

Control de Velocidad a través del Seguimiento de Perfiles.

García O. H. ¹, Tovar L. ², Quintanar S. ³,
 Ing. Mecatrónica, Instituto Tecnológico Superior de Huichapan,
 Saucillo, Hidalgo, México, a15021179@iteshu.edu.mx

Resumen— En el siguiente reporte se muestra el control de un motor DC usando un controlador PID en el cual se sintonizan sus constantes por asignación de polos. Esta práctica se realiza por medio de una tarjeta de adquisición de datos, la cual usa un microcontrolador, el cual nos permite leer señales analógicas que provienen sensores que se usaron para obtener los parámetros del motor.

Posteriormente se realiza el control de velocidad a través del seguimiento de perfiles, donde se realiza en Matlab una función para obtener los parámetros necesarios para generar un perfil en un tiempo rápido para obtener una buena respuesta con las constantes calculadas y obtener el control de una forma más precisa para el sistema con el mínimo error.

Abstract— The following report shows the control of a DC motor using a PID controller in which its constants are tuned by pole assignment. This practice is carried out by means of a data acquisition card, which uses a microcontroller, which allows us to read analog signals that come from sensors that were used to obtain the motor parameters.

Later the speed control is done through the tracking of profiles, where a function is performed in Matlab to obtain the necessary parameters to generate a profile in a quick time to obtain a good response with the calculated constants and obtain control in a way more accurate for the system with the minimum error.

Palabras clave — PID, asignación de polos, encoder, amplificador, RPM.

I. INTRODUCCIÓN

Es interesante señalar que más de la mitad de los controladores industriales que se usan hoy en día utilizan esquemas de control PID.

Como casi todos los controladores PID se ajustan en el sitio, se han propuesto muchos tipos diferentes de reglas de sintonización, que permiten llevar a cabo una sintonización delicada y fina de los controladores PID en el sitio. Asimismo, se han desarrollado métodos automáticos de sintonización y algunos de los controladores PID poseen capacidad de sintonización automática en línea. Actualmente se usan en la industria formas modificadas del control PID, tales como el control I-PD y el control PID con dos grados de libertad. Es posible obtener muchos métodos prácticos para una conmutación sin choque (desde la operación manual hasta la operación automática) y una programación del aumento.

A. Control proporcional

da una salida del controlador que es proporcional al error, es decir: $u(t) = K_P \cdot e(t)$.

Un controlador proporcional puede controlar cualquier planta estable, pero posee desempeño limitado y error en régimen permanente (off-set).

B. Control integral

da una salida del controlador que es proporcional al error acumulado, lo que implica que es un modo de controlar lento. La señal de control $u(t)$ tiene un valor diferente de cero cuando la señal de error $e(t)$ es cero. Por lo que se concluye que, dada una referencia constante, o perturbaciones, el error en régimen permanente es cero.

C. Control derivativo

Esta acción de control es proporcional a la derivada de la señal de error $e(t)$. La derivada del error es otra forma de llamar a la "velocidad" del error. A continuación, se verá porqué es tan importante calcular esta velocidad. Cuando el sistema se mueve a una velocidad alta hacia el punto de referencia, el sistema se pasará de largo

debido a su inercia. Esto produce un sobre pulso y oscilaciones en torno a la referencia. Para evitar este problema, el controlador debe reconocer la velocidad a la que el sistema se acerca a la referencia para poder frenarle con antelación a medida que se acerque a la referencia deseada y evitar que la sobre pase.

II. MATERIALES Y MÉTODOS

Materiales

- 2 amplificadores operacionales LM324
- Motor con encoder pololu
- Sensor de corriente ASC712
- Fuente de alimentación
- Tarjeta de adquisición USB
- Pickit 3
- Multímetro
- Resistencias (según los cálculos)
- Driver L298N
- Tacómetro

Software

- MPLAB
- Visual Studio
- PIC C Compiler
- Matlab

Fórmulas de desarrollo:

$$Va(s) = Ia(s)Ra + LasIa(s) \tag{1}$$

$$Va(s) + Ia(s)(-Ra - Las) = 0 \tag{2}$$

$$\frac{Ia(s)}{Va(s)} = \frac{1}{Las+Ra} \tag{3}$$

$$Tm(s) = KaIa(s) \tag{4}$$

$$T(t) = \frac{Idw(t)}{dt} + Bw(t) + Ti(t) \tag{5}$$

$$T(s) = w(s)[Js + B] \tag{6}$$

$$\frac{T(s)}{W(s)} = Js + B \tag{7}$$

$$\frac{W(s)}{T(s)} = \frac{1}{Js+B} \tag{8}$$

Despejando 4 en 3:

$$Ia = \frac{Tm(s)}{Ka} \tag{9}$$

$$\frac{Tm(s)}{Va(s)} = \frac{Ka}{Las+Ra} \tag{10}$$

De (8) despejamos Tm(s).

$$Tm(s) = w(s)(Js + B) \tag{11}$$

$$Ia(s)Ka = w(s)(Js + B) \tag{12}$$

$$Ia(s) = \frac{w(s)(Js+B)}{Ka} \tag{13}$$

Sustituir en (1) y multiplicamos por Ra

$$Va(s)Ka = w(s)[LaJs^2 + s(JRa + LaB) + Ka^2 + BRa] \tag{14}$$

Función de Transferencia de Velocidad

$$\frac{W(s)}{Va(s)} = \frac{ka}{LaJs^2+s(JRa+LaB)+Ka^2+BRa} \tag{15}$$

Función de Transferencia de Posición

$$\frac{W(s)}{Va(s)} = \frac{ka}{LaJs^2+s(JRa+LaB)+Ka^2+BRa} \left[\frac{1}{s} \right] \tag{16}$$

Función de Transferencia de Corriente

$$\frac{Ia(s)}{Va(s)} = \frac{Js+B}{LaJs^2+s(JRa+LaB)+Ka^2+BRa} \tag{17}$$

Para obtener los valores de las constantes de la ecuación característica es necesario despejar las siguientes formulas:

$$\frac{Ia(s)}{Va(s)} = \frac{a1s+a2}{a3s^2+a1s+a5} \tag{18}$$

$$\frac{1}{Lae} = a1 \tag{19}$$

$$\frac{Be}{Laje} = a2 \tag{20}$$

$$a4 = \frac{Rae}{Lae} + \frac{Be}{Je} \tag{21}$$

$$a5 = \frac{ke^2+BeRae}{LaeJe} \tag{22}$$

Despejando cada una de las ecuaciones anteriores se obtiene:

$$Lae = \frac{1}{a1} \tag{23}$$

$$Rae = \frac{a4 - \frac{a2}{a1}}{a1} \tag{24}$$

$$Ke = \frac{Vol}{Vel} \tag{25}$$

$$Je = \frac{a1Ke^2}{a5-a2Re} \tag{26}$$

$$Be = \frac{a2Je}{a1} \tag{27}$$

Contando con estos valores se lleva a cabo el desarrollo de las fórmulas del control PID paralelo.

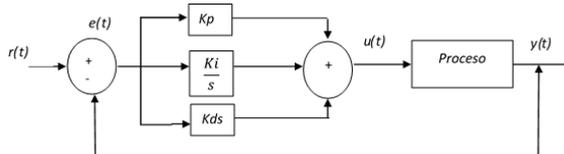


Fig. 1 PID paralelo.

Para este control vamos a suponer que tenemos el modelo matemático (función de transferencia) que representa nuestro proceso real y que esta FT es de segundo orden:

$$Gd(s) = \frac{K\omega_n^2}{s^2+2\xi\omega_n s+\omega_n^2} \tag{28}$$

donde K es la ganancia del sistema, ω_n es la frecuencia natural del sistema, ξ es el factor de amortiguamiento.

$$\omega_n = \frac{3}{\xi+s} \tag{29}$$

Por lo que M_p viene dada por la siguiente ecuación.

$$M_p = 100e^{\left(\frac{-\pi\xi}{\sqrt{1-\xi^2}}\right)} \tag{30}$$

Vamos a utilizar la ecuación del controlador PID convencional (Nota: pueden usar cualquier configuración del controlador, cada configuración dará características diferentes en la dinámica del sistema).

$$G(s) = \frac{K\omega_n^2}{s^2+2\xi\omega_n s+\omega_n^2} = \frac{k}{s^2+as+b} = \frac{A}{B} \tag{31}$$

$$C(s) = \frac{k_p s+k_i+k_d s^2}{s} = \frac{k_d s^2+k_p s+k_i}{s} = \frac{D}{E} \tag{32}$$

donde sabemos que la función de transferencia de Lazo cerrado viene dada por:

$$G(s) = \frac{AD}{BE+AD} \tag{33}$$

Ahora solo nos resta reemplazar los valores de los polinomios:

$$\frac{k_d s^2+k_p s+k_i}{s} \tag{34}$$

Ahora solo nos resta reemplazar los valores de los polinomios:

$$G(s) = \frac{a1d1s^2+a2d1s+a3d1}{e1s^3+e2s^2+e3s+a1d1s^2+a2d1s+a3d1} \tag{35}$$

$$G(s) = \frac{a1d1s^2+a2d1s+a3d1}{e1s^3+s^2(e2+a1d1)+s(e3+a2d1)+a3d1} \tag{36}$$

igualando coeficientes:

$$\alpha_0 = e1 \tag{37}$$

$$\alpha_1 = e2 + a1d1 \tag{38}$$

$$\alpha_2 = e3 + a2d1 \tag{39}$$

$$\alpha_3 = a1d1 \tag{40}$$

Reemplazando valores del controlador si:

$$a1 = kd \tag{41}$$

$$a2 = kp \tag{42}$$

$$a3 = ki \tag{43}$$

Entonces:

$$kd = \frac{\alpha_1-e2}{d1} \tag{44}$$

$$kp = \frac{\alpha_2-e3}{d1} \tag{45}$$

$$ki = \frac{\alpha_3}{d1} \tag{46}$$

Para realizar la parametrización de corriente y voltaje es necesario tomar en cuenta los datos del sensor de corriente a utilizar ya que consta de la siguiente formula:

$$I = \frac{V-2.5}{sensibilidad} \tag{47}$$

Donde se toma en cuenta el valor de la sensibilidad del tipo de sensor a utilizar, en este caso es de 100 mV/A. De acuerdo al desarrollo del programa en MPLAB junto con PICC

Compiler y Visual Studio de obtendrá la parametrización del sensor de corriente y voltaje.

Esto se cumplió con l ayuda del desarrollo del cálculo de un circuito con amplificadores operacionales para obtener un voltaje de salida de 5 Volts mandado a la salida ADC de la tarjeta de adquisición. El circuito consto de un amplificador inversor, amplificador con ganancia y un seguidor de tensión. Para obtener el voltaje deseado de salida se hacen los cálculos necesarios del valor de las resistencias:

$$V_o = -V_{in} \left(\frac{R_2}{R_1} \right) \tag{48}$$

$$5V = -12V(k) \tag{49}$$

$$k = \frac{5V}{12V} = 0.4166 = \frac{R_2}{R_1} = \frac{4.2K}{10K} = 0.42 \tag{50}$$

En la figura 2 se muestra el circuito con amplificadores operacionales:

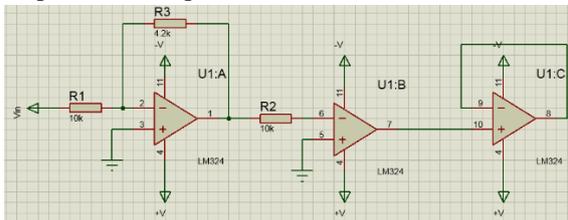


Fig. 2 Amplificador mediante inversor.

III.RESULTADOS Y DISCUSIÓN

Como primeros resultados se obtuvieron las gráficas de los datos adquiridos, como lo son voltaje, corriente y velocidad como se muestra 3, 4 y 5.

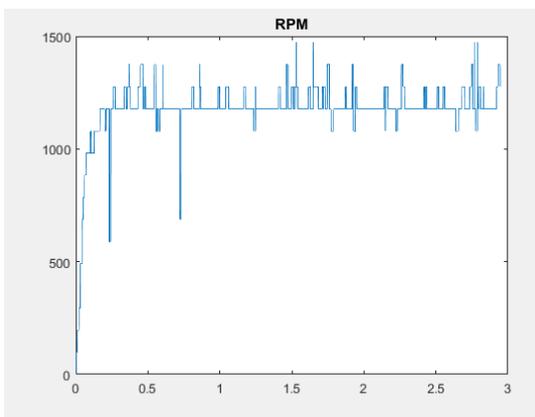


Fig. 3 Grafica de los datos obtenidos de la velocidad a 12 Volts.

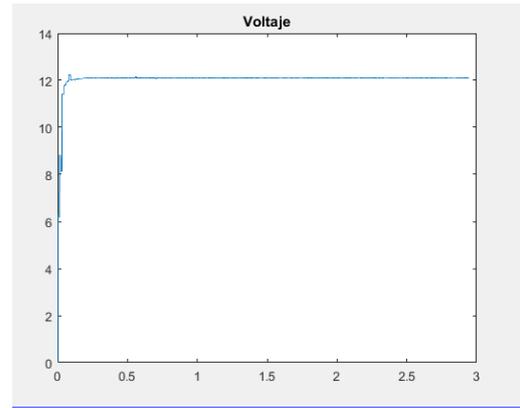


Fig. 4 Grafica de los datos obtenidos del comportamiento del voltaje de 12.

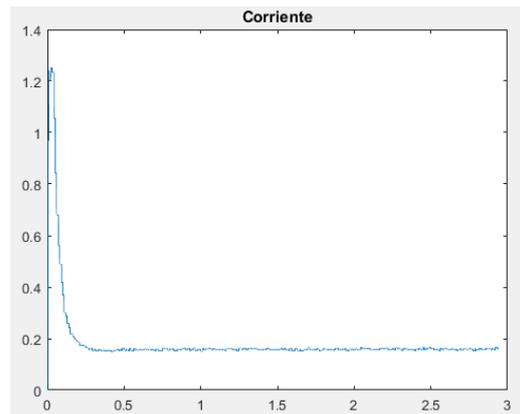


Fig. 5 Grafica de los datos obtenidos de la corriente a 12 Volts

Una vez que se obtuvieron los comportamientos de la velocidad, corriente y voltaje se prosiguió a conseguir la función de transferencia de la corriente como se muestra en la figura 6, donde la gráfica de color azul muestra la señal medida y la roja la estimada.

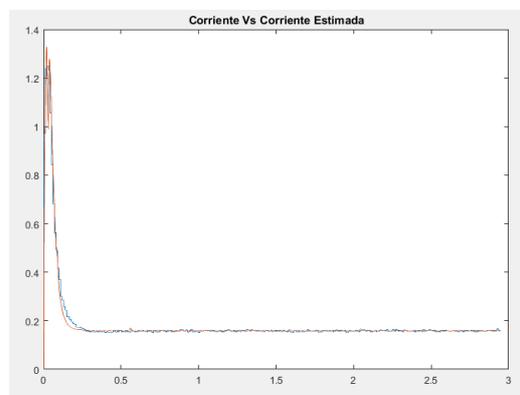


Fig. 6 Grafica de corriente estimada contra corriente medida.

Una vez que se tuvo la corriente estimada Vs corriente medida se prosiguió a conseguir de

igual forma y con el mismo código la velocidad estimada Vs la velocidad que se obtuvo por parte de la lectura del encoder se obtuvo lo que se muestra en la figura 7, donde la señal azul muestra la señal media y la roja la estimada.

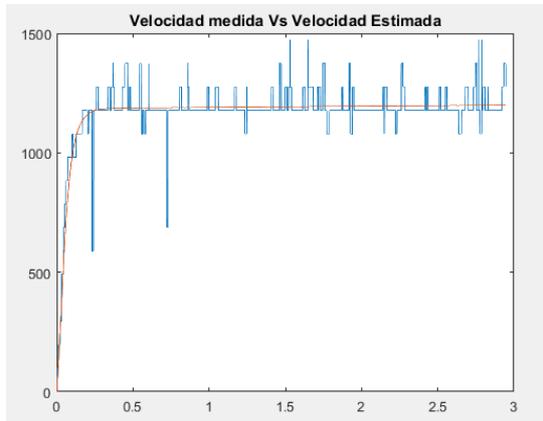


Fig. 8 Grafica de velocidad medido Vs velocidad calculado.

Una vez obtenido el comportamiento medido Vs el comportamiento calculado se prosiguió a obtener los parámetros del motor como lo son, la resistencia de armadura(Ra), voltaje(Va), inductancia (La), constante de torque e inercia del rotor(J).

Donde los resultados para el motor usado (info marco teórico) fueron los siguientes:

$$Lae = 0.0232$$

$$Re = 4.8378$$

$$Ke = 0.0104$$

$$Je = 7.8887e-07$$

$$Be = 1.5263e-06$$

Una vez obtenidos los parámetros del motor se prosiguió a obtener las constantes para nuestra ecuación característica deseada por medio de ξ y ω_n realizando lo que es nuestra ecuación de transferencia mediante el comando en Matlab tf donde nuestro numerador y denominador se conforma por los parámetros del motor previamente calculados obteniendo la ecuación 51.

$$FunTraG = \frac{5.687e05}{s^2 + 210.1s + 6331} \quad (51)$$

Donde la respuesta de nuestra función de transferencia a un step en Matlab se muestra en la figura 9.

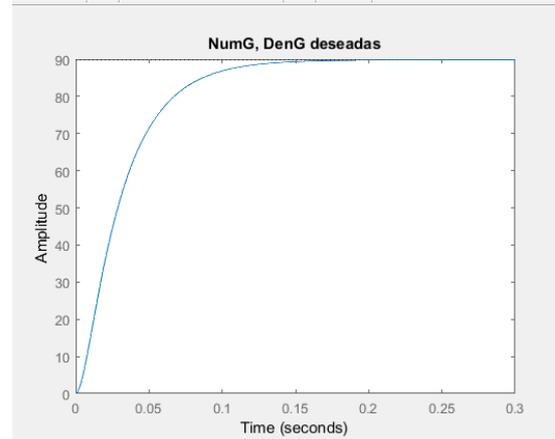


Fig. 9 Respuesta de la función de transferencia a un Step.

Una vez obtenido la respuesta de la función de transferencia se prosiguió a hallar los valores de las constantes con el fin de poder asignar nuestro polo a nuestra ecuación. Para esto fue necesario ingresar valores como lo son m_p y t_s , donde m_p es el sobrepaso de nuestra respuesta de nuestro sistema y t_s es el tiempo de establecimiento, donde nuestro sistema se logra estabilizar por completo.

Por medio de la función roots en Matlab se agregan las raíces del polinomio, estas se componen por nuestros valores de nuestra ecuación característica deseada y mediante la herramienta poly se le asigna el otro polo.

Una vez agregado el polo, ahora se podrán hacer igualaciones entre nuestra ecuación característica deseada y la desarrollada de nuestro sistema, donde por medio de despejes se obtienen las constantes de nuestro sistema quedando de la siguiente forma.

$$Kd = 2.1779e - 04$$

$$Kp = 0.0293$$

$$Ki = 0.0740$$

Una vez obtenidas las constantes se procede a hacer la prueba del funcionamiento y cómo es su desempeño del sistema mediante estas constantes en la interfaz realizada anteriormente para el control del motor.

Como primera prueba se realizó para una velocidad de 1000 RPM como referencia y se agregaron a la interfaz las constantes del controlador quedando de la siguiente forma.

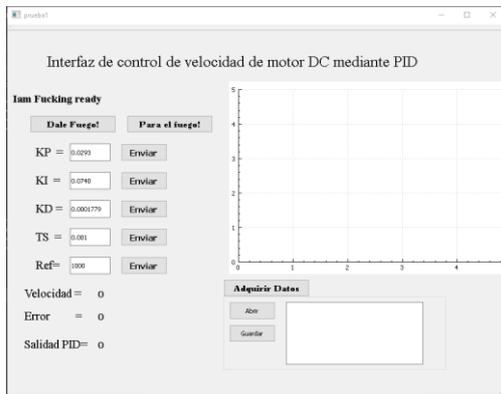


Fig. 10 Datos ingresado en la interfaz para la prueba.

Una vez agregadas las constantes y la referencia se corrió el programa obteniendo los resultados mostrados en la figura 11.

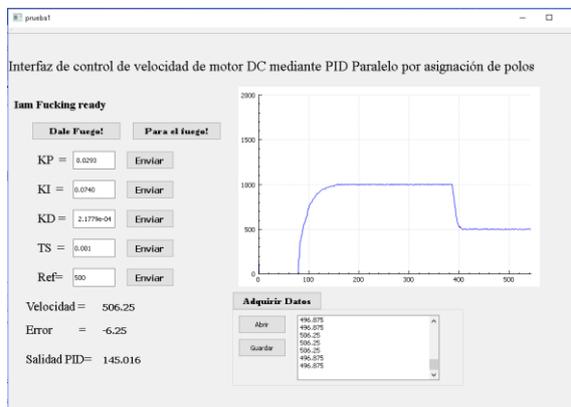


Fig. 11 Respuesta de nuestro sistema a la referencia con las constantes calculadas.

IV. CONCLUSIONES

De acuerdo a la practica realizada se obtuvieron los resultados esperados, obteniendo el control PID con asignación de polos mediante el uso de una tarjeta de adquisición para mostrar un perfil de velocidad deseado por lo que primero se hizo la parametrización de un motor con encoder mediante el uso de un sensor de corriente ASC712 para conocer los datos del vector de voltaje, corriente, tiempo y RPM. Posteriormente se aplicó la asignación de polos mediante el desarrollo de un programa en matlab para conocer los valores de las constantes kp, ki y kd y de acuerdo a eso controlar la velocidad de un motor mediante el control PID para obtener el seguimiento de perfiles.

REFERENCIAS

- [1] S. C. "> Control PID por Asignación de Polos - [abril, 2019]", *Control Automático Educación*, 2019. [Online]. Available: <https://controlautomaticoeducacion.com/control-realimentado/control-pid-por-asignacion-de-polos/>. [Accessed: 01- Apr- 2019].
- [2] "Polynomial with specified roots or characteristic polynomial - MATLAB poly- MathWorks España", Es.mathworks.com, 2019. [Online]. Available: https://es.mathworks.com/help/matlab/ref/poly.html?searchHighlight=poly&s_tid=doc_srchttitle. [Accessed: 03- Apr- 2019].
- [3] J. Tal. "STEP.BY-STEP DESIGN OF MOTION CONTROL SYSTEMS". Ed SunnyVale, CA. 1994.
- [4] Ingeniería de Control Moderna, Katsuhiko Ogata, 5ta. Ed., P568, Pearson Education, Madrid, España, 2010.