# Laboratory 8: Discrete control, position control of a mass using Microcontroller

Galvis, David. Lopez, Daniel.

{u1802584,u1802530}@unimilitar.edu.co

Universidad Militar Nueva Granada

*Abstract---* For this laboratory practice, the transfer function of a plant is found, for a system that allows the position control of a mass, using a hoist reduction of ¼ of the force exerted by the motor. The main objective is to control by means of a PID, design and discretize the controller, and implement it in a microcontroller, which in this case will be the Arduino UNO platform. The position of the mass, the reference, the control signal and the error signal are displayed on a graphical user interface of the software.

*Abstract---* For this laboratory practice, the transfer function of a plant is found for a system that allows control of the position of a mass, using a hoist with a ¼ reduction of the force exerted by the motor. The main objective is to control by means of a PID, design and discretize the controller, and implement it in a microcontroller which for this case will be the Arduino UNO platform. The position of the mass, the reference, the control signal and the error signal are displayed in a graphical user interface of the software.

*Keywords---* Discretization, Microcontroller, Digital Control, Graphical User Interface, Serial Communication.

*General Objective---* Design and build a system that ensures the position of a mass anchored to a hoist system through the use of microcontrollers.

*Specific objectives--*

* Design a hoist system that allows the ascent and descent of a mass, this must have a DC motor as actuator.

* Identify the mathematical model that represents the functioning of the prototype.

* Design a discrete PID that ensures the position of the mass.

* Implement the discrete controller by using a microcontroller, selecting the reference through a user interface and displaying the system status graphically.

## 1. INTRODUCTION

**Theoretical framework:**

Within control processes, most of the time, their integration into the real world is done through digital systems. The current advance of low-cost processing systems has made it possible to have a discrete control system in almost any process.

The progress in the miniaturization of processing systems, the improvement of their speed in executing instructions, today allow the implementation of controls in most of the systems that interact in our environment. When seeking to program a discrete control in microcontrollers, computers, PLC's, it is necessary to proceed from an analysis and interpretation of the behavior of the system to be controlled.

For the approach of the difference equation (the one that allows integrating the discrete control to a processing system), one can start from a continuous design and apply some discretization technique for the control found, or discretize the system model and design in the discreet world control.

For the first method, any of the steps that have been discussed in prior practices or known continuous control theories are followed. With the continuous control fulfilling the design criteria, its discretization is carried out and from this discrete equation (in Z), the difference equation is proposed. In this procedure, it is desirable that the method used for discretization maintains most of the temporal and frequency characteristics.

There are several theories that allow finding the discrete representation. One can be that *s* transforms to $\frac{1-z^{-1}}{t_m}$, where $t_m$ is the sampling time. It is important for the above equation to select a good sampling time. In the literature it is found that *tm* is related to the kind of system to be controlled, with the response time or its bandwidth.

For the second method, the first step is to find a discrete model of the system and from this, apply one of the many ways that exist to calculate the coefficients of a discrete regulator. The design of a discrete PID regulator, whose equation is represented by $\frac{q0+q1Z^{-1}+q2Z^{-1}}{1-z^{-1}}$, can be faced by the pole assignment technique [1].

Fast stabilization systems require fast response control systems and, in turn, require electronics or mechanical actuators suitable for this dynamic. Within the theory necessary for the design of this type of control, there is an adequate selection of the sampling time *tm*. The generation of the control signal involves several steps: the acquisition of the output variable (sensed signal), conditioning or filtering of it, calculation of the control signal and its output (conversion from digital to analog) must be done. . The driver or actuator is an important element in the integration of the control signal and the system.

**What is a microcontroller?**

A microcontroller is an integrated circuit that contains a central processing unit (CPU), memory units (RAM and ROM), input and output ports, and peripherals. These parts are interconnected within the microcontroller, and together they form what is known as a microcomputer. It can be properly said that a microcontroller is a complete microcomputer encapsulated in an integrated circuit.

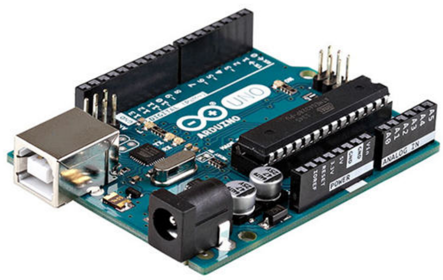


*Figure 1: Arduino platform with ATMEL microcontroller.*

2. MATERIALS

* White coat.

* Plant: DC motor, support, pulleys, rope, mass, linear potentiometer.

* Digital control: National Instruments DAQ acquisition card, Labview 2015.

3. PROCEDURE

Solution **Questionnaire**:

*How is serial between a microcontroller and a PC made and implemented?*

A- To perform serial communication between a PC and a microcontroller, a communication channel known as COM ports is configured, which allow this to be done, there are various software that allow creating a user interface when performing serial communication, such as the case of Matlab, LabView, Netbeans, C++ among others.

*How is the storage of the data received and sent by the port guaranteed, such that they subsequently allow the state of a system to be reconstructed?*

It can be guaranteed by storing the output data in a variable, and that it corresponds to a sampling time in which that output elapsed or was presented.

### Practice

then have that the transfer function of the system plant with a hoist is as follows:

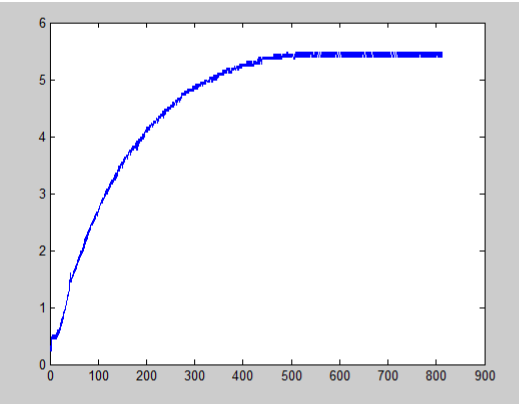


*Figure 2: Response of the plant in closed loop*

This is how the graph of the response of the data obtained by the excel file of the oscilloscope is obtained.

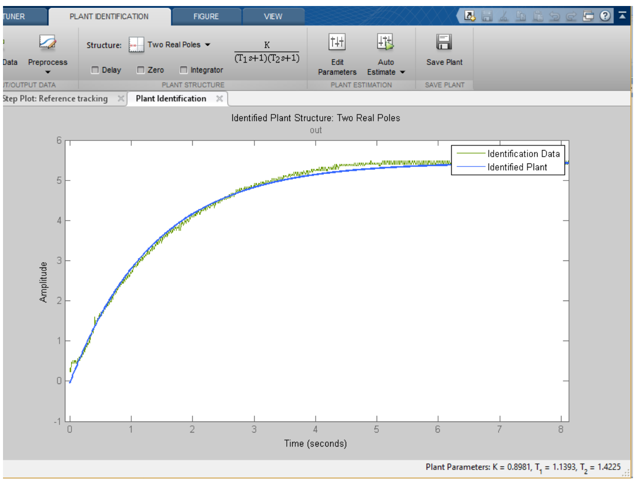



*Figure 3: Approximation of the transfer function of the plant*

It was obtained using the "identify new plant" tool of the Matlab command window tool "pidtool".

Approximate the result obtained from the oscilloscope, to a transfer function of a plant that behaves the same, for this case

$$\text{Transfer function} = \frac{0.8981}{(1.1393s+1)(1.4225s+1)} = \frac{0.5541}{s^2+1.5807s+0.6170}$$

The simulation carried out in MATLAB by block diagram is shown in the following figure:

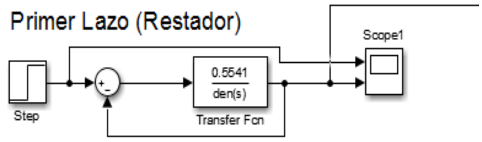



*Figure 4: Simulation of the transfer function in closed loop*

whose response is evidenced in **figure 5.**

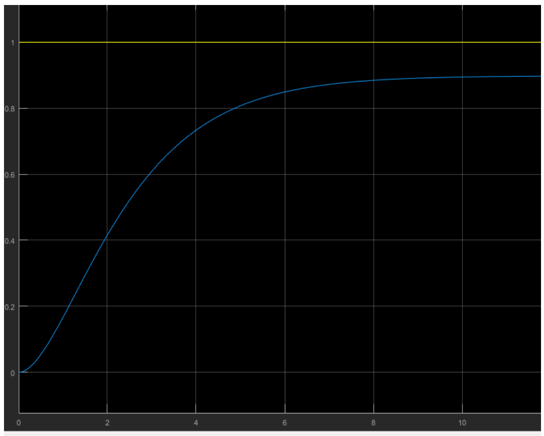


*Figure 5: Response in closed loop of the plant*

For the design of the continuous controller it is observed that the system is of type 0, second order, so it is decided to implement a PID; such that the system presents the following parameters:

- Establishment time = 10s
- Coef. damping = 0.8
- Stabilization criterion = 2%

With this, a desired polynomial is sought:

$$s^3 + 4.8s^2 + 3.45s + 1$$

And consequently the characteristic polynomial of the controlled system:

$$1 + (kp + \frac{Ki}{s} + kds) * (\frac{0.5541}{s^2+1.587s+0.617}) = 0$$

$$s^3 + (1.587 + 0.5541 * kd)s^2 + (0.617 + 0.5541 + kp)s +$$

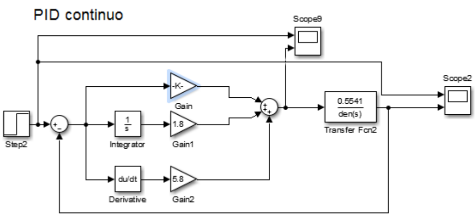
Equating the characteristic polynomial with the desired one and clearing the controller constants gives that:

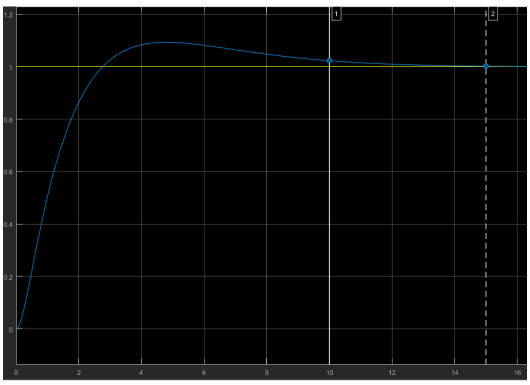
$$Kp = 6.11$$

$$Ki = 1.8$$

$$Kd = 5.8$$

Simulating the results, what is shown in **figure 6 is**.



| | Time | Value |
|---|---|---|
| 1 | 10.000 | 1.023e+00 |
| 2 | 15.000 | 1.002e+00 |

*Figure 6: Response of the system with a controller in continuous time*

Once the correct operation of the controller designed in continuous has been verified, it is proceeded to discretize, it is decided to use the Euler method in delay where the given equivalence is the following:

$$s = \frac{1-z^{-1}}{tm}$$

For the sampling time to be used, it is decided to implement the tr/30 criterion, with this:

$$tr = \frac{\pi-\theta}{Wd}$$

$$Wd = Wn\sqrt{1 - \xi^2}$$

$$\theta = Cos^{-1}(\xi)$$

This gives us a sampling time of 0.277 s

Continuing with the discretization of the control, it is given that the corresponding equivalences are:

$$q0 = Kp + Ki * tm + \frac{Kd}{tm}$$

$$q1 =- Kp - \frac{2Kd}{tm}$$

$$q2 = \frac{Kd}{tm}$$

So that in discrete controller of general form is as follows:

$$Gc(z) = \frac{q0+q1*z^{-1}+q2*z^{-2}}{1-z^{-1}}$$

For the case presented, it follows that:

$$q0 = 27.54$$

$$q1 =- 47.98$$

$$q2 = 20.93$$

Once the constants of the controller have been obtained, the resulting recursive equation is implemented in the microcontroller.

```
int pinRef=0;
float Referencia=0;
float Error=0;
float Error1=0;
float Error2=0;
float Control=0;
float Control1=0;
float q0=27.54;
float q1=-47.98;
float q2= 20.93;
float tm=277;
int pinCont=3;
int pinSen=4;
float Sensor=0;
float Salida=0;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}
```

*Figure 7: Declaration of pins and variables*

```
void loop() {
  // put your main code here, to run repeatedly:
  Referencia=analogRead(pinRef)*5.0/1023.0;
  Sensor=analogRead(pinSen)*5.0/1023.0;
  Error=Referencia-Sensor;
  Control=(q0*Error)+(q1*Error1)+(q2*Error2)+Control1;
  Control1=Control;
  Error1=Error;
  Error2=Error1;
  if(Control>=5){
    Control=5;
   }else{
    if(Control<=0.0){
      Control=0.0;
      }
      else{}
    }
  analogWrite(pinCont, (Control*51));
  //Serial.print(Referencia);
  //Serial.print(",");
  Serial.println(Sensor);
  Serial.print(",");
  Serial.print(Error);
  Serial.print(",");
  Serial.print(Control);
  Serial.println();
  delay(tm);
}
```

*Figure 8: Main structure of the program*

As can be seen, the reference value is determined by an analog voltage signal coming from a potentiometer connected to pin A0 of the board.

The interface designed in the LabView software is as follows:

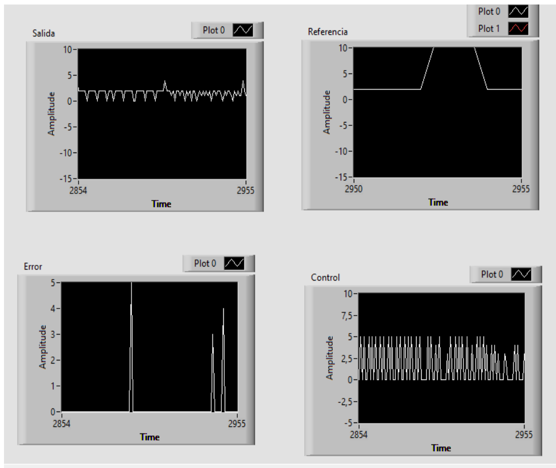



*Figure 9: NI LabView user interface*

Where the functional block diagram is shown in the following figure:

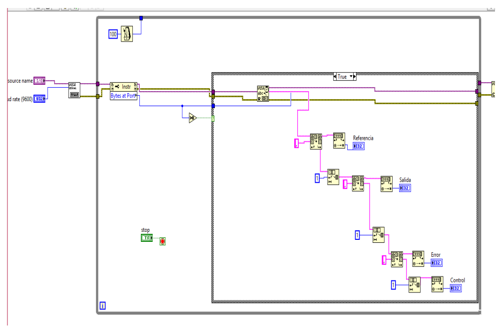


*Figure 10: Block diagram implemented in LabView*

From left to right, the operation is asnext:

- The serial port to which the Arduino board is connected is identified, the baud rate to work is determined.
- The communication is recognized and the transmission and reception channel parameters are obtained.
- arrives at the serial reading block **figure 10,** where it is broken down into the 4 required values, through a search block and segmentation of character strings and transformation of strings to decimal numerical values, which are subsequently displayed.
- Communication is closed if the work cycle is stopped.

Now, for the second part of this practice, a controller is requested from discrete time, for this, the work plant is first discretized, for this the delayed euler method was re-implemented, with a sampling period of 0.45 seconds, resulting in the following:

$$\frac{0.5541}{s2^{1.587s}+0.617}=\frac{0.04478\,z+0.03532\,z2}{+^{-1.481}z0.4896}$$

Comparing both sides of the equivalence gives us what is shown in the following image:

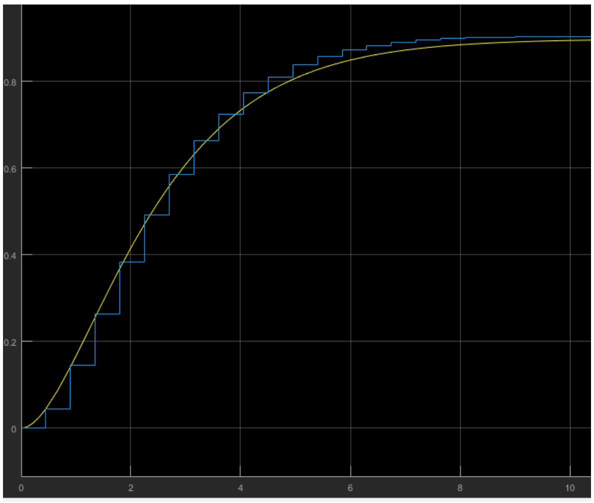


*Figure 11: Continuous system compared to discrete system.*

4. ANALYSIS OF RESULTS

- From what corresponds to the continuous design, for its subsequent discretization, it is shown that the controller in the S domain, as has been verified in previous practices and simulations, responds adequately and meets the criteria for which it was designed, This is verified by looking at **figure 6,** where the output signal reaches an error of 2.3% in a time of 10 seconds. The control and error signal presented in the controlled system are as follows:

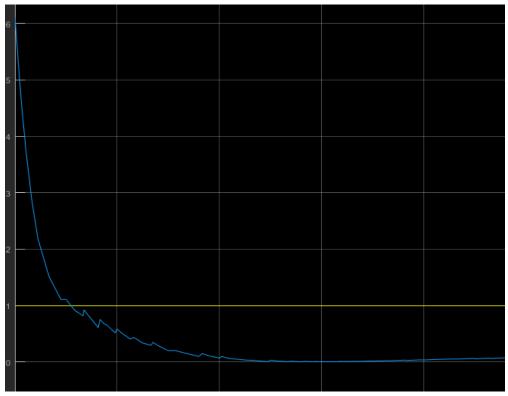
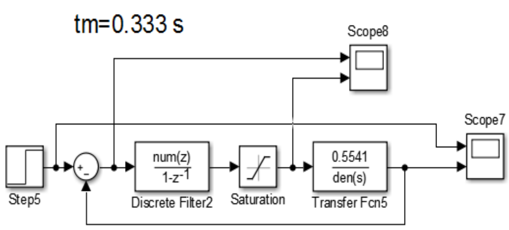


*Figure 12: Control signal and error in continuous time.*

- For the discretization of the controller, the following block diagram is implemented with its respective response.



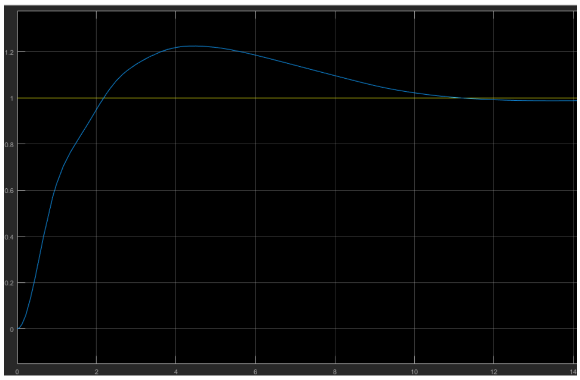



*Figure 13: Response of the discrete system.*

The observed response presents certain errors with respect to the continuous signal, this may occur due to the selection of the sampling time, the discretization method and, consequently, the coefficients of the filter to be implemented. The error signal and the control signal are presented below:

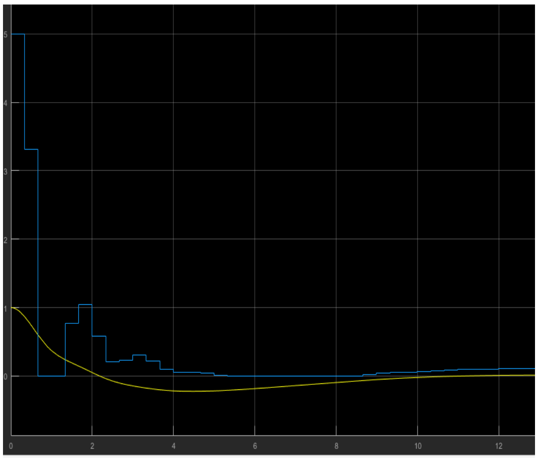


*Figure 14: Error and control signal in discrete time*

- Implementing the interface with the micro-controller, and starting the transmission and reception of data, the following results are observed:

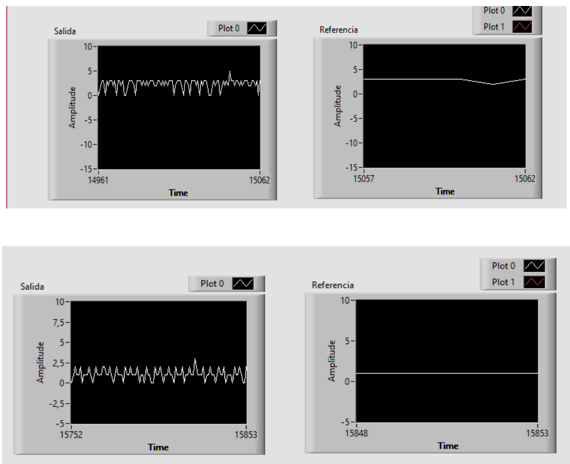



*Figure 15: System graphical interface REF=1*

As can be seen, the output signal tends to follow the reference value, it exhibits oscillations due to the influence of external factors that make the values taken intermittent or variable.

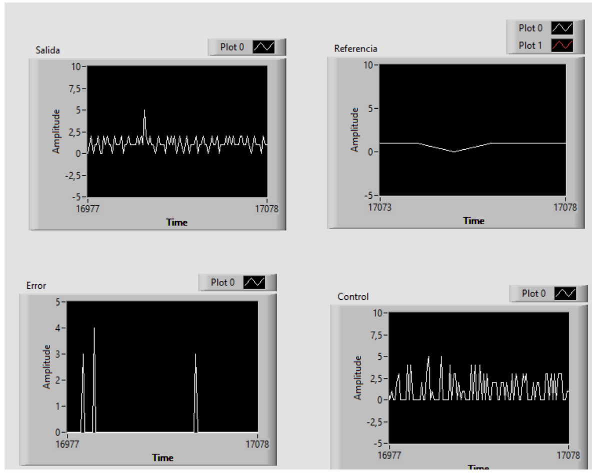



*Figure 16: Graphical user interface.*

In the previous image, it is observed that in the interface the error signal remains mostly at zero, however, due to oscillations produced by the reference value sent by the potentiometer. The output value follows the reference value.

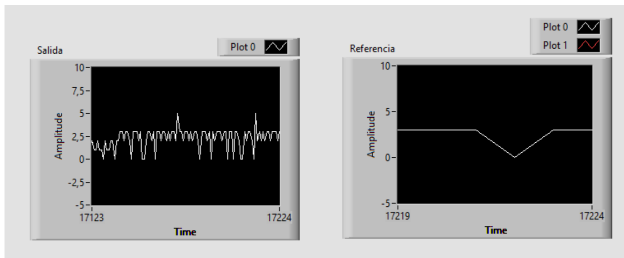



*Figure 17: Output 3 Volts (Left) and Reference 3 Volts (Right).*

**NOTE**: The left image is the output of the system, and the right image is the reference value to which you want to reach, the output presents oscillations in the stable state and falls to zero due to the voltage supply via USB of the Arduino microcontroller, however, it is observed that the error is less than 10%.

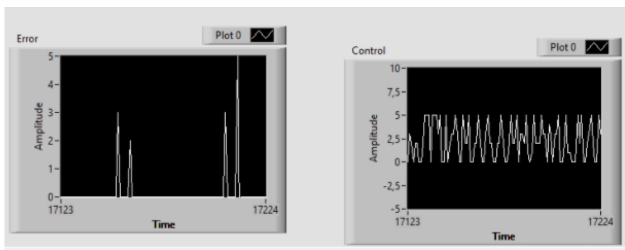



*Figure 18: Error signal (Left) and control signal (Right).*

For the previous image, it is observed that the error signal at a point where the reference falls, increases.

## 5. CONCLUSIONS

* The continuous-discrete control carried out for this laboratory practice, responded adequately in terms of controlling the system, however some oscillations are presented in the control, output, and reference signal.

* An influencing factor in these oscillations is the sampling time, since there is the possibility that the data collection system (LabView) takes data during the time in which the micro is processing the information and has not sent it through the communication channel.

* Although the system controls correctly, it is not the most appropriate, since if it is implemented in systems where great precision is required, these oscillations present drawbacks, so if in an application that is not practical (such as this session) it is recommended to use a microcontroller with a higher range oscillator.

## 6. REFERENCES

[1] K. Ogata, Discrete-Time Control Systems, Prentice Hall International, 1996.

[2] N. Instruments, "National Instruments," [Online]. Available: http://www.ni.com/data-acquisition/what-is/esa/. [Last access: 06 15 2016].