

# Laboratorio 8: Control discreto con microcontrolador de un polipasto.

Galvis, David. López, Daniel.

{u1802584,u1802530}@unimilitar.edu.co

Universidad Militar Nueva Granada

*Resumen---* Para esta práctica de laboratorio se halla la función de transferencia de una planta, para un sistema que permita el control de posición de una masa, utilizando un polipasto de reducción de  $\frac{1}{4}$  de la fuerza ejercida por el motor. El objetivo principal es controlar mediante un PID, diseñar y discretizar el controlador, e implementarlo en un microcontrolador que para este caso será la plataforma Arduino UNO. Se visualiza en una interfaz gráfica de usuario del software la posición de la masa, la referencia, la señal de control y la señal de error.

*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  $\frac{1}{4}$  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.

*Palabras clave---* Discretización, Microcontrolador, Control Digital, Interfaz gráfica de usuario, Comunicación serial.

*Objetivo General---* Diseñar y construir un sistema que asegure la posición de una masa anclada a un sistema de polipasto mediante el uso de microcontroladores.

*Objetivos específicos--*

- \* Diseñar un sistema de polipasto que permita el ascenso y descenso de una masa, este debe tener como actuador un motor en DC.

- \* Identificar el modelo matemático que representa el funcionamiento del prototipo.

- \* Diseñar un PID discreto que asegure la posición de la masa.

- \* Implementar el controlador en discreto mediante la utilización de un microcontrolador, seleccionando la referencia mediante una interfaz de usuario y mostrando el estado del sistema en forma gráfica.

## 1. INTRODUCCIÓN

### Marco teórico:

Dentro de los procesos de control, la mayoría de veces, la integración de ellos al mundo real se hace a través de sistemas digitales. El avance actual de sistemas de procesamiento de bajo costo, ha permitido tener casi en cualquier proceso un sistema de control discreto.

El avance en la miniaturización de los sistemas de procesamiento, la mejora de su velocidad en ejecutar instrucciones, permiten hoy en día la implementación de controles en la mayoría de sistemas que interactúan en nuestro entorno. Cuando se busca programar un control discreto en microcontroladores, computadores, PLC's, se debe proceder de un análisis e interpretación del comportamiento del sistema a controlar.

Para el planteamiento de la ecuación en diferencias (la que permite integrar el control

discreto a un sistema de procesamiento), se puede partir de un diseño en continuo y aplicar alguna técnica de discretización para el control hallado, o discretizar el modelo del sistema y diseñar en el mundo discreto el control.

Para el primer método, se sigue cualquiera de los pasos que se han tratado en las prácticas anteriores o teorías de control continuo conocidas. Con el control continuo cumpliendo los criterios de diseño, se procede a realizar su discretización y a partir de esta ecuación discreta (en  $Z$ ), se plantea la ecuación en diferencias. En este procedimiento, es deseable que el método que se use para la discretización, mantenga la mayor parte de las características temporales y frecuenciales.

Existen varias teorías que permiten encontrar la representación discreta. Una puede ser que  $s$  se transforma en  $\frac{1-z^{-1}}{t_m}$ , donde  $t_m$  es el tiempo de muestreo. Es importante para la anterior ecuación, seleccionar un buen tiempo de muestreo. En la literatura se encuentra que  $tm$  está relacionado con la clase de sistema que se quiere controlar, con el tiempo de respuesta o su ancho de banda.

Por el segundo método, el primer paso es encontrar un modelo discreto del sistema y a partir de este, aplicar una de las muchas maneras que existen para calcular los coeficientes de un regulador discreto. El diseño de un regulador PID discreto, cuya ecuación está representada por  $\frac{q_0+q_1Z^{-1}+q_2Z^{-2}}{1-Z^{-1}}$ , puede ser enfrentado por la técnica asignación de polos [1].

Sistemas de estabilización rápida, requieren sistemas de control de respuesta rápida y a su vez, exigen electrónica o actuadores mecánicos adecuados a esta dinámica. Dentro de la teoría necesaria para el diseño de este tipo de control, se tiene la adecuada selección del tiempo de muestreo  $tm$ . La generación de la señal de control involucra varios pasos: se debe hacer la adquisición de la variable de salida (señal sensada), acondicionamiento o filtrado de ella,

cálculo de la señal de control y su puesta al exterior (conversión de digital a analógico). El driver o actuador es un elemento importante en la integración de la señal de control y el sistema.

### ¿Qué es un microcontrolador?

Un microcontrolador es un circuito integrado que en su interior contiene una unidad central de procesamiento (CPU), unidades de memoria (RAM y ROM), puertos de entrada y salida y periféricos. Estas partes están interconectadas dentro del microcontrolador, y en conjunto forman lo que se conoce como microcomputadora. Se puede decir con toda propiedad que un microcontrolador es una microcomputadora completa encapsulada en un circuito integrado.



Figura 1: Plataforma arduino con microcontrolador ATMEL.

## 2. MATERIALES

- \* Bata blanca.
- \* Planta: Motor DC, Soporte, poleas, cuerda, masa, potenciómetro lineal.
- \* Control digital: Tarjeta de adquisición DAQ de National Instruments, Labview 2015.

## 3. PROCEDIMIENTO

Solución del **cuestionario**:

*¿Cómo se hace y se implementa la comunicación serial entre un microcontrolador y un PC?*

A- Para realizar la comunicación serial entre un

PC y un microcontrolador, se configura un canal de comunicación conocidos como puertos COM, que permiten realizar esto, hay diversos software que permiten crear una interfaz de usuario al momento de realizar una comunicación serial, como es el caso de Matlab, LabView, Netbeans, C++ entre otros.

*¿Cómo se garantiza el almacenamiento de los datos recibidos y enviados por el puerto, tal que, permitan posteriormente reconstruir el estado de un sistema?*

Se puede garantizar mediante el almacenamiento de los datos de salida en una variable, y que corresponda a un tiempo de muestreo en el que cual transcurrió o se presentó esa salida.

### Práctica

Se tiene entonces que la función de transferencia de la planta del sistema con polipasto es la siguiente:

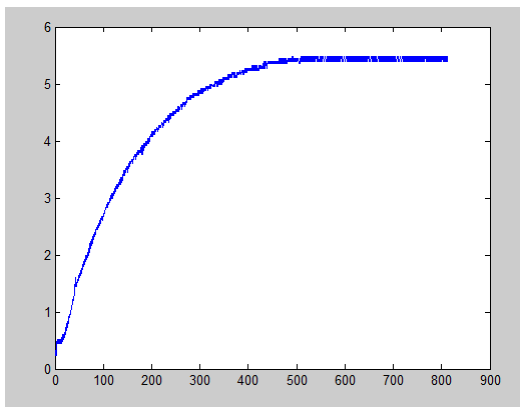


Figura 2: Respuesta de la planta en lazo cerrado

Así se obtiene la graficación de la respuesta de los datos obtenidos por el archivo en excel del osciloscopio.

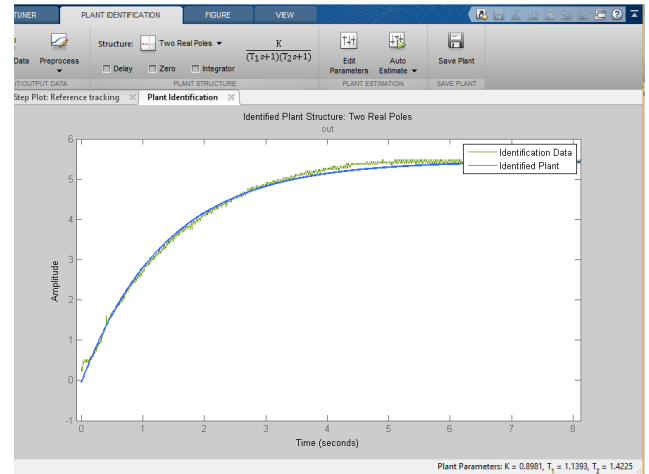


Figura 3: Aproximación de la función de transferencia de la planta

Fue obtenida utilizando la herramienta "identify new plant" de la herramienta del command window de Matlab "pidtool".

Aproxima el resultado obtenido del osciloscopio, a una función de transferencia de una planta que se comporte igual, para esta caso

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

La simulación realizada en MATLAB por diagrama de bloques se muestra en la siguiente figura:

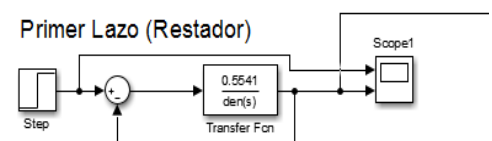


Figura 4: Simulación de la función de transferencia en lazo cerrado

Cuya respuesta se ve evidenciada en la **figura 5**.

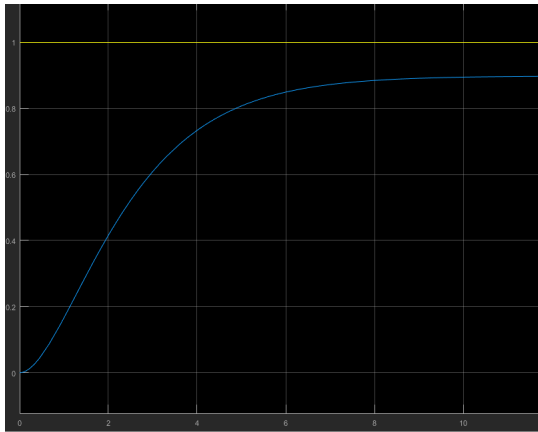


Figura 5: Respuesta en lazo cerrado de la planta

Para el diseño del controlador en continuo se observa que el sistema es de tipo 0, segundo orden, por lo que se decide implementar un PID; tal que el sistema presente los siguientes parámetros:

- Tiempo de establecimiento = 10s
- Coef. de amortiguamiento = 0.8
- Criterio de estabilización = 2%

Con esto se busca un polinomio deseado:

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

Y consecuentemente el polinomio característico del sistema controlado:

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

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

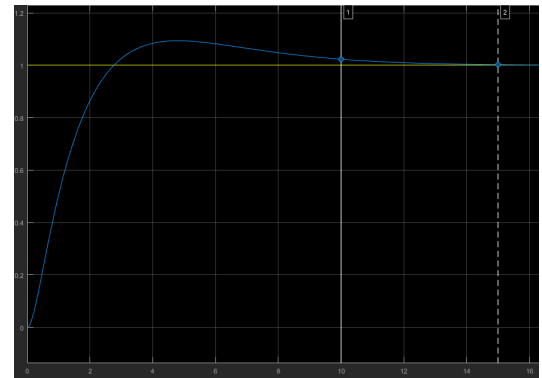
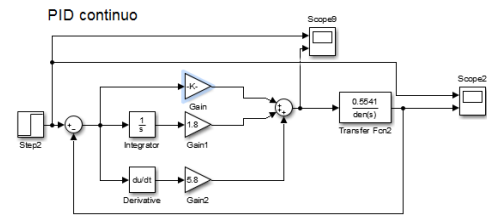
Igualando el polinomio característico con el deseado y despejando las constantes del controlador se da que:

$$Kp = 6.11$$

$$Ki = 1.8$$

$$Kd = 5.8$$

Simulando los resultados se da lo mostrado en la **figura 6**.



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

Figura 6: Respuesta del sistema con un controlador en tiempo continuo

Una vez verificado el correcto funcionamiento del controlador diseñado en continuo, se procede a discretizar, se decide utilizar el método de euler en atraso en donde la equivalencia dada es la siguiente:

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

Para el tiempo de muestreo a utilizar se decide implementar el criterio de  $tr/30$ , con esto:

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

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

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

Esto nos da un tiempo de muestreo de 0.277 s

Continuando con la discretización del control, se da que las equivalencias correspondientes son:

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

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

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

De forma tal que en controlador en discreto de forma general queda de la siguiente forma:

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

Para el caso presentado, se da que:

$$q0 = 27.54$$

$$q1 = -47.98$$

$$q2 = 20.93$$

Ya obtenidas las constantes del controlador se procede a implementar la ecuación recursiva resultante en el microcontrolador.

```
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);
}
```

Figura 7: Declaración de pines y 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);
}
```

Figura 8: Estructura main del programa

Como se observa, el valor de la referencia es determinado por una señal de voltaje analógica proveniente de un potenciómetro conectado al pin A0 de la placa.

La interfaz diseñada en el software LabView es la siguiente:

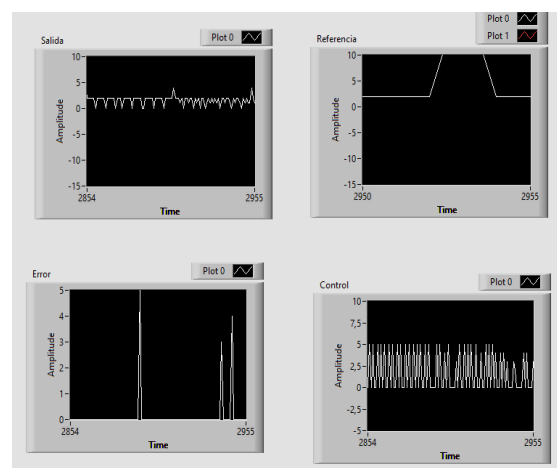


Figura 9: Interfaz de usuario en NI LabView

Donde el diagrama de bloques funcional se observa en la siguiente figura:

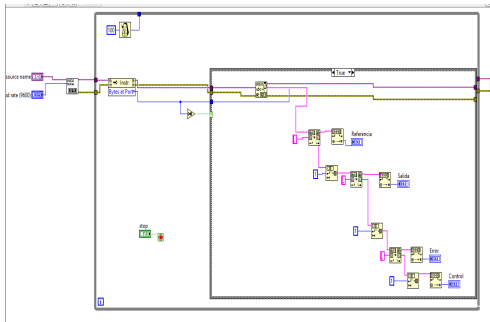


Figura 10: Diagrama de bloques implementado en LabView

De izquierda a derecha el funcionamiento es el siguiente:

- Se identifica el puerto serial al cual está conectada la placa arduino, se determinan los baudios a trabajar.
- Se reconoce la comunicación y se obtienen los parámetros del canal de transmisión y recepción.
- Al bloque de lectura serial llega la información enviada, mostrada en la **figura 10**, en donde se descompone en los 4 valores requeridos, mediante bloque de búsqueda y segmentación de cadenas de caracteres y transformación de cadenas a valores numéricos decimales, que posteriormente se visualizan.
- Se cierra comunicación en caso de ser detenido el ciclo de trabajo.

Ahora, para la segunda parte de esta práctica, se pide realizar un controlador desde el tiempo discreto, para esto, primero se discretiza la planta de trabajo, para esto se re-implementó el método de euler en atraso, con un periodo de muestreo de 0.45 segundos, dando como resultado lo siguiente:

$$\frac{0.5541}{s^2 + 1.587s + 0.617} = \frac{0.04478z + 0.03532}{z^2 - 1.481z + 0.4896}$$

Comparando ambas lados de la equivalencia nos da lo expuesto en la siguiente imagen:

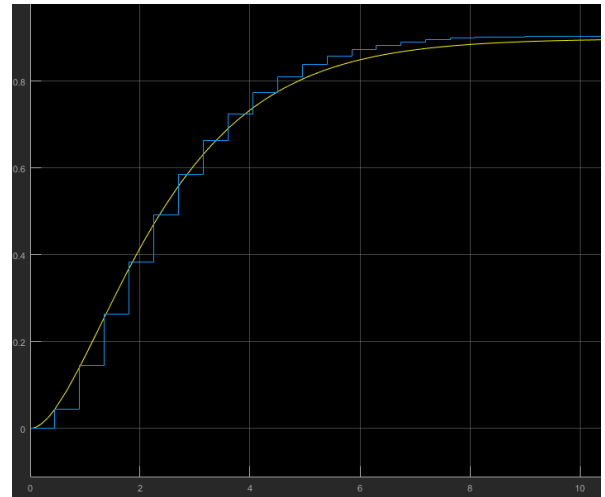


Figura 11: Sistema continuo comparado con sistema discreto.

#### 4. ANÁLISIS DE RESULTADOS

- De lo correspondiente al diseño en continuo, para su posterior discretización, se muestra que el controlador en el dominio S, como se ha comprobado en prácticas y simulaciones anteriores, responde de manera adecuada y cumple con los criterios para los cuales fue diseñado, esto se comprueba al observar la **figura 6**, en donde la señal de salida alcanza un error del 2.3% en un tiempo de 10 segundos. La señal de control y error presentadas en el sistema controlado son las siguientes:

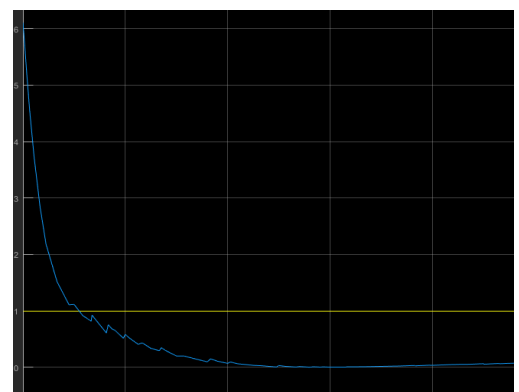


Figura 12: Señal de control y error en tiempo continuo.

- Para la discretización del controlador se implementa el siguiente diagrama en

bloques con su respectiva respuesta.

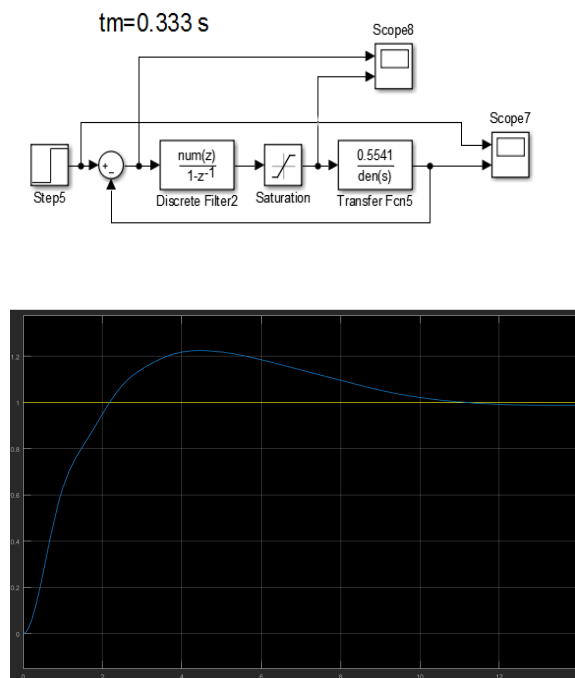


Figura 13: Respuesta del sistema discreto.

La respuesta observada presenta ciertos desaciertos respecto a la señal en continuo, esto puede darse debido a la selección del tiempo de muestreo, el método de discretización y en consecuencia los coeficientes del filtro a implementar. La señal de error y la de control se presentan a continuación:

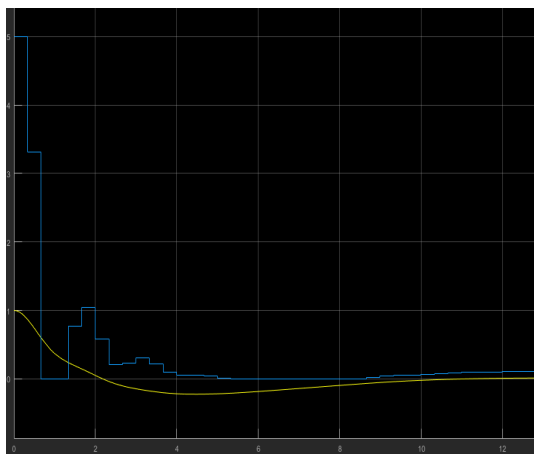


Figura 14: Señal de error y control en tiempo discreto

- Implementando la interfaz con el micro-controlador, y empezando la transmisión y recepción de datos se

observan los siguientes resultados:

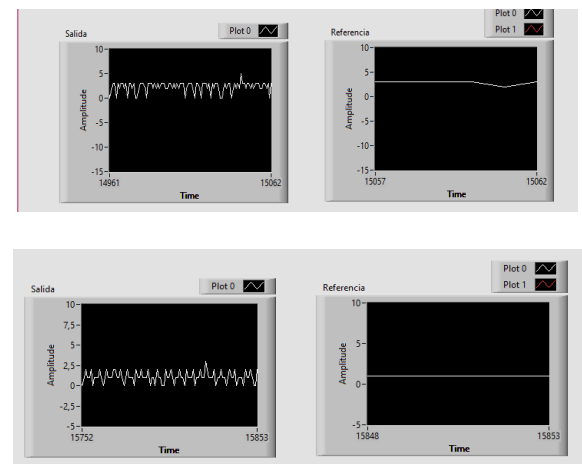


Figura 15: Interfaz gráfica del sistema REF=1

Como se observa, la señal de salida tiende a seguir el valor de la referencia, presenta oscilaciones debido a la influencia de factores externos que hacen intermitente o variable los valores tomados.

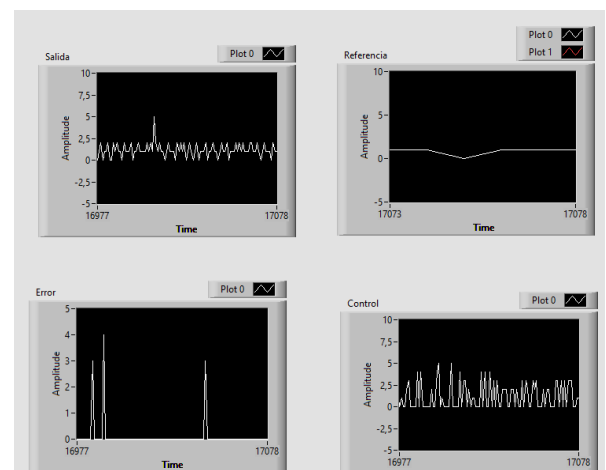
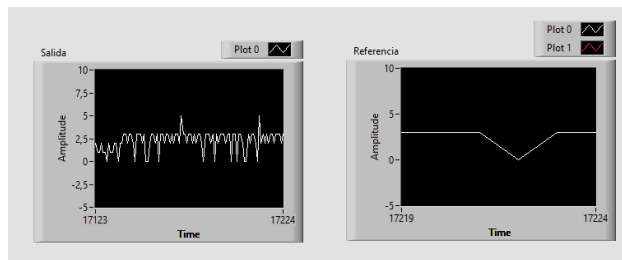


Figura 16: Interfaz gráfica de usuario.

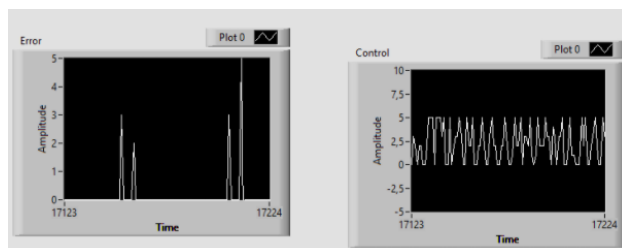
En la anterior imagen, se observa que en la interfaz la señal de error se mantiene en su mayoría en cero, sin embargo, debido a oscilaciones producidas por el valor de la referencia enviado por el potenciómetro. El valor de la salida sigue el valor de referencia.





*Figura 17: Salida 3 Volts (Izquierda) y Referencia 3 Volts(Derecha).*

**NOTA:** La imagen izquierda es la salida del sistema, Y la imagen derecha es el valor de referencia al cual se quiere llegar, la salida presenta oscilaciones en el estado estable y caídas a cero debido a la alimentación de voltaje vía USB del microcontrolador Arduino, sin embargo se observa que el error es menor al 10%.



*Figura 18: Señal de error (Izquierda) y señal de control (Derecha).*

Para la anterior imagen, se observa que la señal de error en un punto en el que la referencia baja, aumenta.

## 5. CONCLUSIONES

\* El control continuo-discreto realizado para esta práctica de laboratorio, respondió de la manera adecuada en cuanto a controlar el sistema, sin embargo se presentan algunas oscilaciones en la señal de control, salida, y referencia.

\* Un factor influyente en estas oscilaciones es el tiempo de muestreo, ya que se da la posibilidad de que el sistema de toma de datos (LabView) tome un dato durante el tiempo en el cual el micro está procesando la información y no la ha enviado por el canal de comunicación.

\* Aunque el sistema controla de forma correcta,

no es la más adecuada, ya que de ser implementados en sistemas en donde se requiera de una gran precisión, dichas oscilaciones presentan inconvenientes, por eso si en una aplicación que no sea de práctica (como esta sesión) se recomienda usar un microcontrolador con un oscilador de rango más alto.

## 6. REFERENCIAS

- [1] K. Ogata, Sistemas de control en tiempo discreto, Prentice hall International, 1996.
- [2] N. Instruments, «National Instruments,» [En línea]. Available: <http://www.ni.com/data-acquisition/what-is/esa/>. [Último acceso: 15 06 2016].