

# Control de Levitador Neumático

Carlos A. Moreno (2011203042)  
dte\_cmoreno639@pedagogica.edu.co  
Sistemas de Control II

**Abstract**—*This article is the result of a laboratory report of Control II course, it tries to explain how a small scale pneumatic levitation is made, using examples and techniques of digital control, this is because the use and implementation of this type of controllers are economic and practical in a real plant, although it has some disadvantages according to the controller, the microprocessor to be used and the implementation of the sensors since they must be as precise as possible to avoid possible errors.*

**Keywords:** Digital Control, PID, Linearization, Identification, Plant.

**Resumen**—Este artículo es un informe de laboratorio para curso de Control II, pretende explicar cómo se hace un levitador neumático de pequeña escala, empleando ejemplos y técnicas del control digital, esto se debe a que el uso y la implementación de este tipo de controladores son económicos y prácticos en una planta real, aunque posee algunas desventajas según el controlador, el microprocesador a usar y la implementación de los sensores ya que deben ser lo más precisas posibles para evitar posibles errores.

**Palabras Claves:** Control Digital, PID, Linealización, Identificación, Planta.

## I. INTRODUCTION

En ocasiones, las perturbaciones tanto externas como internas presentes en los procesos pueden lograr desestabilizar un sistema de control en lazo abierto, los cuales requieren mantener ciertas condiciones de estabilidad para garantizar la calidad del producto.

En este caso los sistemas de control en lazo cerrado nos pueden permitir de alguna manera estabilizar un sistema dinámico, de esto depende el tipo de controlador, los sensores y los actuadores que estén trabajando en el sistema, al lograr una correcta linealización e identificación del mismo se puede garantizar un trabajo y un rendimiento a ciertos criterios de operación.

La planta o sistema que se planteó para este proyecto es un Levitador Neumático, controlado desde una tarjeta PSoC 5LP, (en el transcurso de este artículo se hablara de la implementación de esta), que permite generar señales para operar el aire que circula dentro un tubo, por otra parte esta tarjeta también permite adquirir los datos suministrados por un sensor ultrasonido, para poder observar y determinar la dinámica de un objeto dentro del tubo a partir de ciertos impulsos y señales.

## II. OBJETIVOS DE LABORATORIO

### Objetivo General

Identificar el comportamiento de una esfera dentro de un tubo con aire, junto con los parámetros que le afecta, haciendo que la masa permanezca estable bajo una acción y, quede suspendida de manera vertical en un punto de referencia específico.

### Objetivos Específicos

1. Realizar el montaje a pequeña escala de la planta.
2. Hacer medición de la posición según los datos suministrados por el sensor para poder linealizarlo.
3. Observar el comportamiento de la planta ante un impulso y una señal sinusoidal para poder hacer la identificación de la misma.
4. Implementar el diseño de un controlador digital para ajustar la esfera en un punto de referencia.

## III. MARCO TEÓRICO

### III-A. Dispositivos electrónicos usados para el Proyecto

#### III-B. ARM® Cortex®-M3 CY8CKIT-059 PSoC® 5LP

El procesador ARM® Cortex®-M3, es el líder en la industria de procesadores de 32-bits, debido a que los dispositivos que lo usan son de baja potencia, tienen un costo bastante bajo a pesar de las múltiples aplicaciones embebidas que se pueden realizar con dispositivos que manejen este procesador, este procesador es ampliamente configurable e ideal para aplicaciones donde se requiere controlar la memoria y el espacio que ocupa el dispositivo. Como se puede apreciar en la figura 1.



Figura 1. Ilustración de la tarjeta microcontrolador usada en el proyecto

El PSoC programable está compuesto por una serie de periféricos analógicos y digitales de alta precisión y programables con una CPU ARM® Cortex®-M3 en un solo chip, tiene bloques programables, esto se refiere a que desde el

software de PSoC se pueden intercomunicar diferentes bloques como son comunicación, PGA, pines, filtros, comparadores, etc., además de estar diseñado para flexibilidad, permitiendo el acceso a todas las entradas y salidas de esta tarjeta, conectividad USB 2.0, y acceso a un hardware bastante pequeño.

### III-C. Sensor UltraSonido HC-SR04

Este es un sensor de distancias por ultrasonidos capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 2 a 450 cm (esto depende del ambiente de trabajo donde este se encuentre y la programación que posea).



Figura 2. Ilustración del sensor para medir la distancia de la esfera

Contiene toda la electrónica encargada de hacer la medición ver figura 2. Su uso es simple y práctico, consiste en enviar el pulso de arranque y medir la anchura del pulso de retorno. De precisión medianamente buena y bajo precio por lo que está reemplazando a los sensores polaroid en los robots más actuales.

### III-D. Actuador Motor de corriente continua

Este dispositivo electrónico trabaja de 12 a 24 voltios. en corriente continua, consume en su máxima carga aproximadamente 1.5 amperios, en el proyecto se usó como actuador para enviar aire por dentro del tubo, se implementó con un PWM para variar su velocidad y así poder manipular el aire que circulaba. En la figura 3 se puede apreciar el motor con unas hélices de forma horizontal rodeando el eje esto hace que pueda manipular el aire y enviarlo a la zona deseada.



Figura 3. Ilustración del sensor para medir la distancia de la esfera

### III-E. Identificación del Sistema

**III-E1. Método basado en la respuesta a escalón:** Aunque existen diversas formas de identificar un sistema dinámico, uno de ellos es usando la fórmula de sistemas de segundo orden ver la ecuación (1), que permite a partir de sus variables como:  $\zeta$ ,  $\omega_n$  y  $k$  ajustar el comportamiento a una ecuación, hay métodos mejores que a partir de la adquisición de datos permite computar y generar resultados más aproximados y reales.

$$H(s) = \frac{k\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (1)$$

Aunque se trató de usar este método, la planta no era lo suficientemente estable para recrear la ecuación usando esta fórmula, pues la esfera oscilaba de manera diferente y el margen de error variaba, haciendo que la identificación fuera más errada.

El algoritmo implementado para identificar este tipo de sistemas es el que se muestra en la figura 4, que se implementa de manera analítica para poder identificar los parámetros de la función de transferencia de la ecuación (1), también funciona para el método de **Anderson** que es similar al de ecuación (1) pero con polos reales.

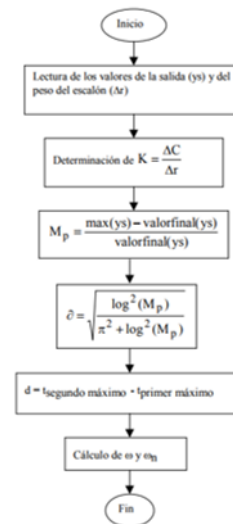


Figura 4. Algoritmo de identificación de sistemas oscilatorios

**III-E2. Método basado en ARX:** Es una representación polinomial de una función de transferencia, esta se encarga de analizar los datos adquiridos y computarlos ante una señal introducida en la entrada del sistema, en las ecuaciones planteadas ver ecuaciones (2), representan un conjunto de polinomios que actúan bajo el operador de retraso  $q^{-nk}$  y permiten crear un conjunto polinomial de las funciones  $G(q)$  y  $H(q)$ .

$$G(q) = \frac{q^{-nk}(B(q))}{A(q)} \quad (2)$$

$$H(q) = \frac{1}{A(q)}$$

Despejando y obteniendo los valores de la ecuaciones:(2), se pueden tener los valores en términos de un polinomio de  $na, nb$  orden, donde  $nk$  es el número de retrasos entra la entrada y la salida del sistema, el polinomio que queda es el resultado del denominador y numerados de la función de transferencia identificada como de muestra en la ecuación:(3).

$$A(q) = 1 + a_1q^{-1} + a_{na}q^{-na}$$

$$B(q) = b_1 + b_2q^{-1} + b_{nb}q^{-nb+1}$$
(3)

Este método es que se usó para identificación de la planta de manera computacional, para su eficiente trabajo los datos adquiridos deben ser lo más limpios y puros de ruido posible debido que este método trabaja por medio de iteraciones y relaciones del dato pasado con el dato presente, adicionalmente matlab me expresa los resultados ver ecuación: (9) de la sección de procedimientos e implementación, en términos de una función de transferencia discreta.

#### IV. PROCEDIMIENTO E IMPLEMENTACIÓN

La planta que se trabajó en este documento es la que se muestra en la figura 5, consta de: un tubo de acetato, un sensor de distancia, un micro controlador, un potenciómetro que se implementó como guía de referencia, finalmente un actuador, que se dio las características en el marco teórico de algunos de los elementos importantes del proyecto.



Figura 5. Fotografía de la planta Funcionando en un punto de referencia

##### IV-A. Linealización del Actuador

Se hace la linealización del actuador teniendo en cuenta el punto de operación del motor, es decir, la zona muerta o donde

le motor no responde, que esta ubicada en el aproximadamente 25,49% del PWM generado por el microcontrolador.

$$M_{pwm} = \left( \frac{194}{5} \right) Voltage + 66; \quad (4)$$

Se tiene en cuenta que 255 es el equivalente en a tener un resultado de 0 a 5 voltios entregados por la tarjeta de desarrollo PSoC esto en el rango de operación de 8 bits y por tal motivo se hace la conversión como se muestra en la ecuación: (4) donde  $ref$  es igual a las variaciones deseadas para que la pelota se ubique medianamente en un punto y  $M_{pwm}$  es el resultado que se entrega al generador de PWM de la tarjeta para que actué el motor, que me permite hacer trabajar el motor desde la zona mínima de operación hasta su máximo poder.

Cuadro I  
TABLA DE VOLTAJE VS ALTURA DE LA ESFERA

Voltios (X)	Altura (Y)
0.5	1.2
0.7	3.2
0.9	4.4
1.1	5.3
1.3	6.07
1.5	6.8
1.7	7.2
1.9	9.03
2.1	9.1
2.5	10.85

En el cuadro I, se muestra la tabulación de resultados entre los voltajes entregados por el pwm, y la altura a la que llega la esfera estabilizándose totalmente. Nótese que cada 0.2 pasos varía la el voltaje que se entrega al PWM como se muestra en la ecuación (4), esto se debe a que el motor generaba cambios después 0,2voltios.

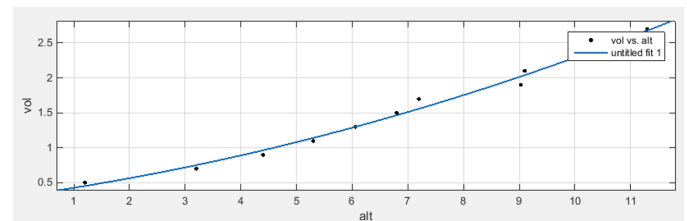


Figura 6. Fotografía de la planta Funcionando en un punto de referencia

La figura 6, muestra el grafico de los datos tabulados en color negro son los puntos, y la línea azul es la linealización, que no es nada más que un polinomio de segundo orden como se muestra en la ecuación: (5) que permite enviar datos en modo de altura al motor.

$$A_{motor} = 0,0087Alt^2 + 0,1107Alt + 0,4 \quad (5)$$

A continuación se muestra el código de implementación de la ecuación para el actuador.

```
float planta(float up){
    float yp=0.0, alt=0.0;
    alt=0.0087*up*up+0.1107*up+0.4;
```

```

yp=sensor;
actuador(alt);
return yp;
}

```

#### IV-B. Linealización del Sensor

Se procede a verificar los puntos de referencia con los resultados entregados por sensor, este tiene un rango de operación eficaz después de los 4cm hasta los 300cm en ambientes ideales y con objetos planos, en el caso de la planta se usó dentro de un tubo con aire lo que se hizo fue enviar pulsos al motor y luego medir los resultado medidos en los sectores donde se pudo controlar la esfera y capturar un dato como se muestra en el cuadro II.

Cuadro II

EN ESTA TABLA SE MUESTRA LA ALTURA REAL OBSERVABLE VERSUS LA ALTURA MEDIDA POR EL SENSOR

Sensado (X)	Altura Observable (Y)
38.5	1.0
37.5	2.0
33.5	3.0
29.3	4.0
21.1	5.0
19.5	6.0
18.5	7.0
12.5	8.0
10.1	9.0
9.8	10.0

Para la eliminación del ruido, se usó una tapa de silicona que ayudo a ajustar el sensor al chasis y, evitar que este Vibrara por las ráfagas de aire, sin contar que adicionalmente se aplicó un filtro descreo de primer orden de aproximadamente 0.35 milisegundos de respuesta como se muestra en la ecuación 6 usando el método ZOH, para que detectara la esfera ayudara con el ruido y a mejorar la respuesta del sensor.

$$H(z)_{filtro} = \frac{0,9989}{z - 0,001052} \quad (6)$$

Las frecuencias de operacion deben estar por debajo de los 35 milisegundos, de lo contrario empezara a atenuar la señal ajustándola a la posición de la referencia, como se muestra en la figura 7 la relación de la magnitud y fase ante las frecuencias.

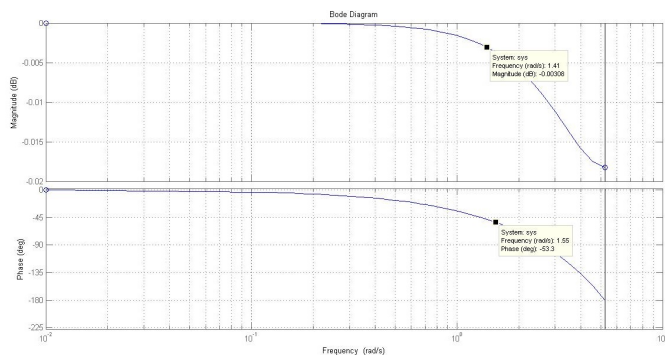


Figura 7. Diagrama de bode del Filtro implementado en el sensor

$$Alt_{real} = -0,4155 * lec_{Sensor} + 17,8; \quad (7)$$

Ahora usando los datos del cuadro II, se obtiene la ecuación (7), de esta ecuación se puede obtener el grafico de la figura:8.

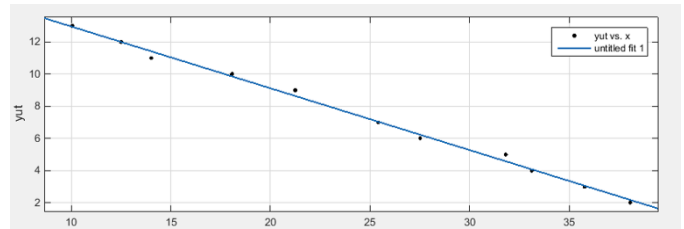


Figura 8. Diagrama de bode del Filtro implementado en el sensor

A continuación se muestra el código de implementación de la ecuación para el sensor.

```

float sensorUltraSonido(){
float y=0.0, lectureD=0.0;
while(echo_Read()==0){
triggerSal_Write(1);
CyDelay(10u);
triggerSal_Write(0);
CyDelay(1);
}
while(echo_Read()==1){}
valueconter=65535-Timer_ReadCounter();
lectureD=valueconter/58.0;
// y=-0.3843*lectureD+16.8;
y=-0.4155*filtroz(lectureD)+17.8;
CyDelay(30);
if(lectureD<11) y=14;
if(lectureD>40) y=0;
return y;
}

```

#### IV-C. Ajuste de los Puntos de Referencia con un Potenciómetro

En esta sección se ajusta la referencia (Set-Point) con un potenciómetro, para ello se procede a lineal izar los valores medidos por el pin análogo de la tarjeta PSoC, ajustándolos a unas escalas específicas como se muestra en el cuadro III, donde se muestra la relación del voltaje medido por la tarjeta y los set Pont.

$$y = 5,167 * x - 7,548; \quad (8)$$

A partir de la tabla obtenida, se puede generar el grafico de la figura9, y la comparación de la ecuación 8, se puede observar que casan los puntos con la función sin inconvenientes.

A continuación en el cuadro puede apreciar la implementación de la ecuación el código fuente.

```

int LinealizacionPot(float32 x){
int y=0;

```



Cuadro III

EN ESTA TABLA III SE MUESTRA LA REFERENCIA DESEADA OBSERVABLE VERSUS EL VOLTAJE SUMINISTRADO POR EL POTENCIÓMETRO A LA TARJETA

Voltaje Pot (X)	Referencia Observable (Y)
1.51	0.0
1.67	1.0
1.83	2.0
2.03	3.0
2.22	4.0
2.43	5.0
2.60	6.0
2.81	7.0
3.00	8.0
3.19	9.0
3.38	10.0
3.60	11.0
3.77	12.0

```

y=5.167*x-7.548;
if (x<1.51) y=0;
if (x>4.22) y=14;
return y;
}

```

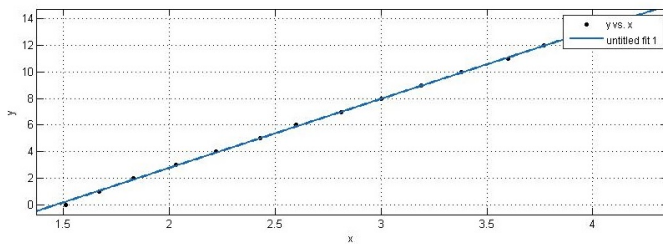


Figura 9. linealización del potenciómetro

#### IV-D. Identificación de la Planta

A continuación se procede a mandar una señal cuadrada con una frecuencia de 0,05Hz a la planta, para observar su respuesta obteniendo los resultados de la figura 10, esta muestra los estímulos en la región azul y la respuesta censada en rojo.

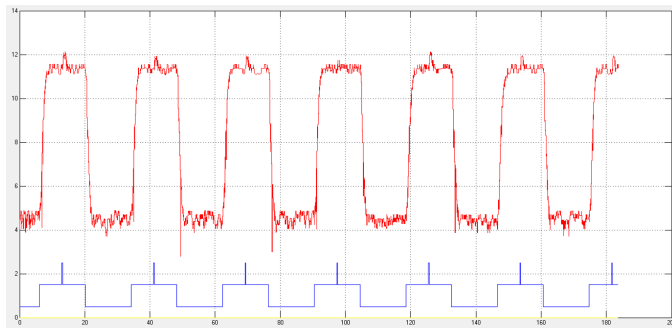


Figura 10. Datos de la planta ante una señal cuadrada

Después de hacer esto se calcula de manera computacional

usando *MATLAB* los datos obtenidos para tener la ecuación de transferencia, este software usa por defecto el método de ZOH, para la identificación de sistemas discretos con el método ARX, y se obtiene la función de transferencia de la ecuación: (9) con un margen de acierto del 92,6 %

$$H(z) = \frac{0,01635z^{-1}}{1 - 1,768z^{-1} + 0,7844z^{-2}} \quad (9)$$

Téngase en cuenta que la ecuación y el periodo de muestreo  $T_m = 0,6$ , se usó este periodo debido a que se ajusta con el tiempo de espera del sensor, para poder capturar los datos de manera eficiente.

Luego de calcular la función de transferencia se ejecuta el código fuente en el microcontrolador y se comparan los resultados obtenidos como muestra la figura 11, se puede ver que casa de manera ideal.

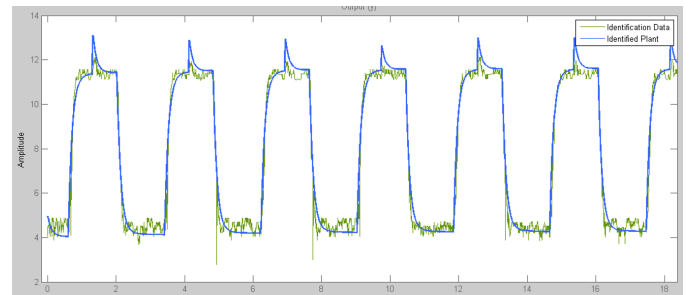


Figura 11. Comparación de la planta real y discreta

A continuación se implementa la función de transferencia en el microcontrolador para poderla comparar con la planta real.

```

float plantid(float up){
float p2=0.01635*uid_1 + 0.0*uid_2;
ypx=1.768*yid_1 - 0.7844*yid_2 + p2;
yid_2=yid_1;
yid_1=ypx;
uid_2=uid_1;
uid_1=up;
return ypx;
}

```

#### IV-E. Análisis de estabilidad

Para poder hacer este paso se tienen en cuenta tres consideraciones:

1. para que un sistema sea estable los polos en lazo cerrado deben estar dentro del círculo unitario.
2. si existe un polo en  $z = 1$  el sistema es críticamente estable, al igual que si hay dos polos complejos sobre el círculo unitario
3. la posición de los ceros en lazo cerrado no afecta la estabilidad del sistema

Si estas condiciones no se cumplen se puede decir que el sistema no es estable y, puede causar daños en la implementación del mismo sin un buen controlador.

Para ejecutar esta acción se emplea el test de jury, que permite determinar si el sistema es estable o no con diferentes condiciones haciendo la implementación más completa.

**IV-EI. Estabilidad de Jury:** Al aplicar las reglas del test para la ecuación (9) tenemos la siguiente información:

$$G(1) = (1)^2 - 1,768(1) + 0,7844 > 0$$

$$G(1) = 0,0164 > 0$$

como  $G(1) > 0$  entonces el sistema es estable en esa sección y se puede continuar:

$$(-1)^2 G(-1) = (-1)^2 ((-1)^2 - 1,768(-1) + 0,7844) > 0$$

$$(-1)^2 G(-1) = 1 * 3,5524 > 0$$

Como  $(-1)^n G(-1) > 0$  entonces el sistema es estable en esa sección y se puede continuar: Al ver que las anteriores condiciones se cumplen se procede a armar el arreglo del test donde:

las ecuaciones para conformar el arreglo son:

$$b_k = \begin{bmatrix} a_0 & a_{n-k} \\ a_n & a_k \end{bmatrix}$$

$$b_k = \begin{bmatrix} b_0 & b_{n-k-1} \\ b_{n-1} & b_k \end{bmatrix}$$

Siendo así: se forma la tabla para verificar el resto de condiciones restantes:

row	$z_0$	$z_1$	$z_2$
1	0.7844	-1.768	1
2	1	-1.768	0.7844
3	-0.3847	-4.9228	
4	-4.9228	-0.3847	

se verifican las siguientes condiciones:

$$|a_0| < a_n$$

$$|b_0| > |b_{n-1}|$$

entonces

$$|0,7844| < 1$$

$$|-4,9228| > |-0,3847|$$

**como las dos condiciones se cumplen entonces se puede afirmar que el sistema es estable en la región identificada.**

A continuación se muestra que en el gráfico de la figura 12, los polos se encuentran dentro del círculo unitario tal como se expresó en el test de jury la planta es estable aunque tiene un cero hacia el infinito.

#### IV-F. Control en la Planta

En el gráfico de la figura: 13, se pueden observar cuatro gráficos, el azul representa la respuesta de la planta, el rojo la ecuación ideal del sistema, la amarilla es el error que simplemente, es la diferencia de la referencia con la señal

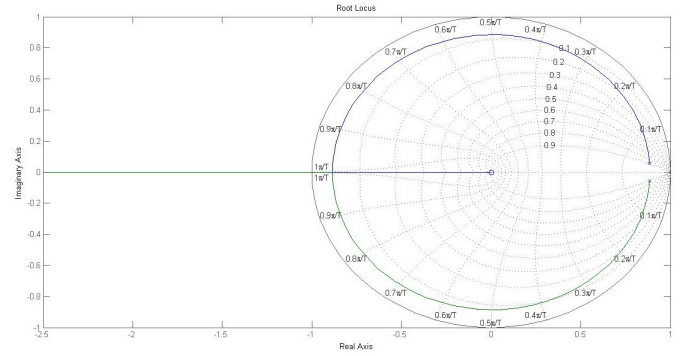


Figura 12. Diagrama de Polos en el Plano Z

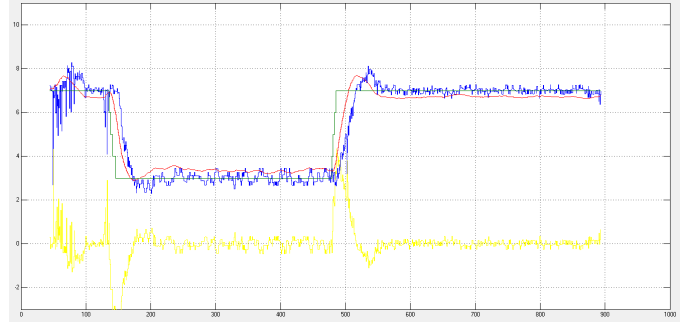


Figura 13. Gráfico Res: Planta, ecuación ideal, el error, y referencia

entregada por la planta, la verde es la referencia a la que se quiere poner el objeto que levita dentro del tubo.

El PID es la suma de tres componentes importantes, como se muestra en la ecuación (10), que si se trabaja en términos de Laplace y luego se discretiza con algún método se puede lograr la ecuación (11).

$$C(t) = K_p e(t) + K_i \int_0^t e(t) dt + k_d \left( \frac{de(t)}{dt} \right) \quad (10)$$

La función de transferencia característica del PID implementado, es la que se muestra en la ecuación: (11), es un PID estándar, a pesar que existe otros modos como el parallel, use este por la facilidad de implementación en el código fuente.

$$C(z) = K_p + \frac{K_p}{T_i} \frac{T_s}{z-1} + \frac{T_d(K_p)(z-1)}{T_s} \quad (11)$$

La función de transferencia de la ecuación (12), esta discretizada con el método de backward y da como resultado, se implementa la función de transferencia  $C(z)$  de la ecuación (12).

$$C(z) = \frac{(K_p + \frac{K_d}{T_s})z^2 + (T_s K_i - kp - \frac{2K_d}{T_s})z + \frac{K_d}{T_s}}{z(z-1)} \quad (12)$$

Para ajustar la sintonía de este controlador se usa la tabla que se muestra en la figura: 14 se zigger y nicols, esta permite tener una implementación teórica de la sintonía del sistema.

Regler	$K_r$	$T_n$	$T_v$
P-Regler	$\frac{T_s}{K_s \cdot T_t}$	-	-
PI-Regler	$0.9 \cdot \frac{T_s}{K_s \cdot T_t}$	$3.33 \cdot T_t$	-
PID-Regler	$1.2 \cdot \frac{T_s}{K_s \cdot T_t}$	$2.0 \cdot T_t$	$0.5 \cdot T_t$

Figura 14. Tabla de sintonía de un PID con zigger y Nicols

Los valores obtenidos para el sistema con un periodo de muestreo  $T_s = 0,6$  fueron los que se muestran en la ecuación (13)

$$\begin{aligned} K_i &= 0,361 \\ T_p &= 0,0423 \\ T_d &= 0,31 \end{aligned} \quad (13)$$

A continuación se muestra la implementación del PID en el microcontrolador

```
float a=0.0, b=0.0, c=0.0;
float PIDz(float error){
    float32 const kp=0.0000000084;
    float32 const kd=0.004, ki=0.0043;

    a=kp+kd/0.6;
    b=0.6*ki+kp*(2*(kd/0.6));
    c=kd/0.6;

    uk=uk_1+a*error+b*ek_1+c*ek_2;
    if(uk>13) uk=13;
    if(uk<=0) uk=0;
    ek_2=ek_1;
    ek_1=error;
    uk_1=uk;
    return uk;
}

float controlPID(int ref){
    float y=sensor;
    y = planta(PIDz(err));
    err= ref-y;
    return y;
}
```

Los valores fueron alterados para ayudar a mejorar la respuesta del sistema.

## V. CONCLUSIÓN

Identificar el comportamiento de la planta lo más eficaz posible, puede llevar a un exitoso control del sistema, en el

caso de este proyecto lo más cercano que se logro fue superior al 90 % e inferior al 93 % esto se debe a que se omitieron varios criterio por cuestiones de tiempo, se puede observar que a pesar que el error es muy bajo se alcanza a desviar la función original del sistema con la planta ideal ver la figura 11 pero más sin embargo el control PID funciona de manera eficaz.

Usando la identificación propuesta las constantes del PID, varían de manera considerable en la planta real ver ecuación 14, donde se proponen unas nuevas constantes que ayudan al rendimiento y operacion de la misma.

$$\begin{aligned} K_i &= 0,00000000841 \\ T_p &= 0,0043 \\ T_d &= 0,004 \end{aligned} \quad (14)$$

En cuanto los parámetros que permitieron identificar el sistema, se encuentra una serie de elementos de instrumentación electrónica, debido a que los sensores y actuadores no trabajan de manera ideal y toca hacer un reconocimiento de operacion, para poder hablar de identificación de un sistema, en caso de hacerse mal puede llevar a errores que son multiplicables, cuando se quiera trabajar con control serán errores difíciles de identificar y que pueden traer varios problemas en las variables dinámicas.

## REFERENCIAS

- [1] Ultrasonic Ranging Module HC - SR04. *Elec Freacks*, March 2008.
- [2] Programmable System-on-Chip. *cypress*, june 2015.
- [3] Katsuhiko Ogata. *Sistemas de Control en Tiempo Discreto*. Pearson, 2nd edition.
- [4] IMPLEMENTACIÓN DE UNA NUEVA TÉCNICA DE CONTROL DIGITAL PARA CONVERTIDORES DC-DC Y DC-AC, Universidad Nacional de Colombia, sede bogota. <https://revistas.unal.edu.co/index.php/dyna/article/view/25589>