Revista Internacional de Investigación e Innovación Tecnológica

Página principal: www.riiit.com.mx

Implementación de un controlador PID difuso con rechazo a perturbaciones aplicado a un motor de corriente directa basado en sistemas embebidos de bajo costo

Fuzzy PID controller implementation with disturbance rejection applied to a direct current motor based on low cost embedded systems

Beltrán-Escobar, A.M.[1], Rumbo-Morales, J.Y.[2], Valdez-Martínez, J.S.[1], Azcaray-Rivera, H.R.[2], Rodríguez-Cerda, J.C.[3]

[1] Universidad Tecnológica Emiliano Zapata del Estado de Morelos. Av. Universidad Tecnológica No. 1, Col. Palo escrito, Emiliano Zapata, Morelos, México. C.P. 62760.
[2] Universidad de Guadalajara. Centro Universitario de los Valles. Carretera Guadalajara-Ameca Km.45.5, Ameca, Jalisco, México. C.P. 46600.
[3] Instituto Tecnológico Superior de Poza Rica. Av. Luis Donaldo Colosio s/n, Ejido Arroyo del Maiz, Poza Rica de Hidalgo, Veracruz. México. C.P. 93230.
Correspondencia: miguelbeltran@utez.edu.mx, jorgevaldez@utez.edu.mx; jjrumbo@hotmail.com, hector.azcaray.rivera@gmail.com; julio.rodriguez@itspozarica.edu.mx

Innovación tecnológica: Sistema de control de bajo costo**.**

Área de aplicación industrial: Tecnificación de procesos industriales.

Enviado: 03 junio 2019.
Aceptado: 20 octubre 2019.

Abstract

This paper shows the implementation of a fuzzy PID controller applied to the speed control of a direct current motor (DCM) in order to show the robustness of the controller when rejecting mechanical disturbances using low cost embedded system. A didactic tests bank was built consisting of a DCM and a power stage coupled to a CD generator that fulfills the function of sensing the speed. The results obtained for a fuzzy PID controller are presented using an Arduino MEGA which is based on the ATmega2560 microcontroller. For the implementation of the fuzzy PID control, LabVIEW was used, the programming of the control algorithm is described, as well as the design of the graphical interface that allows to show the dynamic response of the DCM for the tracking of fixed and variable speed references to the rejection of disturbances applied to the

motor. The tests carried out consist of the tracking of four undisturbed velocity trajectories and subsequently applying unknown mechanical perturbations.

Keywords: CD motor, PID control, Fuzzy control.

Resumen

En este trabajo se muestra la implementación de un controlador PID difuso aplicado al control de velocidad de un motor de corriente directa (MCD) con el fin de mostrar la robustez del controlador ante el rechazo a perturbaciones mecánicas utilizando sistemas embebidos de bajo costo. Se construyó un banco de pruebas didáctico que consiste en un MCD y una etapa de potencia con acoplamiento a un generador de CD que cumple la función de sensar la velocidad. Se presentan los resultados obtenidos para un controlador PID difuso utilizando un Arduino MEGA el cual se basa en el microcontrolador ATmega2560. Para la implementación del control PID difuso se utilizó LabVIEW, se describe la programación del algoritmo de control, así como el diseñó de la interfaz gráfica que permite mostrar la respuesta dinámica del MCD para el seguimiento de referencias de velocidad fijas y variables ante el rechazo de perturbaciones aplicadas al motor. Las pruebas realizadas consisten en el seguimiento de cuatro trayectorias de velocidad sin perturbaciones y posteriormente aplicando perturbaciones mecánicas desconocidas.

Palabras clave: Motor de CD, Control PID, Control difuso.

1. Introducción

Hoy en día se han reportado trabajos en los que se han desarrollado diseños y simulaciones de sistemas de control que combinan el uso de un controlador PID convencional con técnicas que involucran a la lógica difusa como parte de sus estrategias de compensación ante el rechazo de perturbaciones y reconfiguración de los parámetros del controlador, así como acciones de compensación en una o más variables a controlar ante cambios en la referencia o en las condiciones de operación. Con el fin de llevar a la implementación un control PID difuso en este trabajo se desarrolló un prototipo experimental que consiste en un banco de pruebas en el que se acopló un motor de corriente directa (MCD) con un generador del mismo tipo que otorga una salida de voltaje que varía proporcional y linealmente ante el incremento de velocidad del MCD, con el generador se consigue sensar el valor de velocidad presente en el motor.

El banco de pruebas permite estudiar la respuesta dinámica y el desempeño del sistema de control con dispositivos de bajo costo, se utiliza una computadora para programar y ejecutar el algoritmo de control y se cuenta con la posibilidad de introducir perturbaciones mecánicas desconocidas en instantes de tiempo durante la realización de las pruebas.

La organización de este trabajo es la siguiente: en la sección de marco teórico se hace una revisión general del estado del arte a cerca de las características de las plataformas de hardware utilizadas para la implementación del control PID difuso, así

como de la plataforma de programación que se utilizó. En esta misma sección se tratarán las ecuaciones del modelo matemático del motor de corriente directa.

En la sección de desarrollo se exponen las ecuaciones de un controlador PID y la teoría de la lógica difusa, así como las relaciones que se han considerado para hacer una combinación del control PID convencional con el control difuso. Se expone la construcción del banco de pruebas, la descripción del hardware, así como la programación del algoritmo de control y su implementación en las pruebas, en esta misma sección se explica el desarrollo de los algoritmos de control en LabVIEW que es la plataforma de programación utilizada.

En la sección de resultados se exponen las pruebas realizadas mostrando las gráficas de la respuesta de la velocidad del MCD operando con el controlador PID difuso ante una entrada tipo escalón, así como el seguimiento de tres trayectorias de velocidad diferentes, se introducen tres perturbaciones desconocidas para cada tipo de entrada, así como para cada trayectoria de velocidad.

Por último, en la sección de conclusiones se realiza un análisis de los objetivos y las expectativas cumplidas con la metodología de estudio desarrollada a lo largo de este trabajo, se analizan las ventajas y desventajas de un control PID clásico contra un control PID difuso y se mencionan aspectos para ser considerados en trabajos futuros.

2. Fundamentación teórica

2.1. Revisión del estado del arte

Actualmente se han desarrollado trabajos, tanto de diseño y simulación como de implementación, en los que se aborda el control PID en combinación con diferentes estrategias o modificaciones en el esquema de control convencional en mención, con el fin de lograr un mejor desempeño que el que ofrecen los controladores PID clásicos. Muchas de estas estrategias se abordan desde un punto de vista de optimización y basan su funcionamiento en la aplicación de la lógica difusa. Particularmente los controladores basados en lógica difusa han tomado especial interés en el ámbito del control automático ya que en algunos casos ofrecen buenos resultados en el seguimiento y regulación. No obstante, los controladores difusos aun presentan desventajas en su implementación y desempeño que los hacen objeto de estudio y mejoramiento. En el caso particular de los sistemas lineales como el motor de corriente directa (MCD) el diseño e implementación de controladores se centra en la regulación de la velocidad angular y la posición del eje del motor y, en algunos casos, del par mecánico.

En (Pantoja, 2013) se presenta el diseño e implementación de controladores PID clásicos y difusos, ambos por separado, con el fin de realizar un análisis comparativo del desempeño de ambos esquemas de control aplicado al control de velocidad de un motor de corriente directa, el diseño de los esquemas de control y su análisis en simulación se realiza mediante MATLAB Simulink, la implementación se lleva a cabo mediante una FPGA Spartan 3E y Simulink como plataforma de interfaz gráfica para el monitoreo de la respuesta dinámica del motor. Muestra resultados aceptables en simulación, sin embargo, en el aspecto de la implementación se usa un tiempo de muestreo grande debido a que la comunicación entre la FPGA y Simulink es lenta. Adicionalmente el costo del sistema se eleva al utilizar la FPGA como tarjeta de adquisición de datos.

Otros trabajos como el de (Rairán, 2010) presenta el diseño e implementación de un controlador PID difuso para el control de posición de un motor de corriente directa,

mediante una plataforma comercial que utiliza una tarjeta de adquisición de datos PCI6024E de la marca National Instruments. Los resultados muestran un seguimiento de la posición con buen desempeño. No obstante, la plataforma no es de bajo costo, aunque sí bastante eficaz y con una velocidad de muestreo que permite observar claramente la respuesta dinámica del motor.

En (Ruiz, 2007) se realizó un análisis comparativo de controladores PID clásicos y controladores difusos aplicados al control de nivel de líquido en un sistema de tanques acoplados, el análisis muestra la sintonización de los controladores PID. En los resultados se presentan las gráficas de regulación del nivel del líquido usando ambos controladores, sin embargo, no se tienen resultados en implementación.

Trabajos como el de (Gualdrón, 2015), se enfocan al diseño e implementación de controladores PID autosintonizados mediante técnicas de lógica difusa aplicado al control de posición de un péndulo invertido, cabe resaltar que tanto en (Ruiz, 2007) y en (Gualdrón, 2015) el control se realizó sobre sistemas no lineales. Aunque en (Gualdrón, 2015) se presenta, además, la identificación del sistema, sus resultados en el control de posición tienen buen comportamiento dinámico, la implementación se realiza mediante un prototipo experimental diseñado en el laboratorio y una tarjeta de adquisición de datos NI-USB-6210 de National Instruments con buena velocidad de adquisición pero un costo un tanto elevado en comparación con el propuesto en este trabajo.

En (Rodríguez, 2004), (Lemus, 2004) y (Espítia, 2010) se presentó diseño de controladores difusos para sistemas no - lineales enfocándose en los sistemas de estabilización bola-viga y otros, mostrando las bondades que el control difuso tiene frente al control PID clásico. El punto de inmunidad al ruido y el rechazo a perturbaciones se menciona en algunos de los trabajos relacionados con estrategias de control difusas en combinación o no con esquemas de control clásicos, sin embargo, en ninguno de estos ha sido el tema central ni caso de estudio.

En (Vera, 2010) se presentó el diseño y simulación de un controlador difuso aplicado al control de velocidad de un motor de inducción trifásico, se hace presente el uso de un modelo matemático para un inversor trifásico y explica como el controlador modifica las señales encargadas de generar los correspondientes estados de conmutación en el convertidor electrónico de potencia en mención. Así como en trabajos mencionados anteriormente, se hacen evidentes las ventajas que un sistema de control difuso presenta como una respuesta con menor sobreimpulso ante cambios en la referencia y la posibilidad de modificar las reglas de control que rigen el funcionamiento y desempeño de todo el sistema. En este caso la implementación no se realizó.

Por otro lado, el control PID clásico se utiliza ampliamente en la industria y la academia dada su versatilidad de adaptación y facilidad de ajuste paramétrico, buen desempeño (dependiendo la planta a controlar) así como una implementación que no exige un procesamiento matemático complicado. En algunos casos, bajo una sintonización en regulación o autosintonización en seguimiento, en ambas muy precisas, muestra robustez ante el rechazo a perturbaciones.

Trabajos como (Gavran, 2016), (Samikshya, 2017) y (Vikhe, 2014) se enfocan en el estudio, sintonización y simulación de sistemas de control que incluyen controladores PID clásicos con técnicas de optimización y mejoramiento del desempeño, el caso particular de (Gavran, 2016) y (Vikhe, 2014) es que muestran resultados en

implementación, el primero utilizando una tarjeta Arduino Uno como tarjeta de adquisición de datos y Matlab Simulink como plataforma de desarrollo para el controlador e interfaz gráfica de monitoreo, el segundo utiliza a la tarjeta Arduino Uno como tarjeta de adquisición y LabVIEW como plataforma de desarrollo del algoritmo de control y monitoreo, aunque en (Gavran, 2016) se muestran los resultados de regulación para una referencia constante o entrada tipo escalón no se incluye una modificación en el esquema de control clásico, para (Vikhe, 2014) se muestra la implementación de un controlador PID clásico sin presentar las gráficas de resultados para el control de velocidad. En estos dos casos no se realizan pruebas considerando perturbaciones mecánicas o eléctricas conocidas o desconocidas.

En (Mohamed, 2017), (Berahim, 2014) y (Khairul, 2014) se desarrollan trabajos de simulación e implementación de controladores PID aplicados al control de velocidad en motores de corriente directa desde el enfoque de utilización de hardware libre y de bajo costo, de estos trabajos destaca el control por voltaje de armadura en (Mohamed, 2017) y por corriente de armadura en (Berahim, 2014), el uso de Matlab Simulink y LabVIEW como plataformas para el desarrollo de los algoritmos de programación, manipulación de parámetros del controlador así como el monitoreo de las señales adquiridas y generadas. Particularmente en (Berahim, 2014) se realiza un análisis comparativo entre el desempeño de un controlador PID clásico y un controlador difuso, no obstante, el control difuso se desarrolla en simulación sin ser llevado a la implementación.

Persisten desventajas como las que se observan en (Bista, 2016) en donde se realiza la implementación y sintonización heurística de un control PID aplicado a la regulación de temperatura de un sistema de enfriamiento por disipadores de aletas. El algoritmo de control se programa en lenguaje C para Arduino, se utiliza la tarjeta Arduino Uno para la adquisición de los datos y el monitoreo de los mismos se realiza en Matlab Simulink. Se presentan resultados de regulación con baja inmunidad al ruido y rechazo a perturbaciones, principalmente por la naturaleza de la variable controlada y la técnica de sintonización utilizada.

Otro ejemplo similar al anterior es el mostrado en (Chan, 2007) en el que se realiza la implementación de un controlador PID para la regulación de temperatura en un banco de pruebas, en este caso se demuestra que el controlador puede, en cierto grado, tener rechazo a perturbaciones, pero no con buen desempeño, adicionalmente la implementación se realiza con una FPGA lo que incrementa el costo del banco de pruebas.

Trabajos como (Allam, 2016) y (Sáenz, 2013) realizan diseño e implementación de controladores PID clásicos aplicados al control de velocidad en un motor de corriente directa utilizando hardware libre y de bajo costo, destaca el uso de Matlab Simulink y LabVIEW para la programación de los algoritmos.

Al presentar los resultados del seguimiento de velocidad se observa un desempeño considerablemente mejor al de otros trabajos. No se aborda el caso de estudio del rechazo a perturbaciones, sin embargo, los resultados ofrecen la posibilidad de que la programación realizada por (Sáenz, 2013) posea un grado aceptable de rechazo a perturbaciones.

2.2. Modelo matemático del MCD

En esta sección se describirá la planta a controlar que corresponde a un motor de corriente directa o MCD (ver figura 1) de

imán permanente. Sin embargo, las ecuaciones diferenciales que describen su comportamiento dinámico pueden aproximarse sin mayor dificultad a las de cualquier motor de CD con imán o con devanado de campo.



Figura 1. Motor de corriente directa utilizado. Fuente: (Allam, 2016).

Para realizar el análisis se usará el circuito equivalente mostrado en la figura 2 en que se aplica un análisis de mallas utilizando ley de Ohm y Leyes de voltaje de Kirchhoff como se hizo en (Cerón, Vazquez, & Aquino, 2011).

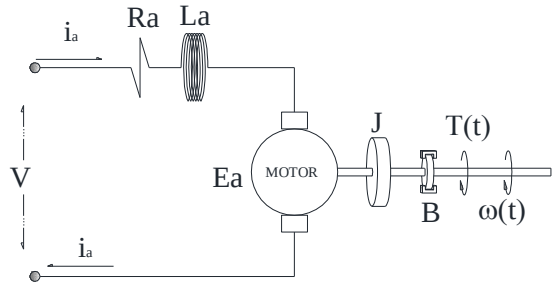


Figura 2. Circuito equivalente del motor de corriente directa. Fuente: *Autoría propia.*

De este circuito equivalente, se obtienen las ecuaciones para la velocidad desarrollada incluyendo el efecto de los parámetros mecánicos en el MCD. En este trabajo se utilizó un motor de imán permanente por lo que le efecto del devanado de campo no se considera para este estudio.

$$v(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + E_a(t) \qquad \textbf{(Ec. 1)}$$

$$T_m(t) = J\frac{d\omega(t)}{dt} + B\omega(t) \qquad \textbf{(Ec. 2)}$$

La ecuación (1) representa el modelado puramente eléctrico mientras que la ecuación (2) representa el modelado de la parte mecánica. Donde $v(t)$ es el voltaje de excitación, R_a es la resistencia interna del devanado de armadura, $i_a(t)$ es la corriente del devanado de armadura, L_a es la inductancia del devanado de armadura, T_m es el par mecánico de carga, J es el momento de inercia rotacional del motor, B es el coeficiente de fricción viscosa en los rodamientos del motor, $\omega(t)$ es la velocidad angular del motor y E_a (t) es el voltaje contraelectromotriz.

Con el fin de obtener la ecuación diferencial que gobierna al MCD se plantean las ecuaciones (3) y (4), usando la transformada de Laplace es posible obtener una función de transferencia que represente la respuesta dinámica de la velocidad.

$$E_a(t) = K_a\omega(t) \qquad \textbf{(Ec. 3)}$$

$$T_m(t) = K_m i_a(t) \qquad \textbf{(Ec. 4)}$$

El parámetro K_m es la constante mecánica de construcción del motor y K_a es la constante eléctrica interna del motor, cabe señalar que dependen de la construcción del mismo. Aplicando la transformada de Laplace a las ecuaciones (1), (2), (3) y (4) se obtiene:

$$V(s) = R_a I_a(s) + L_a s I_a(s) + E_a(s) \qquad \textbf{(Ec. 5)}$$

$$T_m(s) = Js\Omega(s) + B\Omega(s) \qquad \textbf{(Ec. 6)}$$

$$E_a(s) = K_a\Omega(s) \qquad \textbf{(Ec. 7)}$$

$$T_m(s) = K_m I_a(s) \qquad \textbf{(Ec. 8)}$$

Donde $\Omega(s)$ es la velocidad angular e $I_a(s)$ es la corriente de armadura. Sustituyendo ecuación (8) en ecuación (6) y ecuación (7) en ecuación (5) y realizando las factorizaciones y simplificaciones adecuadas en (9) y (10) se obtiene una sola ecuación con la cual es posible obtener la función de transferencia mostrada en la ecuación (11).

$$V(s) = \frac{(R_a + L_a s)T_m(s)}{K_m} + K_a\Omega(s) \quad \textbf{(Ec. 9)}$$

$$V(s) = \frac{(R_a + L_a s)T_m(s)}{K_m} + \frac{K_a T_m(s)}{Js + B} \quad \textbf{(Ec. 10)}$$

$$\frac{\Omega(s)}{V(s)} = \frac{K_m}{L_a J s^2 + (R_a J + L_a B)s + R_a B + K_m K_a} \quad \textbf{(Ec. 11)}$$

La función de transferencia del MCD que se expresa en (11) es un modelo matemático en el dominio de la frecuencia y que puede ser utilizado para obtener la respuesta dinámica en lazo abierto mediante simulación.

En la tabla 1, se puede observar lo valores de los parámetros empleados para el desarrollo de este trabajo.

Tabla 1. Valores de los parámetros considerados (Cerón, Vazquez, & Aquino, 2011).

Símbolo	Valor	Unidades
***v*(*t*)**	12	[*V*]
***R*$_a$**	4.1884	[Ω]
***i*$_a$(*t*)**	180x10^{-3}	[*A*]
***L*$_a$**	0.590001	[*Hy*]
***E*$_a$(*t*)**	0.4	[*V*]
***T*$_m$(*t*)**	0	[*Nm*]
J	8.4x10^{-5}	[*Nm/seg*^{2}]
***ω*(*t*)**	0-1000	[*RPM*]
B	6.8x10^{-5}	[*Nm/rad/seg*]
***K*$_m$**	53.68 x10^{-3}	[*Adimensional*]
***K*$_a$**	33.05 x10^{-3}	[*Adimensional*]

3. Desarrollo

3.1. Control PID Clásico

La teoría del control PID clásica establece que la ecuación que determina la ley de control para este tipo de controladores es:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \quad \textbf{(Ec. 12)}$$

Donde K_p, K_i y K_d son la ganancia proporcional, ganancia integral y ganancia derivativa respectivamente, $e(t)$ es el error que se procesa en el controlador producto de la resta entre la referencia y la variable retroalimentada. De otra forma esta misma ecuación puede expresarse en términos de los parámetros de un controlador PID como:

$$u(t) = K_p e(t) + K_p \frac{1}{T_i}\int_0^t e(t)dt + K_p T_d \frac{de(t)}{dt} \quad \textbf{(Ec. 13)}$$

Donde T_i y T_d son los tiempos integral y derivativo respectivamente que influyen y determinan la respuesta dinámica que el controlador otorga a la planta bajo el desempeño de este. Con base en la ecuación (13) se establece que la representación algorítmica de un controlador PID digital puede expresarse de la forma mostrada en (14):

$$u(k) = K_p e(k) + \frac{K_p T}{T_i}\sum_k^{n-1} e(k) + \ldots + \frac{K_p T_d}{T}\left(e(k) - e(k-1)\right) \quad \textbf{(Ec. 14)}$$

Donde el parámetro T es el periodo de muestreo, inherente a todo sistema que discretiza señales de naturaleza analógica. El término "k" va desde k = 0,1,2,3, hasta (n-1) datos muestreados en el algoritmo de control. El caso de los controladores PID que se implementan dentro de dispositivos programables requieren de la programación de la ecuación (14) para llevar a cabo las acciones de control sobre la planta.

La ecuación (14) es programada dentro del entorno de LabVIEW para obtener así la acción correspondiente al controlador PID digital que se ha utilizado para el algoritmo de control principal. Una multiplicación del error actual y un parámetro constante del controlador representa a la acción proporcional, una sumatoria recursiva de

errores presentes multiplicados por parámetros internos del controlador en cada iteración representa a la acción integral y una resta recursiva entre el error presente y anterior multiplicada por parámetros internos del controlador representa la acción derivativa. La ecuación (14) sintetiza en una expresión matemática lo anterior.

3.2. Control PID Difuso

La utilización de lógica difusa en un algoritmo de control del PID digitalizado conforma al controlador PID digital difuso que se ha implementado en este trabajo. Mediante un conjunto de reglas y criterios establecidos previamente y una herramienta matemática de inferencia estadística el controlador se encarga de otorgar seguimiento de las trayectorias de velocidad impuestas al MCD, esto se explica a continuación. La implementación de este controlador se realiza mediante el banco de pruebas didáctico del cual se explica su construcción más adelante.

El control difuso recibe este nombre por basar su operatividad en la lógica difusa, centra su funcionamiento en la utilización de tres etapas básicas dentro del esquema; la fuzzificación del sistema, un método de inferencia estadístico y una etapa de defuzzificación (Zavala, 2010) que se explican a continuación.

La fuzzificación del sistema consiste en definir cuáles serán las variables de entrada, en este caso se tiene solo una, el error producido por la diferencia entre la referencia de velocidad y la velocidad real medida, y de esta manera asignar a esta entrada rangos de validez mediante un conjunto de funciones llamadas funciones de pertenencia o de membresía. Cada función de pertenencia está definida mediante términos llamados "variables lingüísticas", cuando un determinado valor de la variable de entrada recae sobre alguna de las funciones de pertenencia se le asigna entonces el rango del término lingüístico correspondiente.

Una vez que la variable de entrada tiene los rangos de pertenencia asignados y por lo tanto los términos lingüísticos, entran en juego los algoritmos de inferencia, estos se basan en un conjunto de reglas programadas llamadas "Reglas difusas". Estas reglas determinarán qué acción de control se debe obtener del controlador. La o las variables de salida también tienen que ser fuzzificadas, por tanto, se realiza sobre estas el mismo proceso descrito anteriormente. En este caso la variable de salida es la señal de control, físicamente esta señal de control es un valor de modulación de ancho de pulso o PWM (Pulse Width Modulation) que regula el voltaje inyectado al MCD y por tanto la velocidad que desarrolla.

Por último, es necesario que los valores obtenidos del proceso de inferencia mediante las reglas difusas se escalen y conviertan en valores adecuados para otorgar la o las señales de control de salida, este proceso es el que recibe el nombre de defuzzificación del sistema.

3.3. Hardware Utilizado

En este caso se eligió a la tarjeta de desarrollo Arduino MEGA ya que presenta características que lo hicieron idóneo en comparación con otros modelos de la familia de Arduino tal y como una comunicación estable y sencilla en el envío de datos al sistema de cómputo para la adquisición y generación de señales de control, otras de sus ventajas son la de poseer una velocidad de procesamiento suficiente, periféricos de entrada analógicos (con una resolución adecuada), entradas-salidas digitales suficientes para el control del MCD y su bajo

costo. En la tabla 2, se describen las características de la tarjeta Arduino MEGA.

Otros modelos de tarjeta Arduino poseen las mismas características, sin embargo, son más limitado en puertos de entrada/salida y su costo no es significativamente menor al del Arduino MEGA por lo que sería más complicado escalar el sistema para futuras implementaciones de control en otras plantas.

Se entiende que hay opciones en el mercado a bajo costo que podrían igualar o superar las capacidades del microcontrolador Atmega2560 del Arduino MEGA. Sin embargo, el método propuesto puede ser implementado en otro tipo de plataforma y dispositivos programables y no se cierra a utilizar únicamente la tarjeta Arduino en mención.

Esta tarjeta programable forma parte de la gama de dispositivos fabricados bajo la filosofía de hardware libre lo que los hace sencillos en su programación, bajo costo y arquitectura flexible sin sacrificar desempeño y robustez al momento de establecer parámetros de velocidades de procesamiento, resoluciones y capacidad de memoria.

1.1. Algoritmo de Control

A continuación, se detallan las partes que conforman el algoritmo para el control PID difuso desarrollado en LabVIEW. En la figura 3 se muestra el panel frontal del programa desarrollado y en la figura 4 se muestra el algoritmo del control PID difuso. La primera parte que debe establecerse en el programa es la comunicación con la tarjeta de adquisición, en este caso el Arduino MEGA, posteriormente los bloques que se encargan de adquirir la señal analógica del generador de CD (sensado de velocidad) así como de la generación del PWM que regula el voltaje de alimentación al MCD. La figura 4 muestra esta parte del algoritmo.

Tabla 2. Caracteristicas de la tarjeta arduino Mega.

Microcontrolador:	ATmega2560 (8-bit)	**Voltaje de operación**	5V
Voltaje entrada:	7-12V	**Rango de voltaje: entrada:**	6-20V
Pines digitales I/O:	54 (15 PWM output)	**Pines Analógicos:**	16 (10-bit resolution)
Pines de I/O (corriente):	40 mA	**Pines de 3.3V:**	50 mA
Memoria Flash:	256 KB	**SRAM:**	8 KB
EEPROM:	4 KB	**Velocidad del reloj:**	16 MHz

Como siguiente paso se tiene la programación del controlador PID digital. En esta parte se realiza en función de la ecuación (14) en la que se consideran tanto los términos de sumatoria y diferencia, así como los parámetros del controlador. En la figura 4 se muestra la sección en donde se efectúa el cálculo de la acción de control PID.

Los parámetros del controlador PID se obtuvieron de forma heurística, no se descarta para trabajos futuros el mejoramiento del desempeño del controlador mediante la utilización de una técnica de sintonización o del uso de un sistema autosintonizado

proponiendo la implementación del mismo control difuso para este fin.

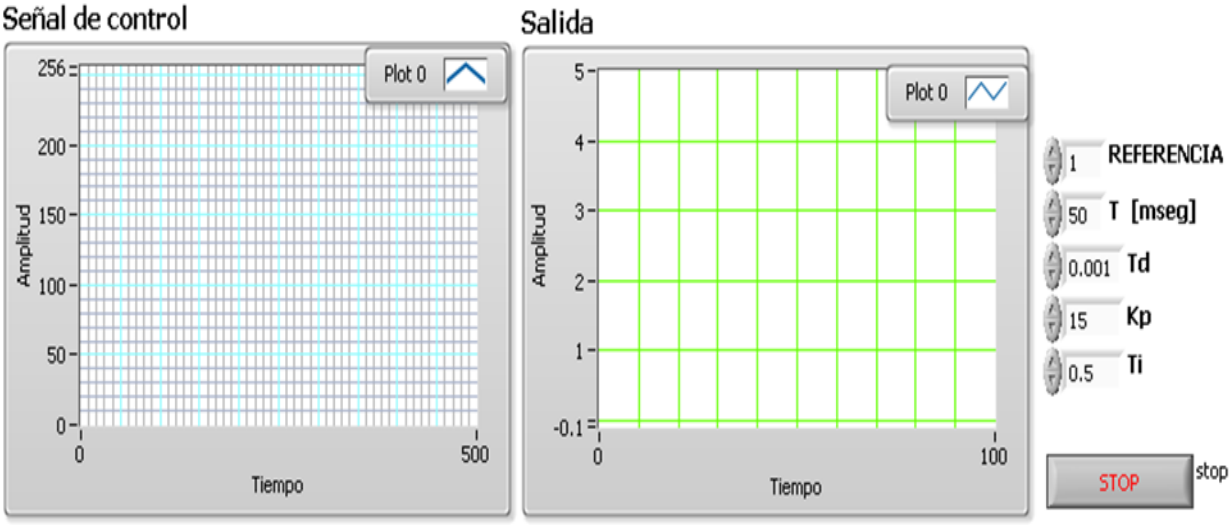


Figura 3. Panel frontal de la interfaz gráfica de usuario desarrollada del Control PID difuso. Fuente: Autoría propia.

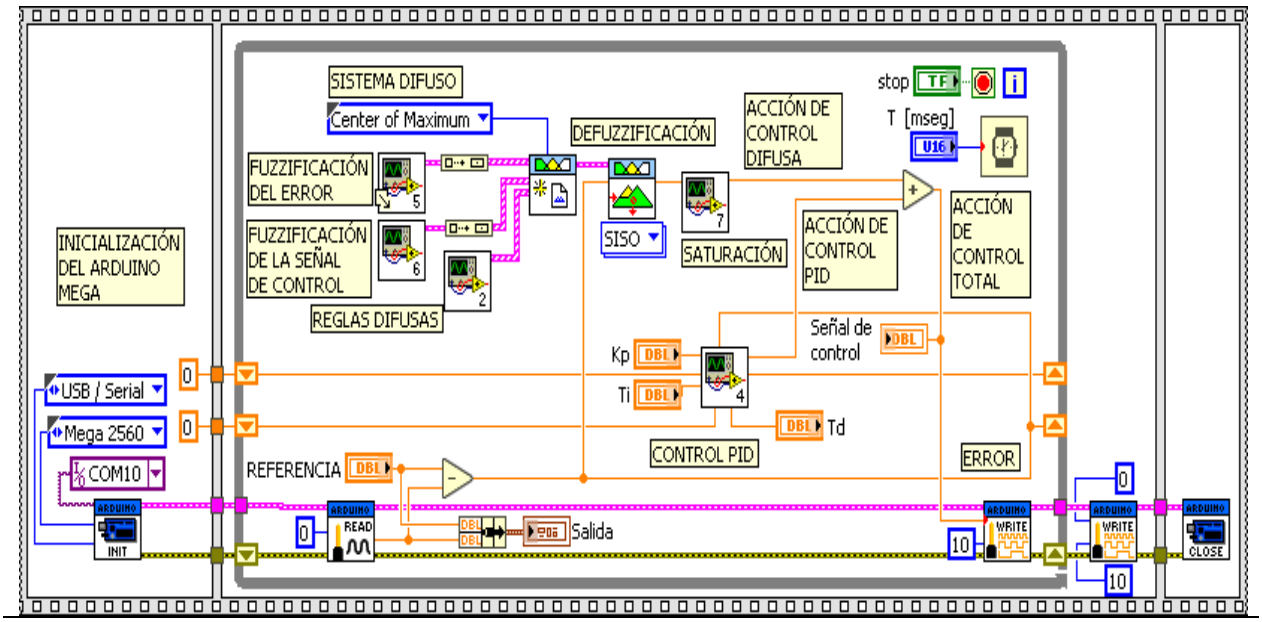


Figura 4. Control PID Difuso (programación en LabVIEW). Fuente: Autoría propia.

Una vez que se ha calculado la acción de control PID se programa la parte correspondiente al control difuso.

Los bloques usados en la figura 4 (para la sección del sistema difuso) corresponden a la herramienta de *Fuzzy Logic* utilizada para introducir a las funciones de pertenencia y el tipo de las mismas, los rangos de validez para las variables de entrada y de salida, así como definir el método de inferencia y, finalmente, realizar la fuzzificación y defuzzificación del sistema.

El método de inferencia que se utilizó es el de Mamdani y el método de defuzzificación

seleccionado para el control es el del centroide máximo. El funcionamiento consiste en generar un error de velocidad que es procesado como entrada tanto por el control PID clásico, como por el control difuso.

Ambos controles generan una señal de control que es sumada y enviada al Arduino MEGA para que esta, a su vez, genere la señal PWM requerida para dotar al motor del voltaje y por tanto de la velocidad que debe desarrollar durante el seguimiento de la referencia.

Una vez obtenida la acción de control difusa y el control PID programado en LabVIEW, se suman para obtener la acción de control total, que se define como controlador PID difuso, cuyo resultado es conducido hacia el bloque de generación de señal PWM proporcionada al MCD por la tarjeta Arduino.

1.2. Banco de Pruebas

El banco de pruebas utilizado se muestra en las figuras 5 y 6.

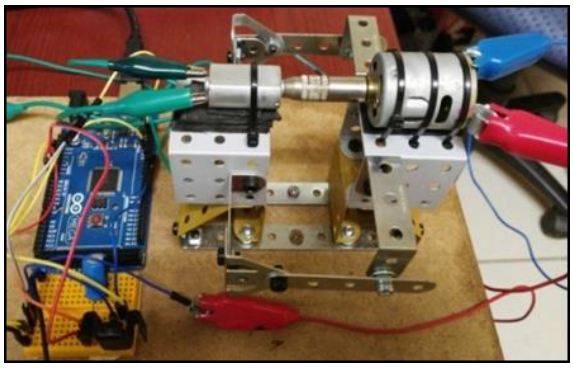


Figura 5. Banco de pruebas (lado izquierdo). *Fuente: Autoría propia.*

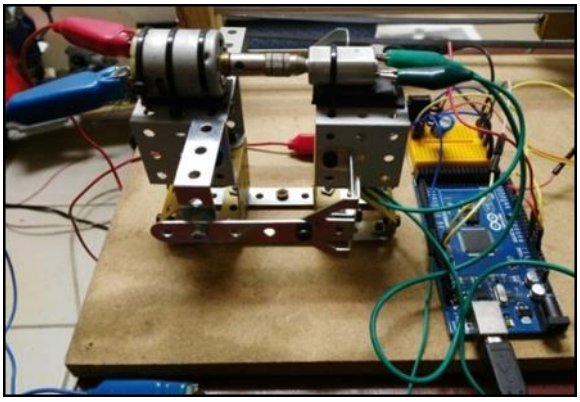


Figura 6. Banco de pruebas (lado derecho). *Fuente: Autoría propia.*

La tabla 3, muestra un listado de materiales y costos de cada componente del banco de pruebas en la que se puede observar que es más económico que otros esquemas de pruebas que involucran el uso de tarjetas de adquisición más costosas y que cumplen el mismo propósito bajo el mismo desempeño como se muestra en (Pantoja, 2013), (Rairán, 2010) y (Gualdrón, 2015) en las que el costo de solo la tarjeta de adquisición utilizada supera los 2,000 pesos mexicanos.

El generador de CD es utilizado para que, acoplado al eje del motor, proporcione un voltaje de salida equivalente a la velocidad angular del MCD y de esta forma se obtenga una medición de la velocidad que pueda ser procesada por el sistema de adquisición.

Tabla 3. Materiales y costos del banco de pruebas.

No. parte	Descripción	Cantidad	Costo
1	Tarjeta Arduino MEGA	1	$600.00
2	Motor de CD	1	$100.00
3	Materiales de montaje		$200.00
4	Transistor TIP41 NPN	1	$15.00
5	Disipador de calor	1	$20.00
6	Placa para circuito impreso	1	$50.00
7	Generador de CD	1	$85.00
8	Capacitores, resistencias	5	$40.00
9	Pieza de acoplamiento mecánico	1	$100.00
10	Cable de comunicación serial USB	1	$45.00
11	Fuente de alimentación a 12 volts	1	$280.00
12	Cables para conexiones		$52.00
	TOTAL		$1,587.00

Aprovechando las características de linealidad del generador de CD se realizó una caracterización de la salida de voltaje de este con respecto a la velocidad de giro que se tiene en el eje acoplado al MCD, obteniendo una salida del generador de 100 milivolts por cada 50 revoluciones por minuto del motor. En la figura 7 se muestra el circuito de la etapa de potencia utilizada para alimentar al MCD, se utilizó una fuente comercial de 12 volts y 2 amperes y un transistor NPN TIP41 con un disipador de tamaño reducido.

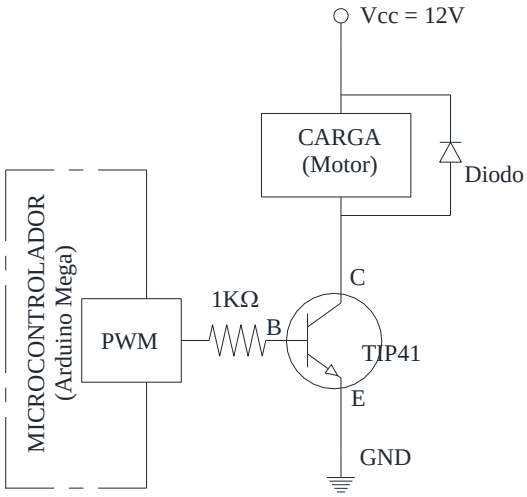


Figura 7. Etapa de potencia para el MCD. Fuente: Autoría propia

Cabe señalar que se genera la señal de control PWM proveniente del Arduino MEGA con una frecuencia de 490 Hz, esta es la frecuencia que por defecto está configurada en el pin de salida, sin embargo, no es un requisito indispensable para el control que este valor de frecuencia sea el mismo siempre, únicamente se debe considerar las características de la etapa de potencia sobre todo por la frecuencia de conmutación permisible en los dispositivos semiconductores utilizados.

En la figura 8 se muestra un diagrama de bloques del banco de pruebas. Para obtener una salida de voltaje del generador más libre de ruido externo, se observó que podía ser filtrada a través de un capacitor electrolítico conectado en paralelo con las terminales de salida, el valor del capacitor se estableció en 1000 microfaradios.

La dinámica que el generador de CD que puede introducir en la respuesta del MCD no se contempla al no influir de manera significativa con base en las pruebas de caracterización de este como sensor, no obstante, su efecto no se descarta y se toma como un punto de análisis para compensar este efecto en trabajos futuros.

2. Resultados

En esta sección se describen los resultados obtenidos para el controlador PID difuso que se ha implementado a través del banco de pruebas mostrado anteriormente. Primeramente, se muestran las gráficas de seguimiento de velocidad, las gráficas del error de velocidad, así como el comportamiento de la señal de control PID difuso durante la prueba respectiva. Se consideran cuatro trayectorias de velocidad con distinto perfil dinámico con el fin de observar el desempeño y dinámica del sistema ante distintos perfiles de velocidad.

La prueba realizada para el seguimiento de la trayectoria 1 mostrada en la figura 9, tiene una duración de 30 segundos tiempo durante el cual no se aplican perturbaciones, se observan sobreimpulsos en los instantes donde la velocidad de referencia cambia bruscamente, no obstante, el controlador recupera rápidamente la velocidad deseada con un tiempo de estabilización de 3 segundos. El tiempo de asentamiento de la velocidad real alcanza los 5 segundos en algunos puntos del seguimiento, sin embargo, el error muestra que después del transitorio más relevante las oscilaciones se atenúan rápidamente (ver figura 9).

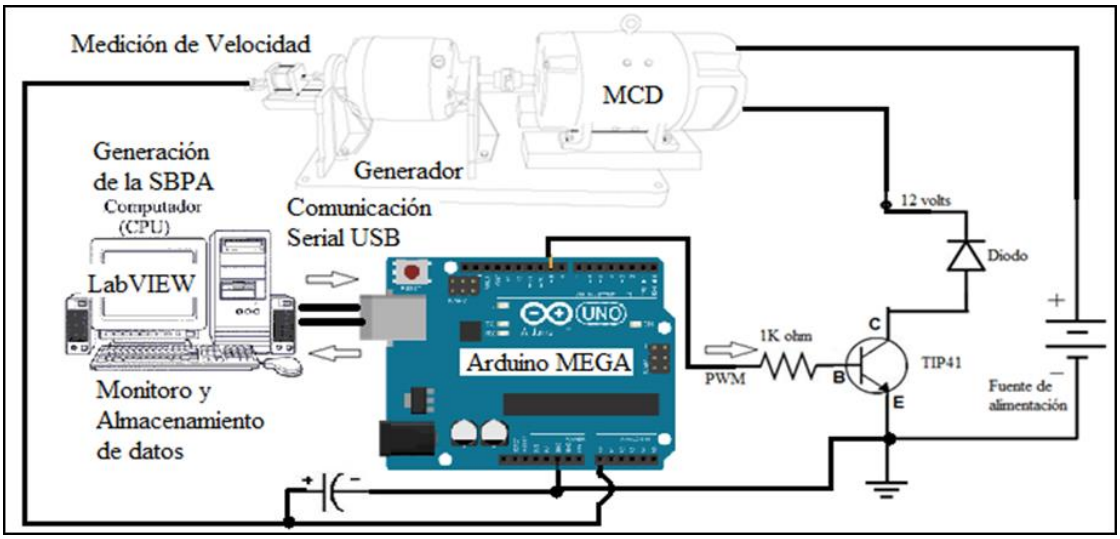


Figura 8. Diagrama de funcionamiento del Banco de Pruebas. *Fuente: Autoría propia.*

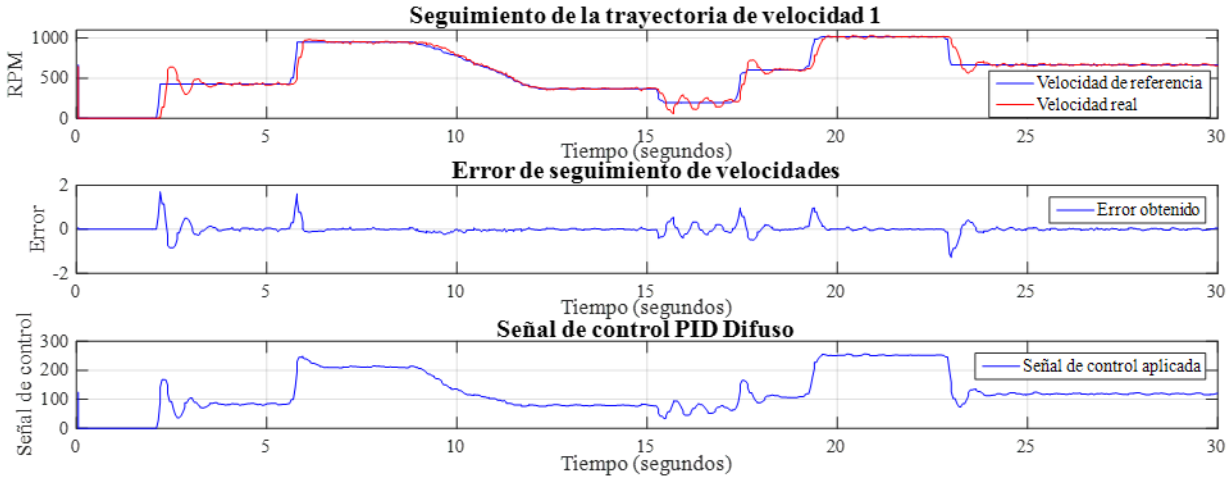


Figura 9. Seguimiento de la Trayectoria 2 (Azul: Velocidad de referencia; Roja: Velocidad real). Fuente: Autoría propia.

La segunda prueba de seguimiento consiste en utilizar un segundo perfil diferente al anterior y, de la misma forma que en la primera prueba, observar el comportamiento dinámico de la velocidad, el error de velocidad y la señal de control generada. Tal como se observa en la figura 10 los instantes de mayor sobreimpulso son a los 2.3 y 8 segundos, el tiempo de asentamiento oscila entre los 2 a 3.5 segundos.

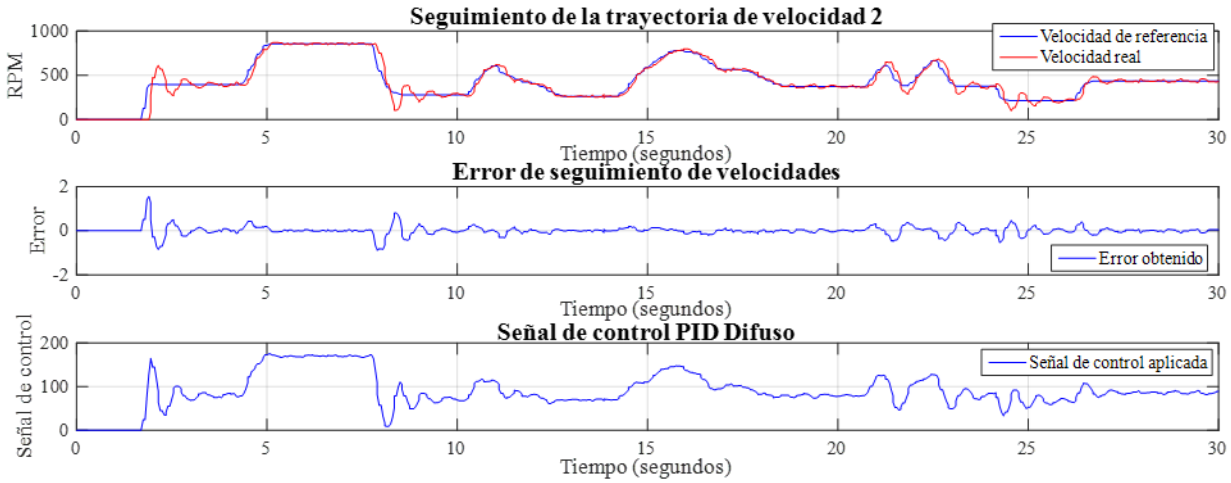


Figura 10. Seguimiento de la Trayectoria 2 (Azul: Velocidad de referencia; Roja: Velocidad real). Fuente: Autoría propia.

La tercera prueba consiste en generar una tercera trayectoria de velocidad distinta a las dos anteriores y observar la respuesta dinámica tal como se hizo en las dos previas. En la figura 11, se observa que persisten los sobreimpulsos en los instantes de cambio abrupto de referencia (2.3, 4.9, 15.8 y 23.8 segundos) no obstante la recuperación del seguimiento no toma más de 3 segundos en los instantes de sobreimpulsos más notorios. En la misma figura se observa que el error oscila en torno a cero (comportamiento deseado) salvo los instantes donde se presentan los sobreimpulsos de seguimiento.

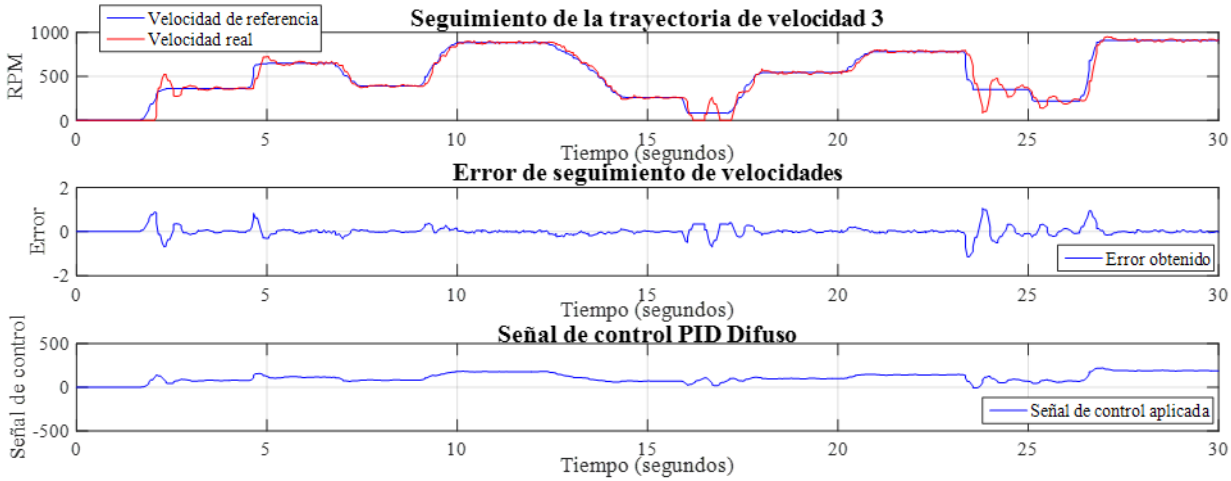


Figura 11. Seguimiento de la Trayectoria 3 (Azul: Velocidad de referencia; Roja: Velocidad real). Fuente: Autoría propia.

Con el fin de llevar al MCD a cambios de referencia un tanto más abrupto se genera la cuarta trayectoria en la que se observan sobreimpulsos de amplitud mayor que en las tres pruebas anteriores, sin embargo, el tiempo de asentamiento continua entre los 2 a 3.5 segundos en los instantes de mayor sobreimpulso (ver figura 12).

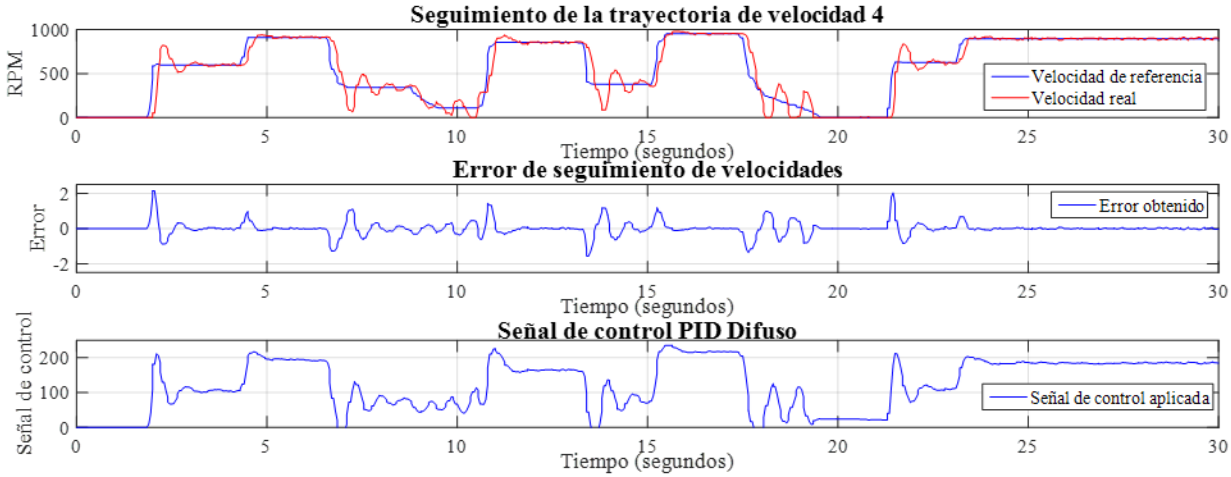


Figura 12. Seguimiento de la Trayectoria 4 (Azul: Velocidad de referencia; Roja: Velocidad real). *Fuente: Autoría propia.*

En los instantes donde el seguimiento no presenta sobreimpulsos el error de seguimiento se conserva en torno a cero. La señal de control muestra una respuesta de reacción ante los cambios en la referencia con el fin de dotar al MCD del voltaje con el que desarrolle más o menos velocidad y con ellos lograr el seguimiento de la trayectoria.

El segundo escenario de pruebas consistió en generar las cuatro trayectorias de seguimiento presentadas en las gráficas anteriores aplicando perturbaciones desconocidas en el MCD. Las perturbaciones consisten, en este caso, en una serie de bloqueos mecánicos en el eje del motor, de tal forma que provoquen el frenado repentino del mismo y posteriormente la liberación abrupta del eje para dejarlo girar sin restricciones, con ello se observará el efecto en la respuesta de compensación del controlador PID difuso ante estas perturbaciones. La magnitud de las perturbaciones es desconocida y se aplican en instantes de tiempo aleatorios. Las gráficas obtenidas se pueden apreciar en las figuras siguientes.

En el caso de la figura 13 se aplica una perturbación constante durante el segundo 6 al 7. Se aplica una perturbación y liberación inmediata del eje en el segundo 12 y 14.5 así como en el segundo 24, 25.1 y 27. De esta forma se observa que el comportamiento del controlador ante los efectos de perturbación mostrados.



Figura 13. Seguimiento de la Trayectoria 1 con perturbaciones (Azul: Velocidad de referencia; Roja: Velocidad real). *Fuente: Autoría propia.*

En la figura 14 se presenta el seguimiento con perturbaciones para la trayectoria 2. Se aplican en los instantes 4, 6, 9.5, 12.5, 14.4, 16.2, 17.3, 20, 25.5 y 27.5 segundos, aplicando un bloqueo y liberación inmediata del eje del MCD.



Figura 14. Seguimiento de la Trayectoria 2 con perturbaciones (Azul: Velocidad de referencia; Roja: Velocidad real). Fuente: Autoría propia.

En la figura 15 se presenta el seguimiento con perturbaciones para la trayectoria 3. Las perturbaciones se aplican en los instantes 4, 5.8, 10.8, 12.4, 15, 18, 20, 20.6, 22.2, 23.8 y 27. 5 segundos, aplicando un bloqueo y liberación inmediata del eje del MCD.

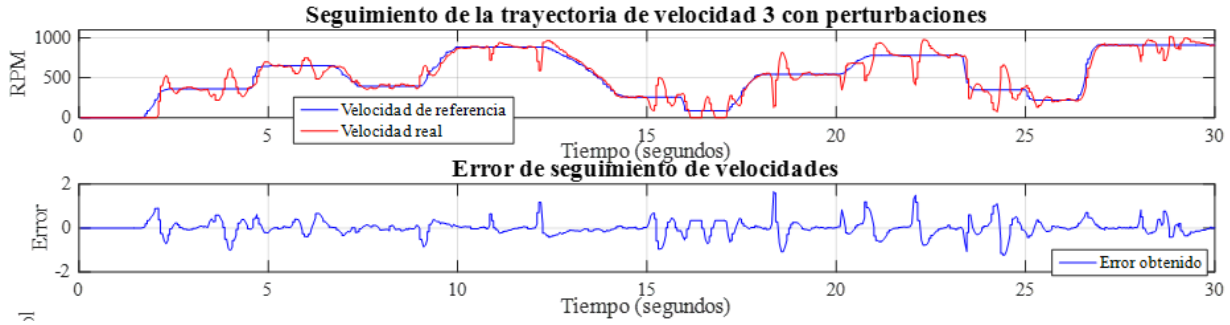


Figura 15. Seguimiento de la Trayectoria 3 con perturbaciones (Azul: Velocidad de referencia; Roja: Velocidad real). Fuente: Autoría propia.

En la figura 16 se muestran las perturbaciones para la trayectoria 4 aplicadas al MCD, con la característica de que se presentan con una frecuencia de ocurrencia más alta cada dos segundos, aplicando un bloqueo y liberación inmediata del eje del MCD.



Figura 16. Seguimiento de la Trayectoria 4 con perturbaciones (Azul: Velocidad de referencia; Roja: Velocidad real). *Fuente: Autoría propia.*

En los 8 casos presentados, se observa que las curvas de error en cada una de las trayectorias con y sin perturbación, son reducidas a 0 con ciertos sobretiros, lo que asegura que el seguimiento de la referencia por la acción de control PID difuso es cumplido.

Con el fin de cuantificar el comportamiento de los lazos de control, se utilizó un índice de desempeño basado en la señal de error $e(t)$, la cual es la diferencia entre el valor deseado de la variable controlada y su valor real (Arrieta & Alfaro, 2003). El índice de desempeño optimizado que se utilizo fue el ITAE el cual interpreta la minimización de la integral de la función de error en valor absoluto multiplicada por el tiempo, definido por (Ec. 15).

$$ITAE = \int t\left|e(t)\right|dt \qquad \textbf{(Ec. 15)}$$

En la figura 17 se resume los resultados obtenidos para el caso particular del controlador PID difuso optimizados bajo el índice ITAE ante un cambio escalón en el valor deseado (servomecanismo), y en la figura 18, las características de las respuestas a un cambio escalón ante una perturbación (regulador).

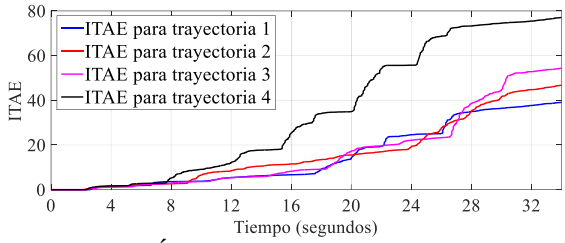


Figura 17. Índice de desempeño ITAE para las trayectorias sin perturbación. *Fuente: Autoría propia.*

Los resultados muestran claramente cómo existen diferencias entre el comportamiento dinámico que se obtiene cuando el controlador (PID difuso) trabaja bajo una perturbación.

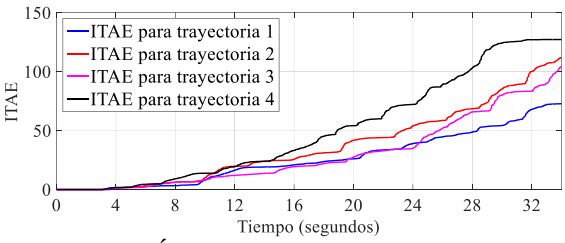


Figura 18. Índice de desempeño ITAE para las trayectorias con perturbación. *Fuente: Autoría propia.*

Como se observa, esta diferencia es más significativa en el caso del regulador (perturbaciones). Se puede apreciar que la

disminución en este índice, al emplear el modelo sin perturbaciones, el valor del ITAE llega apenas un 50 (respuestas promedio). Por lo tanto, con el controlador PID difuso, la reducción en el índice ITAE como servomecanismo (sin perturbación) varía entre un 35 y un 75 y en el caso del regulador (con perturbación) entre un 65 y un 125.

Como se puede observar en las figuras 17 y 18, las respuestas obtenidas con el controlador PID difuso sintonizado con el valor de error de un ITAE de 40 (sin perturbación) y 70 (con perturbación), son las que tienen, en todos los casos, el menor sobrepaso o error máximo, el menor tiempo al pico, el menor tiempo de asentamiento y el menor índice de error integral.

En resumen, estas son las mejores respuestas para el regulador (con perturbación) y para el servomecanismo (sin perturbación), por lo que se han destacado ligeramente sobre las demás.

Conclusiones

En esta sección se consolida la discusión acerca de los resultados obtenidos durante las pruebas del desempeño del controlador PID difuso aplicado al motor de corriente directa (MCD).

Se cuenta con un banco de pruebas para observar el desempeño de un controlador PID difuso que se programó utilizando LabVIEW versión 2015. Las cualidades de este banco de pruebas ofrecen versatilidad en la implementación y facilidad de uso al ser de propósito didáctico y poseer hardware de bajo costo como lo es la tarjeta de desarrollo Arduino MEGA utilizada para adquirir los datos y generar la señal de control. Cabe mencionar que estas cualidades no se pudieron observar en las referencias bibliográficas mencionadas en la revisión del estado de arte, además de que en los otros trabajos se aborda la implementación física de sistemas de control basados en PID e incluso PID difuso, pero no implementados sobre plataformas de bajo costo.

Con relación a las gráficas, las cuatro primeras pruebas de seguimiento que no incluyen aplicación de perturbaciones mecánicas muestran que el seguimiento de la referencia de velocidad se realiza adecuadamente en instantes donde no existe el cambio abrupto y donde se presentan cambios de referencia que persisten los sobreimpulsos, esto se muestra en las gráficas de la figura 9, 10, 11 y 12, no obstante como trabajo futuro se tiene la consigna de obtener los parámetros del MCD para realizar una sintonización del controlador PID que minimice los sobreimpulsos y mejore la respuesta dinámica ante cambios bruscos en la referencia, la utilización de subsistemas con lógica difusa para la autosintonización de controladores PID es una propuesta que se trabajará en investigaciones futuras.

Por otro lado, la aplicación de perturbaciones durante las pruebas de seguimiento de velocidad con el controlador PID difuso presentadas en las gráficas de las figuras 13, 14, 15 y 16 muestran que en instantes donde estas se presentan el controlador actúa compensando el efecto mismo con sobreimpulsos en las respuesta de la velocidad que duplican la amplitud del mismo valor de velocidad deseado (referencia) no obstante el seguimiento es recuperado en un tiempo de 2 a 4 segundos dependiendo la magnitud de la perturbación y el valor de referencia lo cual muestra que el rechazo a perturbaciones es posible aunque el desempeño del controlador se minimiza en comparación al comportamiento sin perturbación.

El efecto de la magnitud de los sobreimpulsos y el tiempo de asentamiento puede ser

mitigado el rediseño del algoritmo de control difuso considerando otras variables de entrada al sistema de control, este aspecto se verá en trabajos futuros que se enfoquen a la resolución de la problemática del rechazo a perturbaciones en el MCD utilizando técnicas de control difuso.

Como trabajo futuro, se podría exponer la escalabilidad del banco de pruebas desarrollado hacia plantas o procesos que necesiten limitar o corregir su salida, incluso a motores de mayor potencia en donde se consideraría los cambios en la etapa de potencia y los costos que implica, así como los parámetros característicos del motor. Y con ello, se demostraría que la metodología usada para controlar el MCD en este trabajo, se aplicaría a motores de alta potencia.

Agradecimientos

Se agradece el apoyo al Consejo Nacional de Ciencia y Tecnológica (CONACYT) y a la Universidad Tecnológica Emiliano Zapata del estado de Morelos por las facilidades otorgadas para la realización de este trabajo.

Referencias

1. Pantoja, W. (2013). *Análisis y Comparación de Controladores PID y PID Difuso en Lógica Reconfigurable.* Oaxaca: Tesis, Universidad del Itsmo, Campus Tehuantepec. Recuperado el 24 de Octubre del 2018 de la página de internet http://www.unistmo.edu.mx/~computacion/tesis/Tesis_Williams_Antonio_Pantoja_Laces.pdf.

2. Rairán, J. (2010). "Diseño de controladores de tipo proporcional integral derivativo (PID) y difuso para la posición de un motor de corriente directa". *Ingeniería y Universidad 14(1),* 137-160. Recuperado el 24 de Octubre del 2018 de la página de internet http://www.redalyc.org/articulo.oa?id=47715438007.

3. Ruiz, A. (2007). *Control difuso vs control PID: Análisis y simulación numérica.* Hidalgo: Tesis, Universidad Autónoma del Estado de Hidalgo. Recuperado el 24 de Octubre del 2018 de la página de internet https://repository.uaeh.edu.mx/bitstream/handle/123456789/10911?show=full.

4. Gualdrón, J. (2015). "Desarrollo de un controlador PID por sintonización difusa aplicado a un sistema rotacional (péndulo invertido)". *Revista Facultad de Ingeniería 4(40),* 95-105. Recuperado el 24 de Octubre del 2018 de la página de internet https://dialnet.unirioja.es/servlet/articulo?codigo=5265645.

5. Rodríguez, F. (2004). *Modelado y control PD-Difuso en tiempo real para el sistema barra-esfera.* México: Tesis, CINVESTAV-IPN. Recuperado el 24 de Octubre del 2018 de la página de internet https://www.ctrl.cinvestav.mx/~yuw/pdf/MaTesFOR.pdf.

6. Lemus, M. (2004). "Simulación de sistema difuso para el control de velocidad de un motor de cd." *Memorias del 3er Congreso de cómputo AGCOMP* (págs. 25-34). Morelos: Tecnológico Nacional de México campus Zacatepec, UAEM.

7. Espítia, H. (2010). "Diseño y simulación de un controlador difuso para un motor dc basado en relaciones boolenas". *Revista Visión Electrónica, 4(1),* 12-22. Recuperado el 24 de Octubre del 2018 de la página de internet https://dialnet.unirioja.es/descarga/articulo/4169233.pdf.

8. Vera, H. (2010). "Diseño y simulación de un controlador PD difuso para el control de la velocidad de un motor de inducción." *Revista Ingeniería & Desarrollo, 27*, 62-92. Recuperado el 24 de Octubre del 2018 de la página de internet http://www.redalyc.org/articulo.oa?id=85215207005.

9. Gavran, M. (2016). "PI controller for DC Motor Speed Realized with Arduino and Simulink." *40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO).* Croatia: Faculty of Electrical Engineering and Computer Science. Recuperado el 24 de Octubre del 2018 de la página de internet https://ieeexplore.ieee.org/document/7973669.

10. Samikshya, T. (2017). "Comparison of Speed Control of DC Motor using PID Controller and Various Optimization Techniques a Review." *International Journal of Engineering Science and Computing, 7.*, 13859-13864. Recuperado el 24 de Octubre del 2018 de la página de internet http://ijesc.org/upload/63c708644989cca6701bf58d48439c42.Comparison%20of%20Speed%20Control%20of%20DC%20Motor%20using%20PID%20Controller%20and%20Various%20Optimization%20Techniques%20a%20Review.pdf.

11. Vikhe, P. (2014). "Real Time DC Motor Speed Control using PID Controller in LabVIEW." *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering. 3(9),* 12162-12167. Recuperado el 24 de Octubre del 2018 de la página de internet http://www.rroij.com/open-access/real-time-dc-motor-speed-control-using-pidcontroller-in-labview.pdf.

12. Mohamed, N. (2017). "Implementation of a PID Control System on Microcontroller (DC Motor Case Study)." *International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE).* Sudan: IEEE. Recuperado el 24 de Octubre del 2018 de la página de internet https://ieeexplore.ieee.org/document/7866088.

13. Berahim, N. (2014). *Development of PID Voltage Control For DC Motor Using Arduino.* Malaysia: Tesis, Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia. Recuperado el 24 de Octubre del 2018 de la página de internet http://eprints.uthm.edu.my/6659/1/NAWI_BERAHIM.pdf.

14. Khairul, M. (2014). *Development of PID Courrent Control For DC Motor Using Arduino.* Malaysia: Tesis, Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia. Recuperado el 24 de Octubre del 2018 de la página de internet http://eprints.uthm.edu.my/6651/1/MOHD_KHAIRUL_AKLI_AB_GHANI.pdf.

15. Bista, D. (2016). *Understanding and Design of an Arduino-based PID Controller*. USA: Tesis, Virginia Commonwealth University. Recuperado el 24 de Octubre del 2018 de la página de internet https://scholarscompass.vcu.edu/cgi/viewcontent.cgi?article=5737&context=etd.

16. Chan, Y. F. (2007). "Design and Implementation of Modular FPGA-Based PID Controllers." *IEEE Transactions on Industrial Electronics, 54(4),* 1898-1908. Recuperado el 24 de Octubre del 2018 de la página de internet https://ieeexplore.ieee.org/document/4271566.

17. Allam, T. (2016). "Design of PID controller for DC Motor Speed Control Using Arduino Microcontroller." *International Research Journal of Engineering and Technology (IRJET), 3(9),* 791-794. Recuperado el 24 de Octubre del 2018 de la página de internet https://www.irjet.net/archives/V3/i9/IRJET-V3I9142.pdf.

18. Sáenz, M. (2013). *Control en cascada no lineal de un conjunto motor generador*. Costa Rica: Universidad de Costa Rica Facultad de Ingeniería Escuela de Ingeniería Eléctrica. Recuperado el 24 de Octubre del 2018 de la página de internet https://ruie.ucr.ac.cr/catalogo/Record/INII-CEDI-CD-20233.

19. Cerón, I., Vazquez, J., & Aquino, E. (2011). *Modelo matemático del motor de corriente directa.* Puebla: Universidad Politécnica de Puebla. Recuperado el 24 de Octubre del 2018 de la página de internet https://www.researchgate.net/profile/Isra_Cero/publication/313215329_Modelo_matematico_del_motor_de_corriente_directa/links/5892b56ea6fdcc1b4146c965/Modelo-matematico-del-motor-de-corriente-directa.pdf.

20. Zavala, S. (2010). *Diseño de un sistema de control Lógico Difuso Basado en microcontroladores de 8 bits.* Hidalgo: Facultad de Ingeniería Eléctrica de la Universidad Michoacana de San Nicolás de Hidalgo.

21. Arrieta, O. & Alfaro, V. (2003). "Sintonizacion de controladores PI y PID utilizando los criterios integrales IAE e ITAE" *Revista Ingeniería,* 13(1,2), 31-39. Recuperado el 16 de Septiembre del 2019 de la página de internet https://www.academia.edu/4649655/SINTONIZACI%C3%93N_DE_CONTROLADORES_PI_Y_PID_UTILIZANDO_LOS_CRITERIOS_INTEGRALES_IAE_E_ITAE.