La Mecatrónica en México, Mayo 2020, Vol. 10, No. 2, páginas 47 – 60
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

Sistema de Control PID Sintonizado por Algoritmo de Evolución Diferencial

Cervantes Marquez Aldo[1]✉, Gorrostieta Hurtado Efrén, Ramos Arreguín Juan Manuel, Torres Hernández Carlos Miguel, Rivas Araiza José Erik

[1]Facultad de Ingeniería, Universidad Autónoma de Querétaro,
✉aldocema04@gmail.com, efrengorrostieta@gmail.com

Resumen

El presente trabajo, muestra la sintonización de un controlador PID mediante un método heurístico llamado Evolución Diferencial (abreviado como ED), el cual está basado en la teoría de evolución de Darwin y siendo comparado con un método de sintonización clásico llamado Ziegler-Nichols, con el fin de observar la capacidad de computo evolutivo y proponer un método alternativo, evitando que los operadores y encargados de los lazos de control, seleccionen ganancias a los controladores de una manera empírica y con poca fiabilidad, debido a su desconocimiento sobre la teoría de control necesaria. Teniendo como finalidad mejorar los entornos industriales, mediante la reducción tiempos de sintonización, mientras se tiene una condición de paro no esperada o un paro programado por mantenimiento en el proceso. En esta ocasión se tiene como caso de estudio un motor de corriente directa con escobillas (motor DC), mediante un adquisidor de datos (data-logger), obteniendo su función de transferencia, para posteriormente ser aplicado el algoritmo de búsqueda ED en la obtención de las ganancias apropiadas en los requerimientos solicitados, cumpliendo con los altos estándares actuales de calidad en procesos, sin la necesidad de la adquisición de controladores más costosos.

Palabras clave: Motor de DC con escobillas, Sistema de control, Controlador PID, Algoritmo de Evolución Diferencial, Parámetros de sintonización.

Abstract

The present work shows tuning of a PID controller through a heuristic method called Differential Evolution (abbreviated as ED), which is based on Darwin's theory of evolution and being compared with a classical tuning method called Ziegler-Nichols, in order to observe the evolutionary computing capacity and propose an alternative method, avoiding that operators and managers of the control loops, select gains to the controllers in an empirical way and with little reliability, due to their lack of knowledge in the control theory. Aiming to improve industrial environments by reducing tuning times while having an unexpected stop condition or a scheduled stop for maintenance in the process. On this occasion, a direct current motor with brushes (DC motor) is used as a case study, through a data-logger where its transfer function obtained, and the search algorithm ED will be applied later in order to obtain the appropriate gains in the requested requirements, complying with the high standards of quality in processes, without the need for the acquisition of more expensive controllers.

Keywords: DC brushed motor, Control System, PID controller, Differential Evolution algorithm, Tuning parameters.

La Mecatrónica en México, Mayo 2020, Vol. 10, No. 2, páginas 47 – 60
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

1. Introducción

En el mundo existe la necesidad de realizar control sobre las cosas, esto nos permite obtener un mejor provecho y optimización de ellas. Los motores de corriente directa son un elemento fundamental en la industria y en la vida cotidiana, por lo que conocer sus características para ser controlados son aspectos muy importantes para realizar un mejor desempeño [1].

Actualmente es muy solicitada la necesidad de aplicar teorías de este tipo, en específicamente la identificación de sistemas lineales, lo cual permite conocer la ecuación característica (función de transferencia) de dicho sistema que requiere ser controlado, como consecuencia, es posible hacer un control más fino y preciso del sistema, teniendo como resultado un proceso más eficiente, estable y con mejores resultados.

Los controladores Proporcional-Integral-Derivativo (PID), son usados frecuentemente para regular procesos lineales de sistemas dinámicos. Estos controladores son muy populares debido a su simplicidad y excelente respuesta en un sistema cerrado. Sin embargo, encontrar la combinación adecuada de ganancias en estos sistemas suele ser complicado. El método de sintonización por Ziegler-Nichols es popular, debido a la facilidad de aplicarlo. No obstante, es complicado encontrar parámetros que cumplan con requerimientos específicos, en un lazo de control más sensible [2].

Debido a la poca variabilidad del método de Ziegler-Nichols, se han desarrollado métodos alternativos para la sintonización de controladores PID, mediante métodos exactos y métodos heurísticos, con el fin de obtener las ganancias adecuadas. Cada método tiene sus ventajas y desventajas. En el presente trabajo, se desarrollará un método heurístico adaptado para la búsqueda y recombinación de datos, con el fin de obtener las constantes del controlador adecuadas.

El algoritmo de evolución diferencial fue por primera vez propuesto por Rainer Storn y Kenneth Price en 1997 [3], es un algoritmo de optimización heurístico, esto quiere decir, que es un método no exacto y se basa en la experiencia y el aprendizaje del propio algoritmo. Siendo parte de los algoritmos genéticos en su desarrollo y operación para la convergencia de soluciones óptimas, más no exactas [4].

El algoritmo de Evolución Diferencial para la resolución de problemas multi-objetivo (abreviado como MODE por sus siglas en Inglés), consiste en la obtención un vector solución y evaluarlo, para posteriormente obtener una aproximación de la función objetivo (solución que cumple con criterios establecidos), y en función de la dispersión con respecto a las restricciones designadas, el algoritmo muta al vector solución, a través de vectores de prueba que se encuentran dentro de la población inicial acotada cruzándose entre sí, logrando mediante de un numero de iteraciones finita, un vector solución con un margen de error al ser evaluada en la función objetivo [5].

1.1 Importancia de los Métodos Heurísticos para la Resolución de Problemas de Optimización.

Los métodos de optimización a través de heurística, son aplicables para problemas con soluciones que no son exactas, y/o existen una n cantidad de soluciones correctas con algún margen de error (tolerancia), en donde un método exacto tardaría mucho tiempo (miles de cálculos) para llegar al resultado, y probablemente la capacidad de cómputo sea insuficiente para resolver el problema [6].

Los algoritmos heurísticos se basan en comportamientos de la naturaleza que puedan permitir realizar una mejora continua sobre los métodos exactos, como es el caso de una sintonización de un controlador PID, donde existen n cantidad de combinaciones, que permiten desarrollar correctamente el control del proceso como es requerido [7].

La Mecatrónica en México, Mayo 2020, Vol. 10, No. 2, páginas 47 – 60
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

2. Justificación

El problema de realizar una sintonización correcta es un tema importante, debido a que es complicado realizar este tipo de cálculos, y a pesar de realizarlos es muy complicado obtener resultados totalmente exactos, debido a las propias condiciones del proceso y muchos otros factores externos que no se contemplan en el modelo matemático. Por lo que, la computación evolutiva y los métodos heurísticos para la resolución de problemas de sintonización, son herramientas factibles, para la resolución de este tipo de problemas.

Mediante la aplicación de teoría de control y computación evolutiva, es posible obtener la función de transferencia de un sistema (en este caso, un motor de corriente directa con escobillas). Para posteriormente poder aplicar un algoritmo evolutivo para obtener las ganancias que satisfagan y/o mejoren las condiciones del sistema.

3. Materiales y métodos

El sistema de control para el desarrollo del proyecto, consta de un adquisidor de datos (data-logger) que tiene la capacidad de recibir y procesar los datos provenientes del encoder. Así como también, posee un modo controlador para realizar un esfuerzo de control, mediante el controlador PID, en donde se contempla una parte eléctrica y una parte de software (véase Figura 1).

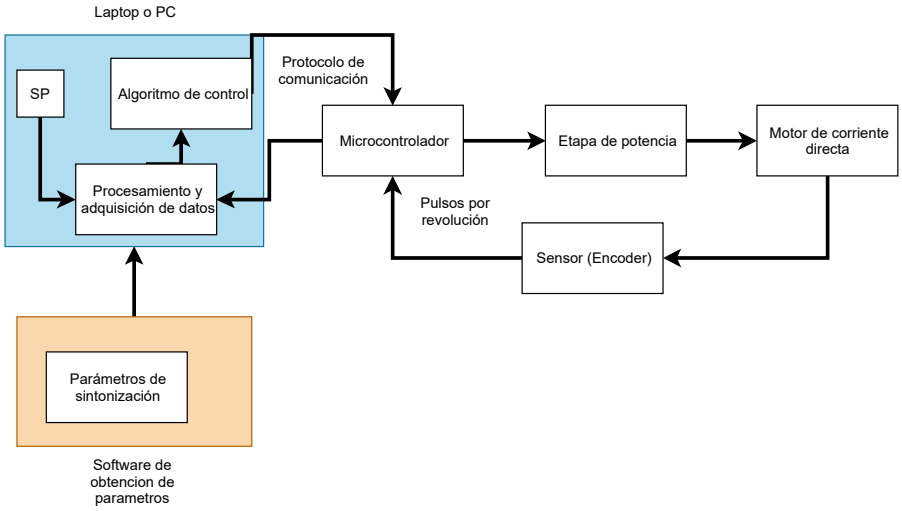

Figura 1. Diagrama de bloques del prototipo experimental (data-logger).

3.1 Equipo de medición.

En la Tabla 1 se muestra el equipo de medición que se utilizó, siendo este equipo de utilidad en la validación de mediciones y de resultados.

1.1 Material electrónico.

En la Tabla 2 se muestra el equipo electrónico a utilizar, este se caracteriza por funcionar con corriente directa y serán componentes principales del data-logger, para realizar las pruebas correspondientes.

Tabla 1. Equipo de medición.

Cantidad	Nombre	Modelo	Características
1	Multímetro	MUT-33	0 a 20V en corriente directa, corriente con un rango de 0 a 10A de corriente directa, resistencia de 0 a 20MΩ.
1	Osciloscopio	DSO138	Amplitud de 50V con ancho de banda de 0 a 200KHz a 12 bits, tiempo de muestreo de 10μs a 500s.

Tabla 2. Material electrónico.

Cantidad	Nombre	Modelo	Características
1	Motor de DC	625500/C	Motor con relación de engranajes 100:1 con una velocidad angular máxima de 80R.P.M. a 10.53V.
1	Encoder	V90-021	Encoder magnético (sensor efecto hall) con 100 PPR (pulsos por revolución).
1	Microcontrolador	PIC18F4550	Microcontrolador de 8 bits con una frecuencia de reloj de 20MHz y un tiempo de muestreo de 0.01s.
1	USB a TTL	TIt PL2303	Adaptador de niveles lógicos de USB a TTL con capacidad de transmisión de hasta 115200Baud.
1	Puente H	Módulo L298N	Módulo de amplificación de potencia mediante señales PWM.

1.2 Software de apoyo y de uso.

Estos componentes son adicionales y de apoyo para el prototipo experimental como son los programas computacionales ocupados y que fueron utilizados para el procesamiento de datos (véase Tabla 3). Los programas fueron ejecutados en una computadora tipo Laptop de la marca acer modelo Aspire 5 A515-51-52BQ.

Tabla 3. Software utilizado en el proyecto.

Lenguaje	Programa	Función
C	Pic C compiler	Programa que sirve para introducir un programa ejecutable en el microcontrolador
C#	Visual Studio	Interfaz gráfica del data-logger donde se visualizan y se mandan esfuerzos de control al sistema
Python	Spider 3.3.6	Programa donde se realizó el algoritmo de evolución diferencial con simulaciones del sistema
--	Proteus 8	Programa que se utilizó para simular y crear placas PCB para el data-logger
Matlab	Matlab/Simulink	Programa en donde se realizaban simulaciones del sistema de control desarrollado

1.3 Métodos.

La metodología a seguir para el desarrollo, integración y análisis de resultados del trabajo, es mostrada en la Figura 2 y posteriormente descritas.

La Mecatrónica en México, Mayo 2020, Vol. 10, No. 2, páginas 47 – 60
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

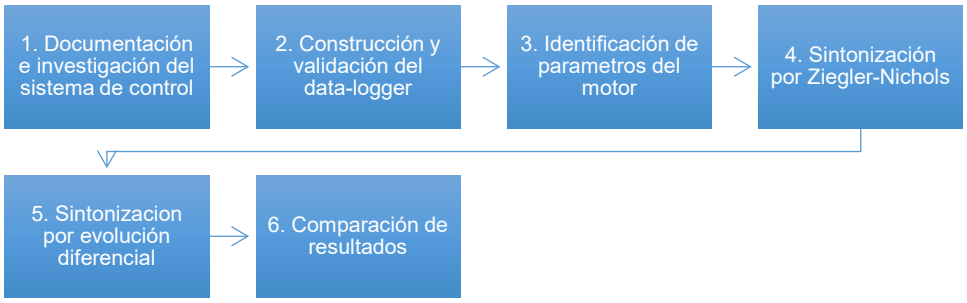

Figura 2. Diagrama de metodología del proyecto.

1. Será necesario realizar una investigación sobre los conceptos necesarios para tener los conocimientos necesarios para construcción del data-logger, así como también la programación de la interfaz gráfica y la teoría de control.
2. La construcción del data-logger implica desarrollar los circuitos necesarios, así como también la programación de la interfaz gráfica, en la que se obtendrán y almacenaran los datos mediante la creación de archivos .TXT y .PNG.
3. Para identificar los parámetros del motor de corriente directa, se realizará una prueba de escalón, la cual consiste en introducir al sistema una referencia tipo escalón y graficar el comportamiento del motor, para poder identificar el orden del sistema. En caso de ser de orden 1 (sin sobrepaso o sobrepaso despreciable) se aplicará la ecuación (1) (véase Figura 3) según la regla de identificación por Ziegler-Nichols [8] [9].

$$\frac{Y(s)}{U(s)} = \frac{Ke^{-Ls}}{\tau s + 1} \tag{1}$$

Donde:

- $\frac{Y(s)}{U(s)}$ es la función de transferencia del motor.
- K es el valor en estado estacionario.
- τ es la constante de tiempo del motor
- L es el tiempo de retardo

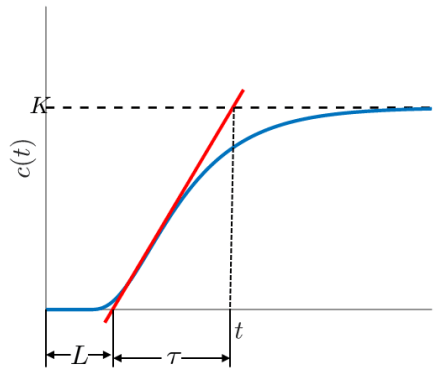

Figura 3. Método de identificación de un sistema de primer orden [8].

La Mecatrónica en México, Mayo 2020, Vol. 10, No. 2, páginas 47 – 60
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

4. Una vez que se tienen las constantes de la ecuación (1) será necesario aplicar la regla se sintonía propuesta en el método de Ziegler-Nichols [8] de un controlador PID (véase Tabla 4), para posteriormente realizar una prueba con el sistema compensado.

Tabla 4. Regla de sintonía de Ziegler-Nichols basada en la respuesta escalón de la planta.

Tipo de controlador	k_p	T_i	T_d
P	$\frac{\tau}{L}$	∞	0
PI	$0.9\frac{\tau}{L}$	$\frac{L}{0.3}$	0
PID	$1.2\frac{\tau}{L}$	$2L$	$0.5L$

Con lo que cabe resaltar en (2) la forma del controlador PID paralelo en transformada de Laplace [10] [11] [12] y en (3)(4) la conversión de las ganancias k_p, T_i, T_d a k_p, k_i, k_d.

$$\frac{U(s)}{E(s)} = k_p + \frac{k_i}{s} + k_d s \tag{2}$$

$$k_i = \frac{k_p}{T_i} \tag{3}$$

$$k_d = k_p * T_d \tag{4}$$

Una vez introducidas las ganancias en el controlador PID se graficará el comportamiento del sistema con esta configuración de ganancias.

5. Para realizar la sintonización por algoritmo de Evolución Diferencial se utilizará un programa en el lenguaje de Python para realizar los cálculos necesarios y la lógica correspondiente, introduciendo parámetros de entrada como son la función de transferencia y restricciones deseadas para el comportamiento del motor. Con la finalidad de obtener la combinación de ganancias adecuadas. Se muestra en la Figura 4, los pasos que el programa llevará a cabo para realizar la sintonización adecuada [13].

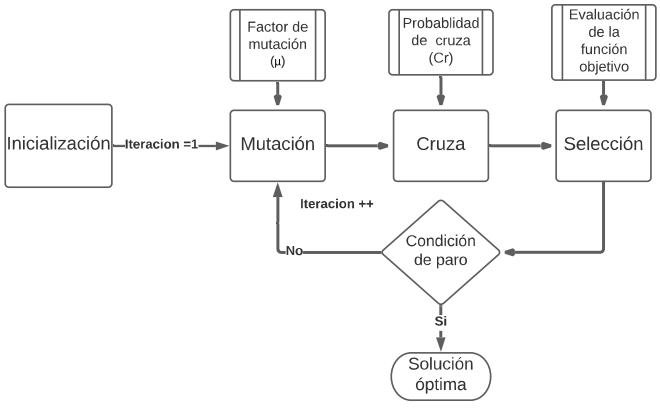

Figura 4. Diagrama de bloques de método de Evolución Diferencial.

La Mecatrónica en México, Mayo 2020, Vol. 10, No. 2, páginas 47 – 60
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

6. Finalmente una vez graficados los resultados, se compararán los datos obtenidos y se interpretaran dichos datos para obtener una conclusión del proyecto, con ayuda de simulaciones y validaciones de los procedimientos para obtener fiabilidad y certeza en los datos [14] [15] .

2. Resultados

2.1 Construcción del adquisidor de datos (data-logger).

Se realizaron dos circuitos para el data-logger (véase Figura 5), un circuito de control, el cual alberga al PIC18F4550, además de una serie de puertos I/O para la recepción y envío de datos, así como su respectivo puerto de alimentación de energía eléctrica, proveniente de una fuente lineal.

El segundo circuito es un circuito de aislamiento, que está conformado por un aislamiento óptico (opto acoplador) y una compuerta lógica para la digitalización de los datos, la finalidad de esta placa es aislar los niveles de tensión del microcontrolador, de los niveles de tensión del motor de corriente continua, el cual tiene una fuente de alimentación independiente a la del microcontrolador. Esto para evitar variaciones de corriente en el circuito de control, y evitar que el microcontrolador pueda dañarse o resetearse el sistema.

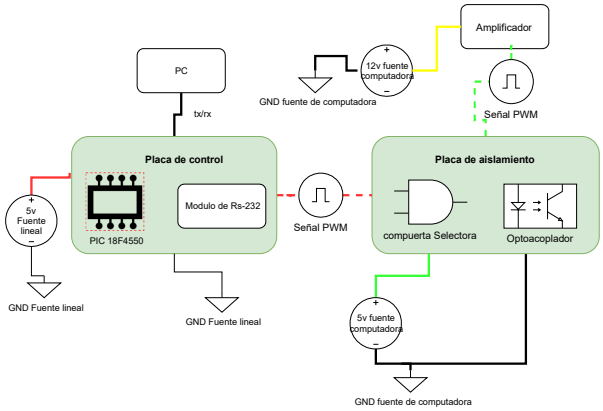

Figura 5. Diagrama de bloques de circuitos de control y aislamiento.

Dichos circuitos fueron convertidos a placas PCB, con la finalidad de evitar cortos circuitos y falsos contactos, teniendo el sistema electrónico completo en la Figura 6.

Posteriormente se realizó una integración y validación del sistema electrónico con la interfaz gráfica, obteniendo la visualización de los datos mientras el sistema electrónico trabaja a la par (véase Figura 7).

2.2 Software de sintonización.

Se realizó un programa en Python donde se muestra el resultado de cada iteración (véase Figura 8), además de que los datos son almacenados en variables para poder ser graficados posteriormente. Una vez finalizadas las iteraciones, se muestra una gráfica aproximada donde se simula la respuesta del sistema, y despliega los resultados más importantes del cálculo realizado.

La Mecatrónica en México, Mayo 2020, Vol. 10, No. 2, páginas 47 – 60
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

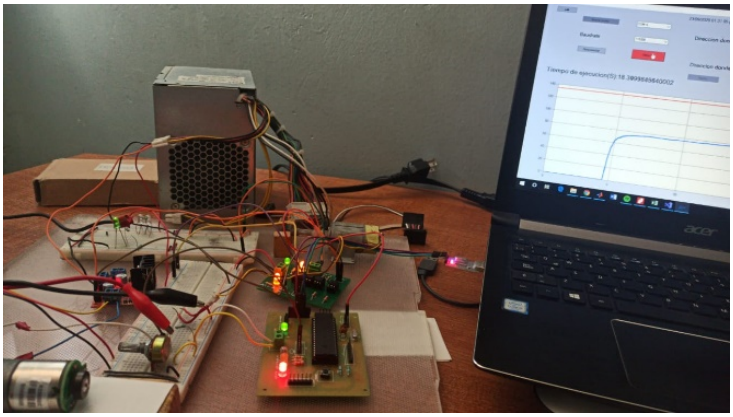

Figura 6. Prototipo experimental.

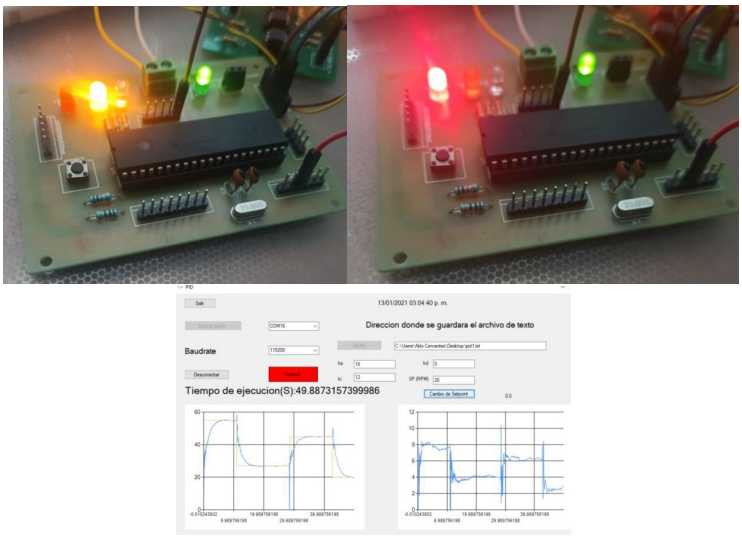

Figura 7. Placa de control con detalle a etapas de funcionamiento e interfaz gráfica.

```
[10.89570395  6.53153528  0.692563  ]
[10.89570395  5.93766129  0.692563  ]
[10.89570395  5.93766129  0.692563  ]
[20.          5.93766129  0.57647202]
[20.          5.93766129  0.40732797]
[13.25293065  5.93766129  0.24328143]
[13.25293065  5.93766129  0.24328143]
[13.25293065  1.95643653  0.24328143]
[13.25293065  1.95643653  0.68721145]
[13.25293065  1.95643653  0.68721145]
[13.25293065  1.95643653  0.42631831]
[13.25293065  1.95643653  0.42631831]
[13.25293065  1.95643653  0.42631831]
[12.01501063  1.95643653  0.30537943]
[12.01501063  2.22387381  0.51080266]
[12.01501063  2.22387381  0.2764848 ]
[12.01501063  2.22387381  0.2764848 ]

Numero de iteraciones= 200
referencia [R.P.M.]= 55
Valor pico [R.P.M.]= 54.99946682945927
sobrepaso [%]= 0
y_f= 54.99946682945927
kp= 12.01501063046201 ki 2.223873812204859 kd= 0.27648479584654084
```

Figura 8. Pantalla de software en Python para sintonización de controlador PID.

La Mecatrónica en México, Mayo 2020, Vol. 10, No. 2, páginas 47 – 60
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

2.3 Parámetros del prototipo experimental (prueba de escalón).

Mediante la metodología propuesta, se obtuvieron los siguientes resultados de los parámetros del prototipo experimental (motor de DC con data-logger) mostrados en la Tabla 5.

Tabla 5. Parámetros del prototipo experimental.

Símbolo	Valor
K	7.613
τ	0.14
L	0.0095

Como se puede observar, el retardo L es muy pequeño, y al realizar la experimentación necesaria, se comprobó que es despreciable en la dinámica del sistema, esto puede ser atribuido al tiempo de muestreo ($t_m = 0.01s$), que al momento de que iniciar la lectura de datos del data-logger, no puede medir valores actuales ni anteriores, por lo que considera la velocidad como 0, y por consiguiente ese intervalo se encuentra vacío. Una vez dicho esto, se encontró una aproximación a un sistema sin retardo en la ecuación (5).

$$\frac{7.613}{0.14s+1} \approx \frac{7.613e^{-0.0095s}}{0.14s+1} \tag{5}$$

En la Figura 9 se observa la función (5) evaluada de 0 a 3.25 segundos, en donde se puede apreciar la aproximación de la función obtenida con el comportamiento del data-logger.

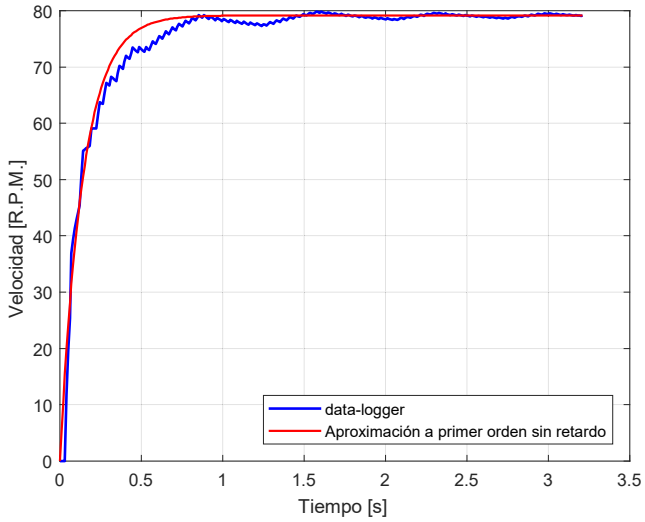

Figura 9. Aproximación del sistema a un primer orden sin retardo a partir de prueba de escalón.

2.4 Ganancias y respuesta del controlador PID por Ziegler-Nichols.

Las ganancias obtenidas para el controlador PID en la Tabla 6, fueron obtenidas mediante la aplicación de la regla de sintonía propuesta en la Tabla 4, y siendo sustituidas por los valores obtenidos de la Tabla 5.

La Mecatrónica en México, Mayo 2020, Vol. 10, No. 2, páginas 47 – 60
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

Tabla 6. Ganancias del controlador PID por Ziegler-Nichols.

Símbolo	Valor
k_p	18.667
T_i	0.018
T_d	0.0045
k_i	1037.0.74
k_d	0.084

Una vez introducidos los valores de las ganancias en la interfaz gráfica, fue sometido a un valor deseado de 55 R.P.M. para observar el comportamiento del sistema compensado, siendo expuesto en la Figura 10.

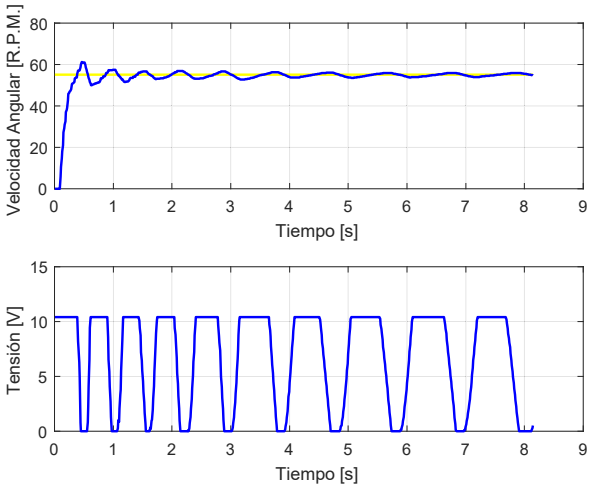

Figura 10. Respuesta del sistema compensado con sintonización por Ziegler-Nichols.

Como se observa en la Figura 10 se tiene un comportamiento oscilatorio, debido a la alta ganancia integral, afectando al actuador (puente H), el cual sufre cambios bruscos de tensión y corriente, soportando picos muy altos de tensión.

2.5 Ganancias y respuesta del controlador PID por evolución diferencial.

Para realizar la sintonización por evolución diferencial, fue necesario introducir algunos parámetros de entrada y restricciones (véase Tabla 7), para el correcto funcionamiento del programa.

Tabla 7. Restricciones del algoritmo de evolución diferencial.

Símbolo	Descripción	Valor
n	Número de iteraciones	200
μ	Factor de mutación	0.40
C_r	Probabilidad de cruce	0.56
t_s	Tiempo de subida	$< 7\ s$
m_p	Sobrepaso	$\leq 27\ \%$
s_p	Valor deseado	55 R.P.M.
k_p	Ganancia proporcional	$0.01 \leq k_p \leq 20$
k_i	Ganancia integral	$0.02 \leq k_i \leq 10$
k_d	Ganancia derivativa	$0 \leq k_d \leq 1$

Una vez corrido el algoritmo con las restricciones solicitadas, se obtuvieron los resultados de la Tabla 8 correspondientes a la simulación del algoritmo de evolución diferencial en Python.

Tabla 8. Resultados obtenidos por el algoritmo de evolución diferencial.

Símbolo	Descripción	Valor
y_f	Valor aproximado final	54.9989 R.P.M.
m_p	Sobrepaso	0 %
t_s	Tiempo de subida	5.2 s
k_p	Ganancia proporcional	6.8134
k_i	Ganancia integral	7.9740
k_d	Ganancia derivativa	0.7391

Con los valores de las ganancias obtenidas en la Tabla 8 se sustituyeron en el prototipo experimental y se graficó la respuesta del sistema compensado obteniendo la Figura 11.

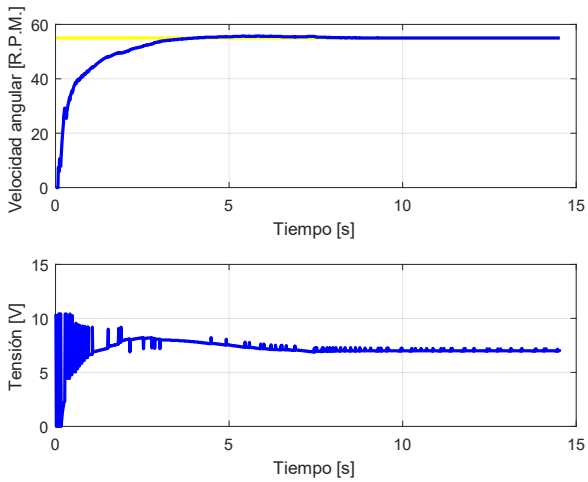

Figura 11. Respuesta del controlador con ganancias obtenidas por el algoritmo.

En la Tabla 9 se muestran los valores más relevantes de la Figura 11, que siendo comparados con la Tabla 8, es posible observar una diferencia en el sobrepaso, debido a la saturación del sistema en ese intervalo de operación, la cual no es considerada por la simulación del algoritmo, que en el prototipo experimental es representada por la limitación de tensión de la fuente de alimentación y del puente H, así como también, en la discretización del sistema en tiempos de muestreo constantes de $0.01\ s$, existiendo perdidas en los datos de adquisición y la respuesta del controlador [16].

En la Figura 12 se muestran las iteraciones que realizó el algoritmo para poder llegar a la solución final tras las 200 iteraciones, se muestra con un triángulo rojo la iteración $n = 0$ y con una estrella la iteración $n = 200$. Es posible observar que el camino que sigue no tiene un patrón de dirección establecido, debido a que se tienen las constantes de mutación y probabilidad de cruce (μ y C_r). Estas constantes permiten que la siguiente iteración no dependa completamente de la iteración anterior, sino también del elemento de la población nuevo, el cual es aleatorio. Logrando evitar que los resultados se estanquen en un valor mínimo o máximo local. Comprobando que es posible obtener una gran cantidad de valores combinatorios que puedan satisfacer el rango de restricciones.

La Mecatrónica en México, Mayo 2020, Vol. 10, No. 2, páginas 47 – 60
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

Tabla 9. Resultados obtenidos por el prototipo experimental.

Símbolo	Descripción	Valor
y_f	Valor aproximado final	54.9784 R.P.M.
m_p	Sobrepaso	1.317 %
t_s	Tiempo de subida	4.036 s
V_{ss}	Tensión en estado estacionario	7.9 V
k_p	Ganancia proporcional	6.8134
k_i	Ganancia integral	7.9740
k_d	Ganancia derivativa	0.7391

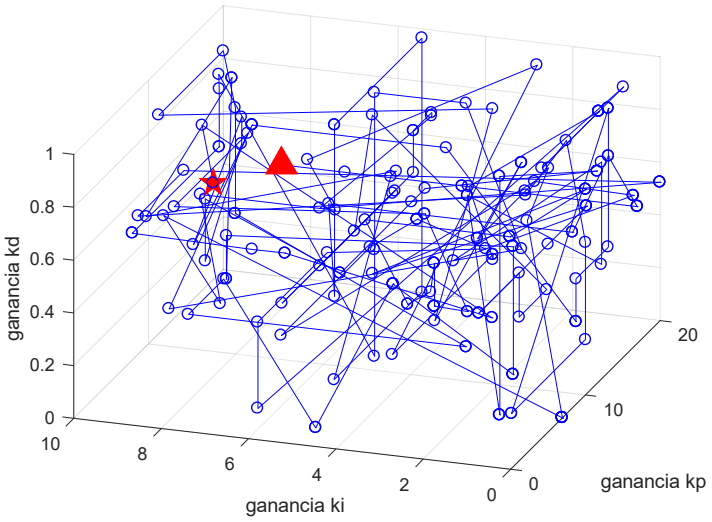

Figura 12. Dispersión y dirección de las iteraciones.

3. Conclusiones

Se construyó un prototipo experimental para obtener las mediciones de un motor de corriente directa con encoder, así como también el software embebido para las operaciones necesarias, visualización y almacenamiento de datos.

Se tuvieron inconvenientes al momento de integrar el prototipo experimental con la interfaz gráfica, pues se tuvo que sincronizar el tiempo de muestreo con el tiempo de procesamiento y almacenamiento de datos, con el fin de obtener un tiempo de muestreo de 0.01 s. También cabe resaltar que las placas PCB permitieron reducir el ruido electromagnético externo y falsos contactos, permitiendo una mejor reproductibilidad de las mediciones. Sin embargo cabe resaltar que el prototipo experimental tiene la capacidad de poder controlar cualquier tipo de motor, siempre y cuando su actuador pueda ser ajustado a los requerimientos del motor, adicionalmente el elemento sensor (encoder) que pueda ser compatible con el sistema.

Cabe destacar que el sistema sintonización puede servir para cualquier sistema de n orden, siempre y cuando se pueda medir y conocer su función de transferencia aproximada, esto quiere decir que, no solamente este algoritmo podrá servir para motores, sino también para otro tipo de sistemas como hornos, mezcladores, tanques, robots manipuladores, etc.

En este trabajo se observó el comportamiento de la sintonización de controladores PID por dos métodos diferentes. Por un lado se tiene un método clásico y muy utilizado para sintonizar lazos de control, teniendo el inconveniente de que son necesarias pruebas adicionales para ajustar algunos valores de la función de transferencia, que en consecuencia implican invertir más tiempo en realizar ese ajuste manual de ganancias, de una manera fina y empírica.

Por otro lado se tiene el método heurístico, el cual permite realizar una simulación de varios escenarios en los que será sometido el proceso, gracias a las restricciones es que se pudo suavizar y acotar el comportamiento del sistema, teniendo como principal ventaja su versatilidad para distintos parámetros de sintonización, en comparación al método clásico en el que únicamente existe una combinación para el controlador PID, sin tomar en cuenta las restricciones o requerimientos que se deseen en ese sistema.

Referencias

[1] Leal G. Importancia de la Instrumentación y el Control en el Sector Industrial n.d. https://www.lymcapacitacion.com/blog/16158/instcontrol (accessed March 10, 2021).
[2] Chiha I, Liouane H, Liouane N. International Review of Modelling and Simulations (IREMOS) A Hybrid Method Based On Multi-objective Ant Colony Optimization and Differential Evolution to Design PID DC Motor Speed Controller. International Review of Modelling and Simulations 2012;5.
[3] Storn R, Price K. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. Journal of Global Optimization 1997.
[4] Algoritmo de búsqueda - EcuRed n.d. https://www.ecured.cu/Algoritmo_de_b%C3%BAsqueda (accessed March 11, 2021).
[5] Santana L. Un Algoritmo Basado en Evolución Diferencial para Resolver Problemas Multiobjetivo 2004:142.
[6] Martínez DMG. Cómputo Evolutivo 2019:29.
[7] Sibalija T. Metaheuristic Algorithms in Industrial Process Optimisation: Performance, Comparison and Recommendations 2020;1198:270–83. https://doi.org/10.1007/978-981-15-5232-8_24.
[8] Ogata K. Ingeniería de control moderna. Pearson educación; 2010.
[9] Rodríguez F. Dinámica de Sistemas. Trillas; 1989.
[10] Nasreldrin M. IMPLEMENTATION OF A PID CONTROL SYSTEM ON MICROCONTROLLER (DC MOTOR CASE STUDY). International Conference on Communication, Control, Computing and Electronics Engineering 2017.
[11] Hernández V, Silva R. Automatic Control with Experiments. Springer; 2019.
[12] Hernández V, Silva R, Carrillo V. Control Automático: Teoría de diseño, Construcción de Prototipos, Modelado, Identificación y Pruebas Experimentales. México: Colección CIDETEC del Instituto Politécnico Nacional; 2013.
[13] Idir A, Kidouche M, Bensafia Y, Khettab K, Tadjer SA. Speed Control of DC Motor Using PID and FOPID Controllers Based on Differential Evolution and PSO). International Journal of Intelligent Engineering & Systems 2018.
[14] Convert model from discrete to continuous time - MATLAB d2c n.d. https://www.mathworks.com/help/control/ref/lti.d2c.html;jsessionid=47b20ec7a19920df7816acd12c15 (accessed March 11, 2021).
[15] Continuous-time or discrete-time PID controller - Simulink n.d. https://www.mathworks.com/help/simulink/slref/pidcontroller.html#br9ejg0-18 (accessed March 11, 2021).
[16] Romero A, Muñoz A, Gómez J. Realimentación de velocidad con encoders de baja resolución en Simulink. Comité Español de Automática de La IFAC (CEA-IFAC) 2015.

Autores

Aldo Cervantes Marquez. Estudiante de la carrera de ingeniería en automatización con línea terminal en instrumentación y control de procesos de la facultad de ingeniería de la Universidad Autónoma de Querétaro (UAQ). Miembro del PMI y miembro del comité estudiantil ISA-UAQ. Desarrollador de proyectos (Project manager) de automatización y sistemas de control con aplicaciones en procesos industriales.

Efrén Gorrostieta Hurtado. Doctorado en ingeniería con especialización en mecatrónica, maestría en ciencia y tecnología con especialización en automatización y control. Graduado en 1992 del instituto tecnológico y de estudios superiores de oriente (ITESO) en Guadalajara Jalisco México, como ingeniero en electrónica con especialidad en control e ingeniería biomédica. Ha trabajado como profesor en el ITESO en el área de control. En el instituto tecnológico de Morelia y Querétaro, como profesor, coordinador y jefe del área de posgrado. También trabajó en la universidad Lasalle bajío. Actualmente es profesor investigador de la Universidad Autónoma de Querétaro (UAQ) en la facultad de ingeniería. Es autor de varios artículos y editor de libros sobre robótica y automatización. Sus intereses actuales de investigación son en el campo de sistemas de control mecatrónicos, robótica y machine learning. Miembro fundador de la Asociación Mexicana de Mecatrónica (AMM, Mecamex), actualmente trabajando como editor de varios libros de robótica y automatización.

Juan Manuel Ramos Arreguín. Tiene Doctorado en Ciencias y Tecnología con Especialidad en Mecatrónica en el Centro de Ingeniería y Desarrollo Industrial. Con Maestría en Ingeniería Eléctrica con opción en Instrumentación y Sistemas Digitales en la Universidad de Guanajuato. La ingeniería en Comunicaciones y Electrónica en la Universidad de Guanajuato. Fue Presidente de la Asociación Mexicana de Mecatrónica en el periodo 2013 a 2016. Pertenece al SNI en nivel I. Cuenta con reconocimiento al perfil PRODEP. A partir de 2017 es Senior Member en el IEEE. Profesor de tiempo completo en la Universidad Autónoma de Querétaro, en la Facultad de Ingeniería.

Carlos Miguel Torres Hernández. Egresado de la carrera de Ingeniería en Mecatrónica en 2013 del Instituto Tecnológico y de Estudios Superiores de Monterrey Campus San Luis Potosí, con una maestría en Ciencias con línea terminal en Instrumentación y Control Automático en la Universidad Autónoma de Querétaro (UAQ), continuando con los estudios de doctorado en Gestión Tecnológica e Innovación en la misma institución. Miembro Activo desde 2013 del Institute of Electrical and Electronics Engineers, vicepresidente del IEEE Sección Querétaro de 2016 a 2018 y presidente de la Sección 2019-2020. Miembro del Comité Latinoamericano de Vinculación con la Industria de IEEE. Docente de la Facultad de Ingeniería y Coordinador de Vinculación Académica Empresarial de la UAQ. Representante de la UAQ ante la Red de Vinculación Centro Sur y la Red de Innovación en Educación Superior de la Asociación Nacional de Universidades e Instituciones de Educación Superior.

José Erik Rivas Araiza. Tiene más de 10 años de experiencia como ingeniero de software de backend con experiencia en automatización y desarrollo web en proyectos nacionales e internacionales. Es un apasionado en la creación de módulos de software de backend y colaborador en proyectos académicos. Le gusta expandir su conocimiento mediante aprendizaje continuo sobre ingeniería de datos, ciencia de datos y enseñando en la academia. Cuenta con maestría en instrumentación y sistemas de control.