

# Reto: Implementación de un control de lazo cerrado en ROS2

1<sup>st</sup> Daniel Nava Mondragon

*Ingeniería en Robótica y Sistemas Digitales*

*Instituto Tecnológico y de Estudios Superiores de Monterrey*

CDMX, México

A01661649@tec.mx

2<sup>nd</sup> Juan Carlos Jiménez Tapia

*Ingeniería en Robótica y Sistemas Digitales*

*Instituto Tecnológico y de Estudios Superiores de Monterrey*

CDMX, México

A01750115@tec.mx

**Abstract**—Este reporte tiene como objetivo explicar como fue la realización y resolución del reto de este periodo. El reto es implementar un sistema de control de lazo cerrado para controlar la velocidad angular de un motor con encoder de cuadratura de fase, todo esto en un ambiente de ROS2 controlado mediante un microcontrolador ESP32.

## I. INTRODUCCIÓN

## II. MARCO TEÓRICO

### A. Control de Lazo Cerrado

El control de lazo cerrado, también conocido como control de retroalimentación, es una metodología de control donde la salida de un sistema se mide y comparada con una referencia o valor deseado. Cualquier diferencia o error se utiliza para ajustar la entrada del sistema y así acercar la salida al valor deseado. Este enfoque es fundamental en sistemas dinámicos donde las perturbaciones internas y externas pueden afectar la estabilidad y el rendimiento del sistema. El control de lazo cerrado es ampliamente utilizado en la industria para el control de procesos, sistemas robóticos y aplicaciones de mecatrónica debido a su capacidad de reducir el error y mantener el comportamiento del sistema dentro de parámetros aceptables.

### B. Motor con Encoder

Un motor con encoder es una configuración en la que un motor eléctrico está acoplado a un encoder, que es un dispositivo de sensado que proporciona retroalimentación de posición o velocidad. Los encoders pueden ser de tipo incremental, proporcionando pulsos que indican movimiento relativo, o absolutos, dando una posición precisa del eje del motor. La combinación de motores con encoders es esencial para aplicaciones de control precisas donde se necesita saber la posición exacta del motor o controlar su velocidad con gran precisión.

### C. Motorreductor

Un motorreductor es un conjunto mecánico que consta de un motor eléctrico y un sistema de reducción de engranajes. Este sistema tiene como objetivo disminuir la velocidad de salida del motor, aumentando al mismo tiempo su torque. Los motorreductores son críticos en aplicaciones donde se requieren movimientos controlados y fuerzas considerables,

como en sistemas de automatización, robots industriales y vehículos eléctricos.

### D. ROS2

ROS2 (Robot Operating System 2) es la nueva generación del popular middleware ROS, diseñado para la robótica. Proporciona una estructura estandarizada para la comunicación entre diferentes partes de un sistema robótico, como sensores, actuadores y procesadores. ROS2 introduce mejoras significativas sobre su predecesor, incluyendo un modelo de comunicación más robusto y seguro, soporte de tiempo real, y una mejor estructura de red para sistemas distribuidos y de múltiples robots.

### E. ESP32

El ESP32 es un microcontrolador de bajo costo y bajo consumo energético con capacidades Wi-Fi y Bluetooth integradas, fabricado por Espressif Systems. Es ampliamente utilizado en proyectos de IoT (Internet de las cosas) debido a su módulo de radio dual que permite la comunicación y el control a través de redes inalámbricas. Además, el ESP32 es compatible con una amplia variedad de entornos de desarrollo, incluyendo Arduino IDE, Espressif IDF y otros, lo que lo hace versátil para el desarrollo de prototipos rápidos y productos comerciales.



Fig. 1. ESP32

### III. METODOLOGÍA

#### A. Implementación práctica

Para poder controlar el motor con un controlador de lazo cerrado requeríamos de un microcontrolador, en este caso se utilizó la ESP32, y un encoder para obtener su valor actual, se implementó un encoder de cuadratura de fase de efecto Hall, específicamente el scx-555.

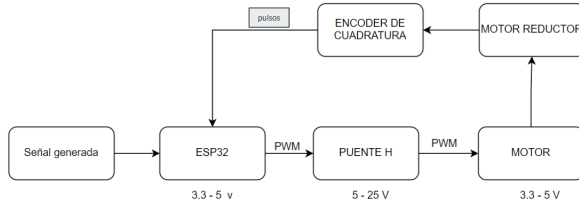


Fig. 2. Diagrama de bloques

Como se observa en el diagrama, la ESP32 va a recibir una señal de entrada que va a servir de referencia para la velocidad del motor. Al microcontrolador se le conecta el Puente H BTS7960, el cual cuenta con dos pines de entrada de PWM, siendo cada uno el encargado de una dirección u otra. Posteriormente se encuentra un módulo de motor, el cual cuenta con el motor, un reductor de rotaciones y el encoder scx-555. Este mismo va a estar enviando constantemente la velocidad actual del motor, para que la ESP32 ajuste la velocidad a su posición requerida.

#### B. Caracterización de la planta

Para poder desarrollar un controlador al motor, tenemos que entender su funcionamiento en tiempo continuo, por lo que se utilizó MATLAB como herramienta para caracterizar el funcionamiento de nuestro motor.

Gracias a MATLAB fue que obtuvimos una función de transferencia de la siguiente manera:

$$G(n) = \frac{0.0008009}{(s^2 + 0.03753s + 0.000426)} \quad (1)$$

Esta función de transferencia cuenta con dos polos y ningún cero, lo que facilita tanto su análisis de comportamiento como el diseño de su control. A la par, este cuenta con una estimación del 85%.

#### C. Diseño de control

Una vez teniendo la función de transferencia de nuestro motor proseguimos con el desarrollo de nuestro controlador. En un inicio se tenía pensado implementar un controlador PID, sin embargo, debido a diversos errores que tuvimos con el procesamiento de la ESP32, y el desbordamiento de las variables, se optó por realizar un controlador P.

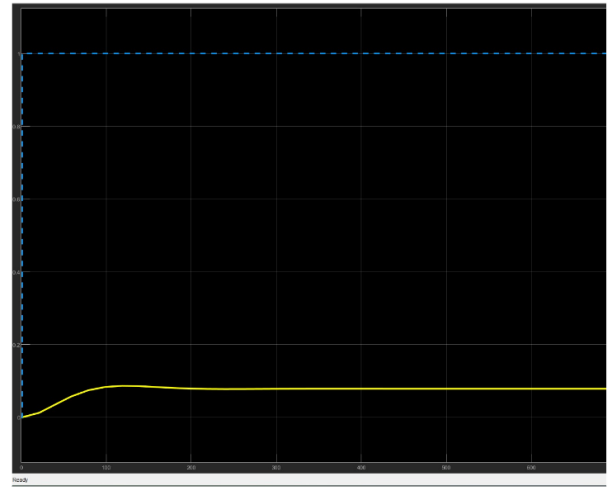


Fig. 3. Respuesta del controlador P ante un impulso o escalón

Dicho controlador cuenta con un valor de  $K_p = 0.1$  y tiene un tiempo de establecimiento de aproximadamente 230 segundos. Se observó su comportamiento al ingresarle al sistema un impulso escalón.

#### D. Implementación Práctica

Una vez determinado nuestro sistema de control, se prosiguió con la programación en Arduino. Se creó el nodo del controlador el cual va a tener un subscriptor que reciba la información de la señal, donde en su callback se implementa la ley de control. Se desarrollaron dos publicadores, los cuales van a enviar los valores de la referencia en rpm y las rpm actuales. Dichas variables van a ser desplegadas por medio de una interrupción de timer. A la par del nodo en el microcontrolador, también se realizó otro nodo pero este corriendo desde la computadora. Este segundo nodo se encarga de realizar una función, en nuestro caso la programamos tanto con una función sinoidal de la forma  $y = \sin(x)$  tanto con una suma de senos de la forma  $y = \sin(2x) + \sin(x)$ . Estas señales se publicaron para ser recibidas por el microcontrolador. Estas señales se diseñaron con el objetivo de corroborar el funcionamiento del controlador a lo largo de una entrada variante en el tiempo.

### IV. RESULTADOS

#### A. $\sin(x)$

Ante una entrada sinusoidal, el sistema reacciona de la siguiente manera como se puede observar en la figura 4. Se puede observar como a pesar de ser un controlador P, llega adecuadamente a la referencia dada, sin embargo, debido a una carencia de la parte integral, por momentos tiene un error considerable.

#### B. $\sin(2x) + \sin(x)$

Para continuar con la comprobación de nuestro sistema, se le ingreso una función diferente, esto para ver el control en el cambio de velocidades.

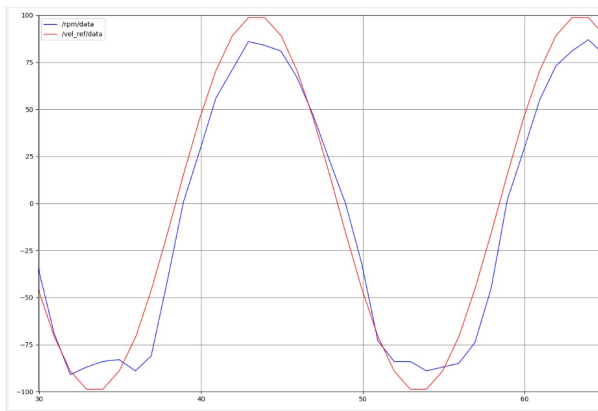


Fig. 4. Respuesta del sistema ante una señal seno

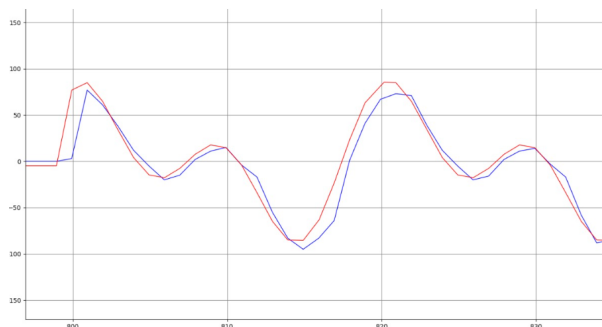


Fig. 5. Respuesta del sistema ante una segunda señal

## CONCLUSIONES

### C. Daniel Nava Mandrón

Debido a la correcta implementación de un controlador de lazo cerrado, y la programación de la ESP32 en el ambiente ROS2, fue que se pudo cumplir con los criterios del reto. Gracias a este trabajo fue que pudimos aprender, por medio del acompañamiento de los profesores del curso, a tener un manejo del sistema operativo Linux y Ubuntu, a la par que la creación e implementación de nodos, subscribers y publishers en el ambiente de ROS2. A pesar de las dificultades que se presentaron, como lo fueron no cambiar el sentido de giro del motor, o no controlar su velocidad adecuadamente, estos mismos se pudieron sobrellevar y superar.

### D. Juan Carlos Jiménez Tapia:

El objetivo de diseñar un sistema de control de lazo cerrado utilizando un controlador proporcional para regular la velocidad de un motor con encoder se alcanzó con éxito. La ejecución comenzó con la caracterización del motor, cuya función de transferencia guió el diseño del controlador. A pesar de las intenciones iniciales de implementar un controlador PID, las limitaciones técnicas relacionadas con la capacidad de procesamiento de la ESP32 y el manejo de variables condujeron a la decisión de aplicar un controlador P. Este proyecto no solo

culminó en la creación de un sistema de control operativo sino que también facilitó una comprensión más profunda de ROS2 en comparación con ROS1, destacando su arquitectura de comunicación avanzada. El desarrollo reforzó conocimientos clave en sistemas de control y demostró la aplicabilidad de ROS2 en el campo de la robótica y la automatización.

## REFERENCES

- [1] Shackleford, W. W. (2004). Force Sensors for Microelectronic Packaging Applications. Wiley-Interscience.
- [2] The MathWorks, Inc. (s.f.). MATLAB. Recuperado de <https://la.mathworks.com/products/matlab.html>.
- [3]

## V. VIDEO DE FUNCIONAMIENTO

Challenge ROS Evidencia 1 \_YOUTUBE