Gezamenlijke opleiding UHasselt en KU Leuven

2018•2019
Faculteit Industriële ingenieurswetenschappen
master in de industriële wetenschappen: energie

# Masterthesis

## Design of an exhaust bypass system for high temperature control in an automotive thermoelectric generator

PROMOTOR :
ing. Eric CLAESEN

PROMOTOR :
Prof. dr. ir. Lino MONTORO

Tom Croes

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: energie, afstudeerrichting automatisering

2018•2019
Faculteit Industriële ingenieurswetenschappen
**master in de industriële wetenschappen: energie**

# Masterthesis

Design of an exhaust bypass system for high temperature control in an automotive thermoelectric generator


**PROMOTOR :**
ing. Eric CLAESEN

**PROMOTOR :**
Prof. dr. ir. Lino MONTORO

**Tom Croes**

**Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: energie, afstudeerrichting automatisering**

# Preface

Dear reader,

I was fortunate enough to get the opportunity from the Uhasselt and the KU Leuven to accomplish my master's thesis, as an energy-automation student of the collaborative study program in Engineering Technology, abroad. As part of my master's thesis, the University of Gerona proposed to collaborate with the research group GREFEMA on the development of an automotive thermoelectric generator. A project that is very interesting since it covers many different domains (such as control engineering, microprocessors, thermodynamics, etc.) that were explored during my education for Industrial Engineer.

I would like to thank a few people in particular who have assisted me in this instructive adventure. First of all, I would like to thank my promoters Prof. dr. Lino Montoro and Prof. ing. Eric Claesen for offering me this subject. Also for supporting and advising me when it was needed.
I am also grateful to the internationalization coordinators Mrs. Greet Raymaekers and Mrs. Anna Critg for their administrative support and tips to complete this master's thesis abroad.
Finally, I would like to thank my family and my girlfriend for their mental support, but also in the first place to give me the opportunity to experience this Erasmus project.

Thank you,

Girona
1 June 2019

Croes Tom

# Table of contents

**Preface** ..... **1**

**List of tables** ..... **5**

**List of figures** ..... **7**

**Abstract** ..... **9**

**Abstract (Nederlands)** ..... **11**

**1 Introduction** ..... **13**

*1.1 Problem statement* ..... *13*

*1.2 Objectives* ..... *14*

*1.3 Material and method* ..... *14*

**2 Literature study** ..... **15**

*2.1 Automotive thermoelectric generator* ..... *15*

2.1.1 Principle ..... 15

2.1.2 Structure ..... 16

*2.2 Control theory* ..... *19*

2.2.1 Performance indicators ..... 19

2.2.2 Control algorithm ..... 20

**3 Material and method** ..... **27**

*3.1 Concept* ..... *27*

*3.2 Hardware* ..... *28*

3.2.1 Microcontroller ..... 28

3.2.2 Temperature measurement ..... 29

3.2.3 Motor ..... 30

*3.3 Software* ..... *31*

*3.4 Experimental setup* ..... *31*

**4 Control program ..... 33**

*4.1 Interface between LabVIEW and Arduino ..... 33*

*4.2 Temperature reading of the thermocouple ..... 33*

4.2.1 MAX31855 converter module ..... 33

4.2.2 DAQ device ..... 34

*4.3 Control program in LabVIEW ..... 35*

4.3.1 PID controller ..... 35

4.3.2 Fuzzy logic controller ..... 36

*4.4 Graphical monitor panel ..... 38*

**5 Results ..... 41**

*5.1 PID based temperature control ..... 41*

*5.2 Fuzzy Logic based temperature control ..... 45*

*5.3 Discussion ..... 47*

**6 Conclusion ..... 49**

**Bibliography ..... 51**

**Appendices ..... 53**

# List of tables

Table 1: Impact of increasing control parameters on performance indicators [18] ................. 23
Table 2: Formula set offered by Ziegler-Nichols method for control parameter determination in a closed loop [12] ........................................................................................................ 24
Table 3: Performance of PID expressed with time response parameters ................................ 41
Table 4: Results of Ziegler-Nichols method for a PID control system .................................... 43
Tabel 5: Performance of fuzzy control expressed with time response parameters .................. 45

# List of figures

Figure 1: Architecture of a TE module [8] .................... 15
Figure 2: Schematic diagram of ATEG with a) hexagonal-prism-shaped b) plate-shaped heat exchanger [10] .................... 16
Figure 3: Overview of the ZT of recent TE materials as a function of temperature [7] .................... 17
Figure 4: A typical response of a PID closed-loop system .................... 20
Figure 5: The operation of a bang-bang controller and the corresponding behavior of the control system .................... 21
Figure 6: An adjustable deadband/hysteresis with on/off control .................... 21
Figure 7: Schematic of a (parallel) PID controller [17] .................... 23
Figure 8: Different types of membership functions shape [22] .................... 25
Figure 9: Rule matrix for rules consisting of one input and one output linguistic variable [19] .................... 25
Figure 10: Block diagram of a Fuzzy Controller [22] .................... 26
Figure 11: Fuzzification [19] .................... 26
Figure 12: Design principle for the temperature control of the ATEG .................... 27
Figure 13: Block diagram of the proposed control system .................... 28
Figure 14: Photographic view of the experimental setup .................... 32
Figure 15: Initializing connection between Arduino and LabVIEW .................... 33
Figure 16: Part of the PID control program based on MAX31855 .................... 34
Figure 17: Part of the PID control based on the NI 9211 .................... 34
Figure 18: Block Diagram of the PID based control program .................... 35
Figure 19: (Upper) Automatic mode; (Lower) Manual mode in a PID control program .................... 36
Figure 20: Block Diagram of the Fuzzy based control program .................... 36
Figure 21: Membership function for the temperature error .................... 37
Figure 22: Membership function for the fuzzy output .................... 37
Figure 23: Rules of a fuzzy temperature control system .................... 38
Figure 24: Front panel of the PID control system .................... 39
Figure 25: Small automatic-mode window of the Front Panel for a Fuzzy control system .................... 39
Figure 26: Output response of PID control system with set temperature 150°C .................... 41
Figure 27: Output response of a PID control system subjected to disturbances .................... 42
Figure 28: Output response of a PID control system with changing setpoint .................... 42
Figure 29: Output response of a PID control system with set temperature 200°C .................... 43
Figure 30: Process of increasing Kp until oscillating behavior (Ziegler-Nichols method) .................... 43
Figure 31: Output response of a PID control system tuned by parameters of Z-N method .................... 44
Figure 32: Output response of a PID control system with adjusted parameters of Z-N method .................... 44
Figure 33: Output response of a fuzzy control system with set temperature 150°C .................... 45
Figure 34: Output response of a fuzzy control system subjected to a disturbance .................... 46
Figure 35: Output response of a PID control system with gains of figure 26, except Kp = 9.5 .................... 47

# Abstract

GREFEMA group at the University of Gerona is currently developing an automotive thermoelectric generator (ATEG). A major challenge is the temperature management on the hot heat exchanger side of the ATEG. On the one hand, the high temperatures of the exhaust gases must not exceed the temperature limit of the thermoelectric (TE) modules. On the other hand, it is more efficient to keep the temperature as high as possible so that the temperature difference between the hot and the cold side of the TE junctions is maximal. The aim of this master's thesis is to design a control system that keeps the temperature within the ATEG almost constant at its optimal target value.

The temperature measurement is executed with a K-type thermocouple in two different ways. In the first method, the MAX31855 converter and the libraries in the Arduino IDE are used to read out the temperature. In the second method, a DAQ device transmits the temperature reading directly into LabVIEW. In this graphical programming environment, the control operation takes place. The control loop has been designed two times, with another control algorithm for each. One is based on PID control, the other one on Fuzzy Logic. Finally, a graphical interface has been built to monitor in real time and possibly control the process manually.

Experimental results illustrate the performances of both control loops and a discussion at the end clarifies in which conditions which control algorithm could be suitable for the temperature control of an ATEG.

# Abstract (Nederlands)

GREFEMA groep van de universiteit van Girona ontwikkelt een automotive thermo-elektrische generator (ATEG). Een grote uitdaging hierbij is de temperatuursbeheersing aan de warme warmtewisselaar kant van de ATEG. Enerzijds mogen de hoge temperaturen van de uitlaatgassen de temperatuurslimiet van de thermo-elektrische (TE) modules niet overschrijden. Anderzijds is het efficiënter de temperatuur zo hoog mogelijk te houden zodat het temperatuurverschil tussen de warme en de koude kant van de TE juncties maximaal is. Het doel van deze master thesis is om een controlesysteem te ontwerpen dat de temperatuur in de ATEG vrijwel constant houdt op zijn optimale gewenste waarde.

De temperatuurmeting wordt uitgevoerd met een K-type thermokoppel op twee verschillende manieren. De MAX31855 converter en bibliotheken uit de Arduino IDE worden gebruikt om de temperatuur uit te lezen in de eerste methode. In de tweede methode stuurt een DAQ device de temperatuurmeting rechtstreeks naar LabVIEW. In deze grafische programmeeromgeving gebeurt de controle werking. De regelkring is twee keer ontworpen, maar met een ander controle algoritme. De ene is gebaseerd op PID, de andere op Fuzzy Logica. Tot slot is er een grafische interface gebouwd om het proces in real time te monitoren en eventueel manueel aan te sturen.

Experimentele resultaten illustreren de prestaties van beide regelkringen en een discussie aan het einde verduidelijkt onder welke voorwaarden welk regelalgoritme geschikt zou kunnen zijn voor de temperatuurcontrole van een ATEG.

# 1 Introduction

GREFEMA is a research group of the University of Girona, situated in the Polytechnic School at Campus Montilivi. This engineering group is active in the fields of fluid mechanics, energy and environment. Particularly, GREFEMA attempts to design energy-saving strategies, and integrate more renewable energy in the residential and industrial sector. Furthermore, the group explores techniques to increase the efficiency of conventional systems [1].

A major focal point of GREFEMA is the thermoelectric generator (TEG) in waste heat recovery applications, to reduce greenhouse gases and boost the performance of the thermal device [1]. The automotive industry is probably the most appealing and active sector for thermal energy recovery, since approximately 60% of the energy produced in an internal combustion engine (ICE), is dissipated as waste heat through the exhaust gases and the engine coolant [2]. So the goal of the group now is to develop a prototype of a TEG for automotive applications. There are a lot of aspects that should be taken into account when developing an automotive TEG. While a part of the research group is concentrating on the design and materials, the purpose of this master thesis is limited to the thermal management of the automotive TEG.

## 1.1 Problem statement

Automotive thermoelectric generators are TEGs built into a road vehicle. An ATEG converts the waste heat available in the exhaust gases coming from the ICE into electricity. This electrical energy can be stored and utilized for various electrical inputs of the car which reduces fuel consumption and emissions [3].

One of the main challenges with the implementation of a TEG in a vehicle is their thermal management under variable load.
Firstly, there are temperature limitations. The maximum temperature of most thermoelectric materials being considered for automotive TEG, which is approximately 250°C in this case, is often exceeded by the high temperatures of the exhaust gases, especially under high engine load. These temperatures vary from 500-600°C for gasoline engines and 700-800°C for diesel engines [3]. This causes damage to the TE module and eventually leads to a complete breakdown of the thermoelectric generator. To avoid those temperature overshoots under high exhaust load, it is highly necessary to regulate the temperature at the hot side of the module junctions to a level which is tolerable for TE materials.
Secondly, the output power of a TEG depends on the temperature differential across the TE junctions. So the exhaust gas temperature still needs to be high enough to have a significant temperature difference between the cold side and the warm side of the TEG.
Considering both previous conditions, the ATEG needs to operate near its limits. And in order to reach such a constant and optimized temperature, a bypass system is required. This control system, consisting of sensors and actuators, deflects a certain amount of the exhaust gases based on the engine load, so that module overheating is prevented and output power is maximized at the same time.

## 1.2 Objectives

Concretely, the main goal of this master thesis is to design an appropriate control system that bypasses the exhaust gases, and thus rejects the excessive thermal energy to respect the limit temperatures of the chosen TE materials. This system will consist of a valve, whose position is regulated in function of the varying temperature at the TE modules' hot side. This valve is built into the design of the prototype. Since the mechanical aspects are the focus of the other part of the research, this thesis does not discuss the mechanical hardware choices, but only determines the possible electrical/electronical hardware and software for realizing the bypass system. This extensive goal is divided into sub-objectives.

The first objective is to research which hardware and software components are suitable for this application and justify every choice.

As the bypass system is a control loop, the next step will be to make a selection of one or more possible control algorithms. A control algorithm is a type of logic programmed into a controller, which helps him to process between the process variable and the setpoint, called the error signal, in order to generate an appropriate action for the system to equalize both values.

The control program for the bypass system is the third objective. This program will also contain a graphical user interface (GUI) to monitor the process clearly.

Finally, the system will be implemented in the prototype. The setup will be optimized until the requirements of the research group are achieved.

## 1.3 Material and method

The first phase of this project involves determining the hardware and software. The program and interface will be created in LabVIEW. This systems engineering software is a graphical development environment in which every aspect of the application is visualized. LabVIEW offers easy communication with measurement instruments, data acquisition and design methods for custom user interfaces [4]. The main hardware components are the motor to manipulate the bypass valve and a microcontroller to manage the complete system. Based on a literature study, different types and brands will be compared. It is important to decide this hardware at the beginning, because both components are crucial for the iterative process of writing and testing of the source code.

In the second phase, different control algorithms will be evaluated and analyzed. Every algorithm will be explored by theoretical facts, as well as by practical studies about similar cases that are accomplished in the past will be examined. As a result, a clear representation of which control techniques are commonly used and which one will meet our requirements is obtained.

For the last phase a LabVIEW course [5], internet and the advice of co-researchers will be the auxiliaries for the realization of the program. Initially, a structural diagram of the program will be made on paper. This visual structure is a really helpful basic to start from, since the source code in LabVIEW is also visually reproduced. Then, the communication between the microcontroller and LabVIEW will be set up. After the program is built, a graphical interface will be created to help the user with monitoring the process or even allowing him to control the system manually.

# 2 Literature study

Theoretical aspects are discussed in this chapter in order to gain more insight into the various components that are offered during this thesis. First, a literature study was carried out about the principles and structure of the thermoelectric generator.
In order to assess the performance of the control system, knowledge about some common time-domain specifications is required. So some parameters will quickly be discussed
Finally, some possible control algorithms that can be used for temperature control were investigated. Only the most relevant ones for this thesis will be treated in detail.

## 2.1 Automotive thermoelectric generator

### 2.1.1 Principle

An ATEG is based on one of the three thermoelectric effects, namely the Seebeck effect. Concretely, this means that an electrical voltage can be generated by applying a temperature differential. This temperature difference is located on both surfaces of the TE module(s), which form(s) the core of an ATEG. Figure 1 shows that such a TE module is composed of P-type and N-type semiconductors. These are all electrically connected in series but parallel thermally [6], [7].

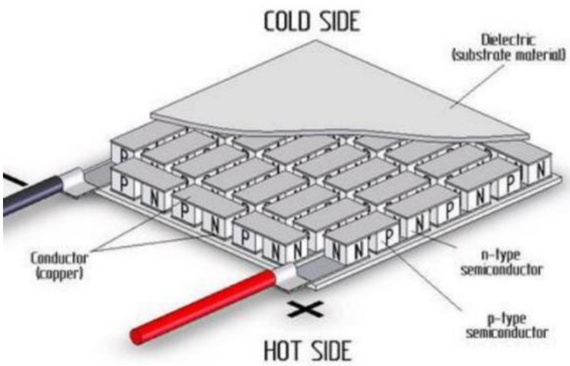



*Figure 1: Architecture of a TE module [8]*

As the temperature rises on one side, the load carriers migrate from the warm side to the cold side. If the majority of the charge carriers are positive (P-type), a positive charge will be built upon the cold side with a net positive potential as a result. Conversely, the negative charge carriers in an N-type semiconductor will cause a negative potential on the cold side. When connecting the ends of the two types of semiconductors on the hot side and connecting a charge on the cold side, the voltage due to the Seebeck effect will cause a current to flow [6]. Electric power is now supplied. Summarized, the temperature difference provides the voltage, but it is the heat flow that activates the electric current.

The hot and cold side of figure 1 is created in an ATEG by the exhaust gases flowing through the ATEG and coolant or ambient air on the outside. According to this principle, an ATEG thus converts a part of the lost heat present in the exhaust gases, originating from an internal combustion engine, into electricity. The generated electricity can then be transported to the battery and the rest of the vehicle's electrical system. This extra energy from the ATEG shortens the operating time of the alternator, and hence also the engine load.
Less load means less energy must be supplied. Briefly, the amount of fuel consumed will reduce as well as emissions [9].

### 2.1.2 Structure

The ATEG has a sandwich structure. The thermoelectric modules are as it were 'sandwiched' between two heat exchangers. As mentioned before, a TE module has a hot and cold side [6]. As a result, one of the heat exchangers will be at a high temperature and is called the warm side of the ATEG. The other heat exchanger at a lower temperature is consequently referred to as the cold side. Between the metal heat exchangers and the TE module, there is an electrically insulating/thermally conductive layer (see substrate layer figure 1). In figure 2 two possible types of ATEG construction for exhaust heat conversion are illustrated.

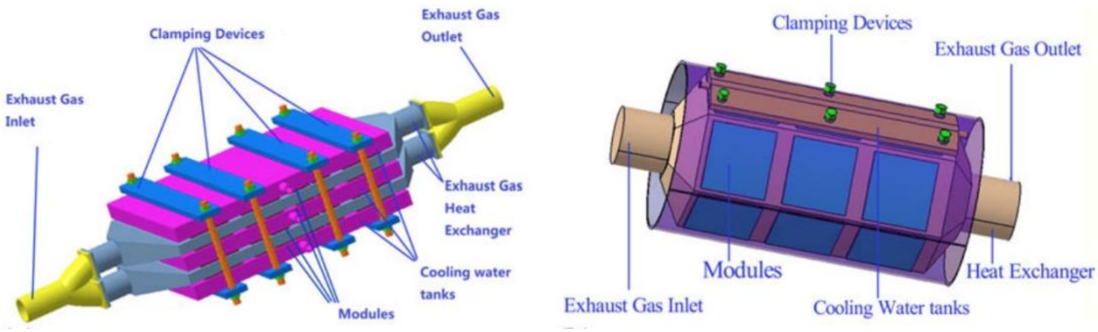



*Figure 2: Schematic diagram of ATEG with a) hexagonal-prism-shaped b) plate-shaped heat exchanger [10]*

#### *2.1.2.1 TE modules*

The TE modules are the main components of a thermoelectric generator since they are responsible for the electric production and thus play a key role in an ATEG its overall efficiency. For this reason, there is a lot of research into innovations of TE materials. Today, there is a wide range of TE materials. They can be subdivided according to different properties, such as conversion efficiency, crystal structure, and temperature range [3], [7].

For the latter property, the TE materials are classified into 3 groups: low (200-500 K), moderate (500-800 K) and high-temperature materials (> 800 K) [7].
The first group is, of course, the most critical one. With TE materials of this group, in case of too long exposure to the hot gases, parts of the TE legs will be dissolved through sublimation. Qualitative temperature control can reduce this sublimation loss. This is exactly what this research is all about. Filling the spaces between the TE legs with an extremely low thermal conductivity aero-gel could be another possible solution [3].

Another important feature in addition to the temperature range is conversion efficiency. The efficiency of a TE material is related to the dimensionless figure of merit ZT. The ZT is given by

$$ZT = \sigma . S^2 . T / \lambda \qquad (1)$$

where T is absolute temperature, S the Seebeck coefficient, $\lambda$ thermal conductivity and $\sigma$ electrical conductivity [11]. When the operating temperature of the modules is approximately around the point where the ZT of the leg materials is highest, the maximum efficiency will be achieved. In other words, the search for more efficient TE materials goes hand in hand with an improvement of this ZT [3], [7]. Figure 17 below gives an example of such a figure of merit ZT. It describes some recently developed TE materials, more specifically TE nanocomposites with the ZT according to their temperature range.

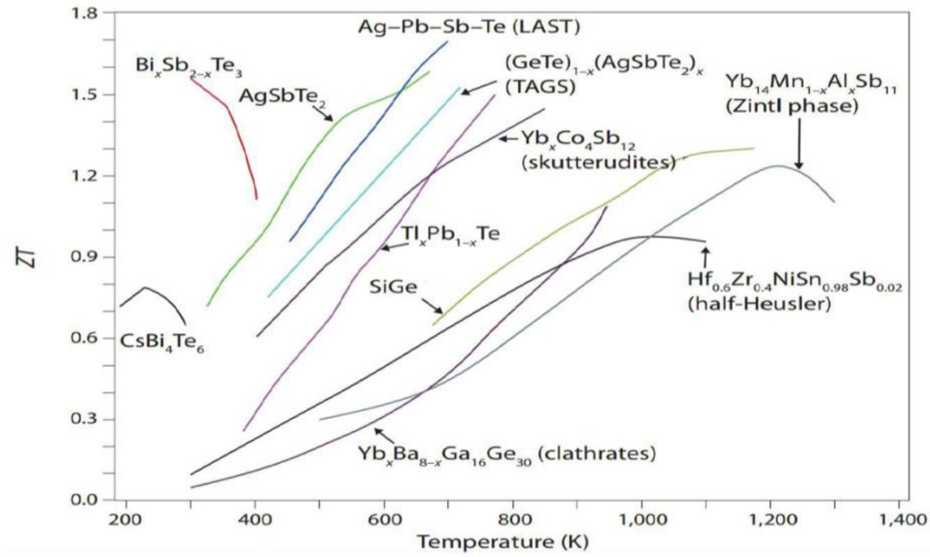



*Figure 3: Overview of the ZT of recent TE materials as a function of temperature [7]*

## 2.1.2.2 Hot side heat exchanger

The function of the heat exchanger on the warm side of the TE module is to extract the heat from the exhaust gases. It is also next to the TE modules crucial for the electrical power output. The ideal heat exchanger aims for the lowest possible weight combined with high efficiency, without causing too much back pressure that disturbs the flow of the exhaust gases. This back pressure would increase fuel consumption, as the engine would have to work harder to overcome this "extra resistance". Since the purpose of an automotive exhaust thermal generator is to reduce fuel consumption, the back pressure should be kept as low as possible at all times [3], [9].

The temperature in the heat exchanger can, according to the exhaust gas flow direction, be lower towards the end of the heat exchanger than at the exhaust gas inlet. This leads to a reduced power output of the TE modules which are more at the end of the flow. Other flow configurations of the exhaust gases and refrigerant, such as parallel flow or counter flow, can provide a solution depending on the system.

#### *2.1.2.3 Cold side heat exchanger or heat sink*

The electrical power output not only depends on the hot side temperature (TH) of the TE but also on the temperature difference across the TE junctions. Therefore it is also fundamental to minimize the cold side temperature (TC). This can be achieved with the use of a separate insulated cooling system such as normal cooling fins or a water-cooled heat sink. In addition, the cold side heat exchanger can also be integrated into the existing cooling circuit of a vehicle, that is used to cool the engine. In contrast to a separate cooling circuit, a complete integration avoids the extra required space and the increased weight. However, the capacity of the pump in the motor cooling circuit should probably be increased to avoid overheating due to the additional heat from the cold side of the TE modules [3].

## 2.2 Control theory

The temperature inside the ATEG, that needs to be hold at the limit of a TE module, depends on the amount of exhaust gases flowing through it. The mass flow rate of the exhaust gases is controlled with a valve, driven by a motor. The temperature of the ATEG will be measured and used by a controller, that will compare this measured value with the desired value. Depending on the difference between those two, a controller will generate a signal for the motor. As a result of the signal the motor gets, he will put the valve in a position, which will change flow rate so that process temperature will evolve towards the desired value. Since the mass flow rate of the exhaust gases changes, as well as their temperature, the process above will be executed continuously. This is also called a closed-control loop or feedback loop. The essential information for this thesis about such control loops is discussed in this part of the literature study.

### 2.2.1 Performance indicators

A control system can be analyzed in 2 ways, the time domain, and the frequency domain. In this master thesis, the temperature control system will be evaluated in the time domain. In order to be to analyze the performance of a control system in the time domain correctly, some indicators are required that characterize the behavior/response of the system. Figure 4 summarizes the main ones in a typical step response of a PID controlled closed-loop system.

**Rise time ($t_r$)**
For under-damped systems, the rise time is the time the system needs to evolve from its current value (0%) to its final value (100%). In the case of an over-damped system, the time is calculated between 10% and 90% of the steady state. [12].

**Peak overshoot ($M_p$)**
The overshoot indicates how much the process exceeds the final value at peak time, expressed as a percentage of the final value. Peak time is the time when the response reaches its peak value for the first time [12].

**Settling time ($t_s$)**
The settling time is the time needed for the process variable to get within a tolerance band of about 5% of the final value and stay within it [12].

**Steady-State error (SSE)**
The Steady-State error is the difference between the process variable and the setpoint at the end when the process variable remains stable and unchangeable [12].

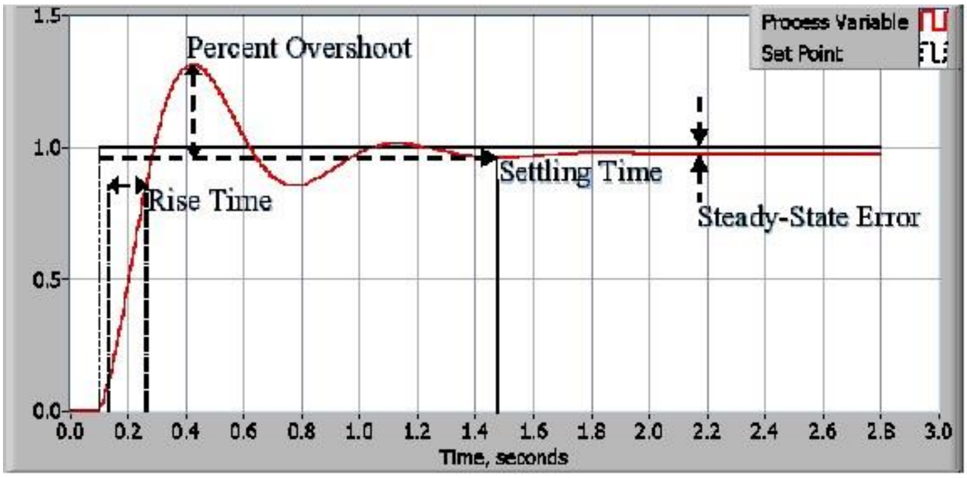



*Figure 4: A typical response of a PID closed-loop system*

### 2.2.2 Control algorithm

A control algorithm is what is programmed into a closed-loop controller and determines the behavior of it. Such a controller will compare a measured value of the process variable with the desired value that the variable should achieve. The error signal, which is the difference between the measured value and the setpoint value, is converted into an appropriate actuator signal. The actuator is manipulated through this signal in such a way that it brings the process variable closer to its desired value. In fact, the control algorithm is thus responsible for how fast, accurate and stable the closed-loop control system is regulated. The most common control algorithms are discussed below.

#### *2.2.2.1 Bang-bang control*

Bang-bang control, also known as on/off control or hysteresis control, is the simplest and cheapest algorithm in control engineering. The output of this controller is always 0% or 100% and nothing in between. Consider, for example, an air conditioning system. An on/off controller will measure the temperature and compare it with the desired value. If the measured temperature is higher than the desired value, the output is true (100%) and as a result, the air conditioning is turned on. If, because of this action, the temperature drops below the setpoint, the air conditioning system will switch off again [13], [14].

Figure 4 shows the behavior at the output of the control loop in comparison to the position of the on/off control for a given case. this case exposes immediately a problem of the on/off controller. At point A, the controller will do the same action as at point B. However, at point B, the process variable direction converges to the desired value and actuation may not even be required. While in point A the process deviates further and further away from the setpoint and a strong action may be needed to reverse this undesired pattern of behavior [14].

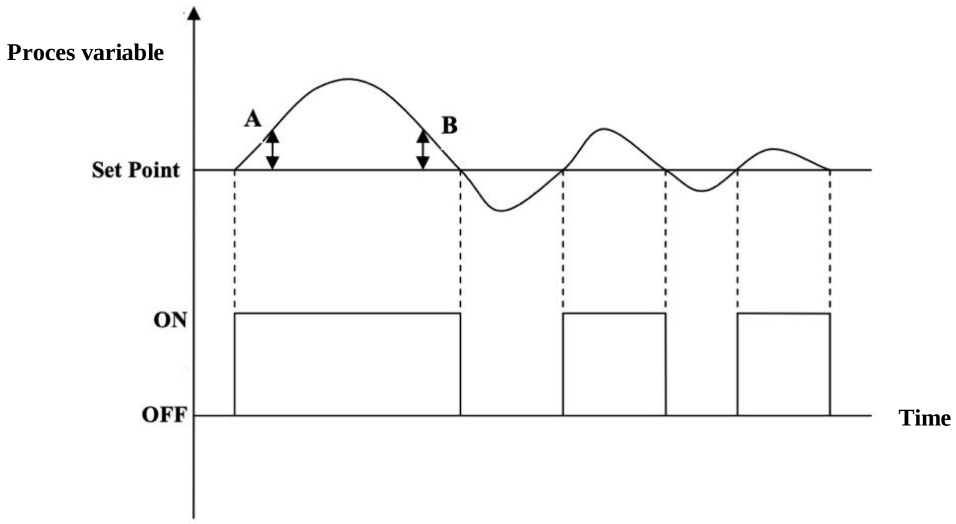



*Figure 5: The operation of a bang-bang controller and the corresponding behavior of the control system*

Theoretically, an on/off controller changes its mode at the setpoint. In practice, there will be processes where the output variable quickly oscillates around the setpoint, causing the output device to switch very fast as well. As a result, the system will be unstable, and valves or contacts can be damaged [14]. This can be solved by introducing a narrow range around the setpoint. This narrow range is also called the adjustable deadband or hysteresis (see figure 5). Instead of switching mode at the setpoint, the on/off controller will now change its mode at the limits of the deadband [13]- [15].

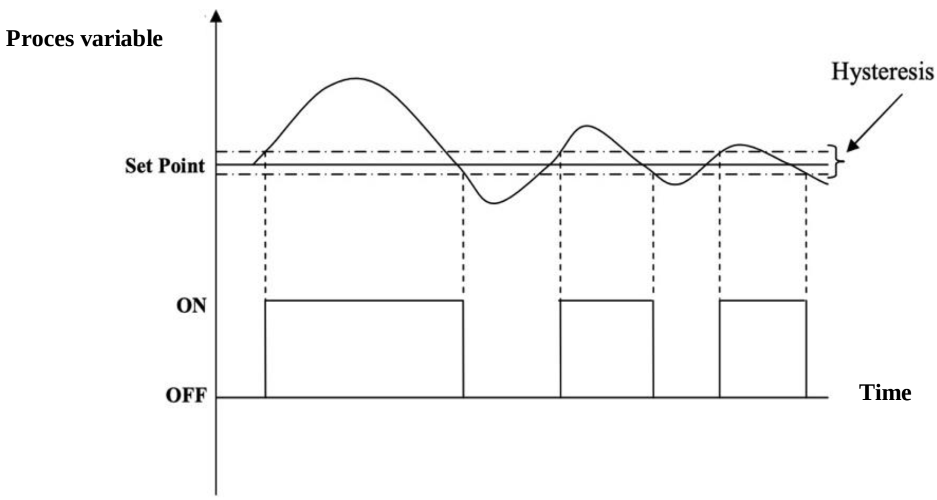



*Figure 6: An adjustable deadband/hysteresis with on/off control*

In short, an on/off controller provides a cyclically varying response. In applications where the output variable can be controlled within a tolerable range, on/off control is a suitable and inexpensive solution. However, in industrial processes, it is often required that the process variable is controlled more quickly and very accurately. In this case, on/off control is not applicable.

### 2.2.2.2 *Proportional-Integral-Derivative (PID) control*

The PID controller is without a doubt the most commonly used form of feedback in the industry today. Although most of them are often PI because the D term is not used regularly, as will be seen later. A PID consists of three terms: a proportional (P), a derivative (D) and an integrative (I) term. In fact, the I-action represents the past, the P-action the present and the D-action the future. This will become more clear in the next sections when each of these three terms is reviewed separately. Afterward, all three terms are combined into the PID controller [16].

**Proportional term P**

As the name already implies, the proportional term will produce a control signal that is proportional to the current error signal e(t). Its output can be obtained simply by multiplying the error by a constant proportional gain $K_p$.

$$u(t) = K_p\, e(t) \tag{2}$$

Raising the $K_p$ will reduce the system's steady state error. Despite the reduction of SSE, the P-action will never be able to eliminate the steady state error completely. In addition, a higher proportional gain also increases the response speed of the system (or lowers the rise time) when a disturbance occurs. A too high $K_p$ often leads to a violent reaction, resulting in overshooting and oscillation [17], [18].

**Integral term I**

The Integral term takes into account both the size of the error signal and the duration of the error. The error signal e(t) is integrated over time, hence an ever larger control signal is produced. This process continues until the error has completely eliminated. The output of an I controller can be obtained as follows.

$$u(t) = K_i \int_0^t e(t)dt \tag{3}$$

Therefore, the integral term aids in the disappearance of the remaining steady state error of the P term. However, it also increases the overshoot and instability of the system if the $K_i$ gain gets too high [17], [18].

**Derivative term D**

With the previous two terms, a zero steady-state error and a high response speed are already obtained. By implementing the derivative term, stability and settle time can be improved. The D term takes into account the speed of the error change and tries to reduce it to zero. The output is then the derivative of the error, multiplied by the derivative gain $K_d$ [17], [18].

$$u(t) = K_d \frac{de(t)}{dt} \tag{4}$$

Typically, the $K_d$ is taken quite small, because an oversized derivative gain results in a very turbulent controller behavior. This should be avoided at all costs.

**PID**

By combining the previous terms, a PID controller is created as shown in figure 7. There are several methods for fine-tuning a PID. Which techniques are best depends on whether the system can be taken offline. If this is the case, a step function is often applied at the input. By measuring the output, the control parameters can be optimized by means of the response. Two of the more common techniques are discussed below. These are also used in this thesis [17].

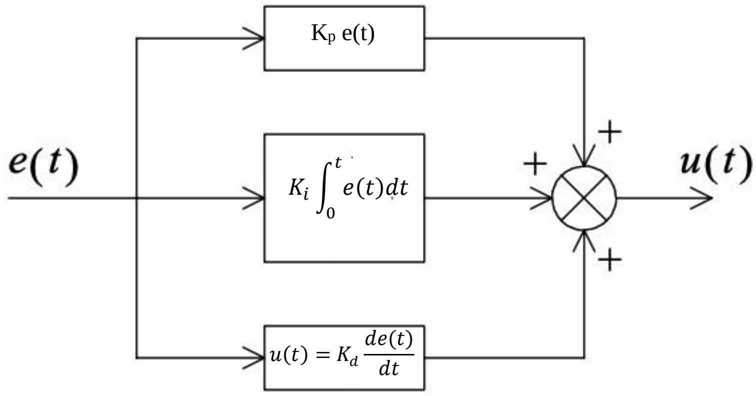



*Figure 7: Schematic of a (parallel) PID controller [17]*

## Manual tuning

In this technique, the $K_i$ and $K_d$ gains are first set to zero. The $K_p$ is raised until the output response starts to show oscillating behavior. Then the gain of P term is reduced to half of the obtained value. Next, the $K_i$ is set to the value at which the process eliminates the offset within the desired time. However, a too large $K_i$ causes an unstable behavior as mentioned before. Finally, if necessary, the $K_d$ is increased slightly to attenuate the oscillation and overshoot. In some systems a small overshoot is acceptable and the D-action is kept very small or even zero in order to reach the desired value as soon as possible. Other systems cannot tolerate overshoots, hence an overdampened closed-loop system is required. Table 1 below presents a brief overview of all control parameters and their influence on the performance of the closed-loop system [12].

*Table 1: Impact of increasing control parameters on performance indicators [18]*

| Parameter | Rise time | Overshoot | Settling time | Steady-state error |
|---|---|---|---|---|
| $K_p$ | Decrease | Increase | Small change | Decrease |
| $K_i$ | Decrease | Increase | Increase | Eliminate |
| $K_d$ | Small decrease | Decrease | Decrease | No effect in theory |

## Ziegler-Nichols method

Ziegler-Nichols is a heuristic method. Here, as well, the I-action and D-action are switched off at the beginning. The $K_p$ is increased until a periodical oscillation is created at the output. The value is defined as the ultimate gain $Kp_u$. Also, the period of oscillation $T_u$, which is called the ultimate period, is measured. Based on these values value, the Ziegler-Nichols technique prescribes the following formulas for the three control parameters [12].

*Table 2: Formula set offered by Ziegler-Nichols method for control parameter determination in a closed loop [12]*

| | $K_p$ | $K_i$ | $K_d$ |
|---|---|---|---|
| **P controller** | $0.5Kp_u$ | $\infty$ | 0 |
| **PI controller** | $0.45Kp_u$ | $T_u/1.2$ | 0 |
| **PID controller** | $0.6Kp_u$ | $T_u/2$ | $T_u/8$ |

PID is and will still stay the dominant closed-loop controller in the industry. Nevertheless, PID control has some limitations. The main problem is that a PID performs worse with non-linear systems, as PID controllers are linear. Some temperature control systems are an example of non-linear systems. In these systems, the temperature is actively increased through a heat source but passively reduced (by switching off the heat source). This means that the drop in temperature is slower than the rise in temperature. In other words, once the process temperature exceeds the setpoint, the overshoot will be corrected very slowly. This can be solved by setting the control parameters so that the closed-loop system is overdamped to reduce or eliminate the overshoot. Though this is not ideal and is at the expense of the performance of PID.
Another disadvantage of PID is that it has fixed parameters and therefore no adaptive or self-adjusting controller can be created. As a result, it does not perform optimally in case of changes in the process [16], [18].

#### *2.2.2.3 Fuzzy Logic control*

In recent years, the use of fuzzy logic has significantly increased. Fuzzy or 'vague' logic, unlike traditional logic, does not work on the basis of 0 and 1 (true or false) but expresses something with a value between 0 and 1. It is a form of soft computing that tolerates the imprecision and uncertainty of the real world. Therefore, no detailed and quantitative information is needed from the system to be able to model it and make decisions about it. As a result, fuzzy is not only limited to highly technical systems. Fuzzy logic is built on the basis of natural language. Although words are not as exact as numbers, their use is closer to human reasoning, which simplifies them. Fuzzy can be found in car control systems, household applications, cameras, etc. [19], [20], [21].

In terms of control theory, fuzzy logic offers a number of advantages over PID. A fuzzy system can handle non-linear systems. In addition, fuzzy logic can be used to design an adaptive or self-regulating system, which is not the case with PID. Also, as mentioned above, the fuzzy logic controller (FLC) does not require precise knowledge of the system. This makes it suitable for when there is little or unreliable information about the system. Briefly, fuzzy logic is a very flexible, robust and cheap technique.
Fuzzy logic also has some shortcomings. Because fuzzy logic is an intuitive method, there is not really an explicit methodology to obtain a solution. A lot of practical experience and a solid basic knowledge of the control technique is required for proper fuzzy implementation [19].
A fuzzy system consists of three main components: fuzzy linguistic variables, membership functions and rules. The architecture of a fuzzy system is discussed in detail below, as well as the fuzzy controller that controls this system [22].

**Fuzzy system**
The linguistic variables are actually the input and output variables of the system that needs to be controlled. Each linguistic variable is divided into a number of fuzzy subsets or linguistic terms. These are categories of values of the linguistic variable. The number of linguistic terms is usually uneven, ranging from three to seven. An example of a linguistic term, for the linguistic variable temperature, is "warm". What "warm" means and to what extent a temperature is "warm" should be described. This is done by means of membership functions [19]- [22].

These membership functions are numerical functions that represent the linguistic terms. They indicate to what extent a considered input value belongs to a linguistic variable within its terms, which is called the degree of membership. Membership functions can take different forms as shown in figure 8. The x-axis shows the elements of the linguistic variable. The y-axis represents the degree of membership. The set of truth levels of the elements for a population concerning a linguistic variable is called a fuzzy set [22].

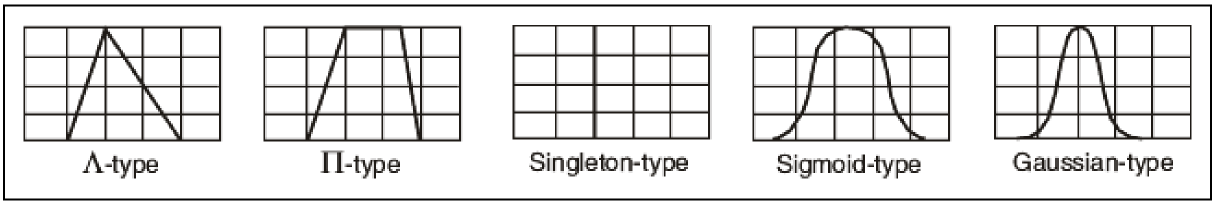



*Figure 8: Different types of membership functions, from left to right: Λ-type (triangular shape), Π-type (trapezoidal shape), singleton-type (vertical line shape), Sigmoid-type (wave shape), and Gaussian type (bell shape) [22]*

Finally, only rules are needed to complete the fuzzy system. These rules are actually relationships between the output linguistic variables and input linguistic variables based on their linguistic terms. A rule is made up of antecedents, also called the "IF" part, and consequents, or "THEN" portions. This can be formulated as follows: "IF A, THEN B". A set of rules is called a rule base and is often represented by a matrix. Figure 9 below shows the control matrix for the relation "IF inlet is LARGE, THEN outlet is LARGE". The linguistic variables inlet and outlet represent the rows and columns [19].

| | Large | Medium | Small |
|---|---|---|---|
| Large | 1 | 0 | 0 |
| Medium | 0 | 0 | 0 |
| Small | 0 | 0 | 0 |

*Figure 9: Rule matrix for rules consisting of one input and one output linguistic variable [19]*

**Fuzzy controller**
The block diagram in figure 10 on the next page illustrates the steps that a fuzzy control performs to obtain a specific crisp output from a crisp input. The fuzzy controller consists of three parts. The control matrix is the heart and forms the connection between the fuzzy input sets and output sets.

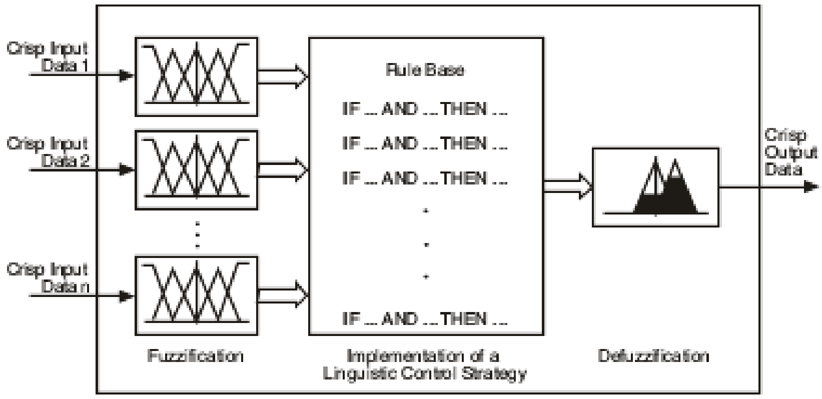



*Figure 10: Block diagram of a Fuzzy Controller [22]*

The first step is fuzzification. For each incoming crisp measurement, the degree of membership of the linguistic input variable within the corresponding linguistic terms is determined. Figure 11 shows an example of the fuzzification step. For the temperature 15°C, there is a degree of membership of 20% with the linguistic term cool and 70% with cold. The other degrees of memberships are zero. At 36°C, it is 87% zero and 12% warm [19].

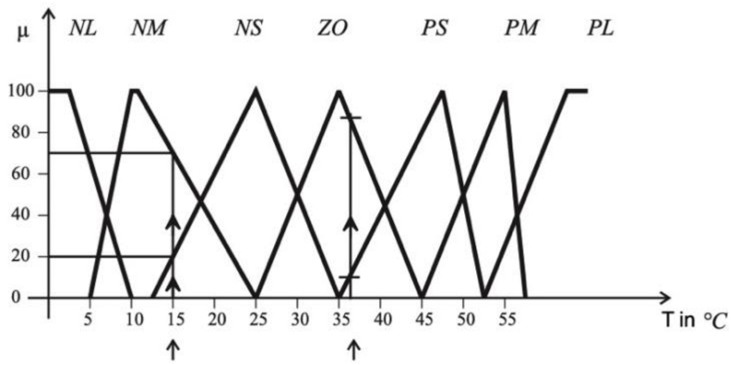



*Figure 11: Fuzzification[1] [19]*

When the input values of the fuzzy system are fuzzified, the controller will use the found input linguistic terms and rule base to determine the linguistic terms of the output linguistic variables [22]. A simple example of what a rule can look like has already been given in Figure 9. But rules can also have multiple antecedents, for example in the case of multiple input variables. It is even possible to combine rules. The minimum and maximum operators "AND" and "OR" are then used to determine the output. An example of such a rule is: "IF current temperature = hot AND desired temperature = cold, THEN heater = HIGH”.
The last step in the process is defuzzification. Here the degrees of membership of the output linguistic variable within its linguistic terms are converted back to crisp output values. There are several defuzzification methods. The most commonly used method is the center of area (CoA). In CoA the geometric center of the area, located under the scaled membership functions and between the range of the output variable, is the value of the output linguistic variable. Another simpler method can for example be the center of maxim (CoM) or singleton method, where the output linguistic variables are defined as singletons [20], [22].

[1] NL (=Negative Large); NM (=Negative Medium); NS (=Negative Small); ZO (=Zero); PS (=Positive Small); PS (=Positive Medium); PL (= Positive Large)

# 3 Material and method

This chapter first describes the principle and model proposed to control the temperature in an ATEG. Next, the material used to realize this system is presented as well as some arguments why this material was chosen. Optionally, alternative components are also offered, which could eventually be integrated in the future. Finally, a more detailed presentation is given of how everything is connected to each other in the actual test setup.

## 3.1 Concept

The idea, outlined in figure 12, is that the position of a butterfly valve, placed at the exit of the ATEG, is precisely controlled according to the read-in temperature. This temperature is read between the warm side of a thermoelectric module and the warm heat exchanger. If the measured temperature is lower than the limit of the TE module, the valve will close more. This will block the exhaust gases at the end of the ATEG. As the exhaust gases spend more time in the ATEG, the accumulated heat has more time to rise through the hot heat exchanger towards the TE modules. So the heat exchange is increased. As a result, the temperature at the hot limit surface of the TE modules will rise. Conversely, if the temperature is too high, the valve is opened further so that the exhaust gases will pass through the ATEG at a high flow rate. Here will be a smaller heat transfer from the gases to the heat exchanger. When closing the valve, the back pressure and suffocation of the engine must be taken into account. Therefore, the valve in the control system must never be completely closed. The heat that is released to the environment is done by means of a cold heat exchanger. This is not shown in the sketch below. Also, note that the heat output is over-drawn to demonstrate the principle.

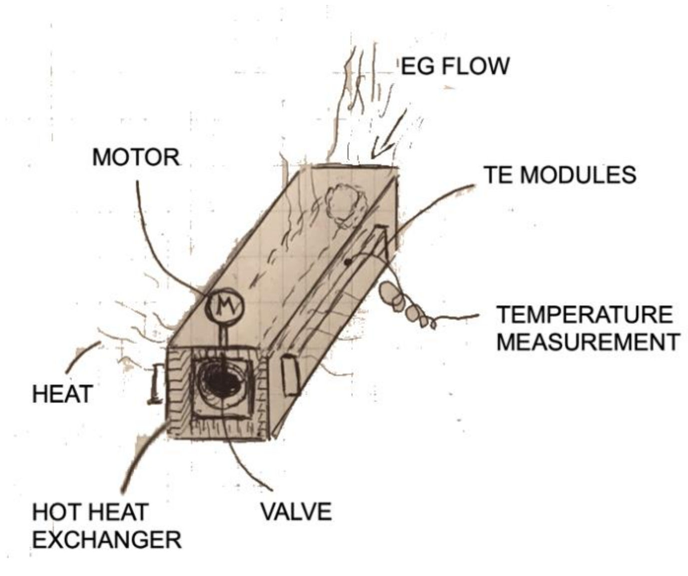



*Figure 12: Design principle for the temperature control of the ATEG*

Starting from the principle above, a more concrete control model was proposed. The diagram of blocks in figure 13 schematically shows the components and the relations between these components for this proposed system. The temperature signal, which is measured by the sensor, will be sent to the microcontroller after any processing. The microcontroller communicates this temperature value to LabVIEW. The chosen type of controller will compare this value with the desired value of 250°C. Based on the error, an output signal is produced, which is sent to the motor via the microcontroller. As a result of this control signal, the motor will turn the valve more open or more closed. There is also a GUI in LabVIEW (indicated by dot line) which monitors the process temperature, the control signal to the motor and the position of the valve during the tests. In addition, the position of the valve can also be adjusted and set manually by means of this graphical interface (see the double arrow). This makes it easier to align the motor and valve. The proposed model is suitable for lab settings in order to evaluate the behavior and response of the system and the actual instruments in real time.

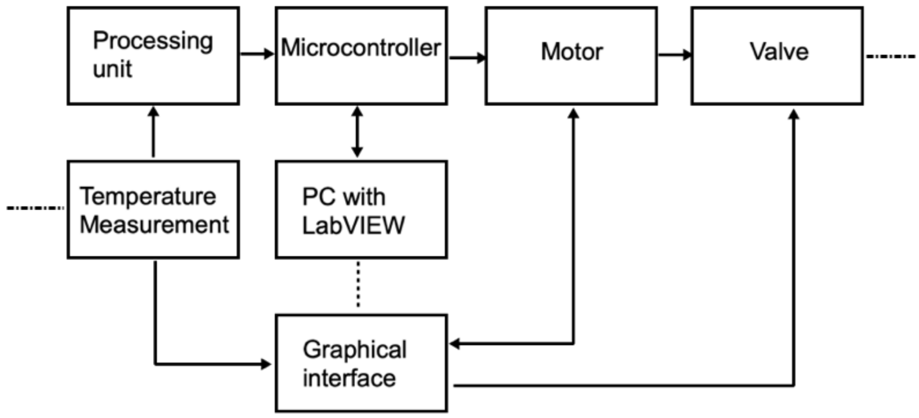



*Figure 13: Block diagram of the proposed control system*

## 3.2 Hardware

### 3.2.1 Microcontroller

**Arduino MEGA 2560**

For this temperature control system, the Arduino Mega 2560 was selected as microcontroller. Arduino Mega 2560 is one of the larger microcontroller boards among the Arduino series, but just as easy to use and implement. Arduino is based on the Atmega2560. It has pretty much everything that other Arduino microcontrollers like the UNO also contain. For example, the 16MHz ceramic resonator, the reset button, analog and digital I/O pins, USB connection, power jack, etc. are on board. What makes the Mega so unique and different from the other controllers, is its capacity in terms of memory and pins. The Arduino Mega 2560 provides up to 54 digital I/O pins, including 16 that can be used as an analog pin. It also has 256 kb flash memory, which is a lot more than the 32kb of the UNO. The Arduino Mega is not only physically larger, but also technically. It makes it suitable for projects with more complex circuits, where more sensors and actuators are used, and more memory is required. This project could be done as well with a smaller Arduino. However, in truck applications, several thermoelectric generators could be placed in parallel to increase the electrical power output. A Mega with its increased capacity will certainly be useful here. Also because the Arduino Mega is capable of maintaining multiple tasks at the same time. Finally, like all Arduino microcontrollers, Mega can also be easily programmed with the Arduino IDE software, which is written in C [23].

### 3.2.2 Temperature measurement

**K-type thermocouple**

A thermocouple relies just like the thermoelectric generator on the Seebeck effect. It consists of two dissimilar metal wires that are welded at one end. At this created junction, the temperature will be measured. If a temperature change occurs at the junction, a proportional voltage will be induced. These voltages are really small. In order to measure accurately, a so-called cold junction compensation module needs to be used. Thermocouples offer a wide temperature range, going approximately from -200°C to 2000°C, at low-cost. This wide measurement range is a very interesting property for this project, regarding the high exhaust gas temperatures.

Thermocouples, however, are not the most precise sensors. The typical error is around 2°C. Considering in this system the sensor will operate over a large range, the error can be neglected. Another problem the noise sensitivity, because of the small produced voltages. Twisting the two insulated parts of the thermocouple cable could be a solution to ensure both wires receive the same noise.

K-type thermocouples are the most popular ones. Consisting of Chromel and Alumel wires, they provide a range from -270°C to 1260°C. In addition, they work even under rugged conditions. Briefly, K-types are small, reliable and inexpensive, and thus ideal as an instrument for measuring temperature in this system [24].

**MAX31855**

Temperature with a thermocouple is measured with the cold junction at 0°C. When attaching a thermocouple to the microcontroller or another data acquisition board, the cold junction is at room temperature. This means the device to which the thermocouple is connected, will receive a lower voltage than with a cold junction temperature of 0°C. Because the generated voltages are really small, as already mentioned above, a little deviation will have a significant impact. In order to compensate the lower voltage and thus measure accurately, a cold-junction compensation is required. The MAX31855 performs such cold-junction compensation. Also, it digitizes the signal from K-type thermocouples. The output consists of a 14-bit signal with a resolution of 0.25°C. Its range is almost the same as that of a K-type thermocouple, namely -270-1800°C. MAX31855 has also the ability to detect short to ground, short to voltage supply or an open thermocouple. This seemed to be an interesting feature during the simulation tests. However, in the last weeks of the project, the MAX31855 broke. Because time was precious and waiting for another one was not an option, the MAX31855 is replaced by a data acquisition (DAQ) device of National Instruments [25].

**NI 9211 connected to cDAQ-9174 (NI)**

The NI 9211 is a thermocouple input module that is used in this case together with the NI CompactDAQ, a four-slot USB chassis. This NI 9211 contains, just like the MAX13855, an analog-to-digital converter, and a cold-junction compensation. It can aggregate 14 samples per second with a maximum accuracy of 2.11°C. This DAQ device may not be as compact and handy to implement in a system as the MAX31855. However, it offers an easy and qualitative solution in a laboratory setting [26], [27].

### 3.2.3 Motor

The valve that controls the amount of exhaust gases flowing through the ATEG, is manipulated by a motor. Three options are proposed.

**Torque motor (Sonceboz)**

The torque motor of Sonceboz (see annex A) is a rotary brushless DC actuator. It generates peak torques between 0.7-2.0 Nm in a compact design. Because this torque motor is a direct-drive actuator without transmission gear, it can deliver a response below 100 ms. It also has no integrated electronics, which makes it resistance against temperatures above 100°C. The output rotary angle of the motor is only 75° but can be extended by means of simple mechanical kinematics. To conclude, this compact and robust torque motor is ideal for fast positioning of exhaust control valves. Therefore this motor is also the primary choice [28], [29].

.

**HS-5065MG Servo (Hitec)**

Unfortunately, it was not possible to receive the torque motor of Sonceboz. So, for the simulation tests of the control system in the laboratory, the HS-50565MG servo was used as an alternative. This servo definitely does not offer the same quality and features as the Sonceboz torque motor. However, it is still good enough to get a reliable result out of the laboratory tests. And maybe in the future, the torque motor can still be integrated and replace the HS motor. The HS-5065MG of Hitec offers a rotary range of approximately 120°C. The servo is driven by a pulse width modulated signal. The “on” time of this PWM signal must be located between 900 and 2100 microseconds.

**Solenoid valve for EGR**

EGR is a technique that reduces the amount of NOx in the emissions by recirculating a portion of the exhaust gases produced by an ICE. This EGR system comes really close to the temperature control system of an ATEG. EGR also uses a valve to determine the flow rate of exhaust gases is also working with exhaust gases. So the requirements of the system in terms of temperature and other factors are pretty much the same as with the temperature control system of an ATEG. This means that a third option can be the motor used in EGR applications. If this motor is detached of the EGR valve, it can be implemented in the ATEG system. In addition, it could maybe even possible to implement the complete package of motor and valve of an EGR system. There are PWM electronic EGR valves that would fulfill perfectly the task of an ATEG valve. Such a PWM EGR valve uses a solenoid, provided with an electric current, to control the valve by means of a magnetic field. These solenoid valves [30] can also take positions from 0-100% by regulating the “on” time of the PWM signal. The use of EGR valves would mean that the mechanical design of the ATEG must be changed. This is the reason why this method for exhaust control is not used. But maybe this can also be, just like the torque motor of Sonceboz, an improvement to consider in the future.

## 3.3 Software

The program for temperature control is realized with the LabVIEW software. LabVIEW contains virtual instruments (VIs) that reproduce physical devices in real life. These VIs can immediately be integrated without the need for connecting a physical device to LabVIEW. This facilitates software design significantly. Generally, LabVIEW's graphical way of programming helps to get more insight into the system and the control of it. Finally, it allows to follow the process graphically in real time and anticipate directly to a problem through a control panel. Two control programs are built in LabVIEW. In the first one, a PID controller is used, while the second is based on fuzzy logic.
PID is the more obvious choice, as it still is the most used control algorithm in the world. It delivers a fast, accurate and robust performance, at least if the parameters are tuned correctly. The parameters will be tuned with two methods described in the theoretical section 2, which were the manual method and Ziegler-Nichols tuning method.
Fuzzy control could also not be missing in this study. However, nothing exceeds a well-tuned PID when there is much knowledge about the system, fuzzy control could still give some satisfying results for this thesis. With its robustness, it can tolerate the non-linearities that are hidden in this ATEG system, as well as the process changes. The natural language of fuzzy logic is also really easy to use and to understand. So fuzzy logic will definitely contribute to a better interpretation of what happens in the process. The most difficult task here will be the selection of the appropriate rules and membership functions since this rests more on experience and intuition.

## 3.4 Experimental setup

Figure 14 shows a complete overview of the experimental setup made in the laboratory. The grey long beam-shaped object in the lower left corner of the figure is the ATEG. As can be seen, the TE modules are attached to the 4 sides of the thermoelectric generator. In the final prototype, the TE modules should be equipped with cold heat exchangers, to transfer the heat from the gases to the environment or a cooling circuit. The absence of these cold heat exchangers makes that the ATEG gets even more inertia during the tests. The system contains already some inertia because the heating is actively done by the hot air flow, but the cooling is done passively by simply opening the valve more. As a result, the measured temperature on the hot surface of the TE module will not change fast enough when the flow rate of the exhaust gases is changed through the valve. This leads to a system that is so slow that both designed control models cannot be tested. This is slightly accommodated by the fan that blows air on the side of the ATEG. The exhaust gases are simulated with a 1500 W hot air gun and flow according to the arrow marked on the ATEG. This heat gun can be switched between 2 positions. In position 1, a hot air flow of 375°C is produced and in position 2, the air flow is 495°C. By switching between these two modes, disturbances to the process can be simulated. However, the flow rate of this air gun is much smaller than that of the actual exhaust gases. This also leads to slow temperature variations and more inertia. A real combustion engine will certainly improve the tests and the results. The external power source supplies energy to the engine. The Arduino is not equipped with enough power to complete this task. So it would reset itself because of the high power demand and lose the connection with LabVIEW. The source is fixed at 5V. Both the Arduino and the DAQ device are connected to the pc. This connection must also be maintained during the simulations, otherwise, the serial connection with LabVIEW would be lost. This communication is required to execute the program and to follow the process in real time. So a stand-alone system is not possible in this case.

The DAQ device in the back of the picture digitizes the signal from the K-type thermocouple so that it can then be sent to the PC.

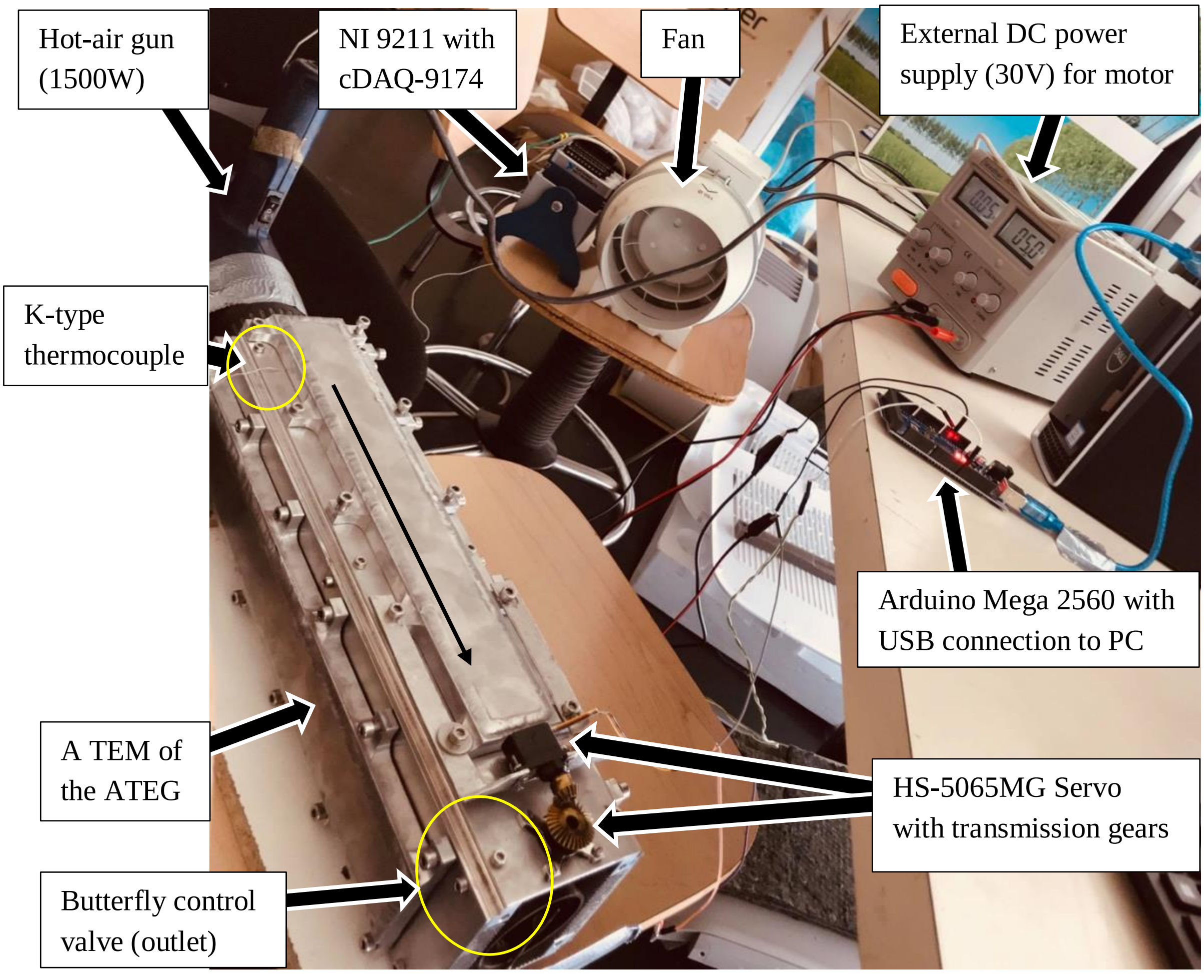



*Figure 14: Photographic view of the experimental setup*

# 4 Control program

## 4.1 Interface between LabVIEW and Arduino

The first step in the development of the control program is the interfacing of the software program LabVIEW First, the LabVIEW interface for Arduino LIFA was used. The package LIFA could easily be installed via VI Package Manager. Then the interface could be started from the Arduino IDE by uploading the LIFA Base sketch to the Arduino microcontroller. This was because LIFA as a form of interface was no longer updated by National Instruments and actually replaced by LINX.
LINX is an open source project from Digilent/LabVIEW Makerhub that provides easy interaction between LabVIEW and its VIs, and embedded systems such as Arduino, myRIO, chipKIT, etc. It also includes a wizard interface with pre-built firmware so the user does not have to start building the firmware from the source. LINX could, just as LIFA, be downloaded and installed in VI package manager. Now the pre-built firmware only needs to be uploaded to the Arduino via the firmware wizard in LabVIEW and the interface is ready to use.
At the beginning of the program, before entering the loop, the connection between LabVIEW and the Arduino must be initialized as shown in figure 15. In addition, servo motor is also initialized. To do this, you have to enter the output channel of the Arduino to which the servo is connected. In this case the digital output is 8. So it is clear that LINX's VIs simplify the initialization part enormously.

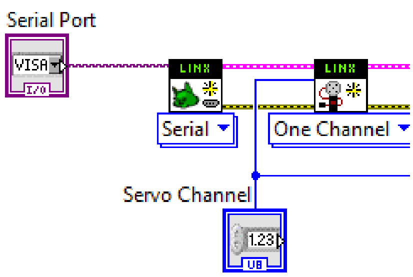



*Figure 15: Initializing connection between Arduino and LabVIEW*

## 4.2 Temperature reading of the thermocouple

### 4.2.1 MAX31855 converter module

The Arduino IDE offers a library that processes the output signal of the MAX31855 and immediately converts it into a numerical temperature. Therefore, although the control program is designed in LabVIEW, the intention was to execute the temperature reading in the Arduino IDE. This means that the temperature value that arrives in Arduino must be communicated to LabVIEW. This can be accomplished by using the VI Custom Commands of LINX. This VI gives the possibility to retrieve a function, written in Arduino, into the LabVIEW environment. This retrieval can be done to acquire data from Arduino. But it is also possible to invoke a function to send data back to Arduino. The data itself, which can be sent back and forth, is in the form of an array of bytes. The data received in LabVIEW is called the response bytes. This is also shown in figure 16, which shows a part of the code for PID control using the MAX31855. Calling a function is done by entering the number, that is assigned to the function in Arduino IDE, in the Custom Commands VI. In figure 16 this is the function 0. In appendix B the functions and the rest of the program written in Arduino IDE is shown. Since the sent data is packed in an array, it is also necessary to indicate which element of that array is needed. Here it is the first and the second element. In this program it is important to do each time the correct conversion of bytes to numeric values and vice versa.

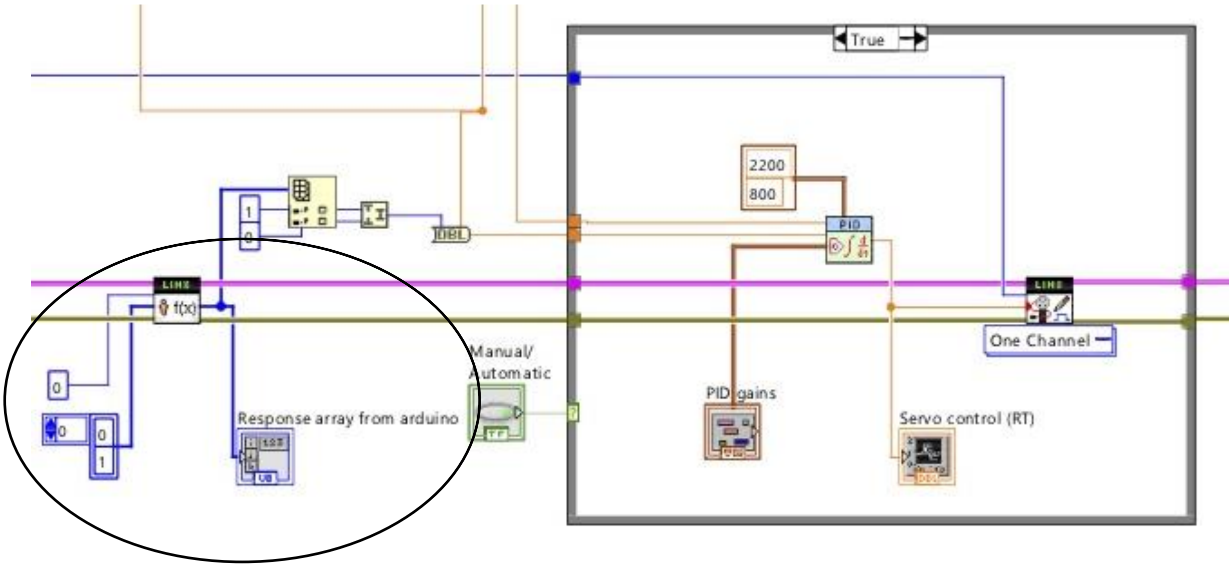



*Figure 16: Part of the PID control program based on MAX31855*

## 4.2.2 DAQ device

The NI 9211 input thermocouple module is attached to the chassis cDAQ-9174. A DAQ device can be implemented effortlessly in LabVIEW. When the thermocouple is connected and the configuration is done, the only thing that needs to be given to the DAQ assistant data module is the sample rate and the number of samples. Every DAQ device has a maximum sample rate. This is 14 samples per second for the NI 9211.

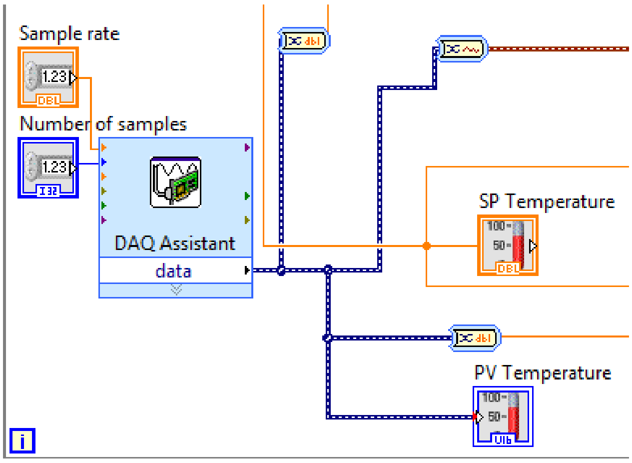



*Figure 17: Part of the PID control based on the NI 9211*

## 4.3 Control program in LabVIEW

A program in LabVIEW is separated into two windows. One window is the Front Panel and represents the user interface with all the controls and indicators. This graphical monitor panel is illustrated in section 4.4. This section deals with the other window, more specifically the Block Diagram, which contains the graphical source code.

### 4.3.1 PID controller

Figure 18 shows the block diagram of the complete control program using  PID control.

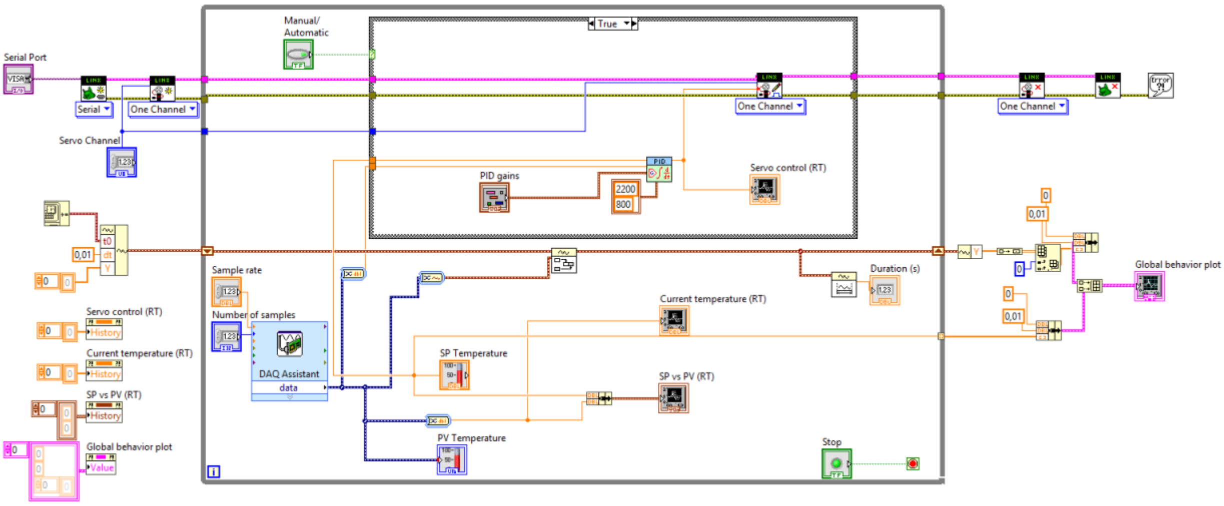



*Figure 18: Block Diagram of the PID based control program*

The program consists of two modes. An automatic mode in which the PID is operating. This PID VI has 3 inputs: the set temperature, the measured and the PID gains. The set temperature is the limit temperature of the TE modules. The measured temperature of the process should be regulated as accurate as possible to that set temperature. The PID is able to do this with the right parameters. The gains are chosen in the Front Panel, which will be discussed in the next section. The PID also has an output that is located inside a specified range. The output of the PID is the input signal for the servo, which is the “on” time of the PWM signal. Thus, the output will be located between minimum and maximum on time or resized between the minimum and maxim rotary angle of the servo. The output the PID is plotted to follow real time the behavior of the controller.
The other mode is made to actuate the valve manually by changing the “on” time of the PWM signal. However, in the Front Panel, the angle is filled in and the program will convert this with the small calculation to the “on” time.
Figure 19 displays both modes. The mode can be changed by pressing a Boolean button in the Front Panel.

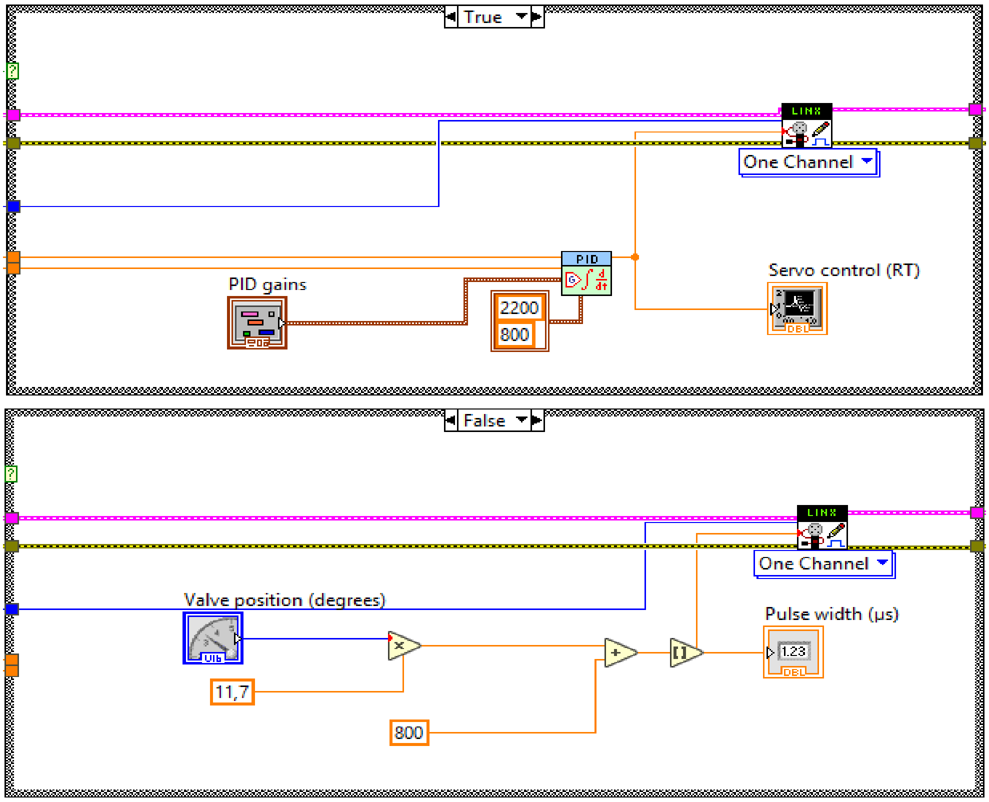



*Figure 19: (Upper) Automatic mode; (Lower) Manual mode in a PID control program*

## 4.3.2 Fuzzy logic controller

In figure 20, it can be seen that the PID controller in figure 19, is now replaced by the fuzzy logic controller. Before the beginning of the while loop, the fuzzy system file will be downloaded from the pc into the Block Diagram of LabVIEW. There a few basic steps in the development of a fuzzy system. These steps are based on the theory of fuzzy logic, that is explained in section 2.2.2.3.

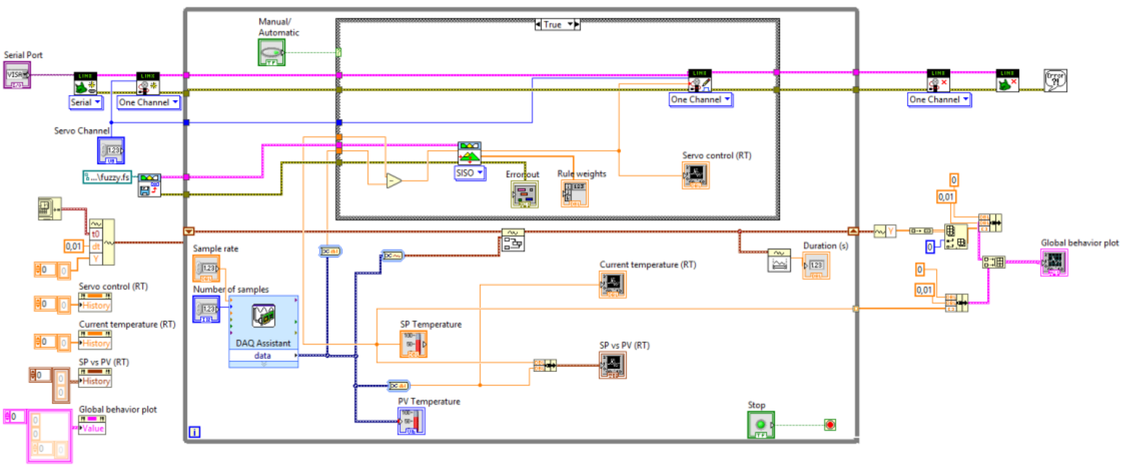



*Figure 20: Block Diagram of the Fuzzy based control program*

## Inputs and outputs

In the first step the linguistic variables or input/ output variables of the temperature control system will be determined. In this project there is only one input and one output. The relative temperature is taken instead of the absolute, because this creates a system that allows a variable set point.

*Input:* Temperature error
*Output:* "On" time of PWM for servo

## Subsets of inputs

Both in- and output are subdivided into five linguistic terms.

*Input:* Large Negative (LN), Small Negative (SN), Zero (Z), Small Positive (SP), Large Positive (LP)

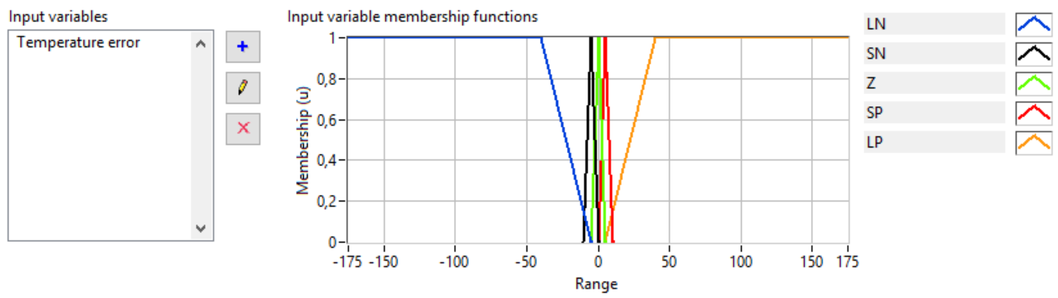



*Figure 21: Membership function for the temperature error*

These values are picked based on intuition, just as the number of membership functions. It is important that for one input value there is minimum one and maximum two membership functions that yield a corresponding value.

*Output:* Fully Closed (FCL), Half Closed (HCL), Close/Open (CL/OP), Half Open (HOP), Fully Open (FOP)

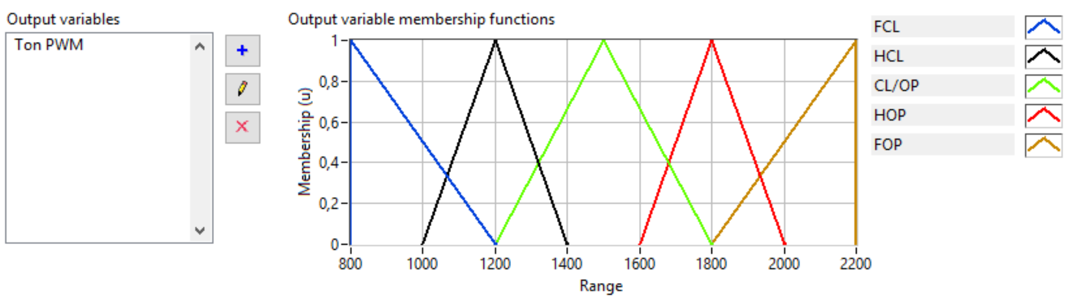



*Figure 22: Membership function for the fuzzy output*

Finally, the rule base is built. The implemented rules are really simple. In this way it will be easier to understand the response. As the fuzzy experience is rather low, this is a good way to start. The used defuzzification method to convert the degrees of membership of the output linguistic variable within its linguistic terms, is Center of Area. Something that is really important to be alert for when testing, is that the output of a defuzzification method always needs to be a continuous signal. A small change at the input can never cause an abrupt change at the output.

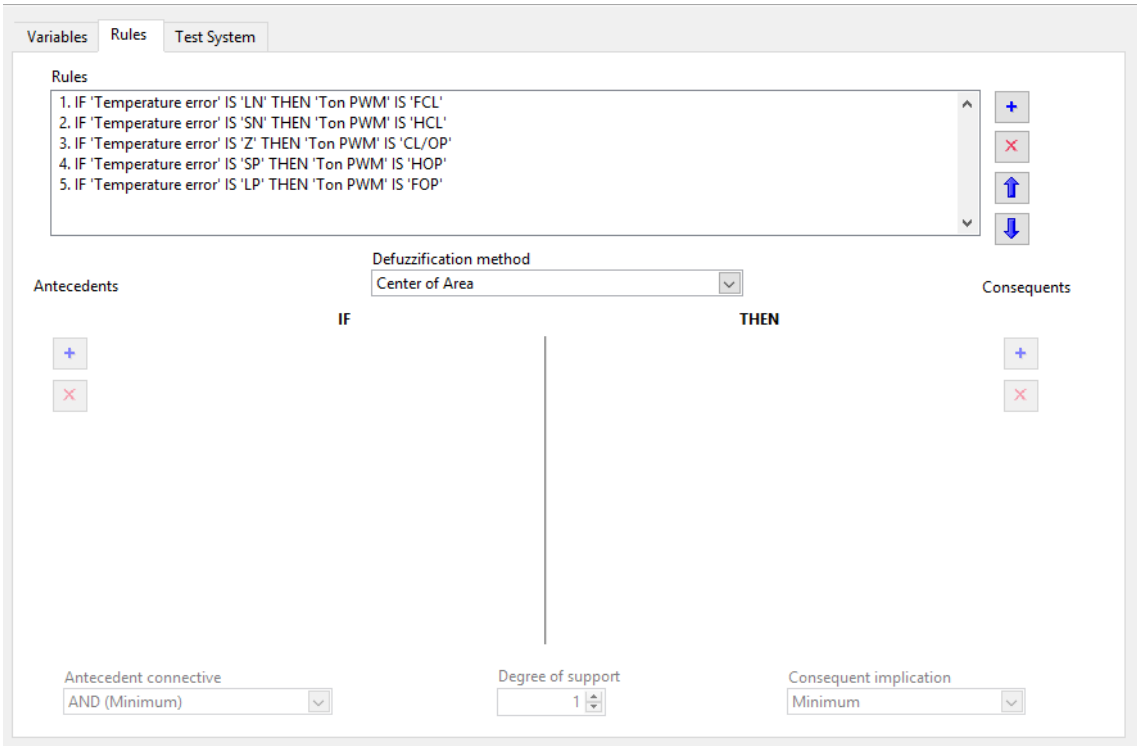



*Figure 23: Rules of a fuzzy temperature control system*

## 4.4 Graphical monitor panel

### PID control

Another important part of the temperature control system is to provide an easy to use and accessible GUI. Figure 24 presents the graphical control and monitor panel for the user. The panel can be divided into three parts. The left part provides more the settings, such as the servo channel, the desired temperature or sample rate of the DAQ. The middle part consists of two modes (automatic and manual). The manual section monitors the behavior of the PID and allows to adjust the PID gains. The right part of the Front Panel has some black graphs These graphs display the process temperature in the ATEG at real-time and plot it against the desired value. The large white graph is used to demonstrate the system’s output response. By reference to this output response, the PID gains can be adjusted for a temperature control behavior.

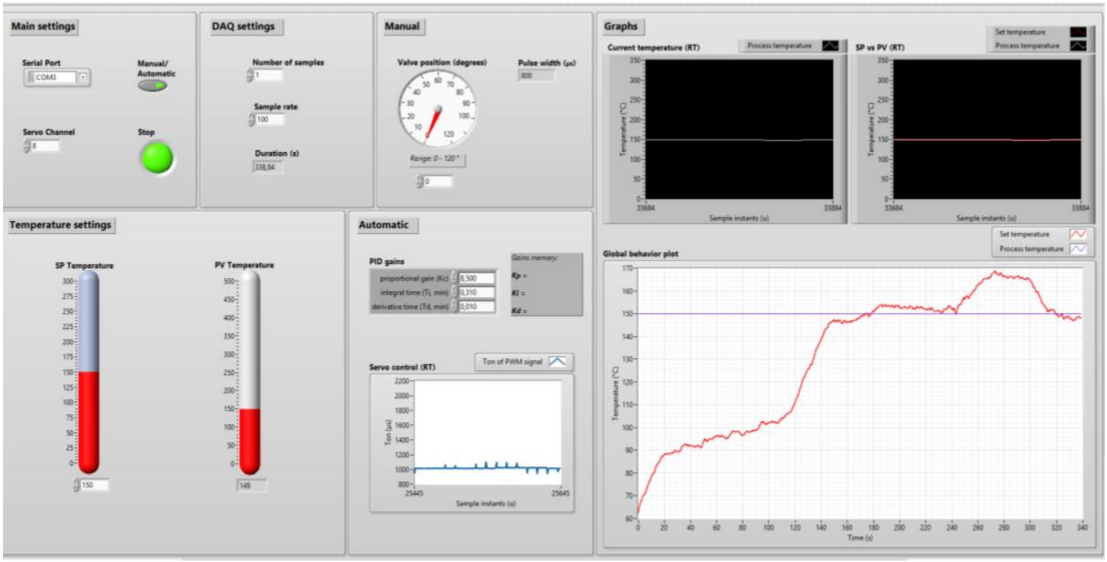



*Figure 24: Front panel of the PID control system*

## Fuzzy logic control

The GUI of the fuzzy control system is nearly the same as the one with PID, except for the automatic window. Instead of gains, now, the weight of the five different implemented rules are displayed.

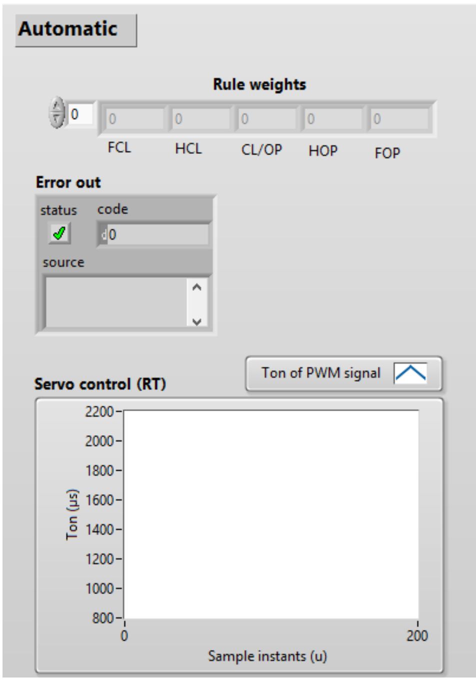



*Figure 25: Small automatic-mode window of the Front Panel for a Fuzzy control system*

# 5 Results

## 5.1 PID based temperature control

**Trial and error method**

The manual tuning method which was discussed in the theoretical part, will not be used, at least the results not. The results of this method were not really meaningful. That's why a classic trial and error method is applied. This is a time consuming method. However, it helps to learn the parameters, the system and the relationship between them better. The tests are executed with the experimental setup of section 3. The hot-air pistol is set up at mode 1. Mode 2 will be used to add a disturbance or to adjust the set temperature. After numerous trials and errors the best achieved result, was the one with the following parameters:

- Kp = 8,5
- Ki = 0,3
- Kd = 0,015

Figure 26 shows the response of temperature control system with these settings for the PID. From the graph, the performance parameters can be found. Table summarizes the results. The results are really satisfying. Despite the time, the response is really accurate and stable.

*Table 3: Performance of PID expressed with time response parameters*

| $t_r$(s) | %$M_p$ | $t_s$(s) | SSE |
|---|---|---|---|
| 80 | 0 | 85 | 0 |

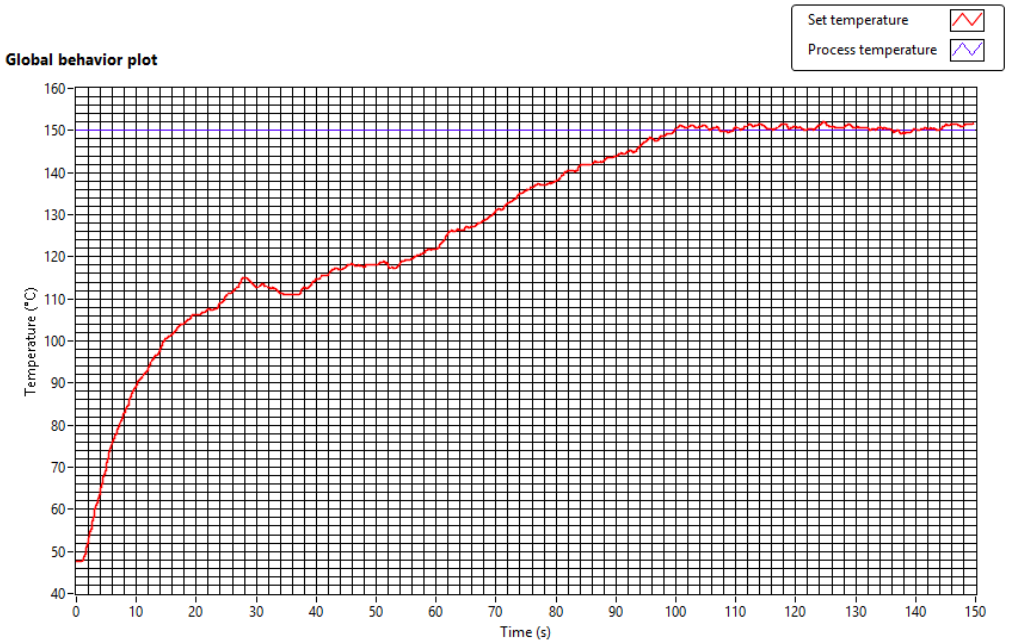



*Figure 26: Output response of PID control system with set temperature 150°C*

Now the mode of the hot-air pistol will be changed two times. After more or less 135 s, the mode will be changed the first time to stand two. As can be seen in figure 27, temperature increases after the first switch up to 168°C and after 240 s it reaches again the target value. This means that the system needed 95 s before the disturbance could be moderated. This is too much, because a long thermal overload will damage the TE modules. At 285 s the mode is switched back to 1. The reaction to this disturbance is even slower than the previous one. Just like the theory already explains, the PID has difficulties to overcome the inertia or non-linearity of the ATEG. This was already noticeable in the previous case, where the system also reacted rather slow.

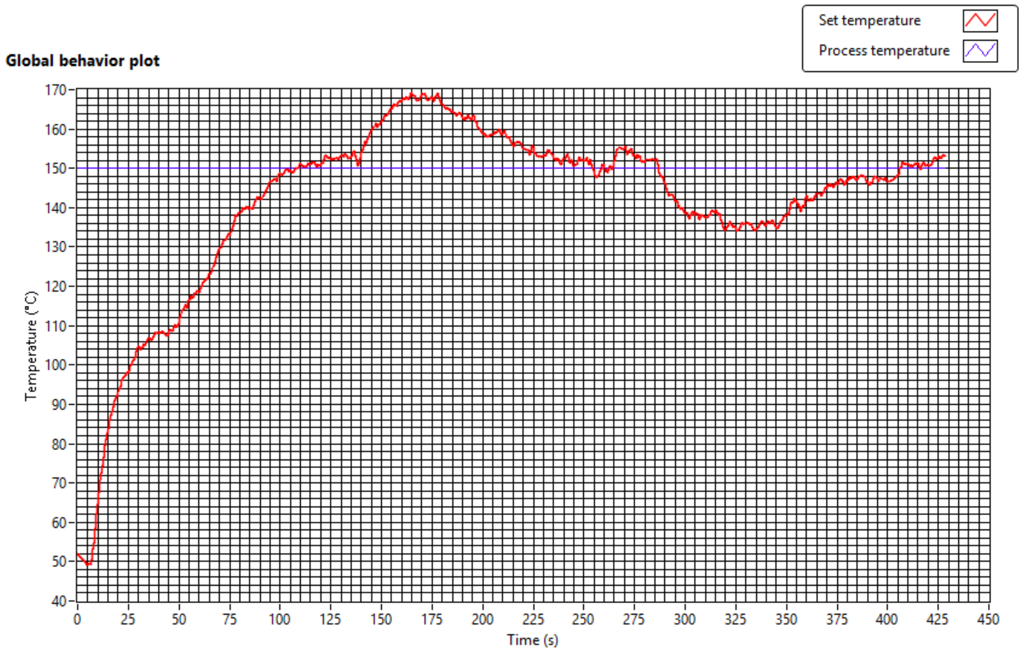



*Figure 27: Output response of a PID control system subjected to disturbances*

In this case the setpoint is changed during the process. As the PID controller is not an adaptive controller, it was predictable that it would lose performance here by changing the process.

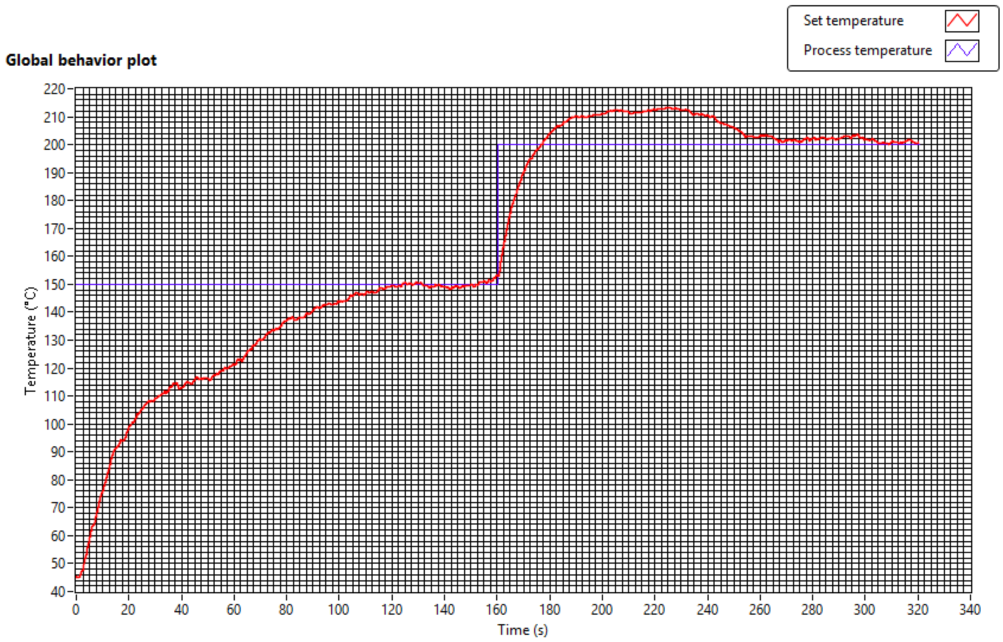



*Figure 28: Output response of a PID control system with changing setpoint*

Also here, the PID will not perform as perfectly anymore as with the setpoint of 150°C. The response is however still stable, but lost a bit of accuracy with a current SSE of 7%.

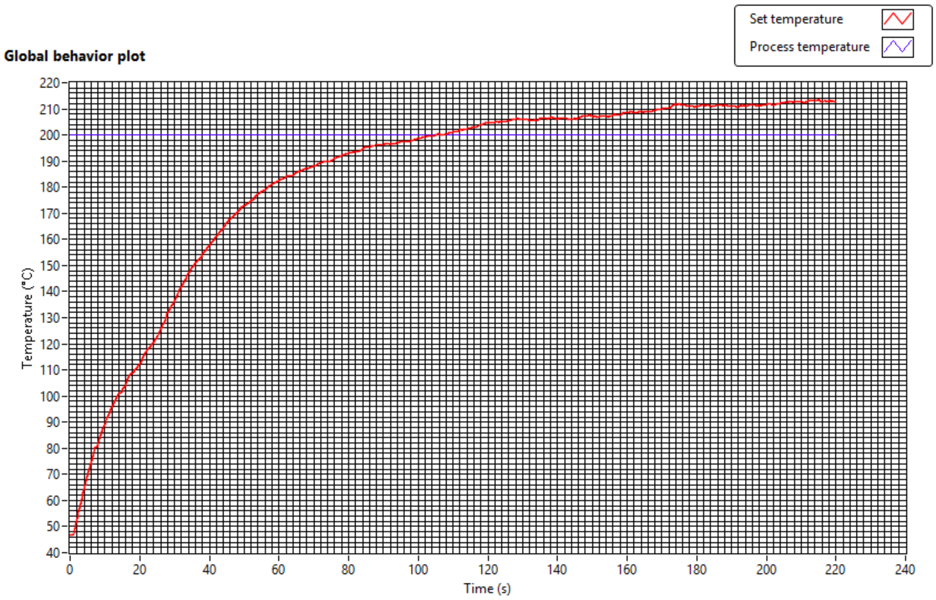



*Figure 29: Output response of a PID control system with set temperature 200°C*

**Ziegler-Nichols method**

The first rule of the Ziegler-Nichols method (Z-N) prescribes that Kp should be increased until a periodical oscillation occurs at the output. The Kp was around 170, where this oscillating behavior took place. That behavior is shown in figure 30 below. The period of this oscillation is 6 s. The results for the gains can now be calculated.

*Table 4: Results of Ziegler-Nichols method for a PID control system*

| | $K_p$ | $K_i$ | $K_d$ |
|---|---|---|---|
| **PID controller** | $0.6Kp_u = 102$ | $T_u/2 = 3$ | $T_u/8 = 0,75$ |

If we use these settings the output response will be like figure 31.

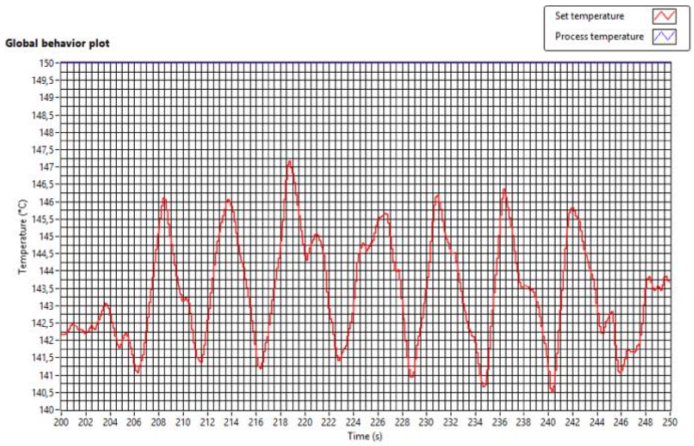



*Figure 30: Process of increasing Kp until oscillating behavior (Ziegler-Nichols method)*

The response of the system with a PID tuned by Ziegler-Nichols method creates a heavily overdamped system and a quite turbulent behavior because of the D term value. It is a logic result, since Z-N uses theoretical prescribed formulas, and according to theory a temperature control system with inertia based on PID should be overdamped. However, the overdamping is too much.

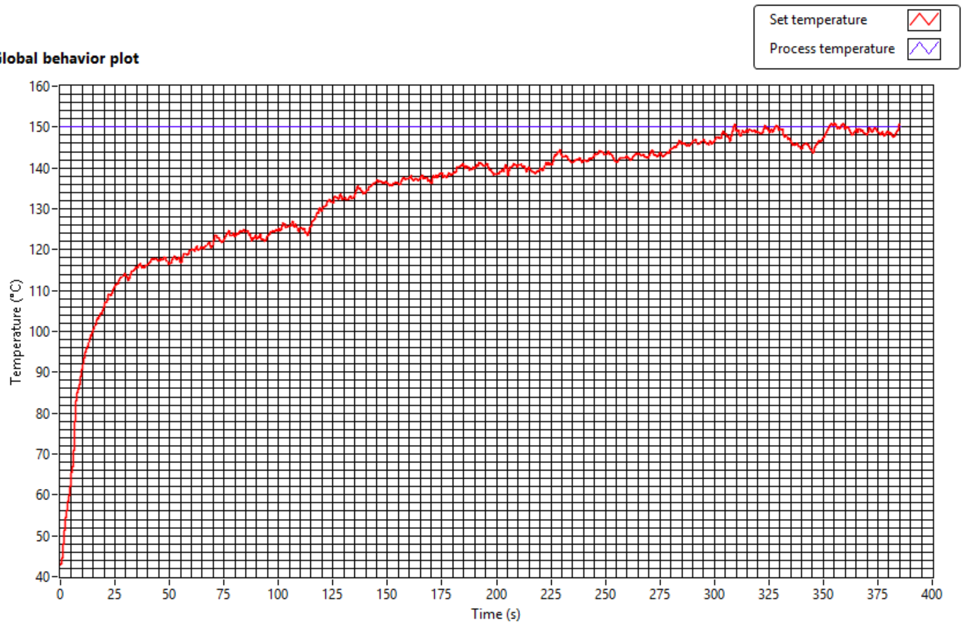



*Figure 31: Output response of a PID control system tuned by parameters of Z-N method*

So because the response was that slow, the D term was modified from 0.75 to 0.2. Figure 32 gives the corresponding response to this new D term value. As we can see, the response is already a lot faster.

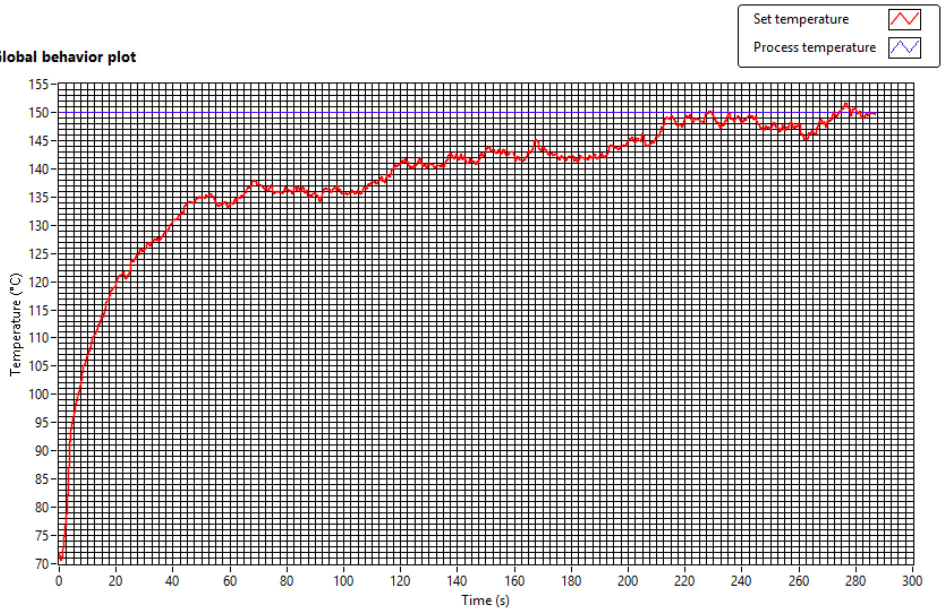



*Figure 32: Output response of a PID control system with adjusted parameters of Z-N method*

## 5.2 Fuzzy Logic based temperature control

The PID controller is now replaced by the FLC. The fuzzy system, discussed in chapter 4, was really simple to establish. It was even so easy that the expectations about the results were really low. However, the result is surprisingly good. It is definitely not perfect, with a small SSE of approximately 5%. But the curve is really smooth and stable. This kind of smoothness is not even achieved with PID control. Considering, that the time that was put in the trial and error tuning of the PID, is almost 10 times larger than time used to build the fuzzy system. However further adjustments of the fuzzy system did not give really much of an improvement. A reason for this is, that in order to eliminate that remaining SSE, an integral term will need to be introduced.

*Table 5: Performance of fuzzy control expressed with time response parameters*

| $t_r$(s) | %$M_p$ | $t_s$(s) | SSE |
|---|---|---|---|
| 65 | 6,25 | 45 | 8 |

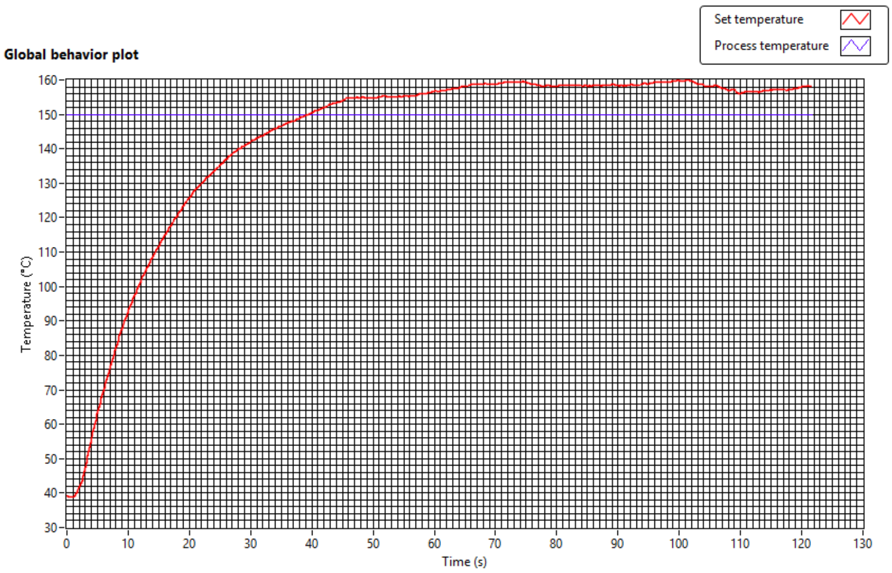



*Figure 33: Output response of a fuzzy control system with set temperature 150°C*

Figure 34 shows the addition of a disturbance to the fuzzy system around 35 s. The result is a curve that migrates further away from the desired value. A reason for this can be found in the membership functions of this system and the input variables. When the disturbance is added, the system will try to react to this. As, there is only one input, the temperature error, this fuzzy system is actually a fuzzy-P. With the proportional term similar to that one of a PID. When the system comes in the situation where all weights are zero, except the large negative (LN) linguistic term of the input linguistic variable like in this example. And the weight of that LN term is changing in such a way that temperature will stop increasing and starts decreasing. Eventually, the moment occurs that the temperature stops increasing. But then the weight also stops changing, so the temperature just stays constant. This could be solved by adding an I-action.

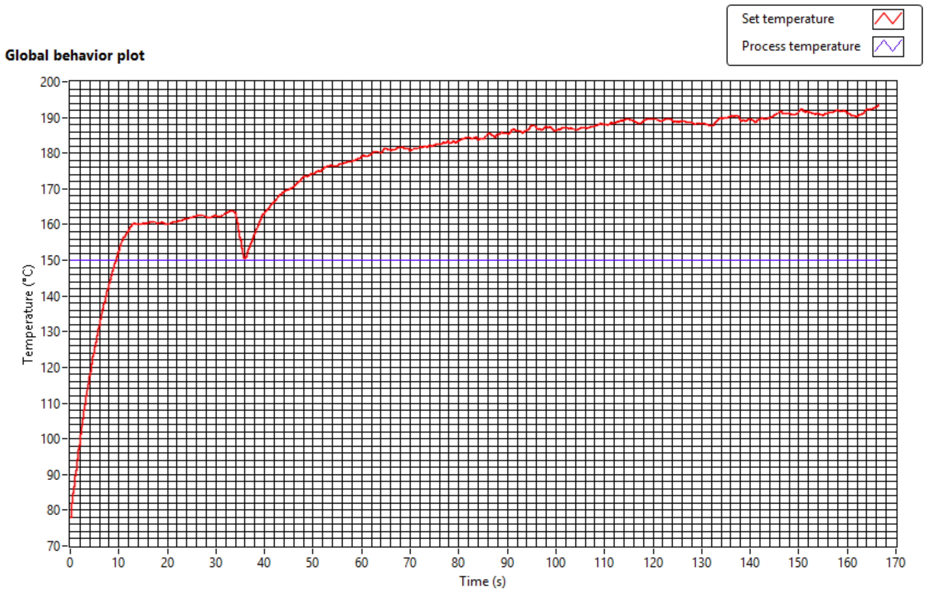
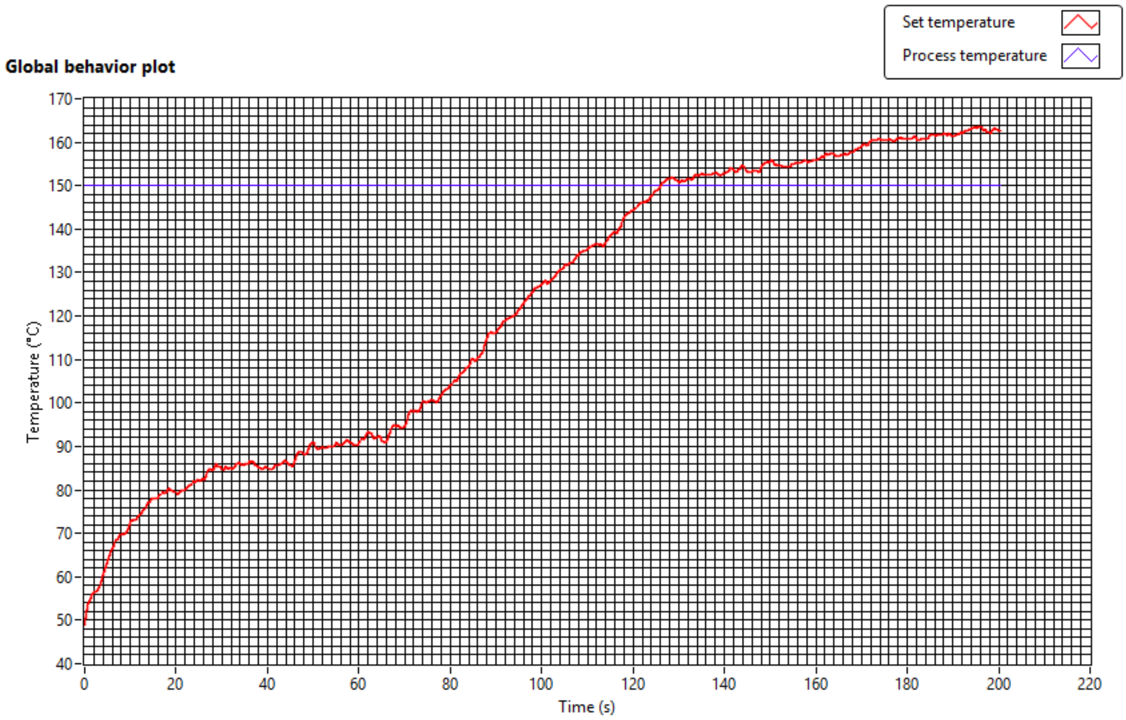



*Figure 34: Output response of a fuzzy control system subjected to a disturbance*

## 5.3 Discussion

The time-consuming trial and error method eventually gave some good results for this temperature control system. Although, the PID controlled system always took quite some time. This is because an ATEG system is a slow changing system, at least where temperature is measured. In addition, it is also not perfectly linear. That is the reason why the PID should be tuned in way that response is really slow, otherwise the performance would be bad. This can also be shown well in the following figure 35.

Here, the same settings are used as in figure 26 with the trial and error method, except for the Kp. This one is raised to 9.5, instead of 8.5, to get a faster response. The result is a behavior that seems to keep increasing and going away from the setpoint. Thus, there is no other way than to tune the PID so that the system will be really slow and bad performing behavior is avoided.




*Figurer 35: Output response of a PID control system with gains of figure 26, except Kp = 9.5*

Ziegler-Nichols method represented the theoretical solution, namely an overdamped system. But it was too much damped and thus not the best solution. However, the Ziegler-Nichols method seems like a good point to start from, but not as a final result. The small changes made in the D-action led already to better results. So it is probably better to finetune with trial and error in this case by using the experience and knowledge about control theory and the system. The PID was in these results more accurate than the FLC. Although, FLC beat PID ahead in terms of speed. It is also remarkable that FLC definitely still have some margin as a controller for the ATEG system, considering the lack now of the I and D actions. In addition, this way of controlling is easy and comprehensible. The natural language where it is based on, really helps you to see understand reactions of the system.

# 6 Conclusion

The purpose of this master's thesis was to design a control system that keeps the temperature on the hot heat exchanger side of an ATEG almost constant. In that way, on the one hand, overheating of the TE modules could be prevented and, on the other hand, the highest possible electrical power at the output could be achieved. The control system consists of an exhaust gas controlling valve, driven by an HS-5065MG motor with position control. The temperature in the ATEG depends on the number of exhaust gases that the butterfly valve passes through the ATEG.

First, we searched for suitable material. As an electric rotary actuator, the torque motor was the first choice due to its combination of compactness, robustness, and speed. However, eventually, the HS-5065MG was used for this lab setting. But the torque engine could certainly be a potential improvement for the future.
The MAX31855 can also be an improvement on the DAQ device. Because of its small size, it is very easy to implement in almost every system.

After that, a literature study was done on the basis of a thermoelectric generator, but especially on control algorithms. Fuzzy Logic and PID seemed to be the most suitable controllers. PID is known for the rather mathematical approach with a single formula. While fuzzy logic control works according to rules and natural language that are at the same level as our human thinking. Both have their own strengths. So, we need to consider which advantages are most useful for this temperature control system and which disadvantages we surely want to avoid..

To verify this, one control program was created based on a PID and another based on fuzzy logic. The fine-tuning of the PID was first done by trial and error, and second by Ziegler-Nichols. It is best to start with the Ziegler-Nichols method. This method immediately indicates the correct direction to follow, concerning the tuning. After that, it is still recommended in this system to continue tuning manually through trial and error. For the fuzzy control system, a SISO system was used. The only input is the temperature error and the output is the control signal for the motor. Both the input linguistic variables and the output linguistic variables consist of 5 subsets. This means 5 membership functions. There are also 5 simple rules.

The results show that PID excelled the most in the completed simulation tests. With no overshoot and a zero SSE, PID seemed definitely to be the most accurate one. However, to obtain these good values, the PID needed to be tuned in such a way that the system was overdamped. As a result the response settle time was in the order of a magnitude of 90 s, which can be considered as too slow. Fuzzy delivered a much faster response with a settle time of 45 s, but could not achieve the desired value. This is also a bit logical because there has been more focus on the adjustment of the PID. If the results were to be viewed in relation to the time spent on each control system, then fuzzy logic would definitely be number one. Fuzzy logic allows to adjust the membership functions in a way that the inertia and non-linearity of the ATEG system can be moderated. This is not possible with the fixed parameters and formula of the PID.

There are still a lot of improvements that can be made to lift this technique of temperature control to a higher level. In terms of material, there already made some suggestions for the future.
For the control algorithm, fuzzy certainly has the potential to reach the same or even a better performance than PID for this application, if it is defined more precise. For example the number of linguistic input variables can be increased. Also a D-action and I-action can still be added. In that way the best of both two worlds are combined in a Fuzzy PID. This method of controlling could have a lot of potential in this domain where the system is too slow for a single PID to perform optimally and accurate control is still required to keep at that limit temperature..

# Bibliography


[1] GREFEMA, 2015. [Online]. Available: www.grefema.udg.edu/en/research/linies-de-recerca/. [Accessed 12 Februari 2019].

[2] D. Champier, "Thermoelectric generators: A review of applications," *Energy Conversion and Management*, no. 140, pp. 167-181, 2017.

[3] P. Aranguren, "Automotive Waste Heat Recovery by Thermoelectric Generator Technology," in *Bringing Thermoelectricity into Reality*, London, IntechOpen, 2018.

[4] National Instruments, "LabVIEW," [Online]. Available: www.ni.com/en-gb/shop/labview.html. [Accessed 20 Maart 2019].

[5] National Instruments, in *LabVIEW Core 1 Participant Guide*, Texas, 2014.

[6] G. Mishra and S. Kumar Sharma, "A review of automotive thermoelectric generator," *International Research Journal of Engineering and Technology (IRJET)*, vol. 4, no. 3, pp. 2787-2798, 2017.

[7] N. Aimable, in *Design, modeling, and fabrication of thermoelectric generator for waste heat recovery in local process industry [thesis]*, Agder: Faculty of Engineering and Science University of Agder, 2017.

[8] A. Fonts Santana, in *Characterization of a Peltier module working as a thermoelectric generator [thesis]*, Barcelona: Faculty of Physics University of Barcelona, 2015.

[9] M. Comamala, I. Ruiz Cózar, A. Massaguer, E. Massaguer and T. Pujol, "Effects of Design Parameters on Fuel Economy and Output Power in an Automotive Thermoelectric Generator," *Energies*, vol. 11, no. 12, 2018.

[10] Y. D. Deng, X. Liu, S. Chen and Q. Tong, "Thermal Optimization of the Heat Exchanger in an Automotive Exhaust-Based Thermoelectric Generator," *Journal of Electronic Materials*, vol. 42, no. 7, pp. 1634-1640, 2012.

[11] B. Beltrán-Pitarch, J. Prado-Gonjal, A. V. Powell, P. Ziolkowski en J. García-Cañadas, „Thermal conductivity, electrical resistivity, and dimensionless figure of merit (ZT) determination of thermoelectric materials by impedance spectroscopy up to 250 °C," *Journal of Applied Physics*, vol. 124, nr. 2, 2018.

[12] Febeg, "PID Theory explained," National Instruments, [Online]. Available: http://www.ni.com/en-gb/innovations/white-papers/06/pid-theory-explained.html. [Accessed 12 April 2019].

[13] A. Ryniecki, A. A. Pilarska and J. Wawrzyniak, "Basics of process: the on-off control system," *Przemysł spożywczy,* November 2015.

[14] B. Mosayebi, „On/off control," Aquarius Technologies, [Online]. Available: https://aquariustech.com.au/wp-content/uploads/bsk-pdf-manager/On_Off_Controller_9.pdf. [Geopend 20 April 2019].

[15] J. Baeten, in *Digitale, Niet-lineaire en Fuzzy Regeltechniek: Deel 2 [course]*, Diepenbeek: Faculteit Industriële Ingenieurswetenschappen UHasselt & KU Leuven, 2006.

[16] K. Åström and T. Hägglund, "The future of PID control," *Control Engineering Practice,* vol. 9, no. 11, pp. 1163-1175 , 2001.

[17] J. Baeten, in *Analoge regeltechniek [course]*, Diepenbeek: Faculteit Industrïele Ingenieurswetenschappen, 2006.

[18] K. Arab Tehrani and A. Mpanda, in *Introduction to PID Controllers - Theory, Tuning and Application to Frontier Areas*, InTech, 2012.

[19] J. Baeten, in *Digitale, Niet-lineaire en Fuzzy Regeltechniek: Deel 3 [course]*, Diepenbeek: Faculteit Industriële Ingenieurswetenschappen UHasselt & KU Leuven, 2006.

[20] F. Dernoncourt, in *Introduction to fuzzy logic [course]*, Cambridge: Massachusetts institute of technology, 2013.

[21] "What Is Fuzzy Logic?," MathWorks, [Online]. Available: https://nl.mathworks.com/help/fuzzy/what-is-fuzzy-logic.html. [Accessed 18 April 2019].

[22] "LabVIEW 2012 PID and Fuzzy Logic Toolkit Help," National Instruments, 2012. [Online]. Available: http://zone.ni.com/reference/en-XX/help/370401J-01/. [Accessed 18 April 2019].

[23] Arduino, "Arduino MEGA 2560," [Online]. Available: http://www.mantech.co.za/datasheets/products/A000047.pdf. [Accessed 3 March 2019].

[24] G. K. McMillan, "Temperature measurement, control key to plant performance," *InTech Magazine,* 2010.

[25] Maxim Integrated, "MAX31855," 2015. [Online]. Available: https://datasheets.maximintegrated.com/en/ds/MAX31855.pdf. [Accessed 18 April 2019].

[26] National Instruments, "NI 9211 Datasheet," 2015. [Online]. Available: http://www.ni.com/pdf/manuals/373466a_02.pdf. [Accessed 6 June 2019].

[27] National Instruments, "NI cDAQ-9174," 2013. [Online]. Available: http://www.ni.com/pdf/manuals/374045a.pdf. [Accessed 6 June 2019].

[28] M.-O. André, C. Gassmann and P. Reghenzi, "Torque motors as actuators in intake and exhaust system," *MTZ worldwide,* vol. 67, no. 6, pp. 21-23, 2006.

[29] Sonceboz, "Torque Motor," [Online]. Available: https://www.sonceboz.com/torque-motor. [Accessed 9 April 2019].

[30] Q. Wang, F. Yang, Q. Yang, J. Chen and H. Guan, "Experimental analysis of new high-speed powerful digital solenoid valves," *Energy Conversion and Management* , vol. 52, no. 5, pp. 2309-2313 , 2011.

# Appendices

Annex A: The torque motor of Sonceboz
Annex B: Temperature reading in Arduino IDE

# Annex A: The torque motor of Sonceboz

SONCEBOZ

## ▶ Torque Motor

**4247**


- **Working angle ............ 75° max**
- **Positioning time ......... < 100 ms**
- **Weight ………………..... 330 g**
- **Peak torque @ 13.5V .. 650 mNm**

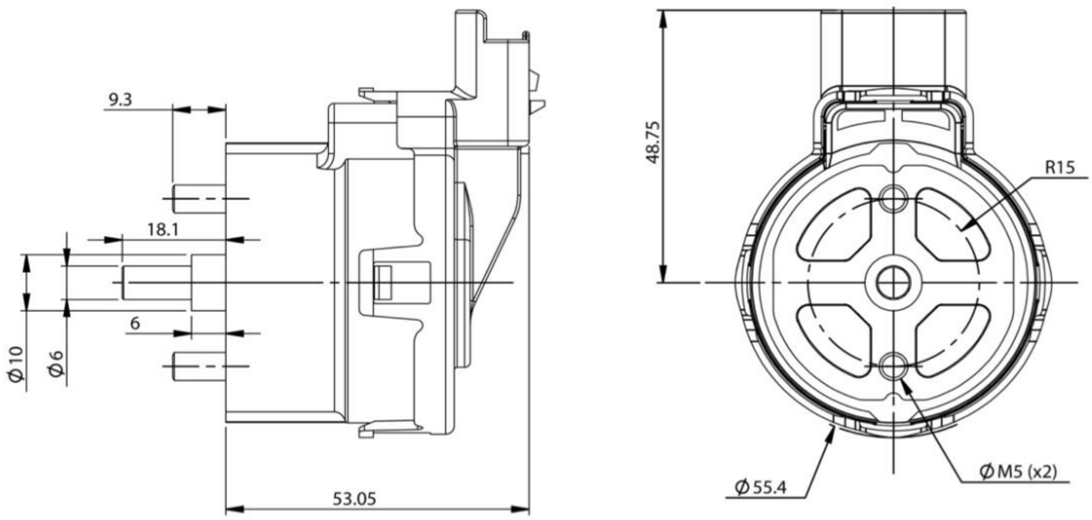
**This actuator is a contactless direct drive device and exhibits both high torque and dynamics.**

### ▶ Technical data

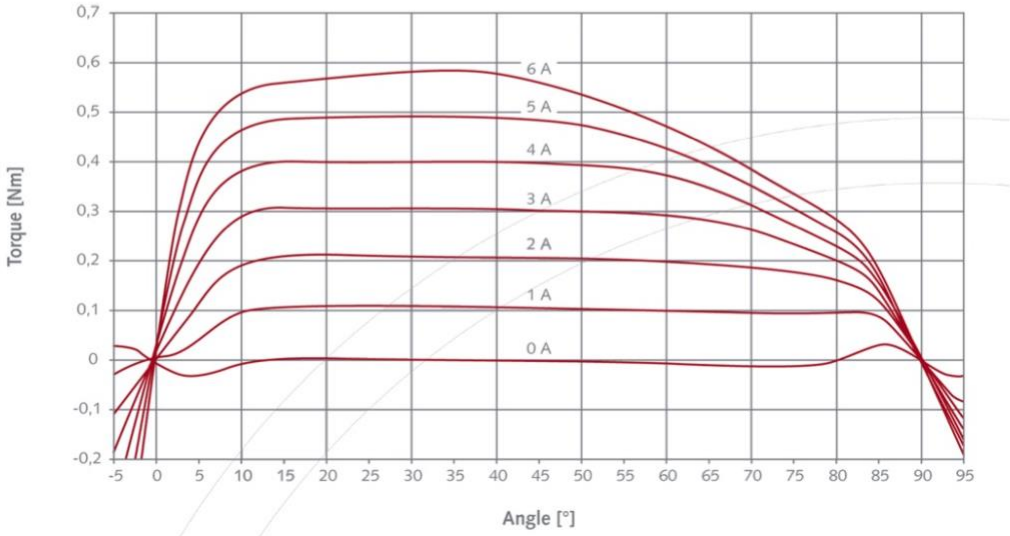
| Resistance [Ω] | Inductance [mH] | Torque constant [mNm/A] | Motor constant [mNm/√W] | Detent torque [mNm] | Sensor signal |
|---|---|---|---|---|---|
| 2 | 14 | 100 | 70 | 38 | Programmable:<br>- Analog: 0.5 to 4.5 V<br>- Digital: PWM / SENT |

### ▶ Dimensions

*Drawing not to scale. All dimensions in mm.*



## ► Dynamic characteristics




## ►Main Features

Compact, robust, contacless
Single phase DC operation (H-bridge compatible)
Nominal torque proportional to current, bi-directional
High peak torque
Programmable contactless position sensor
Integration of fail-safe optional
Waterproof design optional

## ►Operating conditions

**Qualified for on-engine applications:**

Ambient temperature: -40 … > +160° C
Vibrations: 40 g

# Annex B: Temperature reading in Arduino IDE

```
//Include All Peripheral Libraries Used By LINX
#include <SPI.h>
#include <Wire.h>
#include <EEPROM.h>
#include <Servo.h>
#include"Adafruit_MAX31855.h"

//Include Device Sepcific Header From Sketch>>Import Library (In This Case
LinxArduinoMega2560.h)
//Also Include Desired LINX Listener From Sketch>>Import Library (In This Case
LinxSerialListener.h)
#include<LinxArduinoMega2560.h>
#include<LinxSerialListener.h>

#define MAXDO   12
#define MAXCS   10
#define MAXCLK  13

int temperature();
volatile int tempValue = 0;
// initialize the Thermocouple
Adafruit_MAX31855 thermocouple(MAXCLK, MAXCS, MAXDO);

//Create A Pointer To The LINX Device Object We Instantiate In Setup()
LinxArduinoMega2560* LinxDevice;

//Initialize LINX Device And Listener
void setup()
{
  //Instantiate The LINX Device
  LinxDevice = new LinxArduinoMega2560();

  //The LINX Listener Is Pre Instantiated, Call Start And Pass A Pointer To The
LINX Device And The UART Channel To Listen On
```

```
 LinxSerialConnection.Start(LinxDevice, 0);

 attachInterrupt(digitalPinToInterrupt(1),interruptTemperatureMeasure,CHANGE);
 LinxSerialConnection.AttachCustomCommand(0, temperature);
}

void loop()
{
  //Listen For New Packets From LabVIEW
 LinxSerialConnection.CheckForCommands();
  //Functions called in LabView
}

voidinterruptTemperatureMeasure(){
  // tempValue = thermocouple.readCelsius();
      tempValue = 25;
  // delay(500); to mute little temperature fluctuations
}

int temperature(unsigned char numInputBytes, unsigned char* input, unsigned char*
numResponseBytes, unsigned char* response){
    tempValue = thermocouple.readCelsius();
    //tempValue = 25;
    *numResponseBytes = 2;
    response[0] = (tempValue & 0xFF);
    response[1] = (tempValue >> 8) ;

  return 0;
 }
```