

Metodología para el diseño de un controlador discreto aplicado en un sistema embebido

Luis Fernando Jimenez Ibarra¹, Santiago Burgueño Ortega², Olga Natalia Portillo Moreno³, Marcelo Gutierrez Davila⁴

Automatización Industrial – Grupo 601

Ingeniería Mecatrónica

Tecnológico de Monterrey

Monterrey, Nuevo León, México

¹A00841522 ²A00838839 ³A00841932 ⁴A00840727

Abstract—En este documento explicamos el diseño, simulación e implementación de un sistema de control de velocidad para un vehículo autónomo mediante actuación diferencial. Obtuvimos funciones de transferencia independientes para los motores izquierdo y derecho, agregando controladores PID en tiempo discreto mediante la técnica de cancelación de polos y ceros. El sistema fue validado en simulación (Matlab/Simulink) y posteriormente embebido en un microcontrolador STM32 utilizando comunicación inalámbrica por radiofrecuencia (nRF24L01) para su evaluación experimental en tierra.

Index Terms—Control PID, Vehículos Autónomos, STM32, nRF24L01, Tiempo Discreto, Cancelación de Polos.

I. OBJETIVO Y MOTIVACIÓN

El control de velocidad es un subsistema crítico para mantener la seguridad, eficiencia y comodidad en vehículos autónomos. Su diseño debe ser robusto y adaptativo a diversas condiciones de operación para garantizar aceleraciones y desaceleraciones suaves. El objetivo de este proyecto es desarrollar y validar experimentalmente algoritmos de control de velocidad precisos, cerrando la brecha entre el diseño teórico de leyes de control y su transición a aplicaciones prácticas embebidas. Este aspecto es fundamental para el avance de la tecnología de conducción autónoma.

II. SÍNTESIS DEL CONTROLADOR DISCRETO

Debido a las discrepancias mecánicas y eléctricas de los actuadores, se decidió identificar la dinámica de cada rueda de manera independiente y aplicar el método de cancelación de polos y ceros para sintonizar dos controladores dedicados.

A. Identificación de la Planta

Para obtener los modelos matemáticos discretos, operando a un tiempo de muestreo $T_s = 0.1$ s, se usaron señales escalón y senoidales al sistema. Para seleccionar el modelo más confiable, se aplicó una validación cruzada: cada modelo identificado fue evaluado tanto con sus propios datos de entrenamiento como con los datos del otro tipo de señal. El criterio de selección fue el coeficiente de determinación R^2 . Como se observa en la Tabla I y Tabla II, el modelo derivado de la respuesta al escalón superó consistentemente al senoidal en ambas pruebas, obteniendo valores de R^2 más altos tanto en sus propios datos como en la validación cruzada

con datos senoidales. Dado que $G_{p,esc}$ obtuvo el mayor R^2 en ambos motores y en ambas condiciones de validación, se justifica su selección como modelo de planta para el diseño del controlador PID.

Para el Motor 1 (Izquierdo), la función de transferencia estimada arrojó un coeficiente de determinación $R^2 = 0.9795$:

TABLE I: Validación Cruzada Motor 1 (Izquierdo)

Modelo vs Datos	R^2
$G_{p,esc}$ vs datos escalón	0.9795
$G_{p,esc}$ vs datos senoidal (cruzado)	0.6084
$G_{p,sen}$ vs datos senoidal	0.9140
$G_{p,sen}$ vs datos escalón (cruzado)	0.4560

$$G_{p1}(z) = \frac{0.1636z^4 + 0.0677z^3}{z^5 - 0.9143z^4 + 0.1135z^3} \quad (1)$$

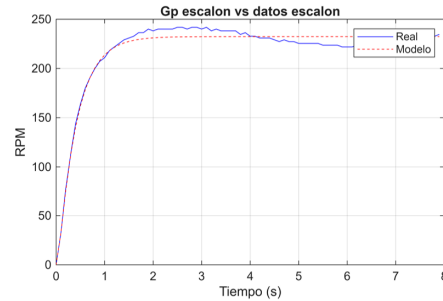


Fig. 1: Ajuste del modelo G_{p1} vs datos de escalón: Motor 1 (Izquierdo)

Para el Motor 2 (Derecho), el modelo presentó un $R^2 = 0.9813$:

TABLE II: Validación Cruzada Motor 2 (Derecho)

Modelo vs Datos	R^2
$G_{p,esc}$ vs datos escalón	0.9813
$G_{p,esc}$ vs datos senoidal (cruzado)	0.5805
$G_{p,sen}$ vs datos senoidal	0.8538
$G_{p,sen}$ vs datos escalón (cruzado)	0.4261

$$G_{p2}(z) = \frac{0.1818z^4 + 0.07952z^3}{z^5 - 0.7755z^4 - 0.002074z^3} \quad (2)$$

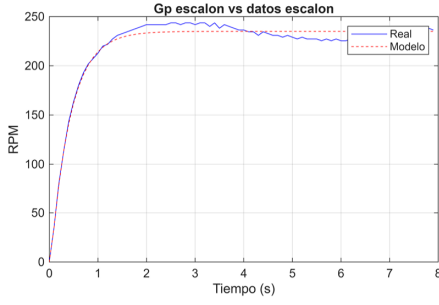


Fig. 2: Ajuste del modelo G_{p2} vs datos de escalón: Motor 2 (Derecho)

B. Parámetros Continuos y Dinámica Deseada

Los requerimientos de diseño establecen un sobreimpulso máximo $M_p = 5\%$ y un tiempo de asentamiento $t_s = 0.5$ s. Para traducir esto, se calcula el factor de amortiguamiento (ζ) y la frecuencia natural no amortiguada (ω_n):

$$\zeta = \frac{|\ln(M_p)|}{\sqrt{\pi^2 + \ln^2(M_p)}} \quad (3)$$

$$\omega_n = \frac{4}{\zeta t_s} \quad (4)$$

Con estos valores, los polos dominantes deseados en el dominio de Laplace (plano S) se ubican en:

$$s_d = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2} \quad (5)$$

C. Mapeo al Plano Z y Estructura por Cancelación

El polo continuo deseado se mapea al dominio digital utilizando la relación de transformación exacta $z_d = e^{s_d T_s}$. Se propone un controlador PID $C(z)$ estructurado para cancelar la dinámica original de la planta $G(z)$. El numerador del controlador se iguala al denominador de la planta, e incluye un polo integrador puro en $z = 1$:

$$C(z) = K_c \frac{den_G}{z(z-1)} \quad (6)$$

Para que el lugar de las raíces cruce exactamente por z_d , se aplica el Criterio de Magnitud $|L(z_d)| = 1$, despejando la ganancia global del controlador (K_c):

$$K_c = \frac{|z_d^2 - z_d|}{|num_G(z_d)|} \quad (7)$$

D. Ecuación de Diferencias y Ganancias PID

Con K_c calculada, se multiplican los coeficientes y se divide el controlador entre z^2 para pasarlo a su forma de diferencias discreta:

$$C(z) = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{1 - z^{-1}} \quad (8)$$

Relacionando mediante Backward Euler, las ganancias son:

$$K_d = q_2 T_s, \quad K_p = -q_1 - 2q_2, \quad K_i = \frac{q_0 + q_1 + q_2}{T_s} \quad (9)$$

E. Parámetros Finales Obtenidos

Para el **Motor 1 (Izquierdo)**: $q_0 = 2.700$, $q_1 = -2.469$, $q_2 = 0.306$. Equivalente a $K_p = 1.856$, $K_i = 5.378$, $K_d = 0.031$. Para el **Motor 2 (Derecho)**: $q_0 = 2.367$, $q_1 = -1.835$, $q_2 = -0.005$. Equivalente a $K_p = 1.845$, $K_i = 5.264$, $K_d = 0.000$.

III. RESULTADOS DE SIMULACIÓN

La simulación representa una etapa previa e indispensable antes de la implementación física, ya que permite validar el comportamiento del controlador en un entorno libre de restricciones de hardware, ruido eléctrico y limitaciones de potencia. A diferencia del sistema real, la simulación garantiza condiciones ideales y repetibles, lo que facilita la detección temprana de errores en el diseño antes de arriesgar el equipo físico. Una vez verificado el desempeño teórico, el controlador se traslada al sistema embebido para su validación experimental bajo condiciones reales de operación.

Los controladores fueron validados en MATLAB simulando pruebas de aceleración y desaceleración. Como se muestra en la Tabla III, ambos motores cumplieron satisfactoriamente con los requerimientos de diseño establecidos.

TABLE III: Índices de Desempeño en Simulación

Índice	Motor 1 (Izq.)	Motor 2 (Der.)
Tiempo de subida (t_r)	0.200 s	0.200 s
Tiempo de asentamiento (t_s)	0.500 s	0.500 s
Sobreimpulso (M_p)	4.05%	4.25%
Error estacionario (e_{ss})	0.00%	0.00%

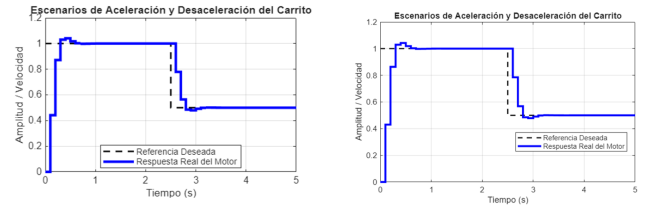


Fig. 3: Simulación en Matlab: Motor izquierdo (izq.) y Motor derecho (der.)

Los índices de desempeño fueron extraídos mediante la función `stepinfo` de MATLAB, aplicando el criterio estándar de estabilización del 2% sobre la respuesta simulada de cada motor.

IV. DESCRIPCIÓN DEL SISTEMA DE CONTROL IMPLEMENTADO

El diseño analítico se implementó en un STM32 (Bluepill) operando a $T_s = 100$ ms. Consta de comunicación inalámbrica (nRF24L01), actuación por Puente H (PWM) y encoders incrementales de cuadratura para la realimentación odométrica. El código gestiona la integración paralela e incluye un mecanismo crítico de Anti-windup, limitando el esfuerzo de control PWM a la banda de $[-255, 255]$ para prevenir la acumulación integral excesiva y la desestabilización durante variaciones bruscas de referencia.

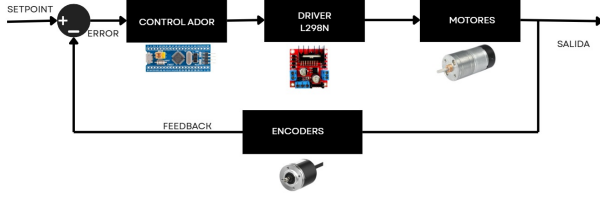


Fig. 4: Diagrama de bloques de lazo cerrado

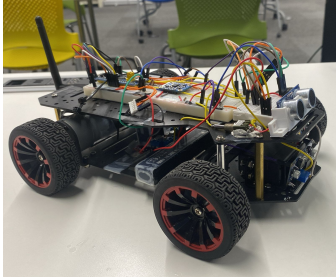


Fig. 5: Fotografía del vehículo autónomo ensamblado con sus componentes

La implementación del algoritmo de control se realizó en Arduino IDE para el microcontrolador STM32 (Bluepill). La ecuación de diferencias (8) se tradujo directamente al código embebido, donde $u[k]$ representa el PWM calculado en cada ciclo de muestreo de $T_s = 100$ ms. El código integra de forma paralela los dos lazos PID discretos, el cálculo del error, la acción integral con Anti-windup y la generación de la señal PWM para cada motor de forma independiente.

```

119 long delta1 = p1 - pulsos1_ant; // Derecho
120 long delta2 = p2 - pulsos2_ant; // Izquierdo
121 pulsos1_ant = p1; pulsos2_ant = p2;
122
123 velocidad_izq = (delta1 / PULSOS_POR_METRO) / (Ts_ms / 1000.0);
124 velocidad_der = (delta2 / PULSOS_POR_METRO) / (Ts_ms / 1000.0);
125
126 // --- CÁLCULO PID IZQUIERDO ---
127 e_k_izq = setpoint - velocidad_izq;
128 u_k_izq = u_k1_izq + (q0_izq * e_k_izq) + (q1_izq * e_k1_izq) + (q2_izq * e_k2_izq);
129
130 // --- CÁLCULO PID DERECHO ---
131 e_k_der = setpoint - velocidad_der;
132 u_k_der = u_k1_der + (q0_der * e_k_der) + (q1_der * e_k1_der) + (q2_der * e_k2_der);
133
134 float FACTOR_ESCALA = 255.0;
135
136 float esfuerzo_pwm_izq = u_k_izq * FACTOR_ESCALA;
137 float esfuerzo_pwm_der = u_k_der * FACTOR_ESCALA;
138
139 // Saturación Izquierda
140 if (esfuerzo_pwm_izq > 255.0) esfuerzo_pwm_izq = 255.0;
141 if (esfuerzo_pwm_izq < -255.0) esfuerzo_pwm_izq = -255.0;
142 if (u_k_izq * FACTOR_ESCALA > 255.0) u_k_izq = 255.0 / FACTOR_ESCALA;
143 if (u_k_izq * FACTOR_ESCALA < -255.0) u_k_izq = -255.0 / FACTOR_ESCALA;
144
145 // Saturación Derecha
146 if (esfuerzo_pwm_der > 255.0) esfuerzo_pwm_der = 255.0;
147 if (esfuerzo_pwm_der < -255.0) esfuerzo_pwm_der = -255.0;
148 if (u_k_der * FACTOR_ESCALA > 255.0) u_k_der = 255.0 / FACTOR_ESCALA;
149 if (u_k_der * FACTOR_ESCALA < -255.0) u_k_der = -255.0 / FACTOR_ESCALA;
150
151 // Prevenir windup
152 if (setpoint == 0.0) {
153     esfuerzo_pwm_izq = 0.0; u_k_izq = 0.0; u_k1_izq = 0.0;
154     e_k_izq = 0.0; e_k1_izq = 0.0; e_k2_izq = 0.0;
155
156     esfuerzo_pwm_der = 0.0; u_k_der = 0.0; u_k1_der = 0.0;
157     e_k_der = 0.0; e_k1_der = 0.0; e_k2_der = 0.0;
158 }
159
160 enviarPWM(esfuerzo_pwm_izq, esfuerzo_pwm_der);

```

Fig. 6: Fragmento del código implementado en Arduino IDE para el control PID de los motores en el STM32

V. PRUEBAS EXPERIMENTALES (INALÁMBRICAS EN TIERRA)

La experimentación se llevó a cabo bajo comunicación inalámbrica continua, enviando un *setpoint* de 1 m/s a través del módulo RF. Las pruebas evaluaron la respuesta transitoria y el esfuerzo de control del PWM frente a las dinámicas reales del vehículo rodando sobre el piso.

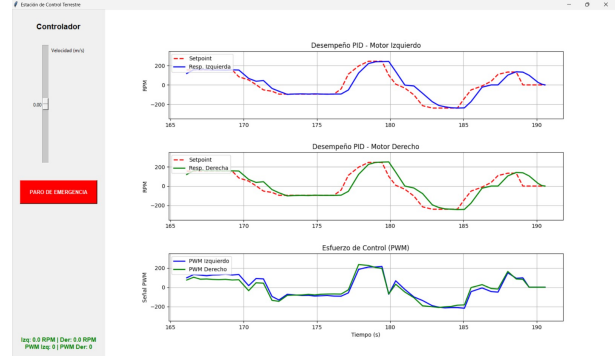


Fig. 7: Datos experimentales: Velocidad vs Setpoint y Esfuerzo de Control PWM

VI. ANÁLISIS DEL DESEMPEÑO EXPERIMENTAL

El comportamiento físico fue diferente al de la simulación ideal, demostrando la complejidad de trasladar un diseño puramente teórico a una implementación mecatrónica real sujeta a fricción y limitaciones de potencia.

En primer lugar, la respuesta real de ambos motores fue completamente sobreamortiguada, registrando un 0.00% de sobreimpulso, lo cual mitigó el $\sim 4\%$ teórico esperado. Este fenómeno se debe a que la inercia del sistema mecánico amortiguó naturalmente la energía dada por los motores.

En segundo lugar, existió una divergencia considerable en el **tiempo de estabilización** (t_s). Mientras que la simulación (libre de límites energéticos) predecía alcanzar el régimen permanente en 0.5 s, el sistema real tardó aproximadamente ~ 1.6 s. Al observar la telemetría del esfuerzo de control, se concluye que en el instante del arranque ($t = 0$), los controladores exigieron el máximo ancho de pulso posible (saturación del PWM en 255). Físicamente, los actuadores estaban operando a su máxima capacidad electromecánica, pero el tiempo de respuesta quedó irremediabilmente limitado por la masa del vehículo y la inercia del arranque inicial.

Adicionalmente, la fricción cinética del terreno y las ligeras caídas de voltaje de la batería bajo carga indujeron un pequeño **error en estado estacionario** (e_{ss}). El error persistente fue de aproximadamente 0.026 m/s, lo que representa un desfase del 2.6% frente al *setpoint* objetivo de 1 m/s.

Este fenómeno derivó en una necesidad de ajuste analítico. Los métodos de evaluación computacional (como la función *stepinfo* de Matlab) utilizan un criterio estándar de estabilización del 2%. Sin embargo, debido a que el error físico constante era ligeramente superior (2.6%), la señal experimental nunca lograba mantenerse dentro de esa estricta banda del 2%, provocando que el cálculo arrojara tiempos fallidos (infinito o *NaN*). Para obtener métricas de diseño representativas del

transitorio físico, **la tolerancia de análisis se ajustó a un criterio del 5%**, permitiendo así una caracterización correcta del asentamiento del carrito.

VII. CÁLCULO DE ÍNDICES DE DESEMPEÑO FÍSICOS

A partir de la data telemétrica analizada bajo un criterio de tolerancia del 5%, se extrajeron las siguientes métricas físicas:

Motor 1 (Izquierdo):

- **Tiempo de Subida (t_r):** 0.9665 s
- **Tiempo de Asentamiento (t_s):** 1.6130 s
- **Sobreimpulso (M_p):** 0.00%
- **Error en Estado Estacionario (e_{ss}):** 0.0228 m/s (2.28%)

Motor 2 (Derecho):

- **Tiempo de Subida (t_r):** 1.0010 s
- **Tiempo de Asentamiento (t_s):** 1.6349 s
- **Sobreimpulso (M_p):** 0.00%
- **Error en Estado Estacionario (e_{ss}):** 0.0265 m/s (2.65%)

TABLE IV: Comparativa Simulación vs Experimental

Índice	Simulación		Experimental	
	M1	M2	M1	M2
t_r (s)	0.200	0.200	0.9665	1.0010
t_s (s)	0.500	0.500	1.6130	1.6349
M_p (%)	4.05	4.25	0.00	0.00
e_{ss} (%)	0.00	0.00	2.28	2.65

VIII. CONCLUSIONES

El proyecto demostró la integración completa de un sistema automotriz autónomo, partiendo del modelado empírico de actuadores y la síntesis matemática discreta, hasta su despliegue embebido e inalámbrico. El uso riguroso de lógica *Anti-windup* demostró ser vital, previniendo fallos sistémicos y matemáticos durante la inevitable saturación física de los actuadores en los instantes de arranque brusco. Los resultados comprueban que el diseño de un PID no sólo reside en su ajuste algorítmico ideal, sino en comprender profundamente la interacción de la planta con perturbaciones del mundo real (fricción, inercia, baterías). Se comprobó que flexibilizar los márgenes analíticos (como pasar del 2% al 5% de tolerancia) es una práctica de ingeniería necesaria para caracterizar sistemas físicos que poseen errores de estado estacionario inherentes a su construcción, validando así la confiabilidad del vehículo para seguir trayectorias de forma controlada.