

Didactic system for identification and control of BLDC motors

F.J. Villalobos-Piña* R. Alvarez-Salas**
J.A. Reyes-Malanche*** C.H. Saucedo-Zarate*
M.S. Esparza-González* L.A. Gonzalez-Murillo**

* *TecNM/Instituto Tecnológico de Aguascalientes, Aguascalientes, Ags., 20256 México (e-mail: fvillalobos@mail.ita.mx, carloshum777@hotmail.com, msespa@hotmail.com).*

** *Facultad de Ingeniería, Universidad Autónoma de San Luis Potosí, San Luis Potosí, S.L.P., 78290 México (e-mail: ralvarez@uaslp.mx, luis.murillo@uaslp.mx).*

*** *Universidad Tecnológica de Aguascalientes, Aguascalientes, Ags., 20196 México (e-mail: josue.malanche@utags.edu.mx).*

Resumen: This work presents a didactic system for the identification and control of Brushless Direct Current (BLDC) motors. The system comprises a controller based on the PIC18F45K50 microcontroller, a Raspberry Pi 4 B+ computer with a graphical interface developed in Python using the Tkinter library. The system allows the identification of the motor's transfer function based on the speed response curve. Once the transfer function is identified, the system is capable of tuning a speed discrete-time PID controller to perform control tasks.

Keywords: BLDC motor, System identification, Control.

1. INTRODUCTION

El presente trabajo describe un sistema didáctico que permite efectuar la identificación básica de un motor tipo Brushless Direct Current (BLDC) con el propósito de implementar un controlador PID para el seguimiento de una referencia de velocidad y con la posibilidad de modificar las ganancias del algoritmo de control en línea.

Para la identificación y el control de sistemas existen muchas alternativas de plataformas digitales, por ejemplo, en [Horita et al. (2020)] se han estudiado métodos para el desarrollo de procesos robustos y escalables. Otros trabajos en la literatura sobre el control de motores BLDC, como es el caso de [Ghignone et al. (2019)], presenta el proceso de desarrollo y verificación de un sistema de potencia para motores BLDC para vehículos eléctricos. En [Kroics and Bumanis (2024)], se describe una propuesta de control de un motor BLDC mediante un control PID difuso adaptable. En [Gamazo-Real et al. (2010)], se presenta una revisión sobre diferentes técnicas para el control de posición y velocidad de motores BLDC sin sensores, indicando las ventajas y desventajas. Otra investigación detallada sobre el estado del arte de técnicas de control de motores BLDC se describe en [Mohanraj et al. (2022)], se muestra un amplio panorama de aplicaciones del motor

BLDC tales como fuente de movimiento en transmisiones industriales, aplicaciones de posicionamiento y control, además del estudio de la vida útil y su confiabilidad. En [Anshory et al. (2021)], se utiliza la lógica difusa para controlar la velocidad de un motor BLDC. Se han reportado también trabajos que describen el comportamiento del motor BLDC a través de señales de vibración utilizando técnicas de elemento finito con modelos 3-D [Cho et al. (2018)]. El uso de observadores deslizantes adaptables para el control de motor BLDC sin sensores se reporta en [Gan et al. (2018)].

En la sección 2, se describe la plataforma experimental. El software desarrollado se detalla en la sección 3. Las características de la interfaz gráfica se muestran en la sección 4. La sección 5 presenta un ejemplo de aplicación de la plataforma para identificar la curva de reacción de un motor BLDC y, finalmente, las conclusiones se discuten en la sección 6.

2. PLATAFORMA EXPERIMENTAL

En la figura 1 se presenta un diagrama de bloques del sistema implementado, cada uno de los elementos que lo integran se describen a continuación:

- Tarjeta Raspberry Pi 4.0 model B+, la cual se muestra en la figura 2.
- Pantalla HDMI marca Sunfounder.

* This work was partially supported by AME-FI-UASLP, UASLP-CA-246 Sistemas Eléctromecánicos y Energía Sustentable, and TecNM/Instituto Tecnológico de Aguascalientes.

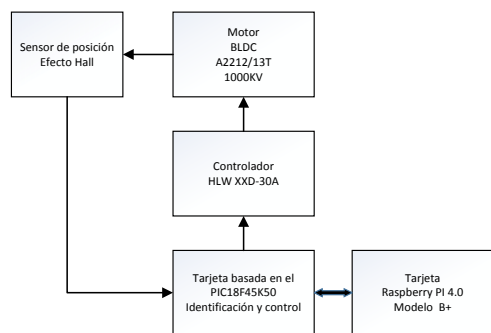


Figura 1. Diagrama de bloques de la plataforma.



Figura 2. Tarjeta Raspberry Pi 4.0 model B+.

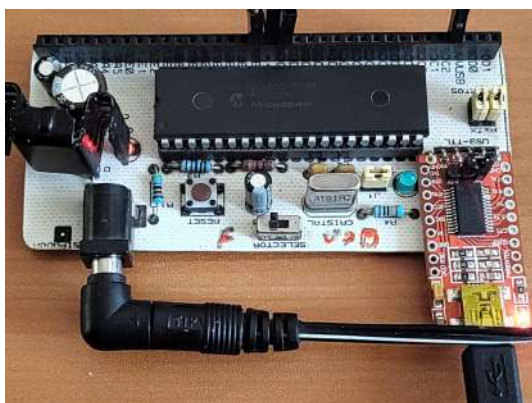


Figura 3. Tarjeta esclava.

- Tarjeta esclava basada en el microcontrolador Microchip PIC18F45K50 (figura 3).
- Motor trifásico BLDC marca QWinOut de 12V modelo A2212/13T 1000KV, al cual se le incorpora un sensor de posición (figura 4).
- Controlador marca Create idea modelo HLW XXD-30A, el cual es usado en sistemas de aeromodelismo y control de drones.

2.1 Tarjeta Raspberry Pi 4.0 model B+

El sistema de adquisición de datos consiste en una tarjeta Raspberry Pi modelo 4.0 model B+ con 8 Gb de memoria RAM, la cual se comunica con la tarjeta esclava mediante un puerto serial asíncrono (UART, por sus siglas en



Figura 4. Motor BLDC con sensor de posición.

inglés). Esta tarjeta opera con niveles lógicos de voltaje de 3,3 V, pero los terminales del UART son tolerantes a los niveles TTL. El sistema operativo utilizado es Raspbian, variante de la distribución Debian Linux.

2.2 Tarjeta esclava con PIC18F45K50

La Tarjeta esclava se basa en el microcontrolador Microchip PIC18F45K50, el cual opera a una frecuencia real de 12 MIPS. El microcontrolador efectúa la medición de velocidad gracias al módulo de captura con el que cuenta, el cual permite estimar la frecuencia de rotación del motor BLDC y, posteriormente, estimar su velocidad en revoluciones por minuto (RPM). El microcontrolador calcula la ley de control a una frecuencia constante establecida, es también capaz de establecer comunicaciones de alta velocidad a través de su puerto serial asíncrono. Las ganancias del controlador se almacenan en la memoria no volátil (Flash), las cuales se cargan al encender el sistema. Además, se pueden modificar en cualquier instante de operación.

2.3 Motor BLDC con sensor de posición

Para estimar la velocidad angular del motor BLDC, fue necesario instrumentar dicho motor, ya que no contaba con ningún tipo de sensor para tal fin. Se diseñó un anillo plástico que cuenta con cinco receptáculos para sujetar cada uno de ellos un imán de neodimio con seguridad, dicho diseño se llevo a cabo con tecnología de adición de filamento con una impresora 3D. En la figura 4, se muestra el motor BLDC, en el cual se montó el anillo plástico y a un lado de éste, se añadió un transistor de efecto Hall modelo U18444. El transistor solo requiere una resistencia de pull-up de 10kΩ para operar, la cual permite generar una señal de lógica de niveles TTL que se envía al módulo de captura del microcontrolador.

2.4 Controlador y señal PWM

Para operar el motor BLDC con el controlador HLW XXD-30A, éste requiere una señal que tenga un periodo de 20 msec, la duración en alto mínima es de 800 μseg y la dinámica para modificar la velocidad de 0% al 100% debe definirse mediante un cierto algoritmo. Esto se hace al aplicar la energía al controlador, éste genera una serie



Figura 5. Estructura de comunicación.

de tonos que son audibles durante los cuales el controlador espera el valor mínimo de PWM y, posteriormente, el valor máximo. Para el presente trabajo se definió una dinámica de $1,5\text{ msec}$, es decir, el algoritmo de PWM genera una señal en alto con una duración entre $800\ \mu\text{seg}$ y $2,3\text{ msec}$ sumando el tiempo mínimo en alto y el tiempo que define la velocidad desde su valor mínimo al máximo. De esta manera, el motor BLDC puede alcanzar una velocidad máxima de 23000 RPM .

3. SOFTWARE

En esta sección se describen las principales funciones del software desarrollado para la tarjeta Raspberry Pi y la tarjeta con el microcontrolador PIC18F45K50.

3.1 Software de la tarjeta Raspberry Pi

En la tarjeta Raspberry Pi se implementaron tres procesos:

- Captura de la curva de reacción de velocidad del motor BLDC.
- Ejecución y paro de la acción de control para regulación de velocidad del motor BLDC.
- Modificación en línea de las ganancias del controlador PID.

Respecto al proceso de comunicación para efectuar la captura de la curva de reacción de velocidad, un algoritmo solicita al microcontrolador PIC18F45K50 una trama de datos con información de la velocidad en RPM de tamaño de 1 KB , la restricción en el tamaño de la trama obedece a la limitación de la memoria RAM del microcontrolador. La velocidad de muestreo puede ser variable, para este caso se utilizó 1 Khz . Por lo que se captura un periodo de un segundo de información de la velocidad en tiempo real.

Para efectuar el proceso de ejecución y paro de la acción de control en la regulación de velocidad del motor BLDC, se envía un comando específico en el software de la tarjeta.

Es posible modificar y recuperar las ganancias del controlador PID, para lo cual se alojan en la memoria no volátil del microcontrolador tres palabras, cada una de ellas de 32 bits. Estas almacenan, en formato de punto flotante, las ganancias proporcional K_p , integral K_i y derivativa K_d .

Para lograr un dialogo confiable con la tarjeta Raspberry Pi, se implementa un esquema de comunicación con el microcontrolador en el cual se intercambian tramas con el formato mostrado en la figura (5).

3.2 Software de la tarjeta esclava

El microcontrolador PIC18F45K50 efectúa básicamente tres procesos, de igual forma que se llevan a cabo en la tarjeta Raspberry Pi:

- Adquisición de datos con la curva de reacción de velocidad del motor BLDC.
- Control PI(D) para la regulación de velocidad del motor BLDC.
- Modificación en línea de las ganancias del controlador PID.

El algoritmo del microcontrolador hace uso de un temporizador que permite el muestreo y almacenamiento de datos en memoria RAM, la información capturada es el periodo de una señal cuadrada producida mediante un sensor de efecto Hall y el grupo de imanes de neodimio al girar el rotor del motor. El hardware utilizado del microcontrolador para estimar la frecuencia de rotación es el módulo de captura. Dicho módulo produce una interrupción que se programa por flancos de subida y/o bajada o cierta cantidad de flancos de subida y/o bajada. Una vez que se genera dicha interrupción en conjunto con el temporizador se logra medir el tiempo entre flancos. Así, se calcula la frecuencia y, finalmente, se obtiene la velocidad de rotación en RPM del motor.

Para efectuar la regulación de velocidad el motor, se implementó un algoritmo de control PID en tiempo discreto. La ecuación en diferencias se muestra en (1), dicha ecuación en diferencias se ejecuta en la rutina de servicio a la interrupción por sobreflujo de un temporizador del microcontrolador. Cabe mencionar que el microcontrolador es de tecnología RISC, el cual cuenta con una unidad lógica aritmética de solo 8 bits. El cómputo de dicha ecuación en diferencias consume un tiempo máquina de aproximadamente $300\ \mu\text{seg}$, que es un tiempo adecuado, pues el periodo de muestreo de la ley de control es $T_s = 1\text{ msec}$.

$$Y_c[kT] = K_p e[kT] + K_i I_T[kT] + K_d D_a[kT] \quad (1)$$

en donde k es el índice de la secuencia, T es el periodo de muestreo, $e[kT]$ es la señal de error, $I_T[kT]$ es la ecuación en diferencias de un integrador trapezoidal, $D_a[kT]$ es la ecuación en diferencias de la aproximación de la derivada de $e[kT]$, K_p es la ganancia proporcional, K_i es la ganancia integral y K_d es la ganancia derivativa.

Finalmente, el microcontrolador es capaz de modificar las ganancias del controlador PID, las cuales se alojan en la memoria no volátil. Es posible leer y enviar a la tarjeta, así como recibir una nueva ganancia, la cual es decodificada en un grupo de cuatro bytes (32 bits) que representan una palabra de punto flotante.

4. INTERFAZ GRÁFICA

Se desarrolló un programa en Python con las librerías Tkinter y Matplotlib para la tarjeta Raspberry Pi con

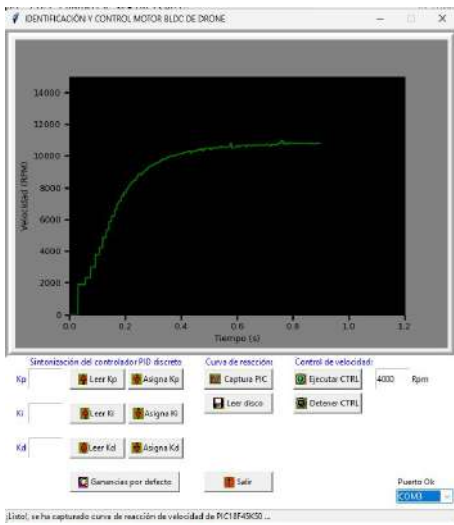


Figura 6. Captura de la curva de reacción de velocidad.

el fin de disponer de un entorno de desarrollo que facilite el análisis de los resultados. La interfaz gráfica permite efectuar la captura de la curva de reacción en tiempo real, el guardado de ésta en un medio de almacenamiento secundario, la lectura de datos que se tengan previamente guardados y su visualización, cambiar el puerto serial en caso de que la tarjeta tenga más de uno.

4.1 Captura de datos

Mediante un proceso de comunicación serial asíncrona *RS232*, se establece la comunicación con la tarjeta esclava a una velocidad de 256000 *Bauds*, 8 bits de datos, 1 bit de paro, sin paridad. Además, se muestrea la velocidad del motor BLDC con una frecuencia de muestreo de 1 *Khz* y se captura una cantidad de 900 muestras con un tamaño de palabra de 16 bits, como se ilustra en la figura 6.

4.2 Control y ajuste de ganancias del controlador

La interfaz gráfica de usuario permite efectuar la regulación de la velocidad, así como el proceso de lectura y ajuste de las ganancias del control de forma individual como se muestra en la figura 7

5. IDENTIFICACIÓN Y CONTROL

En esta sección se describe la identificación y el control del motor BLDC.

5.1 Identificación del sistema

Una vez que se cuenta con una base de datos de curvas de reacción de velocidad del motor BLDC, es posible efectuar un proceso de identificación del sistema. La curva de velocidad se aproximó a la de un sistema de primer orden dado por (2).

$$G(s) = \frac{K}{\tau s + 1} \quad (2)$$

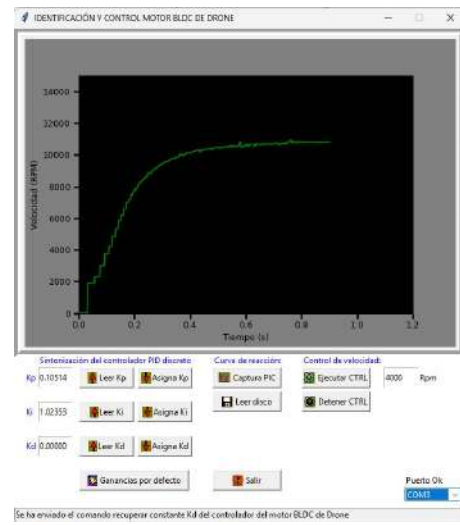


Figura 7. Control y ajuste de ganancias del controlador.

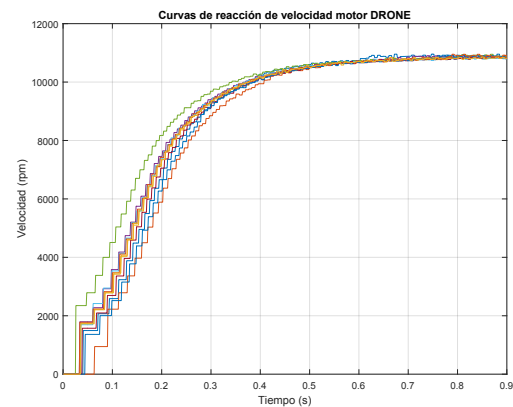


Figura 8. Curvas de reacción de velocidad experimentales.

En la figura 8, se presentan las curvas de 20 experimentos.

El cálculo de la ganancia K de la función de transferencia se obtuvo considerando el voltaje de excitación máximo aplicado al sistema, $V_s = 12$ *V*, y la velocidad máxima de rotación angular alcanzada por el motor V_{max} de acuerdo con $K = V_{max}/V_s$.

La ganancia fue $K = 822,7667$ y la constante de tiempo $\tau = 0,0960$, dando la siguiente función de transferencia identificada:

$$G(s) = \frac{822,8}{0,096s + 1} \quad (3)$$

En la figura 9, se presenta la comparativa entre la curva de reacción promedio experimental y la respuesta al escalón de amplitud V_s utilizando la función de transferencia identificada. Cabe mencionar que se trata de una aproximación básica que sirve como base para aplicar algoritmos más elaborados de identificación de sistemas.

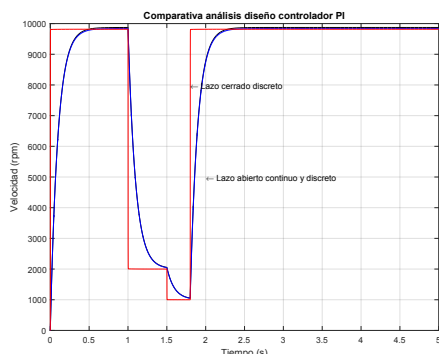


Figura 12. Resultados de simulación del sistema de control.



Figura 13. Resultado experimental del control velocidad sin perturbación de par de carga.



Figura 14. Resultado experimental del control velocidad con perturbación de par de carga.

escenario, pero con par de carga. Se puede observar la compensación, a través de la señal PWM, al presentarse la perturbación de par de carga.

6. CONCLUSION

Se presentó el desarrollo de una plataforma embebida de bajo costo para realizar tareas de identificación y de control para el seguimiento de velocidad de un motor BLDC. Se implementó el controlador PID en tiempo discreto en la plataforma esclava basada el microcontrolador PIC18F45K50, se obtuvieron resultados de simulación de la operación del controlador y experimentales del desempeño del sistema de control ante perturbaciones de par de carga del motor.

REFERENCIAS

- Anshory, I., Hadidjaja, D., and Sulistiyowati, I. (2021). Measurement, modeling, and optimization speed control of blcdc motor using fuzzy p so based algorithm. *Journal of Electrical Technology UMY (JET-UMY)*, 5(1), 1–9, <https://journal.umy.ac.id/index.php/jet/article/view/12113>, ISSN: 2550–1186.
- Cho, S., Hwang, J., and Kim, C.W. (2018). A study on vibration characteristics of brushless dc motor by electromagnetic-structural coupled analysis using entire finite element model. *IEEE Transactions on Energy Conversion*, 33(4), 1–16, <https://doi.org/10.1109/TEC.2018.2833493>.
- Gamazo-Real, J.C., Vázquez-Sánchez, E., and Gómez-Gil, J. (2010). Position and speed control of brushless dc motors using sensorless techniques and application trends. *Sensors MDPI*, 10(1), 6901–6947, www.mdpi.com/journal/sensors, ISSN: 1424–8220.
- Gan, M.G., Zhang, M., Zheng, C.Y., and Chen, J. (2018). An adaptive sliding mode observer over wide speed range for sensorless control of a brushless dc motor. *Control Engineering Practice Elsevier*, 77(1), 52 – 56, <https://doi.org/10.1016/j.conengprac.2018.05.004>, ISSN: 0967–0661.
- Ghignone, R.A., Calá, F.J., Giampetruzzi, J.G., Dománico, S.M., and Juarez, C.G. (2019). Control de motores dc sin escobillas para vehículos eléctricos usando un uc doble núcleo. *elektron*, 3(2), 84 – 90, ISSN: 2525–0159.
- Horita, A.Y., Loubach, D.S., and Bonna, R. (2020). Analysis and identification of possible automation approaches for embedded systems design flows. *MDPI Information*, 11(2), 1 – 29, doi:10.3390/info11020120.
- Kroics, K. and Bumanis, A. (2024). Bldc motor speed control with digital adaptive pid-fuzzy controller and reduced harmonic content. *Energies MDPI*, 17(6), 1–15, <https://doi.org/10.3390/en17061311>, ISSN: 1996–1073.
- Mohanraj, D., Arul david, R., Verma, R., Sathiyasekar, K., Barnawi, A.B., Chokkalingam, B., and image of Lucian Mihet-Popa Lucian Mihet-Popa, A. (2022). A review of bldc motor: State of art, advanced control techniques, and applications. *Sensors MDPI*, 10(1), 54833 – 54869, 10.1109/ACCESS.2022.3175011, ISSN: 2169–3536.