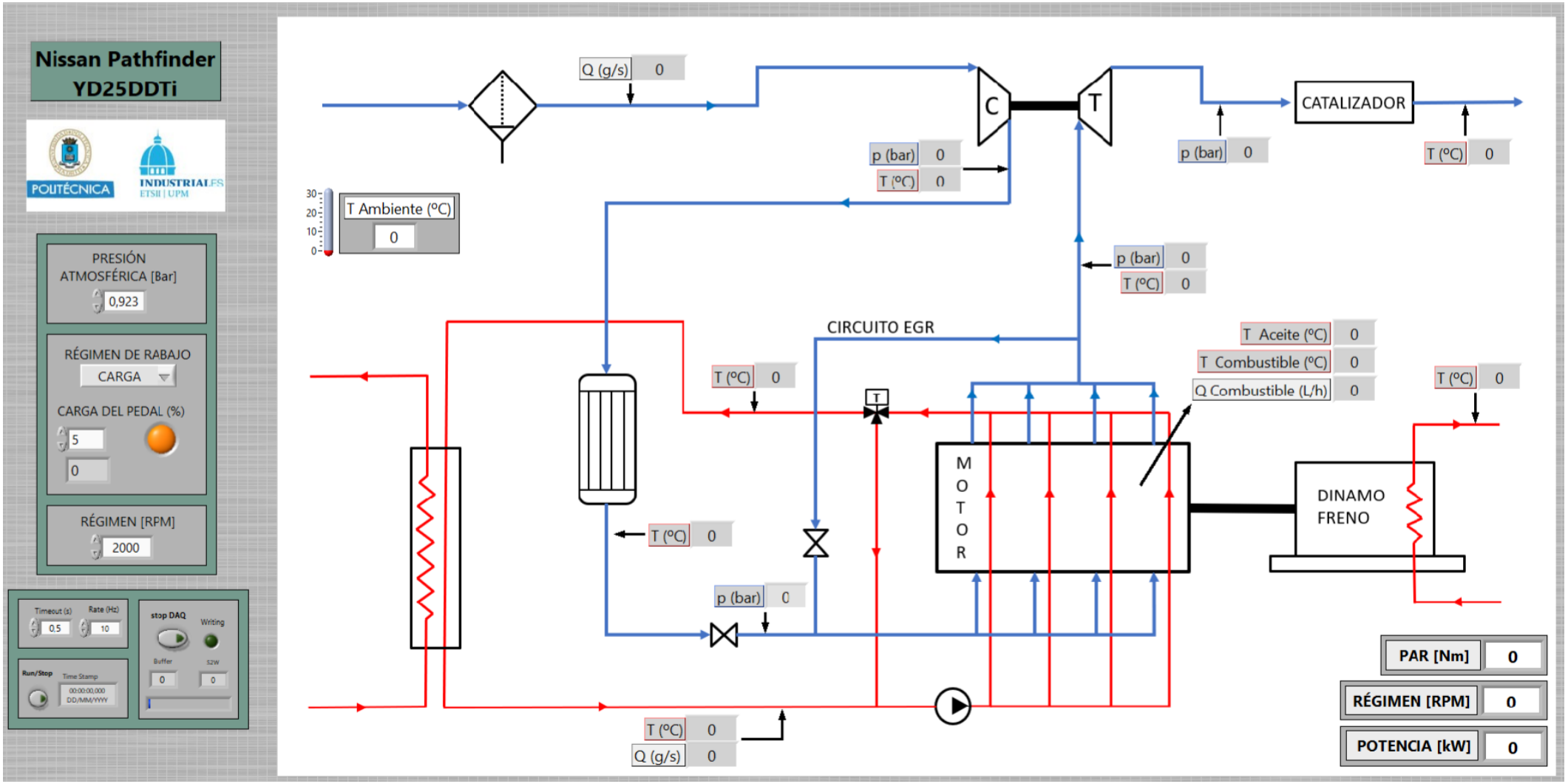

Figure 11: Old front panel

engine control through two indicators, one for speed (*régimen*) and the other for the pedal percentage (*carga del pedal*), along with a dropdown menu called *regimen de trabajo* or working regime, allowing the choice between *carga* or load regime and *ralentí* or idle. This last option in the dropdown menu puts the engine in idle working condition, where the engine does not generate useful power, hence the torque measured by the dynamometer is zero. It sets the engine to the minimum speed to provide power only to auxiliary engine accessories. During the idle condition, controls for rotation speed or power are deactivated. The other condition is the load regime, where the engine provides useful power. In this case, the user chooses the engine's speed using the appropriate control. Regarding power, it varies by sending a signal that controls the percentage of the throttle pressed. Finally, below the engine control part, there is the data acquisition control. Here, the user has the ability to change the data acquisition frequency and stop saving data to an external file.

After discussing the main panel thoroughly, let's move on to discussing the block diagram, which is the environment for programming the front panel. Here, the programmer can combine blocks representing different functions, operations, and data flows to process signals obtained from acquisition systems and present them on the front panel. On the following page, the block diagram of the current state of the program is shown. We can appreciate four different purposes of this program:

1. The program part highlighted in red is related to the acquisition of data coming from the sensors on the engine through the DAQ. The DAQ function processes all the data coming from the data acquisition card.
2. The part highlighted in green is related to data processing and visualization. The input data move through the program via blue lines and is sent to different blocks. These can be the blocks of the indicators on the front panel, which, by receiving the obtained signals, display them on the user interface, or they can be functions that process the obtained data to get more relevant data.
3. The fourth part of the program in orange is responsible for sending output signals to the engine for control via the output module.
4. The remaining part on the bottom is responsible for saving data on an external file for post-data analysis.

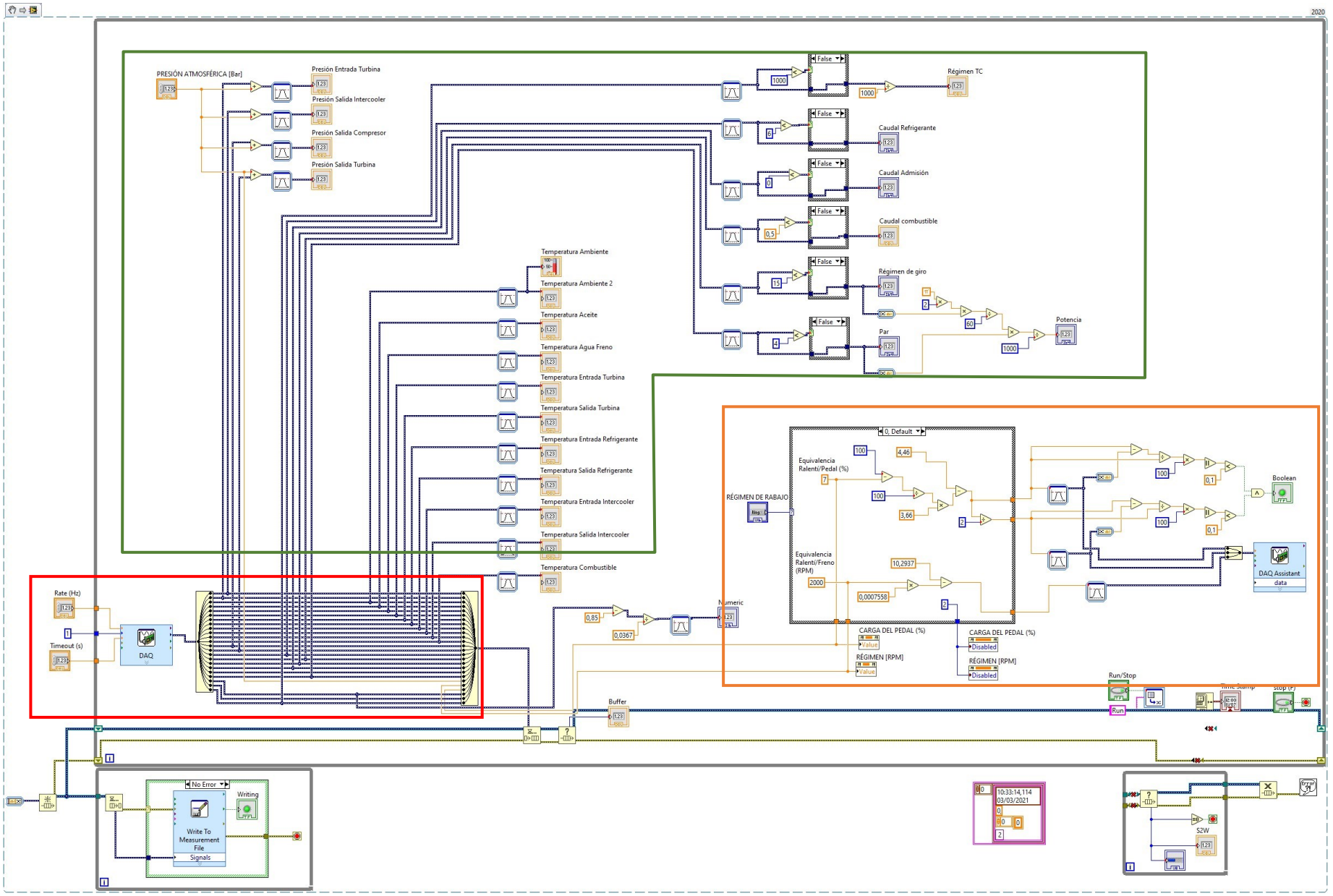

Figure 12: Old block diagram

3. Accomplished work

The project of innovation, updating, and automation of the test bench previously described has made good progress thanks to the work of two other thesis students who have worked towards this goal. However, much work still needs to be completed to conclude this project. In this chapter, we will address the changes that have been implemented during this thesis work. The goal of the thesis is not to improve just a specific aspect of the test bench but rather to enhance it in every possible way. Both small modifications to solve specific and minor issues of the test bench and more significant improvements aimed at completely enhancing the test bench have been made. For this reason, various areas of the test bench have been addressed during this work, including the hardware part related to sensors and data acquisition, the software part, involving programming and data management, and the user interface for data processing and visualization. The summary table below outlines the 8 modifications that we will discuss in the following chapters, categorized into small and large upgrades and by area of interest:

	Hardware	User interface	Software
Small upgrades	3.1. Atmospheric pressure sensor 3.2. Thermocouples' reorganization	3.3. Front panel improvements	
Big upgrades		3.4. Time-based graphs 3.5. KPIs graphs 3.6. Engine energy balance	3.7. Load percentage control 3.8. Cycle programming function

Table 2: Categorised improvements

3.1. Installation of an atmospheric pressure sensor

The test bench was not equipped with an atmospheric pressure sensor, and such parameter had to be manually entered by assuming that the pressure in the laboratory was the same as that in the *Parque del Retiro*, a park at approximately the same altitude and about 3 kilometres away from the

university. The city of Madrid provides online the average hourly pressure in this park, and this pressure had to be manually entered in LabVIEW. This assumption, of course, is not precise and introduces a significant error into all pressure measurements since the pressure sensors installed on the engine is relative pressure, and atmospheric pressure must be added to obtain absolute pressure in each point. It is obvious that the installation of an atmospheric pressure sensor is necessary to measure this variable more accurately, integrating it into the LabVIEW program, and avoid the manual entry of this measurement.

Therefore, an MPXH6115AC6U pressure sensor was purchased—a cost-effective yet sufficiently accurate sensor for the straightforward measurement of atmospheric pressure. The sensor has the following specifications from the catalogue:

Maximum pressure	400	kPa
Operating temperature	-40 to 125	°C
Minimum rated pressure	15	kPa
Maximum rated pressure	115	kPa
Output voltage at minimum rated pressure	0,2	V
Output voltage at maximum rated pressure	4,7	V
Error	±0,0675	kPa

Table 3: Atmospheric pressure's specifications

The error indicated in the table is related to a temperature between 0 and 80°C, a range in which we expect the sensor to operate under nominal conditions. The sensor is indeed located approximately 3 meters away from the engine and will never exceed a temperature of 80°C.

The sensor is sold disassembled, so it needs to be assembled and soldered. Below is a schematic of the circuit that needs to be created for the sensor:

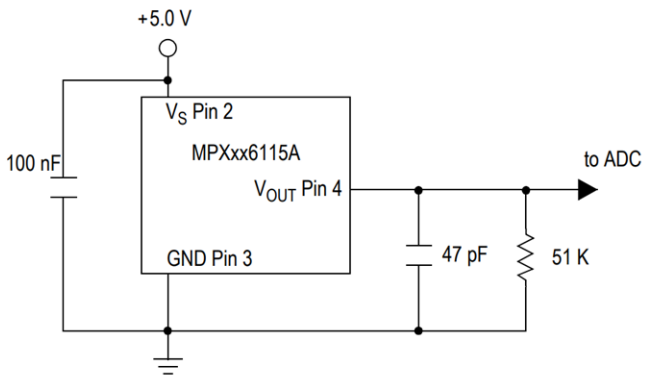

Figure 13: Atmospheric pressure's circuit

After assembling the circuit, the sensor is affixed to a box with a label that reads *Presión atmosférica*, meaning atmospheric pressure, and placed in the engine room. Once this is completed, it is necessary to integrate the hardware with LabVIEW. To do this, we connect the sensor to a 5V power supply through Pin 2 (see Figure 13), while Pins 3 and 4 are connected to the input module, one to the ground and the other to the voltage input.

Figure 14: Atmospheric pressure's installation

The final step is the calibration of the sensor directly in LabVIEW. To do so, we must set the maximum and minimum pre-scaled values and the associated values of pressure. The pre-scaled variable is the voltage of the signal that is sent from the sensor to the computer through the module. Hence, we take the data from Table 3 and put it into the program, which include the maximum and minimum pressure values along with the corresponding voltage readings from the pressure sensor. In

just a few steps, thanks to the ease of integrating new hardware equipment with LabVIEW, a ready-to-use atmospheric pressure sensor has been installed.

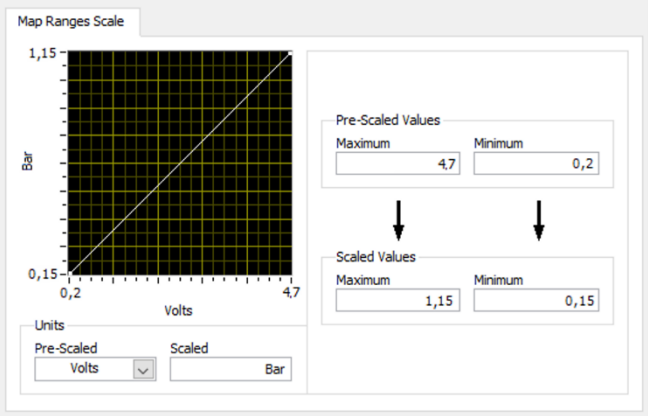

Figure 15: Atmospheric pressure sensor calibration

3.2. Reorganization of thermocouples' cables

We previously discussed how the project's objectives not only aimed to innovate and automate the test bench with significant changes but also to implement small modifications to enhance overall efficiency. This was clear in the reorganization of the cables for the thermocouples installed on the engine.

There are 11 thermocouples on the engine measuring temperature at various points, and the cables from each of these must run from the measuring point to the data acquisition module in the control room, passing through some holes in the wall that separates the two rooms. All these cables were initially disorganized, causing confusion and a messy appearance. To address this issue, a multi-pair thermocouple cable was used, a wire which has a diameter of about 1.5 cm and a semi-rigid coating through which a large number of thermocouple cables pass. At its ends, the thermocouple cables are free, so one end is placed near the engine and connects to all its thermocouples. The other end of the multi-pair cable extends to the thermocouple input module, to which we connect all the thermocouple wires. This cable has the significant advantage of connecting the thermocouples to the input module despite the long distance in a very organized and straightforward manner.

The room, having also removed many cables for the obsolete equipment that was taken out, is visually more organized and has a more professional appearance. However, the beneficial aspects of this modification are not only visual; having cables well-organized and protected helps preserve the integrity of sensors and cables over time, and it allows for easier identification of a problem in the event of a malfunction in the connection between the thermocouple and the computer.

3.3. Front panel improvements

In Chapter 2.4, we observed the front panel of the LabVIEW program for the test bench. This screen is the interface between the test bench and the user, so it is essential that it is clear and simple yet rich in information. On the following page, in Figure 16, we observe the new main screen with the improvements made and compare it with the old screen (Figure 11) to note the following main changes:

- The section related to the customization of data saving has been eliminated. Indeed, although saving data is crucial for later analysis, the main screen offered the freedom to customize the data acquisition frequency and timeout, which, with experience, was observed to be unnecessary to change. For this reason, we have fixed a data acquisition frequency of 10 Hz and a timeout of 0.5 s. The index related to the number of rows in the buffer was removed as it always remained at zero. This section has been streamlined, leaving only the option to stop data acquisition and an indicator to confirm that data writing is in progress.
- A tab has been added to allow the user to navigate between different sections of the program. We have already mentioned that with this thesis project, we want to provide more information to students, so the tab allows choosing which section to display by selecting the relevant tab. Each section will be described exhaustively in the following chapters.
- Other minor changes have been made to the general appearance, reorganizing sensor indicators, adding the pressure sensor described in paragraph 3.1, and redesigning the engine diagram.

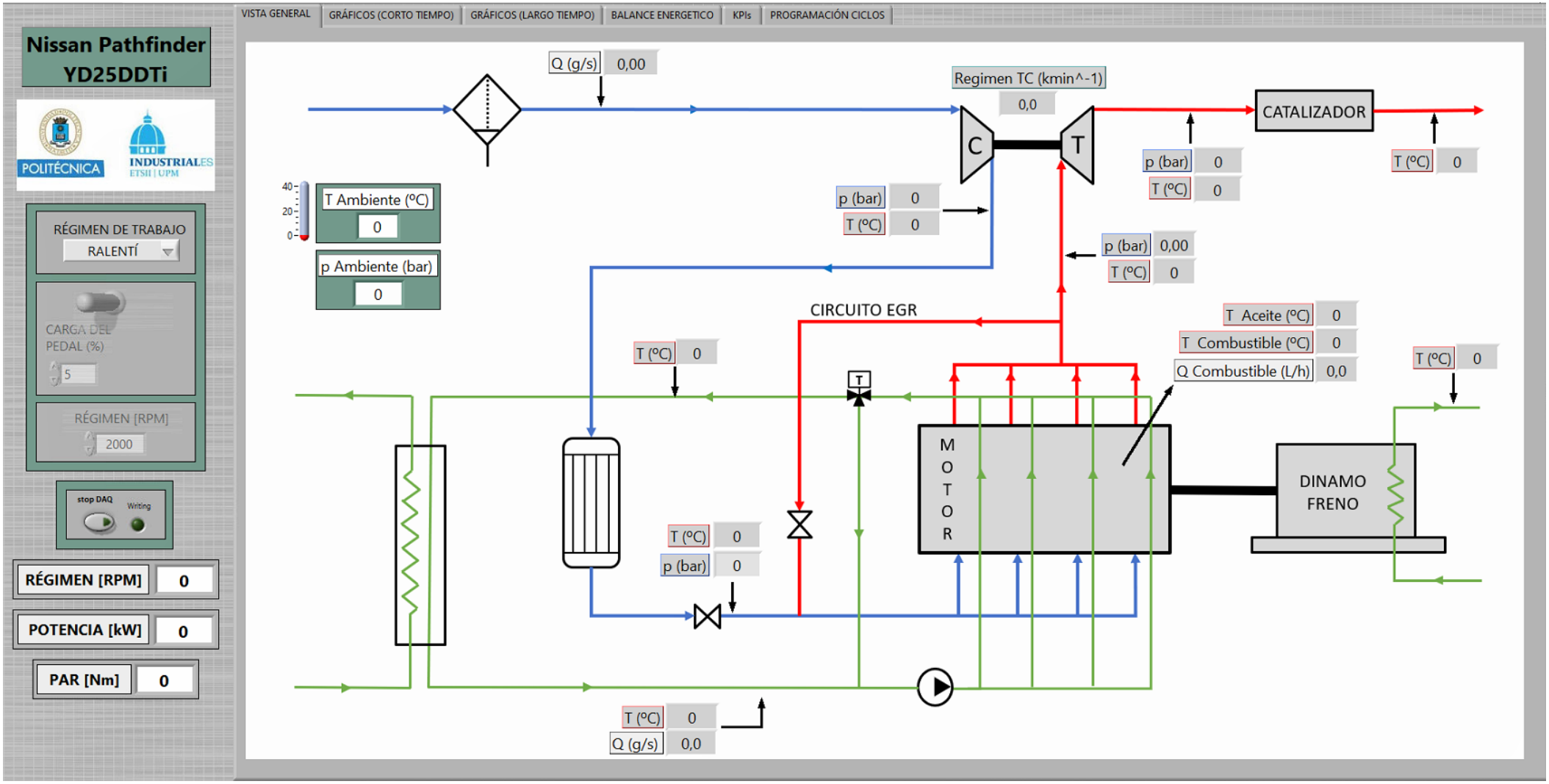

Figure 16: New front panel with improvements

3.4. Time-based graphs

Integrating the time-based graphs of all the variables measured by the engine sensors was one of the initial objectives set at the beginning of the project. The intention was precisely to help students understand how engine variables change with varying operating conditions. This was not possible during the exercise until now because only instantaneous values of engine variables were shown: students could observe changes in conditions at home during data analysis, but it is more appropriate to be able to see it during the practical exercise with the professor.

At this point, it was chosen to display the graphs of all variables in two different ways. In the first screen, short-term graphs are represented, showing all variables over a time span of 90 seconds. These graphs aim to focus on transient conditions between two different engine operating points. Let's observe an example of the screen that appears on the tab *GRAFICOS (CORTO TIEMPO)*, which in English would be "Graphs (Short-term)":

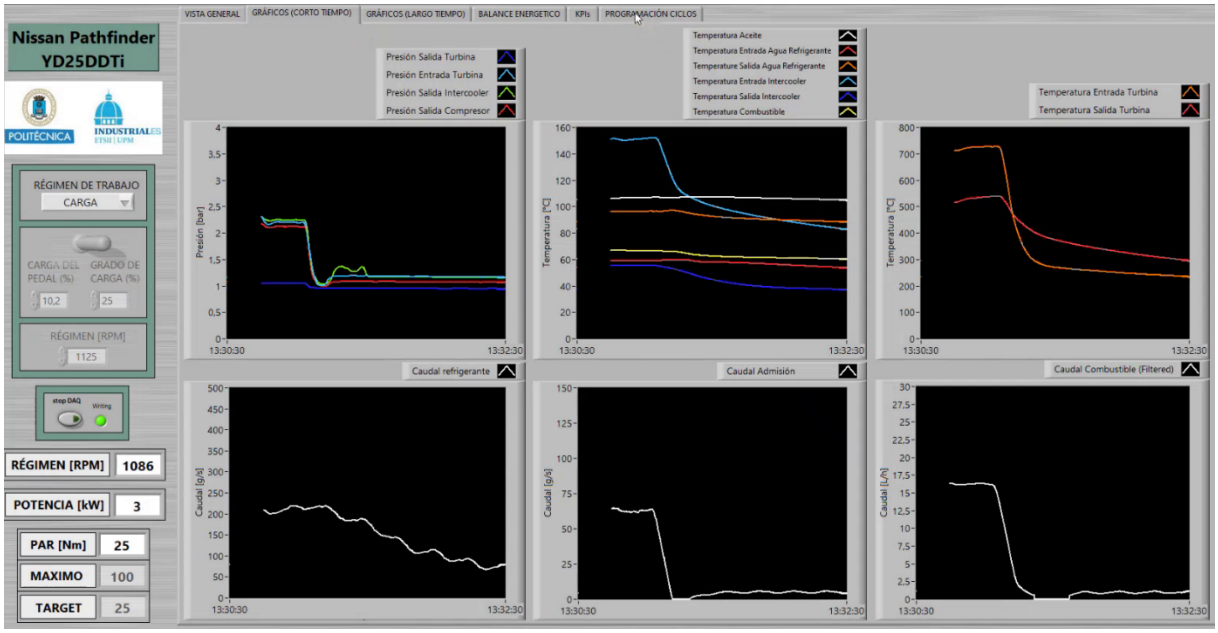

Figure 17: Short-term graphs' tab

In this tab, we have six different graphs which from the top left corner and proceeding from left to right are:

- Pressure graph which shows, referring to the legend from top to bottom, the pressure at the turbine outlet, turbine inlet, intercooler outlet, and compressor outlet.

- Temperature graph displays measurements at various points within and outside the engine. These include oil temperature, coolant temperature at the inlet and outlet, intercooler inlet and outlet temperatures, and fuel temperature, with values ranging from 0 to 160°C.
- Exhaust gas temperature graph depicts temperatures at the turbine inlet and outlet. These are presented separately because exhaust gas temperatures, having undergone the combustion phase in the engine, are higher with greater variability, sometimes reaching up to 800°C.
- Two graphs represent mass flow rates of the refrigerant and intake air, located at the bottom row, left and centre, respectively.
- A single graph represents the volumetric flow rate of the fuel.

The graphs in the figure illustrate the transition between two engine operating conditions: one at 1550 rpm with a 100% throttle, and the other at 1100 rpm with a 25% throttle.

As for the second screen displaying engine variable graphs, a different approach was chosen. This time, the focus is not on transitions between two conditions but rather on illustrating how engine parameters such as temperatures, pressures, and flow rates change as engine speed and power output vary. These graphs have a time range of 15 minutes because, over a longer period, the engine will experience a wider range of equilibrium conditions, allowing the observation of each variable's evolution with changing operating conditions. In the following figure, we provide an example of the *GRAFICOS (LARGO TIEMPO)* or “Graphs (Long-term)” tab:

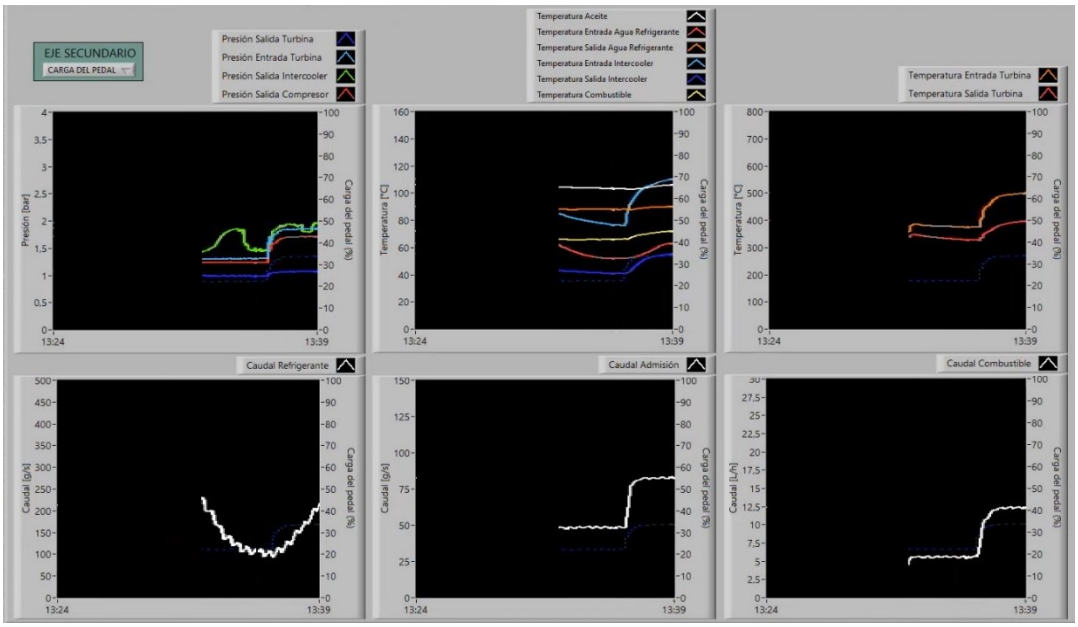

Figure 18: Long term graphs' tab

The screen in Figure 18 is very similar to that of the short-term graphs. All the graphs represent the same variables in the same positions, but over a longer time span. The other difference is that the graphs on this screen also have the ability to represent either the percentage of throttle pressed or the engine speed over time on the right secondary axis. The user can choose which of the two to represent through the dropdown menu on the top left corner labelled *EJE SECUNDARIO* (SECONDARY AXIS in English): the selected parameter will be displayed on each graph in dashed blue. This detail proves to be particularly effective when conducting tests with one of the two control variables, either speed or pedal percentage, held constant. For example, if the engine speed remains constant and we vary the pedal percentage, it is easier to appreciate the effect of the variation of this control variable on all the parameters measured by the sensors, hence on the engine. Conversely, when both speed and pedal percentage vary, it is difficult to determine which of the two variations caused the measured variables on the engine to change.

3.5. Calculation of the KPIs of the engine

Another fundamental aspect to which we wanted to give great importance was the possibility of visualizing some Key Performance Indicators (KPIs), which are quantitative indicators used to measure and evaluate the effectiveness and efficiency of an engine in terms of performance. Indeed, there are many variables measured by the engine and displayed on the user interface which are often challenging to interpret in real-time. Take, for example, the fuel flow rate consumed by the engine, and suppose we reduce the load percentage from 80% to 50%. It is clear that the fuel flow rate will decrease as the power has dropped by 37,5%. However, during the practical exercise, students may not be able to interpret whether this has decreased in percentage more or less than the engine power.

For this reason, it is important to calculate some Key Performance Indices (KPIs) of the engine to be able to assess the engine's performance live and convey the maximum amount of information to the students. Regarding the example just mentioned, there are two KPIs that can provide much more information than just the fuel flow rate. The first is the Specific Fuel Consumption (SFC), which is the ratio of the mass flow rate of fuel $\dot{m}_f$ to the output power P_m:

$$SFC = \frac{\dot{m}_f}{P_m}$$

expressed in g/kWh and indicates the mass of fuel used to produce one kWh of energy by the engine. Similarly, we can define the effective efficiency as:

$$\eta_e = \frac{P_m}{\dot{m}_f \cdot H_c}$$

where H_c is the diesel's heating value. Hence, the effective efficiency is inversely proportional to the specific fuel consumption.

The third KPI which is represented is the volumetric efficiency, defined as the ratio between the mass of air aspirated by the engine and the theoretical quantity of air that the engine would aspirate under ideal conditions. It is a measure that indicates how effectively an engine draws in and expels air during the working cycle. This parameter has a value between 0 and 1: the closer it is to 1, the better the engine can completely exhaust the combusted gases and refill the chamber with fresh air.

In ideal conditions, the engine aspirates air mass at each revolution is:

$$m_{a,th} = V_t \cdot i \cdot \rho_{a,ref} = V_t \cdot i \cdot \frac{p_{ref}}{R \cdot T_{ref}}$$

Where V_t is the total engine displacement, i is a factor equal to ½ to account for the fact that the engine is 4-stroke, so the intake phase occurs every 2 revolutions of the crankshaft, while $\rho_{a,ref}$, p_{ref} e T_{ref} are density, pressure, and temperature at the outlet of the intercooler downstream of the supercharging system and at the combustion chamber inlet. At this point, the volumetric efficiency can be calculated as:

$$\eta_v = \frac{\dot{m}_{a,real}}{\dot{m}_{a,theoretical}} = \frac{\dot{m}_{a,real}}{V_t \cdot i \cdot n \cdot \frac{p_{ref}}{R \cdot T_{ref}}}$$

With *n* being the engine speed. The displacement V_t, the *i* factor and the gas constant *R* are all known in advance, while for the air mass flow rate $\dot{m}_{a,real}$, the speed *n* and the pressure and temperature at the outlet of the intercooler, we have sensors installed on the engine that measure these quantities instantaneously. This way, we can calculate and display the trend of volumetric efficiency over time.

The fourth and last KPI is the mean effective pressure (MEP), which represents the average pressure inside the combustion chamber during the engine's working cycle. MEP is related to the engine's combustion efficiency: a higher MEP indicates a higher amount of energy released per stroke per unit volume. From a calculation perspective, MEP is calculated as:

$$mep = \frac{P_m}{V_t \cdot i \cdot n} = \frac{2\pi \cdot T_m}{V_t \cdot i}$$

where T_m is the torque.

At this point, we have defined and explained exhaustively all four KPIs that we want to display on the dedicated screen. Now, let's take a look at an example of this screen:

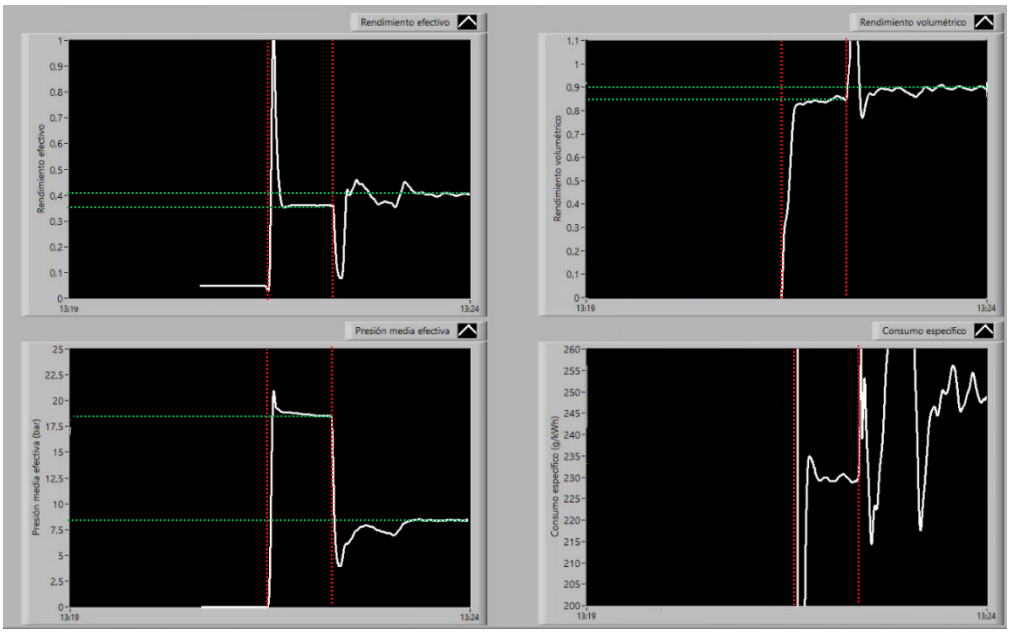

Figure 19: KPIs tab

In this example, the engine goes through three different conditions which are separated on the figure by the vertical red line. Firstly, it is idle mode, hence the mep (*Presión media efectiva*), the effective efficiency (*Rendimiento efectivo*) and the volumetric efficiency (*Rendimiento volumetrico*) are zero or close to zero and the specific consumption (*Consumo especifico*) is very high. Then, the engine goes to 2400 rpm and 100% load for 50 seconds and finally it is brought at 25% at the same speed. For this transition from 100% to 40% load at 2400 rpm, we can observe the following variations in the KPIs

- A small increase in the effective efficiency. From this, we can infer that the engine, in the second condition, operates in more optimized conditions closer to the conditions of maximum efficiency.
- A small increase in volumetric efficiency. This is probably due to the fact that the point of maximum volumetric efficiency is near the point of maximum engine efficiency, and therefore, we have approached better conditions regarding the filling of the chamber too.
- A significant decrease in the mep resulting from the decrease load demand.
- In this case, no information can be obtained from the specific fuel consumption due to its significant variations.

These pieces of information would not have been possible to assimilate with the sole visualization of the parameters detected by the sensors. However, through the use of the visualization of KPIs over time, we have managed to convey them with great simplicity and clarity.

These four graphs were created on the block diagram:

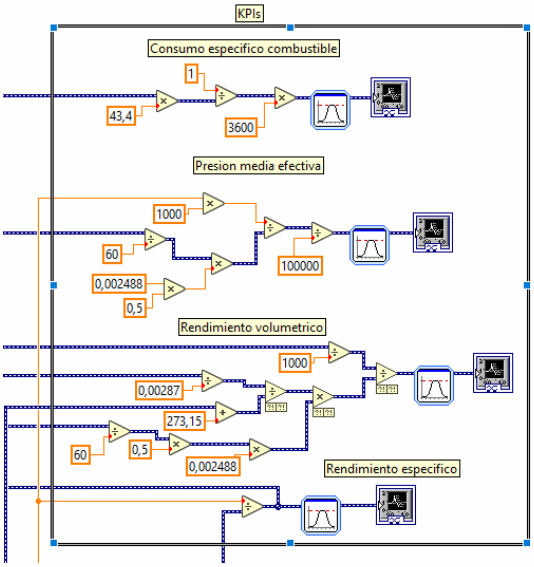

Figure 20: KPIs calculation on Block diagram

The blue lines coming from the bottom and left of Figure 20 are the values of the physical parameters measured by the sensor on the engine. This data is processed through operations to recreate the 4 equations we showed above for each of the KPIs. Finally, the values are filtered and sent to the graph icons to create the graphs in Figure 19.

3.6. Engine energy balance

Another important function that has been added to the LabVIEW program is the implementation of a tab for in real-time engine's energy balance, i.e., how the internal energy of the fuel supplied to the engine transforms into other forms of energy. To conduct the energy balance, it is necessary to choose a control volume and identify the mass flows and energy exchanges that pass through the control volume and apply the first law of thermodynamics. We choose the control volume that covers all the fluid passing through the engine, from intake to exhaust. The energies exchanged with the fluid

from the outside are responsible for the difference in energy between the incoming and outgoing flows.

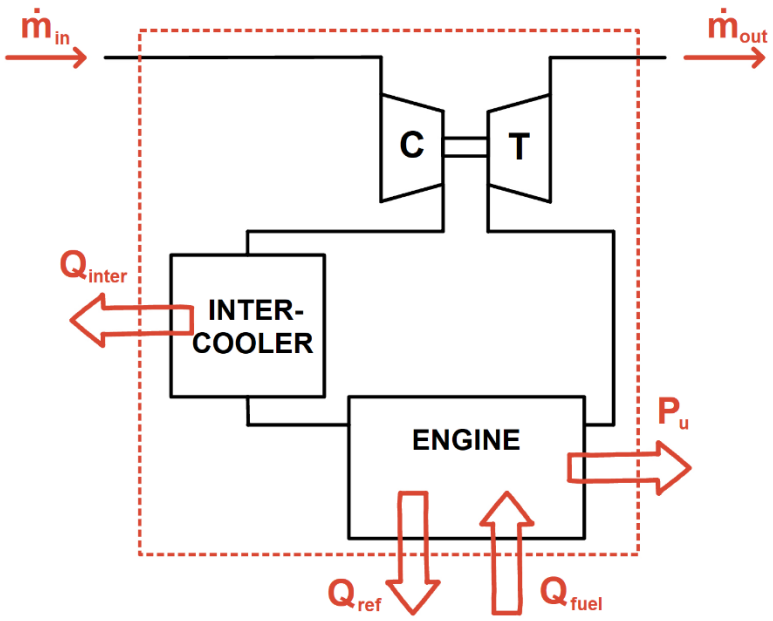

Figure 21: Engine's energy balance

Let's proceed to calculate the value of each of these identified energy flows in the figure. The power supplied to the control volume will be equal to the heat generated by the combustion of the fuel:

$$\dot{Q}_{fuel} = \dot{m}_f \cdot H_c$$

Ideally, the engine's goal is to transform the highest portion of this energy into useful power at the output shaft P_u. However, there are some losses that cannot be eliminated, and in our case, the main ones are the heat transferred to the coolant, in the intercooler, and the heat lost in the exhaust gases. The latter is not precisely heat transferred to the external environment, but it is a term to account for the fact that the gas flow at the exit of the engine $\dot{m}_{out}$, and therefore at the exit of the control volume, is hotter than the inlet flow $\dot{m}_{in}$. Since this is not a desired effect, we consider it as a heat loss. Let's calculate mathematically the three main causes of energy dissipation.

The power transferred to the coolant, which in this case is a mixture composed mostly of water with some additives, is calculated as:

$$\dot{Q}_{refrigerant} = \dot{m}_{ref} \cdot c_{p,ref} \cdot \Delta T_{ref}$$

where $\dot{m}_{ref}$ is the mass flow of the refrigerant liquid, $c_{p,ref}$ is its specific heat capacity and has a value of 3600 J/kgK, while ΔT_{ref} is the difference in temperature between the coolant at the exit and that at the entrance to the engine. The heat loss to the coolant can result in a loss of 20-35% of the power provided by the fuel.

The other main cause of power loss is the heat lost from the exhaust gases. This loss can account for 22-35% of the provided heat. This heat is calculated as:

$$\dot{Q}_{eg} = \left(\dot{m}_{air} + \dot{m}_{fuel}\right) \cdot c_{p,eg} \cdot \Delta T_{eg}$$

with $c_{p,eg}$ the specific heat capacity of the exhaust gases and ΔT_{eg} the temperature difference between the exhaust gasses at the outlet and the environment. To obtain the value of the specific heat capacity of the exhaust gasses, we assumed that it was the weighted average on the mass of the specific heat capacities of the substances that make up the exhaust gases. To do this, it was necessary to study the combustion reaction of diesel in the combustion chambers:

$$C_{12}H_{23} + \frac{71/4}{\lambda}(O_2 + 3{,}76N_2) \rightarrow 12CO_2 + \frac{23}{2}H_2O + \frac{3{,}76 \cdot \frac{71}{4}}{\lambda}N_2 + \frac{71}{4} \cdot \left(\frac{1}{\lambda} - 1\right)O_2$$

We observe that the composition of the exhaust gases is not constant but depends on the air-fuel equivalent ratio λ. From this, we deduce that the value of the specific heat of the burned gases will change depending on the operating conditions of the engine. It was then assumed that the gases forming the exhaust gas mixture have constant specific heat capacities and are not dependent on the exhaust temperature, and they have the following values:

Gas	C_p (J/kgK)
CO_2	942
H_2O	2014
N_2	1046
O_2	942

Table 4: Specific heat capacities of exhaust gases

With the specific heat capacities of all the components of the exhaust gases, we can derive the formula for the specific heat capacity of the exhaust gases as a function of the air-fuel equivalent ratio:

$$c_{p,eg} = 1023{,}6 + 78{,}786 \cdot \lambda$$

The last important loss to consider for the energy balance is the heat exchanged in the intercooler. The exchanged heat, having a thermocouple both before and after the intercooler, can be calculated as:

$$\dot{Q}_{intercooler} = \dot{m}_{air} \cdot c_{p,air} \cdot \Delta T_{intercooler}$$

where $\dot{m}_{air}$ is the mass flow of air, $c_{p,air}$ is the specific heat capacity of air with value 1007 J/kgK and $\Delta T_{intercooler}$ is the air's temperature difference between before and after the intercooler.

At this point, we have defined the three forms of energy dissipation with the highest values and that are easier to calculate. However, the sum of these three and the useful power does not equal the power supplied to the system because there are other components that we have not taken into consideration, such as:

- Losses due to incomplete combustion of fuel in the combustion chamber, which amount to a few percentage points of the total power and are challenging to calculate as it is not possible to directly measure the amount of unburned fuel that does not participate in the reaction. This would require sophisticated tools not present in this test bench.
- Losses due to heat exchanged by convection and radiation from the engine, amounting to 2 - 6% of the total power. The engine has a complex geometry, resulting in non-uniform temperature distribution, a non-uniform heat exchange coefficient with air along the surface, and a variable airflow over the engine. For these reasons, mathematically calculating this form of dissipation is very complicated.
- A component present during transients, due to the fact that the engine is not in equilibrium but with flows, temperatures, and all other parameters evolving over time.

From this, we conclude that in the energy balance equation, we will have a component that we will call *Resto* or "Remainder" to take into considerations all these components that we are unable to calculate singularly. This portion is obtained through the difference between the total energy supplied by the fuel and the other four forms of energy.

The final equation for the energy balance will be:

$$Q_{fuel} = P_{useful} + Q_{refrigerant} + Q_{eg} + Q_{intercooler} + REMAINDER$$

Now let's observe an image of the screen displayed on LabVIEW for the energy balance:

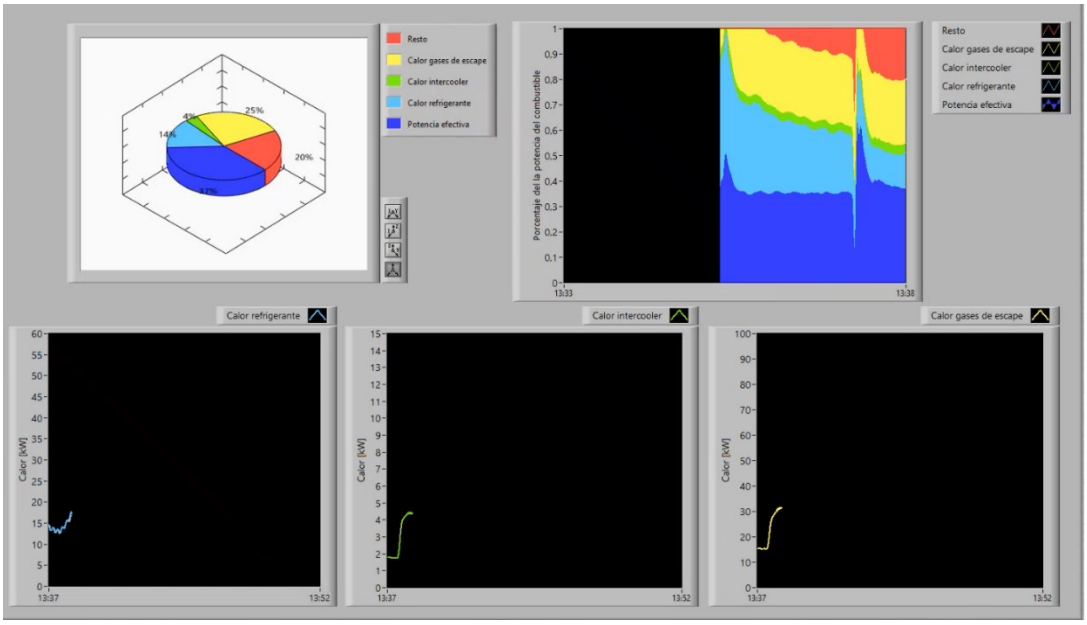

Figure 22: Engine energy balance graphs' tab

On the top row we have two graphs representing the energy balance of the engine. The left one is a pie chart to visualize the instantaneous energy distribution. The 100% of the pie chart is the energy supplied by the fuel and the slices of the pie are: in blue, the useful power; in yellow, green, and light blue, the three calculated dissipation heats mentioned earlier, while in red, the slice of the remainder is represented.

In the right graph, we have the engine balance focused on the temporal evolution of energy components. This graph is scaled relative to the total power, so it does not provide information on the absolute power over time and between two different operating conditions but rather on the proportion of each component to the total.

The three lower graphs provide the missing information to the two upper graphs, which are the absolute values of the heats of the coolant, intercooler, and exhaust gases.

These graphs are obtained by implementing the formulas for each heat dissipation and for the remainder on LabVIEW, substituting all the constants with their actual values and connecting with

appropriate graphical functions the data acquired from the engine. The block diagram part relative to the engine energy balance is the following:

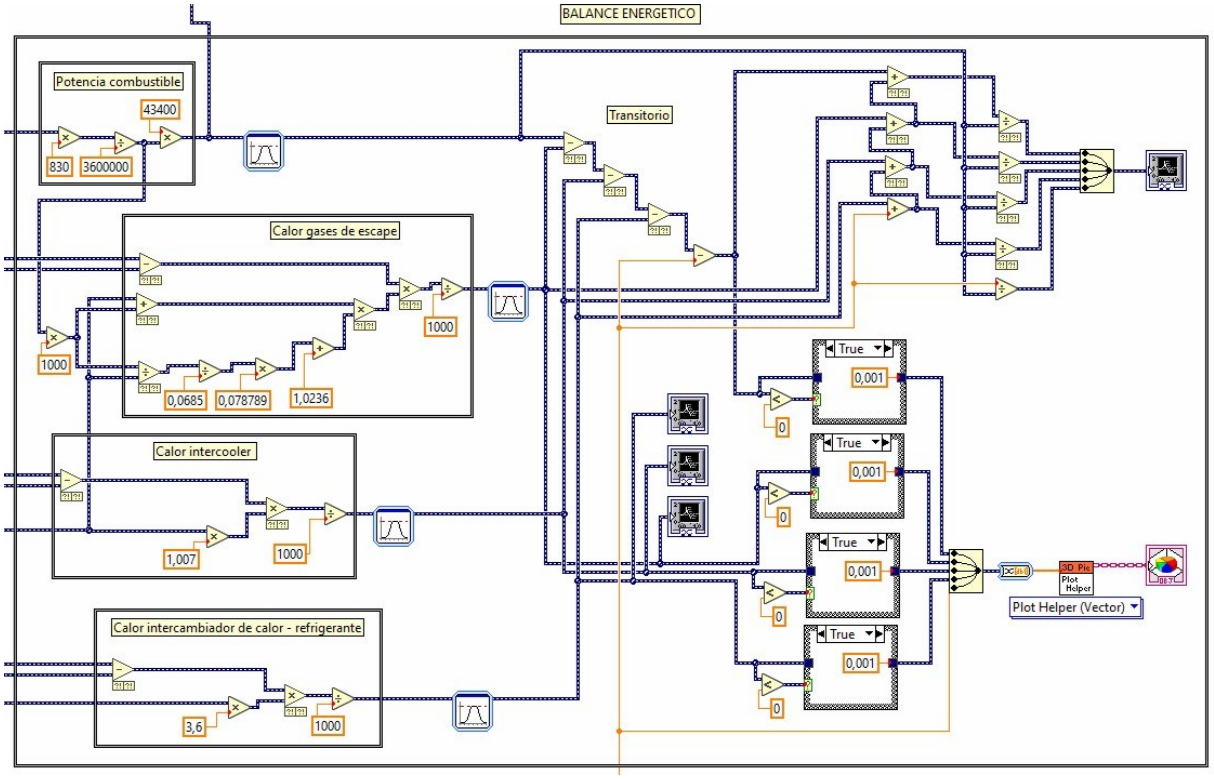

Figure 23: Engine energy balance on Block diagram

3.7. Improvement to engine control

In Chapter 2.4, we saw the front panel and observed that to choose the engine operating point, we have the option to control the speed and the percentage of the pressed pedal. When a driver is behind the wheel of a car, they press and release the accelerator pedal to maintain or change the speed of their vehicle according to their needs and the situation. The vehicle driver can be satisfied with this way of controlling their vehicle. In the context of a test bench, where tests are conducted to measure engine performance, controlling it with the pedal percentage can be limiting in many respects. Thus, we aim to improve the engine control in such a way that we can choose the engine operating point not only by selecting the pedal percentage but also by choosing the engine load percentage.

To do so, let's review how the power control of this specific engine works. Power is indirectly changed by controlling the percentage of the pressed pedal. This is achieved by sending a voltage signal directly to the powertrain control module (PCM) within the Engine Control Unit (ECU), equal to the signal the physical pedal would send if pressed by the same percentage. The physical pedal, in reality, doesn't send just one signal but two—in our case, one being double the other. Having two signals instead of just one allows for a safety measure in case there is a malfunction in one signal. These signals have the following trends:

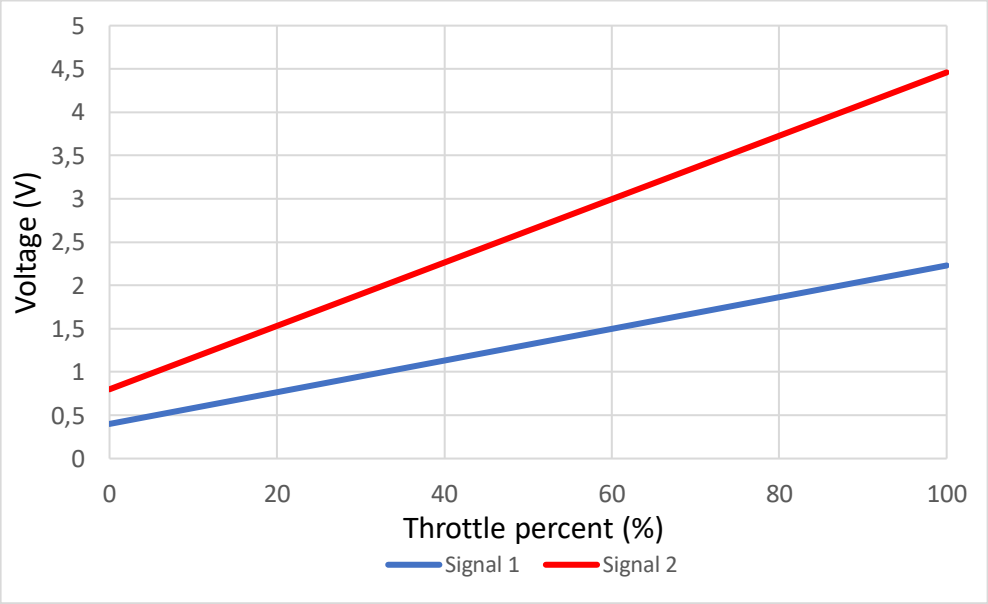

Figure 24: Correlation between throttle percent and output signal

Given *TH%* as the throttle percent chosen by the user, the two voltage signals that must be sent to the ECU will be:

$$V_1 = 0{,}4 + 0{,}0183 \cdot TH\%$$

$$V_2 = 2 \cdot V_1 = 0{,}8 + 0{,}0366 \cdot TH\%$$

As a result of these input signals, the PCM acts on the injection system and consequently the delivered power changes.

Let's now observe the relationship between the percentage of the pressed pedal and the percentage of engine power at a rotational speed of 2500 rpm:

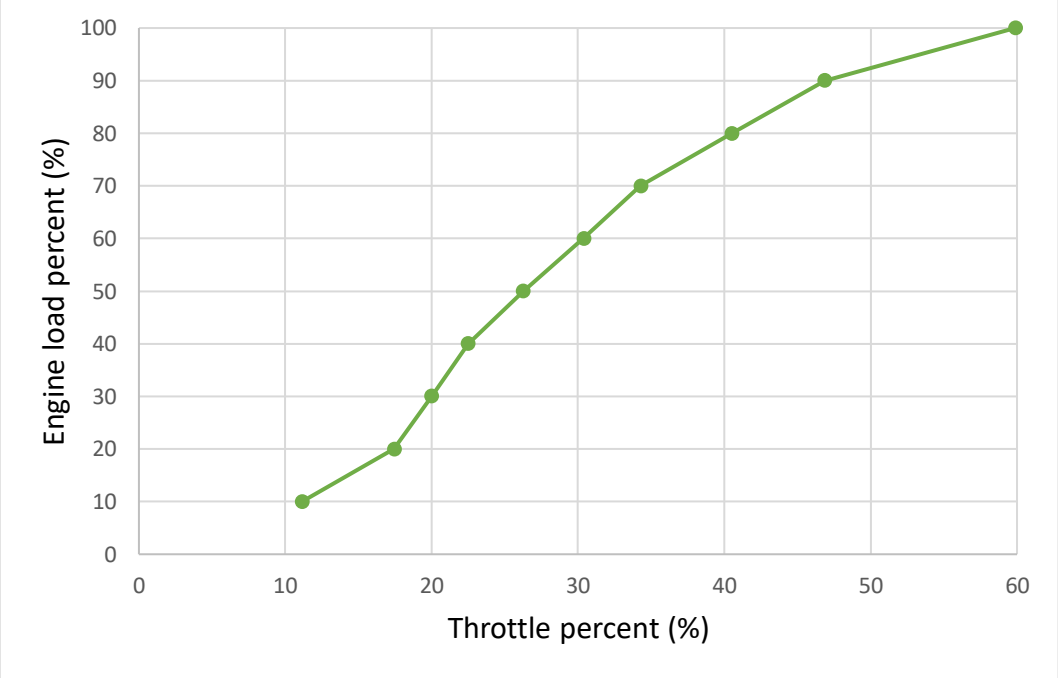

Figure 25: Throttle-load percent correlation

Notice that the relationship between these two quantities is not linear. Instead, the curve has a very low slope below 20% load, while above the slope is bigger and progressively decreases. A lower slope of the curve indicates that the driver, for the same load variation, will have to vary the percentage of the pressed pedal more, which is less comfortable. Hence the highest slopes are in the central part of the curve where the engine is mostly used. Also, note that for a pedal percentage of 60%, the engine already delivers 100% of power. Further increases in the pedal percentage would have no effect on the engine's operating conditions.

Now let's observe how the relationship between the load percentage and the pedal percentage varies with the engine speed:

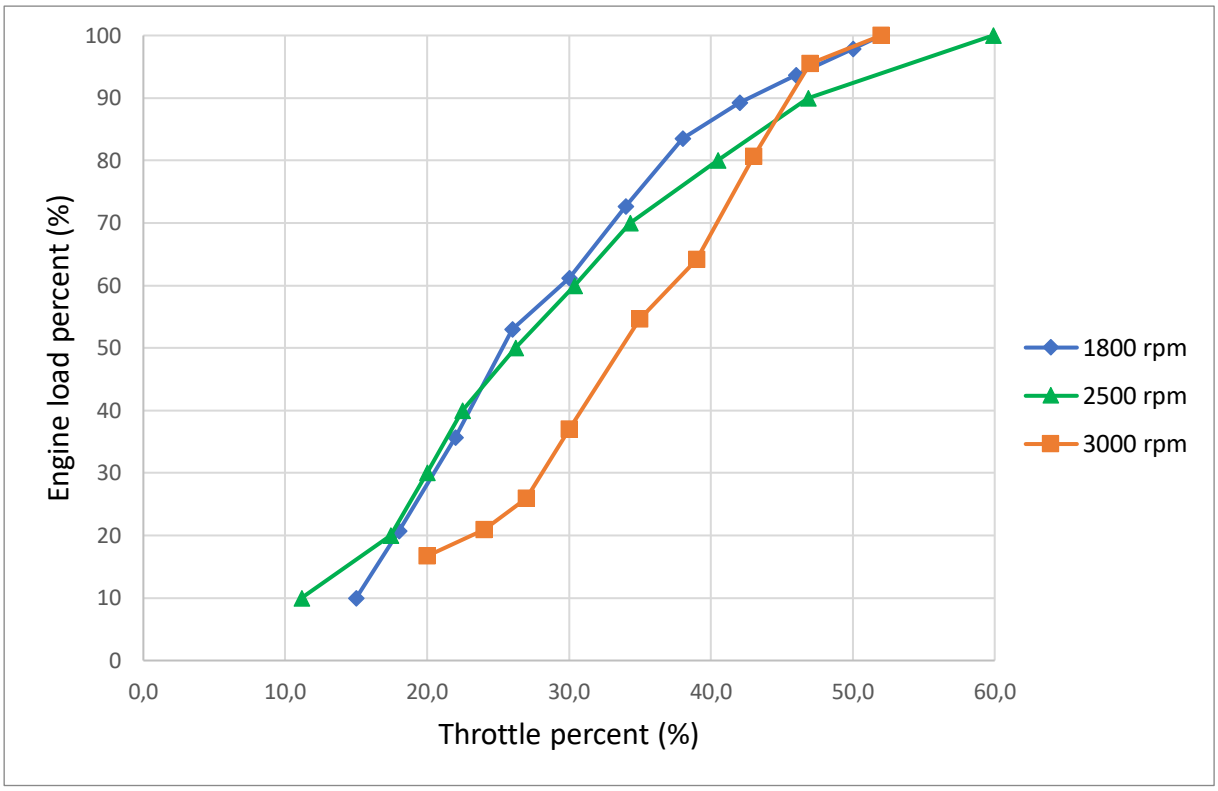

Figure 26: Throttle-load percentage correlation at different speeds

From this graph, we can see the significant variation in the curve's behaviour with the engine speed. With a pedal percentage of 26% at 3000 rpm, we obtain an engine load percentage of 26%, while at 1800 rpm with the same pedal percentage, we have an engine load percentage of 54%, double the value. Note that the minimum pedal percentage for achieving 100% power is 60% at 2500 rpm, but it drops to almost 50% at 1800 and 3000 rpm.

Moreover, these data refer to specific operating conditions, so the curves might change even at the same speeds for different tests. During transients or with a warm or cold engine at startup, the curves will have a different trend from those just shown. For this reason, it is clear that it is not possible to find a unique equation that relates engine load and pedal percentage. Instead, a method must be implemented to intervene on the engine during its operation to adjust the pedal percentage to bring the engine to the desired load percentage.

The first method attempted to perform this task was the PID method, implemented with a function directly from LabVIEW; this method, for various reasons that will be discussed later, did not work, and for this reason, it was necessary to implement an algorithm that performed a task similar to that of PID but simpler. Each of these two methods and decision-making processes will be discussed in the following paragraphs.

3.7.1. PID method

The PID control, an acronym for Proportional-Integral-Derivative, is a feedback control method aimed at bringing and maintaining a system variable (the controlled variable) as close as possible to a desired reference value. The PID constantly evaluates the difference between the reference value and the controlled variable, generating a control signal based on three main components:

- Proportional (P): This component generates a control signal proportional to the instantaneous error. In other words, the larger the error, the larger the applied correction.
- Integral (I): This component keeps track of past errors and accumulates a correction over time. This helps eliminate steady-state errors, ensuring that the system reaches its desired position without persistent oscillations.
- Derivative (D): This component is proportional to the rate of change of the error. It serves to prevent excessive oscillations by providing an early correction when the system starts to react too quickly to changes.

For each of the three components, it is necessary to define a parameter, namely the Proportional, Integral, and Derivative Gains. These parameters depend on the nature of the system, such as stability, precision, settling time, and oscillation attenuation. Choosing the correct gain values is crucial in a PID algorithm as it allows for a perfectly controlled variable. Conversely, incorrect gain values for the studied system can result in poor control quality, such as slow reaching of desired conditions and long transients or, worse, a condition of strong instability leading to rapid system oscillations.

A block diagram of a PID control algorithm is shown in the figure below[1]:

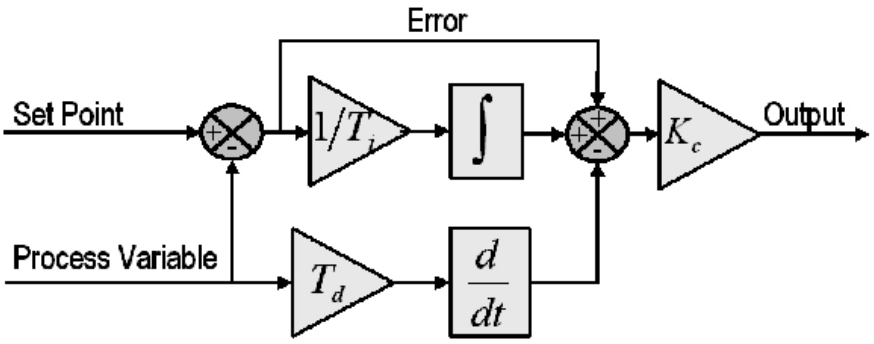

Figure 27: PID's block diagram

In our case, we have the load percentage set to a specific Set point and a Process variable that is the instantaneous load percentage. The PID algorithm outputs the signal voltage which is sent to the ECU for controlling the pedal percentage. In this way, we have gained the advantage of completely separating the engine control from the throttle percentage-load percentage curve, which, as discussed earlier, is challenging if not impossible to obtain.

Let's now observe how the PID function works in LabVIEW[1]:

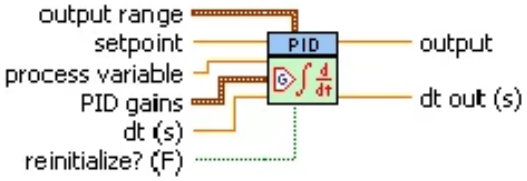

Figure 28: PID function on LabVIEW

In input, the function requires:

- The range of the output signal, which is the range within which the output signal can vary. In our case, the range is [0.4, 2.4] because Signal 1 for 0% and 100% of throttle percentage has these values, as observed in Figure 24. The second signal to be sent to the ECU will simply be double the first one.
- The setpoint, namely the target value of the load percentage that the user chooses directly from the front panel.
- The process variable, which is the instantaneous value of the load percentage. This is calculated by dividing the instantaneous power measured by the test bench by the

maximum power at the operating engine speed. To do this, the curve of maximum power was obtained for the entire range of engine speeds.

- The 3 PID gains.
- The *dt* interval between each recalculation of the process variable and, consequently, the frequency with which the output value is updated, which we set to 4 seconds as it was a reasonably small amount of time interval for the engine to reach the new condition.

In output, the PID function will give the voltage of the signal to be sent to the engine's ECU to control the pedal percentage.

The values of the 3 PID Gains still need to be calculated. We mentioned earlier the importance of choosing correct values of the gains, but this is as important as it is challenging, especially in such a complex and variable system as an engine test bench. As the first method to evaluate these values, the Ziegler-Nichols method was used. This method consists in setting the integral and derivative gains to zero and conducting setpoint change tests starting with a small value of the proportional gain and gradually increasing it. When the system, for a certain proportional gain, does not reach the setpoint but tends to have stable and consistent oscillations (of period T_u), the proportional gain is recorded, and it is denominated the critical gain K_c. The values of the 3 gains will be:

$\mathbf{K_p}$	$\mathbf{K_i}$	$\mathbf{K_d}$
$0{,}6\ K_c$	$0{,}5\ T_u$	$T_u/8$

Table 5: Ziegler-Nichols' gains

These are first-trial values and therefore need to be confirmed for the analysed system and adjusted through a trial-and-error process to reach the optimal solution. Some tests were conducted to try to find the critical gain, but what was observed was that this value was not constant; it varied with the engine operating conditions. Depending on the initial and especially the final conditions of the transient, the critical gain and the period of stable oscillations changed. For this reason, several values of the PID gains obtained with the Ziegler-Nichols method were tried and modified. What was noticed is that they worked well under certain operating conditions, especially those similar to the test

conducted with the Ziegler-Nichols method to obtain those values. However, for very different conditions, the values resulted in a highly unstable response, which is unacceptable for the studied application.

Another way to obtain PID gains is through a manual trial-and-error process. To do this, it is necessary to understand the effects that each gain's variation on the PID. A summary of this is shown in the following table[2]:

Change	Rise time	Overshoot	Settling time	Steady-state error	Stability
Increase K_P	Decrease	Increase	Small Increase	Decrease	Degrade
Increase K_I	Small Decrease	Increase	Increase	Large Decrease	Degrade
Increase K_D	Small Decrease	Decrease	Decrease	Minor change	Improve

Table 6: Effects of PID gains on transients

This method is certainly more burdensome and requires more time to find the ideal combination of the three gains, and it may not always lead to a solution. In our case, many attempts were made by varying the three gains over a wide range of values, taking into account the effects reported in Table 6. However, these attempts were unsuccessful because it was possible to find values that worked well for certain operating conditions but not for all others. Below are two tests in two different operating conditions using the same PID gains. The first graph represents a condition where the PID gain values are almost optimal:

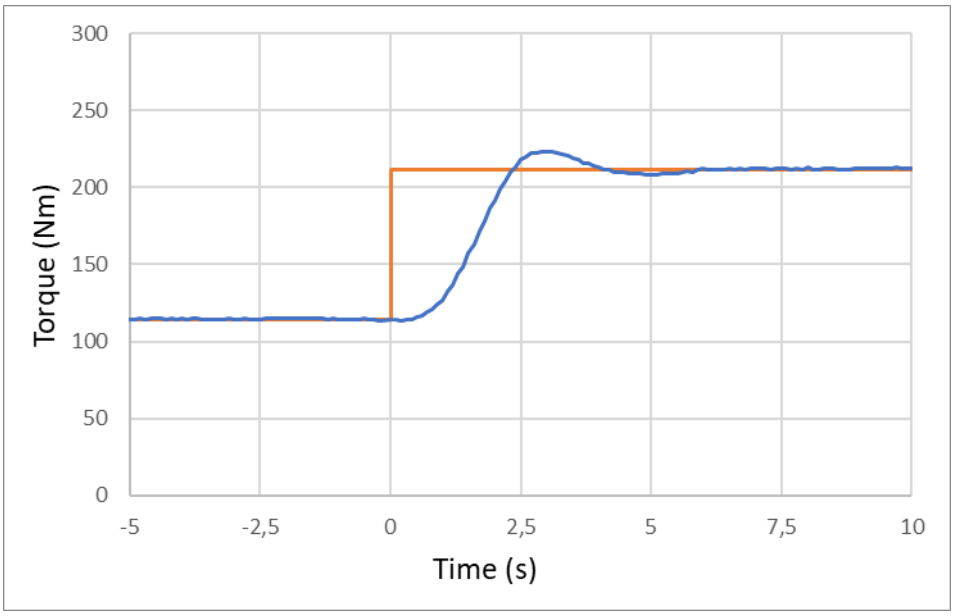

Figure 29: Example 1 of PID transient

At time t = 0 s, the engine is delivering a torque of 115 Nm, and the user requests a torque of 210 Nm. The PID, at this moment, adjusts the signal for controlling the engine percentage and keeps updating the output voltage according to the feedback, and by doing so the engine increases the delivered torque, slightly overshooting, and finally stabilizing at the requested value. In this case, the PID worked perfectly, but unfortunately it will not behave the same way in all situations. If a similar test is performed, with the same PID gain values, at 1200 rpm and transitioning from a torque of 20 Nm to 40 Nm, the result is as follows:

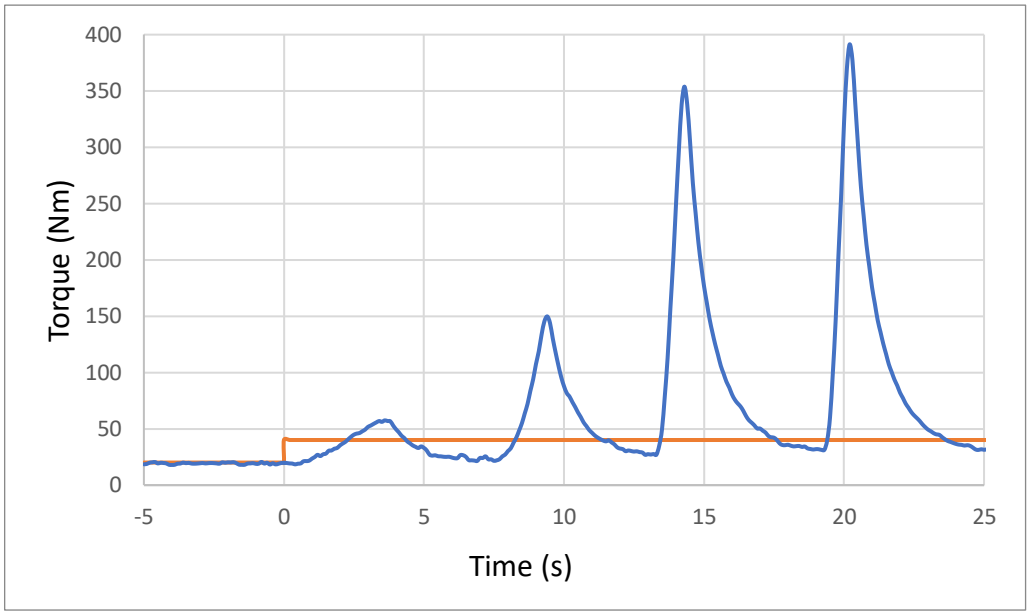

Figure 30: Example 2 of PID transient

In this case, the PID algorithm fails to transition to the desired condition, initiating very rapid and diverging oscillations from the target condition.

The last method attempted to overcome the issue of not being able to find PID gain values that satisfy all conditions was using LabVIEW's PID Autotuning function, which automatically calculates PID gains during operation and adjusts them when these values are not ideal for the operating conditions. The function appears as follows:

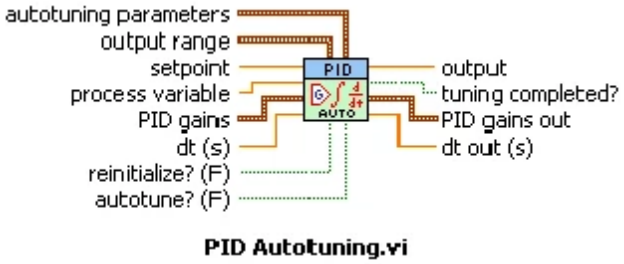

Figure 31: PID Autotuning function on LabVIEW

in which the main difference is that the function calculates the PID gains and outputs them. However, once again, the PID did not yield positive results, and some of the control tests led to imbalance situations similar to those in Figure 30.

After numerous attempts to find values that could work under all engine conditions, it was eventually decided to abandon PID control, justifying its malfunctioning with these three reasons:

- The PID is sensitive to changes in the curve between the percentage of pressed pedal and the engine load percentage studied in the previous paragraph, causing a significant shift in engine operating conditions.
- When the pedal percentage is changed, the engine speed undergoes variations, sometimes even of significant magnitude, even when PID is not used. Let's observe two striking cases:

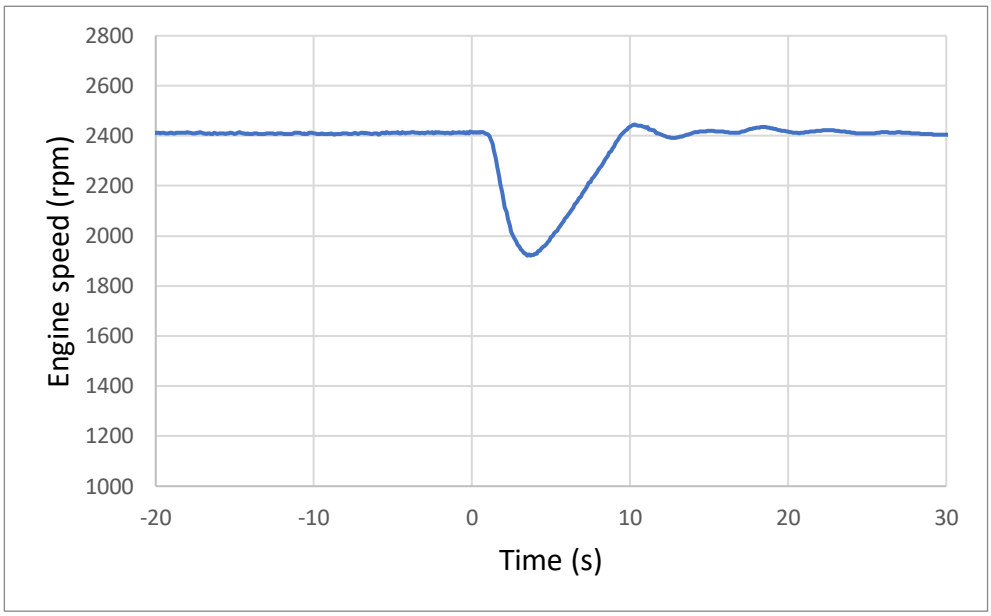

Figure 32: Temporary speed variation upon pedal percentage's variation

In the first case, the throttle percentage is changed while engine speed is kept constant, so the input signal to the dynamometer remains the same. In theory, the speed should remain the same, but it is observed that the speed can change, sometimes more, sometimes less, and return to its nominal value after a short period of time. In this case, the speed should be fixed at 2400 rpm, but decreases to almost 1900 rpm.

In other cases where both the pedal percentage and speed are modified, the speed experiences significant overshooting, as in the following case:

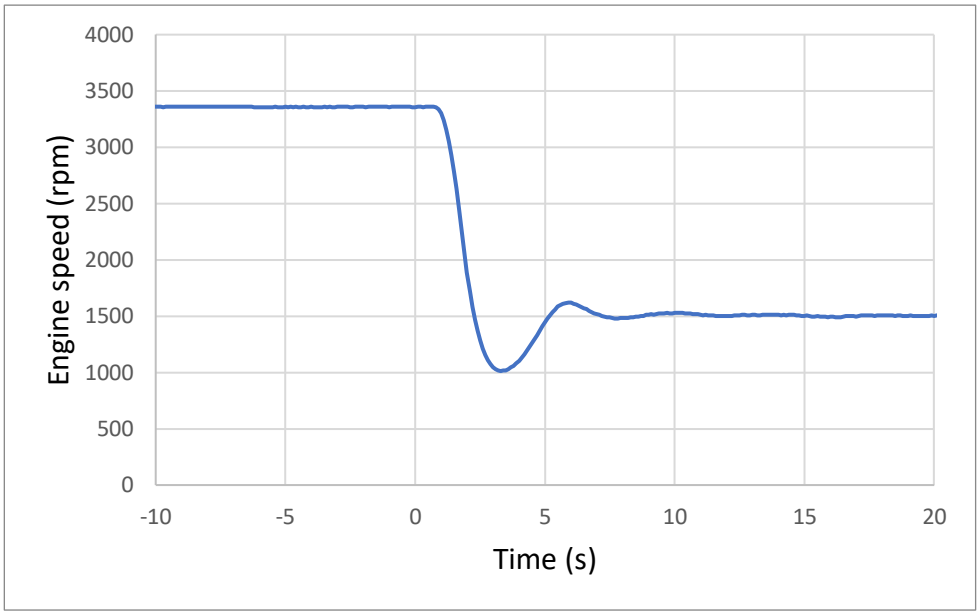

Figure 33: Speed's overshooting in transient

We observe that from a speed of 3350 rpm, the engine needs to decrease its speed to 1500 rpm, but in the transient period, the engine actually drops to 1000 rpm.

From these two graphs, it is clear that the speed also varies significantly during transient conditions, further altering the operating conditions of the engine and thus disturbing the PID algorithm.

- The PID is also less efficient in systems where there is a time delay between the signal and the effect of the signal. In our case, all signals sent to the engine are temporally filtered to have a more gradual change in conditions and avoid sudden changes that could damage the engine. However, by doing so, we delay the effect of the output signal and worsen the effect of the PID.

3.7.2. The proportional algorithm

Given that the first attempt with PID did not yield positive results, it is necessary to find another way to control the engine based on the load percentage. Any function developed in LabVIEW as an alternative to PID must have the following two very important characteristics:

- It must be able to reach the desired load percentage without entering unstable oscillation conditions, for any initial and final conditions of the engine, as was the case in Figure 30.

- The engine must be able to reach the desired condition as quickly as possible, or at least within a reasonable timeframe.

The first alternative considered was to develop a simple open-loop control algorithm where, at each iteration, the difference between the desired load percentage and the actual one is evaluated. The algorithm then adjusts the signal by an amount proportional to this difference. This method was initially chosen because it could be effective while being simple and quick to program.

Let's explain more in details how the algorithm works. When the engine is in a stable condition, and the user requests a new desired load percentage, the algorithm predicts what the corresponding throttle percentage could be for that desired load percentage. This predicted throttle percentage is referred to as the "first attempt" value. The output signal corresponding to this first attempt pedal percentage is sent to the engine, which starts adjusting to these conditions. After a short period, the engine will be at a load percentage different from the initial one, closer to the desired load but possibly different to it because the sent signal was only a first attempt. Hence, the engine calculates the percentage error between the instantaneous and desired load:

$$\%error = \frac{\%load_{required} - \%load_{istantaneous}}{\%load_{required}}$$

At this point, the algorithm needs to calculate a throttle percentage that brings the engine closer to the desired load input by the user. We decided to vary the pedal by a proportionally amount based on the error on the load percentage $\%error$. This translates to the following formula:

$$\frac{throttle_{required} - throttle_{current}}{throttle_{current}} = K \cdot \frac{\%load_{required} - \%load_{istantaneous}}{\%load_{required}}$$

Or

$$\frac{throttle_{required} - throttle_{current}}{throttle_{current}} = K \cdot \%error$$

With K being a constant. Developing the equation, we obtain the formula for calculating the required throttle percentage to be sent to the engine for the next iteration:

$$throttle_{required} = throttle_{current}(1 + K \cdot \%error)$$

Regularly, the load instantaneous load must be compared to the desired value to calculate the $\%error$ in order to recalculate the required throttle $throttle_{required}$. After a certain number of iterations, this process brings the load to the desired value, implying $\%error$ equal to 0 and a constant $throttle_{required}$.

There are still two steps left to program and test the algorithm on LabVIEW. The first one is choosing the initial guess value for the throttle percentage. It is evident that the more accurate the initial guess value is, the fewer iterations will be needed to go from the initial guess condition to the desired condition. Let's recall the relationship between load and throttle percentages:

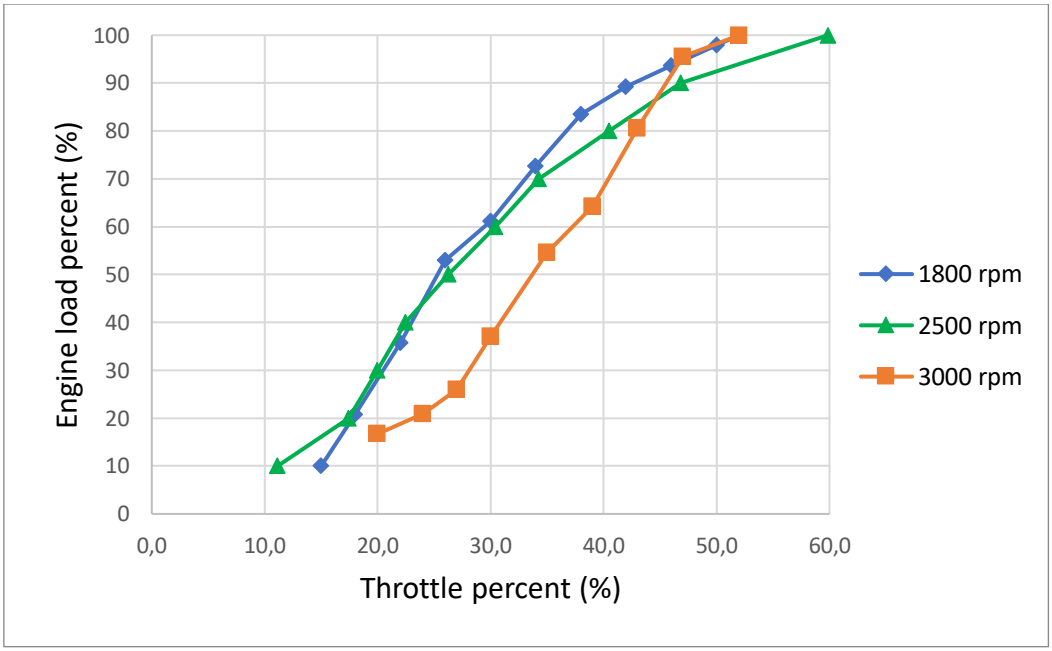

Figure 34: Throttle-load percentage correlation at different speeds

We must remember how variable and unstable these curves are, and we also recall that this variability was undoubtedly a factor that caused the PID malfunction. It is clear that hypothesizing a first trial guess for the throttle percent is very difficult, and we should expect that, in some cases, it may be very different from the real value. Fortunately, an imprecise initial guess value does not affect the stability of the algorithm but just increases the time needed to stabilize at the desired value. Consequently, we choose the initial guess value for the throttle percentage $throttle_{first\ trial}$ for a required load $load_{required}$:

$$throttle_{first\ trial} = \begin{cases} load_{required} & load_{required} < 20 \\ 0{,}33 \cdot load_{required} + 13{,}3, & 20 < load_{required} < 80 \\ 0{,}75 \cdot load_{required} - 20 & load_{required} > 80 \end{cases}$$

This first guess function approximates, in the simplest way possible, the relationship between the pedal percentage and the load percentage, as shown in the figure:

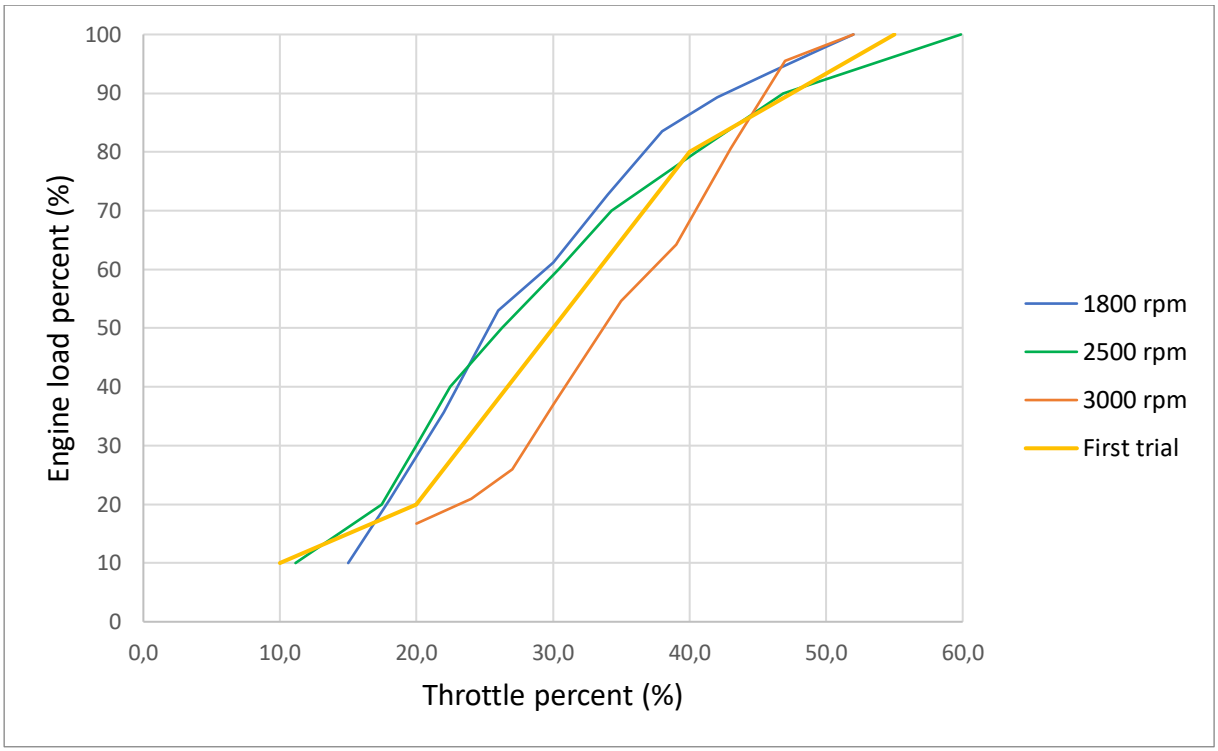

Figure 35: First-trial function

Finally, it is necessary to define a value for the proportionality constant K between the percentage error on the load and the percentage change in the pressed pedal between one iteration and the next one. This constant is closely related to the slope of the curves in the figure. If K is very similar to the reciprocal of the slope of the curve, the algorithm will approach the convergence point very quickly at each iteration. However, the algorithm becomes less efficient if the value of K is far from the optimal one. If K is lower or much lower than the inverse of the slope, the required load percentage will be reached with a high number of iterations, reducing the efficiency of the algorithm since it is slower to reach the target. If K is too high, every pedal variation from one iteration to the next one would be excessive, and at each iteration where the instantaneous load percentage is below the target, it would move above on the next iteration, while if it was above, in the next iteration, it would be below. This leads to oscillations around the target point, which in the best case slowly

converge, while in the worst case, they diverge, moving further away from the target point with each iteration.

For example, let's take 3 points on the pedal-load percentage curve at 2500 rpm with 30%, 60%, and 90% load percentage. We observed that the slope of the curve in these points decreases as the load percentage increases. From what was said earlier, we know that the optimal K will be the highest for the point at 90% load and decrease for the other points. Let's simulate how the algorithm operates the engine with a K value of 1.5 and a pedal-load curve at 2500 rpm, checking at which operating points the algorithm puts the engine into operation:

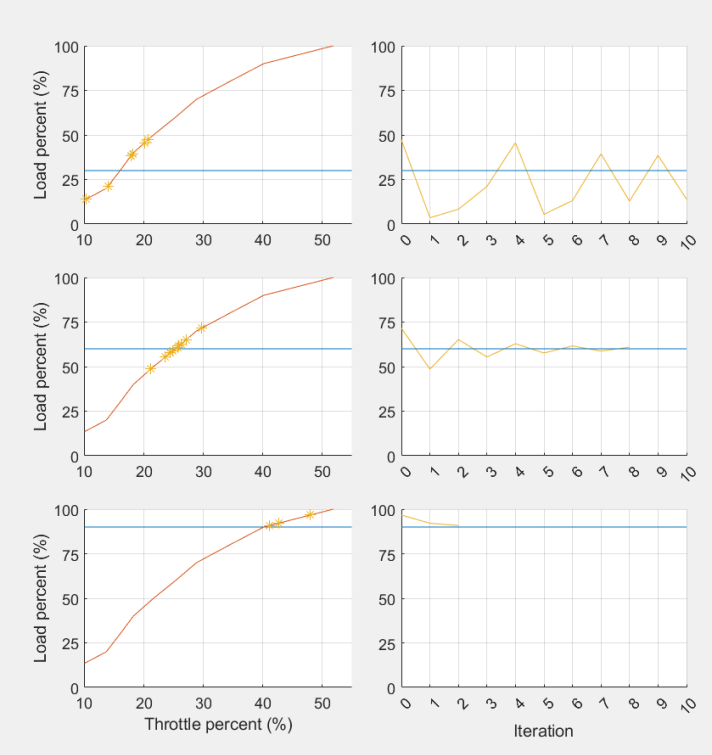

Figure 36: Algorithm's simulation with K=1.5 at 2500 rpm

On the right graphs, we have the engine load percentages at each iteration, while on the left, we have the load-pedal curve at 2500 rpm with the intermediate points of the iterations. Let's focus on the case where we set the target load percentage to 90% (chart on the last row on the right):

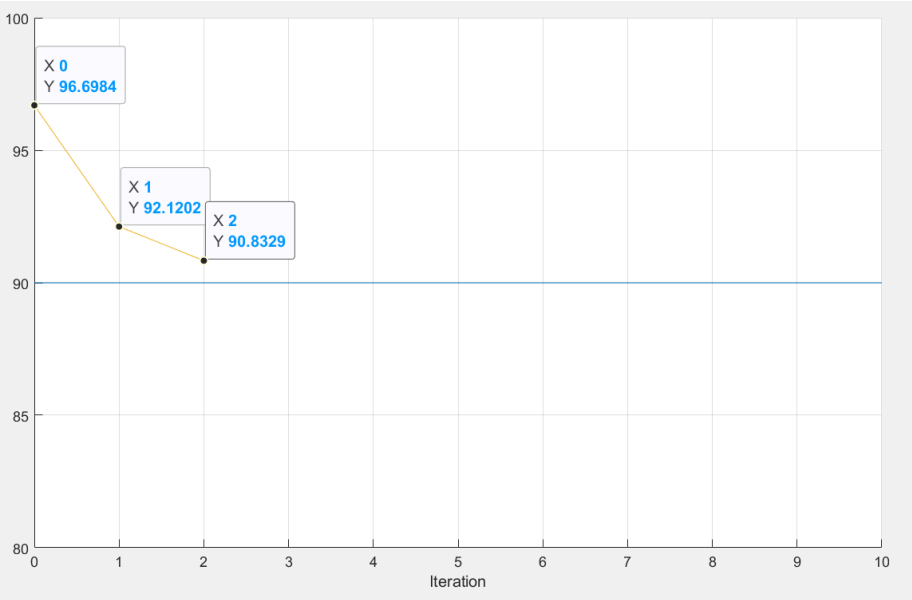

Figure 37: Iterations with 90% target load

The algorithm will provide a first attempt value (obtained from the curve equation in Figure 35) that will initially lead to a load percentage of 96.7%, at the first subsequent iteration to 92%, and already at the second iteration, we will have reached a load of 90.8% with an error of less than 1% compared to the target value. We can deduce that the K value of 1.5 is close to the optimal value for a target load pedal of 90% at 2500 rpm.

If, however, we set a target load percentage of 60%, whose slope in this point is higher, the optimal K will be lower, and as we discussed earlier, this can lead to oscillations (graph in the middle row on the right in Figure 36):

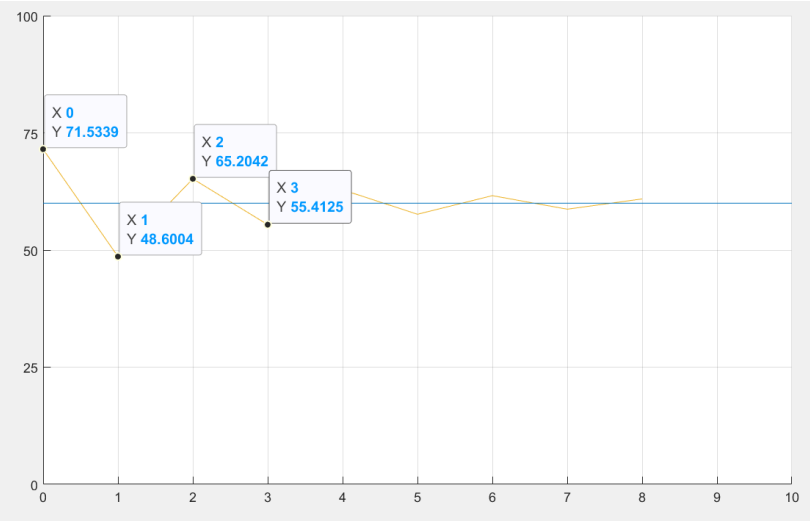

Figure 38: Iterations with 60% target load

As a matter of fact, in this case, the first attempt brings the engine to a load percentage of about 71%, higher than the target value of 60%, while the next iteration will be closer to the target value but below it. Subsequent iterations will continue to approach the target value but always shifting from above to below it or vice versa. In other words, the relative error of the load percentage decreases in absolute value, but it still changes sign with each iteration. By the eighth iteration, the engine has reached a load percentage of 60.9%, a value already sufficiently close to the target. This situation, although not ideal, is still acceptable.

Finally, let's focus on the point with the steepest slope and lowest optimal K of the three, specifically at 30% load, and observe the algorithm's simulation (graph on the top right of Figure 36):

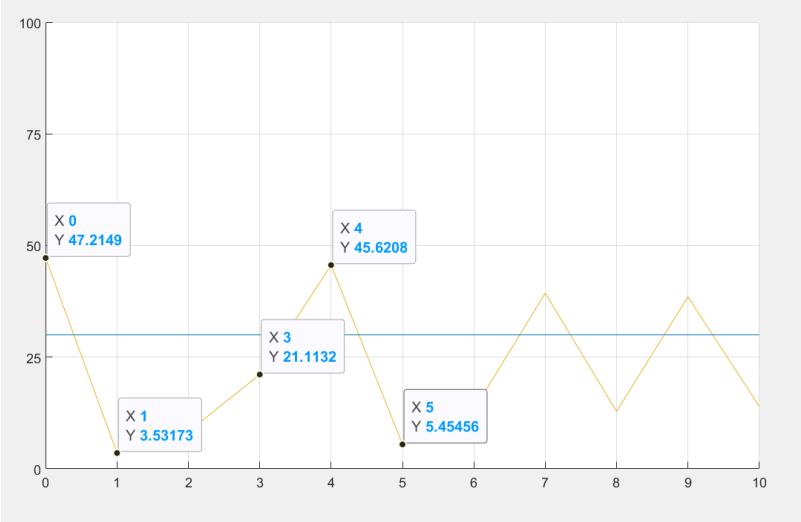

Figure 39: Iterations with 30% target load

In this case, we observe oscillations around the target value of 30% load, which, in this instance, do not converge. It can happen that between one iteration and the next one, the error not only changes sign but also increases. At the tenth interaction, the algorithm hasn't brought the engine to the desired load value yet, and it won't reach it even with a higher number of iterations.

What needs to be done is to decrease the value of K so that it is lower than all the optimal K values for the three points. Let's try with a K value of 0.9 and observe the simulations:

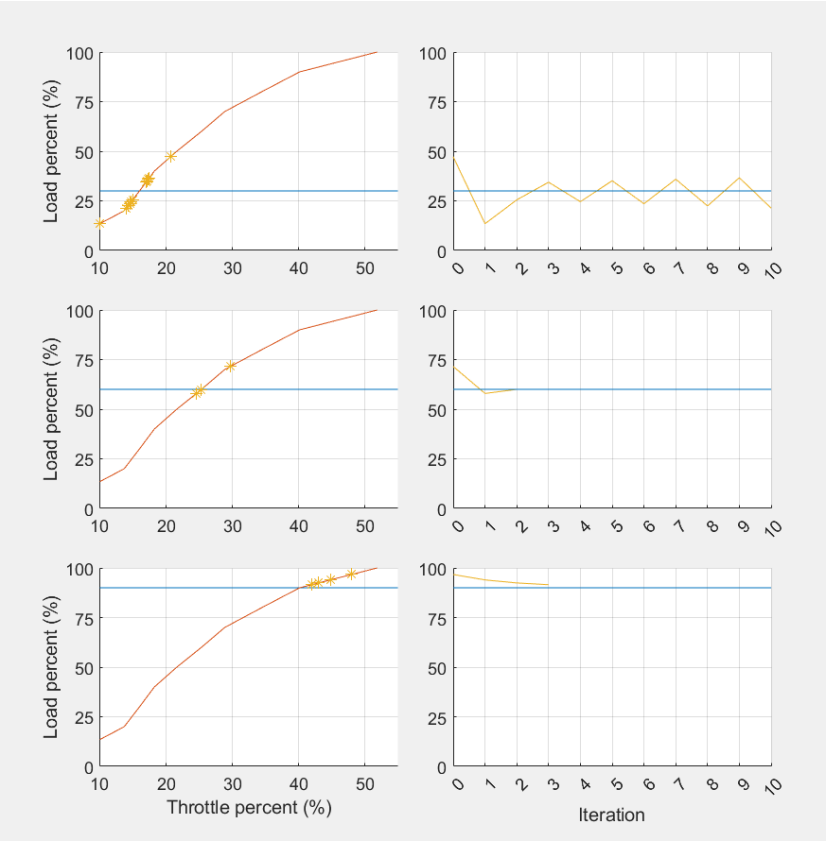

Figure 40: Algorithm's simulation with K=0.9 at 2500 rpm

We notice three things:

- For a target load of 90% (bottom graphs), the number of iterations to reach convergence has increased from two to three.
- For a target load of 60% (middle row graphs), convergence is achieved after only 2 iterations, so we can assume that the value of K=0.9 is close to optimal under these conditions.
- For a target load of 30% (top graphs), the algorithm still fails to bring the engine to the target load percentage. However, the situation has improved compared to the previous case because the oscillations around the target value are of smaller amplitude.

Lowering the value of K further to 0.5 gives us these last results:

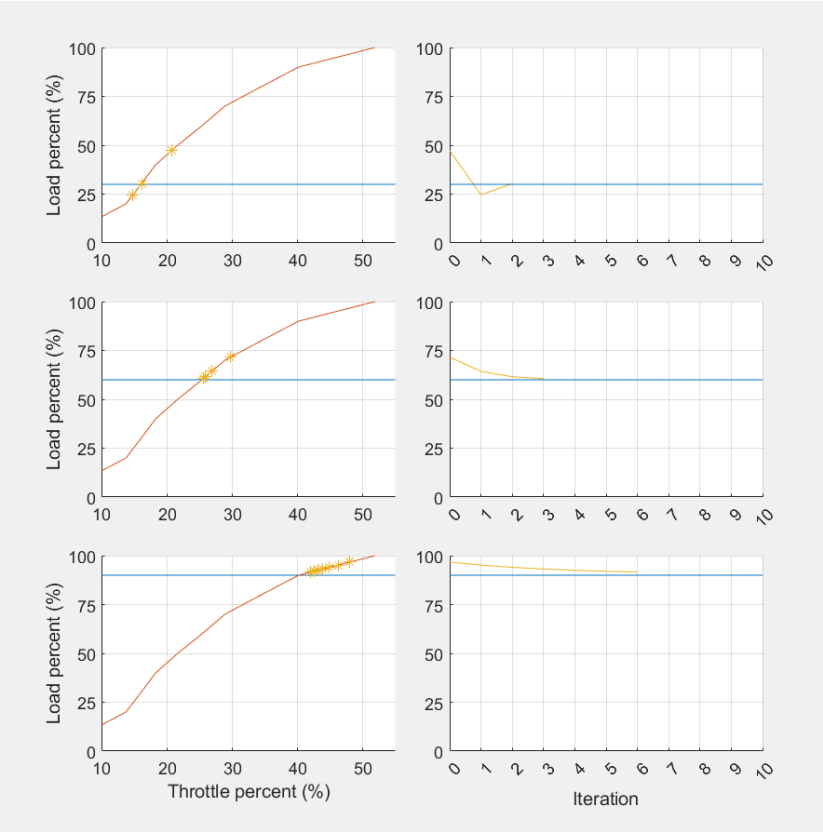

Figure 41: Algorithm's simulation with K=0.5 at 2500 rpm

It is crucial to note that with this value of K, a target load of 30% achieves convergence in just 2 iterations, without oscillations as in the previous cases. However, it is also important to note that the number of iterations to reach convergence in the case of a 90% target load has doubled, reaching 6.

From this analysis, we conclude that the best option would be to have a variable K for each operating point, allowing reaching the target value in a few iterations. However, as mentioned earlier, this is impossible because load-throttle percentage curves are heavily influenced by engine speed and general conditions, making it impossible to calculate or derive the optimal K value for each and every condition. Therefore, a constant value of K is chosen, but it must be selected carefully. A value that is too low allows convergence for any condition and avoids conditions of unstable oscillations, but it sacrifices convergence time for some conditions. On the other hand, a value that is too high can lead to faster convergence in many conditions but may also generate unacceptable divergence situations.

In conclusion, after numerous tests with different K values, it was found that a value of 0.2 provided the best compromise between convergence speed and safety margin regarding the divergence

conditions that need to be avoided at all costs. The effectiveness of this algorithm for controlling the engine load percentage will be thoroughly discussed in Chapter 4.3.

The algorithm is then implemented in LabVIEW:

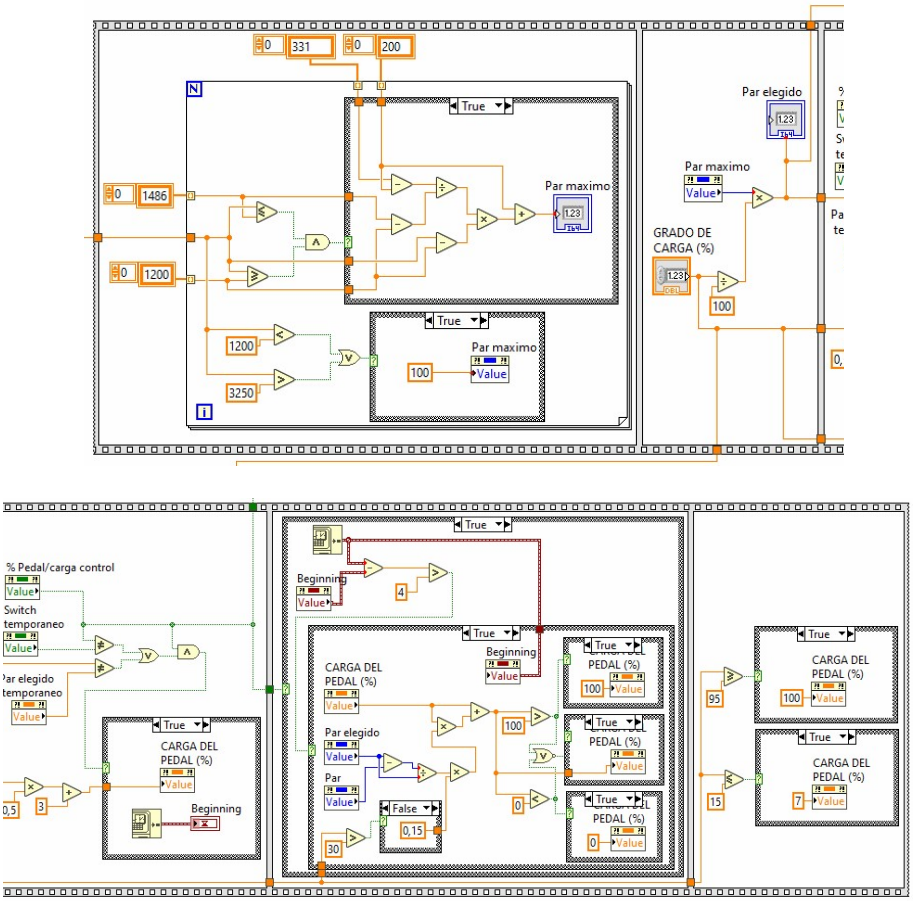

Figure 42: Control algorithm on LabVIEW

The algorithm needs to do the steps explained above, which are:

- Calculate the maximum torque at the set engine speed, calculating it through interpolation of the performance curve of the engine.
- Calculate the first trial throttle percent based on the equation given before.
- Calculate the throttle percent at every iteration.

The front panel is modified too by adding a switch to give the user the option to control the engine with either the throttle percentage or the load percentage:

Figure 43: Load percentage switch and controller

When the switch is clicked, the control of the engine shifts to the load percentage, so the throttle percentage controller is deactivated. The user will enter the desired load percentage, and the algorithm will bring the engine to that value. At the same time, two new indicators will appear on the screen:

Figure 44: Maximum and target torque's indicators

that indicate the maximum torque at that speed (M*aximo*) and the target torque (*Target*), which is the maximum torque multiplied by the load percentage input by the user. This way, the user can see what the target torque is, and during the transient observe how far the engine is from the desired condition.

3.8. Cycle programming function

For a successful automation of a test bench, a cycle programming function has been developed and implemented. This function allows for the automated and controlled management of the engine's behaviour during tests, enabling the precise repetition of specific work cycles. Moreover, automating operating cycles significantly reduces the need for human intervention during tests, thereby decreasing the risk of errors and enhancing the overall efficiency of the test bench. Another crucial aspect is the flexibility of the programmable approach. With the ability to define custom operating cycles, it becomes possible to simulate more complex real-world scenarios, approaching the actual usage conditions of the engine, or to conduct standardized tests such as the World Harmonized Stationary Cycle (WHSC).

The WHSC is a standardized test cycle used to evaluate pollutant emissions from internal combustion engines, particularly those used in heavy-duty vehicles. This test cycle is part of vehicle emission certification and control standards and refers to specific driving conditions that simulate the vehicle's operation in urban and extra-urban traffic. The main goal of the WHSC cycle is to ensure that vehicles on the market comply with emission limits imposed by environmental laws. More details about the WHSC cycle will be provided in the next chapter. This thesis does not aim to conduct a test of an actual WHSC cycle because such a test requires extensive preparation and many instruments, especially for exhaust gas analysis, which our test bench does not have. However, the goal is to develop a programming function that would be sufficient to perform a true WHSC cycle test and would, therefore, be suitable for conducting any other cycle.

The tab for programming the cycle function looks like this:

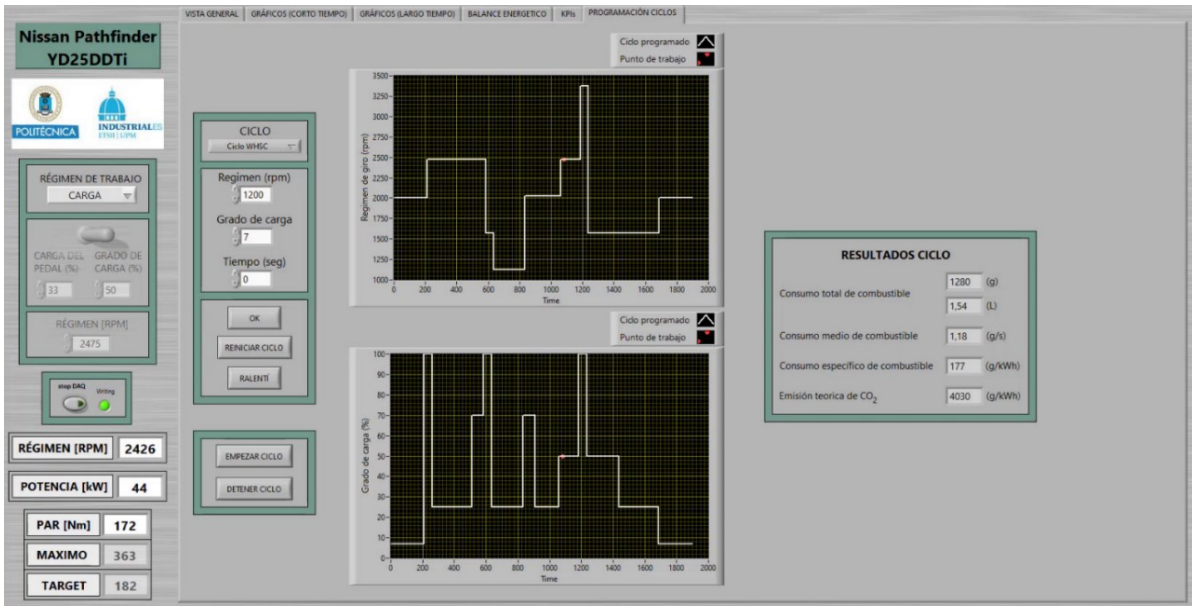

Figure 45: Cycle programming function tab

This is divided into three parts: the leftmost part is for the creation, initialization, and termination of the cycle; in the central part, there are graphs depicting the speeds and load percentages of the programmed cycle; the right part shows the real-time results of the cycle. Let's discuss each part more in detail.

In the first dropdown menu at the top left, we can select *ciclo personalizado* (custom cycle) or *ciclo WHSC* (WHSC cycle).

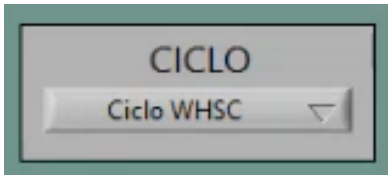

Figure 46: Cycle's dropdown menu

Selecting the second option the WHSC cycle is automatically set, and the speed and load percentage values throughout the duration of the cycle will be displayed on the two graphs in the centre. Choosing the first option (custom cycle) allows the user to create a custom working cycle. The graphs clear, and the user can enter the speed (*regimen*), load (*grado de carga*) and the time for which they want to keep the engine under these conditions.

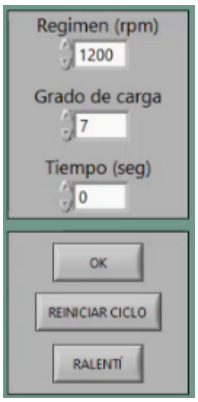

Figure 47: Indicators and controllers for cycles

Pressing the OK button adds the conditions just entered to the end of the cycle. By gradually entering all the conditions, a cycle of any duration and conditions can be created, while the graphs will continue to update automatically. There is also a Restart Cycle button to clear the programmed cycle up to that point and an Idle button to set the engine to idle conditions.

Once the programming is complete, the user can click the Start Cycle button to begin. This deactivates the boxes for manual programming of speed and load, and the computer will receive the values to be output to the engine from the cycle programming section. The operating point of the cycle will be indicated in red on the graphs while time progresses.

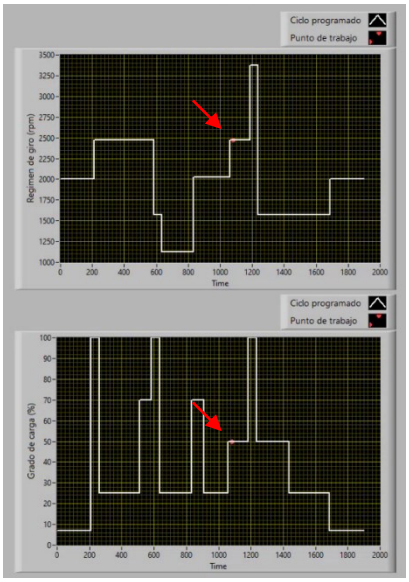

Figure 48: Indicator for progress of the cycle

In the results section of the cycle (*Resultados del ciclo*), some quantities that can be particularly important in the context of engine tests are calculated, which include:

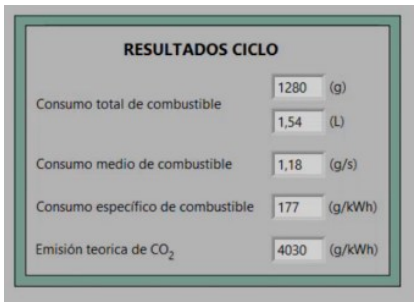

Figure 49: Cycle's results

- Mass and volumetric total fuel consumption.
- Average mass fuel consumption.
- Specific fuel consumption.
- Theoretical specific CO_2 emission.

A particular note should be made regarding the theoretical specific CO_2 emission. The CO_2 emission should be measured using a catalytic converter at the engine exhaust and a specific analysis system. However, in the absence of this instrumentation on the test bench, we show only the theoretical CO_2 emission, which has been calculated based on the combustion reaction of diesel:

$$C_{12}H_{23} + \frac{71/4}{F_R}(O_2 + 3{,}76N_2) \rightarrow 12CO_2 + \frac{23}{2}H_2O + \frac{3{,}76 \cdot \frac{71}{4}}{F_R}N_2 + \frac{71}{4} \cdot \left(\frac{1}{F_R} - 1\right)O_2$$

For each mole of diesel participating in combustion, 12 moles of carbon dioxide are produced:

$$m_{CO_2} = 12 \cdot m_{diesel} \cdot \frac{MM_{CO_2}}{MM_{diesel}}$$

Where *MM* are the molecular masses of carbon dioxide and diesel, obtained as:

$$MM_{CO_2} = 12{,}01 + 16{,}00 \cdot 2 = 44{,}01 \ \frac{g}{mol}$$

$$MM_{diesel} = 12{,}01 \cdot 12 + 1{,}01 \cdot 23 = 167{,}35 \ \frac{g}{mol}$$

From which we obtain:

$$m_{CO_2} = 3{,}16 \cdot m_{diesel}$$

By dividing this quantity by the energy produced during the cycle, we can obtain the theoretical specific CO_2 emission.

In the following image we show the block diagram part which was programmed for the cycle programming function and to perform all the above-mentioned tasks:

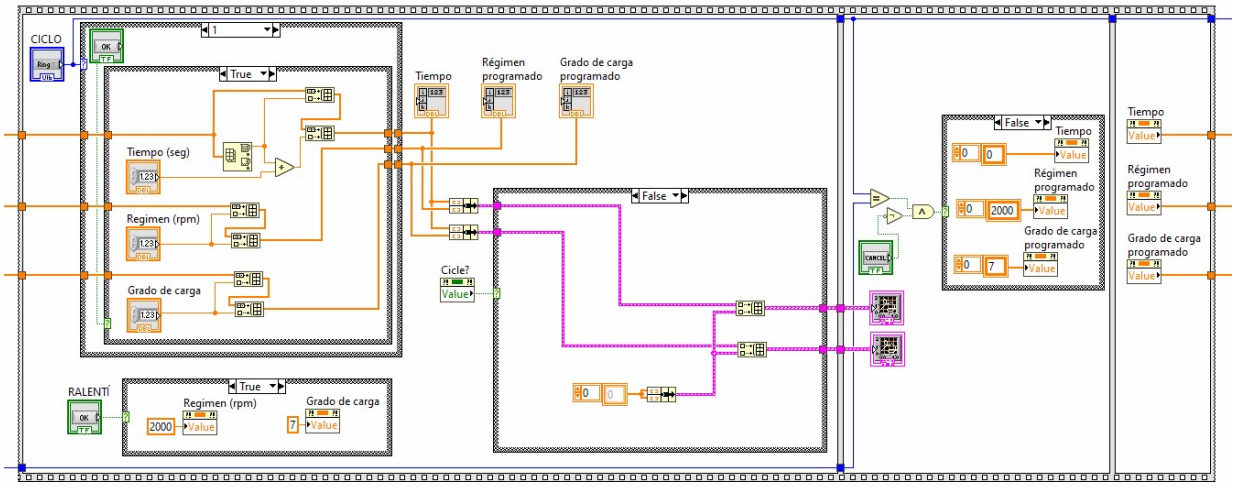

Figure 50: Cycle programming function on the block diagram

4. WHSC Cycle

4.1. Cycle description

The WHSC (World Harmonized Steady State Cycle), along with the WHTC (World Harmonized Transient Cycle), is a cycle born from the proposal of a global technical regulation (GTR) which "aims at providing a world-wide harmonized method for the determination of the levels of pollutant emissions from engines used in heavy vehicles in a manner which is representative of real-world vehicle operation"[4]. The goal of this GTR is to create a globally harmonized procedure for the approval of engines used in heavy vehicles, in order to obtain results that are representative of the actual operation of vehicles under real conditions. This proposal is based on global research on the usage patterns of heavy commercial vehicles and defines test cycles, including the WHSC test, that replicate driving conditions typical in different parts of the world, including the European Union, the United States, Japan, and Australia.

The WHSC test procedure was developed to:

- Be representative of vehicle operations on roads worldwide.
- Provide the highest possible level of on-road emissions control efficiency.
- Match state-of-the-art testing, sampling, and measurement technology.
- Be applicable in practice to existing and foreseeable future exhaust emission reduction technologies.
- Provide a reliable ranking of exhaust emission levels among different types of engines.

Furthermore, the GTR states that "the emissions to be measured from the exhaust of the engine include the gaseous components (carbon monoxide, total hydrocarbons or non-methane hydrocarbons, methane and oxides of nitrogen), and the particulates. Additionally, carbon dioxide is often used as a tracer gas for determining the dilution ratio of partial and full flow dilution systems"[4]. It is important to highlight that the regulations provides precise indications on the preparation, procedure, data acquisition, validation, and emission calculation of the WHSC cycle, but it does not provide any indications on emission limits. This is because the goal is to create a test that is as representative and

standardized as possible, but the task of setting emission limits for engines is the responsibility of the relevant authorities. These limits cannot be the same in different parts of the world, where existing rules may be more or less stringent, and they have not and will not be constant over time, as the issue of climate change and the emission of greenhouse gases from engines has changed them and will undoubtedly change them further in the future.

As we discussed earlier, our test bench does not have any instrument for measuring any of these elements of exhaust gases. As a matter of fact, the WHSC test is not performed to measure the emission level of the engine installed on the test bench, also because it would have no relevance to the subject of the thesis, but to verify the level of innovation and automation provided to the test bench during the work done. We have indeed discussed all the improvements that have been made to the test bench in Chapter 3, and among these are the engine control based on the load percentage and the cycle programming function. Without the latter function, setting the operating conditions during the test would have to be done manually, and it would be inconvenient, as the test lasts about 30 minutes, and secondly, the human factor would be too high, and the risk of a mistake in setting the parameters or a wrong timing would be too high to conduct such a test. The other feature, namely the engine control, is essential to carry out such a test because we will see later that the operating points of the engine in the WHSC test will have to be chosen based on the engine's load percentage.

Therefore, the execution of the WHSC test is intended to be a way to verify the effectiveness of the new improvements, both qualitatively and quantitatively. An initial qualitative analysis will examine engine control, comparing the reference cycle to the real working cycle to verify that the engine has reached the correct operating conditions. A second quantitative analysis will follow the guidelines of the GTR, which provides two verification tests of the cycle to certify that the test conducted conforms to the required standards.

4.2. Reference cycle

The WHSC cycle test consists of various normalized engine speed and load operating modes. The normalized reference cycle is the following:

Mode	Normalized speed (%)	Normalized load (%)	Mode length (s)
0	Motoring		
1	0	0	210
2	55	100	50
3	55	25	250
4	55	70	75
5	35	100	50
6	25	25	200
7	45	70	75
8	45	25	150
9	55	50	125
10	75	100	50
11	35	50	200
12	35	25	250
13	0	0	210

Table 7: Normalised WHSC test

The cycle includes the motoring of the engine, an idle phase at the beginning and end of 210 seconds, and 11 intermediate phases at different speeds and loads for different durations, ranging from 50 to 250 seconds. The total cycle lasts for 1895 seconds, approximately 30 minutes. The cycle is represented by normalized speeds and loads, which will need to be de-normalized to create the characteristic cycle for each specific engine.

Specifically, the normalized load is the ratio of the required torque to the maximum torque at the reference speed. In our case, we will leave the load normalized, without de-normalizing to the required torque, as the engine control we want to test operates through the percentage of load. A different consideration must be made for speed. The formula to obtain the actual speed from the normalized speed is the following:

$$Actual\ speed = n_{norm} \cdot \left(0{,}45 \cdot n_{lo} + 0{,}45 \cdot n_{pref} + 0{,}1 \cdot n_{hi} - n_{idle}\right) \cdot 2{,}0327 + n_{idle}$$

where:

- n_{norm} is the normalized speed from Table 7.

- n_{lo} is the lowest speed where the power is 55 per cent of the maximum power.
- n_{hi} is the highest speed where the power is 70 per cent of maximum power.
- n_{idle} is the idle speed.
- n_{pref} is calculated from the graph of torque vs speed from $n_{idle}$to n_{95}, that is the maximum speed at which power is 95% of the maximum power; n_{pref} is the engine speed where the torque integral is 51% of the whole integral between $n_{idle}$and n_{95}.

The following is the load percentage vs speed graph of our engine with all the necessary speeds for the de-normalization of the cycle, along with a summary table:

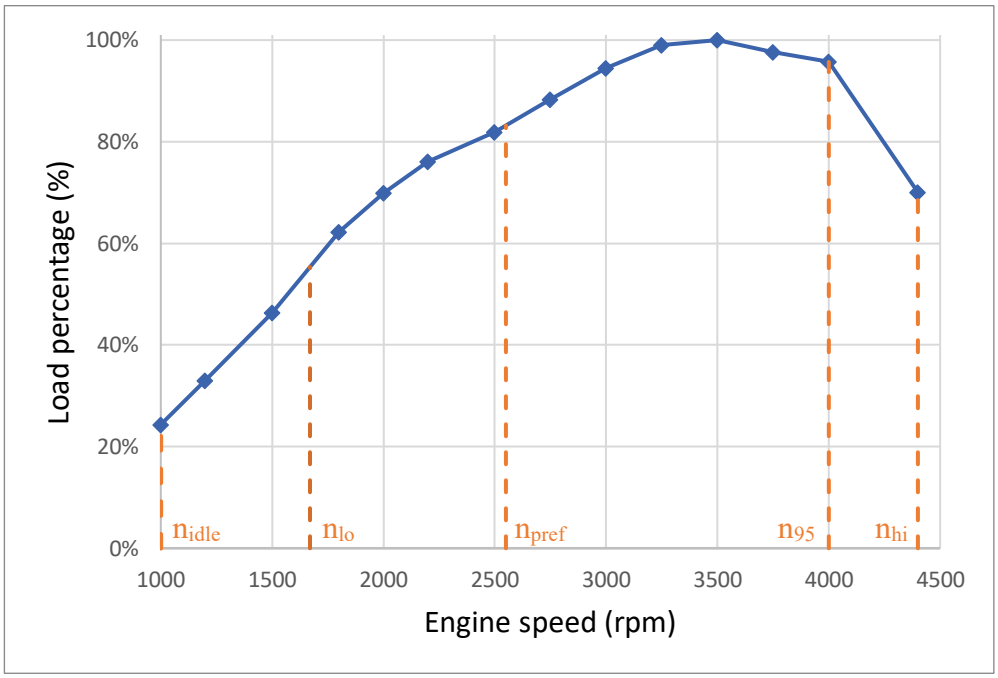

Figure 51: Load percentage vs speed of the engine

Name	Speed (rpm)
n_{idle}	1000
n_{lo}	1670
n_{pref}	2550
n_{95}	4000
n_{hi}	4400

Table 8: Characteristic speeds of the engine

At this point, the formula to calculate the actual speed is:

$$Actual\ speed = n_{norm} \cdot 2721{,}8 + 1000$$

So, let's now examine the specific cycle for our engine, along with the load and speed graphs throughout the cycle:

Mode	Speed (rpm)	Load percentage (%)	Mode length (s)
0	Motoring		
1	1000	0	210
2	1000	100	50
3	2497	25	250
4	2497	70	75
5	2497	100	50
6	1953	25	200
7	1680	70	75
8	2225	25	150
9	2225	50	125
10	2497	100	50
11	3041	50	200
12	1953	25	250
13	1000	0	210

Table 9: De-normalised WHSC test

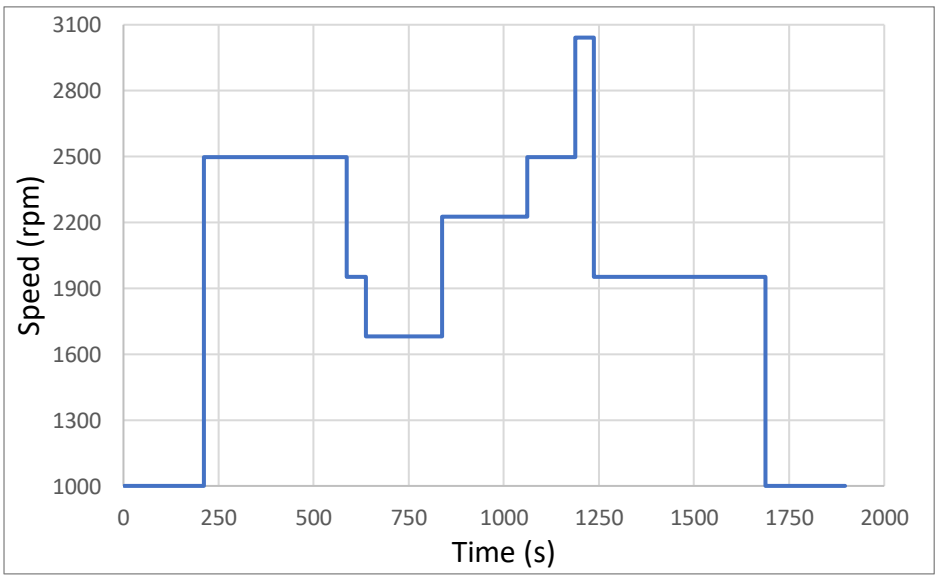

Figure 52: Speed in the WHSC test

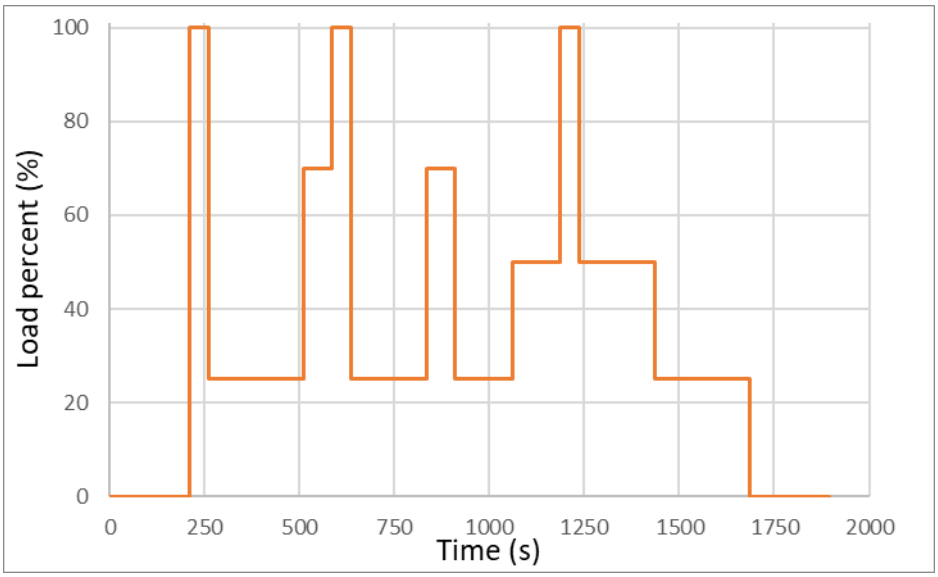

Figure 53: Load in the WHSC

Unfortunately, the calculation of the formula that was taken in the first place for the speed de-normalization was incorrect. Consequently, the first WHSC test conducted on the test bench was carried out with incorrect speed values. Only later did we realize that the test performed was similar but not exactly the same regarding the engine speeds. The possibility of redoing the entire test was considered. However, as mentioned earlier, we are not concerned about the test result being valid in terms of regulations. Therefore, it was assessed that even if the cycle was incorrect, it would still be suitable for testing the improvements in automation and control. The number of modes, duration, and load percentage remain the same, while the speeds differ slightly. Below is the speed diagram used in the test, along with the corrected cycle shown in Figure 52:

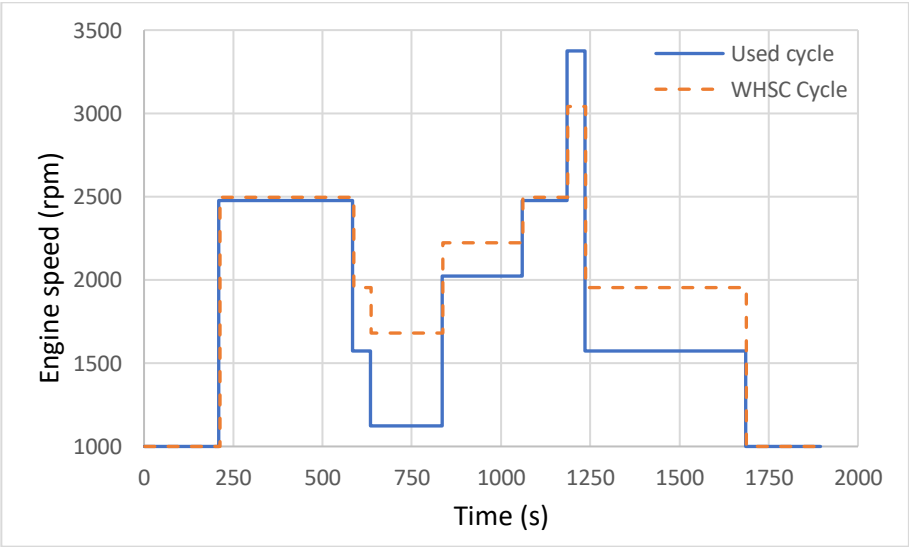

Figure 54: WHSC speed vs used speed

While the following table shows the precise values of the speeds used in the cycle:

Mode	Speed (rpm)	Load percentage (%)	Mode length (s)
0	Motoring		
1	2475	0	210
2	2475	100	50
3	2475	25	250
4	1575	70	75
5	1125	100	50
6	2025	25	200
7	2025	70	75
8	2475	25	150
9	3375	50	125
10	1575	100	50
11	1575	50	200
12	2475	25	250
13	2475	0	210

Table 10: Used reference cycle

4.3. Test results

In this paragraph, we will conduct a qualitative analysis of the results of the WHSC test, specifically to assess the effectiveness of the implemented engine load percentage control algorithm. Let's examine the graph of the reference load percentage and the actual load obtained during the test:

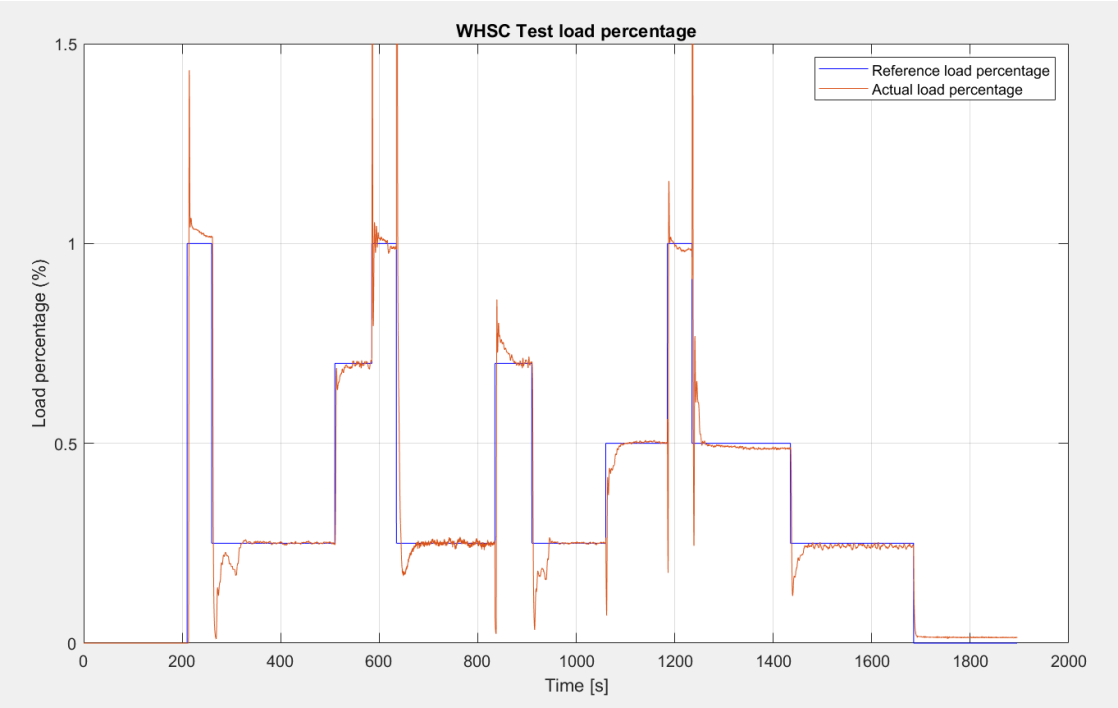

Figure 55: Reference vs actual cycle

We shall remember that the speed throughout the WHSC cycle is not constant, and therefore, points on this graph with the same load percentage in two different zones do not represent the same operating points. Between t = 260 s and t = 510 s, the engine operates at a speed of 2475 rpm, while between t = 1435 s and t = 1685 s, the engine operates at 1575 rpm. Therefore, the conditions are quite different, even though they appear the same on the graph in Figure 55 because they have the same load percentage.

Firstly, let's observe abrupt and sudden load variations where torque sharply increases or decreases. However, these are not due to the inefficiency of the load control algorithm but are intrinsic properties of the engine and the test bench, making them unavoidable. Excluding these instances, we can already qualitatively state that the actual cycle reasonably follows the reference cycle.

The transient to bring the engine to the reference cycle conditions is not always the same, as seen in Chapter 3.7 for being influenced by the slope of the load-throttle percentage curve. Let's observe different cases of the WHSC test:

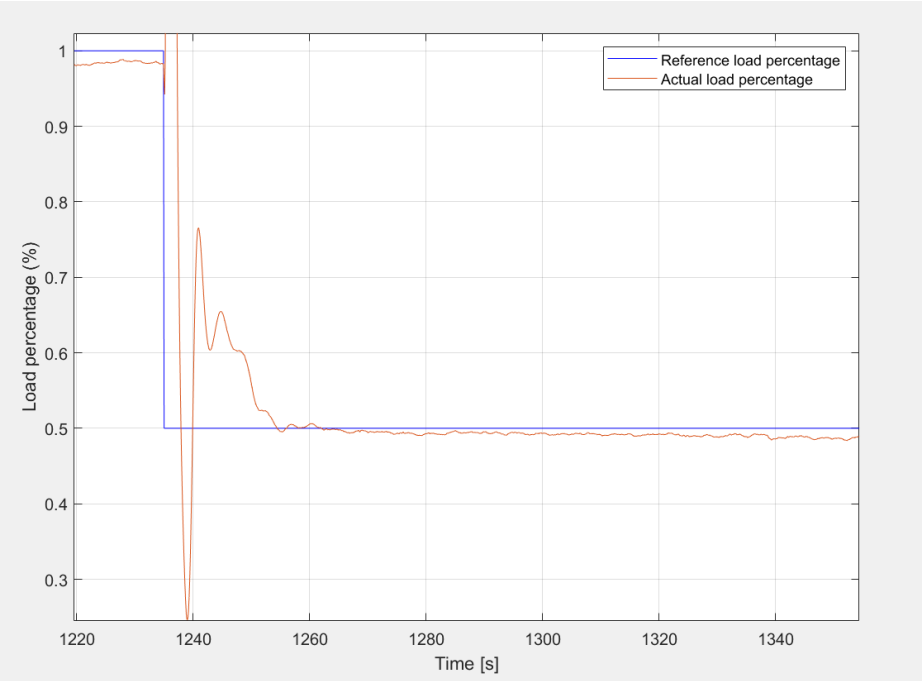

Figure 56: Transient example 1

In this case, we observe inevitable strong fluctuations in the first 5 seconds, partly due to the fact that, at the same time, the engine speed changes from 3375 rpm to 1575 rpm, followed by a subsequent rapid and progressive approach to the desired load of 50%. After about 20 seconds, the system has stabilized at the desired value, and even at steady-state, the engine remains relatively stable.

Let's consider a case where the speed remains constant (1125 rpm) between conditions, and only load percentage changes:

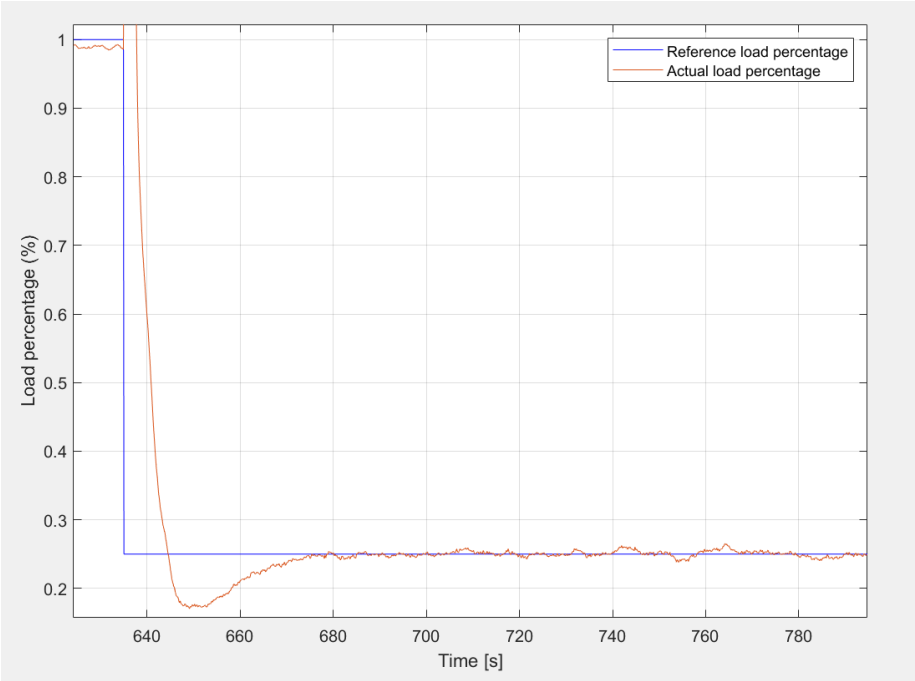

Figure 57: Transient example 2

In this case, the torque is more stable and does not undergo significant fluctuations, as in the previous case. However, at the same time, the time to reach the target value is also longer.

The third case we focus on is the following:

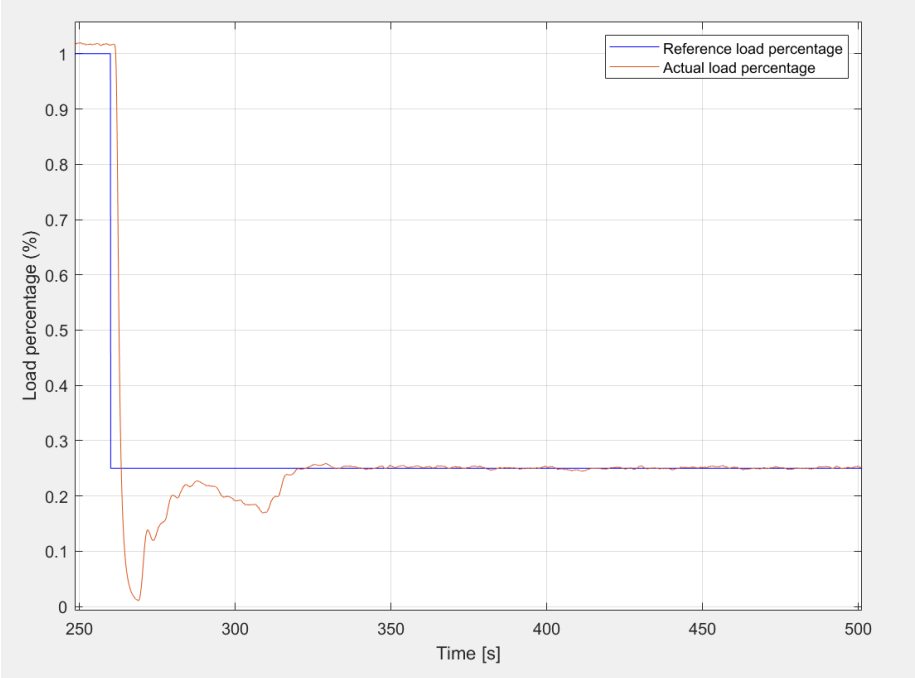

Figure 58: Transient example 3

Note how, during the transition period, the engine gradually moves toward the desired value of 30%, but at a certain moment (t = 290 s) the torque and load decrease. In this instance, despite an increase in the pedal percentage, the load actually decreased. Subsequently, the load began to rise again. This is highlighted to emphasize the strong variability in the relationship between load percentage and pedal, underscoring the importance of developing a simple algorithm that avoids generating unstable conditions even in abnormal situations, as the one described above.

The last transient we examine is the following:

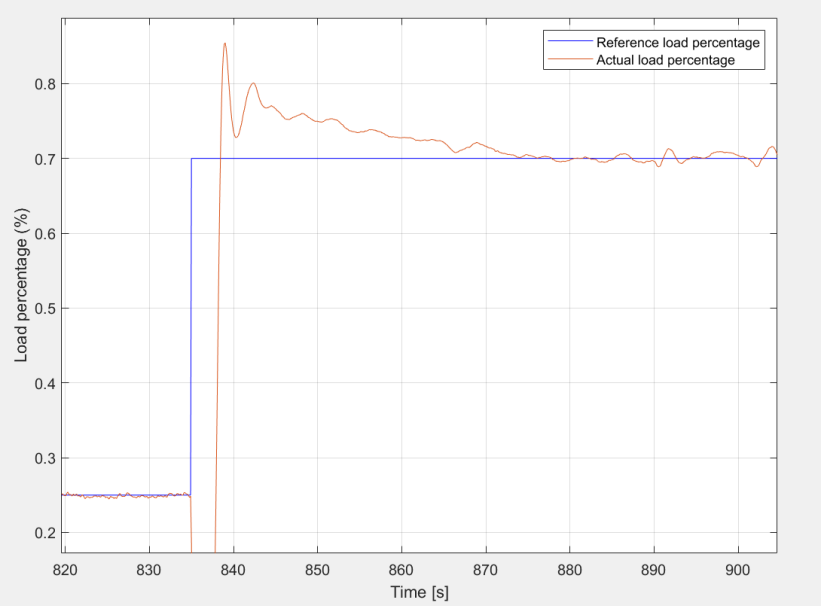

Figure 59: Transient example 4

For these conditions, we can observe a lower efficiency of the algorithm as reaching the target condition is rather slow, taking about 40 seconds to achieve the target value. Additionally, once reached, the system struggles to remain stable at that point, showing non-neglectable fluctuations.

4.4. Validations

From a more quantitative point of view, let's now perform two validations of the test result according to the guidelines of the reference regulation. The regulation outlines two different types of validation. The first requires comparing the actual cycle work to the reference cycle work to ensure that the cycle is sufficiently similar to the reference one. The second one demands verifying that there is no excessive delay between the change in conditions of the reference cycle and the actual achievement of the new condition.

Both validations, in a sense, can serve as quantitative checks for the newly implemented engine control system. If the system does not bring us to the correct condition, there would be a significant difference between the actual and the reference cycles, and the first validation would not be satisfied. If the responses to new conditions were too slow, there would be a substantial delay between input and reaching the goal, and the second validation would not be satisfied.

That said, the first validation requires the actual cycle work W_{act} to be between 85 and 105 percent of the reference cycle work W_{ref}. We calculate the actual work by summing, over the entire 1895 seconds cycle, the product of the measured instantaneous power P and the data acquisition time resolution Δt, which is 0.1 s:

$$W_{act} = \frac{1}{3600} \sum_{t=0}^{1895} P \cdot \Delta t = 12{,}62\ kWh$$

The work of the reference cycle is instead calculated as the sum of the product of speed n_i, torque T_i, and duration t_i for each of the 13 modes of the cycle:

$$W_{ref} = \frac{2 \cdot \pi}{3600 \cdot 1000 \cdot 60} \sum_{i=1}^{13} n_i \cdot T_i \cdot t_i = \frac{\pi}{1{,}08 \cdot 10^8} \sum_{i=1}^{13} n_i \cdot T_i \cdot t_i = 13{,}11\ kW$$

Then, the ratio between the work of the reference cycle and the actual cycle is 96.3 percent, well within the validation threshold of the regulation. The fact that the actual work is not identical to the reference work is obviously given by the fact that the reference cycle sets an instantaneous transition between two conditions while the actual cycle inevitably requires a transition time from getting to one

condition to another. However, the fact that the two works are so close indicates that the transients are still sufficiently rapid.

The second validation is done "to minimize the biasing effect of the time lag between the actual and reference cycle values"[4]. Indeed, this is a validation to directly assess whether the delay introduced by our algorithm between the new operating point and the actual achievement of said operating point is sufficiently low. To do so, linear regressions of the feedback values on the reference values shall be performed for speed, torque and power. The process requires to plot the actual values versus reference values of speed, torque and power, and then through the method of least squares find the best fit equation $y = mx + b$, with y being the feedback values during the whole cycle and x the reference values. Together with the calculation of the slope coefficient m and the y-intercept b, it is required to calculate the standard error of estimate (SEE) of y on x and the coefficient of determination r^2 of the regression line. This process will be done using Matlab, through which we can plot the reference versus actual values and calculate the 4 above-mentioned parameters:

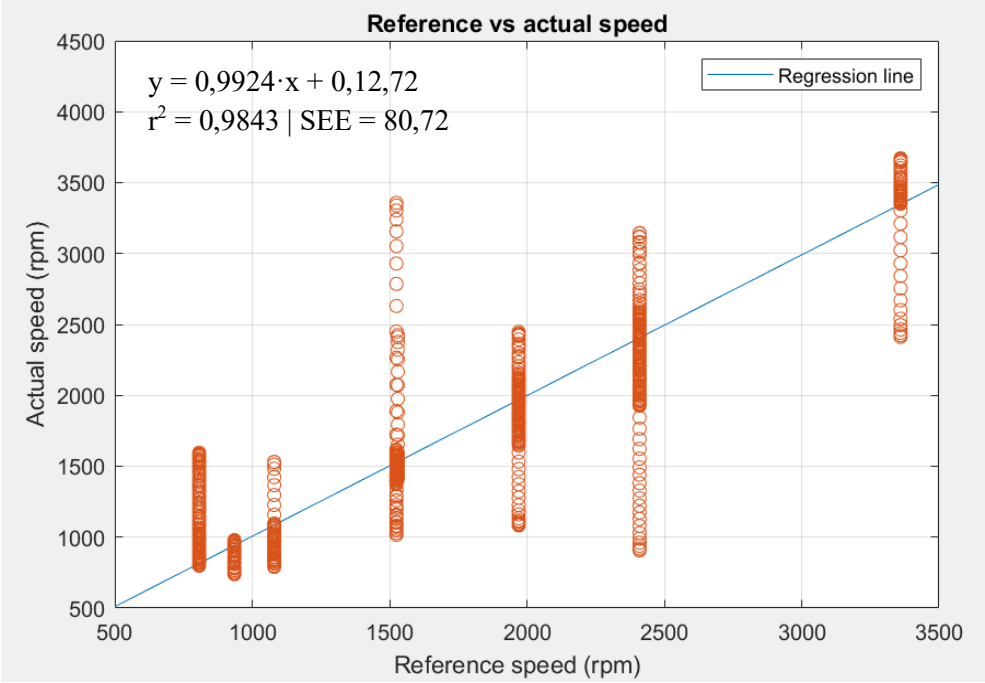

Figure 60: Reference vs actual speed regression line

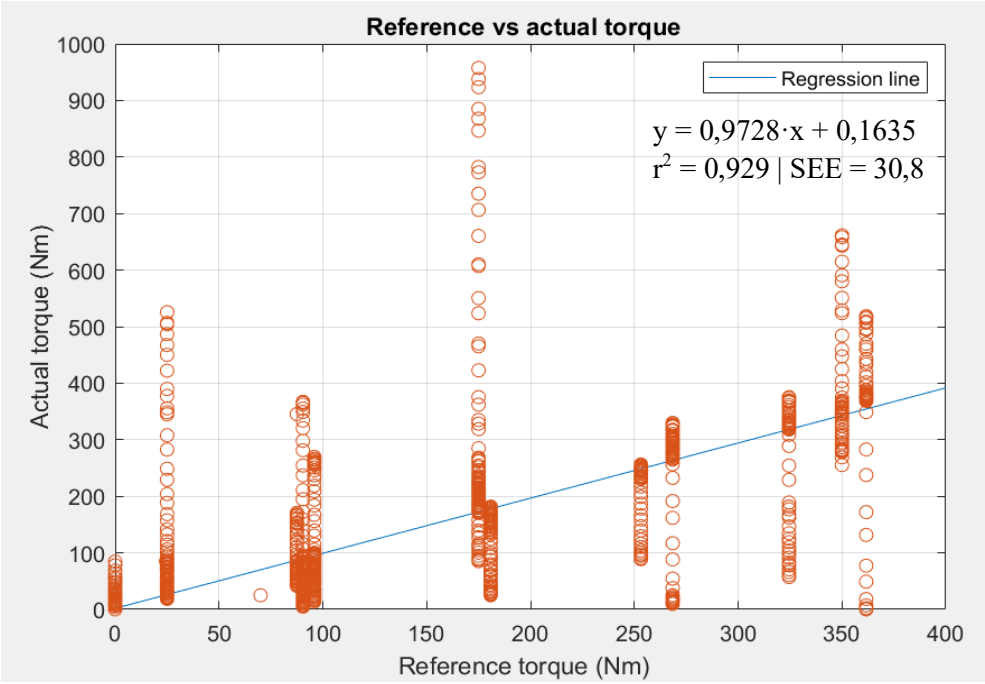

Figure 61: Reference vs actual torque regression line

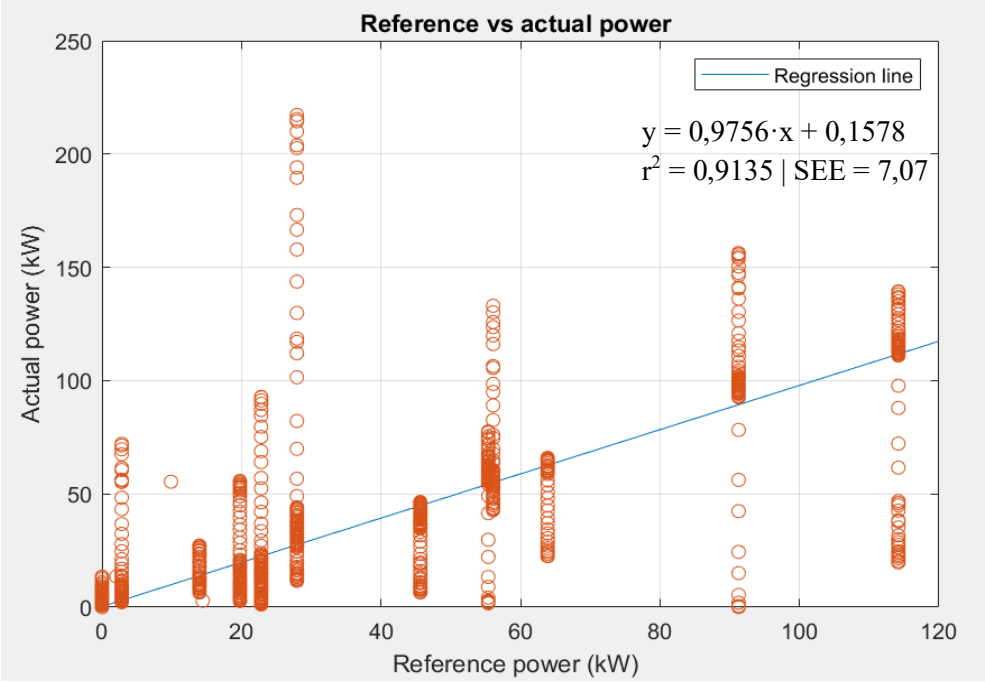

Figure 62: Reference vs actual power regression line

The GTR imposes limits to the values of the parameters just calculated. Being able to satisfy these limits and having values inside the tolerances means confirming quantitively that the delay between the reference cycle and the actual cycle is restrained and that the control algorithm is advanced enough to satisfy strict limitations imposed by a technical regulation. In green, we have all

the values well within the tolerances, while in yellow, the only value within the tolerances but somewhat at the limit is reported:

	Speed		Torque		Power	
	Tolerance	Actual	Tolerance	Actual	Tolerance	Actual
Slope	0.95 – 1.03	0,9924	0.83 – 1.03	0,9728	0.89 – 1.03	0,9756
y intercept	±50 rpm	12,72 rpm	±20 Nm	1,635 Nm	±4 kW	0,158 kW
r^2	> 0.970	0,9843	> 0.850	0,9288	> 0.910	0,9135
SEE	< 100 rpm	80,7 rpm	< 13% of max engine torque = 51 Nm	30,8 Nm	< 8% of max engine power = 8,8 kW	7,07 kW

Table 11: Tolerance and actual validations values

5. Conclusions and future works

5.1. Conclusions

In conclusion, this thesis work has successfully innovated and enhanced a test bench, yielding tangible and significant results. Numerous modifications were made, addressing various issues across different aspects of the test bench, starting from small hardware changes like installing an atmospheric pressure sensor and the reorganization of thermocouple cables. Other changes, on the other hand, were more substantial and radically transformed the test bench.

On one hand, the user interface has been completely revamped, managing to elevate the level of data visualization. Moving beyond merely displaying parameters measured by the engine sensors, the visualization of data has been enriched by providing a time-based representation of these parameters to understand their evolution. Additionally, important tools within the engine studies, such as performance indices and energy balance, are now processed and visualized. Despite representing a large amount of information, the interface has been designed to maintain simplicity, ensuring clarity for the user.

On the other hand, significant progress has been made in test bench automation. Whereas achieving an operating condition characterized by a specific torque value previously required manual control of the pedal percentage, the implementation of an algorithm for control based on torque and load percentage represents a significant step toward reducing human intervention on the test bench. This minimizes the possibility of errors and enhances efficiency. With the new control function, the possibility of creating a cycle programming feature has also emerged, further reducing human involvement during engine operation and allowing the user to focus on the information displayed on the interface rather than controlling the engine. The successful execution of a WHSC test directly validated the improvement in test bench automation, surpassing both qualitative and quantitative analyses of automation, making it suitable for conducting any standardized test.

5.2. Future works

As additional improvements to the test bench, this work proposes the following future enhancements:

- Conduct a more in-depth study of the pedal percentage-load relationship to implement a more complex algorithm that achieves convergence more quickly without encountering instability conditions.
- Add sensors in the combustion chamber to measure the evolution of pressure during the working cycle.
- Replace the fuel flow meter to enhance the measurement of this parameter.
- Install an exhaust gas analysis system.

List of figures

List of tables

Bibliography

[1] “The PID Controller & Theory Explained.” NI, 30 Mar. 2023, www.ni.com/en/shop/LabVIEW/pid-theory-explained.html.

[2] Ang, K.H. and Chong, G.C.Y. and Li, Y. (2005) PID control system analysis, design, and technology. IEEE Transactions on Control Systems Technology 13(4):pp. 559-576.

[3] “Principles of PID Controllers | Zurich Instruments.” Www.zhinst.com, 13 July 2023, www.zhinst.com/europe/en/resources/principles-of-pid-controllers.

[4] UNITED NATIONS. Global Technical Regulation No. 4: TEST PROCEDURE for COMPRESSION-IGNITION (C.I.) ENGINES and POSITIVE IGNITION (P.I.) ENGINES FUELLED with NATURAL GAS (NG) or LIQUEFIED PETROLEUM GAS (LPG) with REGARD to the EMISSION of POLLUTANTS . 25 Jan. 2007.

[5] Martínez González, Imanol. “Transformación Digital Del Sistema De Instrumentación Y Control De Una Celda De Ensayos De Motores Alternativos.” *Universidad Politécnica de Madrid*, 2021.

[6] “Sensor de Presión Absoluta, MPXH6115AC6U, SSOP 8 Pines 115kPa | RS.” *Es.rs-Online.com*, es.rs-online.com/web/p/sensores-de-presion-para-pcb/7176536.

[7] Jesús. Casanova Kindelán, “Tema 4: Balance energético de los motores alternativos. Pérdidas de calor y mecánicas”, *Motores Térmicos*, Madrid, 2019.

[8] Depto. Ingeniería Energética ETSII UPM. (2021, septiembre 28). Guion de prácticas, Ensayo de motores de combustión interna alternativos en banco de pruebas. Madrid.

[9] “Nissan 2.5 YD25DDTi HP Engine.” AutoManiac, www.automaniac.org/engine/nissan/1253/nissan-2488cc-diesel-2.5-yd25ddti-hp-16v-190hp.

Appendix

Appendix 1: Spanish-English translator

Spanish	English	Spanish	English
Aceite	Oil	Máximo	Maximum
Admisión	Admission	Medio	Average
Agua	Water	Motor	Engine
Atmosférica	Atmospheric	Par	Torque
Balance energético	Energy balance	Porcentaje de carga	Load percentage
Calor	Heat	Potencia	Power
Carga del pedal	Throttle percentage	Potencia efectiva	Effective power
Catalizador	Catalyser	Presión	Pressure
Caudal	Flow rate	Presión media efectiva	Mean effective pressure
Combustible	Fuel	Programación ciclos	Cycles programming
Compresor	Compressor	Ralentí	Idle
Consumo	Consumption	Refrigerante	Refrigerant
Consumo especifico	Specific consumption	Régimen	Speed
Corto tiempo	Short-term	Rendimiento efectivo	Effective efficiency
Dinamo Freno	Dynamometer	Rendimiento volumétrico	Volumetric efficiency
Eje secundario	Secondary axe	Resto	Rest
Emisión teórica	Theoretical emission	Resultados ciclo	Cycle’s results
Entrada	Inlet	Salida	Outlet
Gases de escape	Exhaust gases	Temperatura	Temperature
Gráficos	Graph	Vista General	General view
Largo tiempo	Long-term		