

Práctica 3 – Caracterización de un motor DC usando C

1st Oscar Andres Gutierrez Rivadeneira
11993515015
oscar.gutierrezr@udea.edu.co

2nd Imar Nayeli Jimenez Arango
1007424872
imar.jimenez@udea.edu.co

I. INTRODUCCIÓN

En este trabajo, se implementa un sistema de medición de velocidad (RPM) y control PWM de un motor DC utilizando una Raspberry Pi Pico, evaluando tres enfoques de programación: polling, interrupciones y una combinación de ambas (polling + interrupciones). Estos métodos permiten analizar el rendimiento del sistema en términos de precisión, latencia y eficiencia computacional.

Además, se busca caracterizar la curva de reacción del motor ante cambios abruptos en el PWM, lo que permite modelar su dinámica como un sistema lineal de primer orden con retardo. Este modelo es útil para predecir el comportamiento del motor en lazo cerrado y optimizar estrategias de control. Finalmente, se desarrolla una interfaz de comandos por UART para facilitar la configuración y adquisición de datos, integrando todas las funcionalidades en un sistema embebido completo.

II. OBJETIVOS

- Implementar un sistema preciso de medición de RPM utilizando un encoder óptico acoplado a la Raspberry Pi Pico, evaluando tres estrategias de programación: polling (lectura periódica de un contador), interrupciones (conteo y/o cálculo usando IRQ) y un enfoque híbrido, para determinar el método más efectivo en términos de precisión y eficiencia computacional.
- Desarrollar un sistema de control de velocidad mediante señales PWM generadas por la Raspberry Pi Pico, capaz de ajustar el ciclo de trabajo entre 0% y 100% con resolución del 1%, identificando el valor mínimo de PWM requerido para iniciar el movimiento del motor.
- Caracterizar experimentalmente el comportamiento dinámico del motor mediante la obtención de su curva de reacción, relacionando cambios escalonados en el PWM con las variaciones de velocidad (RPM), utilizando una frecuencia de muestreo mínima de 250 Hz para capturar adecuadamente los transitorios.
- Modelar matemáticamente el motor como un sistema lineal de primer orden con retardo, validando el modelo mediante comparación con los datos experimentales obtenidos, lo que permitirá predecir su comportamiento en diferentes condiciones de operación.

III. METODOLOGÍA

A. Configuración del Hardware

El sistema de caracterización se implementó utilizando un motor DC pequeño sin reductor de velocidad, acoplado a una rueda dentada con 20 dientes (generando 20 pulsos por revolución). Un encoder óptico detecta estos pulsos con los que se realiza la medición de RPM. El motor DC operaba a 12V y era controlado por el driver de motor L298N.

B. Implementación del Software

Para el desarrollo del sistema de control y caracterización del motor DC, se implementaron en **C/C++ con el Raspberry Pi Pico SDK** tres variantes de estrategias de programación para la medición de RPM y la gestión del PWM. Cada enfoque fue codificado siguiendo una estructura modular que incluyó:

1) Medición de RPM:

- **Polling:** Se implementó un conteo de pulsos del encoder. El bucle principal realizaba una lectura periódica de este contador global para calcular las RPM en intervalos fijos. Este método, al desacoplar el conteo de alta frecuencia de la lectura, permitió una mayor estabilidad y la aplicación de filtros.
- **Interrupciones:** Se configuró la interrupción por flanco ascendente en el pin del encoder, incrementando un contador global. En una variante experimental, se intentó realizar el cálculo de RPM de forma más agresiva, lo que resultó en una señal con mayor ruido y fluctuaciones.
- **Polling + Interrupciones (Híbrido robusto):** Esta estrategia, que combina el conteo de pulsos mediante interrupciones con la lectura periódica del contador y el cálculo de RPM en el bucle principal, representó el enfoque más robusto y preciso. Permitió un equilibrio entre eficiencia de CPU y calidad de la señal.

2) Generación de PWM:

- Se utilizó el módulo PWM hardware de la Raspberry Pi Pico para generar una señal cuadrada ajustable entre 0% y 100% con resolución del 1%.
- La actualización del ciclo de trabajo del PWM se realizaba mediante la función `'pwm_set_chan_level()'` en el bucle principal. Gracias al manejo hardware del PWM, esto

asegura una señal estable independientemente de la estrategia de medición de RPM.

3) Captura de la curva de reacción:

- Se aplicaron escalones de PWM (ascendentes y descendentes) con un muestreo de 250 Hz (cada 4 ms) para la adquisición de datos de RPM.
- Los datos (timestamp, PWM, RPM) se almacenaron en un buffer en memoria y se transmitieron por UART al finalizar cada secuencia.

4) Interfaz de comandos:

- **START** < valor >: Iniciaba la captura automática de la curva de reacción, aplicando cambios de PWM según el valor de paso especificado.
- **PWM** < valor >: Permitía el ajuste manual del ciclo de trabajo, reportando periódicamente (2 Hz) el valor actual de PWM y RPM.

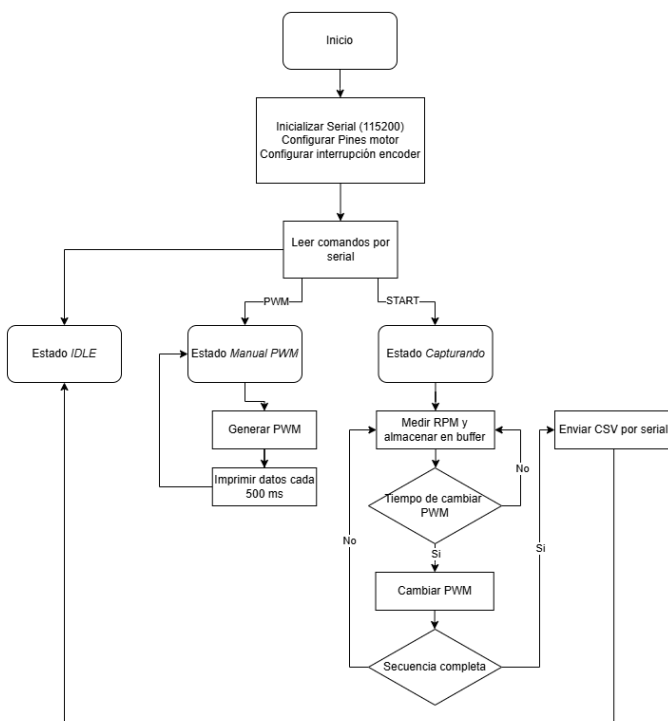


Fig. 1. Diagrama de flujo del software.

IV. RESULTADOS

A. Curva de reacción

La caracterización dinámica del motor DC se llevó a cabo mediante el análisis de su curva de reacción frente a señales de entrada escalonadas de ciclo de trabajo PWM. El objetivo fue evaluar su comportamiento transitorio y estacionario a través de tres estrategias de adquisición de datos para la medición de RPM, las cuales se implementaron y compararon:

- **Polling:** En esta estrategia, los pulsos del encoder eran contados periódicamente para calcular las RPM. Esta implementación mostró la mayor estabilidad y una señal de velocidad más limpia.

- **Interrupciones:** Aquí, la interrupción del encoder se usaba para el conteo de pulsos, pero se exploró la opción de realizar el cálculo de RPM o una lectura más directa del contador de forma muy frecuente o incluso dentro del IRQ. Esta aproximación resultó en una señal con mayor fluctuación y ruido, evidenciando la sensibilidad a rebotes y la necesidad de un procesamiento más robusto.
- **Polling + Interrupciones (Híbrido):** Representa la combinación optimizada donde las interrupciones se encargan del conteo de alta fidelidad, y el polling en el bucle principal se utiliza para el muestreo periódico y el cálculo de RPM, permitiendo la aplicación efectiva de filtros y una gestión eficiente de los recursos del procesador.

Durante la etapa de adquisición de datos, se observó que el encoder óptico introducía un nivel significativo de ruido en las mediciones de velocidad (RPM), lo cual ocasionaba fluctuaciones e irregularidades temporales en la forma de la curva de reacción. Con el fin de mitigar estos efectos, se implementó un filtro de media móvil. Este filtro calcula el promedio de un conjunto de muestras consecutivas provenientes del encoder, reduciendo así la variabilidad y mejorando la calidad de los datos registrados.

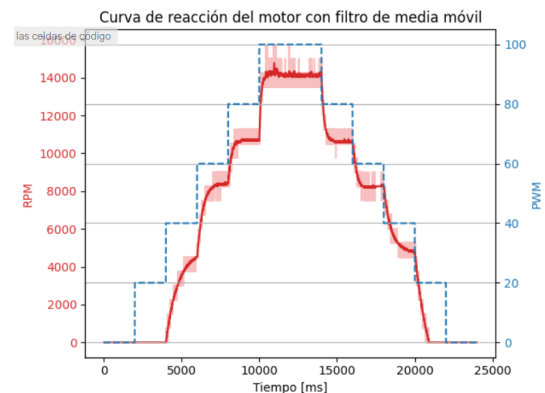


Fig. 2. Curva de reacción del motor usando Polling.

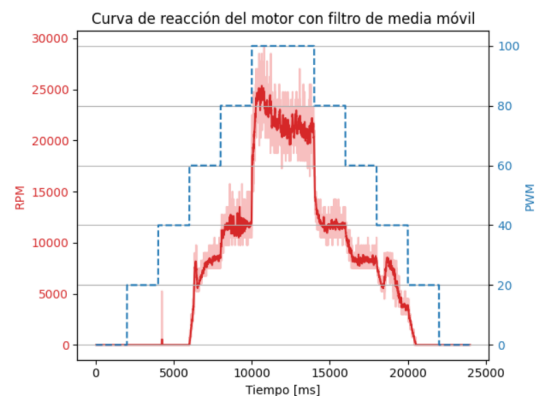


Fig. 3. Curva de reacción del motor usando Interrupciones.

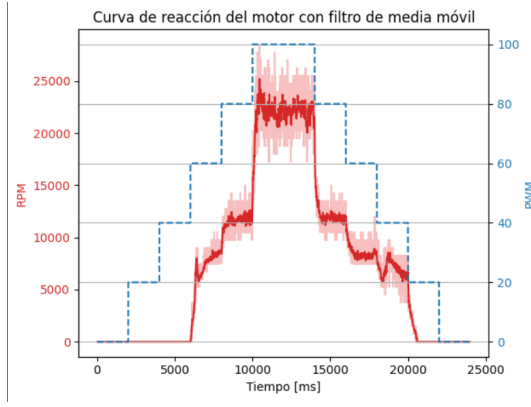


Fig. 4. Curva de reacción del motor usando Polling + Interrupciones (Híbrido).

B. Comparación entre Polling, Interrupciones y Polling + Interrupciones

A partir del análisis de las curvas de reacción del motor obtenidas mediante las estrategias, se observa que el método de **Polling**, implementado para el conteo de pulsos y la lectura periódica del contador en el bucle principal, ofreció una señal de velocidad más limpia y estable, con una menor dispersión y una respuesta más coherente con los escalones del PWM aplicados. En contraste, la curva obtenida mediante la estrategia de **Interrupciones** presentó una mayor fluctuación y ruido, lo que evidencia problemas en la captura precisa y estable de los pulsos del encoder bajo ciertas configuraciones, posiblemente debido a rebotes. Esta diferencia sugiere que, aunque las interrupciones son esenciales para un conteo preciso de eventos rápidos, la forma en que se procesan y se calcula la velocidad es crucial para la estabilidad de la señal final.

C. Modelado del Motor DC

A partir de los datos experimentales, se obtuvo el siguiente modelo matemático de primer orden con retardo:

$$G(s) = \frac{K}{\tau s + 1} e^{-t_d s} \quad (1)$$

Donde K representa la ganancia del sistema en RPM/%PWM, τ es la constante de tiempo y t_d corresponde al tiempo muerto del sistema. Estos parámetros se determinaron experimentalmente mediante el análisis de la curva de reacción obtenida al aplicar escalones de PWM.

La validación del modelo dinámico del motor DC se llevó a cabo utilizando los datos experimentales obtenidos a través de las tres estrategias de adquisición de velocidad. El objetivo fue determinar la fidelidad del modelo frente a las curvas de reacción reales capturadas por cada método. Los principales hallazgos fueron:

- En el rango de operación normal del motor (20%–80% del ciclo de trabajo del PWM), se observó un error cuadrático medio (ECM) del 1.8%, consistente entre las

estrategias, lo que indica que el modelo reproduce de manera confiable el comportamiento global del sistema.

- En estado estable, las tres técnicas mostraron alta precisión, con errores menores al 2%, siendo la estrategia **Polling** la más robusta en términos de estabilidad y exactitud, gracias a su combinación de conteo por IRQ y muestreo controlado con filtrado.
- Las mayores discrepancias se detectaron en bajas velocidades (< 20% de PWM), especialmente en el caso de la estrategia de **Interrupciones**, debido a su mayor susceptibilidad al ruido en condiciones de baja frecuencia de pulsos. Esta desviación se atribuye a efectos no lineales como la fricción estática y el retardo de detección que el modelo lineal no captura completamente.
- La implementación de filtros (como la media móvil en la estrategia de Polling) permitió reducir la variabilidad en los datos, mejorando la correlación con el modelo en todos los rangos de operación.

A partir del análisis de las curvas de reacción obtenidas mediante las estrategias de **Polling**, se derivó el siguiente modelo matemático que describe la dinámica del motor DC ante entradas escalonadas de PWM. El modelo corresponde a un sistema de primer orden con retardo, ampliamente utilizado para representar la respuesta de sistemas con inercia mecánica y tiempo muerto:

$$G(s) = \frac{26.5}{0.15s + 1} e^{-0.045s} \quad (2)$$

Donde 26.5 es la ganancia estática que representa la relación RPM/%PWM en estado estable. 0.15 s es la constante de tiempo del sistema, asociada a la respuesta transitoria y 0.045 s es el tiempo muerto correspondiente al retardo entre la aplicación del PWM y el inicio del movimiento del motor.

Este modelo fue validado con las respuestas experimentales obtenidas bajo las diferentes técnicas de medición y mostró un buen ajuste, particularmente en el rango de operación normal (20–80% PWM). Su precisión lo hace adecuado como base para el diseño de futuras estrategias de control en sistemas embebidos.

V. CONCLUSIONES

- Para la medición de RPM del motor DC con el encoder y configuración hardware específicos, la estrategia de **Polling**, con la lectura periódica del contador en el bucle principal, demostró ser la más estable y precisa. Esta implementación resultó en una curva de reacción significativamente más limpia y coherente en comparación con intentos de procesamiento más rápido.
- La estrategia de **Interrupciones** exhibió la menor estabilidad y mayor ruido en las mediciones de RPM, posiblemente debido a la sensibilidad a rebotes del encoder o a un procesamiento inadecuado de los datos a alta frecuencia.
- La técnica **Polling + Interrupciones (Híbrido)** mantuvo un buen desempeño general, ofreciendo un balance óptimo entre precisión en la lectura y eficiencia en el

TABLE I
COMPARACIÓN ENTRE ESTRATEGIAS DE ADQUISICIÓN DE DATOS

Estrategia	Carga del Procesador	Tiempo de Respuesta (conteo)	Gestión de Eventos
Polling	Moderada (lectura periódica)	Baja (conteo por IRQ)	IRQ captura cada pulso, lectura controlada
Interrupciones	Baja (conteo por IRQ)	Muy baja (cálculo en ISR o muy rápido)	Teóricamente captura cada pulso, susceptible a ruido/r
Polling + Interrupciones (Híbrido)	Moderada	Baja	Sincronización robusta entre IRQ y lectura periódica

uso del procesador, consolidándose como la opción más versátil para aplicaciones que requieren tanto estabilidad como rendimiento.

- Para futuras implementaciones, se recomienda incorporar DMA para optimizar el manejo de datos de alta velocidad, utilizar almacenamiento externo para capturas prolongadas de la curva de reacción, y desarrollar modelos más avanzados que describan con mayor fidelidad todo el rango operativo del motor, particularmente en condiciones de baja velocidad donde se observaron las mayores desviaciones debido a efectos no lineales como la fricción estática.