

## Sintonización de un controlador PID con redes neuronales artificiales Tuning a PID Controller with artificial neural networks

A. Camacho-Ramírez <sup>a,\*</sup>, J.C. Avila-Vilchis <sup>a</sup>, J.M. Benitez-Quintero <sup>b</sup>

<sup>a</sup>Facultad de Ingeniería, Universidad Autónoma del Estado de México, 50000, Toluca, Estado de México

<sup>b</sup>Tecnológico Nacional de México, ITS Purísima del Rincón, Guanajuato, México

### Resumen

Este trabajo presenta una metodología para la sintonización de controladores PID mediante redes neuronales artificiales (ANN) aplicada a Rotatory Servo (RS) equipado con un motor de corriente continua (DC). Se exploran dos enfoques: la sintonización previa y la sintonización dinámica en tiempo real, ambas basadas en datos generados por simulación. La ANN es entrenada exclusivamente con datos simulados, sin requerir un modelo matemático detallado del sistema físico, lo que simplifica significativamente el proceso de diseño. El objetivo de este estudio es demostrar que es posible obtener un rendimiento satisfactorio sin necesidad de realizar análisis complejos ni cálculos manuales por parte del diseñador. Los experimentos abarcan simulaciones y pruebas en un entorno físico, mostrando que la sintonización mediante las ANN, tales resultados validan la capacidad de las ANN para optimizar controladores PID, destacando su eficacia en entornos no lineales y con alta variabilidad. Adicionalmente, se compara con algoritmos genéticos (GA), donde la ANN supera en eficiencia computacional y facilidad de implementación.

**Palabras Clave:** Control, Redes Neuronales Artificiales, Sintonización.

### Abstract

This work presents a methodology for tuning PID controllers using artificial neural networks (ANN), applied to a Rotary Servo (RS) equipped with a direct current (DC) motor. Two approaches are explored: offline pre-tuning and real-time dynamic tuning, both based on simulation-generated data. The ANN is trained exclusively with simulated data, without requiring a detailed mathematical model of the physical system, which significantly simplifies the controller design process. The aim of this study is to demonstrate that satisfactory performance can be achieved without the need for complex analysis or manual calculations by the designer. The experiments include both simulations and physical tests, showing that tuning through ANN is effective. These results validate the ability of ANN to optimize PID controllers and highlight their effectiveness in nonlinear environments with high variability. A comparative analysis with genetic algorithms (GA) highlights ANN's advantages in computational efficiency and implementation ease.

**Keywords:** Control, Artificial neural networks, Tuning.

### 1. Introducción

Los motores DC son esenciales en muchas aplicaciones debido a que su control es preciso en relación a su velocidad y par, simplicidad operativa, y rápida respuesta a cambios de carga, tales como vehículos eléctricos, cintas transportadoras y robots industriales Elmorshedy *et al.* (2021). Son muy utilizados debido a un par alto a bajas velocidades y su facilidad de mantenimiento, especialmente en versiones sin escobillas Kommula y Kota (2022). Los PID siguen siendo la piedra angular del control automático, gracias a su versatilidad y evolución mediante

técnicas modernas. Su integración con métodos adaptativos e inteligentes amplía su aplicabilidad en sistemas complejos, desde robótica hasta biomedicina, manteniendo relevancia en la era de la Industria 4.0 Borase *et al.* (2021). Además, su compatibilidad con tecnologías modernas los hace fundamentales para la automatización y el Internet de las cosas (IoT) Radonjić *et al.* (2022).

Las técnicas de control PID y Proporcional-Integral (PI) siguen siendo las más utilizadas en sistemas de control debido a su simplicidad y efectividad. Se estima que más del 95 % de las

\* Autor para correspondencia: acamachor@uaemex.mx

**Correo electrónico:** acamachor@uaemex.mx (Adrian Camacho-Ramírez), jcavilav@uaemex.mx (Juan Carlos Avila-Vilchis), manuel.bq@purisima.tecnm.mx.com (Jose Manuel Benitez-Quintero)

**Historial del manuscrito:** recibido el 24/01/2025, última versión-revisada recibida el 02/09/2025, aceptado el 04/09/2025, en línea (postprint) desde el 19/09/2025, publicado el 05/01/2025. **DOI:** <https://doi.org/10.29057/icbi.v13i26.14471>



aplicaciones en procesos industriales hacen uso de estos controladores. Sin embargo, su desempeño puede verse afectado en situaciones donde el sistema enfrenta variaciones de carga o condiciones de inestabilidad, limitando su capacidad de respuesta óptima Lee *et al.* (2020). Además, su desempeño está estrechamente ligado a una correcta sintonización de los parámetros y ganancias del controlador, lo cual puede convertirse en un desafío significativo en aplicaciones de alta complejidad Somefun *et al.* (2021).

La sintonización de controladores PID es un proceso esencial para garantizar que el sistema controlado responda de manera eficiente y estable ante diferentes condiciones de operación. La elección de los parámetros  $K_p$  (ganancia proporcional),  $K_i$  (ganancia integral) y  $K_d$  (ganancia derivativa) influye directamente en el rendimiento del sistema, afectando su estabilidad, tiempo de respuesta y precisión. Existen diversos métodos clásicos de sintonización de controladores PID, como los métodos de Ziegler-Nichols, Cohen-Coon y la respuesta en bucle abierto, los cuales son ampliamente utilizados. Sin embargo, presentan limitaciones al aplicarse a sistemas complejos o no lineales Åström y Hägglund (2006). Para superar estas limitaciones, se han propuesto enfoques más avanzados. Por ejemplo, en Gadekar *et al.* (2020), se presenta un algoritmo de sintonización automática del PID que permite adaptar sus parámetros dinámicamente según la velocidad deseada en motores BLDC. En Abdelghany *et al.* (2023), se introduce un controlador PID autoajutable basado en lógica difusa, diseñado para optimizar el rendimiento de servomotores. Por su parte, Amuthambigaiyin Sundari y Maruthupandi (2022) explora la sintonización del PID mediante algoritmos metaheurísticos, analizando su desempeño en sistemas de tanques interactivos y no interactivos bajo condiciones de perturbación. En Li y Gao (2021), se propone un método de sintonización robusta para controladores PID fraccionales, orientado a preservar el margen de fase en lazo abierto ante incertidumbres en los parámetros de la planta. Otros enfoques avanzados recurren a algoritmos de optimización, como los algoritmos genéticos (AG), utilizados en Cao (2020) para la sintonización de PID en sistemas electrohidráulicos complejos, logrando una reducción de oscilaciones y una mejora en la precisión del control. Asimismo, el aprendizaje automático ha cobrado relevancia, destacando el uso de ANN que permiten ajustar los parámetros del controlador en función del comportamiento dinámico del sistema. En Debnath *et al.* (2020), se emplea una ANN para la sintonización adaptativa de un controlador PID destinado a la regulación de frecuencia en sistemas eléctricos con generación distribuida. En conjunto, la sintonización automática y adaptativa de controladores PID resulta especialmente beneficiosa en entornos con rápidas variaciones de carga, al mejorar la robustez y la capacidad de respuesta del sistema de control.

En términos generales, la sintonización de un controlador PID se puede clasificar en tres tipos principales, como se ilustra en la Figura 1, los métodos basados en modelo requieren un conocimiento detallado del sistema, incluyendo su modelo matemático y parámetros específicos. Por otro lado, los métodos libres de modelo se basan puramente en datos y utilizan técnicas de optimización para llevar a cabo la sintonización. Finalmente, los métodos híbridos combinan el uso del modelo del sistema con un análisis basado en datos, aprovechando lo mejor

de ambos enfoques para lograr una sintonización más efectiva y robusta Somefun *et al.* (2021).

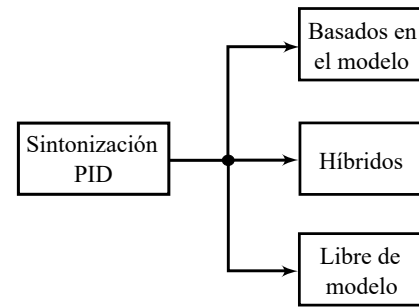


Figura 1: Clasificación para sintonización.

En este contexto, una ANN que cae en la sintonización libre de modelo puede ajustarse para predecir y seleccionar los valores de  $K_p$ ,  $K_i$  y  $K_d$  en tiempo real, mejorando significativamente el control adaptativo del sistema.

En este artículo, se abordará la sintonización de un controlador PID utilizando dos enfoques distintos. En el primer enfoque, las ganancias del controlador PID serán preestablecidas a través de una ANN. Esta red se entrenará con datos obtenidos de simulaciones, buscando una solución óptima para determinar los valores adecuados de las ganancias. En el segundo enfoque, las ganancias del PID serán ajustadas de manera dinámica por una ANN, permitiendo que estas se adapten en tiempo real a los requerimientos del sistema.

El sistema considerado para este estudio es un RS SRV02 que contiene un motor DC. Inicialmente, se realizarán simulaciones para ambos enfoques de sintonización con el objetivo de evaluar su desempeño. Posteriormente, los métodos serán implementados en el sistema físico para validar su eficacia en un entorno real. Este enfoque permite explorar tanto la eficacia de las ganancias predefinidas como la flexibilidad de las ganancias dinámicas, ofreciendo un análisis integral del control adaptativo.

## 2. Control de velocidad de un motor DC

Comúnmente, la selección de las ganancias PID para un motor DC se realiza utilizando métodos basados en modelos, los cuales requieren un análisis detallado como se describirá a continuación.

El modelado matemático de motores DC es un paso fundamental para comprender y diseñar sistemas de control eficientes. Dicho modelo permite establecer una relación entre la entrada, como el voltaje aplicado, y la salida, que puede ser la velocidad angular o la posición del motor. En la mayoría de las aplicaciones, la dinámica del motor puede aproximarse a un sistema de primer orden. Esta simplificación facilita tanto el análisis como el diseño del controlador, permitiendo optimizar el rendimiento y reducir la complejidad del sistema.

El modelo matemático de un motor DC se deriva de las ecuaciones que describen tanto el circuito eléctrico del inducido como la dinámica mecánica del rotor. Las ecuaciones básicas son:

**Ecuación del circuito eléctrico:**

$$V(t) = L \frac{di(t)}{dt} + Ri(t) + E(t) \quad (1)$$

donde:  $V(t)$  es el voltaje aplicado al inducido,  $L$  es la inductancia del inducido,  $R$  es la resistencia del inducido,  $i(t)$  es la corriente en el inducido y  $E(t)$  es la fuerza electromotriz inducida, proporcional a la velocidad angular  $\omega(t)$ , es decir,  $e(t) = K_e \omega(t)$ .

**Ecuación de la dinámica mecánica:**

$$J \frac{d\omega(t)}{dt} + B\omega(t) = K_t i(t) \quad (2)$$

donde  $J$  es el momento de inercia del rotor,  $B$  es el coeficiente de fricción viscosa,  $K_t$  es la constante de par del motor y  $\omega(t)$  es la velocidad angular.

**2.1. Aproximación a un Sistema de Primer Orden**

La simplificación del modelo del motor DC (Ecuaciones 1 y 2) a un sistema de primer orden es ampliamente conocida y reportada en la literatura Franklin *et al.* (2010). Esta simplificación desprecia el término inductivo ( $L \frac{di(t)}{dt} \approx 0$ ), válido cuando  $\tau_e = L/R \ll \tau_m = J/B$  entonces (1) se convierte en:

$$V(t) = Ri(t) + K_e \omega(t) \quad (3)$$

Despejando  $i(t)$  de la Ecuación (2)

$$i(t) = \frac{J}{K_t} \frac{d\omega(t)}{dt} + \frac{B}{K_t} \omega(t) \quad (4)$$

Sustituyendo (4) en (3), entonces

$$\begin{aligned} V(t) &= R \left( \frac{J}{K_t} \frac{d\omega(t)}{dt} + \frac{B}{K_t} \omega(t) \right) + K_e \omega(t) \\ &= \frac{JR}{K_t} \frac{d\omega(t)}{dt} + \left( \frac{BR}{K_t} + K_e \right) \omega(t) \end{aligned} \quad (5)$$

Aplicando transformada de Laplace a (5)

$$\frac{JR}{K_t} s\omega(s) + \left( \frac{BR}{K_t} + K_e \right) \omega(s) = V(s) \quad (6)$$

Entonces la función de transferencia

$$G(s) = \frac{\omega(s)}{V(s)} = \frac{1}{\frac{JR}{K_t} s + \left( \frac{BR}{K_t} + K_e \right)} \quad (7)$$

Por lo tanto el modelo se puede expresar como un sistema estándar de primer orden como

$$G(s) = \frac{K_m}{\tau_m s + 1} \quad (8)$$

donde:

$$K_m = \frac{K_t}{BR + K_t K_e} \quad (\text{Ganancia estática})$$

$$\tau_m = \frac{JR}{BR + K_t K_e} \quad (\text{Constante de tiempo})$$

Esta función de transferencia (Ecuación (8)) muestra que el motor se comporta como un sistema de primer orden, donde la

constante de tiempo  $\tau_m$  define la rapidez de la respuesta del sistema y la ganancia  $K_m$  representa la relación entre la entrada y la salida en estado estacionario.

La representación de un motor DC como un sistema de primer orden simplifica el diseño de controladores, ya que permite el uso de controladores PID y PI que se ajustan de manera efectiva para mejorar la estabilidad y el rendimiento del sistema. Esta aproximación es válida cuando la inductancia es despreciable y el comportamiento del motor puede ser dominado por las características mecánicas y eléctricas principales.

**2.2. Rotatory Servo SRV02**

Para este caso de estudio se llevan a cabo las simulaciones y pruebas experimentales mediante el dispositivo RS SRV02, ilustrado en la Figura 2, el cual constituye una plataforma versátil e intuitiva para la realización de experimentos relacionados con el control de posición angular (en grados) y velocidad angular (en revoluciones por minuto, rpm). Este equipo, desarrollado por Quanser®, es ideal para la introducción y enseñanza de conceptos y teorías fundamentales de control. Además, ha sido adaptado para la adquisición de datos mediante un microcontrolador de bajo costo, el Arduino Due, mejorando su funcionalidad y accesibilidad en entornos educativos y experimentales.



Figura 2: Rotary Servo SRV02

La unidad principal del RS SRV02 consiste en un sistema con engranajes. Un motor DC que acciona un piñón pequeño, el cual está acoplado a un engranaje de mayor diámetro que rota alrededor del eje de interés. La posición angular de este eje se mide con un codificador óptico de alta resolución, o encoder el cual se emplea para estimar la velocidad angular del motor, proporcionando datos precisos tanto de posición como de movimiento, algunos datos que detalla al RS SRV02 se muestran en la Tabla 1.

Tabla 1: Especificaciones del RS SRV02

Dimensiones	15cm x 15 cm x 18 cm
Masa	1.2 kg
Voltaje nominal	6 V
Corriente máxima	1 A
Resolución del encoder	4096 pulsos por revolución
Velocidad máxima	60 rpm
Caja reductora	70:1

Para implementar el control PID en el sistema RS SRV02, fue necesario determinar la función de transferencia del sistema. Esto se logró aplicando un voltaje de entrada específico (PWM = 255) y registrando la respuesta en términos de velocidad angular, como se muestra en la Figura 3. Los datos de entrada (voltaje) y salida (RPM) fueron analizados utilizando el método de *System Identification* en Matlab®, obteniendo una aproximación matemática del sistema RS SRV02 como una función de transferencia de primer orden. Este procedimiento permitió capturar con precisión la dinámica del motor, facilitando su modelado y diseño de control.

La función de transferencia obtenida, presentada en la Ecuación 9, describe el modelo matemático del RS SRV02, permitiendo capturar su dinámica dominante:

$$G(s) = \frac{4.673}{s + 4.676} \quad (9)$$

Este modelo de primer orden, como se mencionó previamente, sirve como la base para el diseño y la sintonización del controlador PID. Su simplicidad facilita tanto el análisis del comportamiento del sistema como el desarrollo de estrategias de control que optimicen su desempeño.

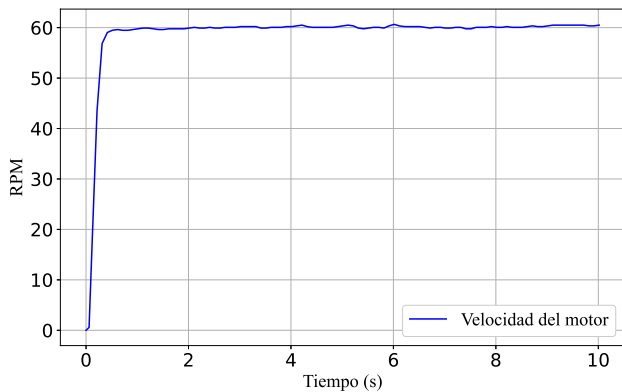


Figura 3: Curva de velocidad del motor con PWM=255.

### 2.3. Plataforma experimental

Para llevar a cabo la parte experimental de la sintonización de un controlador PID mediante una ANN en el sistema RS SRV02, se utilizó una plataforma experimental representada en la Figura 4. Esta figura ilustra el esquema de los componentes fundamentales involucrados en la implementación del control, los cuales se detallan a continuación:

1. Computadora: Equipada con programas como Matlab/Simulink, Arduino IDE y Python, utilizados para el diseño y simulación del sistema. Además, la computadora establece la comunicación con el Arduino Due.
2. Arduino Due: Recibe los datos del encoder, procesa la señal y genera el pulso PWM correspondiente al controlador PID.
3. Controlador de motor L298N: Convierte la señal PWM proveniente del Arduino en un voltaje aplicado al motor. Este voltaje es suministrado por una fuente de voltaje.

4. Planta RS SRV02: Representa el sistema físico cuya velocidad angular es controlada.

Este conjunto de elementos permite la implementación y evaluación práctica del sistema de control de velocidad, integrando hardware y software de forma eficiente.

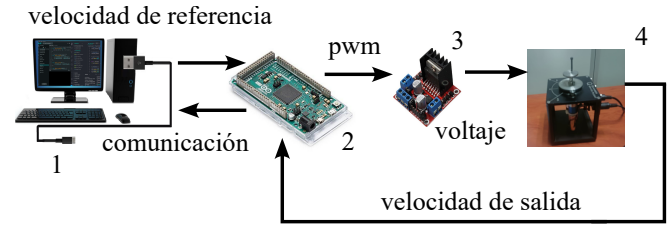


Figura 4: Diagrama del sistema

### 3. ANN para control

El uso de técnicas de aprendizaje profundo, como las ANN, ha sido explorado ampliamente en aplicaciones de control para motores DC, como en Gundogdu *et al.* (2021) donde la combinación de las ANN con identificación de sistemas ha demostrado ser efectivo para el control de velocidad en motores DC, superando métodos convencionales como el Filtro de Kalman Extendido en estados transitorios y estacionarios. Las ANN, cuyo elemento fundamental es el perceptrón, representan una de las formas más antiguas y conocidas de redes neuronales Kruse *et al.* (2022).

El perceptrón, inspirado en el funcionamiento de las neuronas biológicas, procesa entradas ponderadas, las combina linealmente y aplica una función de activación para generar una salida. Aunque originalmente diseñado para resolver problemas de clasificación lineal, su concepto básico ha evolucionado hacia arquitecturas avanzadas que pueden modelar sistemas no lineales, aproximar funciones y optimizar parámetros en tiempo real Géron (2022).

El perceptrón tiene una arquitectura como la que se muestra en la Figura 5, en la cual se identifican los valores de entrada  $x_1, x_2, \dots, x_n$  y un valor de salida  $y$ .

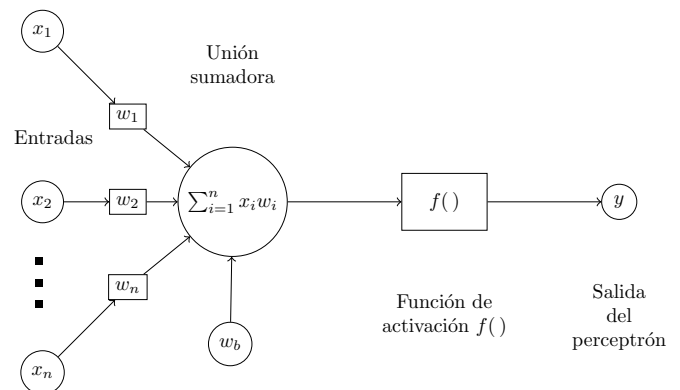


Figura 5: Estructura de un perceptrón

Las entradas corresponden a las características del conjunto de datos de entrenamiento, mientras que la salida representa la respuesta esperada para cada muestra de dicho conjunto. Cada

entrada  $x_i$  está asociada a un peso  $w_i$  donde  $i = 1, 2, 3, \dots, n$ ,  $w_b$  representa el peso de una entrada fija o bias. El valor de salida de la neurona se obtiene calculando la suma ponderada de las entradas y sus respectivos pesos mas el bias. Esta suma es procesada a través de una función de activación  $f()$  que transforma el valor resultante en una nueva salida, ajustada para capturar la no linealidad del modelo y facilitar la toma de decisiones.

Por lo tanto, la capacidad de procesamiento de la ANN se basa en la fortaleza de las conexiones representadas por los pesos. Estas conexiones se ajustan mediante un proceso de adaptación o aprendizaje, que utiliza un conjunto de patrones de entrenamiento para optimizar los pesos y mejorar el desempeño de la red en tareas específicas. Este mecanismo permite a la ANN identificar patrones, modelar relaciones complejas y generalizar a nuevos datos Gurney (2017).

### 3.1. Estructura de la Red Neuronal

La estructura de control propuesta para la sintonización del controlador PID mediante la red neuronal artificial (ANN) se ilustra en la Figura 6.

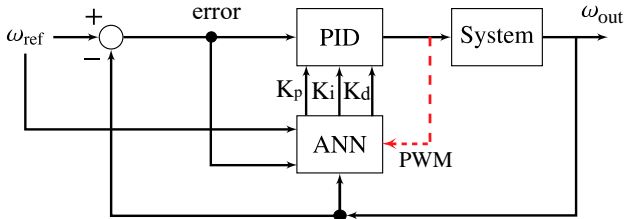


Figura 6: Diagrama de control.

La red neuronal recibe como entradas la velocidad de referencia ( $\omega_{ref}$ ), la velocidad de salida del motor ( $\omega_{out}$ ) y el error ( $e_\omega = \omega_{ref} - \omega_{out}$ ). Como salidas, la red proporciona las ganancias  $K_p$ ,  $K_i$  y  $K_d$  del controlador PID ya sea para una sintonización previa o de forma dinámica. Adicionalmente, la señal de control, representada por la línea en rojo, es generada por el Arduino Due para el control del RS, sustituyendo directamente a  $\omega_{out}$  para la parte experimental.

### 3.2. Entrenamiento

Para llevar a cabo la etapa de entrenamiento mediante la ANN, se propone la estructura mostrada en la Figura 7. Esta configuración consta de cuatro entradas:  $x_1$ , que representa la velocidad de referencia del motor;  $x_2$ , que corresponde al error entre la velocidad de referencia y la velocidad de salida;  $x_3$ , que indica la velocidad de salida y para la parte experimental representa el PWM; y  $x_4$ , que actúa como el término de sesgo o bias. La red incluye dos capas ocultas, cada una con 10 neuronas, y tres salidas, correspondientes a los parámetros del controlador PID:  $y_1 = K_p$ ,  $y_2 = K_i$  y  $y_3 = K_d$ . Esta estructura permite que la ANN ajuste los valores del controlador en función de las condiciones del sistema.

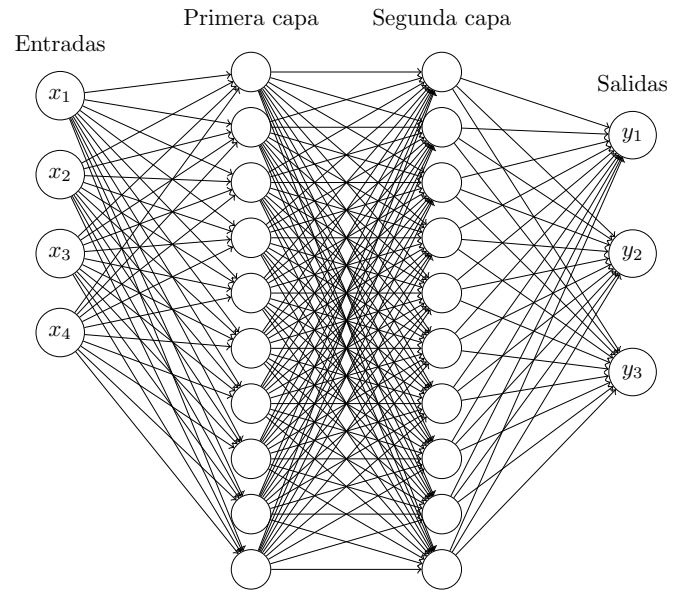


Figura 7: Estructura de la ANN

En las capas 1 y 2 se utilizó la función de activación ReLU (Rectified Linear Unit), una de las más populares y ampliamente empleadas en redes neuronales debido a su capacidad para acelerar el proceso de entrenamiento y mejorar el rendimiento del modelo en diversos escenarios. La representación matemática de esta función se presenta en la Ecuación 10.

$$ReLU(x) = \max(0, x) \quad (10)$$

La función ReLU devuelve la entrada si es positiva y cero en caso contrario, lo que la hace más eficiente que funciones como la sigmoide o la tangente hiperbólica. Además permite entrenar redes neuronales dado que, como se ha demostrado, su comportamiento puede entenderse dentro de un marco variacional que la vincula con soluciones óptimas, facilitando así tanto la interpretación como la optimización de modelos entrenados Parhi y Nowak (2020).

El optimizador utilizado para la red neuronal es ADAM (Adaptive Moment Estimation), un algoritmo ajusta los parámetros del modelo (en este caso, los pesos de la red neuronal) para minimizar una función de costo o pérdida, mejorando así el rendimiento del modelo Lihua (2022). Sus principales ventajas incluyen una mayor eficiencia y la corrección de sesgos en los primeros pasos, lo que evita inestabilidades en la optimización.

Para mejorar la pertinencia del proceso de entrenamiento, se redefine la función de costo como el Error Cuadrático Medio (MSE) entre la velocidad de referencia y la salida del sistema en el dominio temporal. Esta nueva métrica evalúa directamente el desempeño del controlador bajo las ganancias PID propuestas por la red neuronal, permitiendo que el aprendizaje se enfoque en minimizar el error de seguimiento. La expresión formal del MSE queda definida como:

$$MSE = \frac{1}{N} \sum_{i=1}^N (\omega_{ref}(t_i) - \omega_{out}(t_i))^2 \quad (11)$$

donde  $\omega_{ref}(t_i)$  representa el valor de la velocidad de referencia en el instante  $t_i$ ,  $\omega_{out}(t_i)$  es la salida real del sistema a ese



mismo instante, y  $N$  es el número total de muestras de tiempo consideradas durante la simulación. Esta formulación permite que la ANN optimice directamente el desempeño del sistema, enfocándose en reducir el error de seguimiento.

### 3.3. Configuración para el entrenamiento

El entrenamiento se configuró con un total de 1,000 épocas, empleando el 80 % del conjunto de datos para el proceso de aprendizaje. Al utilizar el optimizador ADAM, se estableció una tasa de aprendizaje inicial de 0.01. Este valor se mantuvo debido a la capacidad de ADAM para ajustar automáticamente la tasa de aprendizaje durante el entrenamiento, adaptándola a las características de los gradientes y mejorando así la eficiencia del modelo.

### 3.4. Primer enfoque: sintonización previa

Para entrenar la red neuronal, es fundamental contar con un conjunto de datos representativo que capture cómo el sistema responde bajo diferentes condiciones de operación y configuraciones PID. Con este objetivo, se consideraron los siguientes pasos:

- Se simuló la respuesta del motor de corriente continua (DC) utilizando múltiples combinaciones de ganancias PID.
- Se recopilaron los datos correspondientes a  $\omega_{ref}$ ,  $\omega_{out}$ ,  $e_{\omega}$  asociado para cada configuración y el tiempo de simulación  $t$ .
- Se almacenaron las ganancias  $K_p$ ,  $K_i$  y  $K_d$  empleadas en cada simulación, para usarlas como etiquetas en el proceso de entrenamiento.

Para este propósito, se llevaron a cabo 100 simulaciones destinadas a entrenar la ANN. Cada simulación tuvo una duración de 10 segundos, y los datos fueron recopilados a una frecuencia de 100 Hz, lo que permitió capturar una muestra completa y detallada del comportamiento del sistema bajo diferentes configuraciones, un ejemplo representativo de estas simulaciones se presenta en la Figura 8, donde se ilustran 25 de los 100 experimentos realizados para la recopilación de datos. Esta figura permite observar la diversidad de respuestas obtenidas durante el proceso de simulación, las cuales son fundamentales para el entrenamiento de la ANN.

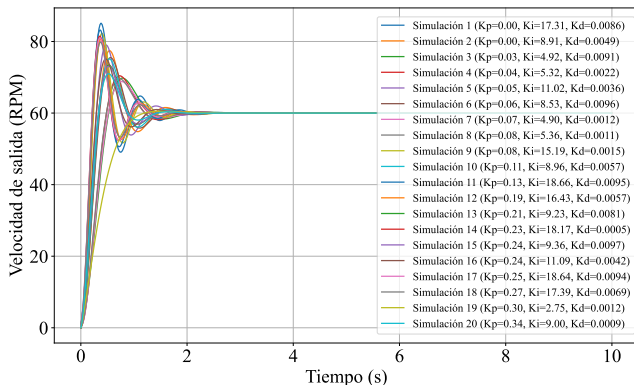


Figura 8: Respuesta del motor ante diferentes valores en las ganancias.

Este enfoque basado en simulación se adoptó de manera intencional como una primera etapa metodológica, permitiendo generar datos de entrenamiento de forma segura, rápida y controlada. La estrategia consiste en entrenar la ANN utilizando un modelo previamente validado, y posteriormente aplicar los resultados obtenidos sobre la planta física. De este modo, se minimizan riesgos operativos y se garantiza una transición más confiable del entorno simulado al sistema real.

### 3.5. Sintonización dinámica

El diseño de un control basado en redes neuronales para ajustar dinámicamente las ganancias  $K_p$ ,  $K_i$  y  $K_d$  de un controlador PID implica utilizar una ANN que modifique estos parámetros en función de las condiciones del sistema. Las entradas de la ANN incluye variables  $\omega_{ref}$ ,  $\omega_{out}$ ,  $e_{\omega} = \omega_{ref} - \omega_{out}$  y la señal de control (PWM). La red neuronal procesa estas entradas para calcular las ganancias del controlador PID, adaptándolas en tiempo real para mejorar la respuesta del sistema frente a perturbaciones externas y cambios en la referencia.

La ANN se diseña con dos capas ocultas y diez neuronas para capturar las relaciones no lineales entre las entradas y las salidas. Su entrenamiento requiere un conjunto de datos obtenido mediante simulaciones del sistema bajo diferentes configuraciones PID. Durante la operación en tiempo real, la ANN recibe los datos del sistema, calcula las ganancias  $K_p$ ,  $K_i$  y  $K_d$ , y ajusta el PID dinámicamente. Esto permite optimizar el desempeño del motor DC frente a condiciones variables y perturbaciones externas, garantizando una respuesta rápida y precisa.

## 4. Implementación

Como se mencionó, la implementación de la ANN para la sintonización de un controlador PID se desarrolló en dos enfoques principales. En el primer enfoque, llevado a cabo de manera previa, la ANN se empleó para determinar los valores iniciales de las ganancias  $K_p$ ,  $K_i$  y  $K_d$ , que posteriormente fueron implementados en el sistema RS SRV02. En el segundo enfoque, la ANN operó en tiempo real, ajustando de forma dinámica los valores de las ganancias directamente en el sistema RS SRV02, lo que permitió una sintonización adaptativa y continua, optimizando el desempeño del controlador frente a cambios y perturbaciones.

### 4.1. Simulación de sintonización previa

Tras el entrenamiento de la ANN, esta proporcionó la sintonización del controlador PID en función de las diversas entradas, simplificando significativamente el proceso de toma de decisiones para su implementación en un entorno real. Las ganancias obtenidas fueron:

$$K_p = 8.61, \quad K_i = 8.52, \quad K_d = 0.01$$

La gráfica correspondiente a la simulación del sistema se presenta en la Figura 9, donde se utiliza una velocidad de referencia de  $\omega_{ref} = 60\text{rpm}$ , la cual representa la velocidad máxima del sistema RS SRV02.

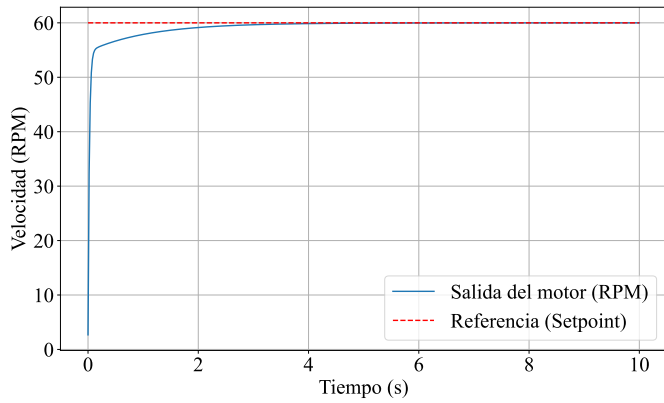


Figura 9: Control de velocidad del RS SRV02 con PID ante una entrada constante con sintonización de ganancias previa.

Si se tiene una referencia que varía con respecto al tiempo, alcanzando los siguientes valores: 10 rpm de 0 a 5 segundos, 20 rpm de 5 a 15 segundos, 40 rpm de 15 a 25 segundos, 60 rpm de 25 a 35 segundos, 30 rpm de 35 a 45 segundos, 15 rpm de 45 a 55 segundos y 0 rpm después de 55 segundos, los resultados obtenidos se muestran en la Figura 10.

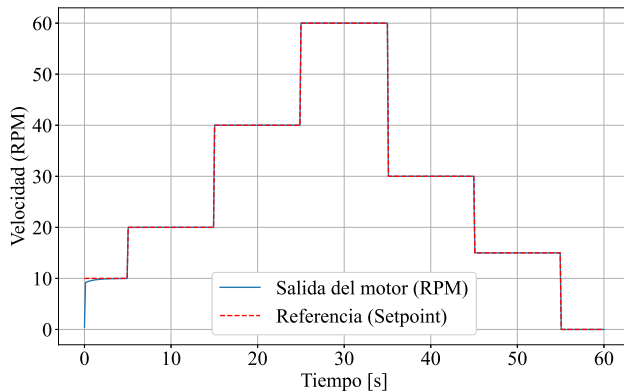


Figura 10: Control de velocidad del RS SRV02 con PID ante diferentes velocidades de referencia con sintonización de ganancias previa.

#### 4.2. Simulación de sintonización dinámica

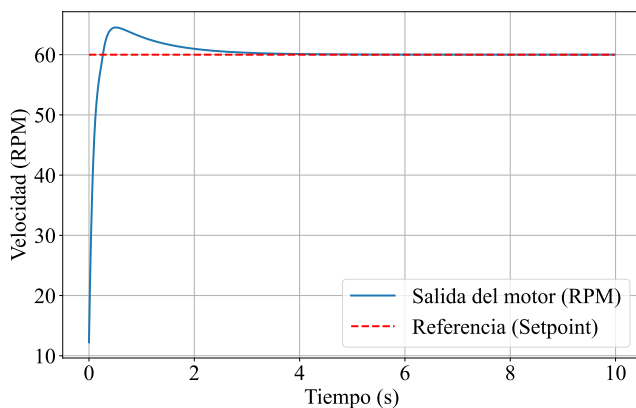


Figura 11: Control de velocidad del RS SRV02 con PID ante una entrada constante con sintonización de ganancias dinámicas.

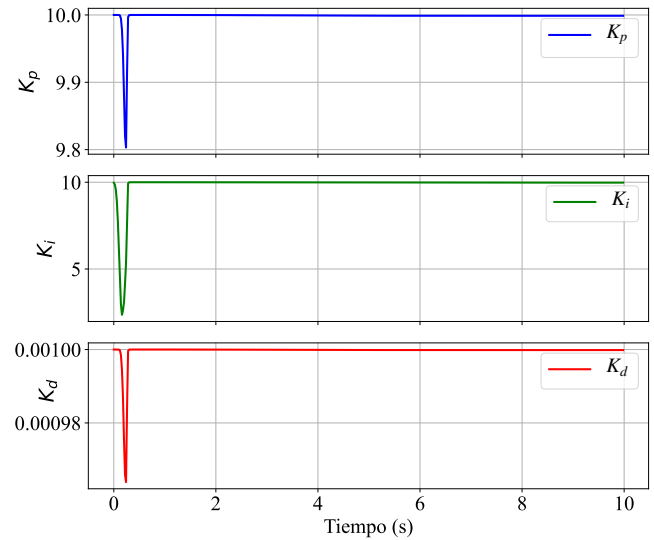


Figura 12: Evolución de las ganancias con referencia constante.

El entrenamiento de la ANN permitió lograr la sintonización dinámica del controlador PID, ajustando las ganancias en tiempo real según las condiciones del sistema. Este proceso se llevó a cabo utilizando los experimentos previamente mencionados como conjunto de datos de entrenamiento. Como resultado, se obtuvo un modelo capaz de generar ganancias adaptativas que optimizan el desempeño del controlador PID, garantizando estabilidad y eficiencia incluso ante cambios en la referencia. Los resultados obtenidos durante el entrenamiento se ilustran en la Figura 11, donde se observa la respuesta del sistema frente a una velocidad de referencia de 60 RPM. Por otro lado, la Figura 12 muestra que, ante una referencia constante, las ganancias del PID experimentan ajustes menores alrededor de sus valores iniciales. Esto se debe a que la ANN prioriza la estabilidad del sistema en condiciones estacionarias, realizando modificaciones sutiles para contrarrestar perturbaciones de baja magnitud.

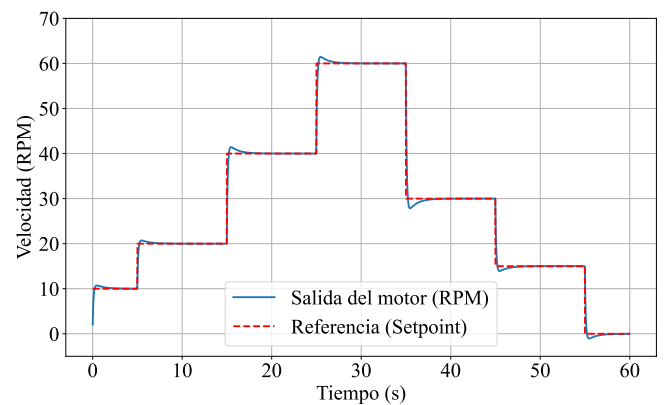


Figura 13: Control de velocidad del RS SRV02 con PID ante diferentes velocidades de referencia con sintonización de ganancias dinámicas.

Al repetir la prueba con una referencia que varía en intervalos definidos, los resultados obtenidos se presentan en la Figura 13, donde se muestra la respuesta del sistema frente a una referencia de velocidad dinámica que alcanza un máximo de 60 RPM. Asimismo, la Figura 14 ilustra la evolución dinámica de

las ganancias del controlador PID, evidenciando cómo se ajustan en tiempo real para adaptarse a la respuesta del sistema.

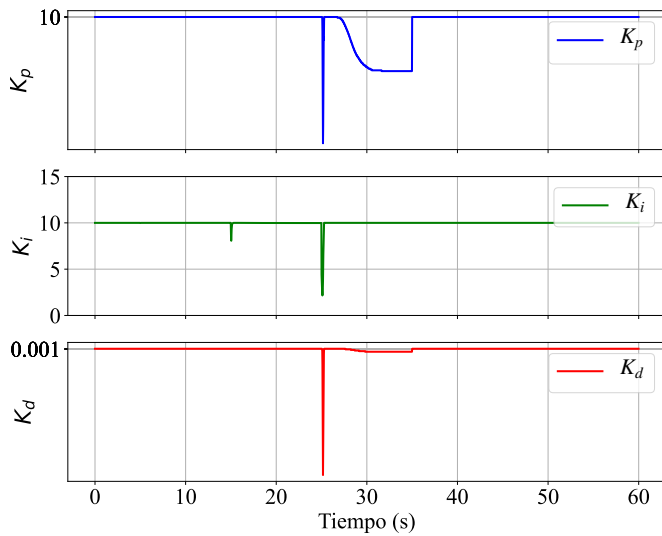


Figura 14: Evolución de las ganancias con diferentes velocidades de referencia.

## 5. Resultados experimentales

Para validar los enfoques previamente mencionados, se llevó a cabo la sintonización del controlador PID y su posterior implementación en el sistema RS SRV02.

### 5.1. Experimentación con sintonización previa

En la Figura 15, se presenta el comportamiento del sistema frente a un setpoint constante de 60 RPM, replicando las condiciones establecidas en la simulación utilizada como referencia. Esto permite evaluar la correspondencia entre los resultados experimentales y los simulados, validando la efectividad de la sintonización realizada previamente en el entorno de simulación. Para efectos de comparación, las ganancias del controlador PID permanecen constantes con los valores  $K_p = 8.61$ ,  $K_i = 8.52$ , y  $K_d = 0.01$ .

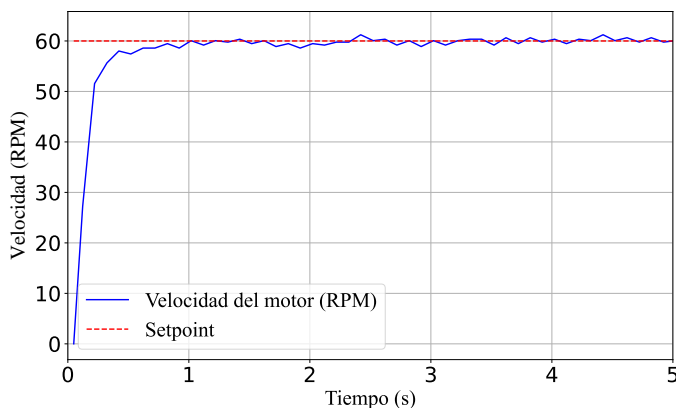


Figura 15: Respuesta el RS SRV02 con control PID ante una entrada constante con sintonización de ganancias previa.

De manera similar, se llevó a cabo una prueba utilizando un setpoint variable, donde la velocidad de referencia cambió en

intervalos de tiempo predefinidos con los siguientes valores: 10 RPM de 0 a 5 segundos, 20 RPM de 5 a 15 segundos, 40 RPM de 15 a 25 segundos, 60 RPM de 25 a 35 segundos, 30 RPM de 35 a 45 segundos, 15 RPM de 45 a 55 segundos y 0 RPM después de 55 segundos. Los resultados obtenidos se presentan en la Figura 16, mostrando la respuesta del sistema frente a este perfil dinámico de referencia.

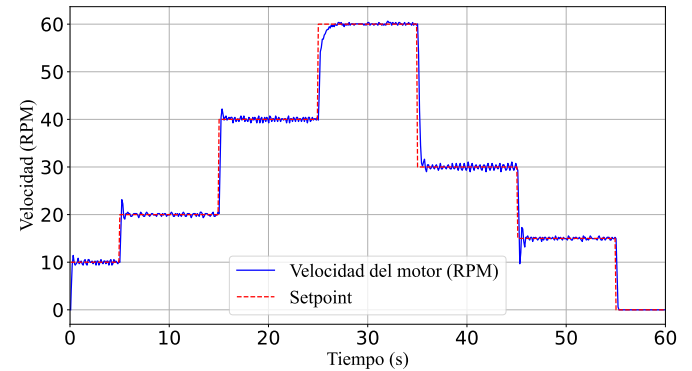


Figura 16: Respuesta el RS SRV02 con control PID ante diferentes velocidades de referencia con sintonización de ganancias previa.

### 5.2. Experimentación con sintonización dinámica

A través de la sintonización dinámica implementada mediante la ANN, se realizó inicialmente una prueba utilizando una referencia constante de 60 RPM. Los resultados obtenidos se presentan en la Figura 17, donde se ilustra la respuesta del sistema ante dicha referencia. Por otro lado, en la Figura 18 se muestra la evolución de las ganancias del controlador PID en función de la respuesta del sistema RS SRV02, evidenciando el ajuste continuo de las mismas frente a la referencia establecida.

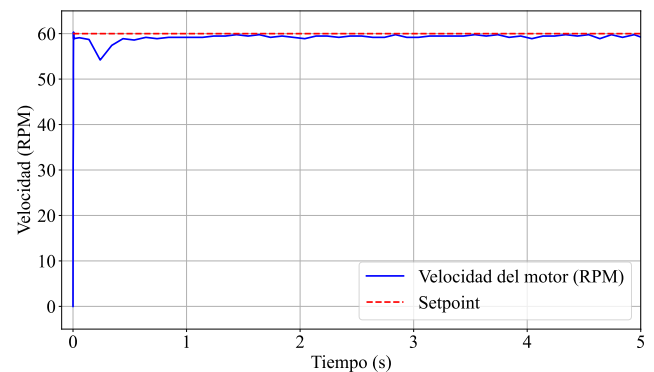


Figura 17: Respuesta el RS SRV02 con control PID ante velocidad constante de referencia con sintonización dinámica.

Para los experimentos con una referencia que varía en distintos intervalos de tiempo, se empleó la misma referencia utilizada previamente en los experimentos y simulaciones anteriores, garantizando una base de comparación consistente. En la Figura 19 se presenta el comportamiento del sistema RS SRV02 frente a los cambios de velocidad, destacando cómo la ANN se adapta progresivamente a estas variaciones, mejorando su desempeño con el tiempo.



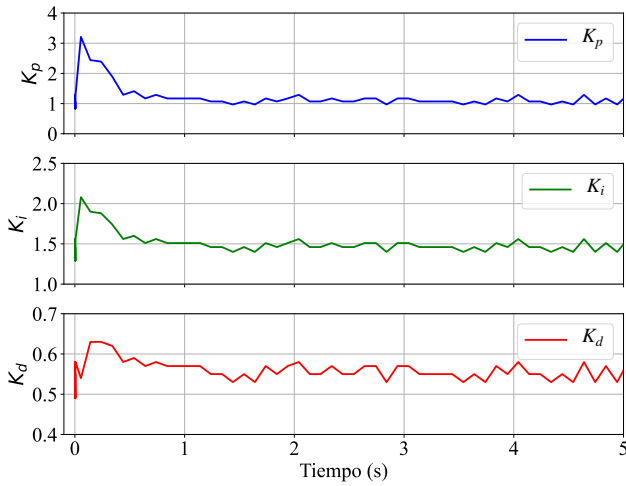


Figura 18: Evolución del valor de las ganancias del PID ante velocidad constante de referencia con sintonización dinámica.

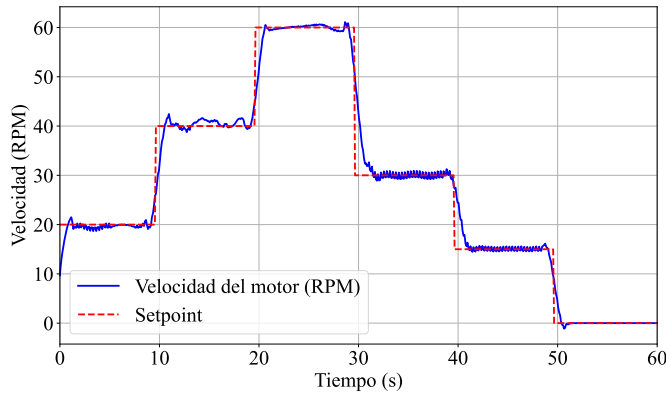


Figura 19: Respuesta del RS SRV02 con control PID ante diferentes velocidades de referencia con sintonización dinámica.

Por otro lado, en la Figura 20, se observa la evolución dinámica de las ganancias del controlador PID en respuesta a las modificaciones de la velocidad de referencia.

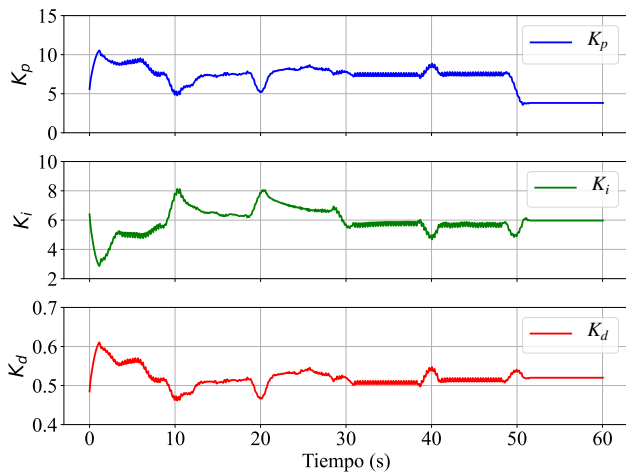


Figura 20: Evolución de las ganancias del PID ante diferentes velocidades de referencia con sintonización dinámica.

### 5.3. Comparación con algoritmos genéticos

Los AG representan otra técnica de optimización avanzada utilizada para la sintonización de controladores PID, especialmente eficaz en sistemas con múltiples óptimos locales y comportamientos no lineales. A diferencia de las ANN, que aprenden patrones a partir de datos y realizan predicciones en tiempo real, los AG utilizan operadores de selección, cruce y mutación para encontrar soluciones óptimas. Para la sintonización de PID, los AG minimizan una función de costo explorando el espacio de búsqueda de ganancias  $K_p$ ,  $K_i$ ,  $K_d$ .

No obstante, los AG requieren un mayor número de evaluaciones del sistema y son computacionalmente más costosos durante la fase de optimización, ya que cada evaluación implica simular el comportamiento completo del sistema. En contraste, una vez entrenadas, las ANN pueden generalizar soluciones rápidamente, permitiendo ajustes en tiempo real sin necesidad de nuevas optimizaciones. Por lo tanto, aunque ambas técnicas presentan ventajas significativas sobre métodos clásicos, las ANN resultan más adecuadas para aplicaciones en tiempo real donde la velocidad de ajuste es crítica, mientras que los AG destacan en escenarios donde la precisión extrema y el tiempo de cómputo no son restricciones severas. A continuación se exponen únicamente los resultados experimentales del sistema RS SRV02 sintonizado mediante AG, bajo distintas condiciones de referencia. Los resultados de simulación se omiten, dado que el objetivo principal del estudio no es el análisis específico de los AG, sino su comparación con otras técnicas de sintonización PID.

#### 5.3.1. AG para sintonización previa

Para garantizar una comparación justa entre los métodos de sintonización PID basados en ANN y AG, se definió que ambas técnicas optimizan el MSE entre las ganancias actuales como se ve en la Ecuación (11) junto con los parámetros mostrados en la Tabla 2.

Tabla 2: Parámetros del AG para sintonización PID

Parámetro	Valor
Tamaño de la población	30 individuos
Número de generaciones	60
Rangos de búsqueda	$K_p, K_i \in [0, 10]$ , $K_d \in [0, 1]$

El algoritmo se diseñó con una estrategia de elitismo que conserva los dos mejores individuos por generación. La probabilidad de cruce fue del 80 %, mientras que la tasa de mutación fue del 20 %. Cada individuo de la población representa un vector de ganancias PID y se evalúa simulando la respuesta del sistema con dichas ganancias. A partir de operadores de selección por torneo, cruce lineal con ponderación aleatoria y mutación aditiva gaussiana, el algoritmo evoluciona hacia una solución óptima. El resultado es una configuración de ganancias que minimiza el MSE y genera una respuesta rápida y estable del sistema, validando la viabilidad del uso de algoritmos evolutivos como alternativa efectiva a métodos convencionales o de aprendizaje supervisado como las redes neuronales artificiales.

La mejor solución encontrada por el AG con una MAE mínimo son los siguientes:

$$K_p = 3.75, \quad K_i = 9.89, \quad K_d = 0.011$$

La Figura 21 muestra los resultados de seguimiento control de seguimiento de una trayectoria aplicando al sistema RS SRV02.

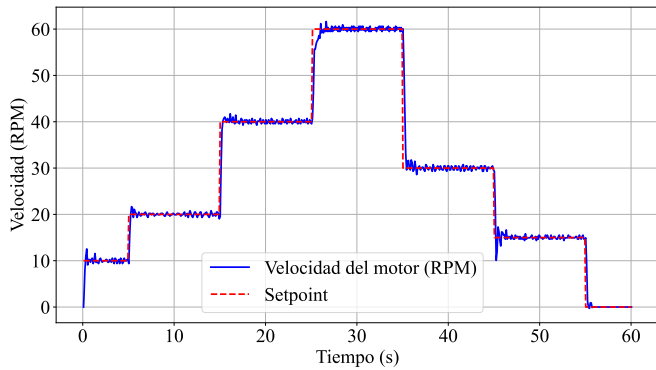


Figura 21: Control PID con sintonización previa mediante AG

### 5.3.2. AG adaptativo para sintonización dinámica

Aunque los algoritmos genéticos tradicionales operan fuera de línea, hemos desarrollado un enfoque híbrido que permite su aplicación en tiempo real. Este método ejecuta mini-optimizaciones genéticas periódicamente durante la operación del sistema, ajustando las ganancias PID en respuesta a cambios en la referencia o perturbaciones. En este trabajo se implementa un AG adaptativo orientado a la sintonización dinámica de un controlador PID aplicado al control de velocidad de un motor DC, tal que el AG actúa en tiempo real, adaptando las ganancias PID ante variaciones en la referencia o condiciones operativas, permitiendo así una regulación continua del comportamiento del sistema. Tomando en cuenta los mismos parámetros de la Tabla 2 con una tasa de mutación del 30 % y elitismo que conserva siempre el mejor candidato. La evaluación de desempeño se realiza mediante una función de costo basada en la Ecuación (11), calculado sobre una ventana deslizante de los últimos 10 errores, a lo que se le suma una penalización si las nuevas ganancias cambian bruscamente respecto a las anteriores, lo que favorece la estabilidad. Cada segundo, el algoritmo realiza 10 generaciones internas donde explora soluciones cercanas a las ganancias actuales mediante una mutación adaptativa, reduciendo la intensidad de exploración conforme avanza el tiempo. Esta estructura híbrida asegura una búsqueda eficiente y local, manteniendo la estabilidad del sistema sin reiniciar el proceso de control. La Figura 22 muestra los resultados de este enfoque aplicado al sistema RS SRV02.

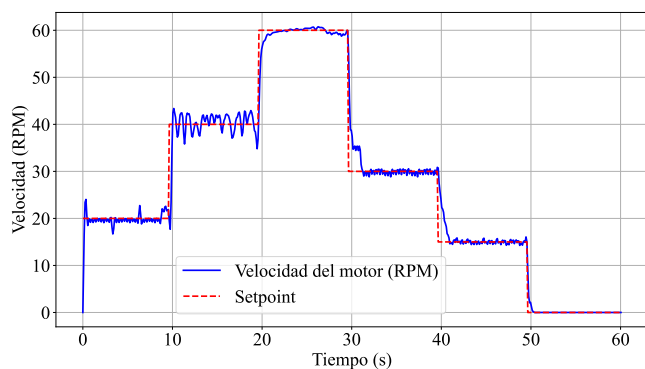


Figura 22: Control PID con sintonización dinámica mediante AG adaptativo

En la Figura 23 se evidencia la capacidad de adaptación dinámica de las ganancias del controlador PID ( $K_p$ ,  $K_i$  y  $K_d$ ) ante variaciones en la referencia de velocidad, demostrando robustez en el desempeño del sistema. Cabe destacar que este enfoque basado en AG adaptativo tiene una limitación inherente la selección manual de ganancias iniciales. Mediante un proceso iterativo de prueba y error, se determinó como configuración óptima inicial los valores  $K_p = 8.66$ ,  $K_i = 4.24$  y  $K_d = 0.56$ . Esta dependencia de parámetros iniciales subóptimos constituye una desventaja significativa del método.

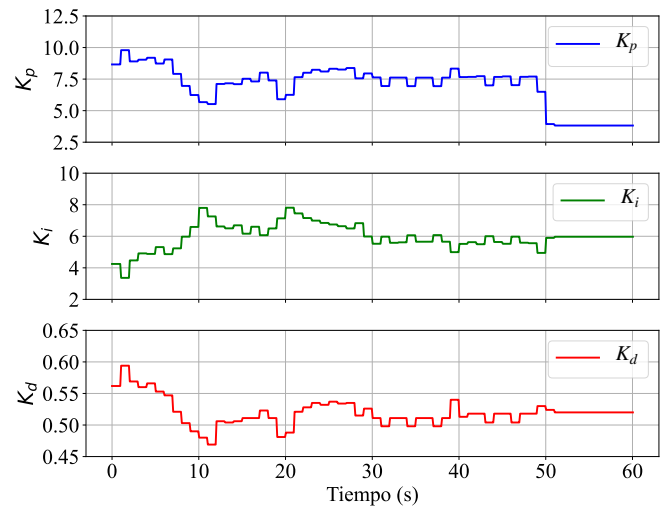


Figura 23: Evolución de las ganancias de PID ante diferentes velocidades de referencia con AG adaptativo.

En la Tabla 3 se presenta una comparación detallada entre los enfoques de sintonización de controladores PID mediante ANN y AG, considerando tanto la sintonización previa como la dinámica. Los datos que sustentan esta comparación provienen de simulaciones y pruebas experimentales realizadas sobre el RS SRV02, en las que se aplicaron ambos métodos de sintonización. Esta tabla permite visualizar las fortalezas y limitaciones de cada técnica, así como su rendimiento en términos cuantitativos, proporcionando una base sólida para la selección del enfoque más adecuado en aplicaciones con condiciones de operación variables.

A diferencia de las ANN, que requieren entrenamiento intensivo y cuya estructura es más difícil de interpretar, los AG proporcionan una solución más transparente, lo que los hace particularmente atractivos para escenarios donde la trazabilidad del ajuste del controlador es prioritaria. No obstante, los costos computacionales asociados a su ejecución periódica son mayores que los de una red neuronal ya entrenada, lo que plantea un compromiso entre capacidad de adaptación, costo y facilidad de implementación ya en el sistema embebido.

## 6. Conclusiones y trabajo futuro

El uso de ANN para la sintonización de controladores PID ha sido demostrado como una herramienta eficaz tanto en escenarios de sintonización previa como dinámica. En la sintonización previa, valores consistentes para las ganancias del controlador fueron proporcionados por las ANN, lo que facilitó el di-

Tabla 3: Comparación entre ANN y Algoritmos Genéticos para la sintonización de controladores PID

Criterio	ANN Previa	ANN Dinámica	GA Previo	GA Dinámico
Tipo de entrenamiento	Offline (simulado)	Online (tiempo real)	Offline (simulación)	Online (optimización local)
Enfoque	Predictivo basado en datos simulados	Adaptación en tiempo real	Optimización evolutiva	Mini-optimizaciones periódicas
Modelo requerido	No	No	Parcial (modelo simple)	Parcial (simplificado)
Dependencia de entrenamiento	Alta	Alta	Nula	Nula
Adaptabilidad	No	Alta	No	Alta
Velocidad de respuesta	Instantánea (post-entrenamiento)	Rápida (inferencia en ms)	Lenta (fase de optimización)	Lenta (10 gen/seg)
Costo computacional	Bajo	Medio	Alto	Alto
MSE (Experimental)	1.65	2.58	1.79	2.71
Implementación física	Sencilla (valores fijos)	Compleja (integración en tiempo real)	Sencilla (valores fijos)	Muy compleja (GA embebido)

seño del sistema y permitió que una correspondencia sólida entre simulaciones y pruebas experimentales fuera mostrada. Por otro lado, el ajuste de las ganancias del controlador en tiempo real fue permitido por la sintonización dinámica, adaptándose a cambios en la velocidad de referencia. Sin embargo, aunque un rendimiento óptimo del sistema no fue alcanzado en la última prueba y un desempeño inferior fue observado en comparación con la sintonización previa, una mejora gradual en el desempeño de la ANN con el tiempo fue detectada.

Estas capacidades han llevado a que las ANN sean consideradas una opción ideal para aplicaciones donde condiciones de operación cambiantes son experimentadas constantemente. La viabilidad de combinar métodos tradicionales con técnicas modernas de aprendizaje automático para optimizar el rendimiento en sistemas de control industrial ha sido validada por la integración de modelos matemáticos con ANN en el sistema RS SRV02.

Aunque el sistema fue representado como una dinámica de primer orden por el modelo matemático identificado, características propias de una dinámica de segundo orden en lazo cerrado fueron mostradas por los resultados experimentales. Esta discrepancia puede ser atribuida a simplificaciones aplicadas durante el modelado, como la omisión de la inductancia y el acoplamiento mecánico. Sin embargo, este modelo fue considerado adecuado para el diseño preliminar del controlador PID y para el entrenamiento inicial de la red neuronal.

Adicionalmente, la comparación con los enfoques basados en algoritmos genéticos permitió evidenciar que estos últimos presentan ventajas relevantes en términos de interpretabilidad y ajuste fino. En su modalidad de sintonización previa, los algoritmos genéticos ofrecieron el menor MSE, mostrando una alta precisión en la etapa de diseño sin necesidad de entrenamiento previo. Por otro lado, la sintonización dinámica basada en AG demostró un buen compromiso entre adaptabilidad y desempeño, logrando un seguimiento robusto frente a cambios en la referencia con ganancias adaptadas en tiempo real.

Como trabajo futuro, se plantea que el modelo sea refinado mediante la inclusión de una descripción dinámica más completa. Además, se propone que la red neuronal sea entrenada con datos reales del sistema RS SRV02 para incrementar la

precisión del modelo, así como que una nueva configuración de capas con la cantidad de neuronas de la red sea creada. Por su parte, una hibridación de ambos enfoques empleando ANN para predecir ajustes iniciales y AG para afinamiento dinámico podría representar una solución aún más robusta y eficiente, integrando lo mejor de ambas metodologías.

No obstante, ha sido demostrado por este estudio que, incluso al ser empleados modelos simplificados y datos generados mediante simulaciones, una sintonización efectiva de controladores PID puede ser lograda por las ANN lo que refuerza la metodología.

## Agradecimientos

Agregue sus agradecimiento hasta la aceptación del manuscrito.

## Referencias

- Abdelghany, M., Elnady, A. O., e Ibrahim, S. O. (2023). Optimum pid controller with fuzzy self-tuning for dc servo motor. *Journal of Robotics and Control (JRC)*, 4(4):500–508.
- Amuthambigaiyin Sundari, K. y Maruthupandi, P. (2022). Optimal design of pid controller for the analysis of two tank system using metaheuristic optimization algorithm. *Journal of Electrical Engineering & Technology*, 17(1):627–640.
- Åström, K. J. y Hägglund, T. (2006). *Advanced PID control*. ISA-The Instrumentation, Systems and Automation Society.
- Borase, R. P., Maghade, D., Sondkar, S., y Pawar, S. (2021). A review of pid control, tuning methods and applications. *International Journal of Dynamics and Control*, 9:818–827.
- Cao, F. (2020). Pid controller optimized by genetic algorithm for direct-drive servo system. *Neural Computing and Applications*, 32(1):23–30.
- Debnath, M. K., Agrawal, R., Tripathy, S. R., y Choudhury, S. (2020). Artificial neural network tuned pid controller for lfc investigation including distributed generation. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 33(5):e2740.
- Elmorshedy, M. F., Xu, W., El-Sousy, F. F., Islam, M. R., y Ahmed, A. A. (2021). Recent achievements in model predictive control techniques for industrial motor: A comprehensive state-of-the-art. *IEEE Access*, 9:58170–58191.
- Franklin, G. F., Powell, J. D., y Emami-Naeini, A. (2010). *Feedback control of dynamic systems*, volumen 10. Pearson Upper Saddle River, NJ.

- Gadekar, K., Joshi, S., y Mehta, H. (2020). Performance improvement in bldc motor drive using self-tuning pid controller. En *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 1162–1166. IEEE.
- Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Inc.
- Gundogdu, A., Celikel, R., y Aydogmus, O. (2021). Comparison of si-ann and extended kalman filter-based sensorless speed controls of a dc motor. *Arabian Journal for Science and Engineering*, 46(2):1241–1256.
- Gurney, K. (2017). *An Introduction to Neural Networks*. CRC Press, London.
- Kommula, B. N. y Kota, V. R. (2022). Design of mfa-pso based fractional order pid controller for effective torque controlled bldc motor. *Sustainable energy technologies and assessments*, 49:101644.
- Kruse, R., Mostaghim, S., Borgelt, C., Braune, C., y Steinbrecher, M. (2022). Multi-layer perceptrons. En *Computational intelligence: a methodological introduction*, pp. 53–124. Springer.
- Lee, J., Chang, P. H., Yu, B., y Jin, M. (2020). An adaptive pid control for robot manipulators under substantial payload variations. *IEEE Access*, 8:162261–162270.
- Li, X. y Gao, L. (2021). Robust fractional-order pid tuning method for a plant with an uncertain parameter. *International Journal of Control, Automation and Systems*, 19:1302–1310.
- Lihua, L. (2022). Simulation physics-informed deep neural network by adaptive adam optimization method to perform a comparative study of the system. *Engineering with Computers*, 38(Suppl 2):1111–1130.
- Parhi, R. y Nowak, R. D. (2020). The role of neural network activation functions. *IEEE Signal Processing Letters*, 27:1779–1783.
- Radonjić, M., Zečević, Ž., y Krstajić, B. (2022). An iot system for real-time monitoring of dc motor overload. *Electronics*, 11(10):1555.
- Somefun, O. A., Akingbade, K., y Dahunsi, F. (2021). The dilemma of pid tuning. *Annual Reviews in Control*, 52:65–74.