# Discrete time control, 7 different methods.

Galvis. David. I restrepo. Camilo.

{u1802584,u1802556}@unimilitar.edu.co

Universidad Militar Nueva Granada

*Summary*--- In this laboratory, a mass-spring-type physical plant introduced in a fluid is going to be assembled, which is connected to a motor that transmits its power through pulleys, the operation of said plant is through a PUSH-PULL circuit and a hall effect sensor.

Various discrete controllers will be applied to the described plant and the behavior of the entire system will be analyzed based on each of the respective controllers in order to make a comparison and determine which ones may be better in operation and ease. of implementation.

*Abstract*--- This practice has the purpose of making the assembly of a physical plant type mass - spring introduced into a fluid which is connected to a motor that transmits its power through pulleys, the operation of this plant is through a PUSH-PULL circuit and a hall effect sensor.

A variety of discrete controllers will be applied to the plant described and the behavior of the entire system will be analyzed based on each of the respective controllers in order to make a comparison and determine which can be better in operation and in ease of implementation.

*Keywords*--- Discrete control, difference equation, saturation.

*General Objective*--- Implement different control strategies in discrete time using different equations.

*Specific objectives*--

* Design and implement a PID controller in continuous time and discretize it.

* Design and implement a PID controller in discrete time.

* Design and implement a controller using the root locus method.

* Design and implement a frequency compensator in discrete time.

* Design and implement a plant override controller.

* Design and implement a controller for dead oscillations.

* Design and implement a servosystem in discrete time.

## INTRODUCTION

The following document presents the design of 7 discrete-time controllers and the results obtained for each of these. Through the use of Matlab and LabView software.

DISCRETE CONTROL:

Discrete-time control systems are those systems in which one or more of the variables can change only in discrete time values. These instants are denoted with kT or tk = (k=0,1,2,3...) and can specify the times in which some physical measurement is carried out or the times in which the data is extracted. from the memory of a digital computer. The time interval between these two discrete instants is assumed to be short enough that the data for the time between them can be approximated by simple interpolation. Discrete-time control systems differ from continuous-time control systems in that the signals for the former are in the form of sampled data or in digital form. If a digital computer is involved in the control system as a controller, the sampled data must be converted to digital data.[1]

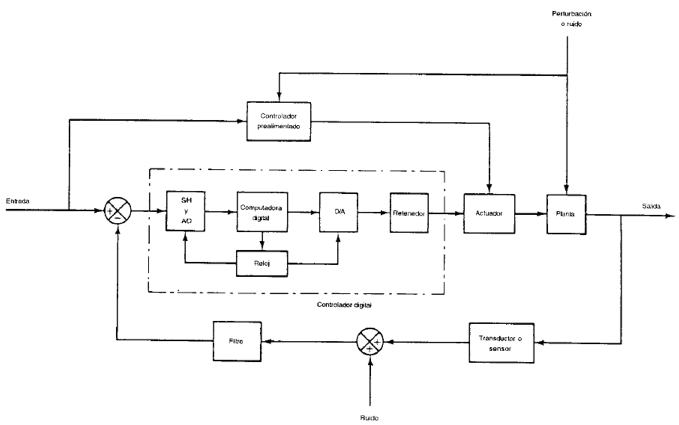


***Figure 1****: Block diagram of a digital control system.*

PUSH-PULL AMPLIFIER

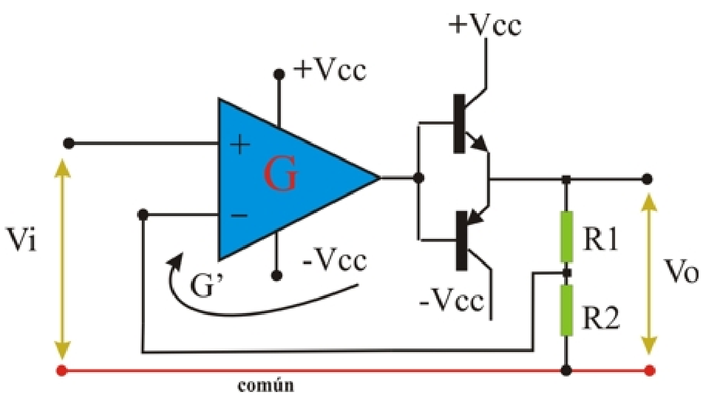


***Figure 2****: Push-pull circuit with operational amplifier*

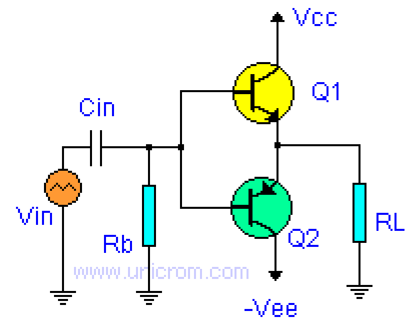


***Figure 3.*** *Push-pull or push-pull*

amplifier This amplifier is called push-pull or push-pull amplifier, since it uses 2 groups of transistors. Each group is responsible for amplifying a single phase of the input wave. [2]

One group is yellow and the other is green. (in this case there is only one transistor per group). When one group comes into operation, the other comes in, and vice versa. A common emitter amplifier is used to amplify small signals. In this configuration the voltage of the output signal has practically the same amplitude as that of the input signal (unity gain) and they have the same phase.

When the input signal is large and what is desired is to increase the current delivery capacity, a push-pull or push-pull amplifier is used. (Power amplifier). The amplifier shown in the following graph is made up of two transistors. One NPN and one PNP with the same characteristics. [3]

1. MATERIALS

* Software: MATLAB ®

* White coat.

* LabView DAQ acquisition card

* Plant (DC motor, pulleys, coconut oil fluid, spring, wooden support, hall effect).

# 2. PROCEDURE

Using the plant characterized in delivery 2 of the laboratory, the following transfer function is used for all the controllers, a new characterization was not necessary.

$$\frac{1.606}{s^2 + 2.861\, s + 2.047}$$

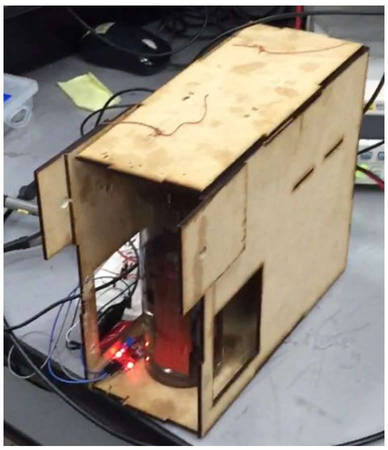

***Figure 4** Physical layout of the plant*

The general graphic interface used for all the controllers is shown below:

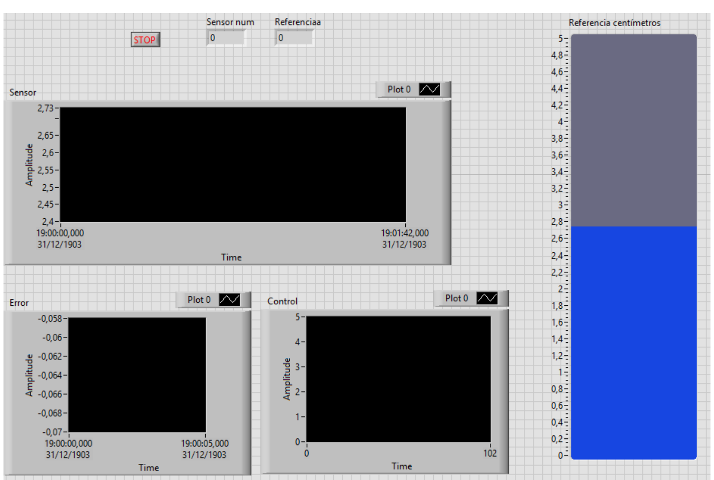


***Figure 5**: Graphic interface in LabView*

For the plant, centimeters or distance begins to be measured from the surface of the table upwards, in this case, 1 cm is the closest to the surface and 5 cm is the highest point to which the mass can reach. In addition to this, the reference of the slider is in centimeters and that of the wavechart or graph is in volts, it enters closer to the surface on the mass, the output voltage is higher, for the hall effect sensor used.

**PID continuous Discretized**

Having the plant in continuous:

$$\frac{1.606}{s^2 + 2.861\, s + 2.047}$$

and applying a continuous PID controller, which has the following polynomial:

$$PID = \frac{kds^2 + kps + ki}{s}$$

The general system is as follows:

$$G(s) = \frac{PID.planta}{1 + PID.planta}$$

The characteristic polynomial results from the denominator of G(s), from which the following is obtained:

$$Pc = s^3 + s^2(1.606kd + 1) + s(1.606kp + 2.047) + 1.606\, ki$$

The desired polynom is of the form:

$$(s^2 + 2\xi wns + wn^2)(s + 5\xi wn)$$

A settling time of 2 seconds and a damping coefficient of 0.7 are required

$$wn = \frac{4}{\xi . ts} = \frac{4}{(0.7)(2)} = 2.8571 \frac{rad}{s}$$

Then the resulting desired polynomial is:

$$Pd = s^3 + 14s^2 + 48.163\, s + 81.633$$

Equating the coefficients of both polynomials:

$$14 = 1.606kd + 1$$

$$48.163 = 1.606kp + 2.047$$

$$81.633 = 1.606ki$$

$$kd = 8.0946$$

$$kp = 28.7148$$

$$ki = 50.83$$

Now we proceed to discretize by tustin, choosing a sampling time of 30 times less than the desired establishment time, the values obtained are:

```
q0t =

  151.8281


q1t =

 -269.8585


q2t =

  121.4190
```

**Discrete PID**

For the discrete PID controller, a damping coefficient (zita) of 0.7 and a desired establishment time of 4 seconds are selected. The sampling time is selected as one tenth of the desired settling time, thus being 400 ms. A PID with filter is then designed and for this case the desired polynomial must be of 3 order, the usual method is taken in discrete time (z-re+j*im)*(z-re-j*im) and also multiplies by the universal non-dominant pole (z-0.05). Performing the feedback of the controller by the plant, equating the denominator of the feedback transfer function with the desired polynomial. Later, using the matlab <solve> function, it is possible to clear and obtain the values q0, q1, q2 and s0 that make up the controller.

**LGR**

have the plant in open loop:

```
sysLA =

          1.606
  ---------------------
  s^2 + 2.861 s + 2.047

Continuous-time transfer function.
```

And in closed loop:

```
sysLC =

          1.606
  ---------------------
  s^2 + 2.861 s + 3.653

Continuous-time transfer function.
```

Which has the following response:

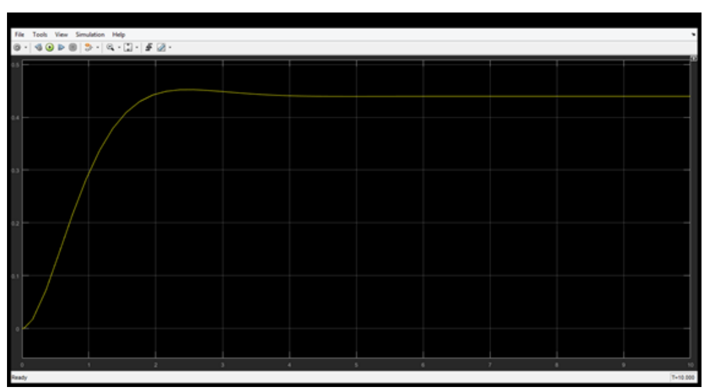

*Figure 6: Response of the plant in open loop*

Calculation of zita and wn of the plant:

$$wn = \sqrt{3.653} = 1.9113 rad/s$$

$$zita = \frac{2}{2 * wn} = 0.7486$$

$$ts = \frac{4}{(zita)(wn)} = 2.7957\ seg$$

Which is in accordance with the response of the plant in LC since its zita is equal to 0.7486, which indicates an underdamped response.

**CONTROL OBJECTIVES:**

A settling time of 50% and a zita of 0.5

$$tsd = 0.5(2.7957) = 1.3979\ seg$$

$$wn = \frac{4}{(0.5)(1.3979s)} = 5.723 rad/s$$

$$wd = \left(\frac{5.723rad}{s}\right) * \sqrt{1 - 0.5^2} = 4.9563 rad/s$$

is desired. A sampling time 20 times less is assumed, and it is verified that it meets the condition (ws>8wd):

$$Tm = \frac{1.3979s}{30} = 0.0466s$$

$$Ws = \frac{2 * pi}{0.466s} = 134.8450$$

$$8wd = (8)(4.9563) = 39.6501$$

$134.8450 > 39.6501$ Cumple la condición

$$s = -(zita)wn \pm jwd$$

$$z = e^{-(Tm)(zita)(wn) \pm jwd}$$

$$|z| = e^{-(Tm)(zita)(wn)} = e^{-(0.046)(0.5)(5.72)} = 0.8752$$

$$ang(z) = Tmwd = (0.046)(3.46) = 0.2309\ rad$$

$z = 0.8519 + j\,0.2003$ Este representa el polo deseado

Now the plant is discretized in MATLAB by c2d with `zoh` and we obtain:

```
sysd =

  0.001668 z + 0.001595
  ----------------------
  z^2 - 1.871 z + 0.8752

Sample time: 0.046596 seconds
Discrete-time transfer function.
```

The plant in discrete as it contains an integrator can also be expressed as follows:

$$H(z) = \frac{0.0017(z + 0.9565)}{(z - 0.9358)(z - 0.9352)}$$

It can be noted that the plant contains a Zero at -0.9565 and two poles at 0.9358 and 0.9352 respectively. The angles formed by the Zero and the poles with respect to the desired pole are calculated as follows:

$$\varnothing 1 - \theta 1 - \theta 2 = -180º$$

$$6.3208º - 112.71º - 112.5728º = -218.96º$$

The excess angle that occurs is:

$$-218.96º - (-180º) = -38.96º$$

To eliminate this effect, the compensator is used in such a way that the sum of the angles is equal to -180º.

**To know the value of alpha:**

It can be noticed that the poles of the plant can be annulled, then the zeros of the compensator are found at 0.9358 and 0.9352 (The final controlled system must contain an integrator) Therefore, the numerator can be easily known that must contain the COMPENSATOR, which for this case is of the form:

$$Gc(z) = K\,\frac{(z - alfa1)(z - alfa2)}{(z - 1)(z - beta)}$$

The compensator until now is remaining:

$$Gc(z) = K\,\frac{(z - 0.9358)(z - 0.9352)}{(z - 1)(z - beta)}$$

**To know the value of beta:**

First, it must be replaced by the angle of the less dominant pole, which has already been canceled with the numerator of the compensator, and the following must be fulfilled:

$$\emptyset 1 - \theta 1 - \theta\beta = -180º$$

$$\theta\beta = 180º + \emptyset 1 - \theta 1$$

$$\theta\beta = 180º + 6.32º - 12.71º$$

$$\theta\beta = 73.6046º$$

$$\tan\theta\beta = \frac{im}{re - \beta}$$

$$\beta = re - \frac{im}{\tan\theta\beta}$$

$$\beta = 0.8519 - \frac{0.2003}{\tan(73.6º)}$$

$$\beta = 0.8973$$

The compensator is left as

$$Gc(z) = K\,\frac{(z - 0.9358)(z - 0.9352)}{(z - 1)(z - 0.8973)}$$

**To know the value of K**

The magnitude of the product between the plant in open loop and the compensator evaluated in the desired pole must be equal to 1.

$$|Gc(z).H(z)|_{z=0.515+j0.428} = 1$$

$$\left|K\,\frac{(z - 0.9358)(z - 0.9352)}{(z - 1)(z - 0.8973)}\cdot\frac{0.0017(z + 0.9565)}{(z - 0.9358)(z - 0.9352)}\right|_{z=0.8519+j\,0.2003} = 1$$

In MATLAB it is done:

$$K = \frac{1}{abs(subs(\,(|Gc(z).H(z)|), z, 0.8519 + j0.2003\,)}$$

K=16.8421

The compensator remained:

$$Gc(z) = 16.8421\,\frac{(z - 0.9358)(z - 0.9352)}{(z - 1)(z - 0.8973)}$$

**Plant**

general equation for any controller using the plant override method is expressed as:

$$Dc(z) = \frac{1}{Gp(z)}\cdot\frac{M(z)}{1-M(z)}$$

Where M (z) is the desired behavior, or mathematically, it can be described as follows:

$$M(z) = \frac{1-e^{\frac{-T}{\tau}}}{z-e^{\frac{-T}{\tau}}}$$

It is defined as $T = ts/5$, $\tau = ts/10$, where ts is the time of desired establishment.

### Dead oscillations

This controller assures us of dead oscillations, meaning error in stable state equal to zero, then the desired behavior is M(z)= $1 - (1 - Z^{-1})^{n} F(z)$, where n is the order of the systema, when the system is of second order then the F(z) must also be of second order. Two variables m1 and a1 are then generated that must be found, and in this case after finding their respective values.

$$Dc(z) = \frac{1}{Gp(z)} \cdot \frac{M(z)}{1-M(z)}$$

$$\frac{M(z)}{1-M(z)} = \frac{(1+zero^{*}z^{-1}) \cdot m1 \cdot z^{-1}}{(1-z^{-1})(1+a1 \cdot z^{-1})}$$

### Frequency compensator

For the design of the frequency compensator, the plant must first be discretized as was done with the previous controllers. Then the plant is moved discreetly to the matlab sisotool tool. Where the compensator is designed by first selecting a desired zita and settling time and observing the response as poles and zeros are located at the root loci as well as adjusting the magnitude and phase gain on the Bode plot. , in order to obtain the controller that allows the plant to respond in the desired way, since it indicates each change made and how it affects the output.

### Servosystem

For the design of the discrete servosystem, the first thing that must be obtained are the matrices G and H of the system. The matrices G and H are obtained in the following way:

$$\mathbf{G}(T) = e^{\mathbf{A}T} = \begin{bmatrix} 1 & \frac{1}{2}(1 - e^{-2T}) \\ 0 & e^{-2T} \end{bmatrix}$$

$$\mathbf{H}(T) = \left( \int_0^T e^{\mathbf{A}t} \, dt \right) \mathbf{B}$$

Where t is the sampling time.

Because the servo system must be used with state space matrices, by Ackerman's method, the packed matrices of the system must be found:

$$\hat{G} = \begin{bmatrix} G & 0 \\ C \cdot G & 1 \end{bmatrix} \quad \hat{H} = \begin{bmatrix} H \\ C \cdot H \end{bmatrix}$$

So that with this value find fi of G packed, which allows us to introduce the alpha values of the polynomial wanted.

$$\varphi(\hat{G}) = \hat{G}^3 + \hat{G}^2 \alpha_1 + \hat{G} \alpha_2 + I \alpha_3$$

And then using Ackerman's method with which we find the feedback constant and the integral constant.

$$K = [0 \quad 0 \quad 1] \cdot [\hat{H} \;\; \hat{G}\hat{H} \;\; \hat{G}^2\hat{H} \;]^{-1} \cdot \varphi(\hat{G})$$

$$K = [k1 \quad k2 \quad ki]$$

And the design of the observer through dead oscillations, as the plant is of second order, the fi term of G becomes $G^{n}$ where n is the order.

$$K_e = G^2 \cdot \begin{bmatrix} C \\ C \cdot G \end{bmatrix}^{-1} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

## 3. ANALYSIS OF RESULTS

### Discretized continuous PID.

Simulating with the continuous PID, the following is obtained:

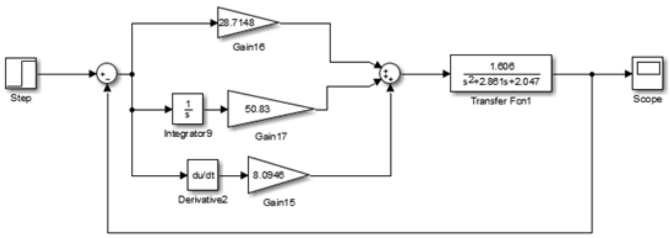


***Figure 7*** *: Continuous PID diagram*

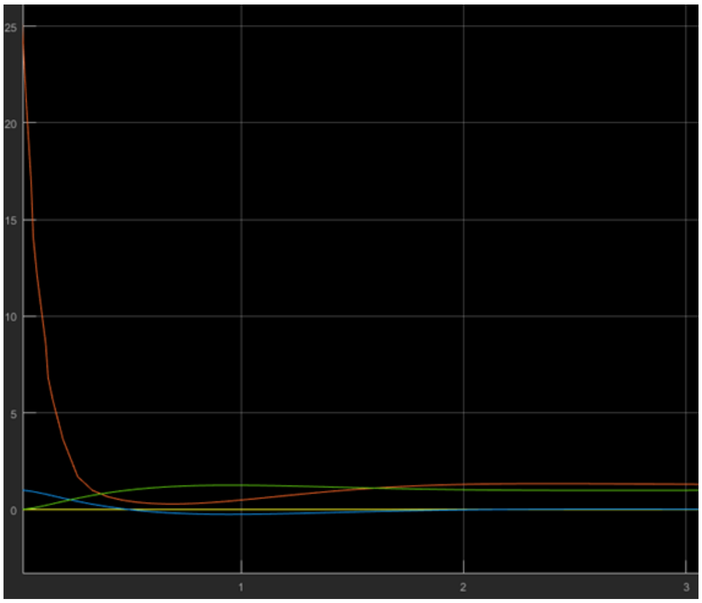

***Figure 8*** *: Continuous PID Scope*

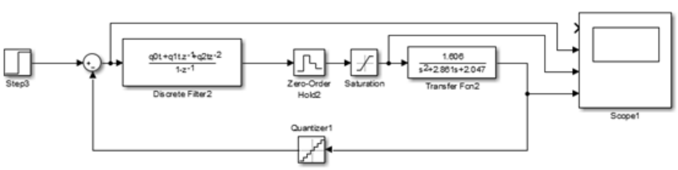


***Figure 9*** *: Discretized PID diagram*

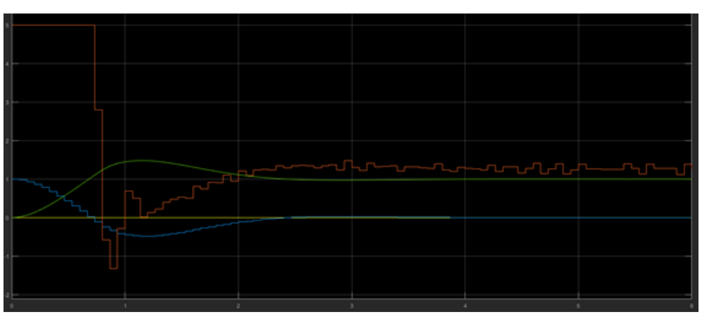

***Figure 10*** *: Discretized PID Scope*

It can be seen that due to the saturator there is a slightly oscillatory behavior for the control signal, which converges to approximately 1.3.

**Discrete PID**

Discretized plant in a sampling time of 0.4 s.

```
Tm =
    0.4000

disc =

   0.08855 z + 0.06044
  ----------------------
  z^2 - 1.129 z + 0.3184
```

The desired polynomial is expressed as:

```
Zdeseado =
z^4 - 1.33*z^3 + 0.575*z^2 - 0.048*z + 0.00112
```

Equating the polynomial with the denominator of the pid+filter, the constants

```
Tm =
    0.4000
q0 =
    5.3695
q1 =
   -5.7617
q2 =
    1.7173
s0 =
    0.3225
```

are obtained: And the following response was obtained in simulink, which can be seen in figure , where: Green=Control, Blue=Reference, Red= Error, Yellow=Exit (system response).

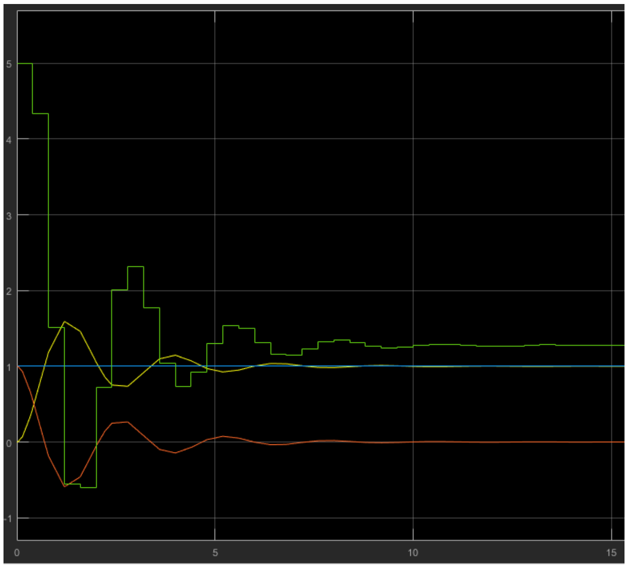


***Figure 11*** *: Discrete-Discrete Scope PID*

## Difference Equation in LabView

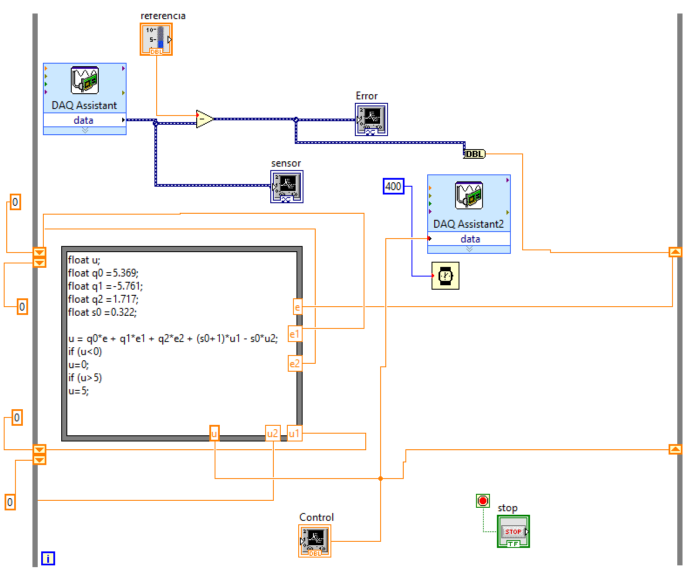


***Figure 12*** *: LabView Discrete-Discrete*

## LGR

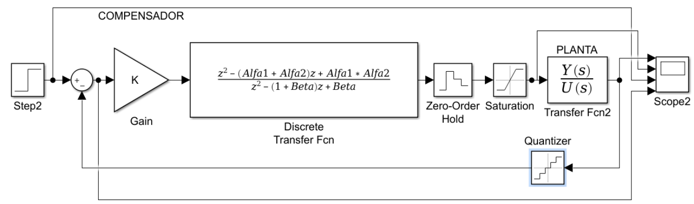


***Figure*** *13: Simulink LGR Diagram*

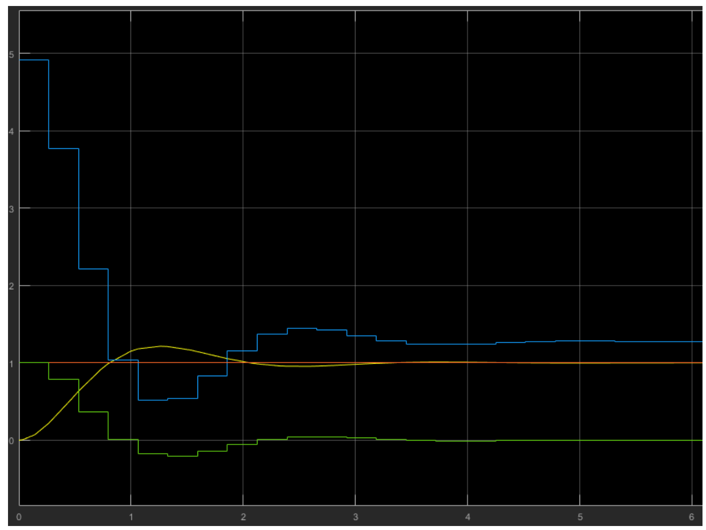

***Figure*** *14: Scope LGR*

For the figure: Blue = control, Orange = reference, Green = error, Yellow = system response.

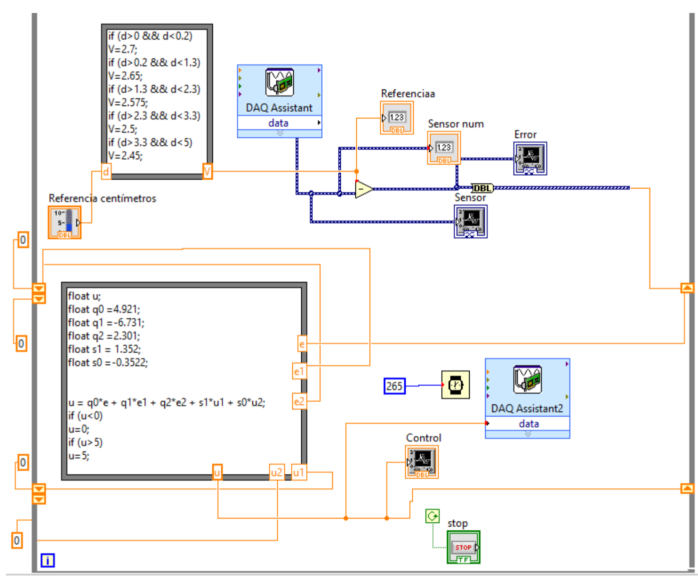


***Figure 15*** *: LabView LGR Diagram*

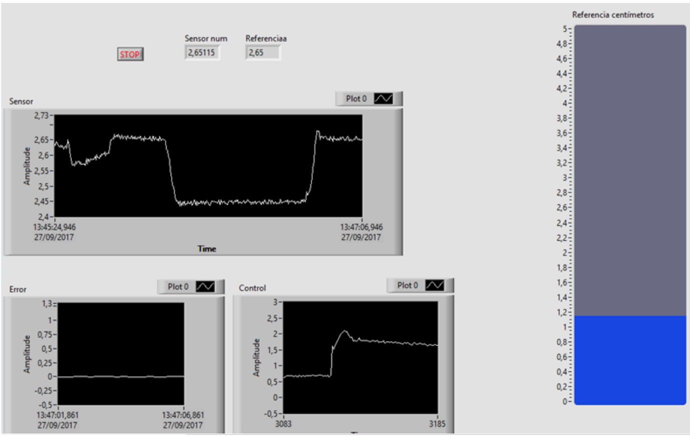


*Figure 16* : *LabView LGR GUI*

The control signal shown in the figure corresponds to the action that displaced the mass and changed the sensor voltage from 2.45 to 2.65, where in the control signal output <Sensor> an underdamped response is observed.

**plant**

is discretized using a sampling time of 419 ms.

```
 0.09551 z + 0.06401
----------------------
z^2 - 1.098 z + 0.3016
```

With this, after performing the named procedure for this controller, the controller equation separated by the numerator and denominator is obtained.

```
numDC1 =
4.1197*z^2 - 4.5234*z + 1.2409
denDC1 =
z^2 - 0.32976*z - 0.67024
```

To use the MATLAB coefficients function, create the transfer function that will be used in Labview

```
ukk =
4.1197*ek - 4.5234*ek1 + 1.2409*ek2 + 0.32976*uk1 + 0.67024*uk2
```

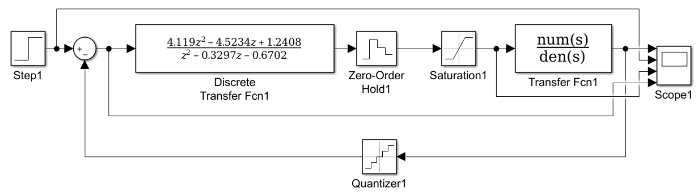


*Figure 17* : *Simulink diagram Plant*

cancellation In the simulation, an overdamped response was obtained just as a plant that has a plant cancellation controller responds.

For figure number : Red=control, Green=error, Blue=reference, Yellow=System response.

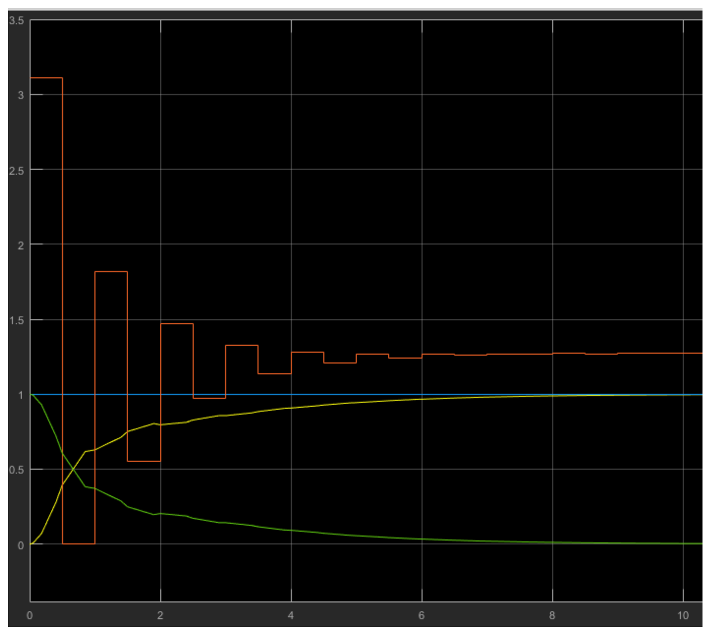

*Figure 18* : *Plant Cancellation Scope*

The constants obtained in the LabView script are assembled.

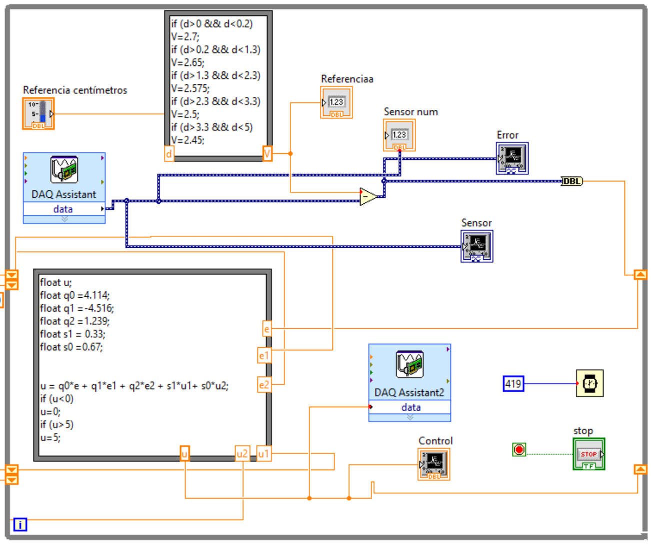


***Figure 19*** *: LabView Plant Override Diagram*

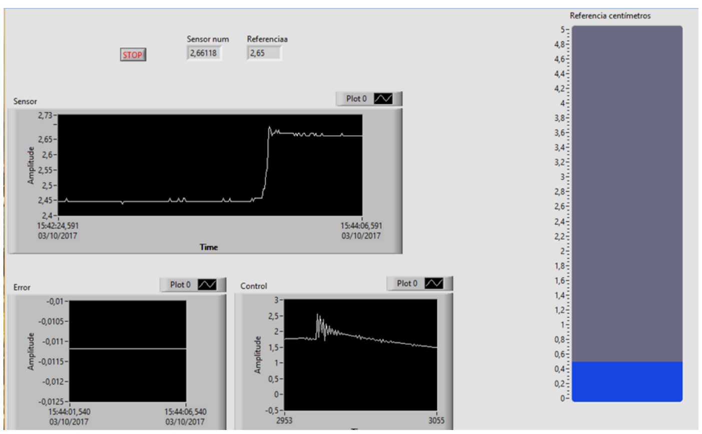


***Figure 20*** *: LabView Plant Override GUI*

It is observed that the control signal, as in the simulation carried out in simulink, oscillates more than the other controllers until it reaches the value that allows the system to reach The reference.

**Dead**

oscillations In dead oscillations the sampling time (ts) is selected as the desired settling time (tss) divided by the plant order. ts = tss/n, then for a settling time of 1 second, the sampling time is 500 ms and the discretized plant is as follows.

$$\frac{0.1265\ z + 0.07842}{z^2 - 0.9781\ z + 0.2392}$$

The values of m1 and a1 using the equation named in the procedure and knowing that the zero of the discretization is located at 0.6202, are the following:

m1 = 0.6172 and a1 = 0.3828, always the result of the sum of both must be of 1, which means that the procedure was carried out well.

Knowing the above, the difference equation is as follows.

```
ukk =
4.881*ek - 4.7739*ek1 + 1.1675*ek2 + 0.61721*uk1 + 0.38279*uk2
```

The response in simulink for this controller is found below, where in the Scope: Green=Control, Orange=Error, Blue=Reference and the yellow one is the response of the system, which stabilizes in 1 second.

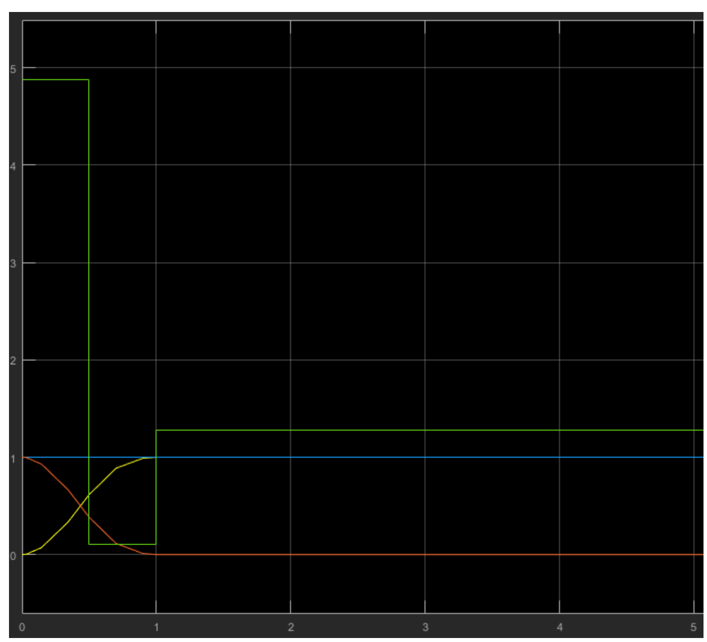

***Figure 21*** *: Dead Oscillations Scope*

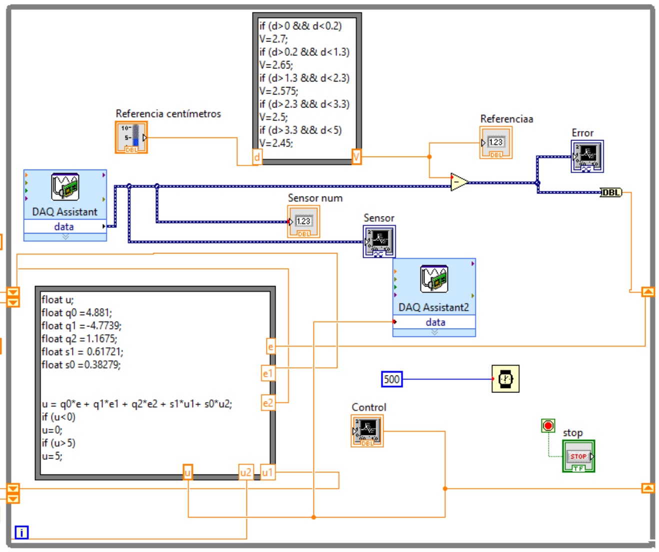


***Figure 22*** *: LabView Dead Oscillations Diagram*

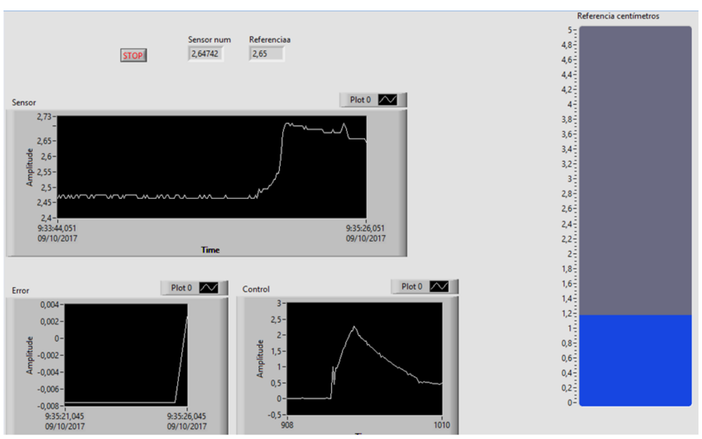


***Figure 23*** *: LabView Dead Oscillations GUI*

It is observed that the control signal, when it has a positive slope, only goes to two values (from 0 to 1, and from 1 to 2.3 )since it stabilizes at 2 samples, as can be seen in the simulink scope. And you can also see that the sensor output has an overdamped response when driving from 2.47 to 2.69. In the screenshot, it is then observed that the reference is lowered in order to appreciate the change in the control signal, which in this case begins after reaching the value of 2.5 and takes a negative slope.

## Frequency compensator

The discretized plant for a sampling time of 0.15 seconds is as follows:

$$\frac{0.01568\ z + 0.01359}{z^2 - 1.614\ z + 0.6511}$$

Then an it of 0.7 and a settling time were taken.

The Bode plot and the root loci that are characteristic of the designed compensator are shown below.

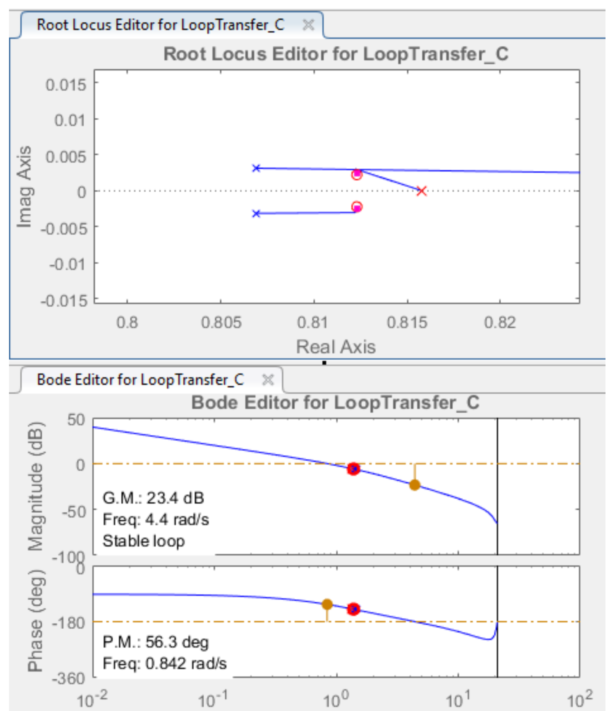


***Figure*** *24: Bode plot and root locus Frequency compensator*

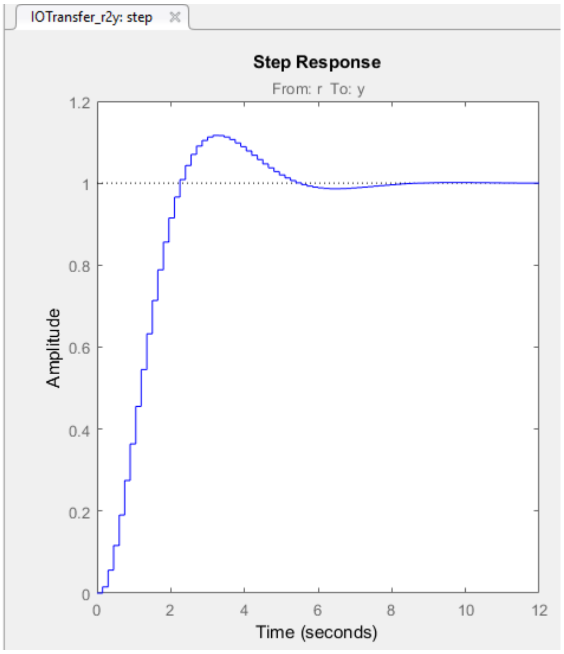


***Figure 25*** *: Visualization of the expected response Frequency compensator*

```
Name: C
Sample Time: 0.15
Value:
  0.97496 (z^2 - 1.625z + 0.6598)
  -------------------------------
         (z-1) (z-0.8158)
```

This is the transfer function of the compensator that will then be exported to the workspace in order to obtain results using the step and matlab feedback.

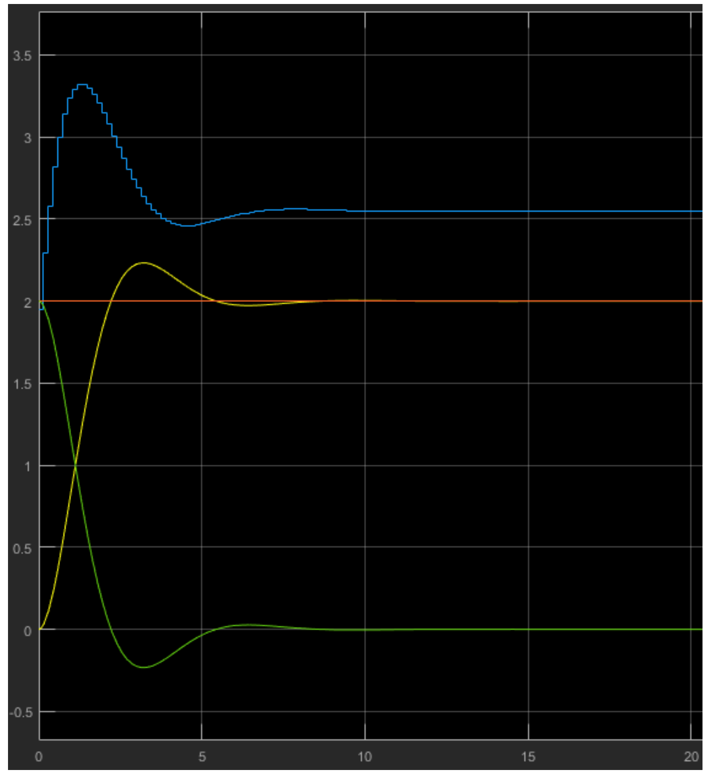


***Figure*** *26: Scope simulink Frequency compensator*

In the previous figure we have the simulation carried out in simulink of the frequency compensator, where the blue signal corresponds to the control signal, green is the error, the orange signal is the reference and the yellow signal is the system response.

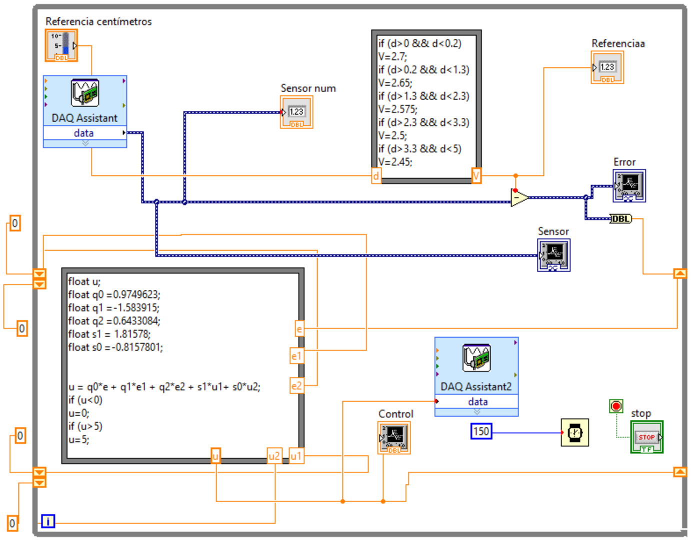


***Figure 27*** *: LabView Frequency Compensator Diagram*

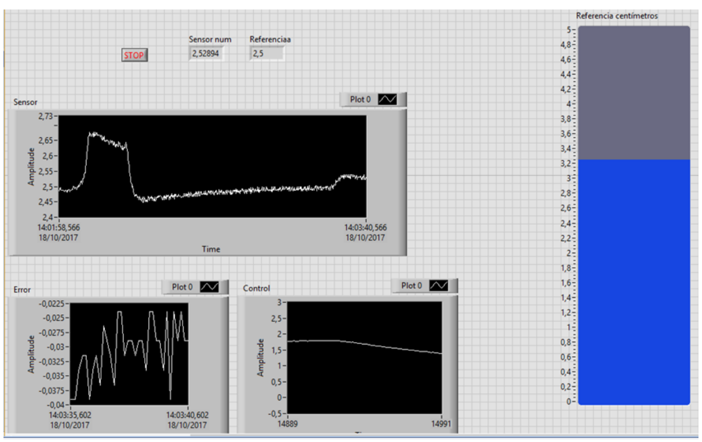


***Figure 28*** *: LabView Frequency Compensator GUI*

It is observed that the sensor response is underdamped when the reference is varied from 2.65 to 2.5, having an overshoot that reaches the value of 2.45 since the establishment time of the plant and as can be seen in the scope of the simulink, is that in 10 seconds it stabilizes completely.

**Servo**

system System matrices using Matlab's tf2ss function.

```
A =
     -2.8610   -2.0470
      1.0000         0
B =
       1
       0
C =
           0    1.6060
D =
       0
```

Desired polynomial

```
Zd =
z^3 - 1.687*z^2 + 0.7522*z - 0.03352
```

Packed matrices

```
Gemp =
    0.4492   -0.3579         0
    0.1748    0.9494         0
    0.2808    1.5248    1.0000
Hemp =
    0.1123
    0.0437
    0.0702
```

Sampling time 0.25, feedback constants, integral constant and the observer constant.

```
t =
    0.2500
Krealim =
    3.7259    5.9835
Ki =
    0.4447
Kee =
    0.4959
    0.8709
```

The obtained state space matrices are assembled.

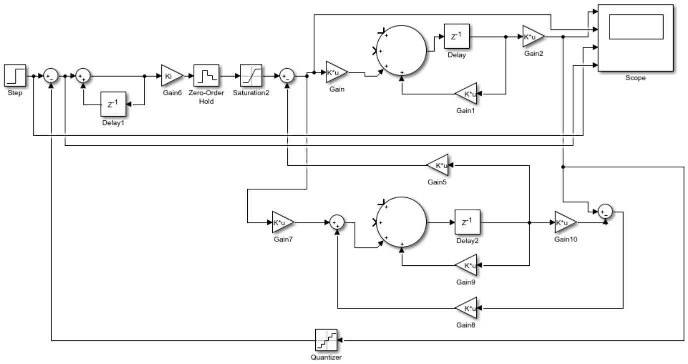


***Figure 29*** *: Servosystem simulink diagram*

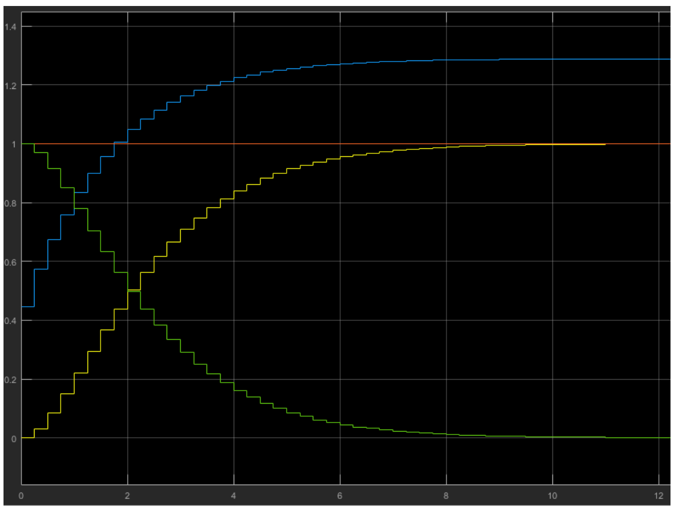


***Figure 30*** *: Servosystem simulink Scope*

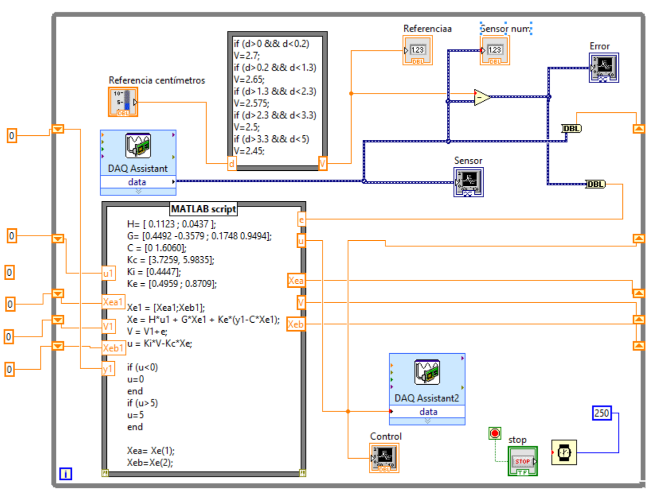


***Figure 31*** *: Servosystem LabView diagram*

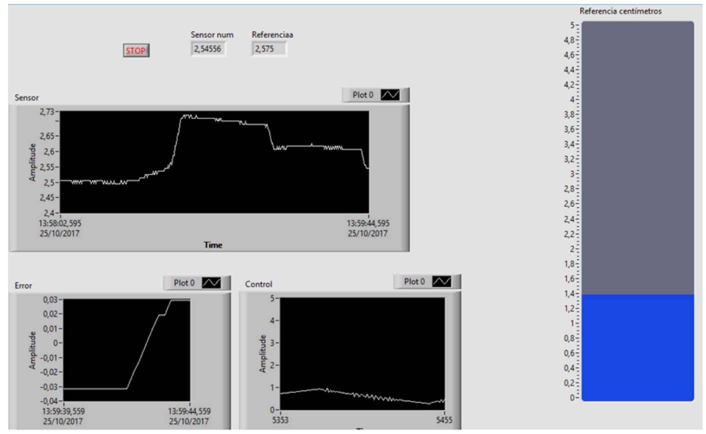


***Figure 32*** *: Servosystem LabView GUI*

## 4. CONCLUSIONS

- The servosystem and dead oscillation control were the controllers that performed best of the 7.
- The discretized PID controller performed in the simulation but not in the implementation because the constants obtained were very large and this caused the control signal to saturate too much.
- It was possible to identify how the control by dead oscillations improved with respect to the plant cancellation, since in the latter the control signal presented small oscillations before reaching its point of convergence, with dead oscillations that oscillatory effect was eliminated.
- It was possible to use SISOTOOL for the design of the frequency compensator manipulating the establishment time and the damping coefficient to obtain an easy-to-use controller in LABVIEW.
- A new transfer function for the plant was not identified, since the identification obtained in the previous practice responded adequately for all the controllers.

## 5. REFERENCES

[1] K. Ogata, Discrete Time Control Systems, University of Minnesota, Prentice Hall, 2ed, 1996.

[2] Introduction to Operational Amplifiers [online] Available at:

http://www.forosdeelectronica.com/f16/intro-operational-amplifiers-mini-tutorial-408/

[3] Contrafasico or push-pull amplifier [online] Available at:

https://unicrom.com/amplificador-contrafasico-o-push-pull/