

# SEGUIMIENTO DE TRAYECTORIAS UTILIZANDO POLINOMIOS DE BÉZIER Y LA TARJETA STM32F4-DISCOVERY

Josue Roberto Rodríguez Cerón<sup>1</sup> Jorge Rivero Gómez<sup>1</sup> Everardo Hernández Manuel<sup>1</sup>  
Benjamín Natanael Santiago Nogales<sup>2</sup> Ramón Silva Ortigoza<sup>2</sup> Magdalena Marciano Melchor<sup>2</sup>

<sup>1</sup> Instituto Politécnico Nacional - UPIITA

<sup>2</sup> Instituto Politécnico Nacional - CIDETEC

Laboratorio de Mecatrónica y Energía Renovable

*jrodriguez1705@alumno.ipn.mx*

Boletín No. 106, 1º de mayo de 2025

## Resumen

En el presente trabajo se muestra un control que resuelve la tarea de seguimiento de trayectoria en la velocidad angular de un motor de CD GM-25-370. Se presentan resultados experimentales logrados mediante una tarjeta STM32F4-DISCOVERY, el software Matlab-Simulink y la biblioteca Waijung. Las pruebas experimentales muestran un buen desempeño del sistema en lazo cerrado ante trayectorias deseadas del tipo Bézier.

## I. Introducción

En el presente apartado se explican las generalidades del controlador proporcional-integral-derivativo (PID).

### 1.1 Control Proporcional

El control proporcional, denotado por la letra P, se utiliza para que la acción del controlador sea proporcional al valor de la señal de error. La representación de este control, tanto en el dominio del tiempo como en el de Laplace, es la siguiente:

**Dominio del tiempo:**

$$u(t) = K_p \cdot e(t) \quad (1)$$

**Dominio de Laplace:**

$$U(s) = K_p \cdot E(s) \quad (2)$$

En diagrama de bloques se muestra la relación entre la señal de error y la salida del controlador proporcional.

### 1.2 Control Integral

Control Integral. Denotado por la letra I, este control integral se usa cuando se requiere que el controlador corrija algún error de estado estacionario producido por un valor constante de la señal de referencia. Este tipo de control se utiliza para eliminar desplazamientos sin tener que usar grandes valores de ganancia en los controladores. La representación de este control, tanto en el dominio del tiempo como en el de Laplace, es la siguiente:

**Dominio del tiempo:**

$$u(t) = K_i \int_0^t e(\tau) d\tau \quad (3)$$

**Dominio de Laplace:**

$$U(s) = \frac{K_i}{s} \cdot E(s) \quad (4)$$

El control integral acumula el error a lo largo del tiempo para corregir desviaciones persistentes.

### 1.3 Control Derivativo

Control Derivativo. La representación de este control, este utiliza la razón de cambio de una señal de error para posteriormente introducir un elemento de predicción dentro del control. La representación para este control tanto para el dominio del tiempo como de Laplace es el siguiente:

#### Dominio del tiempo:

$$u(t) = K_d \cdot \frac{de(t)}{dt} \quad (5)$$

#### Dominio de Laplace:

$$U(s) = K_d \cdot s \cdot E(s) \quad (6)$$

El control derivativo anticipa cambios en el error, mejorando la respuesta del sistema.

En el presente artículo se muestra la aplicación de un control para el seguimiento de una trayectoria utilizando un polinomio de Bézier, haciendo uso de la biblioteca Waijung para Matlab y la placa STM32F407VGT6 (STM32F4-Discovery). Las secciones de este trabajo están divididas de tal forma que se pueda proporcionar la mejor explicación posible de la práctica. En la sección II se mencionan los materiales y el software utilizados. En la sección III se muestran los diagramas de bloques implementados dentro del ambiente de Simulink de Matlab, así como los resultados obtenidos, tanto de forma física con el motor de CD, como dentro del programa. Por último, la última sección de este trabajo está dedicada a las conclusiones obtenidas al finalizar la realización de este trabajo.

## II. Materiales y Software

En la realización de este trabajo se emplearon los siguientes elementos:

1. **Motor de CD GM-25-370:** Motor de corriente directa utilizado como sistema a controlar.
2. **Tarjeta STM32F4-DISCOVERY:** Microcontrolador ARM Cortex-M4 para implementar el controlador.
3. **Driver L295N:** Controlador de motor para amplificar las señales de control.
4. **Encoder:** Sensor de posición para realimentación de la velocidad angular.
5. **Matlab-Simulink:** Software para programación y simulación de los controladores.
6. **Biblioteca Waijung:** Extensión de Simulink para programación de microcontroladores STM32.

## III. Programación e Implementación del Prototipo

En este apartado se muestra la construcción del prototipo del sistema a estudiar. Para la elaboración de este trabajo se hace uso de la metodología empleada en trabajos previos para la construcción de servomecanismos. Así, se creó un primer programa (test\_conexion.slx) para realizar la comunicación entre la computadora, por medio de Simulink, y la placa, haciendo uso de bloques de Waijung.

En este programa se cuenta con bloques para definir el MCU de la placa a utilizar, el compilador (GNU ARM) y el tiempo de muestreo de 0.005 s.

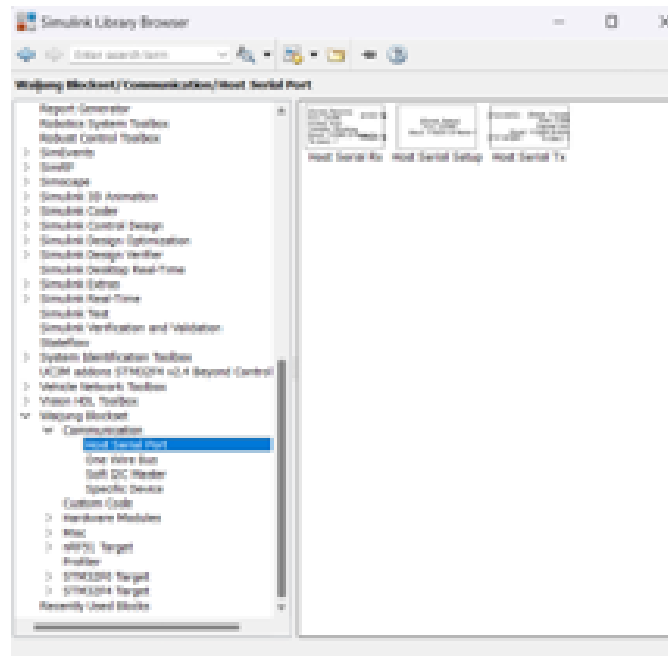


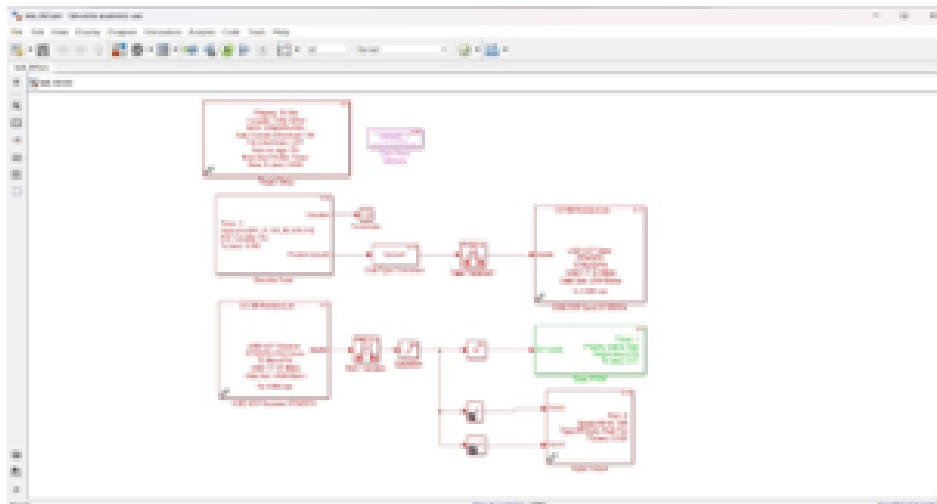
Figura 1 Biblioteca Waijung añadida a Simulink con versión de Matlab 2015b

Posteriormente se realiza la asignación de los pines C6 y C7 para ejecutar la lectura del encoder del motor.



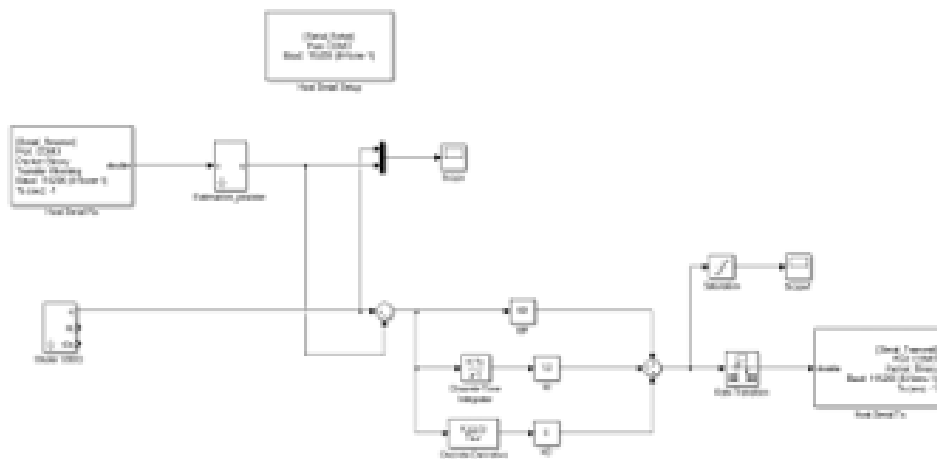
Figura 2 Diagrama para lectura de Encoder del motor y transmisión al VCP para posterior lectura en segundo programa

Por último, mediante el envío de datos a través del USB VCP (Virtual COM Port), se envía una señal PWM ( $T=0.02s$ ) a través del pin A8. Además, se asignan los pines PD12 y PD13 como salidas para el control del sentido de giro del motor. Este primer programa servirá para establecer la comunicación entre la placa y la computadora. Una vez terminado y configurado, se hace clic en el ícono "Build Model" para iniciar la comunicación.



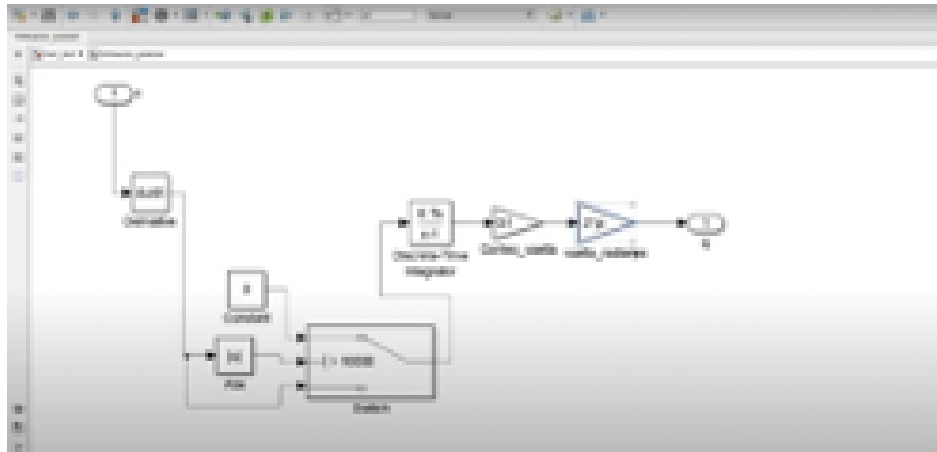
**Figura 3 Programa para realizar lectura de pines PD12 y PD13 para Encoder, envío de PWM al pin A8 para el motor**

Se creó un segundo programa encargado del control PID de la velocidad del motor. Utilizando el Host Serial Setup, se asigna automáticamente el puerto COM de la placa STM32F4 y se configura la velocidad en baudios (bps) para la comunicación. Con los bloques de Host Serial Tx y Rx, se establece la comunicación entre la placa y el programa para controlar el motor.



**Figura 4 Diagrama de bloques Simulink 2015b con control PD**

Mediante bloques de simulink se crea un subsistema (estimación\_posicion) para realizar la estimación de la posición del eje del motor según valores del encoder, convirtiéndolo a radianes. También, se utiliza una ganancia del motor ( $G1$ ), la cual es especificada dentro del Mask de este subsistema.

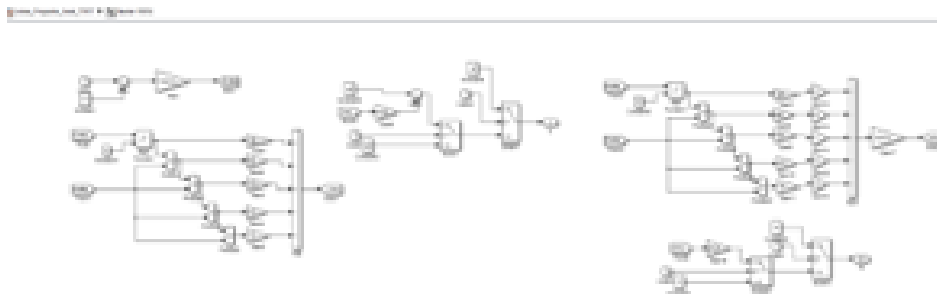


**Figura 5 Subsistema de estimación de posición con ganancia del motor**

Para la programación de la trayectoria deseada de tipo Bezier, se crea un subsistema independiente al controlador. El polinomio de Bézier se define mediante la siguiente ecuación:

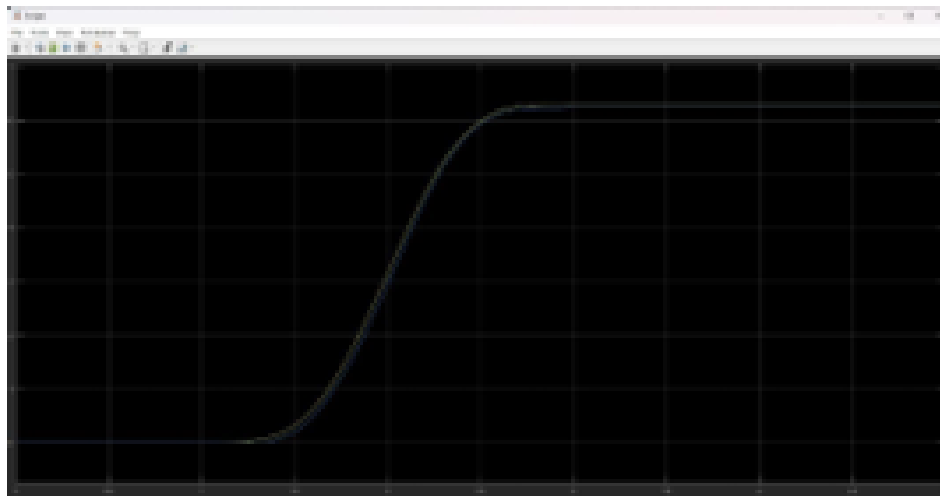
$$B(t) = \sum_{i=0}^n \binom{n}{i} (1-t)^{n-i} t^i P_i, \quad t \in [0, 1] \quad (7)$$

donde  $P_i$  son los puntos de control y  $\binom{n}{i}$  son los coeficientes binomiales.

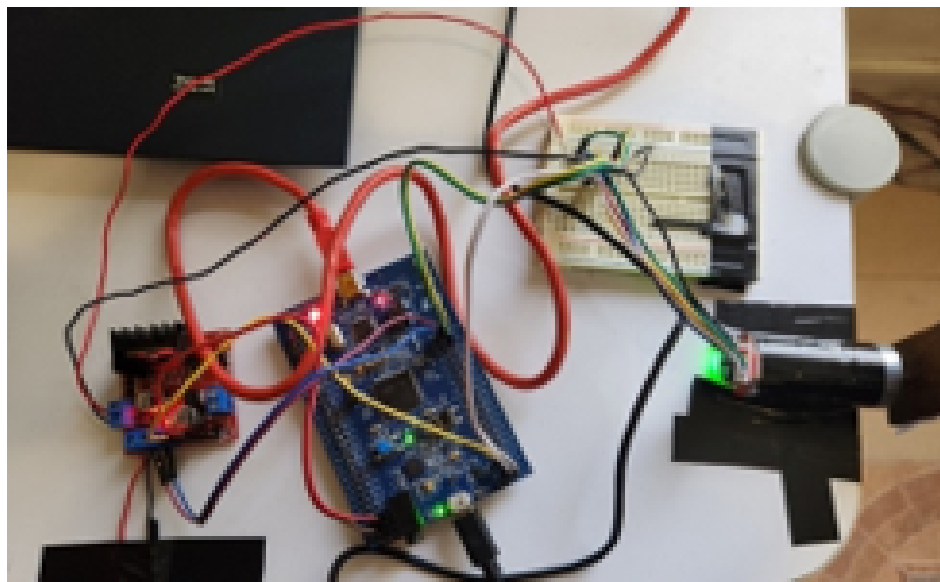


**Figura 6 Subsistema para la obtención de la trayectoria por medio de polinomio de Bézier**

Cabe notar que la sintonización de los valores de las ganancias del controlador PID se realizó por medio del método de Ziegler-Nichols, realizando ajustes hasta obtener el resultado expuesto en la siguiente figura.



**Figura 7 Scope mostrando señal deseada (color amarillo) vs la señal obtenida por lectura de Encoder del motor (color azul)**



**Figura 8 Conexión de placa STM32F4-Discovery con motor y Driver L295N**

#### IV. Conclusiones

El uso de polinomios de Bézier para resolver la tarea de seguimiento de trayectorias es ampliamente empleado en motores. La biblioteca Waijung es un recurso de gran ayuda para establecer la comunicación en tiempo real entre programas de Simulink y una placa de la familia STM32F4. A pesar de la falta de compatibilidad con versiones más recientes de Matlab, sigue siendo una buena opción debido a los diversos problemas encontrados al intentar implementar este tipo de programa con Waijung 2 en versiones más recientes de Matlab utilizando la placa ESP32 de 38 pines.

#### Referencias

- [1] Crowe, K., Tan, K. K., Lee, T. H., Ferdous, R., Katebi, M. R., Huang, H.-P., Jeng, J.-C., Tang, K. S., Chen, G. R., Man, K. F., Kwong, S., Sánchez, A., Wang, Q.-G., Zhang, Y., Zhang, Y., Martin, P., Grimble, M. J., y Greenwood, D. R. (2006). *PID Control*. Springer. ISBN: 978-1-85233-702-5.
- [2] Rojas Castillo, A. R., Luviano Juárez, A., Lozada Castillo, N. B., y Cortez Vega, R. A. (s.a.). *Construcción de un servo-*

*mecanismo de 1 GDL* [En línea]. Disponible: <https://rb.gy/6ky68i>. [Último acceso: 07 11 2024].

**Rodríguez, J., Rivero, J., Hernández, E., Santiago, B., Silva, R., & Marciano, M.** (Mayo - Junio, 2025). Seguimiento de trayectorias utilizando polinomios de Bézier y la tarjeta STM32F4-Discovery. *Boletín UPIITA*. 19 (106).