POLITECNICO DI TORINO

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATIONS

MASTER OF MECHATRONIC ENGINEERING

Experimental characterization and control of a rotary motor driven by shape memory alloy (SMA)

Supervisor: Prof. Terenziano Raparelli
Supervisor: Prof.ssa Daniela Maffiodo

Thesis of ABDELHALIM AHMED
Matr. 256842

Academic year 2019/2020

Contents

Table of figures

List of Tables

Introduction

The purpose of this project is to control a motor that use a certain material with a specific characteristics or in other words a special ability and that is the memory shape alloy (SMA)as the base mechanism of this rotary motor to get a specific torque and speed.

This rotary motor is alimented by current to activate the special ability of this SMA material.

The prototype that it will be used in this experimental project was designed by (Luca Amicucci) in 2017 at Politecnico di Torino university which it will be introduced later with all its specifications and mechanical characteristics as they will influence the calculations and the experiments and control coding.

Memory Shape Alloy (SMA)

Introduction

The Swedish metallurgist Olander (1932) discovered the shape memory effect, or a closely related phenomenon, now called pseudo-elasticity. He discovered the unique rubber-like behaviour of gold-cadmium alloy at a meeting of the Swedish Metallurgical Society. May 27, 1932. Benedicks (1940) and Bystrom and Almin (1947) subsequently studied the "Olander alloy".

In (1951), the rubber-like effect in the martensite product phase was explained. They made important observations on the single-interface transformation and the two-way or reversible shape memory effect.

They also report that if they are cooled by phase change under stress, the cubic straight rod will be permanently deformed. When heated to the cubic phase, the permanent deformation disappears again. This observation is strictly a reverse shape memory under stress, but it is close to the first observation of the true shape memory effect.

In 1965, the first of a series of metal alloys of nickel and titanium was produced by the Naval Ordnance Laboratory. These alloys are called Nitinol, for Nickel Titanium Naval Ordnance Laboratory. Many of the alloys have a rather remarkable property: they remember their shape. This “smart” property is the result of the substance’s ability to undergo a phase change – a kind of atomic ballet in which atoms in the solid subtly shift their positions in response to a stimulus like a change in temperature or application of mechanical stress.

So we can say that **shape memory alloy** is an alloy that can be deformed when it cools but returns to its pre-deformed ("remembered") shape when heated. It may also be called memory metal, memory alloy, smart metal, smart alloy, or muscle wire.

SMAs can be created by alloying zinc, copper, gold and iron but the two most prevalent shape-memory alloys are copper-aluminium-nickel and nickel-titanium (NiTi) , NiTi-based SMAs are preferable for most applications due to their stability and practicability.[1]

The shape memory effect allows these materials to change their shape continuously and reversibly, only with the change in temperature: they basically change from a stable martensitic configuration at low temperature, to a stable austenitic one at high temperature and vice versa.

SMA functions and characteristics

Taking into account the (NiTi) because it is the material used in our project ,NiTi alloys change from austenite to martensite upon cooling; M_f is the temperature at which the transition to martensite completes upon cooling. Accordingly, during heating A_s and A_f are the temperatures at which the transformation from martensite to austenite starts and finishes. Repeated use of the shape-memory effect may lead to a shift of the characteristic transformation temperatures (this effect is known as functional fatigue, as it is closely related with a change of microstructural and functional properties of the material) [2]. The maximum temperature at which SMAs can no longer be stress induced is called M_d, where the SMAs are permanently deformed [3]. Temperature and stress is what is depending for the transition from martensite to austenite phase (it does not depend on time).

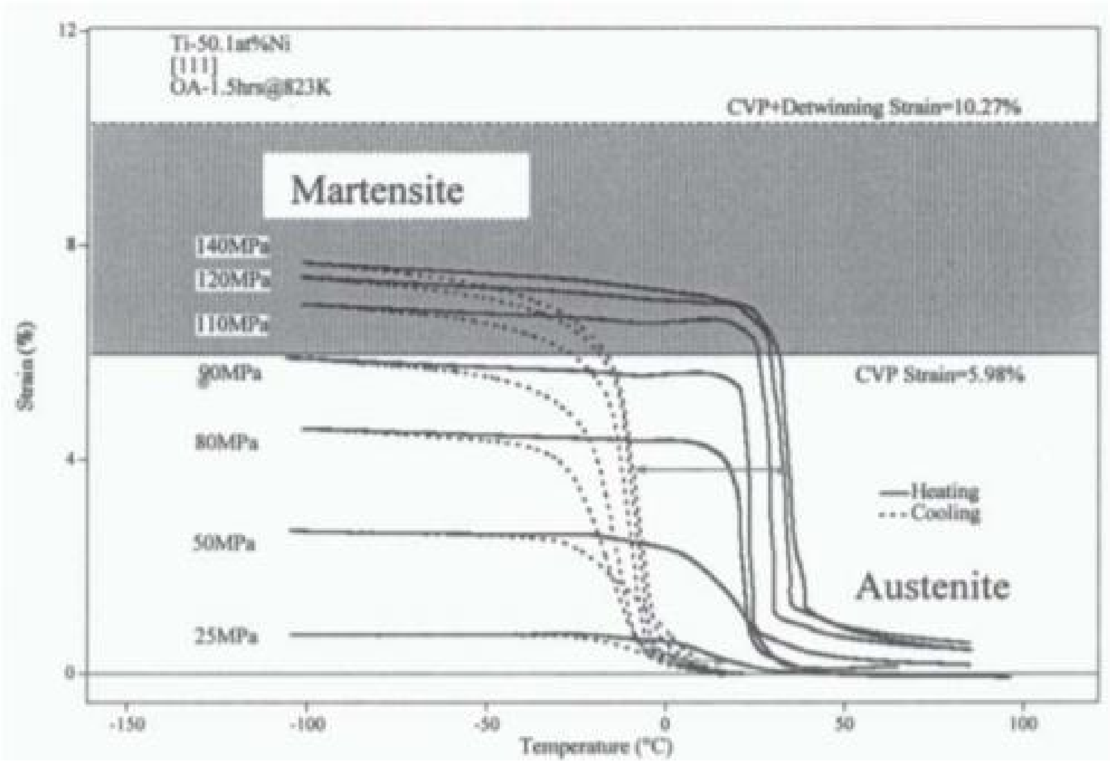

Figure 1 Strain temperature response of 50.1%Ni NiTi alloy under constant stresses. the thermal hysteresis is not constant and increases with increasing stress levels. Arrows indicate the thermal hysteresis measured at low stress and high stress respectively [4]

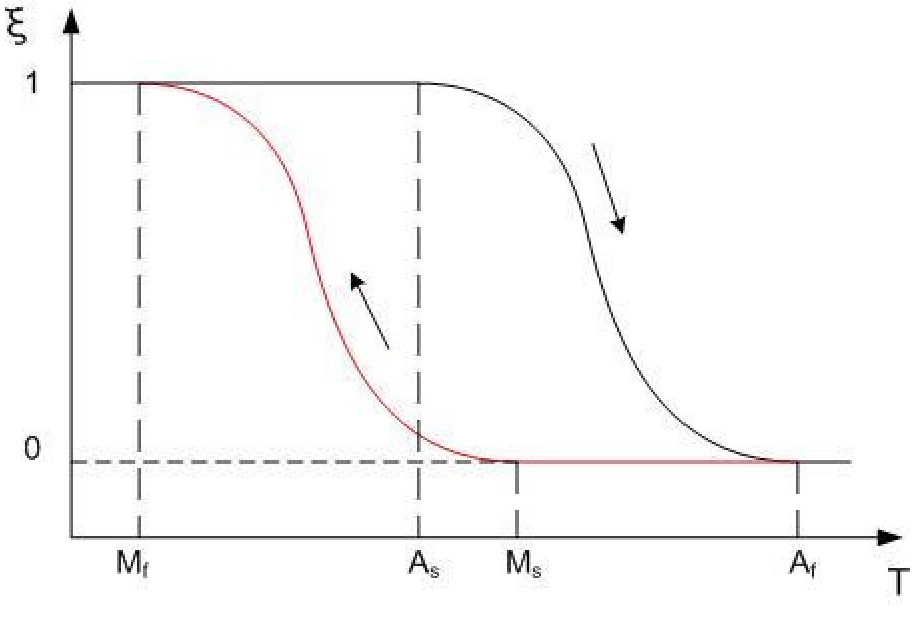

Figure 2 General representation of Strain/temperature hysteresis of SMA

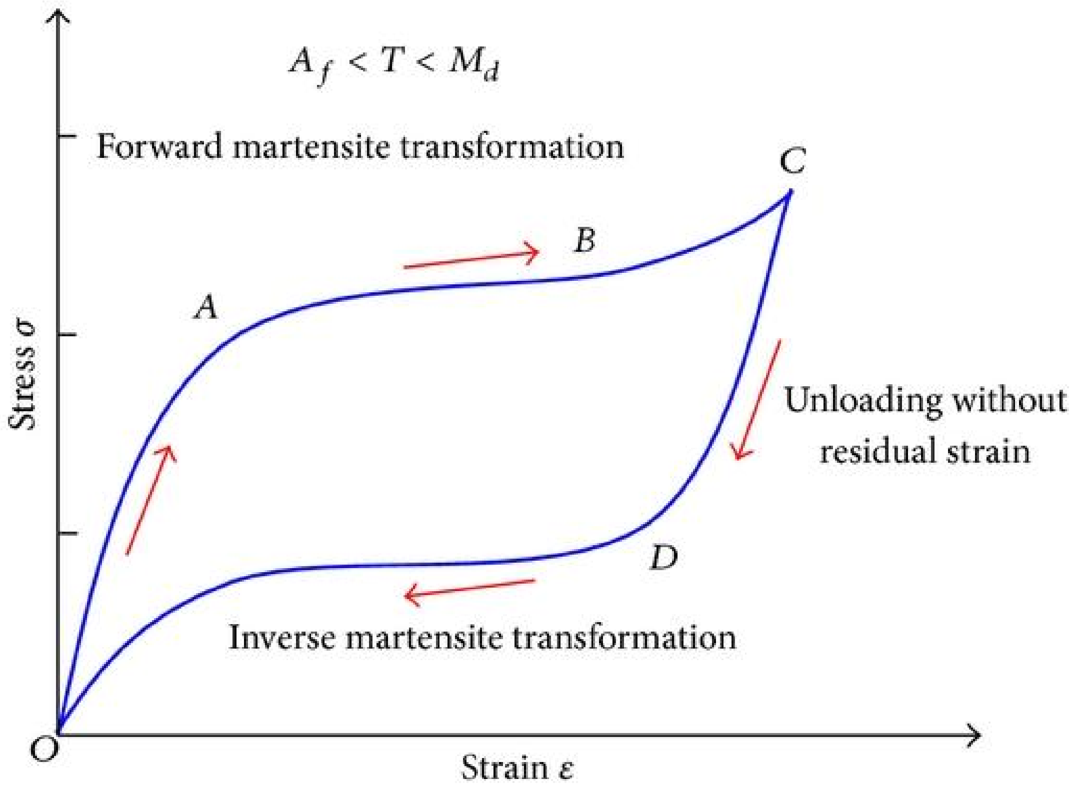

Figure 3:general representation of stress strain (NiTi) hysteresis

If we look at figure 3 that shows a diagram between stress and strain in NiTi we see that the loading path and the unloading path are not identical, surpassing point C will lead to failure. another aspect is that in martensite form (low temperature, high stress) and in austenite form (high temperature, low stress).[5]

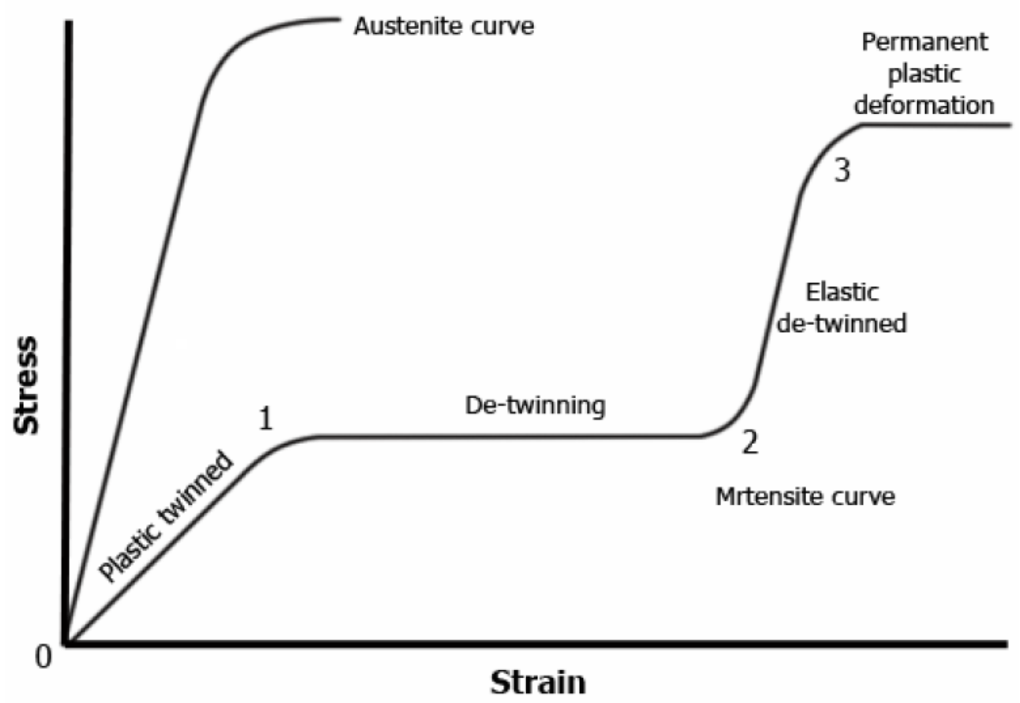

Figure 4:stress/strain hysteresis of SMA

Having the material in the martensite form and deforming it then applying high temperature that buts it in the beginning of the austenite form it will return to its original form this behaviour of the alloy is called the 'one way' shape memory effect, because the original shape of the sample is imprinted in the memory of the material, while the deformed configuration can no longer be obtained except with a new stress. The effect that causes these materials to be used as actuators. In fact, as can be seen in figure 4, when the alloy is in the deformed martensitic state it will present a certain stress-deformation graph, while following a heating that transforms it into austenite it will be able to develop, with the same deformation, considerably greater efforts.

Furthermore, martensite, having overcome the recoverable deformation field (region 1-2 in figure 4), will have a second elastic field, beyond which there will be a permanent deformation which cannot be recovered by converting it into austenite.

Temperature change is frequently achieved using a current pulse, particularly in nickel–titanium alloys, where resistivity is reasonable. A change in crystal structure results in dimensional changes that are typically between 1% and 8%. These strains can be induced under very high loads, leading to high work densities (>1 MJ/m^3). Furthermore, by employing effective cooling, millisecond time scale impulse responses can be induced by heating, followed by quick cooling from nucleate boiling. Typically the shape memory alloy contracts at high temperature, and a tensile stress is needed to return it to its elongated state following cooling. A simplified version of the shape memory cycle is shown in Figure 5, reversible deformation in nickel-titanium alloys is generally produced by cooling the material from the austenitic phase. A twinned martensite structure results. Application of stress leads to deformation. Increased temperature returns the martensite to the original austenitic phase, generating strain. The cycle is repeated to produce actuation.

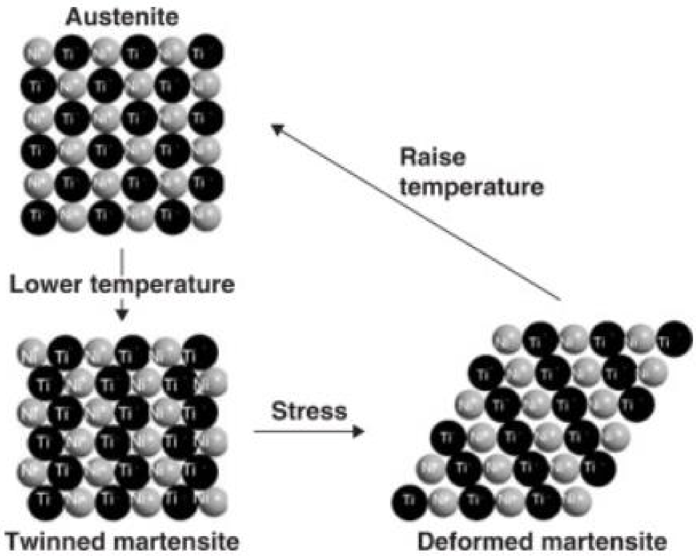

Figure 5 . A simplified version of the shape memory cycle

Two-way shape memory effects are also possible in which physical deformation of the twinned state is not required. Mechanical elongation can be avoided when a two-way shape memory effect is induced.

If we compare between the one-way and the two-way shape memory effects the procedures are very similar: starting from martensite (a), adding a reversible deformation for the one-way effect or severe deformation with an irreversible amount for the two-way (b), heating the sample (c) and cooling it again (d).

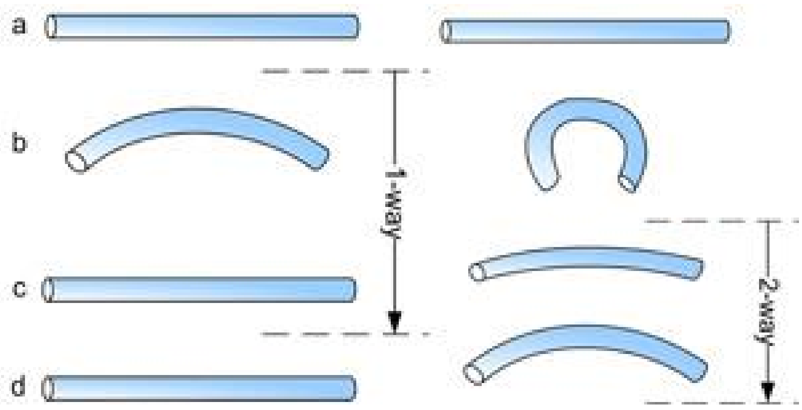

Figure 6 one-way VS two-way shape memory effect

1. One-way shape memory effect

   When a shape-memory alloy is in its cold state (below A_s), the metal can be bent or stretched and will hold those shapes until heated above the transition temperature. Upon heating, the shape changes to its original. When the metal cools again, it will retain the shape, until deformed again.

   With the one-way effect, cooling from high temperatures does not cause a macroscopic shape change. A deformation is necessary to create the low-temperature shape. On heating, transformation starts at A_s and is completed at A_f (typically 2 to 20 °C or hotter, depending on the alloy or the loading conditions). A_s is determined by the alloy type and composition and can vary between −150 °C and 200 °C.

2. Two-way shape memory effect

   The two-way shape-memory effect is the effect that the material remembers two different shapes: one at low temperatures, and one at the high-temperature shape. A material that shows a shape-memory effect during both heating and cooling is said to have two-way shape memory. This can also be obtained without the application of an external force (intrinsic two-way effect). The reason the material behaves so differently in these situations lies in training. Training implies that a shape memory can "learn" to behave in a certain way. Under normal circumstances, a shape-memory alloy "remembers" its low-temperature shape, but upon heating to recover the high-temperature shape, immediately "forgets" the low-temperature shape. However, it can be "trained" to "remember" to leave some reminders of the deformed low-temperature condition in the high-temperature phases.[6]

Innovative rotary actuators

As we know there are many types of actuators, but the one we are interested in are the rotary actuators, so let’s take a tour in what are the rotary actuators, different types and the new and innovative technologies used to improve them now days.

The main function of a **rotary actuator** is to produce a rotary motion or torque.

A pure mechanic actuator is the simplest form of a rotary actuator, where linear motion in one direction gives rise to rotation. The most common actuators are electrically powered; others may be powered pneumatically or hydraulically, or use energy stored in springs.

A rotary actuator can have either continuous rotation, as for an electric motor, or movement to a fixed angular position as for servomotors and stepper motors. A further form, the torque motor, does not necessarily produce any rotation but merely generates a precise torque which then either causes rotation, or is balanced by some opposing torque.

- **Stepper motor**

  Stepper motors are a form of electric motor that move in discrete steps of a fixed size (an angle for each pulse). It can be used to have a continuous rotation at a controlled speed or to move by a controlled angular amount. If the stepper is combined with either a position encoder or at least a single datum sensor at the zero position, it is possible to move the motor to any angular position and so to act as a rotary actuator.

  This kind of rotary actuators are used commonly in floppy disk drives, flatbed scanners, computer printers, plotters, slot machines, image scanners, compact disc drives, intelligent lighting, camera lenses, CNC machines, 3D printers and in In the field of lasers and optics.

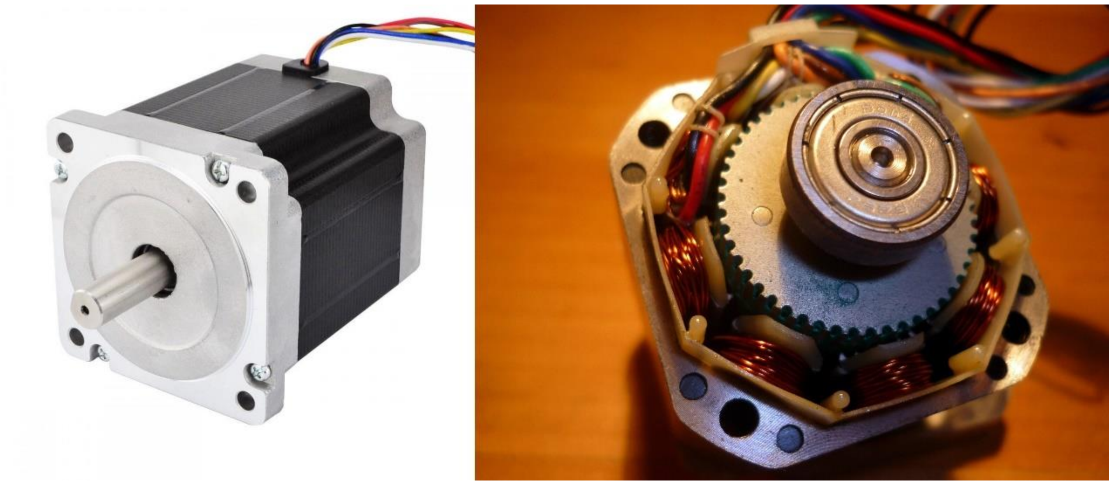

Figure 7 stepper motor (exterior and interior)

- **Servo motor**

  A servo motor is a motor that use a gear train to reduce speed and increase the rotation torque, it has a control system and a position encoder to identify the position of its shaft. The input control signal to the servo indicates the desired output position. Any difference between the position commanded and the position of the encoder gives rise to an error signal that causes the motor and gear train to rotate until the encoder reflects a position matching that commanded (It acts as a closed loop system).

Servomotors are used in applications such as robotics, CNC machinery or automated manufacturing.

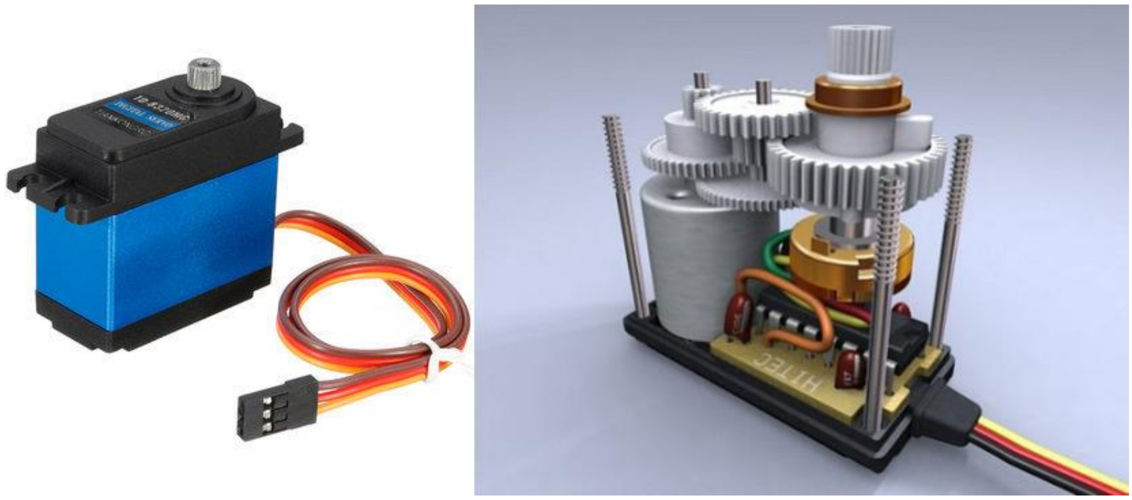

Figure 8 Servo motor (exterior and interior)

- **Memory material based rotary actuators**

  They are ultra-lightweight actuator uses memory wire. By applying current , the wire is heated above its transition temperature and so changes shape, applying a torque to the output shaft. When power is removed, the wire cools and returns to its earlier shape, this type of actuators is the one used in our project.

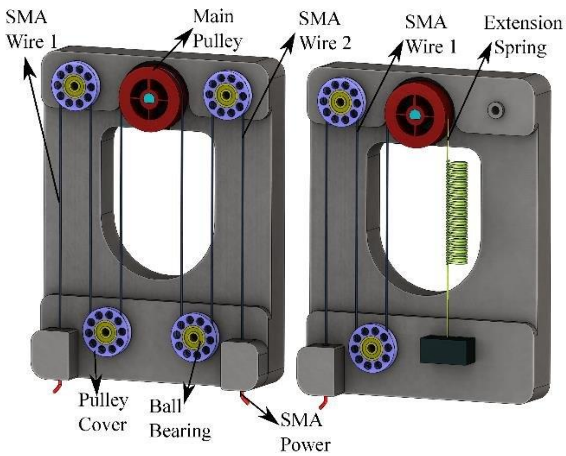

Figure 9 SMA wire based actuator

There are more rotary actuators, let’s take a look at a piezoelectric based actuator for example one develop by and Italian innovative company that starts in collaboration with Politecnico di Torino university ,they called it as high precision rotary actuator ,they work by using **innovative mechanisms** which have originated a number of **patents**. The innovation of our mechanisms is based on the exploitation of the **elastic deformation of materials**.

The rotating drives **has a** rigid ring at his heart, and piezoelectric actuation systems are used to produce its elastic deformation, which is in turn conveyed into a controlled, accurate movement. Changing the dimensions of our components, we can develop **high speeds and torques**, without giving up on a precise positioning.[7]

Figure 10 PHI DRIVE piezoelectric high precision rotary actuator

SMA rotary actuators

They can be divided into two categories, according to the output rotation: Continuously rotary actuators, which have a permanent rotation ability; and non-continuously rotary actuators, which have a finite rotation angle (usually less than 360).

- **Continuously SMA rotary actuators**

  We can say that they are SMA motors and most of them can rotate in clockwise and anticlockwise direction by control them in different sequences.

  A model developed by Sharma in (2008) as shown in the figure (11) is a SMA wire-based poly-phase motor (3 equal phases for the motor), where each phase uses a SMA wire and a tension spring in series. The extension and the pulley are used to fold back the SMA wires and achieve compact dimensions, this motor works bay heating and cooling the SMA wires in a certain sequence. This model can achieve a rotation speed of 10 r/min and 7 r/min with forced air and natural air-cooling methods, respectively. In addition, either clockwise or anticlockwise rotation can be achieved through different phase sequencing schemes.

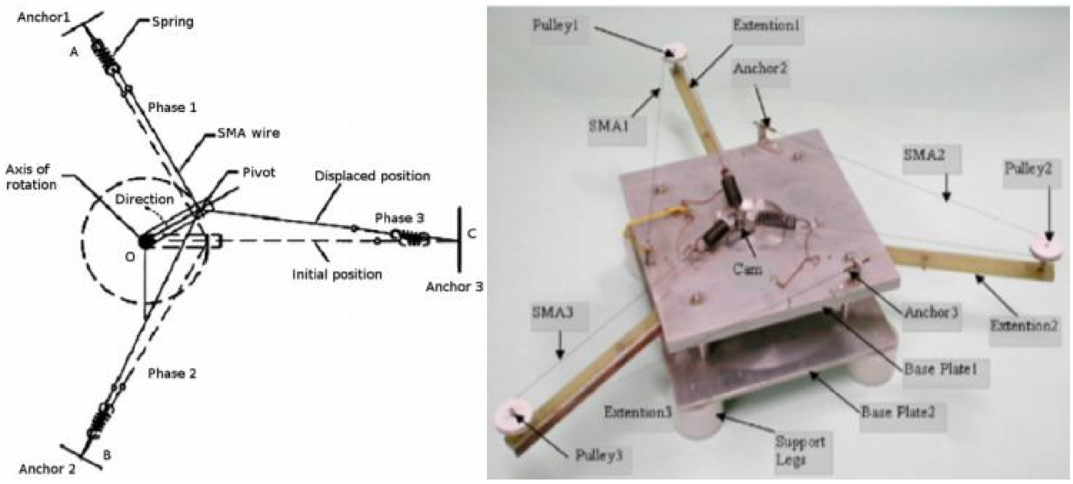

Figure 11 SMA wire-based poly-phase motor Sharma (2008)

Zhang and Yan motor is a continuous one-way rotating motor, driven by multiple segments of SMA wire, based on the friction ratchet mechanism and bias spring device (12) driven by SMA wire, each actuator is activated at the appropriate time, the motor can A continuous speed of 0.28 r/min is achieved with a torque of 1008 Nmm.

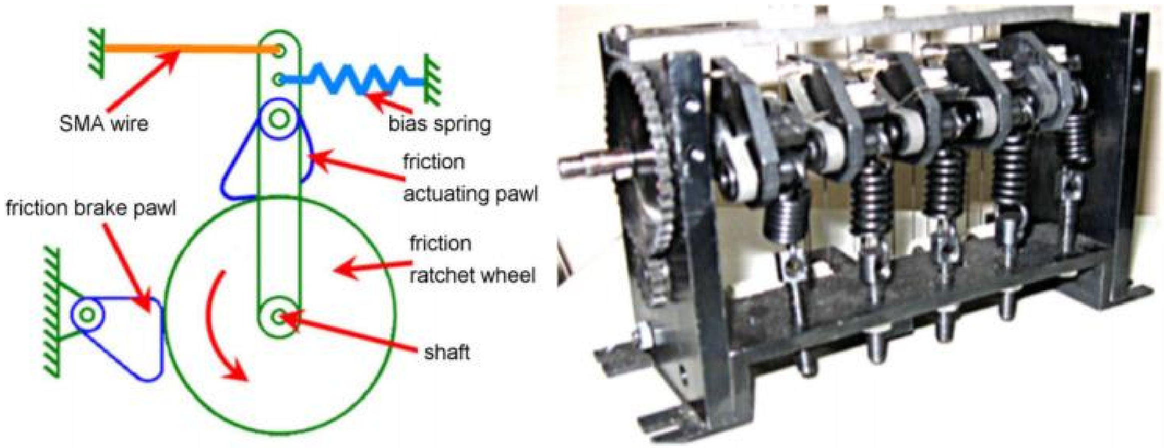

Figure 12 . Continuous SMA motor consisting of 5 ratchet wheel-based unidirectional rotary actuators (Zhang and Yan)

A rotary actuator using SMA wires is proposed and tested in the work by Hwang and Higuchi (2014), the rotor is rotated by the wobbler in a wobbling motion which is driven by the three sequential external forces generated from the three parallel SMA-spring systems placed mechanically in parallel. The three identical crankshafts are designed to guide the wobbling motion. A Ni-Ti-Cu sma wires were heated by current in the prototype that generates a max torque of 2.6 Nmm and a max rotation speed of 11 r/min.

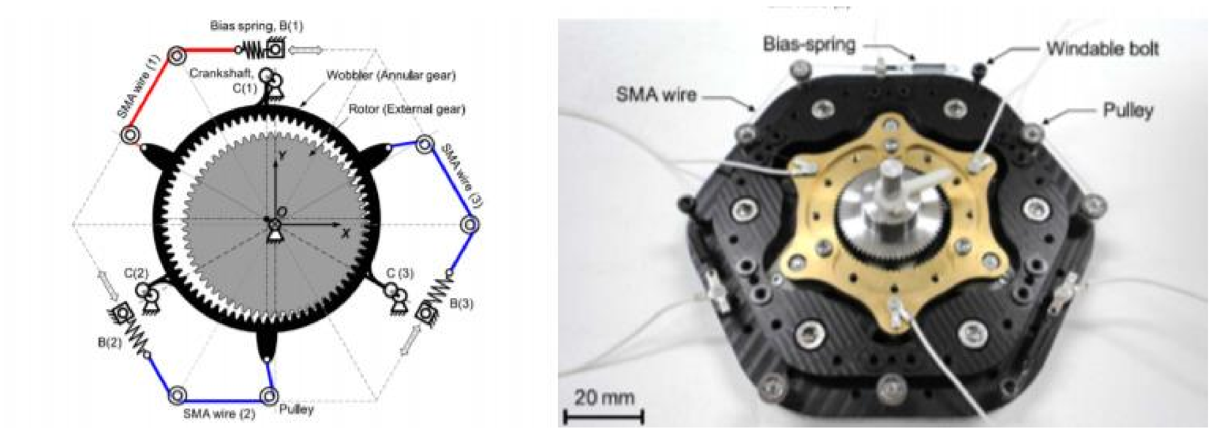

Figure 13 SMA-driven stepping motor based on wobbling motion, Hwang and Higuchi (2014).

Other motors driven by SMAs can be also found in the work by Dilthey and Meier (2009) figure(14), where a stepping motor powered by straight SMA wires is developed based on a strain wave gear mechanism (harmonic drive) achieving a maximum torque of 2500 Nmm, a maximum rotation speed of 0.17 r/min and an accuracy of 0.9.

Figure 14 SMA-powered rotary harmonic drive system, Dilthey and Meier (2009).

- **Non-continuous SMA rotary actuators**

Paik's (2010) work proposed a low-profile torsion actuator suitable for micro-robots. As shown in Figure (15), the actuator is made of a whole piece of SMA thin plate with a thickness of only 100 mm and a weight of only 0.033 g. The transformation temperature from martensite to austenite is about 60°C, which can be achieved by passing current through Ni-Cr (high resistance) wires embedded in the SMA plate. The actuator can perform folding actions (180 rotations) with a maximum torque of 4.5 Nmm. (It is a one-way rotary actuator).

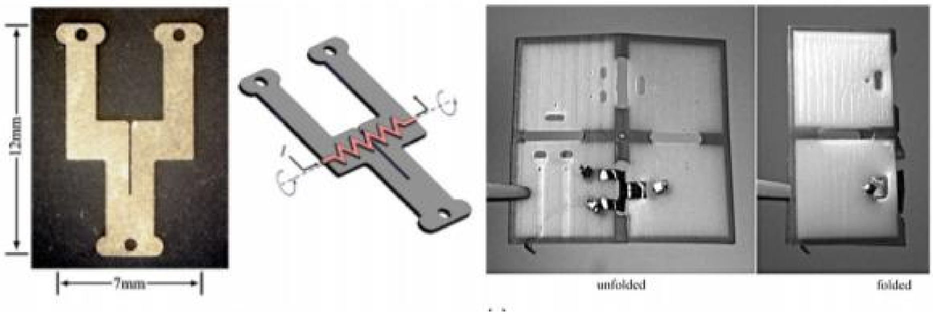

Figure 15 . A small-size folding actuator made of thin SMA sheet, Paik (2010) +folding test

Lan e J.H.'s work developed a bidirectional rotary actuator driven by SMA wires and specially designed flexible parts. Wang (2009). As shown in Figure (16), the pre-strained NiTi SMA wire with a diameter of 125 mm passes through 9 fixed posts and 3 movable pins to form a curved ring on a 60 mm disk. This special design can achieve a longer wire length in a limited space. At the same time, the three pins on the compliant part can get the same force and displacement through the ring. A prototype was made, and the test results showed that when the SMA wire is heated by the current, the actuator can achieve 19° rotation, with a maximum output torque of 31 Nmm, and return to its original position by the elastic force of the compliance pin. Parts when SMA wire is cooled. This design ignores the friction between the SMA wire and the pin. However, if friction is considered, the three centre pins cannot obtain the same force or displacement. In this case, the movement of the central axis is no longer pure rotation. [8]

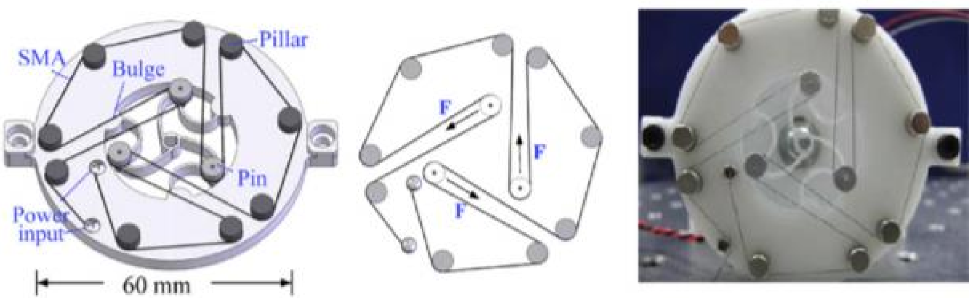

Figure 16 . Rotary actuator using SMA wire-actuated flexure, Lan e J.H. Wang (2009)

A portable magnetic SMA rotary actuator is designed and implemented in the work by Viscuso and Pittaccio (2012) figure (17), where a series of pulleys is mounted on the shaft and a pre-strained nitinol wire with a diameter of 0.25 mm is double coiled on these pulleys. In this way, a linear movement of the SMA wire can be converted into a rotation in a compact space. Moreover, the rotation angle of the actuator increases with the number of coiled turns. Another significant aspect of this actuator is the amagmatic characteristic. By the double coiling of the wire, the electric current flows in opposite directions in the forward and reverse coils. These magnetic transparency is achieved. A rotational movement can be obtained with a maximum angle of 38° and an output torque of 1220 Nmm when the SMA wire is heated by the Joule effect. A linear steel spring (which cannot be seen in figure 17) is used as the bias spring to achieve the return motion. A remarkable feature of this actuator is that it introduces pulleys in the design to decrease friction and thus increase output stroke and torque.

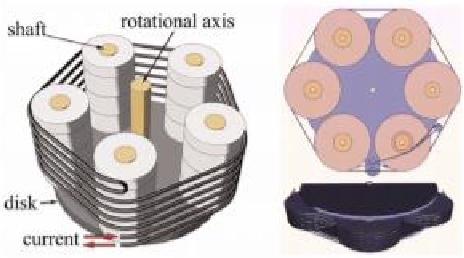

Figure 17Portable amagmatic SMA rotary actuator, Viscuso and Pittaccio(2012)

A SMA-actuated rotary servo is presented in the work by Song (2007). As shown in figure 18, a nitinol SMA wire is spooled on a threaded non-conductive rotary drum. One end of the SMA wire is fixed to the drum and the other end is fixed to a supporting base plate. A torsional spring is connected to the drum in order to achieve two-way rotation. In addition, a sliding-mode-based robust control approach is developed to improve servo accuracy. Experimental results show that this actuator can achieve a rotation angle of 100° with a steady state error of 0.2.

This actuator design is compact and offers a space-saving solution for the use of SMA wires. However, if friction between the drum surface and the SMA wire is considered, only a small part of the spooled SMA wire will be free to retract.

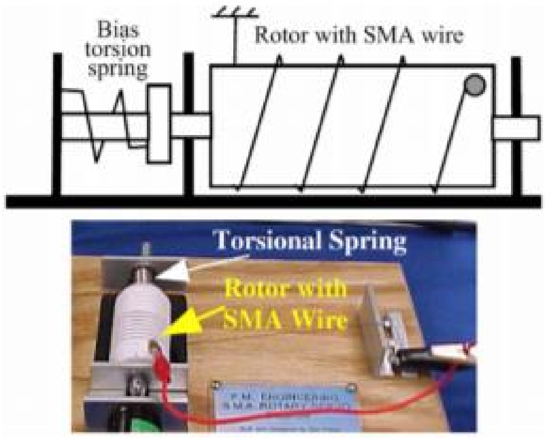

Figure 18 SMA-actuated rotary servo, Song, (2007).

Another rotary helical spire driven by SMA wire is presented in the work by Yuan (2016) figure (19). This actuator is composed of a fully compliant structure made of 3D-printed monolithic parts and a SMA wire, allowing a highly integrated form. The plastic structure features a helical shape. A Φ1 mm Ni-Ti wire is first stretched in the martensitic state at ambient temperature to obtain a 1.4% pre-strain. Then the SMA wire is wrapped onto the helical structure and fixed at its two ends. An electric current is applied to heat the SMA wire by the Joule effect, leading to the rotation of one end of the helix with respect to the other. The return to ambient temperature is obtained simply by turning off the current. The helical structure, which acts as a bias spring, returns to its original shape due to its own elasticity, generating a rotation in the opposite direction. The rotation angle attained depends on the current intensity and the thermal boundary conditions. With a 4 A current, a rotation angle of 250° is achieved. A remarkable feature of this actuator is that the friction between the driven SMA wire and the supporting structure is negligible, since there exists no sliding between the SMA wire and the plastic. In addition, the helical structure reinforces the stroke output capability of the actuator in a limited space.

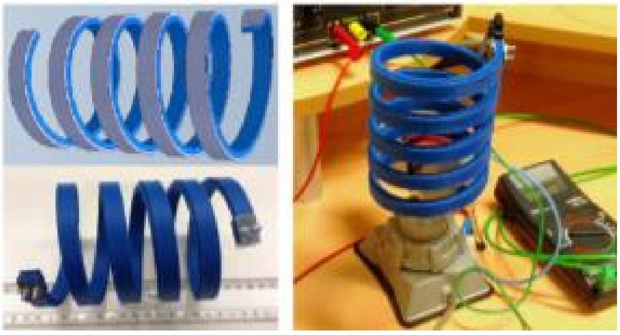

Figure 19 Rotary helical spire driven by SMA wire Yuan (2016)

Project SMA material specifications

It was chosen to use the NiTi SMA wires, In the choice of the implementation wire, the one that developed the major force from Dynalloy Inc, it was decided to use the one with the larger diameter, that is 0.25 mm.

The technical characteristics, supplied with the wire, are shown in the following table:

Name	250
Diameter	0,25 mm
Minimum radius of curvature	12,5 mm
Linear resistance	18.5 Ω/m
Recommended current	1050 mA
Maximum stress	28.77 N
Recommended stress (implementation)	9.1 N
Recommended stress (recovery)	3.5 N
Typical speed of contraction	1 s
Start activation temperature	88 °C
finish activation temperature	98 °C
Relaxation start temperature	72 °C
Relaxation finish temperature	62 °C
Annealing temperature	300 °C
Melting temperature	1300 °C
elongation	3-5 %

Table 1Technical characteristics of the SMA wire used

The Prototype

A prototype of a rotary motor that uses SMA wires was designed by (Luca Amiccui) in 2017 at Politecnico di Torino university and the procedure will continue from what he stopped at.

The conceptual scheme of the actuator is very similar to that of Zhang and Yan, but it uses a ratcheting motion transmission and locking system. That's it In fact, due to the strict tolerances, an attempt was made to solve the problem that has been discussed about the change in the rotation angle between Zhang Heyan's five actuator units.

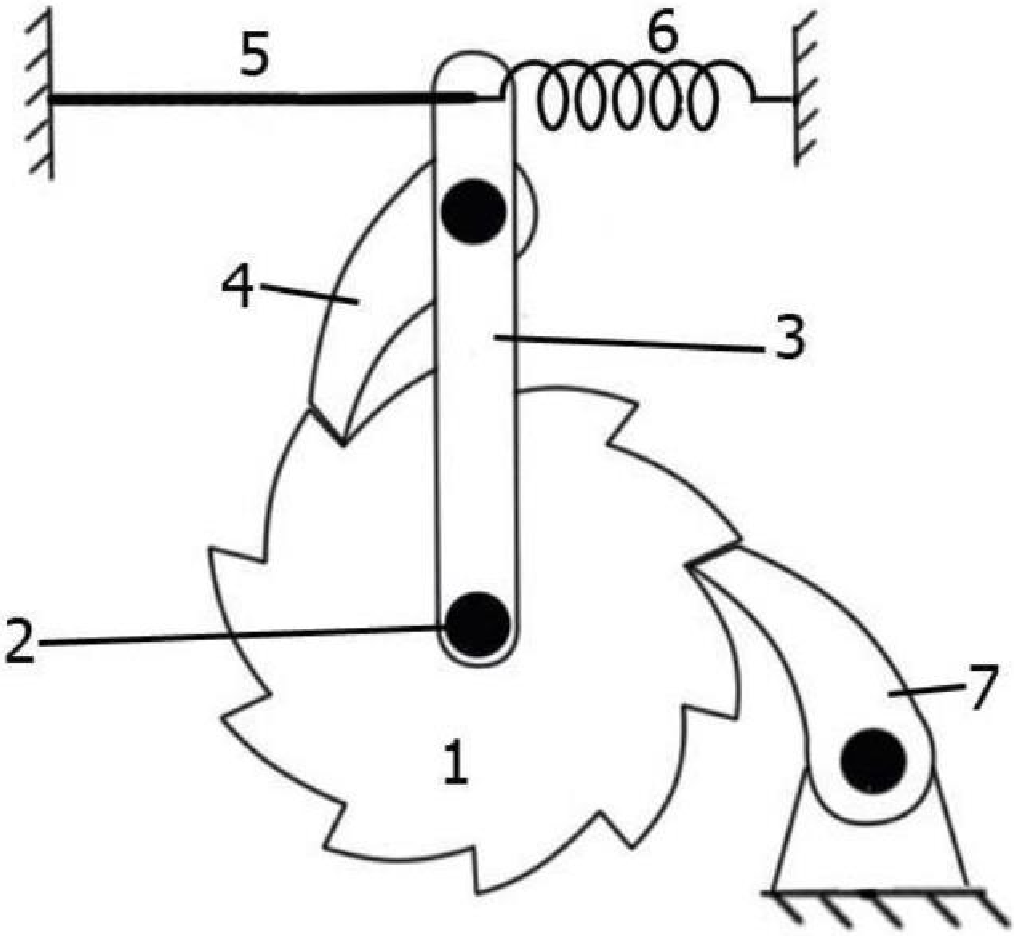

Figure 20: Conceptual diagram of the actuator developed

The basic diagram of the actuator in figure(20) is composed of the following elements:

1. Toothed wheel
2. Shaft, which will transmit the outgoing motion
3. Leverage
4. Motion transmission harpoon
5. SMA wire
6. Recovery spring
7. Stop harpoon

When current flows to the SMA wire fixed at one end to the frame and the other end to the end of the element 3, it will generate heat, and after the temperature is exceeded, it will begin to shorten and generate the force capable of rotating the lever.

This will move the gear (the gear of the hinged lever) through the harpoon 4, which will mesh with one of the teeth. In this way, the movement will be transferred to the axle 2 integrated with the wheel, and the harpoon 7 will not work at this stage because it will only slide on the teeth.

When the current is removed, the SMA wire will cool down and return to martensitic thanks also to the stress imposed by the recoil spring (bound with one end to the frame and the other to the lever), which will deform it up to the initial conditions. In this second phase, the harpoons 4 and 7 will reverse their roles: therefore, while the harpoon 7 will block the retrograde motion of the wheel, the harpoon 4 will slide on the teeth without intervening in the phase. In this way it is possible to obtain a unidirectional rotation of the shaft even in case of imposed load.

The prototype is illustrated by its all components figure (21 and 22) and calculations.

From the exploded view it is possible to see that the 80-tooth wheel(gear) (12) and the lever (5) are fastened to the container (1) by means of an M5 centring screw (13). Between the wheel and the screw there is a cylindrical bushing (10) with internal diameter 6 mm and 3 mm wide (6x3), for the reduction of friction, while between the screw and the lever there is a flanged bushing (9) 6x4 with 1 mm thick flange, which reduces the contact surface between wheel and lever. Under the latter there are two washers (2 and 3) 1 mm thick for the recovery of the thicknesses. The spiral spring (23) is instead integral with the central pin.
The transmission harpoon (7) is constrained to the lever by an M3 centring screw (8). The latter is screwed to a threaded bush M3 (4), 3.8 mm wide and with a 1 mm thick flange, positioned behind the lever. Between the harpoon and the screw there is a 4x3 bushing (11), while between harpoon and lever there is a washer (6) 0.55 mm thick, which also in this case reduces the surfaces in contact.
As for the locking harpoon, there is again a fixing with M3 screw and 4x3 bushing. Under the harpoon there are two washers 0.55 mm thick.
The output shaft (15) is mounted on the 32-tooth wheel (16) and on two 6x8 bushings (14). Finally, the cable guides (17) are mounted on bearings 638 / 8-2Z (18).[9]

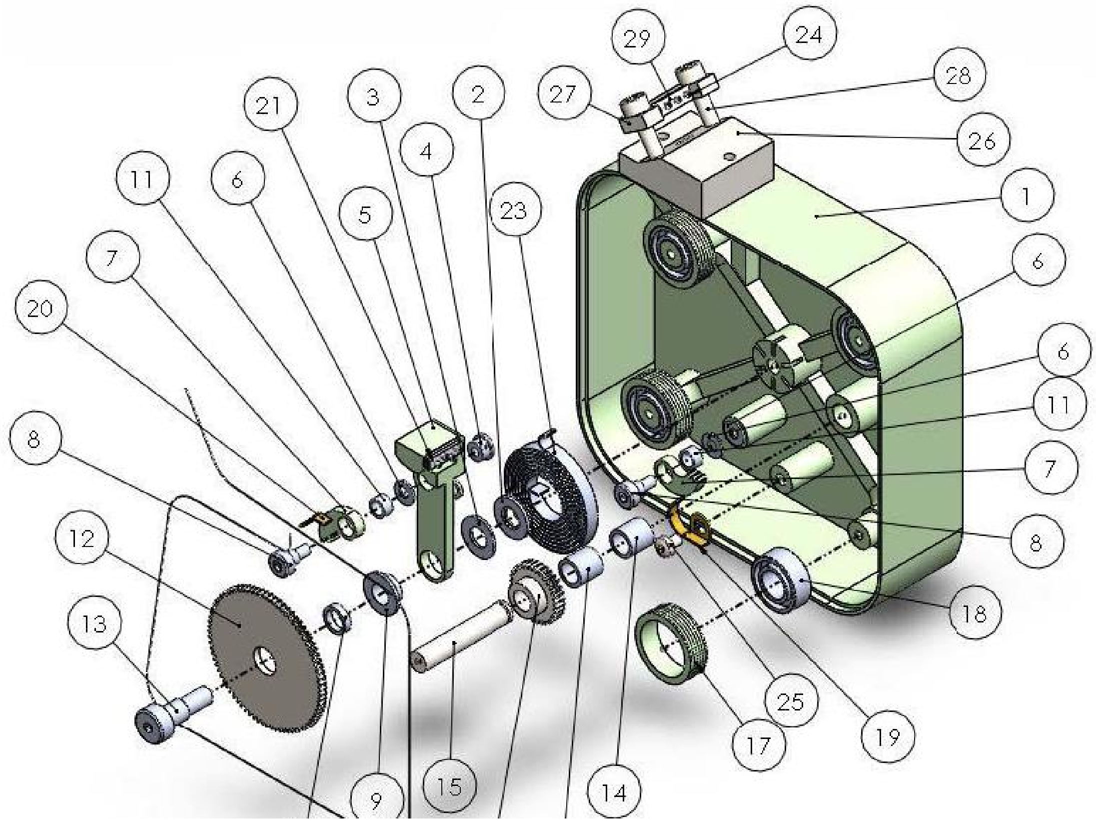

Figure 21:expanded view of the components of the final configuration

Based on some other previous designs in order to have an actuator with incremental performance without weighing on the volume, which happens in the actuators of both Zhang and Yan and Mammano and Dragoni, in which each module adds power but also thickness, it was decided to connect the lever up to with three wire-spring pairs.

In this way, each wire will develop during implementation a maximum force of:

Implementation stress-recovery stress=9.1-3.5=5.6N

So our final theoretical (total) force is :3*5.6=16.8N

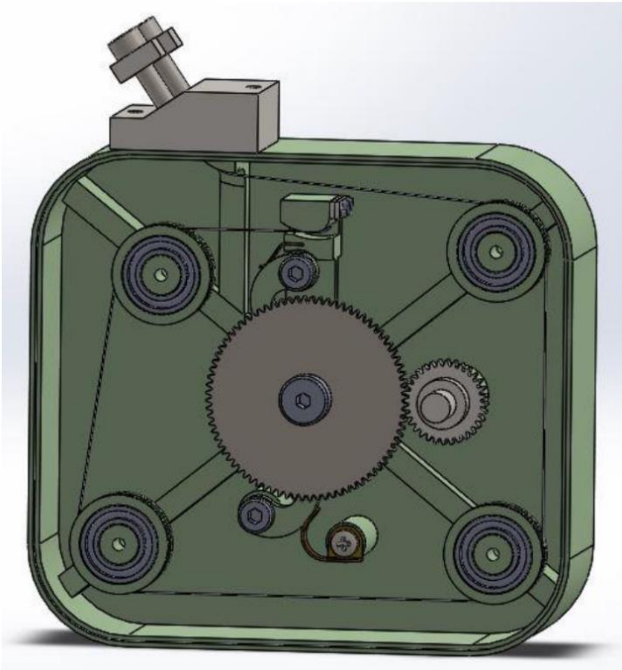

Figure 22 Assembled final configuration

Theoretical Calculation

The idea here was to have a quick implementation about the length of the wire or in other words to know what we want from the output of our device for example angle of rotation and torque so for a certain wanted values it must have a certain specification in the length of the wires and what is varying in our prototype are the diameters of the 4 pulleys in the corners of our prototype so changing there values changes the length of the SMA wire so in consequence changes the deformation of the wire.

Using MATLAB and SIMULINK the theoretical equation and the values based on the design of the prototype were introduced in order to get quick values of the deformation in the SMA wire that will show the final angle of rotation of the transition shaft that will provide us the torque of this motor actuated by the SMA wires.

The design provide two fixed curves so they have a fixed angle to calculate the length of this curves , we have 4 pulleys of the same size but the wire does not touch them with the same length so we need to know the length of the four touching curves that is calculated by the angels and the radius of the pulleys then the rest is easy to measure because they are straight lines.

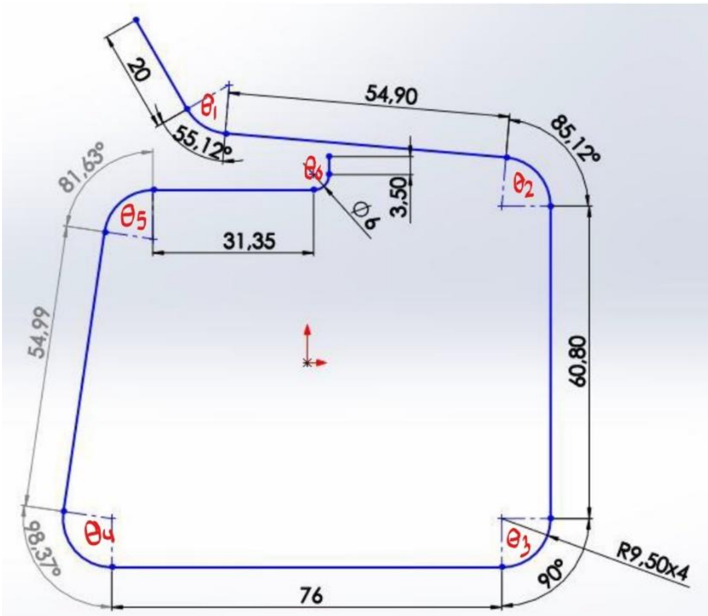

Figure 23:Representation of the SMA wire path

The Code:

```
clear all
clc

% first angle of the touching fixed curve
theta1=55.12;

%angel of the touching for the pulleys
th2=85.12;
th3=90;
th4=98.37;
th5=81.63;

%second angle of the touching fixed curve
theta6=pi/2;

%radius for the pulleys and the first fixed curve
r1=(9.5);

%radius for the second fixed curve
r2=3;

% percentage of shortening
per=0.03;    %from 3% to 5%
simout=sim('calculations_of_length');


%sma wire length
L=simout.yout{1}.Values.Data
```

A Simulink block was used in order to do all the calculations for the length of the SMA Wire that is represented in figure (24).

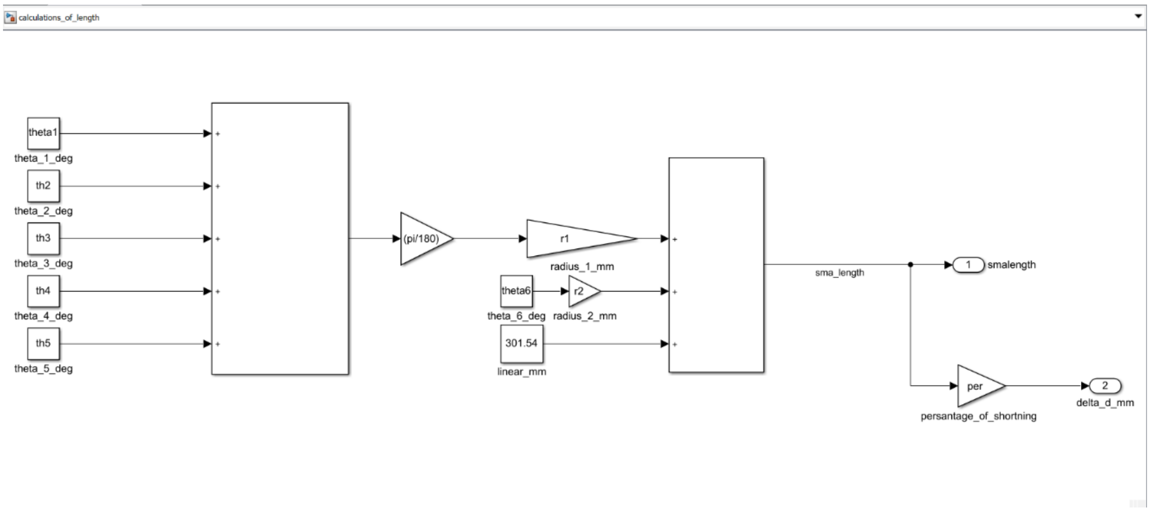

Figure 24:Simulink block for length calculation of the wire

Then here we have the rest of the code that provide us with the deformation (shrinking) of the wire and the final angle of the transition shaft

```
%srinking in leangth
delta_L=simout.yout{2}.Values.Data

%having a tooth in the bigger gear every 4.5 degree
%so our Rotation angle is  (degree)
Ang=(delta_L/34)*(180/pi);
Ang=int16(fix(Ang/4.5));

%angle of iterest for the big gear
ang_int=Ang*4.5

%with a gear ratio of 2.5
Ang2=ang_int*2.5
```

Here the result for the final angle is (45°) that corresponds to 4 teethes of the big gear and 10 teethes of the small gear.

From the calculations the length of the wire should be 374.278 mm and the deformation delta_L = 11.2282 mm.

The table below shows how different size of pulleys can affect the length and the output of the actuator here is the values with an almost 3mm variation in the radius of the pulleys.

Pulleys radius(mm)	Wire length(mm)	Delta_L (mm)	Angle of big gear	Angle of small gear
3	327.73	9.83	14	35
6	349.21	10.47	14	35
9.5	374.27	11.22	18	45
12	392.17	11.76	18	45
15	413.56	12.4	18	45

Table 2:variation of pulleys radios to wires length

Prototype first experiment

For the first implementation on the physical prototype it was a must to study all of its component to be aware of every detail in the device it was observed that our pulleys had a force applied in them due to the wires so they were not rotate very smoothly particularly.

It was noticeable too that the two transmission harpoon due to the difference in material between them and the big gear and the friction that occur between the teeth of the harpoon and the gear due to movement and force with the time and the use of the device they eroded so it was better in the future to change them with a more robust material.

The test bench was composed of a power supply that has a regulable voltage and Current of 30V and 3Amper , two wires that can resist the power of 90W to ensure that the connection is stable between the power source and the device and finally our prototype that it was connected in series direct connection with the power supply one end with the positive and the other end with the positive of the power supply.

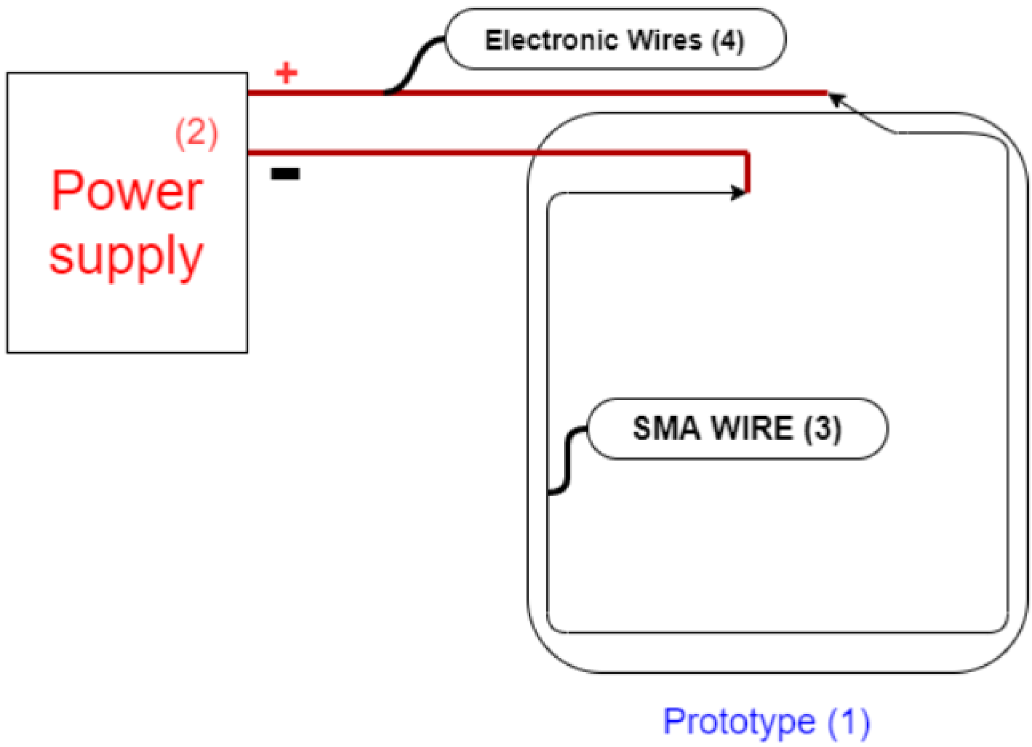

Figure 25:Simple scheme of the first test bench

Continuing the process after Amicucci that he was only able to actuate the best efficiency of the prototype by only one SMA wire at 6.6V and 1050mA so with a power of 6.93W so the experiment was redone with this values to ensure the performance of the prototype with the single wire the results out from the bigger gear was with an average of 3 teeth(that is the output corresponding to 14 degrees on the bigger gear and 35 degree of the small gear) for every tentative and changing the power from the power supply in a certain range affects only the speed of the deformation it was used the theoretical voltage .(Results figure 26)The experiment was done about 5 times. (the power range was between 5.46W and 9.105W)

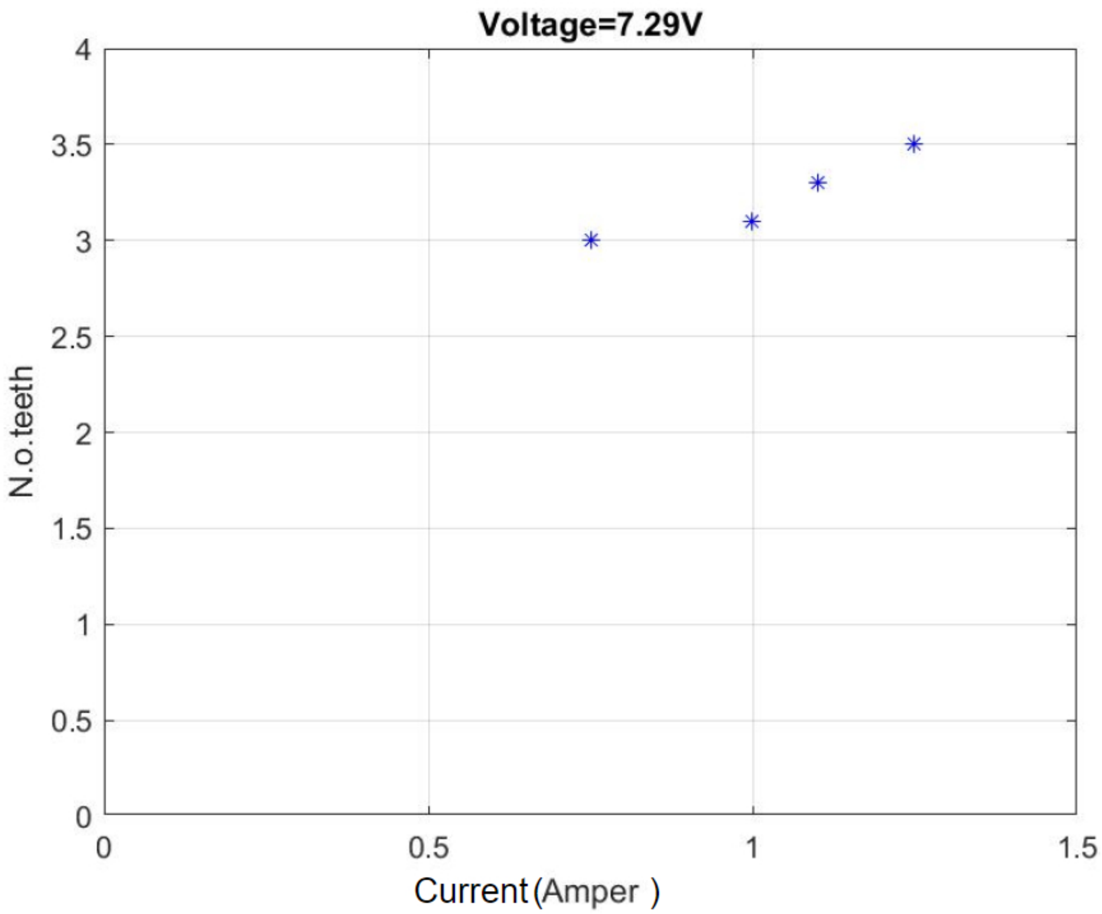

Figure 26:one SMA wire at voltage 7.29V

The retraction of the SMA wire to its zero position (initial potion) was very fast due to the large force produced by the torsion spring that is above the needed value of 3.5N and the average time was from 1s to 2s as maximum.

Doing the same experiment with more than one SMA wire so another wire was mounted and fixed with the same tension applied on the other one, with different power values provided by the power supply until we got the power value that provided us with the best result(between 16.8W and 21.25W) ,but there was a problem noticed that in the cooling phase the wire didn't return to its zero position so the experiment was redone buy attaching an external additional spring(with force equal to 3.5N) to add some force with the torsion spring to see if there is a problem in the wires or that the force of the torsion spring was not enough to ensure the wires to return to its zero position this was done by attaching another spring that was fixed from one end of the testing table and the other side to the lever that has the torsion spring and the wires, then the prototype was fixed to the table in a way that there is no force pre-applied form this additional spring, here a quick representation of it(figure 27).

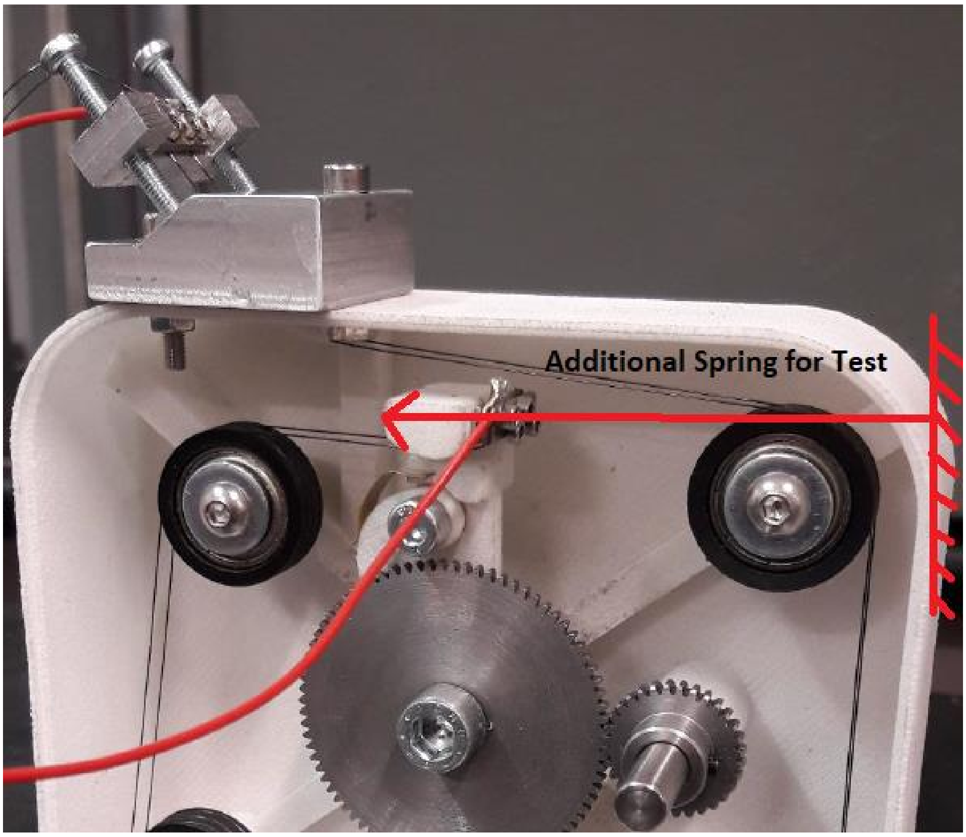

Figure 27: Representation of the test additional spring

The result was that with this additional spring the wires returns normally to there zero position that helped to arrive at a force of 7N as in the characteristics of the used SMA wire (Result figure 28).

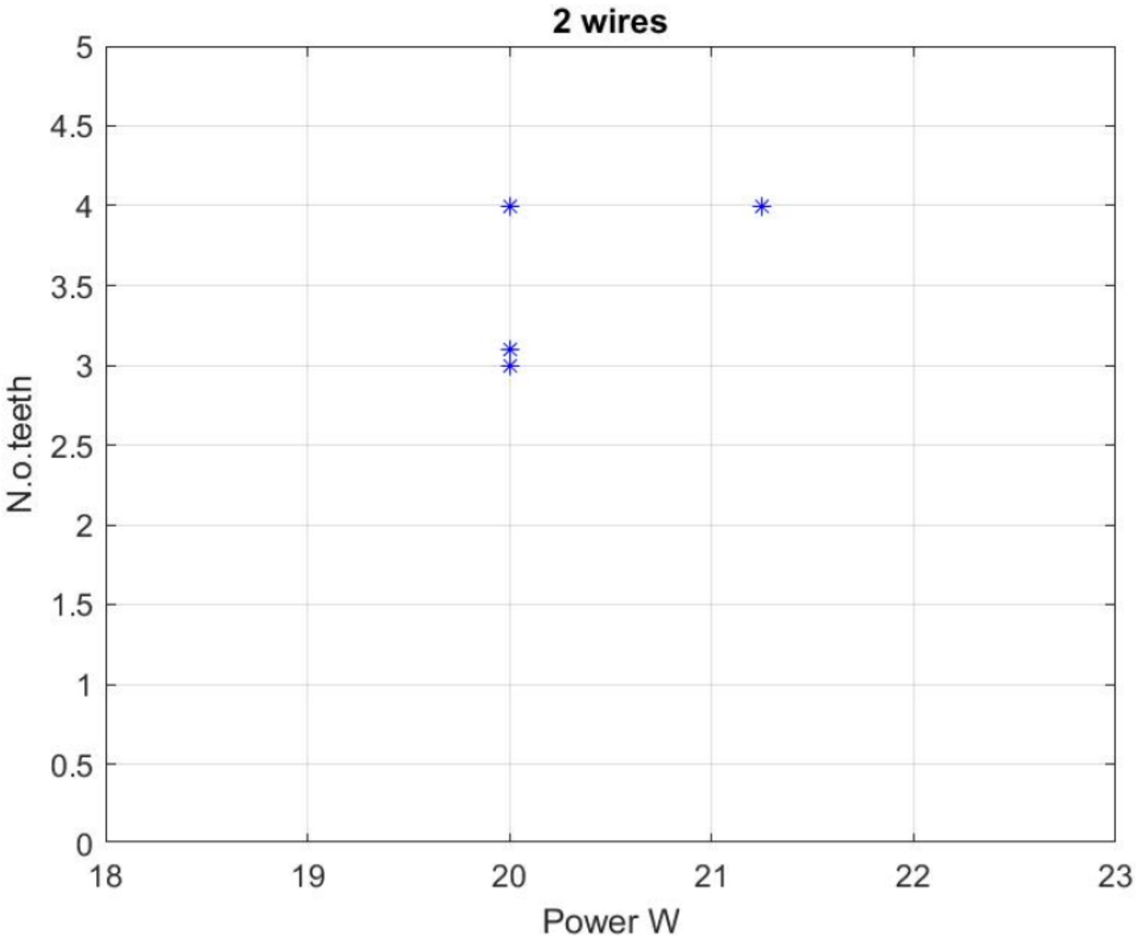

Figure 28:2 wire experiment

As seen from figure (28) the best result was at power equal to 21.25W that gets us 4 teeth for 6 consecutive tentative.

Another fact to be considered when using more than one wire that it is a must to ensure that all the wires has the same pre-applied force to get an accurate result and efficiency and that was done by a tensioning device (figure 29) that it is composed of 3 equal springs each equal to 3.5N to as the force needed in the retraction process but it was not granted that all the wires are tensioned equally due to the tolerance of the springs and of that this process is manual .

Figure 29:Tensioning device

The conclusion of this experiment was that for testing the control methods that were established is to use one SMA wire for simplicity.

Considering that for the use of more than one wire we need to replace the torsion spring to get additional retraction force and it was seen that the bearing needs some lubricant in order to get more smoothness that was observed after this tests in order to improve the performance of the device.

So after knowing how the prototype act and how much deformation is in the wire and the how it retracts in the cooling stage it was time to introduce the ideas of subjected control methods in order to make the prototype work in continuous mood and to save the wires from overheating and the risk to damage (brake) it.

There three main control methods that were decided to be applied to our project:

1. PID control to ensure that our wire temperature does not exceeds a certain temperature that is in our case 90°C to not damage the wire,
   changing the PID constants will make the controller maintain this temperature until the power is cut off.

2. Two methods to make the prototype work in a continuous loop by some conditions:

   • To have a sensor to know the wire temperature and to cut off the power from the wire when we reach a certain temperature

   • To have a sensor on the transmission shaft to know the shaft position and cut off the power from the wire after a certain movement.

The controlling methods explanation:

1. PID Controller

The main idea is to maintain the wire under a certain temperature to not damage the wire and that is done by a PID control generated on an Arduino uno kit.

PID is a control loop mechanism employing feedback from the output, it continuously calculates an error value as the difference between a desired setpoint in this case the temperature of the wires and a measured process variable the output and applies a correction based on proportional, integral, and derivative terms (denoted P, I, and D respectively) In practical terms it automatically applies accurate and responsive correction to a control function.

In our case we have that the austenite final temperature is 90°C so if we reach this temperature and the power is still provided without a control to the wire we have the risk to damage the wire (cut it) so by this PID controller using a thermocouple as the feedback sensor to get values and calculate the error to maintain the same temperature if it was reached and the power is still on, here a simple scheme that represent how the control idea works(figure 30).

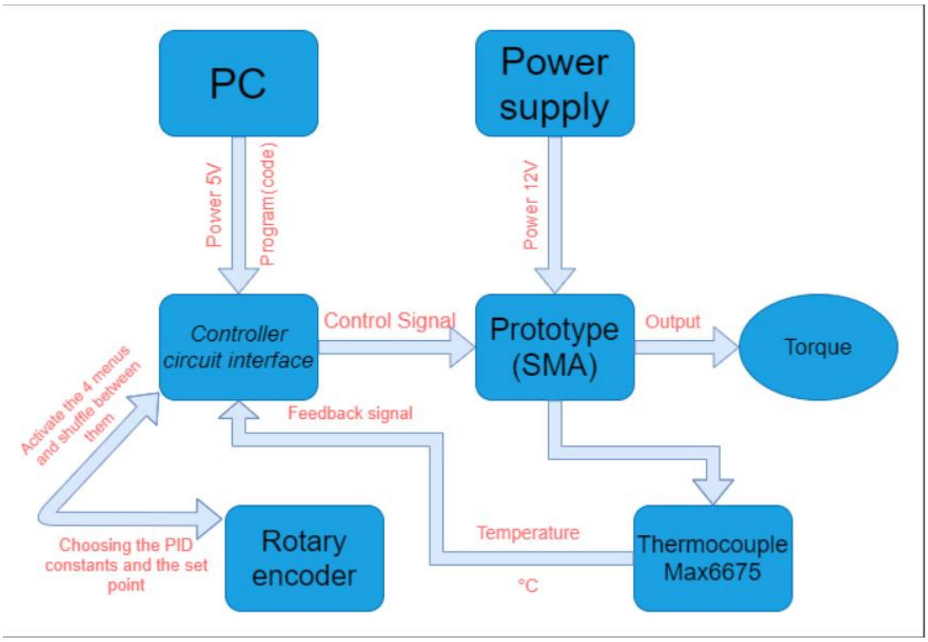

Figure 30:simple scheme of PID control

A hardware as in (figure 31) was established and tested with a low voltage LED as an output reference to ensure the functionality of our program.

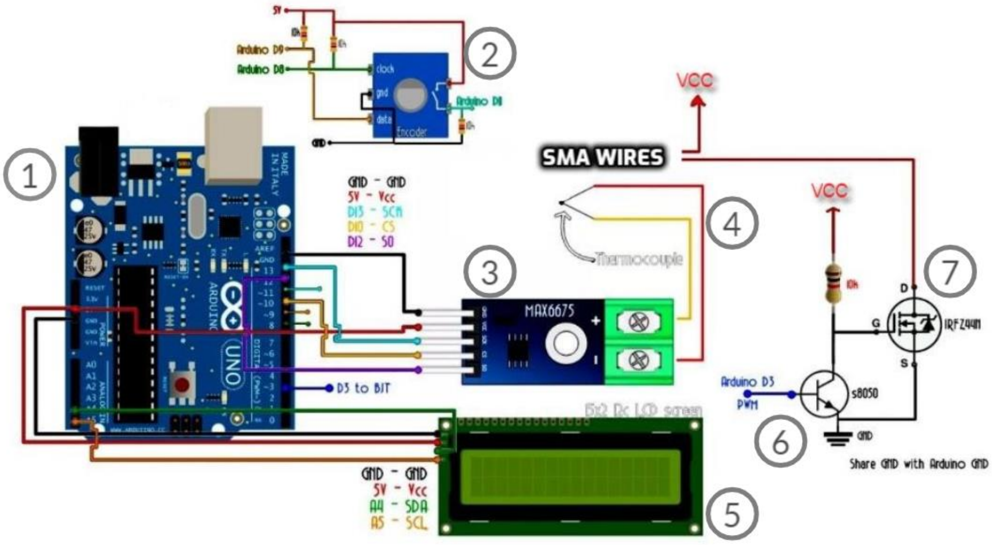

Figure 31:PID control setup

PID control components:

- Arduino uno (1)
- Rotary encoder (2)
- MAX6675 (3)
- K thermocouple (4)
- I2C LCD display (5)
- S8050 N-type transistor (6)
- IRFZ44N (MOSFET) (7)
- Regular Power Supply
- 10K resistance

The setup has an hardware of two transistors one the BJT s8050 to give the PWM signal(so we need to use particular pins on the Arduino to provide this type of signal and it was No.3) to the MOSFET IRFZ44N this hardware was made because our wire need a high input current and voltage that is not provided by the Arduino so w ned another power source.

We have the thermocouple that is our feedback sensor and a MAX6675 module to amplify and compensate the voltage from the thermocouple into the value equivalent to insert in in our PID control.

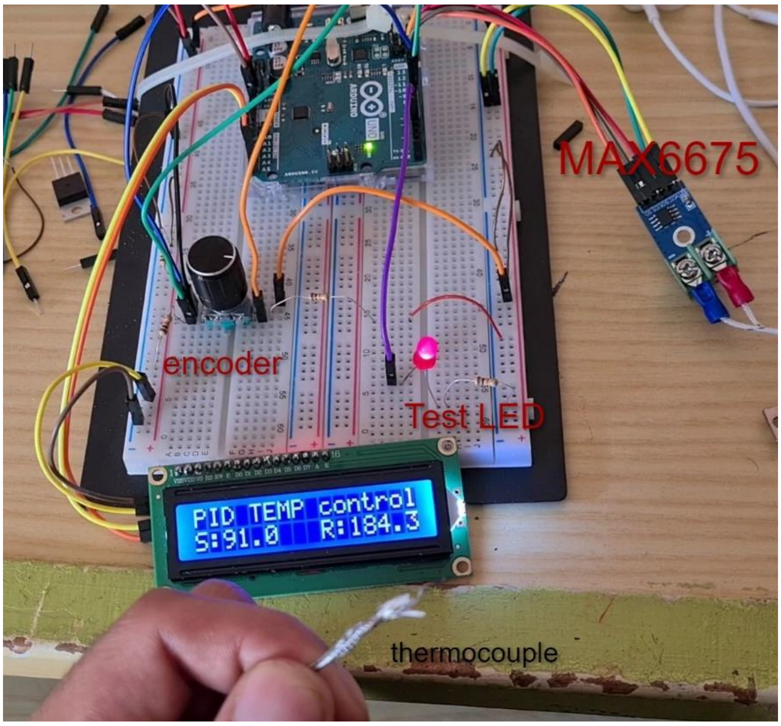

Figure 32:Bench test for PID control

An encoder is our second input that has the scoop to enter in four menus established by the program this idea is to facilitate the control of the program in other words it makes the user interact with the program in real time to modify some values because in this case they are based at the beginning on try and error to reach the desired values, one to set our desired temperature that we can't exceed ,the second to choose the KP constant suitable with our case ,the third for the KI constant and the forth for the KD constant all of that is displayed on an LCD that displays by default the current temperature from the thermocouple and the pre-set temperature value .

To get the access to the 4 menus the encoder must be pressed once for each menu then rotate the encoder to increase or decrease the wanted value.

THE PID Control Code

PID §

```
#include <SPI.h>

//LCD config
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);


//I/O
int PWM_pin = 3;  //Pin for PWM signal to the MOSFET driver (the BJT npn with pullup)
int clk = 8;      //Pin 1 from rotary encoder
int data = 9;     //Pin 2 from rotary encoder


//Variables
float set_temperature = 0;            //Default temperature setpoint.

float temperature_read = 0.0;
float PID_error = 0;
float previous_error = 0;
float elapsedTime, Time, timePrev;
float PID_value = 0;
int button_pressed = 0;
int menu_activated=0;
float last_set_temperature = 0;

//Vraiables for rotary encoder state detection
int clk_State;
int Last_State;
bool dt_State;

//PID constants
//////////////////////////////////////////////////////////
int kp = 90;   int ki = 30;   int kd = 80;
```

Figure 33:PID code part 1

```
int PID_p = 0;    int PID_i = 0;    int PID_d = 0;
float last_kp = 0;
float last_ki = 0;
float last_kd = 0;

int PID_values_fixed =0;

//Pins for the SPI with MAX6675
#define MAX6675_CS   10
#define MAX6675_SO   12
#define MAX6675_SCK  13

void setup() {
  pinMode(PWM_pin,OUTPUT);
  TCCR2B = TCCR2B & B11111000 | 0x03;    // pin 3 and 11 PWM frequency of 928.5 Hz
  Time = millis();

  Last_State = (PINB & B00000001);      //Detect first state of the encoder

  PCICR |= (1 << PCIE0);    //enable PCMSK0 scan
  PCMSK0 |= (1 << PCINT0);  //Set pin D8 trigger an interrupt on state change.
  PCMSK0 |= (1 << PCINT1);  //Set pin D9 trigger an interrupt on state change.
  PCMSK0 |= (1 << PCINT3);  //Set pin D11 trigger an interrupt on state change.

  pinMode(11,INPUT);
  pinMode(9,INPUT);
  pinMode(8,INPUT);

  lcd.init();
  lcd.backlight();
}
```

Figure 34:PID code part 2

```
void loop() {

if(menu_activated==0)
{
  // First we read the real value of temperature
  temperature_read = readThermocouple();
  //Next we calculate the error between the setpoint and the real value
  PID_error = set_temperature - temperature_read + 3;
  //Calculate the P value
  PID_p = 0.01*kp * PID_error;
  //Calculate the I value in a range on +-3
  PID_i = 0.01*PID_i + (ki * PID_error);

  //For derivative we need real time to calculate speed change rate
  timePrev = Time;                            // the previous time is stored before the actual time read
  Time = millis();                            // actual time read
  elapsedTime = (Time - timePrev) / 1000;
  //Now we can calculate the D calue
  PID_d = 0.01*kd*((PID_error - previous_error)/elapsedTime);
  //Final total PID value is the sum of P + I + D
  PID_value = PID_p + PID_i + PID_d;

  //We define PWM range between 0 and 255
  if(PID_value < 0)
  {    PID_value = 0;    }
  if(PID_value > 255)
  {    PID_value = 255;  }
  //Now we can write the PWM signal to the mosfet on digital pin D3
  //Since we activate the MOSFET with a 0 to the base of the BJT, we write 255-PID value (inverted)
  analogWrite(PWM_pin,255-PID_value);
  previous_error = PID_error;     //Remember to store the previous error for next loop.

  delay(250); //Refresh rate + delay of LCD print
  //lcd.clear();
```

Figure 35:PID code part 3

```
  lcd.print("PID TEMP control");
  lcd.setCursor(0,1);
  lcd.print("S:");
  lcd.setCursor(2,1);
  lcd.print(set_temperature,1);
  lcd.setCursor(9,1);
  lcd.print("R:");
  lcd.setCursor(11,1);
  lcd.print(temperature_read,1);
}//end of menu 0 (PID control)


//First page of menu (temp setpoint)
if(menu_activated == 1)
{
   analogWrite(PWM_pin,255);
  if(set_temperature != last_set_temperature)
  {
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Set  temperature");
  lcd.setCursor(0,1);
  lcd.print(set_temperature);
  }
  last_set_temperature = set_temperature;


}//end of menu 1


//Second page of menu (P set)
if(menu_activated == 2)
{

  if(kp != last_kp)
//Second page of menu (P set)
if(menu_activated == 2)
{

  if(kp != last_kp)
  {
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Set   P  value  ");
  lcd.setCursor(0,1);
  lcd.print(kp);
  }
  last_kp = kp;

}//end of menu 2


//Third page of menu (I set)
if(menu_activated == 3)
{

  if(ki != last_ki)
  {
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Set   I  value  ");
  lcd.setCursor(0,1);
  lcd.print(ki);
  }
  last_ki = ki;

}//end of menu 3


//Forth page of menu (D set)
```

Figure 36:PID code part 4

```
//Forth page of menu (D set)
if(menu_activated == 4)
{

  if(kd != last_kd)
  {
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Set   D  value  ");
  lcd.setCursor(0,1);
  lcd.print(kd);
  }
  last_kd = kd;
}//end of menu 4

}//Loop end


//The function that reads the SPI data from MAX6675
double readThermocouple() {

  uint16_t v;
  pinMode(MAX6675_CS, OUTPUT);
  pinMode(MAX6675_SO, INPUT);
  pinMode(MAX6675_SCK, OUTPUT);

  digitalWrite(MAX6675_CS, LOW);
  delay(1);

  // Read in 16 bits,
  //  15    = 0 always
  //  14..2 = 0.25 degree counts MSB First
  //  2     = 1 if thermocouple is open circuit
  //  1..0  = uninteresting status

  v = shiftIn(MAX6675_SO, MAX6675_SCK, MSBFIRST);
  v <<= 8;
  v |= shiftIn(MAX6675_SO, MAX6675_SCK, MSBFIRST);

  digitalWrite(MAX6675_CS, HIGH);
  if (v & 0x4)
  {
    // Bit 2 indicates if the thermocouple is disconnected
    return NAN;
  }

  // The lower three bits (0,1,2) are discarded status bits
  v >>= 3;

  // The remaining bits are the number of 0.25 degree (C) counts
  return v*0.25;
}


//The interruption vector for push button and rotary encoder
ISR(PCINT0_vect){
if(menu_activated==1)
   {
  clk_State =   (PINB & B00000001); //pin 8 state? It is HIGH?
  dt_State  =   (PINB & B00000010);
  if (clk_State != Last_State){
     // If the data state is different to the clock state, that means the encoder is rotating clockwise
     if (dt_State != clk_State) {
       set_temperature = set_temperature+0.5 ;
     }
     else {
       set_temperature = set_temperature-0.5;
     }
  }
  Last_State = clk_State; // Updates the previous state of the clock with the current state
```

Figure 37:PID code part 5

```
if(menu_activated==2)
  {
  clk_State =   (PINB & B00000001); //pin 8 state?
  dt_State  =   (PINB & B00000010);
  if (clk_State != Last_State){
     // If the data state is different to the clock state, that means the encoder is rotating clockwise
     if (dt_State != clk_State) {
       kp = kp+1 ;
     }
     else {
       kp = kp-1;
     }
  }
  Last_State = clk_State; // Updates the previous state of the clock with the current state
}


if(menu_activated==3)
  {
  clk_State =   (PINB & B00000001); //pin 8 state?
  dt_State  =   (PINB & B00000010);
  if (clk_State != Last_State){
     // If the data state is different to the clock state, that means the encoder is rotating clockwise
     if (dt_State != clk_State) {
       ki = ki+1 ;
     }
     else {
       ki = ki-1;
     }
  }
  Last_State = clk_State; // Updates the previous state of the clock with the current state
}

 if(menu_activated==4)

 if(menu_activated==4)
  {
  clk_State =   (PINB & B00000001); //pin 8 state?
  dt_State  =   (PINB & B00000010);
  if (clk_State != Last_State){
     // If the data state is different to the clock state, that means the encoder is rotating clockwise
     if (dt_State != clk_State) {
       kd = kd+1 ;
     }
     else {
       kd = kd-1;
     }
  }
  Last_State = clk_State; // Updates the previous state of the clock with the current state
}



  //Push button was pressed!
  if (PINB & B00001000) //Pin D11 is HIGH?
  {
    button_pressed = 1;
  }
  //We navigate through the 4 menus with each button pressed
  else if(button_pressed == 1)
  {

   if(menu_activated==4)
   {
    menu_activated = 0;
    PID_values_fixed=1;
    button_pressed=0;
    delay(1000);
   }
```

Figure 38:PID code part 6

```
    {
     menu_activated = menu_activated + 1;
     button_pressed=0;
     kd = kd + 1;
     delay(1000);
    }

    if(menu_activated==2)
    {
     menu_activated = menu_activated + 1;
     button_pressed=0;
     ki = ki + 1;
     delay(1000);
    }

    if(menu_activated==1)
    {
     menu_activated = menu_activated + 1;
     button_pressed=0;
     kp = kp + 1;
     delay(1000);
    }


    if(menu_activated==0 && PID_values_fixed != 1)
    {
     menu_activated = menu_activated + 1;
     button_pressed=0;
     set_temperature = set_temperature+1;
     delay(1000);
    }
    PID_values_fixed = 0;

   }
 }
```

Figure 39:PID code part 7

This code works but not very effective due to the high latency of the thermocouple wire respect of the rise time of the temperature of the SMA wire. (It is more recommended to use a high efficiency thermocouple but is has a high cost).

The program was tested on LED to control its efficiency and it was very accurate and response.

Continuous loop Control

For the other two types of control due to the current and voltage needed for the wire an interface circuit between the control kit and the prototype was designed with a special transistor N-type that has two bjt transistors (Darlington Transistor) the first provide additional gain and improved current capability, and internal resistance and a security diode for any reverse current.

It can resist up to 80W and has an Emitter-Base voltage as 5 volts to open the gate between the collector and the emitter to turn on the transistor that is provided by our control signal.

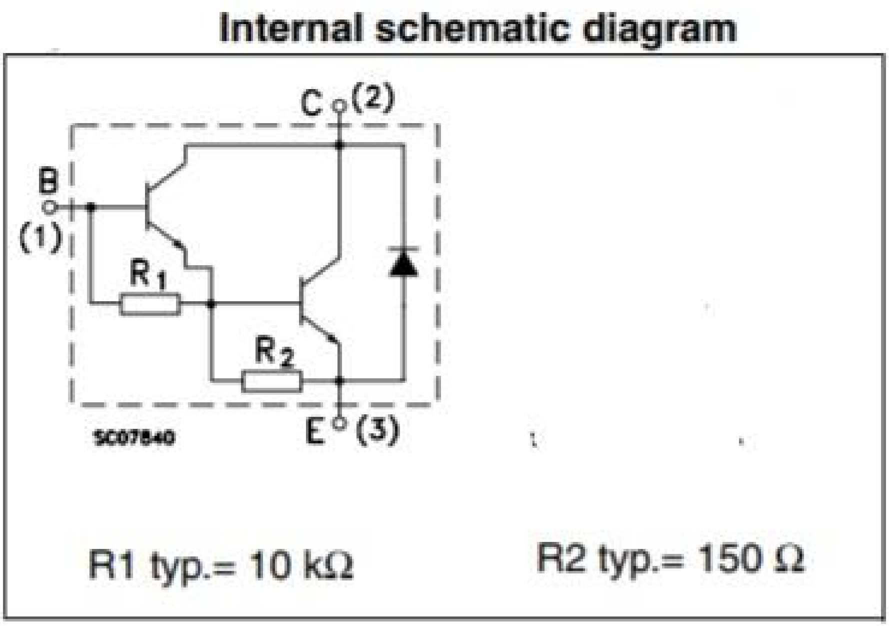

Figure 40:Tansistor internal schematic

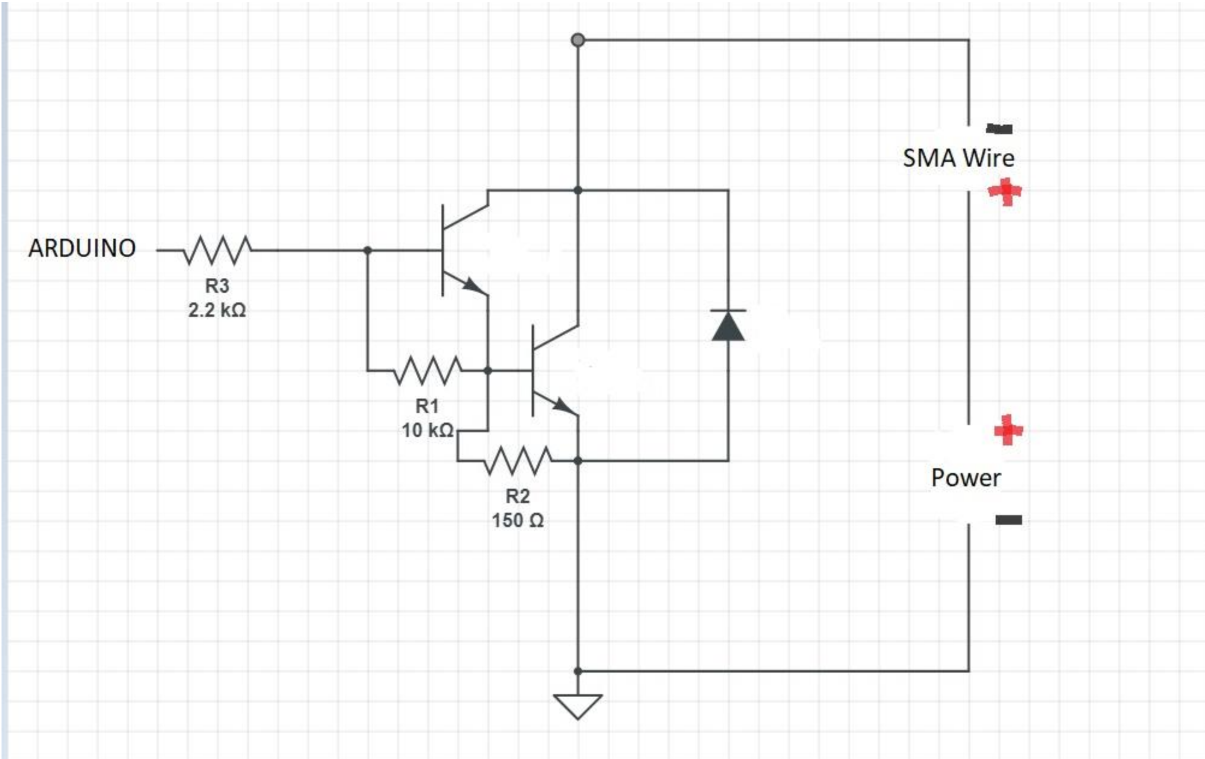

Figure 41:Interface circuit between the controller and the prototype

The circuit above is the interface used, test and works correctly with our purpose.

2. Continuous Loop actuation using temperature sensor

Here the control is by setting a temperature that we do not exceed that is the austinite final temperature 90°C reaching this temperature means that we had arrived at our full stroke so that mean that we need to cut off the power but we want to make this process as a continuous loop so an IR sensor was established in the position of the red dot in (figure 44) to ensure that the wire has return to is initial position and that is has been cooled to provide the power to the wire once again in this way it will run in a continuous loop.

This simple scheme represents a simple idea about the control and the flow of signal, power and data (figure 42)

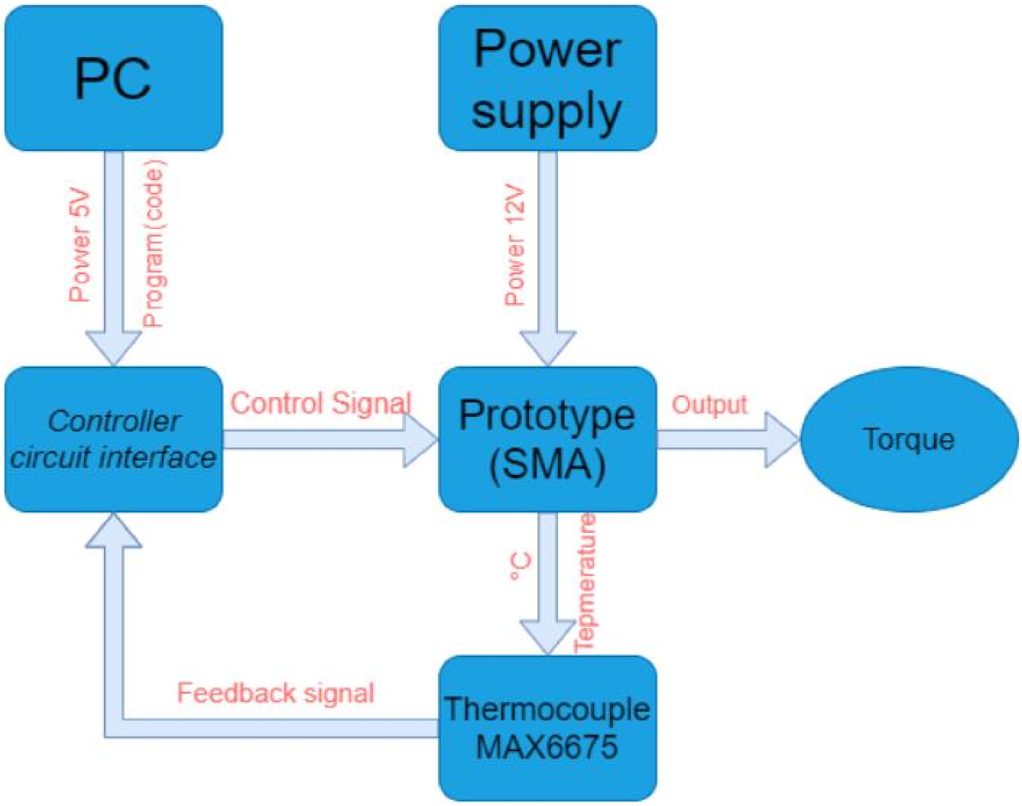

Figure 42;Scheme of temperature control

The testing bench was composed by the power supply that its negative side is connected to the ground of all the circuit and the positive side with one side of the SMA wire, the other side of the

SMA wire was connected to the collector of the transistor and the interface circuit connected with the Arduino that has the thermocouple module installed then getting the tip to the thermocouple very near to the wire but without touching it to avoid short circuit with the MAX6675 module and a LCD to display the current temperature .

The Arduino was connected to the PC to provide it with power and to burn the code on it.

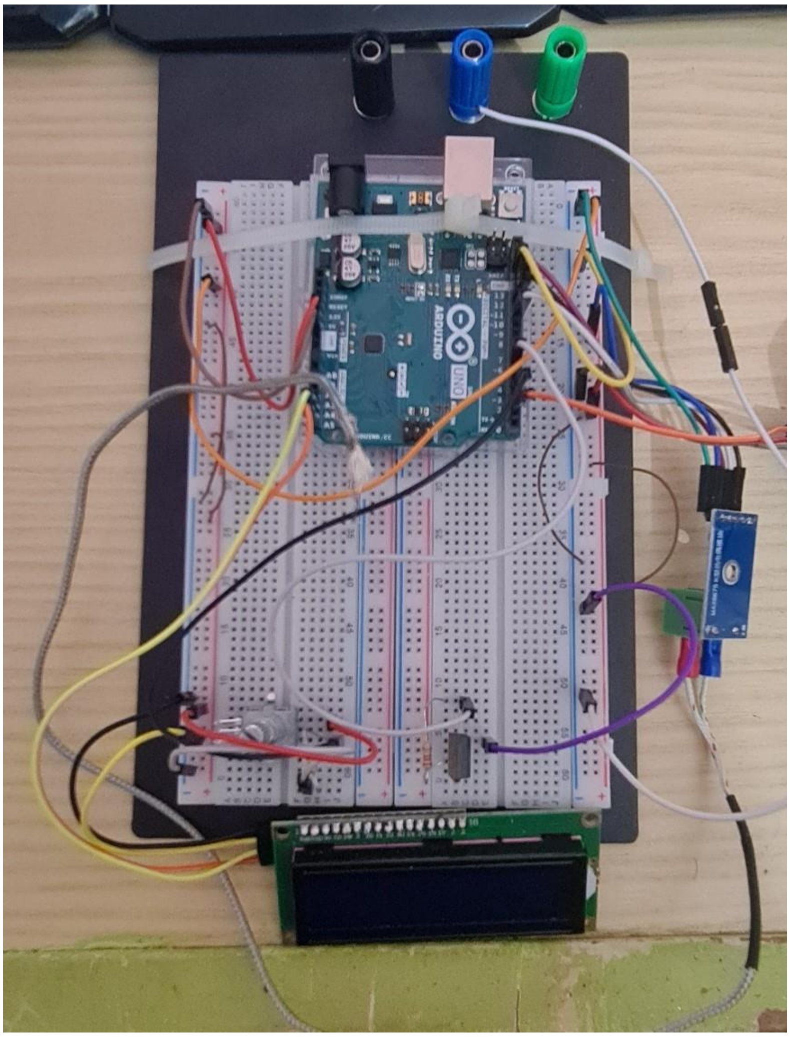

Figure 43:test bench of continuous loop temperature control

It was able to power up the SMA wire based on its temperature so the prototype is alimented by a power of 8W if it is under or equal 90°C then it will turn off until the wire is back to its original position to be sure that the wire is cooled to get the maximum efficiency.

The original position is cheeked by an IR sensor, the experiment was done several times to see the repeatability of the wire experiment and it was an acceptable result.

The experiment was recorded in a video to document the process of the experiment.

The only problem was that our K thermocouple has a big latency that means when the wire reaches its 90°C our thermocouple is still in the range of 40 to 45°C so we used this range in our program as an indication to this 90°C due to the speed of the wire in reaching this temperature that do not exceed 2 seconds.

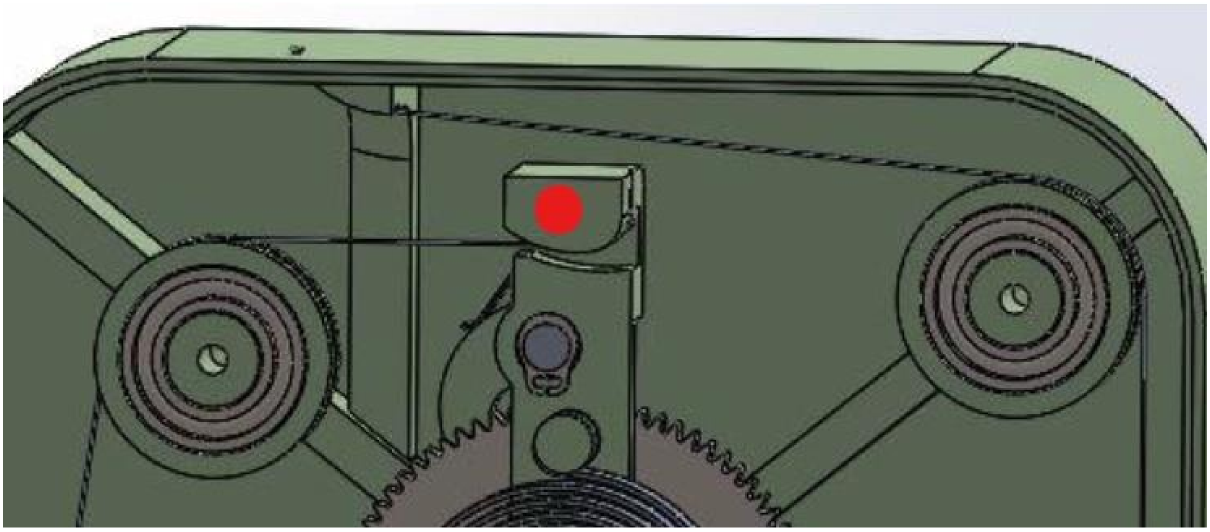

Figure 44:IR sensor position

THE COMPONENTS:

- IR sensor
- Arduino uno
- I2C LCD display
- Push button
- Power supply
- BDX53C transistor
- 2.2K resistance

THE PROGRAM CODE:

```
//LCD config
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);

/*    i2c LCD Module  ==>   Arduino
 *    SCL             ==>     A5
 *    SDA             ==>     A4
 *    Vcc             ==>     Vcc (5v)
 *    Gnd             ==>     Gnd      */

#include <SPI.h>

#define MAX6675_CS   10
#define MAX6675_SO   12
#define MAX6675_SCK  13
int sensor = 4;
int led=7;

boolean flag =true;
void setup() {
  lcd.init();
  lcd.backlight();
  pinMode(sensor,INPUT);
  pinMode(led,OUTPUT);
  int sens=0;
}

void loop() {

  float temperature_read = readThermocouple();
  lcd.setCursor(0,0);
  lcd.print("TEMPERATURE");
  lcd.setCursor(7,1);
  lcd.print(temperature_read,1);
```

```
  lcd.print(temperature_read,1);
  delay(300);

  //if(digitalRead(sensor)==HIGH){
 int sens=digitalRead(sensor);
  //delay(5);
   //flag = !sens;

//&& sens == HIGH

   if (temperature_read <90 && sens==HIGH ){
    //flag=HIGH;
    digitalWrite(led,HIGH);


   }

   else {
    //flag= false ;
    digitalWrite(led,LOW);


  }

 }



double readThermocouple() {

  uint16_t v;
  pinMode(MAX6675_CS, OUTPUT);
  pinMode(MAX6675_SO, INPUT);
  pinMode(MAX6675_SCK, OUTPUT);
```

Figure 45:Temperature continues loop code part 1

```
  digitalWrite(MAX6675_CS, LOW);
  delay(1);

  // Read in 16 bits,
  //  15    = 0 always
  //  14..2 = 0.25 degree counts MSB First
  //  2     = 1 if thermocouple is open circuit
  //  1..0  = uninteresting status

  v = shiftIn(MAX6675_SO, MAX6675_SCK, MSBFIRST);
  v <<= 8;
  v |= shiftIn(MAX6675_SO, MAX6675_SCK, MSBFIRST);

  digitalWrite(MAX6675_CS, HIGH);
  if (v & 0x4)
  {
    // Bit 2 indicates if the thermocouple is disconnected
    return NAN;
  }

  // The lower three bits (0,1,2) are discarded status bits
  v >>= 3;

  // The remaining bits are the number of 0.25 degree (C) counts
  return v*0.25;
}
```

Figure 46:Temperature continues loop code part 2

3. Continuous Loop actuation using position sensor

Here the control is by putting an optical encoder at the end of the transmission shaft after a certain degree of rotation is an indication of our full stroke of our wire that is 45° that corresponds to 10 teeth of the small gear and our encoder disc has 20 slots so after 3 slots that are 3 pulses we need to cut off the power from the SMA wire and to make sure that the wire has returned to its position or we calculate the average time of cooling or by the same IR sensor used above so then we get a continuous loop.

This simple scheme represents a simple idea about the control and the flow of signal, power and data (figure 47).

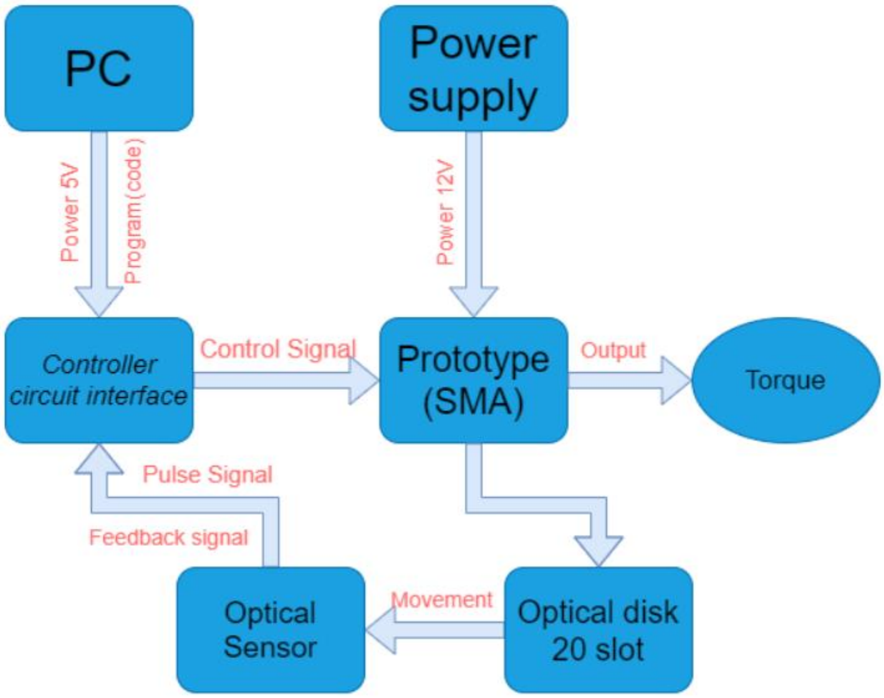

Figure 47:simple scheme of position control

There is a piece made for attaching the encoder disc with the transmission shaft of our prototype (figure 51) and testing the program it was a failure, so it was decided to write a new code to be compatible with our needs and design, the test bench is similar to the previous one but instead we use the optical sensor and the disc as our sensors .

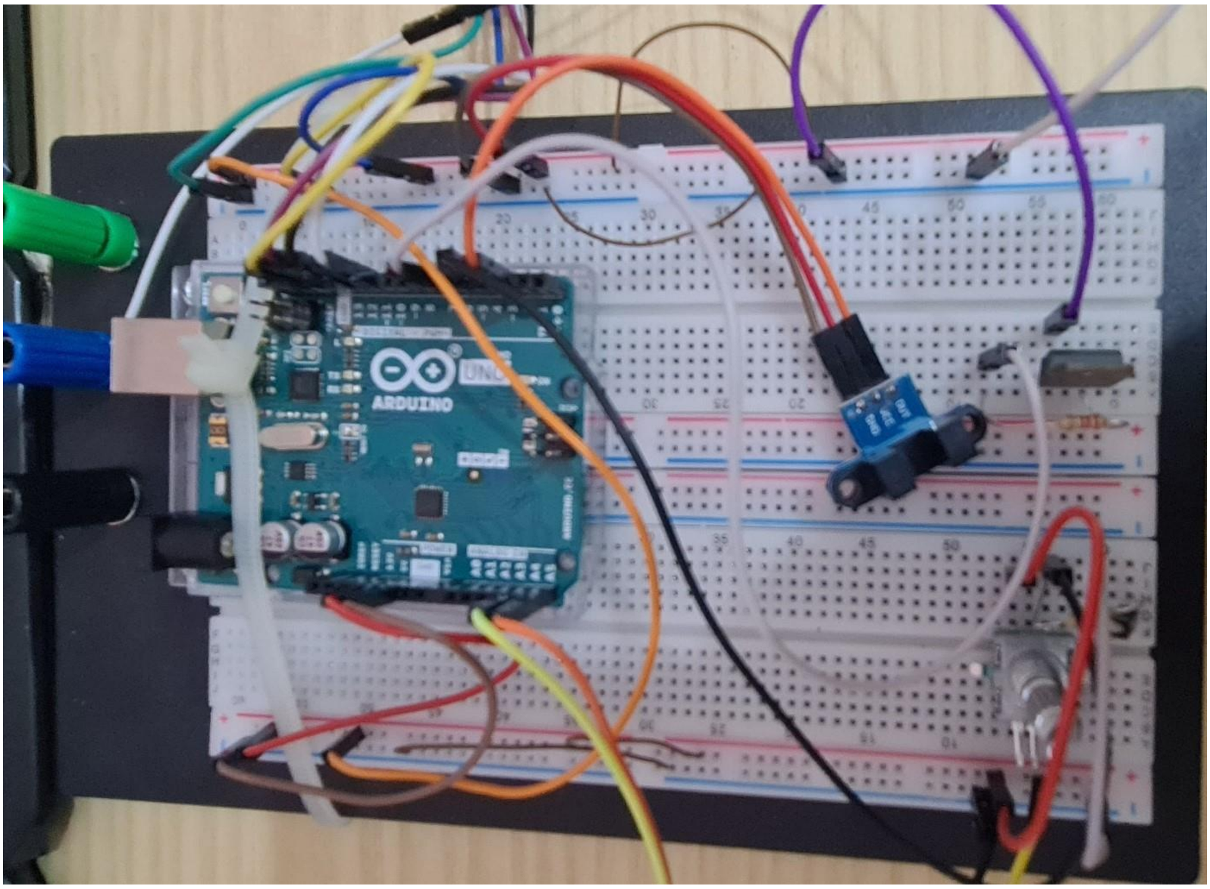

Figure 48:Test bench of position control

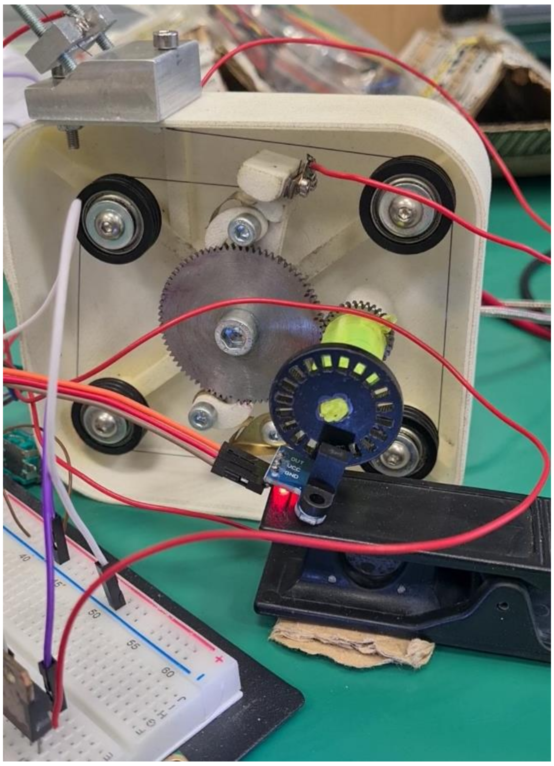

Figure 49:Test bench position control (initial position)

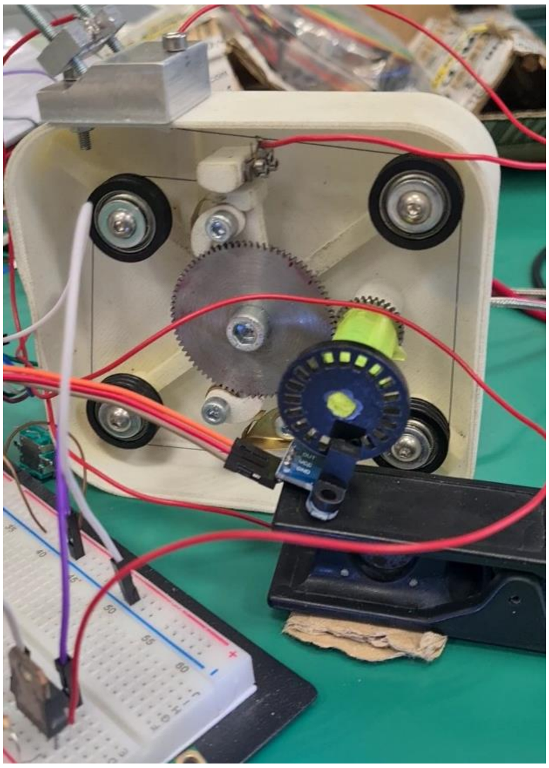

Figure 50:Test bench position control (final position)

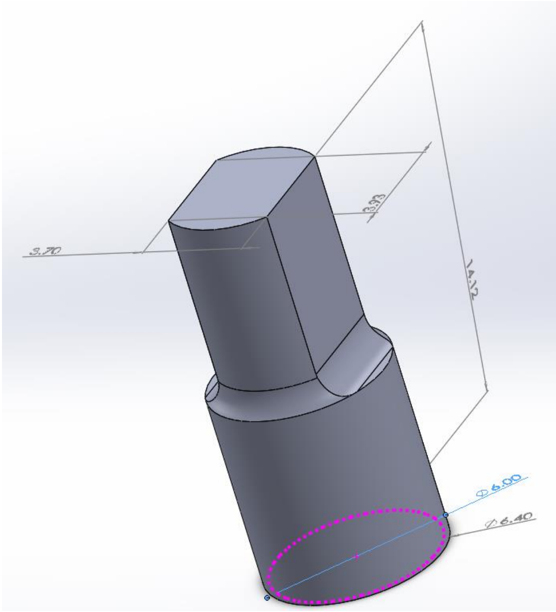

Figure 51:pace made to attach the encoder disc withe the transmission shaft

The program was able to power up the wire with our needs for our full motion we need to count 3 slots out of 20 to get our full motion angle that is 45° and the experiment was done several times to check the repeatability and it was recorded in a video.

An experiment with 2 slots of motion was done too but it was not a full motion process and it was observed that in this case we need to rise up the value of the force out from the torsion spring to be able to bet the wire back to its original position .

THE PROGRAM CODE:

```
volatile unsigned int counter = 0;

void setup() {
  Serial.begin (9600);
  pinMode(2, INPUT_PULLUP);    // internal pullup input pin 2 for encoder input A - green wire
  pinMode(4, INPUT);
  pinMode(6, OUTPUT);


//Setting up interrupt
  //A rising pulse from encodenren activated ai0(). AttachInterrupt 0 is DigitalPin nr 2 on moust Arduino.
  attachInterrupt(0, ai0, RISING);
  }

  void loop() {

  //Serial.println (counter);
  if (counter >=3) {
    digitalWrite(6,LOW);
    counter = 0;
    delay(10000);
  }
 else {
  digitalWrite(6,HIGH);
 }
  }

  void ai0() {
  // ai0 is activated if DigitalPin nr 2 is going from LOW to HIGH
  counter++;

  }
```

Figure 52:Position Control Code

Observations:

Continues loop temperature control:

It was noticed that there is a continues variation in the current needed from the power supply to aliment the wires over the time that has an average between [0.808:0.943] Amper in the same loop if we cut of the power totally and made the system rest then power it up again it vary in 0.02% from this results that are shown in this graphs (figure 53,54)

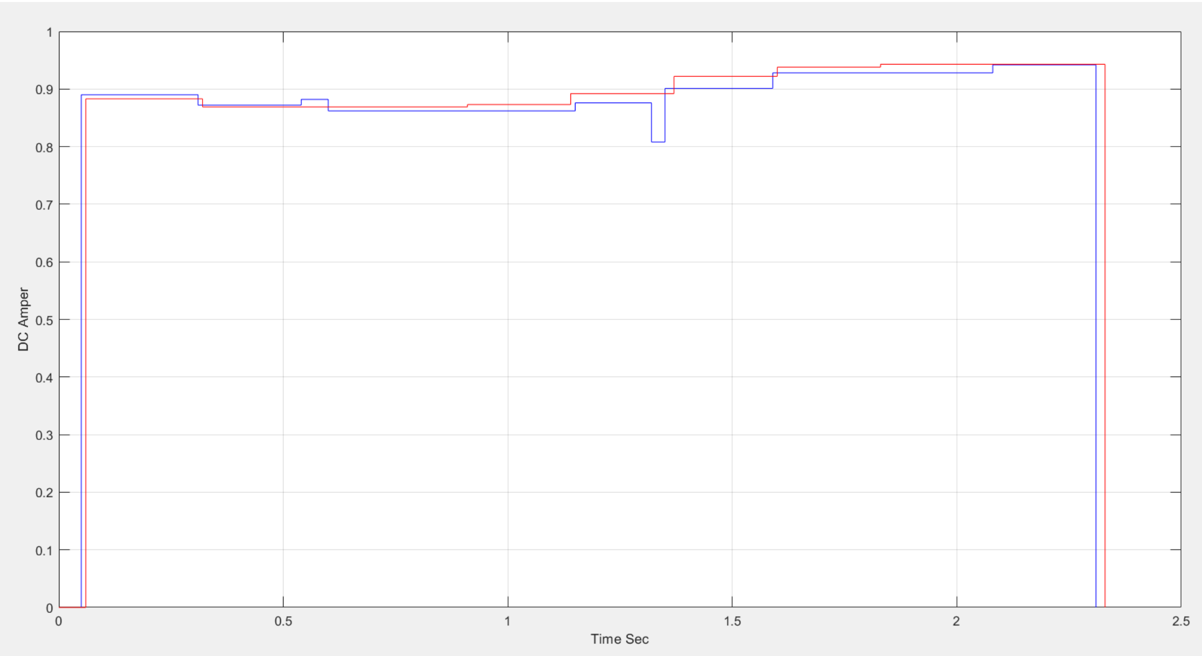

Figure 53:Current variation over time

The blue line represents the first deformation and the red on the next consecutive deformation of the same loop. There was a constant latency of 0.05 sec before each cycle.

It was also calculated that it has an average consumption of power 420,765 Wmin = 25.24 kWh with a cost in Italy of 0,160259 for kWh 4.045€ for an hour

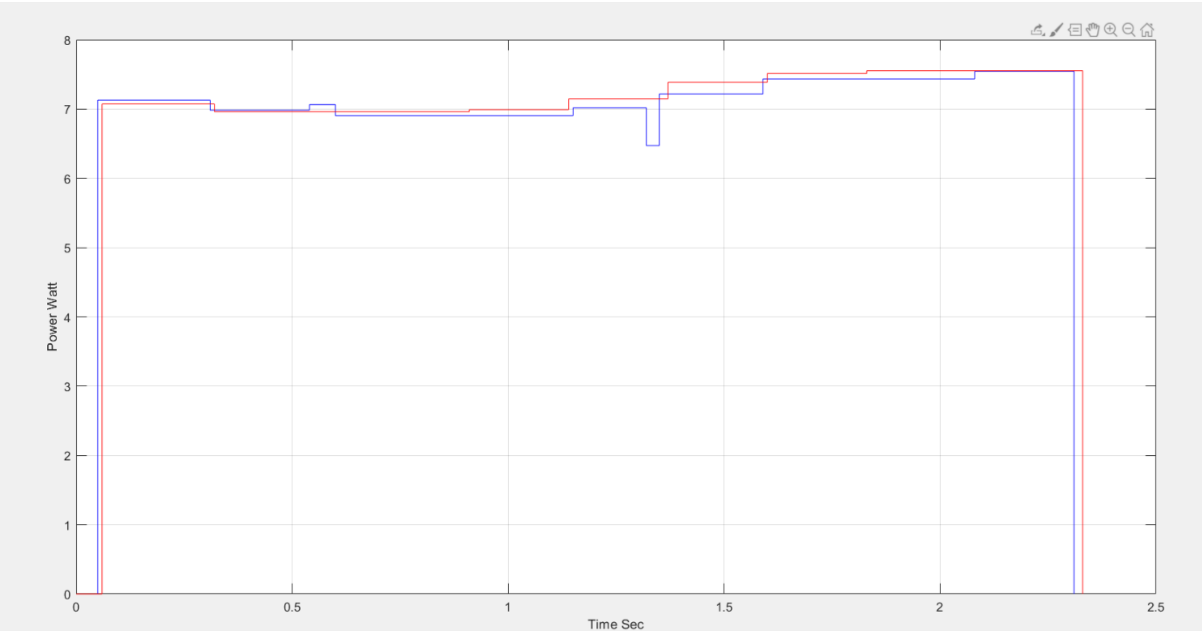

Figure 54:power consumption of temperature control

Type	Time(Sec)
First cycle rise time	2.26
second cycle rise time	2.27
third cycle rise time	2.29
forth cycle rise time	2.26
fifth cycle rise time	2.30
First cycle cool time	9.45
second cool rise time	9.33
third cycle cool time	9.64
forth cycle cool time	9.51
fifth cycle cool time	9.66

Table 3:Rise and cool time of temperature control

So as seen from the table the average rise time is almost the same and it is a very similar speed to the tested one without any control functions but it takes lot of time to get cooled (but it gets almost the same values in each cycle).

This type of control enables us to get an angle of 33.75 degree on the transition shaft that is acceptable.

Continues loop position control:

It was noticed that there is a continues variation in the current needed from the power supply to aliment the wires over the time specially at the end of the cycle because the controller forces the device to get 3 slots of the optical disc so it will not power of until it gets 3 full pulses from the optical sensor so it needs more power for this additional force, due to that it was able to reach 36 degree that is 2.25 degree above than the Temperature control

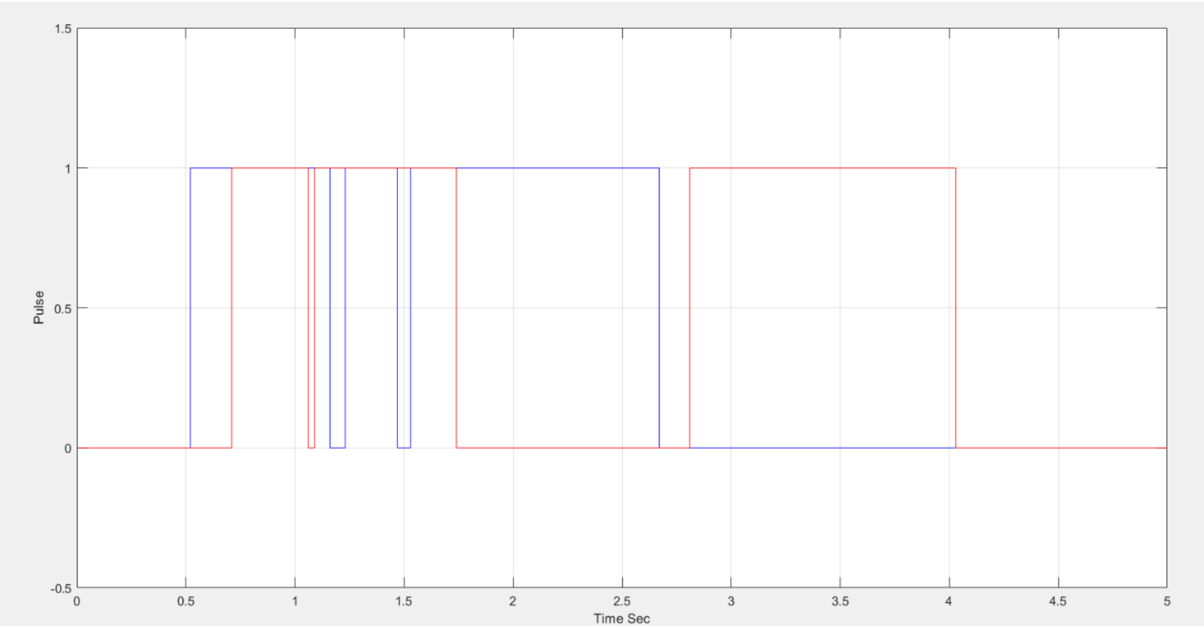

Figure 55:pulses generated from optical sensor for two consecutive cycles

The blue one is the first cycle and the red one for the second one there was a latency in the system of 0.3 sec before the system begins to respond to the commands and for each next cycle it takes more time than the previous one due to the extra stress applied to reach 36 degree from the previous cycle

Type	Time(Sec)
First cycle rise time	2.37
second cycle rise time	3.73
third cycle rise time	3.84
First cycle cool time	6.69
second cool rise time	6.95
third cycle cool time	6.9

Table 4:Rise and cool time of position control

It can be seen that the position control was able to get the maximum of the SMA wires with 36 degree but with a slow rise time for each cycle and faster cooling time ,the temperature was more faster in the rise time category but more slower in the cooling time but with 33.75 degree.

Conclusion

At the end the main objective of this project was to control the prototype (SMA) wires in order to get a continuous automatic output from the transition shaft and to know the behaviour of the SMA wire when current passes through it.

It was able to accomplish three control methods PID control to maintain the SMA wire temperature under control of a certain temperature, continuous loop temperature control and a continuous loop position control.

Results are that the device was able to do the continuous loop of both the control methods, but the wire behave in different ways according to speed, cooling time and angle of movement.

In the future on the short term to redo the tests with a more efficient thermocouple module that has a low latency to be able to track the temperature of the wires in real time without delays, change the torsion spring to correspond the number of wires (the force need to get back the wires to there original form) ,lubricate the pulleys to make their movement more smooth and trying to attach a fan to get a forced cooling by the air to reduce the cooling time of the wires.

On the long term it is proposed to redesign the prototype body with another material such as metal that will help the device to be more robust and it can help the wires to cool down faster by convection temperature transition, to make the body more longer to accept a more longer wire so as a consequence a more longer deformation.

It can also have different paths for the wires by putting other pulleys in different positions and to get then attached to the same lever to have as an output different angels for different paths.

References

[1] Wilkes, Kenneth E.; Liaw, Peter K.; Wilkes, Kenneth E. (October 2000). "The fatigue behavior of shape-memory alloys". JOM. 52 (10): 45–51

[2] K. Otsuka; C.M. Wayman, eds. (1999). *Shape Memory Materials* Cambridge University Press.

[3] Duerig, T.W.; Pelton, A.R. (1994). "Ti-Ni shape memory alloys". In Gerhard Welsch; Rodney Boyer; E.W. Collings (eds.). *Materials Properties Handbook: Titanium Alloys*. American Society for Metals. pp. 1035–48.

[4] H. Sehitoglu1, R. Hamilton, H.J. Maier and Y. Chumlyakov(2004) 'Hysteresis in NiTi alloys' pp 5

[5] Hui Qian, Hongnan Li, Gangbing Song and Wei Guo (2003) 'Recentering Shape Memory Alloy Passive Damper for Structural Vibration Control'

[6] Lisa Case, Zachary Kreiner ,John Redmond ,Brian Trease (2004) 'Shape Memory Alloy Shape Training Tutorial'

[7]https://www.phidrive.eu/en/prodotti-phidrive-attuatori-rotativi-ad-alta-precisione/

[8] Lan e J.H. Wang (2009), '*Optimal design of rotary manipulators using shape memory alloy wire actuated flexures'*

[9] Luca Amicucci (2017) PROGETTAZIONE E REALIZZAZIONE DI UN PROTOTIPO DI MOTORE ROTATIVO AZIONATO MEDIANTE SHAPE MEMORY ALLOY (SMA)