

# Regulación de velocidad robusta de un motor de CD usando un método basado en datos con RNA

## Robust speed regulation of a DC motor using a data-driven method with RNA

M. A. Castro-Liera <sup>a</sup>, C. Higuera-Verdugo <sup>a,\*</sup>, J. A. Sandoval-Galarza <sup>a</sup>, I. Castro-Liera <sup>a</sup>

<sup>a</sup>Área de División de Estudios de Posgrado e Investigación, Tecnológico Nacional de México/ Instituto Tecnológico de La Paz, 23080, La Paz, B.C.S., México.

### Resumen

En este trabajo se presenta el control basado en datos mediante redes neuronales artificiales de un motor de CD. Los principales componentes del esquema de control propuesto son el modelo numérico de la planta que incluye como entrada el voltaje suministrado por la fuente y un modelo inverso, que calcula la señal de control, implementados mediante Redes Neuronales Artificiales (RNA). La novedad del presente método basado en datos con RNA es la incorporación del voltaje en el vector de estados del modelo, ya que interesa por su importancia práctica al modelar el motor de CD. Para validar el esquema propuesto, se presentan resultados experimentales sobre un motor de CD comercial ante perturbaciones variantes en el tiempo.

**Palabras Clave:** Control basado en datos, Identificación de sistemas, Redes neuronales artificiales.

### Abstract

This work presents a data-driven control of a DC motor using artificial neural networks. The main components of the proposed control scheme are the numerical model of the plant, that includes as input the voltage supplied by the source, and an inverse model which calculates the control signal, both implemented using Artificial Neural Networks (ANN). The main novelty of the proposed method is the incorporation of voltage in the state vector of the model which it is of interest due to its practical importance when modeling the DC motor. To validate the proposed scheme, experimental results on a commercial DC motor under time-varying perturbations are presented.

**Keywords:** Data-driven control, Systems identification, Artificial Neural Network.

## 1. Introducción

En general, los esquemas de control basados en modelos requieren conocer los valores numéricos de los parámetros del modelo matemático del motor. En contraparte, el esquema propuesto en el presente trabajo puede prescindir de ese requerimiento y generar un modelo de la planta a partir de la correspondencia uno a uno entre los valores de las variables de entrada y salida obtenidos de forma experimental en lazo abierto.

Al método de construcción de modelos numéricos que aproximan el comportamiento de un sistema dinámico a partir de datos experimentales también se le conoce como identificación de sistemas basada en datos (Naung *et al.*, 2018; Rongmin *et al.*, 2010). A la aplicación de este tipo de modelos numéricos en sistemas de control se le conoce comúnmente como control basado en datos (*Data-driven control*) (Bazanella *et al.*, 2012).

Una alternativa interesante para la generación de este tipo de modelos numéricos son los perceptrones multi-capas o MLP (*Multi-Layer Perceptron*). Este tipo de redes cuentan con la capacidad de aproximar cualquier función con una precisión arbitraria, siempre que se cumplan algunas condiciones en su arquitectura, como la inclusión de al menos una capa oculta (Hornik *et al.*, 1989), por lo que tienen un amplio rango de aplicaciones en ingeniería (Abougarair, 2023; Jayachandra y Mahesh, 2018; Hamoodi *et al.*, 2019; Cao *et al.*, 2019; Li *et al.*, 2020). Con la evolución de equipos de cómputo con mayor poder de procesamiento a finales de los noventa, las redes neuronales profundas, con más de una capa oculta, se han aplicado con éxito en diferentes problemas de reconocimiento de patrones (Lecun *et al.*, 1998).

Una desventaja del procedimiento numérico para aproximar

\* Autor para correspondencia: cesar.hv@lapaz.tecnm.mx

**Correo electrónico:** mareco.cl@lapaz.tecnm.mx (Marco Antonio Castro-Liera), cesar.hv@lapaz.tecnm.mx (César Higuera-Verdugo), jesus.sg@lapaz.tecnm.mx (Jesús Alberto Sandoval-Galarza), iliana.cl@lapaz.tecnm.mx (Iliana Castro-Liera).

**Historial del manuscrito:** recibido el 30/11/2023, última versión-revisada recibida el 30/11/2023, aceptado el 10/03/2024, publicado el 22/04/2024. **DOI:** <https://doi.org/10.29057/icbi.v12iEspecial2.12165>



el modelo de la planta es la necesidad de realizar experimentos en tiempo real. Este proceso puede demorar dependiendo del equipo del que se disponga, al requerir que los datos abarquen el conjunto más amplio posible de condiciones de operación para capturar correctamente la dinámica del sistema a controlar. El modelo numérico obtenido asegura que la entrada de control no exceda el voltaje nominal del motor. Para este fin, se considera en los experimentos variaciones en el voltaje para mejorar la aproximación del modelo numérico del motor de CD. Una característica importante del esquema propuesto es que es robusto ante errores de medición de los sensores, variaciones en la planta y perturbaciones variantes en el tiempo.

El resto del documento está organizado de la siguiente manera. La sección II presenta la metodología de control basado en datos. En la sección III se describe el proceso de la obtención del modelo numérico de la planta, mientras que la sección IV describe el proceso para el diseño del controlador. En la sección V se presentan los resultados experimentales obtenidos durante la validación del controlador y, finalmente, la sección VI describe las conclusiones y el trabajo a futuro.

## 2. Control basado en datos

Una de las estrategias más simples para el desarrollo de un controlador de una planta no lineal, que cuenta con un modelo numérico obtenido experimentalmente, es el método del control inverso (Verbruggen *et al.*, 1999). En dicha estrategia de control se supone que el modelo de la planta puede aproximarse numéricamente por medio de una función

$$\mathbf{m}(\mathbf{x}(k), u(k)) : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n, \quad (1)$$

tal que una variable de salida sea obtenida con:

$$\mathbf{x}(k+1) = \mathbf{m}(\mathbf{x}(k), u(k)), \quad (2)$$

donde  $\mathbf{x}(k)$  es el vector de estados del sistema de dimensión  $n$ ,  $u(k)$  es la señal de control y  $n$  es la dimensión del vector de estados. Para fines prácticos, el instante  $k$  puede ser establecido como el valor del paso de integración utilizado en los experimentos para el registro de la velocidad y el voltaje del motor de CD.

Para el caso de regulación de velocidad del motor, el método de control basado en datos busca calcular un valor de  $u$  que acerque de manera asintótica la velocidad  $\omega(k)$  del motor a una velocidad constante deseada  $\omega_d$ . Matemáticamente, esto puede expresarse como:

$$\lim_{k \rightarrow \infty} [\omega_d - \omega(k)] = 0, \quad (3)$$

donde  $\omega_d$  es asignada por el usuario. Una vez obtenida la función  $\mathbf{m}(\mathbf{x}(k), u(k))$  definida en (1), es posible calcular la entrada de control  $u(k)$  por medio de una función inversa de (1) dada por

$$m^{-1}(\mathbf{x}(k), \omega_d) : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}, \quad (4)$$

de la forma:

$$u(k) = m^{-1}(\mathbf{x}(k), \omega_d). \quad (5)$$

Cabe señalar que la notación  $m^{-1}(\mathbf{x}(k), \omega_d)$  no se refiere al inverso multiplicativo, sino al hecho de que dicha función permite calcular la entrada de control a partir de la salida deseada de la planta.

Con el fin de que el controlador sea robusto ante perturbaciones y errores del modelo numérico, se define la perturbación de velocidad  $\tilde{\omega}(k)$  como

$$\tilde{\omega}(k) = \omega(k) - \omega_m(k) \quad (6)$$

siendo  $\omega(k)$  la velocidad medida del motor y  $\omega_m(k)$  la velocidad calculada a partir de la función  $\mathbf{m}(\mathbf{x}(k), u(k))$ . Una interpretación de  $\tilde{\omega}(k)$  es el efecto de la perturbación  $\tilde{V}(k)$  en la velocidad del motor.

En la Figura 1 se muestra el diagrama de bloques del sistema de control, donde  $\omega_d$  es la velocidad constante deseada del motor de CD.

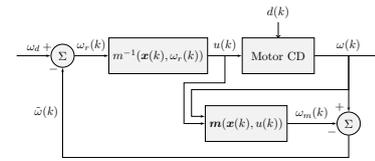


Figura 1: Diagrama de bloques del sistema de control propuesto.

Es conocido que la construcción de  $\mathbf{m}(\mathbf{x}(k), u(k))$  y  $m^{-1}(\mathbf{x}(k), \omega_r(k))$  demanda cálculos complejos, por tanto, para realizar esta tarea en el presente trabajo se utiliza un método de aprendizaje automático (*Machine Learning*) conocido como aprendizaje supervisado basado en redes neuronales. Específicamente se utiliza un perceptrón multi-capas (MLP por sus siglas en inglés). Este método permite aproximar dichas funciones a partir del procesamiento de datos experimentales. Dichos datos se utilizan para ajustar iterativamente los parámetros de un modelo de aprendizaje automático con una estructura preestablecida.

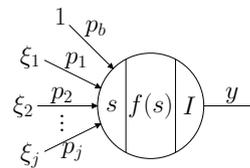


Figura 2: Estructura de un perceptrón.

El elemento básico de construcción de un MLP es el perceptrón, el cual es un modelo matemático inspirado en la neurona biológica, como se muestra en la figura 2. El valor total de la entrada  $s$  al perceptrón se calcula como una suma ponderada de la forma:

$$s = \sum_{i=1}^j p_i \xi_i + p_b, \quad (7)$$

donde  $j$  es el número de entradas a la neurona,  $\xi_i$  es el  $i$ -ésimo valor de entrada y  $p_i$  el peso correspondiente a la  $i$ -ésima conexión, mientras que  $p_b$  es el peso de una entrada fija en 1 conocida como *bias*. A continuación, se aplica una función de activación  $f(s)$  a la entrada, en este trabajo se utilizó la función *ReLU* de la forma:

$$f(s) = \begin{cases} 0, & s < 0 \\ s, & s \geq 0 \end{cases} \quad (8)$$

La función de salida en el perceptrón puede incluir operaciones como el redondeo, la conversión de bipolar a binario o, como en nuestro caso, simplemente la función identidad  $I$ , con la que se obtiene un valor de salida  $y$ . Un solo perceptrón está limitado a generar modelos de tipo lineal. Para resolver problemas no lineales es necesario interconectar un conjunto de perceptrones para formar una red neuronal. Esta debe contener una capa para recibir los valores de entrada al modelo, una o más capas ocultas, que permiten linealizar el problema y una capa proporciona la salida del modelo (ver Figura 3).

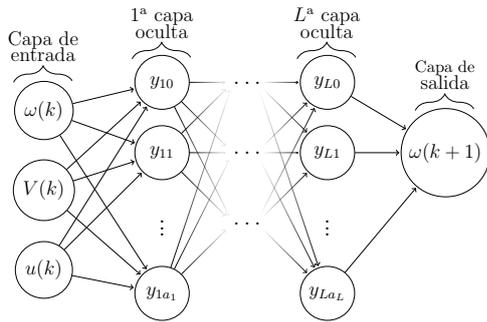


Figura 3: Estructura de un MLP, donde cada nodo representa un perceptrón,  $y_{cb}$  representa la salida del  $b$ -ésimo perceptrón en la  $c$ -ésima capa,  $L$  representa el número de capas ocultas y  $a_d$  el número de neuronas en la  $d$ -ésima capa.

En la Figura 3 se muestra la estructura MLP utilizada en nuestra propuesta. Note que como entrada tenemos la velocidad angular en el instante actual  $\omega(k)$ , el voltaje de entrada en dicho instante  $V(k)$  y la señal de control  $u(k)$  y como salida la velocidad angular en el instante siguiente  $\omega(k+1)$ . Para encontrar los pesos de las conexiones entre los perceptrones que componen la red neuronal, es necesario contar con un conjunto de datos que capturen la dinámica del sistema a modelar. Estos datos deben incluir la salida correspondiente a los valores de las variables de entrada al modelo, a esto se le conoce como *aprendizaje supervisado*. El algoritmo para determinar los pesos de las conexiones, inspirado en el método numérico de optimización conocido como descenso de gradiente, se conoce como *backpropagation* (retropropagación) Rumelhart et al. (1986); Haykin (1999).

La Figura 4 ilustra los principales pasos del método basado en datos para la construcción del modelo y la siguiente sección la describe en detalle. La Figura 5 muestra los principales pasos del proceso que se siguió en este trabajo para el diseño del controlador y en la sección IV describe dicho proceso en detalle.

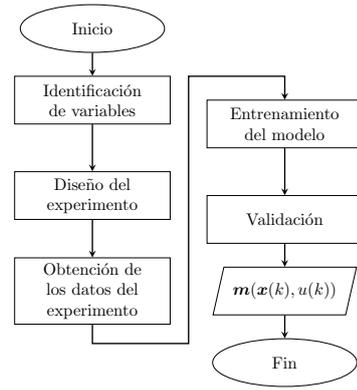


Figura 4: Diagrama de procesos para la identificación del sistema a controlar.

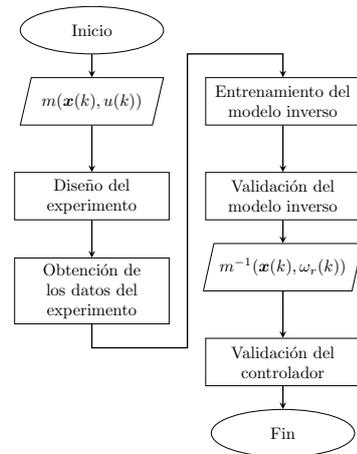


Figura 5: Diagrama de flujo del método basado en datos para el diseño del controlador  $u(k)$ .

### 3. Identificación del sistema

#### 3.1. Planta

Como caso de estudio, en este trabajo se aborda la regulación de la velocidad de un motor de CD.

Los componentes utilizados como plataforma experimental fueron:

1. Fuente de voltaje de CD marca BK precision modelo 1672.
2. Controlador de motor TB6612FNG.
3. Raspberry Pi 4. Requerida para almacenar los datos del experimento y enviar la señal de control al motor.
4. Sensor ACS 712. Para medir el voltaje de entrada al motor.
5. Tarjeta Arduino Uno. Utilizada para monitorear el estado del *encoder* y del sensor de voltaje.
6. Motor CD GA25-371.

En la Figura 6, se muestra el flujo de las señales de control y de potencia mediante la interconexión de los principales componentes utilizados para los experimentos en lazo abierto con el motor. Cabe señalar que se decidió utilizar una tarjeta Arduino dedicada al proceso de muestreo de las señales de los sensores con el fin de obtener una cadencia de muestreo que minimice la pérdida de información.

### 3.2. Identificación de variables

Para el diseño del regulador de velocidad basado en datos, el vector de estados  $x(k)$  está compuesto por  $\omega(k)$  que representa la velocidad angular del motor en un instante  $k$  medida en RPM y  $V(k)$  que representa el voltaje suministrado al sistema en el mismo instante  $k$ . La salida del modelo es la velocidad angular del motor en el instante siguiente,  $\omega(k + 1)$  y la señal de control  $u(k)$  corresponde a un valor del PWM en el rango  $[0,1000]$  milisegundos, proporcional a un ciclo de servicio por cada segundo en el intervalo de 0 a 100 por ciento.

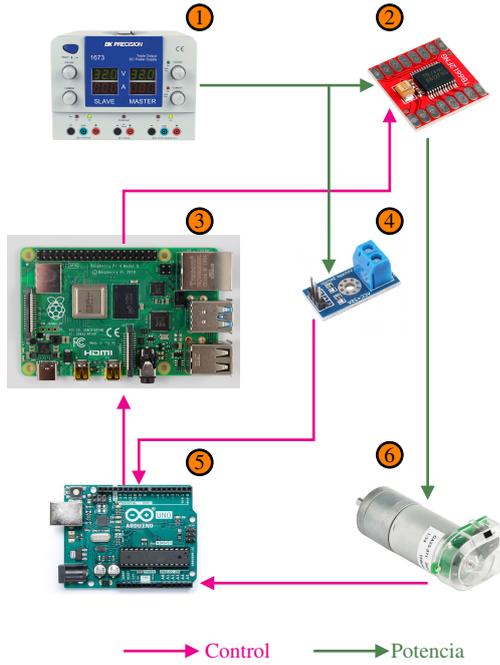


Figura 6: Diagrama del sistema experimental.

Nuestro interés es aproximar una función  $m(x(k), u(k))$ , que cumpla la relación matemática:

$$\omega(k + 1) = m(\omega(k), V(k), u(k)). \quad (9)$$

En adición, buscamos también aproximar una función, referida como el modelo inverso de la planta, de la forma:

$$u(k) = m^{-1}(V(k), \omega_d), \quad (10)$$

que logre el cumplimiento del objetivo de control (3).

### 3.3. Diseño del experimento

Los experimentos se efectuaron en un motor de CD usando tres diferentes tipos de señales de control en lazo abierto:

1.- Función escalón. De los 150 a los 1000 [ms], usando incrementos de 50 [ms] con duración de 2 [s], como se muestra en la Figura 7.

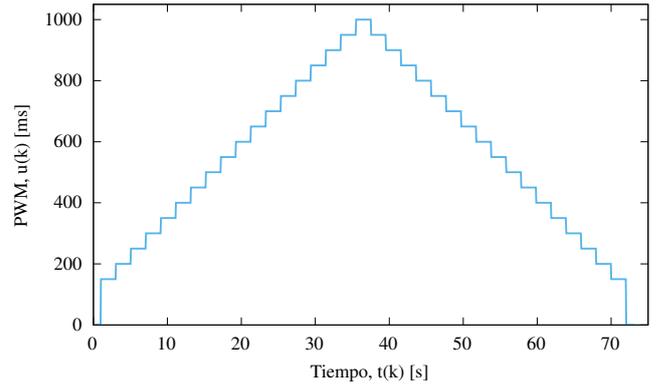


Figura 7: Señal de control del tipo escalón.

2.- Función tangente hiperbólica ascendente. De la forma

$$u(k) = u_f \tanh\left(\pi c \frac{t(k) - t_i}{t_d}\right), \quad (11)$$

con  $c = 1.5$ ,  $u_f = 800$  [ms],  $t_i = 1$  [s],  $t_d = 10$  [s] y descendente de la forma

$$u(k) = u_i \left[ 1 - \tanh\left(\pi c \frac{t(k) - t_i}{t_d}\right) \right], \quad (12)$$

con  $c = 1.1$ ,  $u_i = 800$ ,  $t_i = 1$  [s],  $t_d = 10$  [s] y una meseta de 2 [s], como se muestra en la Figura 8.

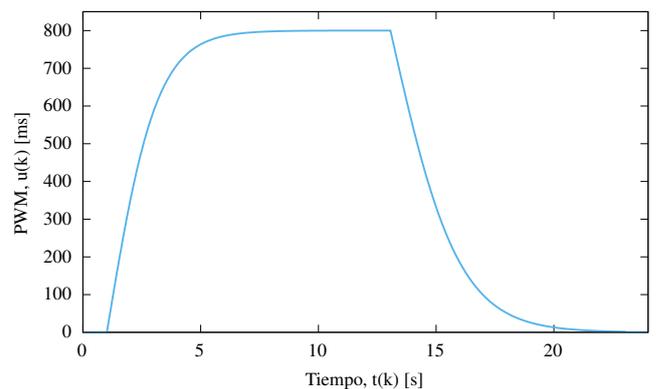


Figura 8: Señal de control basada en la función tanh ascendente (de 1 a 11 [s]), constante (de 11 a 13 [s]) y tanh descendente (de 13 a 23 [s]).

3.- Función rampa positiva y negativa. De 160 a 800 [ms] en 10 segundos con una meseta de 2 [s], como se muestra en la Figura 9.

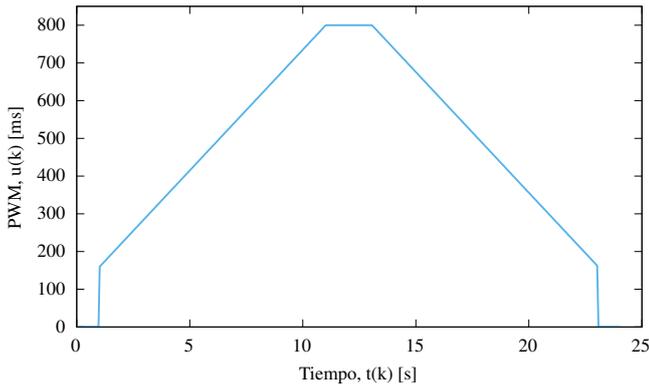


Figura 9: Señal de control basada en funciones tipo rampa ascendente de los 1 a los 11 [s], constante de los 11 a los 13 [s] y descendente de los 13 a los 23 [s].

### 3.4. Registro de los datos experimentales

Se elaboró un programa en lenguaje C capaz de leer un archivo de texto con los parámetros de la señal de control utilizada en el experimento. En cada experimento se generó un archivo de texto con marcas de tiempo con una resolución en microsegundos. Dicho archivo guarda el voltaje medido por el sensor, la señal PWM enviada al controlador electrónico del motor y la velocidad angular del motor en RPM.

Se repitieron los experimentos en un rango de 2 a 7 [V] con incrementos de 0.5. Una vez recolectados los datos experimentales, se procedió a homologar los tiempos de muestreo en intervalos de 0.05 [s], que permiten calcular:

$$t(k+1) = t(k) + 0.05. \quad (13)$$

La homologación consistió en calcular los valores mediante interpolación lineal de los instantes más cercanos disponibles a la muestra requerida. Los valores de las variables de entrada al modelo fueron normalizados mediante el método de máximo y mínimo valor, dado por la expresión:

$$z = \frac{x - x_{min}}{x_{max} - x_{min}}, \quad (14)$$

donde  $z$  es el valor normalizado en el rango  $[0,1]$ ,  $x$  el valor observado,  $x_{min}$  y  $x_{max}$  son los valores mínimo y máximo de cada conjunto de valores a normalizar.

### 3.5. Entrenamiento del modelo de la planta

Para el desarrollo del modelo y el modelo inverso del motor, se utilizaron las bibliotecas de programación *Keras* de *TensorFlow* en *Python 3.10*, con *SciKitLearn* y *talos*.

La estructura del MLP que modela la planta cuenta con 3 neuronas en la capa de entrada ( $V(k)$ ,  $\omega(k)$ ,  $u(k)$ ), 2 capas ocultas con 250 y 40 neuronas respectivamente, las cuales utilizan la función de activación *ReLU* y, finalmente, una neurona en la capa de salida con activación lineal.

Se entrenó <sup>1</sup>el modelo usando el optimizador *Adam*, con un máximo de 1000 épocas <sup>2</sup> de entrenamiento y una estrategia de parada temprana con una paciencia de 10 épocas. Los datos de entrenamiento fueron los obtenidos con la señal tipo escalón. Para evitar el sobre-entrenamiento se evaluó la función de pérdida contra los datos obtenidos con la señal de control tipo tangente hiperbólica. La función de pérdida utilizada fue el error cuadrático medio o MSE (*Mean Squared Error*).

Los datos obtenidos mediante señales tipo rampa se reservaron para una posterior validación del desempeño del modelo con datos no usados durante el entrenamiento.

Se efectuaron 100 ciclos de entrenamiento eligiendo los parámetros del modelo que minimizaron el MSE.

### 3.6. Validación del modelo de la planta

Para evaluar el desempeño del modelo se utilizaron los criterios de índice de correlación lineal  $R^2$  entre las predicciones del modelo y los datos experimentales, así como el error absoluto promedio o MAE (*Mean Absolute Error*). Los resultados con cada conjunto de datos se presentan en la Tabla 1 en la que podemos observar bajos niveles de error absoluto promedio y una alta correlación entre las predicciones y los valores esperados. La Figura 10 muestra la comparación entre el comportamiento medido de la planta y la predicción del modelo. Cabe señalar que la máxima magnitud de error obtenida, denominada ME (*Maximum absolute Error*), fue de 5.086 [RPM] con datos que no fueron utilizados durante el proceso de entrenamiento.

Tabla 1: Evaluación del modelo de la planta

Conjunto de datos	MAE	$R^2$
Entrenamiento	0.511	0.9998
Prueba	0.536	0.9996
Validación	0.691	0.9995

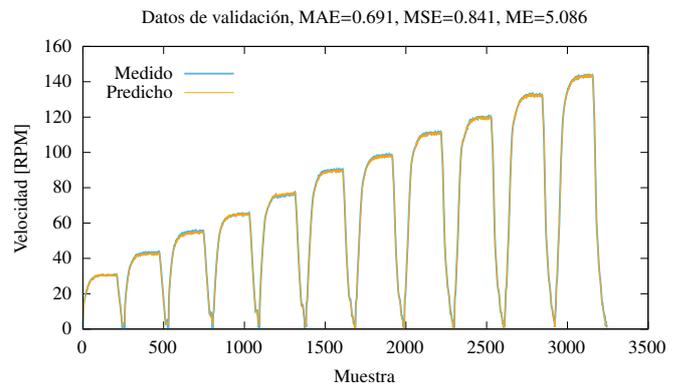


Figura 10: Comparación de datos de validación versus datos de predicción del modelo.

<sup>1</sup>En el contexto del aprendizaje automático, el término entrenamiento se refiere al ajuste de los parámetros del modelo para buscar que su salida sea lo más cercana posible al conjunto de datos experimentales.

<sup>2</sup>Una época se refiere a una iteración completa sobre el conjunto de datos de entrenamiento.

## 4. Diseño del controlador

### 4.1. Modelo inverso de la planta

El modelo inverso de la planta recibe como variables de entrada el nivel de voltaje  $V(k)$ , medido con el sensor ACS 712, y la referencia ajustada  $\omega_r(k)$  expresada en RPM (ver Figura 1). Finalmente en su salida se genera el valor de la señal de control  $u(k)$  correspondiente.

Para recolectar los datos de entrenamiento se realizaron simulaciones con el fin de determinar la velocidad de giro a la que el motor se estabiliza con una serie de rampas de señales PWM operando desde los 3 hasta los 7 [V] con incrementos de 0.5.

La estructura del MLP utilizada cuenta con 2 neuronas en la capa de entrada, 4 capas ocultas de 400, 200, 100 y 50 neuronas, con función de activación *ReLU* y finalmente, una neurona en la capa de salida con activación lineal. Al igual que en el caso mostrado en la subsección *III-E*, se entrenó el modelo inverso usando el optimizador *Adam*, con un máximo de 1000 épocas de entrenamiento y una estrategia de parada temprana con una paciencia de 10 épocas. La mitad de los datos se usaron para el entrenamiento y el resto para evaluar el criterio de parada anticipada.

Tabla 2: Evaluación del modelo inverso de la planta

Conjunto de datos	MAE	$R^2$
Entrenamiento	1.325	0.9999
Prueba	1.322	0.9999

Se efectuaron 100 ciclos de entrenamiento eligiendo los parámetros del modelo que minimizaron el MSE. La Tabla 2 muestra las métricas de desempeño del modelo inverso en la que puede apreciarse un bajo error absoluto promedio y una correlación muy cercana a la lineal dada por un valor de  $R^2$  cercano a uno.

## 5. Resultados experimentales: Desempeño del controlador

Con la finalidad de considerar perturbaciones y validar la robustez del esquema de control propuesto se aplicó una perturbación variante en el tiempo por medio de una rampa de voltaje no considerada en el modelo inverso. Esto se llevó a cabo con una velocidad angular deseada  $\omega_d = 100$  [RPM] a 6 [V], aplicando una perturbación arbitraria de tipo rampa desde -0.5 hasta 0.5 [V], con una duración de 2.5 [s], tal que el voltaje que entra al motor cambia desde los 5.5 hasta 6.5 [V], entre los 2.5 y 5 [s]. Matemáticamente la perturbación de voltaje  $d(k)$  se calcula mediante:

$$d(k) = -0.5 + \frac{t(k) - 2.5}{2.5}. \quad (15)$$

La Figura 11 muestra la comparativa del comportamiento de la planta con y sin el compensador de perturbaciones habilitado. Es notorio que el comportamiento del sistema de control mejora, gracias a que el compensador ajusta la señal de control como se muestra en la Figura 12. Cabe resaltar que la señal

de control se mantiene en todo momento dentro del rango de operación nominal del equipo utilizado.

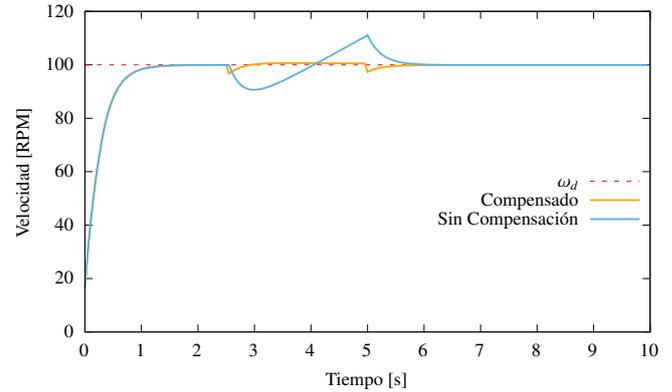


Figura 11: Respuesta del motor de CD ante una perturbación de voltaje tipo rampa con y sin compensación.

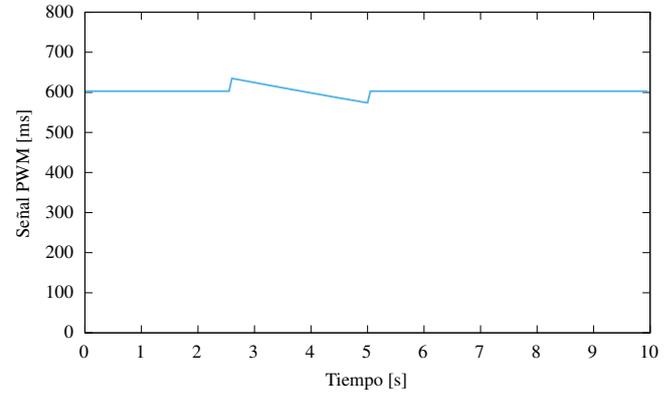


Figura 12: Señal de control de compensación de perturbación de voltaje tipo rampa.

## 6. Conclusiones y trabajo futuro

Se ha aplicado una metodología basada en datos para el diseño de un modelo que aproxima satisfactoriamente la velocidad de un motor de CD. Este se validó al suministrarle datos experimentales no usados durante el entrenamiento y con un rango de voltajes de entrada que puede variar desde los 2 hasta los 7 [V]. Asimismo, se obtuvo un modelo inverso que es capaz de generar señales de control que, dentro de un esquema con realimentación de perturbaciones, es capaz de mejorar la respuesta de la planta ante dichas perturbaciones al cumplir el objetivo de control de regulación de velocidad.

Como trabajo futuro se pretende explorar la aplicación del esquema propuesto en plantas con mayor complejidad y en la sintonía de controladores por moldeo de energía (Sandoval *et al.*, 2022). Asimismo, se pretende implementar un sistema operativo de tiempo real, como *RTLinux*, para conseguir una cadencia de muestreo uniforme y evitar interpolar los datos experimentales al generar el conjunto de datos de entrenamiento.

## Agradecimientos

Este trabajo fue parcialmente financiado por CONAHCYT bajo contrato 166636, por la Red Internacional de Control y Cómputo Aplicado (RICCA) del TecNM y por proyecto TecNM.

## Referencias

- Abougarair, A. J. (2023). Adaptive neural networks based optimal control for stabilizing nonlinear system. En *2023 IEEE 3rd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA)*, pp. 141–148.
- Bazanella, A., Campestrini, L., y Eckhard, D. (2012). *Data-driven Controller Design: The  $H_2$  Approach*. Springer, Netherlands.
- Cao, F., Yang, T., Li, Y., y Tong, S. (2019). Adaptive neural inverse optimal control for a class of strict feedback stochastic nonlinear systems. En *2019 IEEE 8th Data Driven Control and Learning Systems Conference (DDCLS)*, pp. 432–436.
- Hamoodi, S. A., Sheet, I. I., y Mohammed, R. A. (2019). A comparison between pid controller and ann controller for speed control of dc motor. En *2019 2nd International Conference on Electrical, Communication, Computer, Power and Control Engineering (ICECCPCE)*, pp. 221–224.
- Haykin, S. (1999). *Neural networks: a comprehensive foundation* by simon haykin.
- Hornik, K., Stinchcombe, M., y White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.
- Jayachandra, B. y Mahesh, A. (2018). Ann based direct power control of 2-level pwm rectifier. En *2018 International Conference on Power Energy, Environment and Intelligent Control (PEEIC)*, pp. 623–627.
- Lecun, Y., Bottou, L., Bengio, Y., y Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- Li, J., Fang, Y., Ge, Y., y Wu, Y. (2020). Dnn-based implementation of data-driven iterative learning control for unknown system dynamics. En *2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS)*, pp. 1037–1042.
- Naung, Y., Schagin, A., Oo, H. L., Ye, K. Z., y Khaing, Z. M. (2018). Implementation of data driven control system of dc motor by using system identification process. En *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pp. 1801–1804.
- Rongmin, C., Huixing, Z., y Hou, Z. (2010). The permanent magnet linear motor control based on data-driven control theory. En *2010 Chinese Control and Decision Conference*, pp. 3164–3169.
- Rumelhart, D. E., Hinton, G. E., y Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323.
- Sandoval, J., Kelly, R., Santibáñez, V., y Villalobos-Chin, J. (2022). Energy regulation of torque-driven robot manipulators in joint space. *Journal of the Franklin Institute*, 359:1427–1456.
- Verbruggen, H. B., Zimmermann, H.-J., y Babuška, R., editores (1999). *Fuzzy Control Versus Conventional Control*, pp. 59–81. Springer Netherlands, Dordrecht.