

Control por posicionamiento de polos adaptivo usando machine learning para motor DC con inercia variable

Sánchez Aguilar, Luis Ángel

luislasabills@gmail.com

<https://orcid.org/0000-0001-8411-0300>

Universidad Nacional de Trujillo

Trujillo - Perú

RESUMEN

El presente trabajo propone un sistema de control por posicionamiento de polos aplicado a un motor DC con inercia variable, tal que pueda mantener la respuesta de la velocidad, incluso ante algunas variaciones de carga, haciendo uso de algoritmos de Machine Learning. Para ello se obtienen las características de la respuesta de la velocidad del motor sin carga que nos brindará un modelo de referencia para ajustar las ganancias adaptivas. Los datos obtenidos para el entrenamiento fueron los momentos de inercia y ganancias de retroalimentación de estados (velocidad y corriente de armadura) para cargas distintas entre un intervalo definido. Luego, se hizo el entrenamiento de cuatro modelos de Machine Learning: Regresión Polinomial, Árbol de Decisión, Bosque Aleatorio y Máquina de Soporte Vectorial, en base a tres parámetros de validación distintos. Una vez validados, se exportan los modelos que finalmente se comportarán como las ganancias adaptivas a cambios de inercia. Los errores entre el tiempo de establecimiento deseado y simulado fueron los indicadores que nos permitieron concluir si las ganancias adaptivas satisfacen el objetivo. Finalmente, se determinó que el modelo de Árbol de Decisión tuvo mejores prestaciones a la hora de adaptarse al modelo de referencia.

Palabras clave: control por posicionamiento de polos; ganancias adaptivas; inercia variable; machine learning; motor DC.

Correspondencia: luislasabills@gmail.com

Artículo recibido 10 agosto 2022 Aceptado para publicación: 10 septiembre 2022

Conflictos de Interés: Ninguna que declarar

Todo el contenido de **Ciencia Latina Revista Científica Multidisciplinar**, publicados en este sitio están disponibles bajo

Licencia [Creative Commons](https://creativecommons.org/licenses/by/4.0/) 

Cómo citar: Sánchez Aguilar, L. Ángel. (2022). Control por posicionamiento de polos adaptivo usando machine learning para motor DC con inercia variable. *Ciencia Latina Revista Científica Multidisciplinar*, 6(5), 925-943. https://doi.org/10.37811/cl_rcm.v6i5.3152

Adaptive pole-positioning control using machine learning for DC motor with variable inertia

ABSTRACT

The present work proposes a pole placement control system using Machine Learning algorithms applied to a DC motor with variable inertia that can be able to maintain the speed even with some load variations. The response characteristics of the motor speed without load is obtained to provide us a reference model that will serve to adjust the adaptive gains. The data obtained for training includes the moments of inertia and state feedback gains (of velocity and armature current) for different loads within a defined interval. Then, four Machine Learning models were trained, based on three different validation parameters. Once validated, the models that will finally behave as the adaptive gains to inertial changes are exported. The errors between the desired and simulated establishment time were the indicators that allowed us to conclude if the adaptive gains satisfy the objective. Finally, it was determined that the Decision Tree model had better performance when adapting to the reference model.

Keywords: *adaptive gain; DC motor; machine learning; pole placement control; variable inertia.*

INTRODUCCIÓN

Existen distintos tipos de arquitectura para el desarrollo de ingeniería en control de sistemas mecánicos, eléctricos, electromecánicos, entre otros; uno de ellos es el control por posicionamiento de polos (PP), que consiste en controlar mediante la retroalimentación de estados pertenecientes al sistema según su interpretación en espacio de estados. Esta estructura de control depende de las ganancias de los estados para obtener las características deseadas en la salida (Rojas, 2001).

Debido al crecimiento tecnológico y científico, la evolución del control da parte al control adaptivo, que implica el modelado de sistemas linealizados en muchos puntos operativos de interés de aplicación, es decir, este tipo de control permite seguir obteniendo las características deseadas en la salida incluso ante cambios en los parámetros que inicialmente se establecen como constantes para el sistema. Una de las técnicas de este tipo de control es la de retroalimentación adaptiva, que son robustas ante las variaciones paramétricas mencionadas antes (Gómez y Mendoza, 2009).

Los sistemas con variaciones paramétricas son llamados “sistemas variantes en el tiempo” y son más precisos al momento de modelar procesos reales (Ardila, 2018). Siguiendo el enfoque del control por posicionamiento de polos, es clave encontrar las ganancias de los estados retroalimentados, pero el sistema está en constante cambio, por lo que estas ganancias deben adaptarse a los cambios paramétricos.

Existen algoritmos como la función de Ackermann para hallar estas ganancias para sistemas invariantes en el tiempo (Arnáez, 2014), siendo así, para proporcionar un control y un diagnóstico preciso de sistemas variantes en el tiempo, se requiere de un algoritmo diferente que pueda actualizar las ganancias adaptivas que no impliquen los cálculos matemáticos de la función de Ackermann ante un nuevo cambio en los parámetros del sistema.

En este caso, se presenta una solución orientada al Machine Learning, donde exista retroalimentación y actualización de ganancias en tiempo real haciendo uso de inteligencia artificial (Scheinker, 2021) conformando ganancias adaptivas en base a un modelo de referencia del cual aprenda. Se comprueba el uso de este enfoque en un sistema con un solo parámetro característico variable.

El sistema elegido es un motor DC, que en principio posee un momento de inercia constante, que permite una respuesta controlada de velocidad fija (Ann y Shrivastava,

2018), no obstante, al ser la inercia un valor que depende de la carga del motor, es probable encontrarse con cambios en este parámetro. Teniendo esto en cuenta, el análisis del sistema se ajusta a utilizar algoritmos de Machine Learning para resolver el problema de Control.

Ann y Shrivastava (2018) diseñan un controlador PID que sirve como modelo de referencia para la utilización de un controlador mediante una red neuronal artificial para controlar la velocidad de un motor DC. Esta investigación nos indica que la inteligencia artificial no es incompatible con el control de un sistema electromecánico, además de demostrar la necesidad de tener un modelo de referencia que funcione como experiencia de aprendizaje para un modelo de inteligencia artificial.

Scheinker (2021), con su investigación referente a la utilización de un modelo de machine learning como función alternativa para encontrar las ganancias de retroalimentación en un controlador que busca el diagnóstico de vigas en el acelerador de partículas del CERN, muestra el rol de un algoritmo de inteligencia artificial en la arquitectura de control que se seguirá en el presente trabajo al ser el más compatible con el control por posicionamiento de polos.

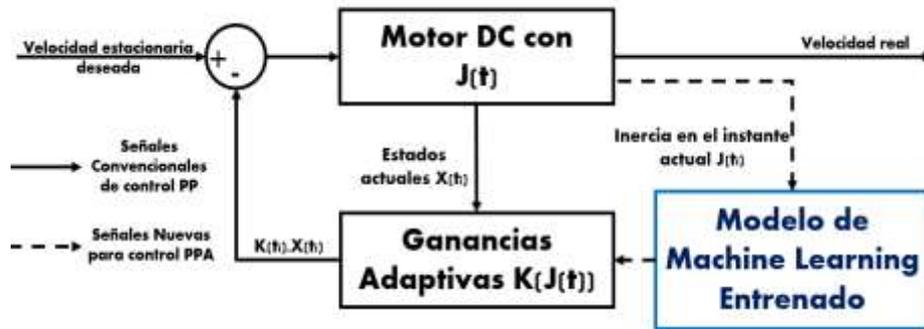
En la presente investigación se proponen cuatro algoritmos de Machine Learning para desarrollar ganancias adaptivas en el control de velocidad por posicionamiento de polos frente al momento de inercia variable de un motor DC. Los algoritmos son: regresión polinomial, árbol de decisión, bosque aleatorio y máquina de soporte vectorial. Finalmente, estas se comparan en base a parámetros de respuesta como lo son el tiempo de establecimiento, el máximo sobrepico, la velocidad en estado estacionario y el error en estado estacionario.

METODOLOGÍA

Para un motor DC simulado se propone una arquitectura donde las ganancias de retroalimentación de sus estados adapten la respuesta de velocidad a distintos momentos de inercia a través del tiempo $J(t)$ para intentar mantener la respuesta con parámetros similares a la del motor sin carga (modelo de referencia con la respuesta de velocidad deseada). La arquitectura propuesta del control por posicionamiento de polos adaptivo (PPA) se muestra en la Figura 1.

Figura 1.

Arquitectura del control por posicionamiento de polos adaptivo PPA

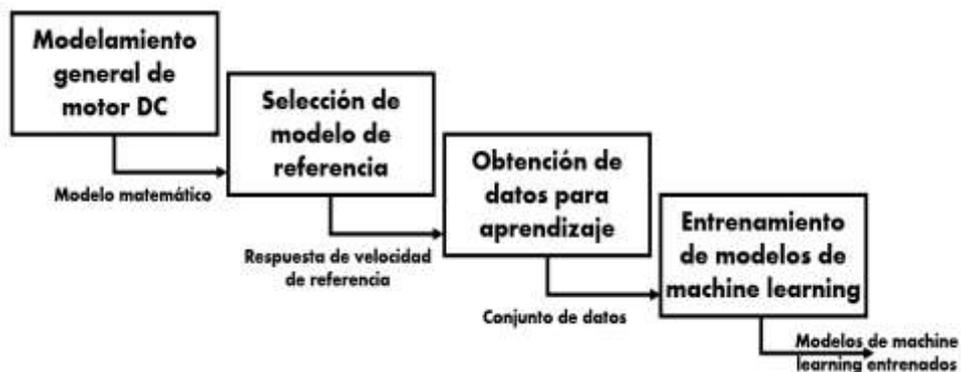


Nota: Los estados son representados como $X(t)$. Las ganancias adaptivas son representadas como $K(J(t))$ que indica su dependencia respecto al momento de inercia $J(t)$. Un instante de tiempo se representa como t_i . **Fuente:** Elaboración propia.

La metodología seguida en el presente trabajo para obtener los cuatro modelos de machine learning entrenados, que permitan conocer las ganancias adecuadas para ciertos momentos de inercia, se muestra en la Figura 2.

Figura 2.

Metodología para obtener los modelos de machine learning entrenados



Fuente: Elaboración propia.

Modelamiento General de Motor DC

El motor DC tiene como entrada al ciclo de trabajo de una señal PWM “ $u(t)$ ”, y la salida sería la velocidad angular del motor “ $y(t)$ ”. Por el tipo de control se requiere de un modelamiento matemático en espacio de estados. Consideramos que el momento inercial depende del tiempo $J(t)$.

$$\dot{x}(t) = \begin{bmatrix} -\frac{R}{L} & -\frac{K_a}{L} \\ \frac{K_m}{J(t)} & -\frac{B}{J(t)} \end{bmatrix} x(t) + \begin{bmatrix} \frac{V_a}{L} \\ 0 \end{bmatrix} u(t) \dots (1)$$

$$y(t) = [0 \quad 1]x(t) + [0]u(t) \dots (2)$$

$$x(t) = \begin{bmatrix} i_a(t) \\ \omega(t) \end{bmatrix} \dots (3)$$

Donde “R” es la resistencia de armadura, “L” es inductancia de armadura, “K_a” es la constante contraelectromotriz, “K_m” es la constante de torque, “B” es constante de fricción, “J” es momento de inercia y “x(t)” es la matriz que contiene los siguientes estados: velocidad angular del motor “ω(t)” y corriente de armadura “i_a(t)”.

Selección del Modelo de Referencia

El modelo con los parámetros específicos que implican cierto comportamiento en la velocidad angular real de salida del motor serán los valores característicos de un motor DC que no tiene carga, por lo que el valor de momento de inercia será fijo (J₀). Los parámetros reales del motor que usaremos son los brindados por Hurtado (2006).

Tabla 1. Parámetros reales de motor DC

Parámetro	Valor	Unidad (SI)
R	2.1975	Ω
L	0.0063	H
K _a	1.78	V/(rad/s)
J ₀	0.0236	kg.m ²
B	0.015	kg.m ² /s

Fuente: Hurtado, 2006.

Se puede aproximar la constante de torque haciendo que su valor sea de igual módulo que la constante contraelectromotriz (Monasterio y Gutiérrez, 2021), por tanto: $K_m = |K_a| \text{ Nm/A} = 1.78 \text{ Nm/A}$.

La velocidad máxima alcanzable por un motor con estos parámetros característicos sin ningún tipo de control es de aproximadamente 1165 rpm (122 rad/s), pero en este caso, se trabajará a una velocidad deseada de 500 rpm (52.46 rad/s). Se simuló el sistema haciendo uso del software Matlab. Se obtuvieron el tiempo de establecimiento (t_s), máximo sobrepico (%MP), velocidad en estado estacionario (v_{ss}) y error en estado estacionario (e_{ss}) de la respuesta de salida del motor sin carga sin controlar (ScSc) y para

el motor sin carga controlado por posicionamiento de polos convencional (ScPP); estos se muestran en la Tabla 2.

Tabla 2. Características de respuesta de ScSc y ScPP

	ScSc	ScPP
tS (s)	0.045	0.0315
%MP (%)	0	0
VSS (rad/s)	52.46	49.32
eSS (%)	N.A.	6

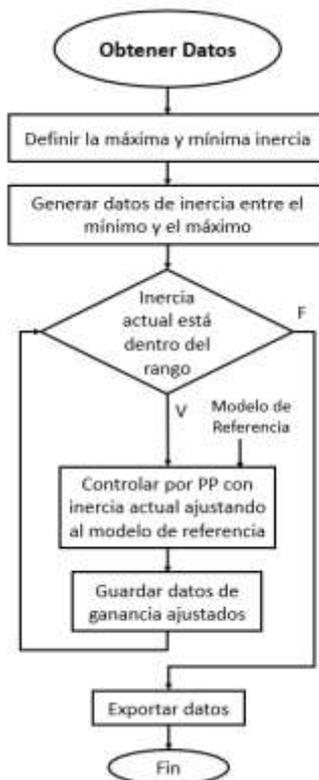
Nota: N.A. significa que no aplica al sistema en cuestión, y esto ocurre porque el error en estado estacionario es un concepto que tiene sentido en un sistema cuando tiene alguna estructura de control. **Fuente:** Elaboración propia.

Las características de respuesta del ScPP (t_s , %MP, v_{ss} , e_{ss}) se escogen como la respuesta de velocidad que debe alcanzar el motor DC cuando la inercia varía respecto al momento de inercia del motor sin carga, obteniendo así el modelo de referencia.

Obtención de Datos para Aprendizaje Automático

El diagrama de flujo del algoritmo diseñado para obtener los datos necesarios se muestra en la Figura 3.

Figura 3. Diagrama de flujo para obtención de datos de inercia y ganancias ajustadas

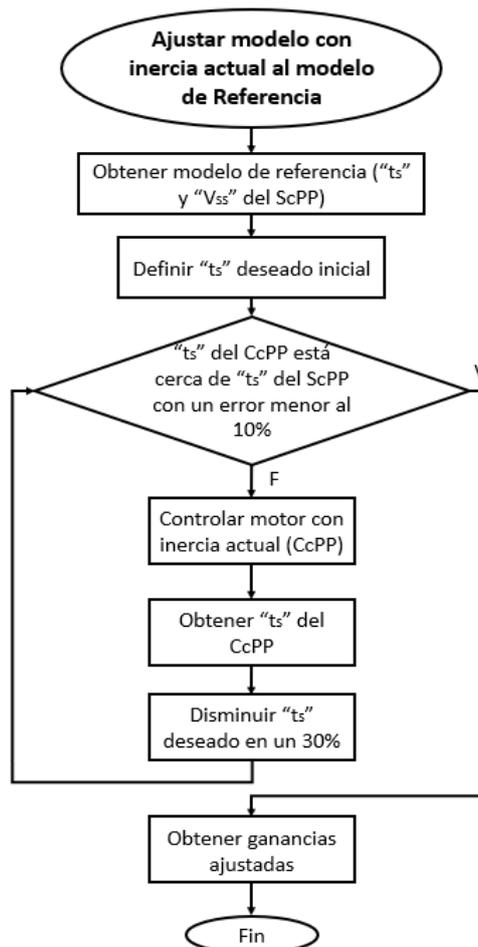


Fuente: Elaboración propia.

El objetivo de los algoritmos de Machine Learning es encontrar una relación entre la inercia " $J(t)$ " y las ganancias " $K_{\omega}(t)$ " y " $K_i(t)$ " que corresponden a la velocidad y a la corriente de armadura, respectivamente. Para la generación de los datos de inercia, determinamos los límites del mismo. Siguiendo recomendaciones empíricas (CONTAVAL, 2018), el intervalo de inercias estará entre $1.005J_0$ y $5J_0$.

Se simula el motor DC para cada dato de inercia y se ajustan las ganancias de retroalimentación de estados haciendo uso de la fórmula de Ackermann para obtener un tiempo de establecimiento cercano al del ScPP. De esta manera, se obtienen los datos de ganancias ajustadas para cada valor de momento de inercia. Finalmente, se logran 2000 muestras que serán unidas en un dataset o conjunto de datos. Esto se utilizará como experiencia de aprendizaje para los algoritmos de machine learning. El ajuste de las ganancias de retroalimentación se realiza según el diagrama de flujo de la Figura 4.

Figura 4. Diagrama de flujo para ajustar motor con inercia actual al modelo de referencia



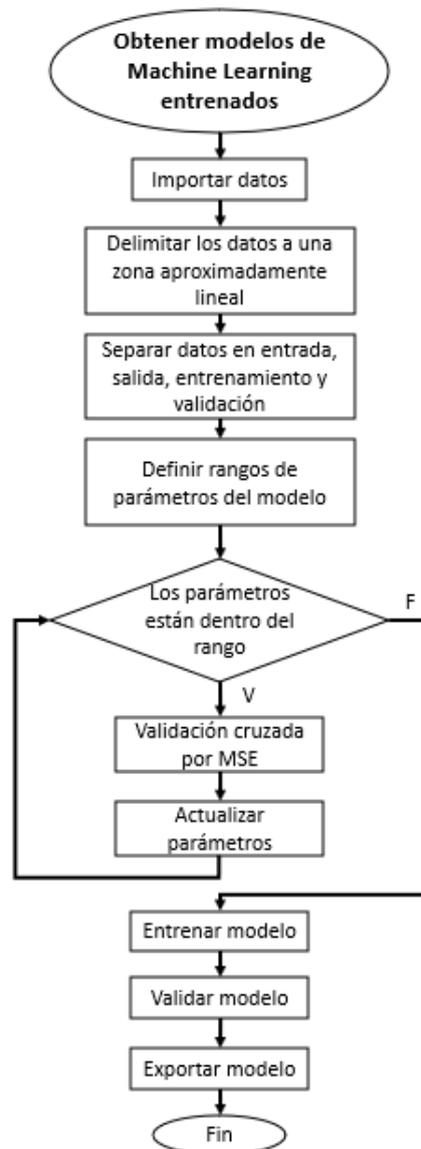
Nota: CcPP se refiere al motor DC con inercia actual (referente al bucle) controlado por posicionamiento de polos. **Fuente:** Elaboración propia.

Al final del proceso se obtienen las ganancias de velocidad y de corriente (variables de estado) que serán las variables objetivo o salidas del modelo de Machine Learning.

Entrenamiento de Modelos de Machine Learning

Se utiliza el lenguaje Python en el IDE de Google Colab para poder programar algoritmos de machine learning. La metodología usada para poder obtener los cuatro tipos de modelos de Machine Learning se puede observar en la Figura 5. El procedimiento es general para todos los modelos, y varían en “parámetros del modelo”.

Figura 5. Diagrama de flujo para obtener modelos de machine learning entrenados

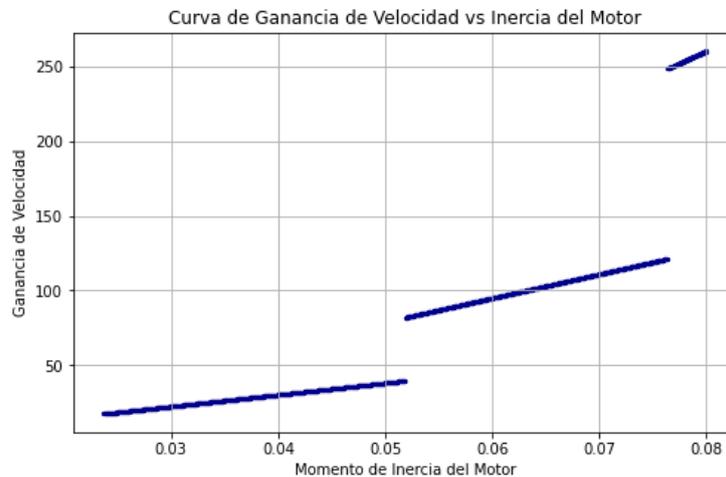


Fuente: Elaboración propia.

Es necesario delimitar los datos para trabajar en zonas aproximadamente lineales o de crecimiento moderado. Para este caso es notorio un crecimiento rápido a partir del valor

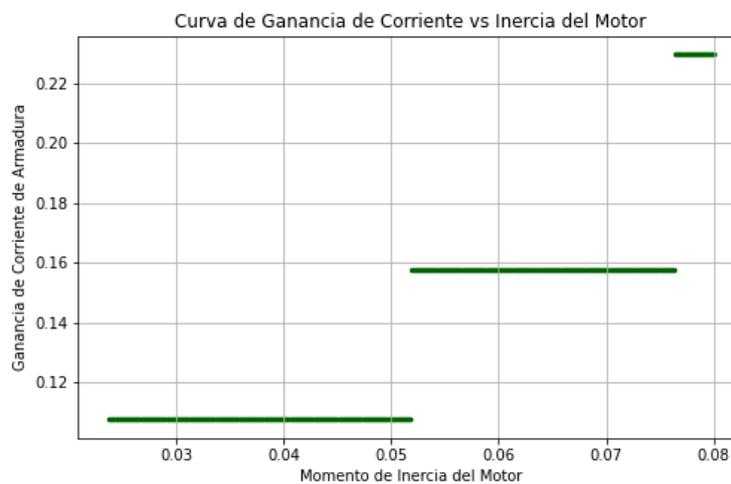
de 0.08 kg.m² aproximadamente, por lo que delimitamos los datos de nuestro modelo hasta este valor, obteniendo las gráficas mostradas en las Figuras 6 y 7.

Figura 6. Curva de Ganancia de velocidad vs Inercia (hasta 0.08 kg.m²)



Fuente: Elaboración propia.

Figura 7. Curva de Ganancia de corriente vs Inercia (hasta 0.08 kg.m²)



Fuente: Elaboración propia.

Posterior a la delimitación de la zona lineal, debemos separar los datos en conjuntos de entrenamiento y validación (75% y 25% respectivamente), además de definir como entrada o variable predictora al momento de inercia, y como salida o variable objetivo a las ganancias. Desde este punto, el procedimiento se bifurca, con el fin de obtener el modelo de machine learning de la ganancia de velocidad y corriente totalmente independientes. Haremos uso del método de “validación cruzada” con el fin minimizar el error cuadrático medio (MSE). Esto se realiza para los 4 algoritmos:

- En el algoritmo de Regresión Polinomial (PR) hay un parámetro que puede mejorar el modelo minimizando el MSE: el grado del polinomio (d).
- En el algoritmo de Árbol de Decisión (TD) hay 3 parámetros a tomar en cuenta: la profundidad (deep), el número mínimo de datos para dividir en una bifurcación de decisión (split) y el número mínimo de datos que debe haber en una hoja o nodo final (leaf).
- En el algoritmo de Bosque Aleatorio (RF) el parámetro a tomar en cuenta es el número de árboles aleatorios (trees).
- En el algoritmo de Máquina de Soporte Vectorial (SVM) hay 3 parámetros a tomar en cuenta: el kernel, el parámetro “ C ” y los grados del polinomio (dS).

Los valores que minimizan el MSE se muestran en el apartado de discusión de resultados en la Tabla 3. Con los parámetros hallados se procede a entrenar los cuatros modelos de Machine Learning y luego se hallan tres parámetros de validación por cada uno: el error estándar residual (RSE), el error cuadrático medio (MSE) y el coeficiente de determinación R^2 , para el caso de PR, TD y SVM, o el Out-of-Bag score para RF (OOB2). Estos también se mostrarán en el apartado de discusión de resultados en la Tabla 4.

Comparación de los Modelos

Para realizar una comparación entre los modelos de machine learning y el control por posicionamiento de polos convencional (PP), se observa el promedio del error entre el tiempo de establecimiento del modelo de referencia (t_s del ScPP mostrado en la Tabla 2) y los tiempos de establecimiento de las respuestas usando los algoritmos para las distintas inercias.

Aprovechando la estructura del procedimiento, también hallamos los promedios de máximo sobrepico (%MPP) y de error en estado estacionario (e_{ssP}), que deberían tender hacia cero. Esto se muestra en la Tabla 5 del siguiente apartado.

RESULTADOS Y DISCUSIÓN

Parámetros de Entrenamiento

Durante el entrenamiento de los cuatros modelos de Machine Learning (ML), se hallaron los parámetros de entrenamiento para minimizar el error cuadrático medio (MSE). Todos estos valores se resumen en la Tabla 3.

Tabla 3. *Parámetros de aprendizaje adecuados para minimizar el MSE*

Ganancia	Parámetro de Entrenamiento		MSE
Velocidad	PR	d: 8	184.8
	TD	deep: 3 split: 40 leaf: 60	52.6
	RF	trees: 10	13.4
	SVM	kernel: "poly" C: 100 dS: 3	4053.6
Corriente	PR	d: 3	2.5e-4
	TD	deep: 2 split: 10 leaf: 2	0.1e-4
	RF	trees: 90	0.06e-4
	SVM	kernel: "poly" C: 10 dS: 2	21.6e-4

Fuente: Elaboración propia.

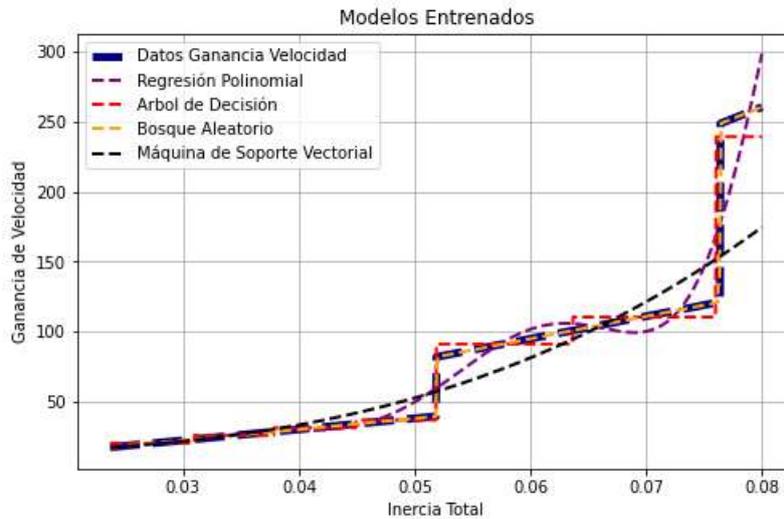
En lo que respecta a la ganancia de velocidad, el modelo que mejor corresponde a los datos es el algoritmo de Bosque Aleatorio RF, con un valor de MSE de 13.42, que se ajusta mejor que los algoritmos de regresión polinomial y de árbol de decisión. Respecto a la ganancia de corriente, también el modelo del Bosque Aleatorio es el que está mucho más cerca a los datos de entrenamiento, al tener el MSE más bajo con un valor de 0.06 e-4.

Tanto en la ganancia de velocidad como de corriente, el algoritmo de Máquina de Soporte Vectorial SVM no responde de buena manera ante los datos, alcanzando MSE muy altos con valores 4053.6 y 21.6e-4 respectivamente.

Estos modelos generan curvas ajustadas a los datos, tanto para la ganancia de velocidad mostrado en la Figura 9, como para la ganancia de corriente mostrado en la Figura 10. A nivel gráfico se puede observar que la curva del modelo de Bosque Aleatorio se ajusta mejor a los datos de entrenamiento en ambas ganancias adaptivas (curva naranja mucho más cercana a la curva azul de la Figura 9 y curva verde de la Figura 10).

Figura 8.

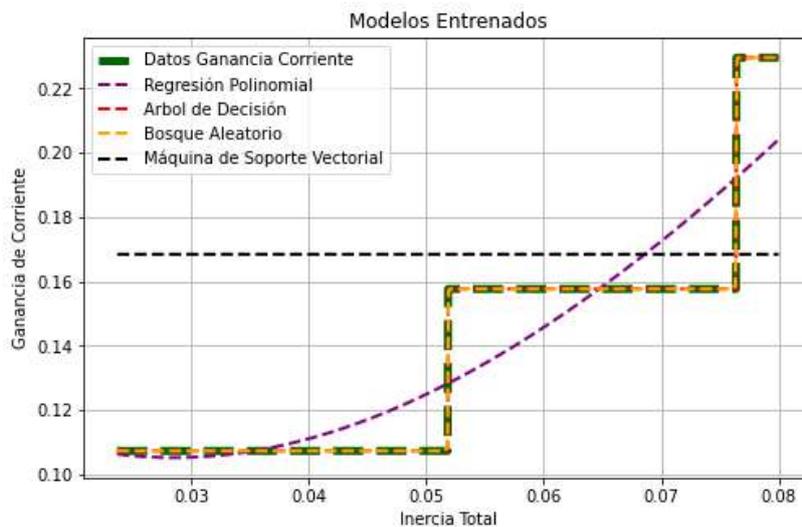
Modelos de ML para la ganancia de velocidad



Fuente: Elaboración propia.

Figura 9.

Modelos de ML para la ganancia de corriente



Fuente: Elaboración propia.

Parámetros de Validación

Los parámetros de validación de los algoritmos de Machine Learning entrenados para ser los modelos de ganancias adaptivas de la arquitectura de control propuesta (PPA) poseen los valores mostrados en la Tabla 4.

Tabla 4. *Parámetros de validación de los modelos de machine learning entrenados*

	RSE	MSE	OOB2 (%)
Ganancia de Velocidad			
PR	18.8	239.7	94.0
TD	0.1	78.2	98.0
RF	7.3e-4	34e-4	97.4
SVM	0.4	807.1	79.7
Ganancia de Corriente			
PR	5.9	2.7e-4	79.3
TD	0.1e-4	4e-12	99.9
RF	2.3e-4	1e-9	99.2
SVM	0.3	21e-4	65.5

Fuente: Elaboración propia.

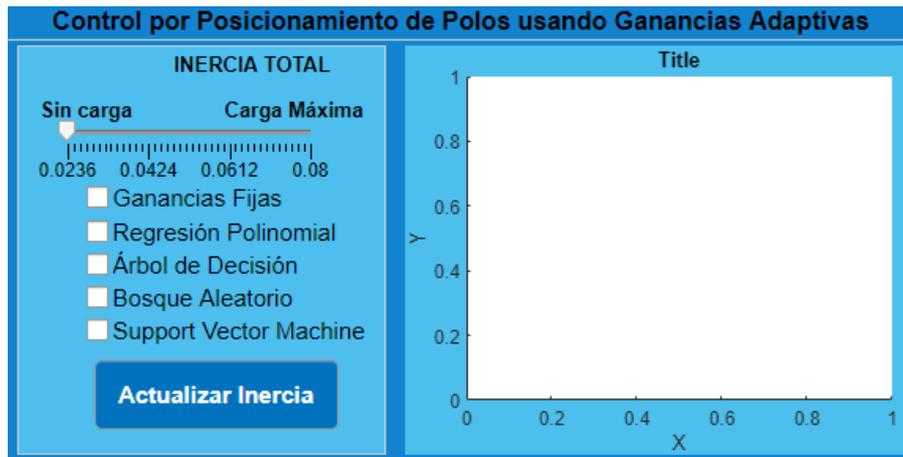
Lo que se desea es tener errores (RSE y MSE) bajos, y un coeficiente de determinación (OOB2) cercano al 100%. Para la ganancia de velocidad, el algoritmo con más bajos RSE y MSE es el de Bosque Aleatorio, aunque el Árbol de Decisión supera su OOB2 ligeramente, mientras que, para la ganancia de corriente, el algoritmo que alcanza los errores más bajos y el OOB2 más alto es el Árbol de Decisión. Es decir, según la Tabla 4, los modelos más adecuados para las ganancias de velocidad y corriente parecen ser TD y RF, respectivamente.

Simulación de la arquitectura propuesta

Con los modelos de machine learning entrenados se procede a implementar a nivel de simulación la arquitectura del Control por Posicionamiento de Polos Adaptivo (PPA) mostrada en la Figura 1.

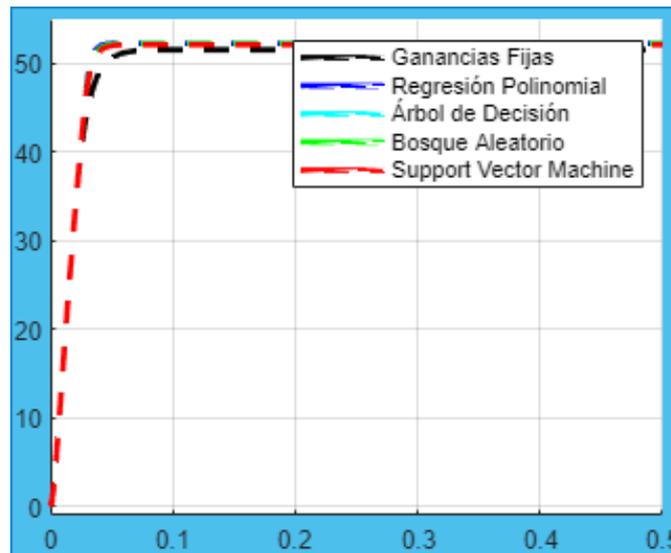
Se realizó una interfaz gráfica de usuario (GUI) en App Designer de Matlab, con el fin de poder variar la inercia desde un mínimo hasta 0.08 kg.m², y además mostrar las respuestas de velocidad del motor DC con la inercia indicada, respecto a 5 métodos: el control por posicionamiento de polos convencional, por PR, por TD, por RD y por SVM. La GUI se muestra en la Figura 11. El bloque de la derecha grafica las 5 respuestas de velocidad a cierta inercia (Figura 12).

Figura 10. GUI para la simulación de la arquitectura propuesta



Fuente: Elaboración propia.

Figura 11. Respuesta de velocidad para los 5 métodos



Fuente: Elaboración propia.

Al simular la arquitectura del control por posicionamiento de polos adaptivo (PPA), se halló el error promedio respecto al tiempo de establecimiento del ScPP (et_sP), el máximo sobrepico promedio (%MPP) y el error en estado estacionario promedio ($essP$). El error del tiempo de establecimiento se halla restando el t_s del modelo de referencia menos el t_s del motor con cierta inercia, para comprobar si se mantiene la respuesta del modelo de referencia. Los resultados se muestran en la Tabla 5, donde no solo se probaron los 5 modelos de control PP (convencional, PR, TD, RF y SVM) sino que también se usó una

adaptación híbrida entre dos modelos de machine learning, al usar RF para la ganancia de velocidad y TD para la ganancia de corriente.

Tabla 5. Error experimental porcentual del tiempo de establecimiento

t_s ScPP = 0.0315 s	etsP (s)	%MpP (%)	essP (%)
Convencional	36e-4	0	3.3
PR	8.2e-4	0	2.7
TD	6.1e-4	0	2.7
RF	6.4e-4	0	2.7
SVM	74e-4	0	2.8
TD+RF	6.5e-4	0	2.7

Fuente: Elaboración propia.

Ninguno de los modelos de Machine Learning genera sobrepico en la respuesta, indicando que, respecto a este parámetro, no empeora ni mejora el método convencional de control por posicionamiento de polos. Donde podemos ver mejoras es que, en promedio, los modelos de Machine Learning generan menos errores de estado estacionario que el método convencional de control PP, por lo que todos llegan a mantener la respuesta de velocidad deseada. Por último, observando el promedio de error del tiempo de establecimiento, los algoritmos de machine learning están bastante más cerca que el método convencional de control (PP), excepto el modelo de máquina de soporte vectorial, que, de hecho, es peor que el método convencional. El modelo híbrido fue bueno, pero no mejor que el modelo TD y RF por sí solos, ya que tuvo, en promedio, un mayor error en el t_s , aunque leve.

CONCLUSIONES

En la presente investigación se logró desarrollar una metodología y arquitectura capaz de lograr que el control por posicionamiento de polos pueda adaptar sus ganancias de retroalimentación de estados con el objetivo de mantener la respuesta ante aumentos de inercia en un motor DC.

Además, según los resultados obtenidos, el algoritmo de machine learning “Árbol de Decisión” tiene mejores prestaciones al momento de crear un modelo adaptivo de ganancias que intentan mantener la respuesta del motor sin inercia de carga ante aumentos del mismo, pues el error promedio respecto al tiempo de establecimiento del modelo es el menor con 6.1e-4 s.

El algoritmo de “Bosque Aleatorio” también es un modelo con buenas prestaciones para ser el modelo de las ganancias adaptivas, pero el modelo por “Árbol de Decisión” es levemente mejor al momento de cumplir nuestro objetivo, ya que es $0.3e-4$ s más cercano al modelo de referencia, en promedio.

Realizar un modelo adaptivo híbrido entre “Bosque Aleatorio” y “Árbol de Decisión” no permitió superar las prestaciones de los algoritmos por sí solos, a pesar de que intuitivamente los resultados anteriores a las simulaciones de prueba sugerían que sí. El error en tiempo de establecimiento del modelo híbrido fue de $6.5e-4$, mientras que de los algoritmos por sí solos llegaron a $6.1e-4$ y $6.4e-4$ s.

El algoritmo de “Regresión Polinomial” no supera la adaptación lograda con los algoritmos de “Árbol de Decisión” y “Bosque Aleatorio” al momento de querer alcanzar los parámetros de la respuesta del modelo, con un error en tiempo de establecimiento de $8.2 e-4$.

El algoritmo de “Máquina de Soporte Vectorial” no consiguió un buen modelo adaptivo, respecto al control por posicionamiento de polos convencional, incluso empeorando la respuesta, en promedio, estando mucho más lejos del modelo en $38e-4$ s.

Los algoritmos de “Regresión Polinomial”, “Árbol de Decisión” y “Bosque Aleatorio” lograron crear modelos de ganancias de corriente y velocidad del motor DC adaptivos; una característica que el método convencional de posicionamiento de polos no puede lograr.

REFERENCIAS BIBLIOGRÁFICAS

Rojas, A. (2001). *Control Avanzado: Diseño y Aplicaciones en Tiempo Real*. Universidad Nacional de Ingeniería. Perú.https://kupdf.net/download/-control-avanzado-diseo-y-aplicaciones-en-tiempo-real-arturo-rojas-moreno_59a8f835dc0d60065d568ee2_pdf

Gómez, J. y Mendoza, G. (2009). *Aplicación Del Control Adaptativo a Procesos Industriales Tipo ISO*. Universidad Pontificia Bolivariana. Bucaramanga.https://repository.upb.edu.co/bitstream/handle/20.500.11912/433/digital_17521.pdf?sequence=1

Ardila, V. (2018). *Control Variante en el Tiempo de Sistemas Multivariables Subactuados*. Universidad Tecnológica de Pereira.

- Colombia.<https://repositorio.utp.edu.co/server/api/core/bitstreams/af4e1ffe-c517-4459-8dc6-f81b8ad522a4/content>
- Arnáez, E. (2014). *Enfoque Práctico del Control Moderno con Aplicaciones en Matlab*. pp. 1-23. Universidad Peruana de Ciencias Aplicadas. Perú.<https://repositorioacademico.upc.edu.pe/handle/10757/345716>
- Scheinker, A. (2021). *Adaptive Machine Learning for Robust Diagnostics and Control of Time-Varying Particle Accelerator Components and Beams*. National Laboratory Los Alamos. USA. Doi: 10.20944/preprints202103.0301.v1
- Ann, J. y Shrivastava, S. (2018). *DC Motor Speed Control Using Machine Learning Algorithm*. International Journal of Engineering Research & Technology, 7(4).https://www.academia.edu/39282755/IJERT-C_Motor_Speed_Control_Using_Machine_Learning_Algorithm
- Hurtado, F. (2006). *Estimación de los Parámetros Físicos de un Motor de Corriente Directa usando el Algoritmo de Steiglitz-Mcbride*. Instituto Politécnico Nacional. México. <https://tesis.ipn.mx/handle/123456789/15660>
- Ogata, K. (2010). *Ingeniería de Control Moderno*. Quinta Edición, Pearson Educación. España.
- Kuo, B. (1996). *Sistemas de Control Automático*. Séptima Edición, Prentice HallHispanoamericano. México.
- Moreno, L., Garrido, S. y Bernaldo, C. (2003). *Ingeniería de Control. Modelado, Análisis y Control de Sistemas*. Ariel Ciencia. España.
- Zdzislaw, B. (2005). *Modern Control Theory*. Springer. Wroclaw University of Technology. Polonia. Doi: <https://doi.org/10.1007/3-540-28087-1>
- Landau, L., Lozano, R., M'Saad, M. y Karimi, A. (s.f.). *Adaptive Control*. Springer. Estados Unidos.
- Rodríguez, F. y López, M. (1996). *Control Adaptativo y Robusto*. Universidad de Sevilla. España.
- Müller, A. y Guido, S. (2016). *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O' Reilly Media. Estados Unidos.
- Deisenroth, M., Faisal, A. y Soon, C. (2020). *Mathematics for Machine Learning*. Cambridge University. Reino Unido. <https://mml-book.github.io/book/mml-book.pdf>

- Shai, S. y Shai, B. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University. Reino Unido.
<https://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf>
- Mirjalili, V. y Raschka, S. (2020). *Python Machine Learning*. Marcombo. España.
<https://books.google.com.co/books?id=5EtOEAAAQBAJ&hl=es>
- Géron, A. (2020). *Aprende Machine Learning con Scikit Learn, Keras y Tensorflow*. Anaya Multimedia. España.
- Monasterio, F. y Gutiérrez, A. (2021). *Modelado de un Motor DC*.
<http://www.robolabo.etsit.upm.es/asignaturas/seco/apuntes/modelado.pdf>
- CONTAVAL (2018). *Como afecta la Inercia a los Servos*. España.
<https://www.contaval.es/afecta-la-inercia-los-servos/>