# Práctica 1: Medición de parámetros de un motor DC

Santiago Andrés Gómez Barbón
*Universidad Distrital*
*Estudiante de Ingeniería Electrónica*
20211005034
Bogotá, Colombia
sagomezb@udistrital.edu.co

Juan Camilo Cárdenas Murcia
*Universidad Distrital*
*Estudiante de Ingeniería Electrónica*
20211005029
Bogotá, Colombia
jccardenasm@.udistrital.edu.co

Alejandro Santamaria Torres
*Universidad Distrital*
*Estudiante de Ingeniería Electrónica*
20211005097
Bogotá, Colombia
asantamariat@udistrital.edu.co

Andres Camilo Patiño Ariza
*Universidad Distrital*
*Estudiante de Ingeniería Electrónica*
20211005105
Bogotá, Colombia
acpatinoa@udistrital.edu.co

***Abstract* - This laboratory practice focuses on the implementation of a PID controller for speed and position control of a DC motor using Arduino and MATLAB. The objective is to develop and analyze the control system, evaluating its performance through real-time data acquisition and parameter tuning. The effects of the control action on system stability and accuracy will also be examined.**

## I. Introducción

El control de motores de corriente continua (DC) es una aplicación clave en la automatización y robótica, donde se requiere precisión en la regulación de velocidad y posición. En esta práctica de laboratorio, se implementará un controlador PID para controlar la velocidad y la posición de un motor DC utilizando Arduino como interfaz de hardware y MATLAB como entorno de desarrollo y simulación.

El motor ya ha sido caracterizado previamente, por lo que en esta etapa se enfocará en la implementación del control PID. Se configurará la comunicación entre MATLAB y Arduino para enviar señales de control y recibir datos en tiempo real. A través del ajuste de los parámetros PID, se buscará optimizar la respuesta del sistema, minimizando el error en la velocidad y la posición deseadas.

Esta práctica permitirá reforzar conceptos de control en sistemas de lazo cerrado, programación en MATLAB y comunicación con hardware embebido, proporcionando una experiencia integral en la implementación de controladores PID en motores DC.

### A. Objetivos

- Implementar un controlador PID en MATLAB para regular la velocidad y posición de un motor DC mediante Arduino.
- Analizar el desempeño del sistema ajustando los parámetros PID y evaluando su respuesta en tiempo real.
- Examinar la acción de control generada y su efecto en la estabilidad y precisión del motor.

## II. MARCO TEORICO

### A. Control PID

El controlador PID (Proporcional-Integral-Derivativo) es un sistema ampliamente utilizado en control automático debido a su capacidad para mejorar la estabilidad y el rendimiento de un sistema dinámico. Su ecuación general se expresa como:

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt} \tag{1}$$

donde:

- $u(t)$ es la señal de control aplicada al sistema.
- $e(t)$ es el error entre la referencia deseada y la salida del sistema.
- $K_p$ es la ganancia proporcional, que ajusta la respuesta en función del error actual.
- $K_i$ es la ganancia integral, que corrige errores acumulados a lo largo del tiempo.
- $K_d$ es la ganancia derivativa, que anticipa errores futuros basándose en la tasa de cambio del error.

El ajuste adecuado de estos parámetros permite obtener una respuesta estable y rápida, minimizando el sobreimpulso y el tiempo de estabilización del sistema.

### B. Control de Velocidad y Posición en Motores DC

El control de motores de corriente continua (DC) se basa en la regulación de su velocidad y posición mediante técnicas de control en lazo cerrado.

*1) Control de Velocidad:* Se busca mantener una velocidad deseada, compensando variaciones debido a la carga o perturbaciones externas. Para ello, se utiliza un controlador PID donde el error es la diferencia entre la velocidad deseada y la medida.

*2) Control de Posición:* En este caso, el objetivo es alcanzar una referencia de posición específica, utilizando un encoder para la retroalimentación. Se emplea un lazo de control PID donde el error es la diferencia entre la posición deseada y la real.

### C. Interfaz Arduino-MATLAB

MATLAB permite la comunicación con Arduino mediante la librería *MATLAB Support Package for Arduino*, facilitando la adquisición de datos y el envío de señales de control. En esta práctica, MATLAB calculará la señal de control PID y la enviará a Arduino, que la aplicará al motor DC mediante modulación por ancho de pulso (PWM).

Esta implementación permite probar diferentes configuraciones y realizar análisis en tiempo real para evaluar el rendimiento del control PID en la regulación de velocidad y posición del motor DC.

## III. Diseño de la planta

### A. Planta del sistema

Para la planta de nuestro sistema obtubimos lo siguiente a partir de la caractelizacion del motor utilizando el coamdo de matlab de sistemIdentification.

$$\frac{5.910}{s^2 + 211.8s + 4311}$$

Figura 1. Planta obtenida a partir de sistem identification

El diagrama de bloques con la planta obtenida entonces seria de la siguiente manera

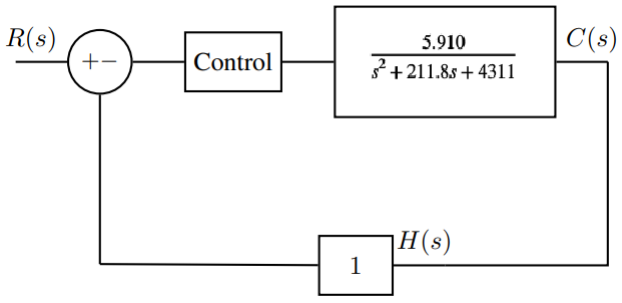


Figura 2. Sistema

De igual manera a este sitema de lazo abierto podemos sacarle la siguiente respuesta de manera que obtenemos parametros como T=50ms L=10ms y el K con un valor de 15,44:

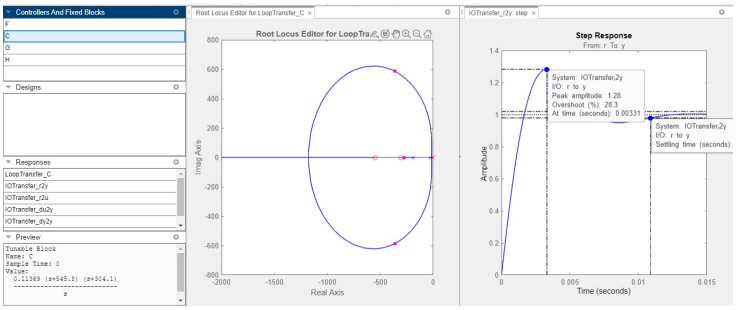


Figura 3. Diseño con controlSistemDesigner

Simulando el PID tenemos la siguiente Entrada siendo el escalon, una salida similar y un pico de error como podemos ver a continuación:

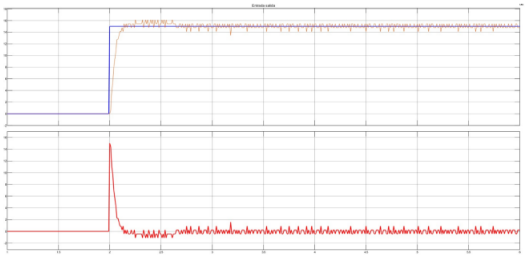

Figura 4. Entrada y salida del PID

Entonces, tenemos la siguiente acción de control para el sistema:

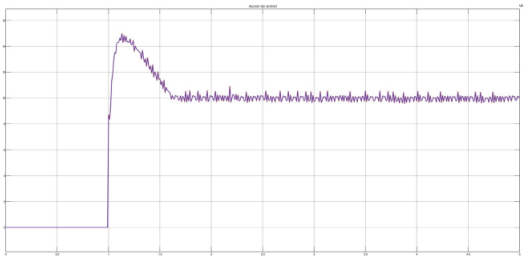

Figura 5. Accion de control

### B. Calculos del PID

$$1 + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + s + K_i}{s}$$

$$= \frac{K_p K_d s^2 + K_p s + K_p K_i}{s}$$

De la misma forma:

$$K_p \left( \frac{K_d s^2 + s + K_i}{s} \right) \cdot \frac{6910}{(s + 182.7)(s + 29)}$$

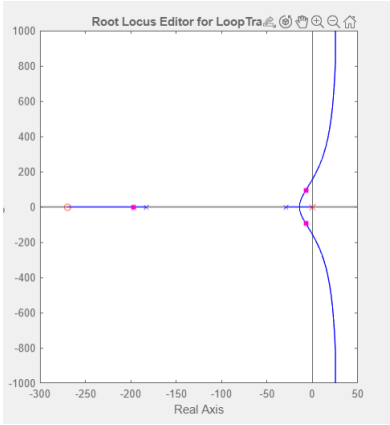


Figura 6. Polos

De la misma forma que con el caso del PI. Pero aqui se diseña para cancelar el polo en 182.7, y mantener el cero en 52 como en el caso del PI.

Tenemos entonces los siguientes calculos:

$$(s+182.7)(s+52) = s^2 + 234.7s + 9500.4$$

$$K_d s^2 + 234.7K_d s + 9500.4K_d$$

$$K_d s^2 + 5s + K_i$$

$$234.7K_d = 1 \implies K_d \approx 0.00426$$

$$9500.4 \cdot 0.00426 = K_i \implies K_i \approx 40.5$$

$$\text{Rlocus: } K_p \approx 0.76$$

El Rlocus del PID es el siguiente:

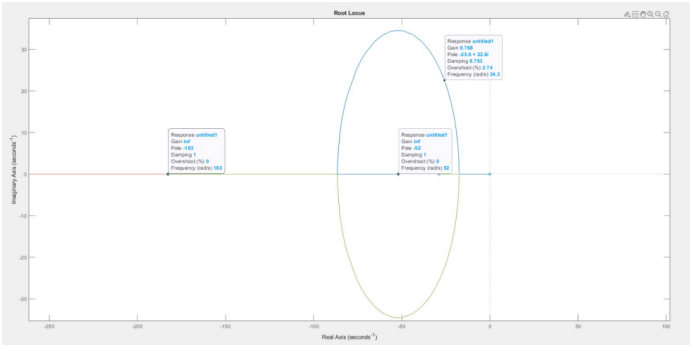


Figura 7. Rlocus del PID

Entonces obtenemos la entrada,la salida y el error de la siguiente manera:

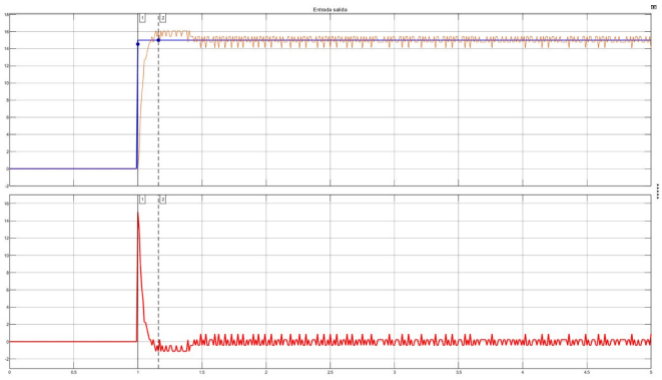

Figura 8. Rlocus del PID

En lazo abierto tenemos lo siguiente:

Lazo abierto:

$$G_p^* = \frac{G_p}{1 + G_p \cdot \frac{1}{s}} = \frac{5.486}{s^2 + 8.886s}$$

Lazo cerrado:

$$\frac{C}{R} = \frac{K_p(1+K_d s)\cdot 5.486}{s^2 + 8.886s + K_p(1+K_d s)\cdot 5.486}$$

$$= \frac{K_p(1+K_d s)\cdot 5.486}{s^2 + (8.886 + 5.486K_p K_d)s + 5.486K_p}$$

Por lo tanto, tenemos un sistema con 2 polos y 1 cero, se requiere:

$$\text{Ov} = 5\% \quad \text{y} \quad t_p = 0.1[s]$$

$$0.05 = \text{Ov}\% = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \implies \zeta = 0.69$$

$$0.1 = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} \implies \omega_n = 43.4$$

$$s \approx -30 \pm j31.4$$

Suponemos un cero en -30:

$$\frac{1}{K_d} = 30 \implies K_d = \frac{1}{30} \approx 0.033$$

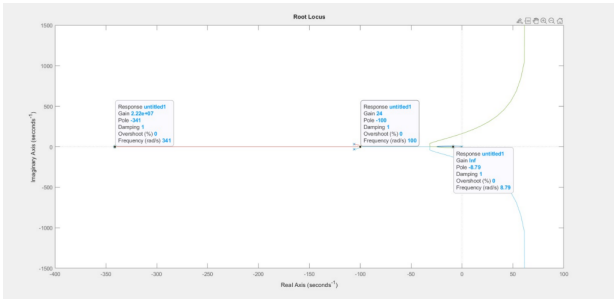


Figura 9. Rlocus del PID posición

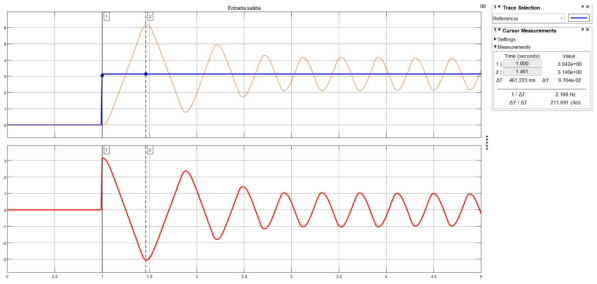

Figura 10. Salida entrada y error de la posición

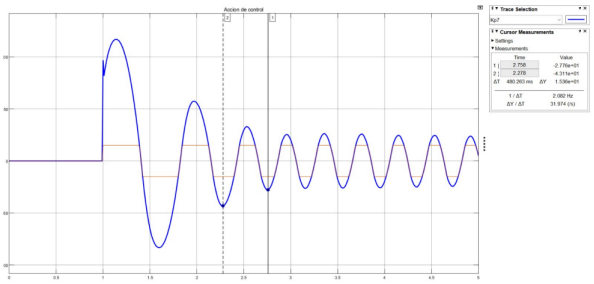

Figura 11. Acción de control

## IV. Resultados

El diseno del controlador PID de velocidad ˜ se obtuvieron con el método de Rlocus, el cual sé analizo la respuesta del motor en lazo abierto, para identificar los parametros que utiliza este metodo y poder aplicar el respectivo control.Para el diseno del control PID de posicion, se opto por el metodo de lugar geométrico de las raices. La funcion de transferencia del controlador (Gc) es la siguiente:

$$G_c = K_p\left(1 + \frac{K_i}{s} + K_d s\right)$$

$$G_p = \frac{5.910}{s^2 + 211.8s + 4311}$$

Para el lugar geometrico se multiplica con la planta, pero se quita el Kp debido a que al realizar Rlocus en matlab lo que hace es colocarle un Kp e ir variandolo, por lo tanto este actua como el valor de Kp del controlador. Se propone un valor de Ki = 10, y Kd = 0.1. El objetivo es escoger un polo con un overshoot del 1 porciento aproximadamente, para verificar la estabilidad del sistema. El valor de Kp elegido fue de Kp = 16.4, debido a que al bloque de PID se le tienen que agregar las constantes de otra manera, entonces los valores quedan de la siguiente manera:

$$P = K_p = 16.4$$

$$I = K_p \cdot K_i = 164$$

$$D = K_p \cdot K_d = 1.64$$

Para el controlador PID se opto simplemente por eliminar ´ la parte Integral del PID para poder comparar ambos controladores, las diferencias que se lograron encontrar fue que al tener un Ki muy alto, empieza a oscilar un poco el motor, haciendo de vez en cuando pequenos movimientos ˜ hasta llegar a la posicion deseada. Como se observó en ´ el laboratorio anterior, ambos controladores presentan un overshoot.

El ajuste para el PID como control de posicion se hizo ´ ajustando los valores P,I,D. Se dividio entre 100 y 5. El ´ resultado obtenido fue:

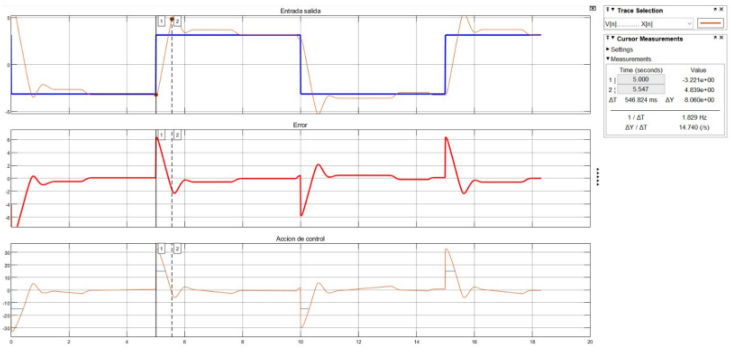

Figura 12. PID entrada salida y error de control

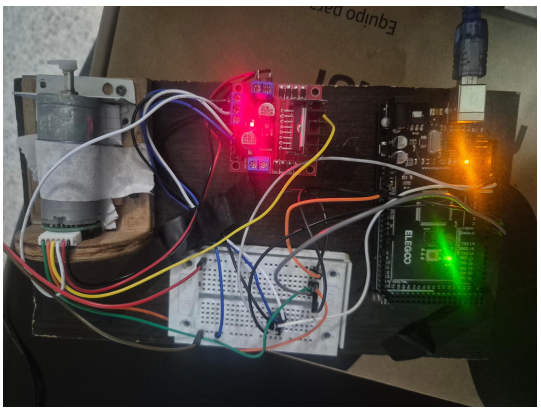

Figura 13. Montaje

## V. Conclusiones

- Los controladores que implican una parte IntegraloDerivativa afectan drasticamente a la respuesta que se est ´ a´deseando, si no se tiene cuidado al asignar estos valores,el sistema se volvera muy oscilatorio e inestable en el sentido de que deja de estar controlada la senal
- Respecto al control de velocidad, los resultados obtenidos fueron satisfactorios de la misma manera, cumpliendo los requisitos solicitados
- El uso del metodo de Ziegler–Nichols para la sintonizacion de controladores puede genera oscilaciones si los parametros no se ajustan correctamente, especialmente en sistemas con alta sensibilidad a cambios en Ki y Kd. Esto resalta la importancia de complementar estmetodo con ajustes e

manuales o tecnicas de optimización para garantizar un desempeno estable y predecible del sistema.

- La implementacion del m ´ etodo del lugar geom ´ etrico de las raíces para el diseno de controladores de posicióncpermite una seleccion más precisa de los polos deseados én comparacion con la sintonización por Ziegler-Nichols. nSin embargo, el ajuste de Ki y Kd en el controlador PID es un poco mas complejo, ya que valores inadecuados pueden inducir oscilaciones no deseadas como sencomento en en la sección de dificultades.

## VI. Referencias


1) Ingeniería De Control Moderna - 3ra Edicion - Katsuhiko Ogata
2) Diana Marcela Ovalle, Luis Francisco Combita - Theaching Basic Control Concepts With A Home-Made Thermal System.
3) System dynamics - William J. Palm III -2da Edicion
4) Control of Electric Machines - D. W. Novotny y T. A. Lipo.