



CONTROL FEED FORWARD, MEDICIÓN Y GRÁFICO DE R.P.M

Richard Gómez, Santiago Alarcón

*Escuela de Ciencias Exactas e Ingeniería, Universidad Sergio Arboleda
Bogotá, Colombia*

richard.gomez01@usa.edu.co, santiago.alarcon01@usa.edu.co

Abstract

Este proyecto presenta el diseño e implementación de un sistema de **control de velocidad para un motor DC con encoder de 300 RPM**, utilizando una tarjeta de desarrollo STM32 y una interfaz gráfica desarrollada en Qt. El objetivo principal es garantizar un control preciso de la velocidad lineal del motor como movimiento principal del objetivo final: El Micromouse, expresada en mm/s, mediante la aplicación de un **control FeedForward** basado en la caracterización experimental del sistema, es decir, la cantidad de pulsos y el comportamiento respecto a diferentes voltajes.

INTRODUCCIÓN

Inicialmente, se midieron los pulsos del encoder en tiempo real utilizando el modo encoder de los temporizadores del microcontrolador, y se realizó la conversión de pulsos a rad/s y posteriormente a mm/s. Principalmente la interfaz Qt permitió seleccionar el puerto serial, enviar consignas de velocidad, visualizar en tiempo real la respuesta del sistema mediante un gráfico de tiempo vs velocidad y registrar los datos en un archivo CSV para su análisis posterior.

Por otra parte, se implementó un **filtro FIR** para mejorar la calidad de la señal de velocidad y se usaron funciones matemáticas para convertir voltaje a PWM y velocidad deseada a voltaje, optimizando así la respuesta del sistema. La parte final de proyecto, consistió en la caracterización

la relación velocidad–voltaje, evaluando el tiempo de estabilización según la gráfica.

METODOLOGÍA

Para medir con precisión la velocidad real del motor, se utiliza un sensor encoder, el cual genera pulsos que indican cuánto ha girado la rueda. A partir de esos pulsos se calcula la velocidad y se compara con la velocidad deseada. Por otra parte, El motor fue controlado mediante señales PWM generadas por la tarjeta STM32. El encoder se conectó a los pines de un temporizador configurado en **modo encoder**, lo que permitió contar pulsos a medida que el motor giraba. La lectura de los pulsos del encoder se utilizó para calcular la **velocidad angular** y posteriormente la **velocidad lineal** de la rueda en mm/s. Esta conversión se realizó a partir del número de pulsos por vuelta y el diámetro de la rueda, relacionando el giro físico con el

desplazamiento lineal. Se implementó un **control FeedForward**, que consistió en aplicar un valor de PWM previamente calculado en función de la velocidad deseada. Para ello, se caracterizó la respuesta del motor midiendo la velocidad real para diferentes niveles de voltaje, obteniendo así una relación lineal que permitió estimar el PWM adecuado para alcanzar una velocidad específica.

Con el fin de mejorar la calidad de la señal de velocidad, se implementó un **filtro FIR**, reduciendo fluctuaciones y ruido en las mediciones. Luego, los datos fueron analizados para evaluar el comportamiento del motor, el tiempo de estabilización y la precisión del control.

Calculo

$$RPM = \frac{encoder_motor1 \cdot 60}{pulsos_vuelta \cdot (tiempo_base \cdot 100 \cdot 10^{-6})}$$

encoder_motor_1 = número de pulsos contados en el intervalo de muestreo.

PULSOS_VUELTA = cantidad de pulsos que genera el encoder por cada vuelta completa.

tiempo_base = intervalo de muestreo (en unidades base del sistema).

100e-6 = factor de corrección (por ejemplo, si el tiempo base está en pasos de 100 µs).

60 = factor de conversión de revoluciones por segundo a revoluciones por minuto (RPM).

RESULTADOS Y DISCUSIONES

Al principio, la conversión de pulsos a mm/s no consideraba correctamente los pulsos por vuelta, lo que producía valores exagerados, por lo tanto, Se recalculó la relación pulsos → rad/s → mm/s utilizando el número exacto de pulsos por vuelta y el diámetro real de la rueda, eliminando lecturas incorrectas.

Por otra parte, se registraron datos incoherentes en la gráfica, Al iniciar las primeras pruebas, la gráfica mostraba valores fuera de escala o picos irregulares que no coincidían con el comportamiento real del motor. Para reducir el ruido, se aplicó un filtro FIR simple que suavizó la señal de velocidad sin afectar el tiempo de respuesta.

CONCLUSIONES

El control FeedForward permite alcanzar velocidades deseadas con buena precisión, siempre y cuando exista una caracterización adecuada de la relación voltaje-velocidad. Al aplicar este enfoque, el sistema mostró tiempos de estabilización cortos y un comportamiento predecible para diferentes consignas.

La comunicación serial entre la STM32 y la interfaz en Qt es un componente crítico, ya que errores en el formato o la frecuencia de envío pueden generar datos incoherentes.

El uso de filtrado FIR y una correcta conversión de pulsos a unidades físicas mejoró significativamente la calidad de la medición, permitiendo eliminar ruido, estabilizar la gráfica y obtener información más precisa para el análisis del sistema.

Referencias

[1] R. Magalhães, “Control de velocidad y dirección del motor de CC con microcontrolador STM32F103 y controlador de motor L293D,” *COMPRACO Indústria e Construção*, Jun. 14, 2024. [En línea]. Disponible en: <https://compraco.com.br/es/blogs/tecnologia-e-desenvolvimento/control-de-velocidade-e-direcao-do-motor-dc-com-microcontrolador-stm32f103-e-driver-de-motor-l293d?srltid=AfmBOoq7tGrIQo0YKO4YZsy7YQ5pAvbf2K4TYqgGLvpxBNC5YWeSVLKN>

[2] T. E. y-desenvolvimento, “Setting Up – Qt Plotting Widget QCustomPlot,”

QCustomPlot, 2025. [En línea]. Disponible en:

<https://www.qcustomplot.com/index.php/tutorials/settingup>

[3] R. Lozano, “¿Cómo medir RPM encoder 400 pulsos?,” *Talos Electronics Tutoriales*, 22 Ene. 2023. [En línea]. Disponible en:

<https://www.taloselectronics.com/blogs/tutoriales/como-medir-rpm-encoder-400-pulsos>