QUANSER MOTOR POSITION CONTROL

Subject: Control 1, Group: 53

Professor: Carlos Mario Paredes Valencia

Jairo Arenas, LuisFelipe Guerrero, Ruben Murillo, Cesar Varela
jairo.arenas_castro@uao.edu.co , luis_felipe.guerrero@uao.edu.co , ruben.murillo@uao.edu.co
cesar.varela@uao.edu.co
Department of Automation
Universidad Autónoma de Occidente
Santiago de Cali, Colombia

ABSTRACT

In this article, given about a practice in the UAO automation laboratory, a data collection was made for the position of the Quanser motor with respect to a voltage input. This shot was made to find the mathematical model or transfer function of said machine. In addition, the theoretical results were compared with the practical ones taken in the laboratory and, likewise, the obtained model was corroborated. From there, different controllers were designed and implemented to satisfy and reduce plant errors, the stages through which this work went through are presented below.

I. INTRODUCTION

Rotating machines are very important systems for the industry. Thanks to this, many companies manage to be in the market. The development of direct current motors was focused for many years on the search to transform alternating current into direct current. However, in this report a position controller for a shunt permanent magnet motor will be designed.

Today there are all kinds of equipment, such as electric motors of various models, sizes and powers to carry out a certain job. These machines work with alternating (AC) or direct (DC) current. However, there are rotating machines that can work in DC and AC, as is the case of the universal motor or the series excitation motor. Systems that require little power to move their mechanisms use motors that are powered by elements that generate DC voltage such as batteries, or DC AC converters. The construction of a simple electric motor, fed with direct current, represents special interest since it allows us to observe the close relationship between a magnet and the circulation of electric current through a coil, which in turn explains various electromagnetic phenomena. A motor is made up of two types of magnets, one of the permanent type and the other transient, which is the one that is formed once the electric current flows through the winding.

In this article about the final control project, we will be analyzing the mathematical model of the dynamic behavior of a DC motor, which in this case is a permuting magnet. This motor is basically a transducer that converts electrical energy to mechanical energy, therefore, as a first instance we must analyze it as a series circuit with its respective input and output of a system, to later model it by means of controllers designed and calculated to improve parameters. of process.

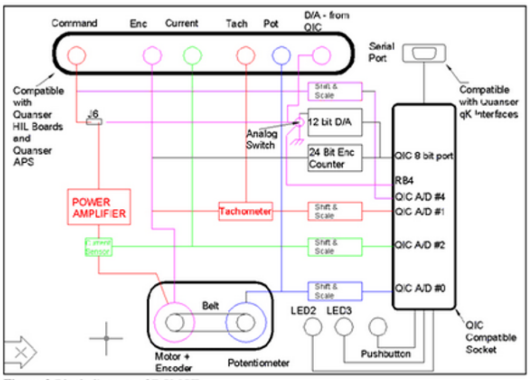


Figure 1: Connection for data acquisition.

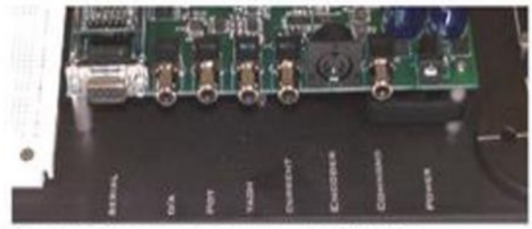


Figure 2: External connections.

II. OBJECTIVES

General objectives

- Obtain the transfer function of the Quanser engine.

- Represent in an analogous way the transfer function of the system by means of block diagrams.

Specific objectives

- Understand the operation of the Quanser engine, by simulating it in open and closed loop in Simulink.

- Calculate the control for the Machine.

Image 1: Quanser Dc Motor.

III. METHODOLOGY

Initially, during laboratory activity 1, the appropriate connections were made between the motor and the analog-digital card, in order to obtain the necessary data from Quanser's mathematical modeling. The connection is presented below:

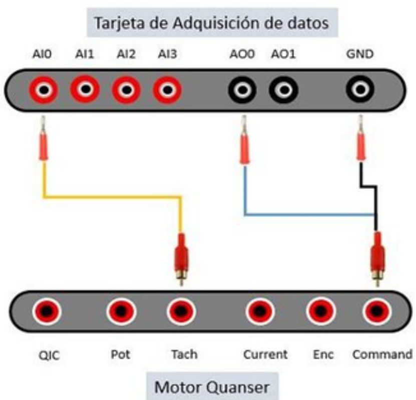


Figure 3: Motor Communication and Data Acquisition Card

Outputs	Device	Range
RCA4	Potentiometer voltage	± 5 VDC
RCA3	Tachometer voltage	± 5 VDC
RCA2	Current measurement	± 5 VDC
RCA5	QIC D/A output	± 5 VDC
5 pin DIN	Encoder output	TTL, A, B
Input		
RCA1	Command signal to power amplifier	± 5 VDC
Serial		
DB9	Serial to QIC	RS232
Power		
6mm jack	AC power to board	15 VAC

Figure 4: Motor Connections

Quanser is a versatile servo system designed to teach and demonstrate the fundamentals of DC motor control in a variety of ways. The system can be easily configured to control motor position and speed, as well as model experiments.

Theoretical modeling of an electromechanical system Transfer function.

Theoretical modeling of an electromechanical system, Transfer function. It is a mathematical expression that characterizes the input and output relationships of a time-invariant linear system. It is defined as the ratio of the Laplace transform of the output (response function) to the Laplace transform of the input (exciting function), under the assumption of zero initial conditions. This is used to know how any type of system behaves with respect to an input and an output. In our case we want to see the response of the position of a direct current motor.

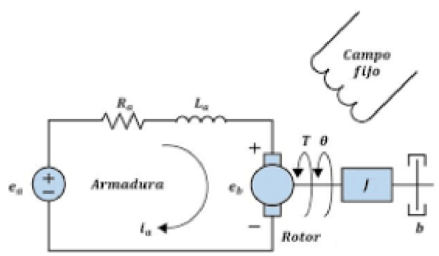


Figure 5. Schematic Equivalent circuit of a CC

In order to meet the objectives set by the professor, the following tools were used:

- Dc Quanser motor, provided by the laboratory. Reference: QNET DC Motor Board 2.0
- Card and data acquisition system.
- Web interface VI
- Matlab
- Simulink plotting program from matlab

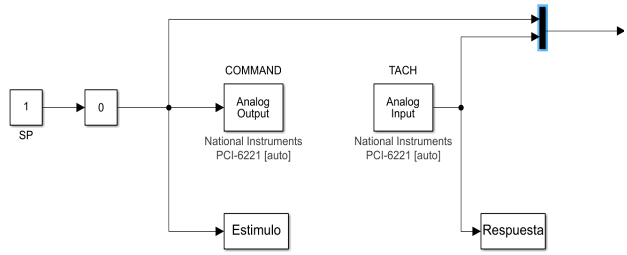


Figure 6: Connection of blocks for position measurement in Matlab-Simulink.

PROCEDURE:

Next, the procedure will be carried out to obtain the transfer function with the data obtained from the quanser engine. The data was saved as a Mat point, and later the values were organized in an excel file, to later transfer that file to the workspace. The data was organized as a vector type so that it could be imported into the “ident”.

	A Estimulo1 Number	B Respuesta1 Number
1	Estimulo	Respuesta
2	0	3.3448
3	0	3.2344
4	0	3.0415
5	0	2.8739
6	0	2.7015
7	0	2.5327
8	0	2.3839
9	0	2.2358
10	0	2.1000
11	0	1.9544
12	0	1.8390
13	0	1.7307
14	0	1.6344
15	0	1.5229
16	0	1.4327
17	0	1.3527
18	0	1.2684
19	0	1.1998
20	0	1.1357
21	0	1.0706
22	0	0.9960
23	0	0.9422

Image 2: Test data.

Through the command "Ident" which for newer versions of Matlab has a new name which is "system identification".

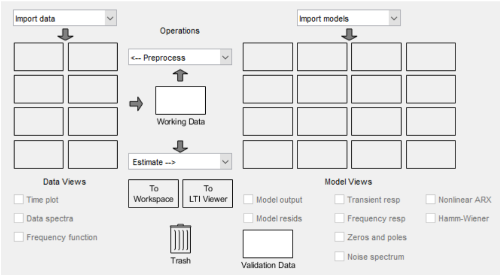


Figure 7: System identification

Then the "import data" data is imported. When carrying out this step, a menu is displayed in which "time domain data" is selected, which opens a new tab in which the input and output data, the sampling time that was 0.1s and initial time of 0s are entered. . Subsequently under these parameters it is given to import.

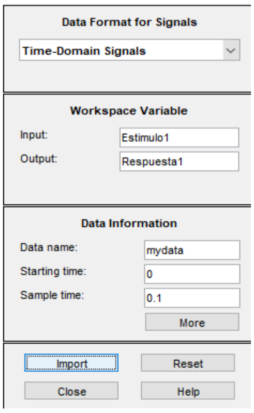


Figure 8: initial data selection format

After pressing import the program, I save a graph in the "import data" section.

With the graphs obtained, they are sent to the "working data" and "validation data" section to then execute the transfer functions models section, in which the number of poles corresponding to the test is entered. Finished this we click "estimate".

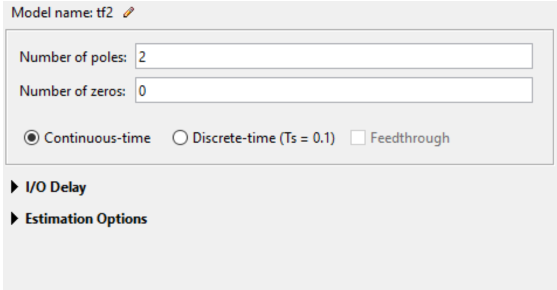


Figure 9: Conditions of the transfer function

With the graphs stored in "import models" in which you click on "model output" you can see the percentage of accuracy or adjustment between the measured and simulated model, it is also possible to see the step response, and the transfer function corresponding to the model as shown below:

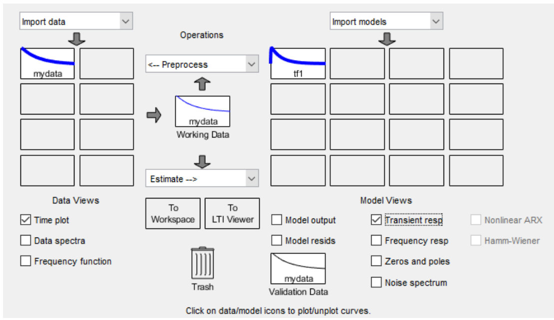


Figure 10: System Identification with saved parameters

```
         10.35
----------------------
s^2 + 20.42 s + 12.33
```

```
Status:
Estimated using TFEST on time domain data "mydata".
Fit to estimation data: 99.55% (stability enforced)
FPE: 0.0001229, MSE: 0.0001229
```

Image 3: Transfer function.

Note: As can be seen in the previous image, the second order transfer function had a 99.55% accuracy with respect to the data taken and an MSE (mean square error) of only 0.0001229. Therefore, the need to carry out more tests to compare them and choose the most approximate was not seen.

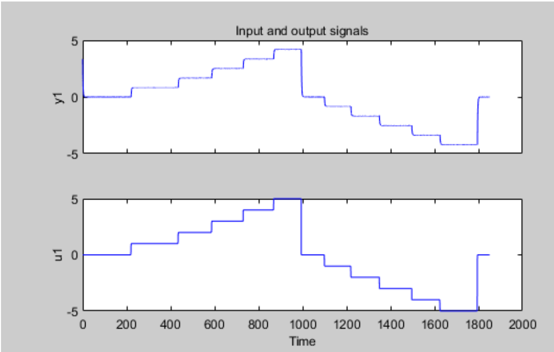


Figure 11: Graphs of the behavior of the quanser motor.

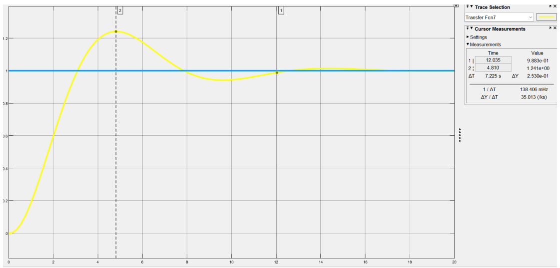


Figure 12: Step response of the system

In figure 14 you can see the graph that shows the transfer function obtained by means of system identification with the data obtained from the quanser for position, it will be possible to observe the behavior of the system without a controller, where you can improve the steady state error.

IV. RESULTS AND ANALYSIS

For the experiment, 3 compensators are designed in continuous time; a.) Lead-lag network, b.) PD controller and c.) Algebraic controller.

Below are the necessary calculations for the design of these controllers and the respective graphs of each controller in continuous and discrete time

a.) Lead-Lag Controller:

It begins by looking for the transfer function of the controller, taking into account the control objectives determined, below are the calculation steps to obtain the parameters that meet the design criteria.

- **Plant:** $G(s) = \frac{10.32}{s(s^2+20.42s+12.33)}$ Transfer function for position.
- A phase controller (lead/lag) is designed for the plant with the following criteria.
- Kv=25 Mf -d=45
- →**Kv**=Lim(s→0) s Gc(s) G(s) =
- $\frac{10.32}{12.33}$Kc$\frac{Z1Z2}{P1P2}$=25 we clear Kc
- Kc=29.87

Now with the data from the generated bode diagram, the crossing frequency

Wn=3.51 rad/sec Z2=0.3511 P2=0.0465 B=7.5486 is obtained.

advancement network Mg=22dB

20= $\frac{0dB-(-1.76dB)}{log(P1)-log(3.51)}$ We solve and we have that P1=4.2984

20= $\frac{-20-(-1.76dB)}{log(Z1)-log(3.51)}$ We solve and we have that Z1 =0.42984.

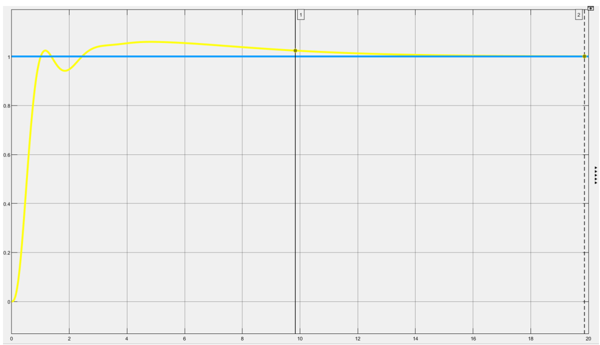


Figure 13: Continuous lead/lag controller.

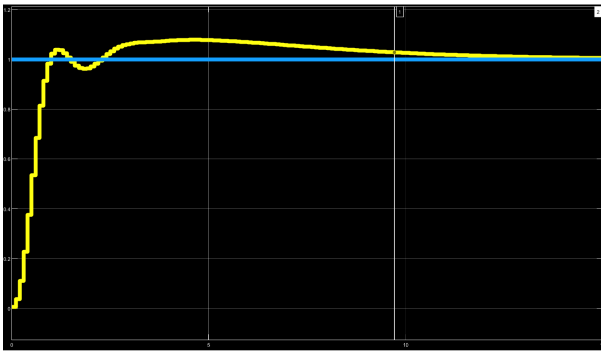


Figure 14: Discrete Lead/Lag Controller.

b.) Analytical PID controller:

for this control a PD controller is used since the plant has an integrator and if another is added it would have two values at the origin which would make the system marginally stable and when making a feedback it would be unstable

C(s) =Kp+Kds

Characteristic Equation

$s^3 + 20.42s^2 + 12.33s + 10.32Kds + 10.32Kp\ Desire$

E.Mp=1.52% Ts=2.5 Remote Point (S+17.42)

$s^3 + 20.42s^2 + 59.1056s + 68.9076$

Now we clear Kd and Kp then

12.33+10.32Kd = 59.1056

10.32Kp= 68.9076

Kd=4.5325

Kp=6.6771

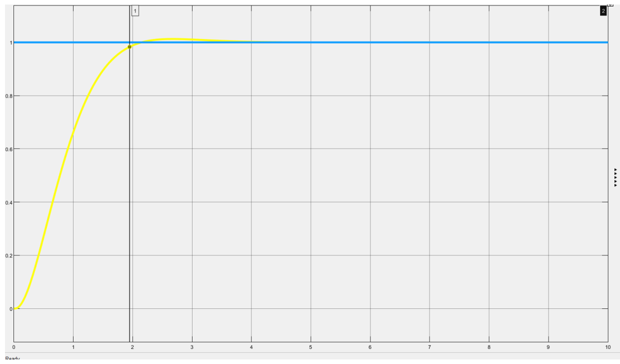


Figure 15: Continuous analytical PID controller.

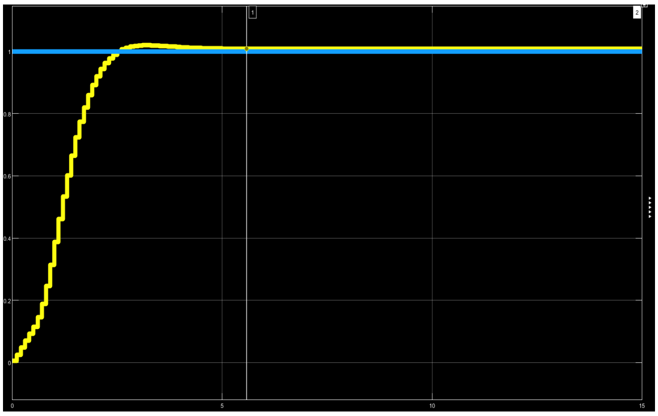


Figure 16: Discrete analytical PID controller.

c.) Analytical algebraic controller:

Desired E.2(plant order)-1

Desired E.2(3)-1=5

3+controller order = Desired

order Controller order = desired order -3

controller order = 5-3 = 2

controller of order 2 $C(s) = \frac{q2s^2+q1s+q0}{Ps^2+P1s+P0}$

Since there are 6 unknowns, a matrix is used to obtain its results in the following way:

the first column of the matrix A are the terms that accompany the S of the denominator of the plant

Second column the numerator of the plant

finally for columns 3, 4, 5 and 6 is the same process adding a 0 to the first term of said column.

$$A = \begin{bmatrix} 0 & 10.32 & 0 & 0 & 0 & 0 \\ 12.33 & 0 & 0 & 10.32 & 0 & 0 \\ 20.42 & 0 & 12.33 & 0 & 0 & 10.32 \\ 1 & 0 & 20.42 & 0 & 12.33 & 0 \\ 0 & 0 & 1 & 0 & 20.42 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

For matrix B, the numerators and denominators of the controller are taken starting from P0 and q0 and alternating descendingly denominator and numerator respectively

$$B = \begin{bmatrix} P0 \\ q0 \\ P1 \\ q1 \\ P2 \\ q2 \end{bmatrix}$$

$$C = \begin{bmatrix} 23566 \\ 22763.7 \\ 9239.45 \\ 1156.65 \\ 57.42 \\ 1 \end{bmatrix}$$

B= inv (A) * C

$C(s) = \frac{82.11s^2+1741.46s+2283.52}{s^2+37s+388.63}$

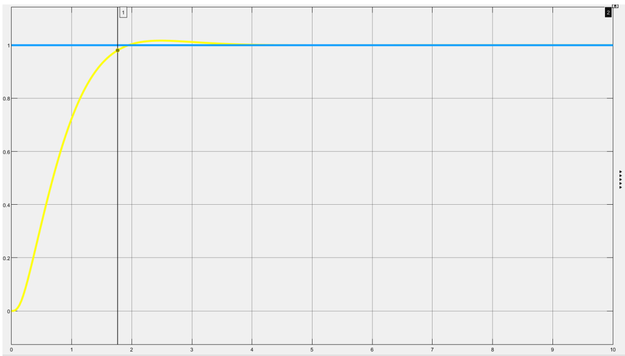


Figure 17: Continuous algebraic controller.

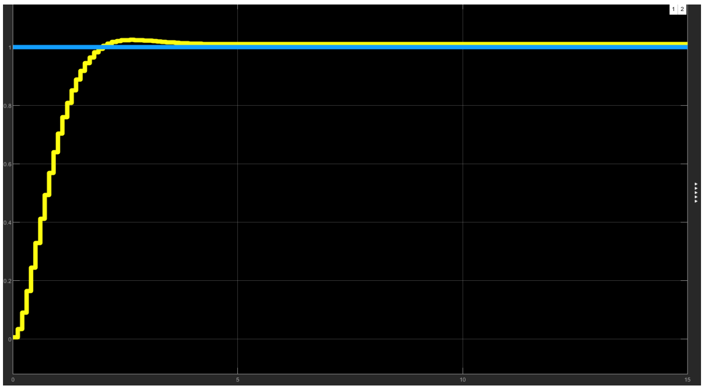


Figure 18: Discrete algebraic controller.

Next, the 3 controllers are graphed in continuous time, where the differences that exist are observed, and the benefits that a controller offers with respect to the others.

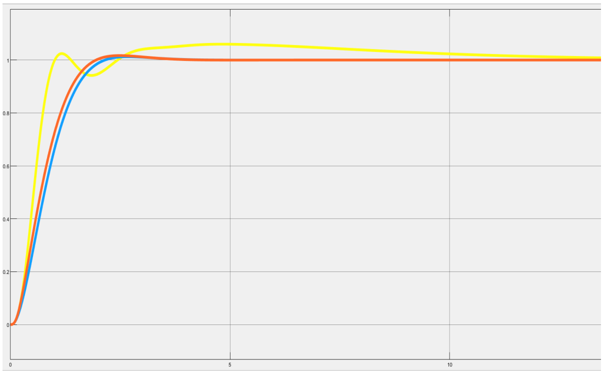


Figure 19: Controllers in continuous time.

```
step(g)
%% discretizacion
gdis=c2d(g,ts,'tustin')
step(gdis)
%% polos=pole(gdis)
a=-polos(1)
b=-polos(2)
c=-polos(3)
zeros=zero(gdis)
d=-zeros(1)
e=-zeros(2)
f=-zeros(3)
%% adelanto atrazo
kc=29.87
z1=0.42984;
p1=4.2984;
z2=0.351141039329555
p2=0.046517174423448
num=conv(kc*[1 z1],[1 z2]);
den=conv([1 p1],[1 p2]);
adelanto=tf(num,den)
adelantoatrazo=c2d(adelanto,ts,'tustin')
%% pid
s=tf('s')
pid=6.6771+(4.5325*(100*s)/(s+100))
pidd=c2d(pid,ts,'tustin')
%% algebraico
s=tf('s')
```

Now the same action is performed but for discrete-time controllers.

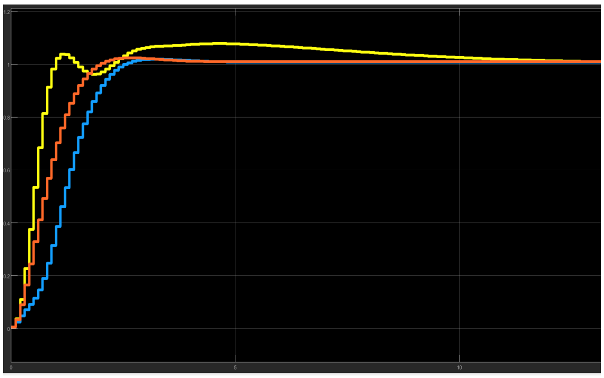
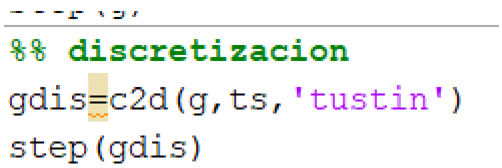


Figure 20: Controllers in discrete time.

Discretization: In order to bring the controllers designed and calculated in continuous time in this section to discrete time, the transfer functions of these controllers and of the plant are taken, they are taken to matlab code. a sampling time (ts) is established as required by the software for the discrete domain, then tf of each controller and silver is called, and the TUSTIN command is used, which takes the laplace systems to Z, taking thus the discrete controllers. In image 4 you can see the script used in Matlab to discretize the systems.



```
%% discretizacion
gdis=c2d(g,ts,'tustin')
step(gdis)
```

Image 4: Discretization in Matlab

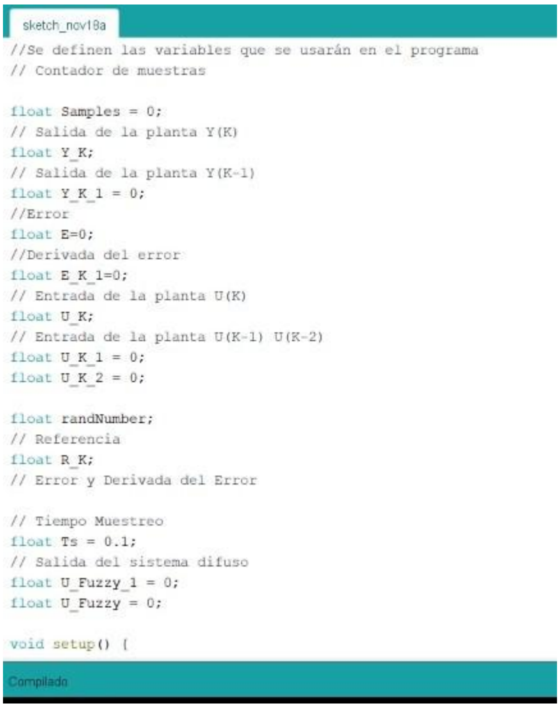
Difference Equation.

The PID controller is taken in discrete time, which in the case of this work is a proportional derivative, having the integral with value 0. Under the following scheme and calculations, the difference equation of this controller is obtained

- $\frac{U(z)}{E(z)} = \frac{82.22z - 7109}{z + 0.6667}$
- ZU(z)+0.6667U(z) = 82.22zE(z) -71.09 E(z)
- U(z)=82.22E(z)-71.09z^{-1}E(z)- 0.6667 z^{-1}U(z)
- U(k)= 82.22E(k)-71.09E(k-1)-0.6667U(k-1)

After obtaining the difference equation of the PD controller, the equation is taken to a microcontroller in arduino.

Already in the arduino scribd the construction of the code begins. The variables are defined, the current sample, the previous sample, the sampling period, the reference, current control action and previous control action. Below you can see the code on the arduino for the micro.



sketch_nov18a

```
//Se definen las variables que se usarán en el programa
// Contador de muestras

float Samples = 0;
// Salida de la planta Y(K)
float Y_K;
// Salida de la planta Y(K-1)
float Y_K_1 = 0;
//Error
float E=0;
//Derivada del error
float E_K_1=0;
// Entrada de la planta U(K)
float U_K;
// Entrada de la planta U(K-1) U(K-2)
float U_K_1 = 0;
float U_K_2 = 0;

float randNumber;
// Referencia
float R_K;
// Error y Derivada del Error

// Tiempo Muestreo
float Ts = 0.1;
// Salida del sistema difuso
float U_Fuzzy_1 = 0;
float U_Fuzzy = 0;

void setup() {
```

Compilado

Image 5: Microcontroller with arduino

CONCLUSIONS.

- It could be verified that the Computer tools greatly facilitate the capture and processing of data. In addition, they are essential when carrying out an adequate characterization of the plant.
- To realize the PID controller, only one PD was made to prevent the system from becoming unstable.
- The algebraic and PID controllers are very similar in the simulation, in general all the controllers (continuous and discrete) meet the control objectives during the simulation.
- The data collection for a control system must be taken into account, which is calculated correctly.

DISCUSSION .

There were some drawbacks at the time of taking the project to the real part, because perhaps failures occurred at the time of data collection from the plant, although the software registered 99% accuracy, it should be checked if it was in the right thing to do to decide if this was really the transfer function of the plant and thus obtain the mathematical models.

It is analyzed at the moment of running the controller already implemented with the hardware that does not stabilize as we wish, which could be caused by a bad parameterization that, as previously indicated, could have been due to a bad modeling of the plant, simulations are made with the other controllers but in all of these it presents the same error, with the help of the teacher the parameters of the PID controller and the sampling times are varied to finally achieve the stability of the system and that it meets the established criteria, finally it remains as an experience for an upcoming practice or a real life task that should always check the results that the software throws in order to achieve a clear and precise solution of the modeling of the controller.

REFERENCES.

[1] Power Electronics. Daniel W. Hart. Universidad de Valparaíso Pearson Education SA Madrid 2001. Full wave and triphasic rectifiers: AC-DC Conversion (Chapter 4).

[2] Electrical Machines. Stephen J. Shapman of Houston. SA Mexico. Fundamentals of dc machines (chapter 4) and dc motors (chapter 6)

[3] Electrical Machines. Jesus Frayle Mora. Fundamentals of DC Motor Machines (Chapter 6)

[4] Katsuhiko Ogata., 2011. Modern Control Engineering. 5th ed.
Pearson Education of Mexico, SA de CV, pp.140,146,150,178

[5] Quanser Consulting Inc. User Guide Quanser Engineering Trainer DC Motor Control.

[6]https://www.revisingenieria.unam.mx/numeros/2018/v19n4-09.pdfocw

[5]https://www.youtube.com/watch?v=d1xfirFBd4Q[

5]https://.unican.es/pluginfile.php/763/course/section/814/ejercicios_3.pdf