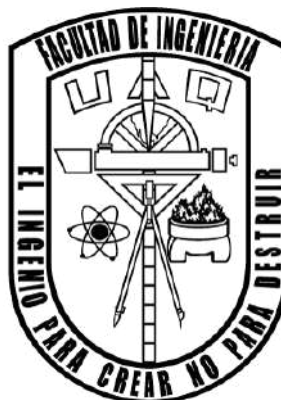


UNIVERSIDAD AUTÓNOMA DE QUERÉTARO



FACULTAD DE INGENIERÍA

Tesis

Sistema de control PID sintonizado por algoritmo de Evolución Diferencial y Ziegler-Nichols aplicado a motores de CD

Que como parte de los requisitos para obtener el grado de ingeniero en automatización con línea terminal en instrumentación y control de procesos

Presenta:

Aldo Cervantes Marquez

Asesor:

Dr. Efrén Gorrostieta Hurtado

Sinodales

Dr. Efrén Gorrostieta Hurtado

Presidente

Dr. Juan Manuel Ramos Arreguín

Secretario

M. en C. Carlos Miguel Torres Hernández

Vocal

M. en C. Jose Erick Rivas Araiza

Suplente

14 de Noviembre de 2020

Dedicatoria

–A mis padres, hermanos y familiares, que a lo largo de mi vida me instruyeron y me guiaron para llegar a este momento. Siendo esta tesis una recompensa por todo el trabajo y tiempo que me dedicaron.–

Agradecimientos

- A la Universidad Autónoma de Querétaro (UAQ) por ofrecerme la oportunidad y el respaldo para formarme como ingeniero.
- A los maestros de la Facultad de Ingeniería (FI) por aportar su enseñanza en mi para llegar a este momento.
- A mi asesor el Dr. Efrén Gorrostieta Hurtado, por su apoyo, enseñanza, confianza y amistad en este trabajo de tesis dándole un mejor enfoque y visión, aclarando dudas y aportando un gran valor al trabajo.
- Al maestro Carlos Miguel Torres Hernández, por su apoyo y cooperación en el desarrollo del trabajo, así como también al ofrecerme su amistad a lo largo de la carrera.
- Al Dr. Juan Manuel Ramos Arreguín, por su enseñanza y sus comentarios para el desarrollo del trabajo de tesis.
- Al maestro José Erick Rivas Araiza por ser un gran mentor en el desarrollo del trabajo y al mismo tiempo ofrecerme su amistad desde los inicios de la carrera.

Índice

Dedicatoria	I
Agradecimientos	II
1. Resumen	1
2. Abstract	2
3. Antecedentes	3
3.1. Importancia del Control y de la Instrumentación en la Industria	3
3.2. Importancia de los Métodos Heurísticos para la Resolución de Problemas de Optimización	4
4. Introducción	5
4.1. Planteamiento del Problema	6
4.2. Hipótesis	7
4.3. Objetivo General	7
4.4. Objetivos específicos	7
4.5. Justificación	8
5. Estado del Arte	9
6. Fundamentación Teórica	11
6.1. Controlador PID	11
6.2. Método de Ziegler-Nichols	12
6.3. Método Heurístico de Evolución Diferencial	13
7. Materiales y Métodos	15

7.1. Equipo de medición	15
7.2. Material Electrónico	16
7.3. Componentes adicionales del prototipo experimental	18
7.4. Métodos	19
8. Prototipo Experimental	23
8.1. Subsistema Electrónico	23
8.2. Software de Control	29
8.3. Operaciones del Software	32
9. Resultados y Discusión	34
9.1. Parámetros del Prototipo Experimental (prueba de escalón)	34
9.2. Parámetros y Respuesta del Controlador PID por Ziegler- Nichols	36
9.3. Parámetros y Respuesta del Controlador PID por Evo- lución Diferencial	39
9.4. Saturación y no linealidades en el sistema de control	42
10. Conclusiones	44
Bibliografía	45
Índice de figuras	
1. Diagrama de bloques de controlador PID paralelo. . .	11
2. Método gráfico de Ziegler-Nichols.	12
3. Diagrama de flujo de algoritmo de Evolución Dife- rencial.	14
4. Multímetro MUT-33.	15

5.	Osciloscopio DSO138.	16
6.	Motoreductor modelo 625500/C y encoder magnético V90-021.	16
7.	PIC 18f4550.	17
8.	Modulo Tlt PL2303.	17
9.	Modulo L298N.	18
10.	Computadora Acer Aspire 5.	19
11.	Respuesta de un sistema de 2do orden al escalón.	20
12.	Diagrama de evolución diferencial.	22
13.	Diagrama de bloques del prototipo experimental.	23
14.	Diagrama esquemático de la placa de control del prototipo experimental.	24
15.	Placa PCB de control y adquisición de datos.	24
16.	Diagrama esquemático de la etapa de aislamiento.	25
17.	Placa PCB de la etapa de aislamiento.	25
18.	Prototipo experimental.	26
19.	Fuente de computadora.	26
20.	Placas PCB de aislamiento y control.	27
21.	Placa de control con detalle a etapas de funcionamiento.	27
22.	Interfaz gráfica en funcionamiento.	28
23.	Pantalla de software en Python para sintonización de controlador PID.	28
24.	Pantalla de inicio de la interfaz gráfica.	29
25.	Ventana de respuesta al escalón.	29
26.	Ventana de PID.	30
27.	Aproximación lineal de voltaje de salida al motor.	31
28.	Aproximación del sistema a un primer orden con retardo.	34

29.	Aproximación del sistema a un primer orden sin retardo.	35
30.	Diagrama de Bode de respuesta en frecuencia del motor 625500/C.	36
31.	Respuesta del controlador con ganancias obtenidas por Ziegler-Nichols.	37
32.	Diagrama de Bode de respuesta en frecuencia del sistema compensado con ganancias obtenidas por Ziegler-Nichols.	38
33.	Respuesta del controlador con ganancias obtenidas por el algoritmo.	40
34.	Diagrama de Bode de respuesta en frecuencia del sistema compensado con ganancias obtenidas por el algoritmo de Evolución Diferencial.	41
35.	Dispersión y dirección de las iteraciones.	42
36.	Respuesta del sistema compensado con saturación de tensión.	43

Índice de tablas

1.	Regla de sintonía de Ziegler-Nichols basada en la respuesta escalón de la planta.	12
2.	Resultados de parámetros de la planta.	34
3.	Ganancias del controlador PID por Ziegler-Nichols.	37
4.	Restricciones de algoritmo de Evolución Diferencial.	39
5.	Resultados obtenidos por el algoritmo de Evolución Diferencial.	39
6.	Resultados obtenidos por el prototipo experimental.	40

1. Resumen

El presente proyecto de tesis presenta la sintonización de un controlador PID mediante un método heurístico de búsqueda llamado evolución diferencial (se suele abreviar como ED) el cual se basa en la teoría de la evolución de Darwin y siendo contrastado con un método clásico llamado Ziegler-Nichols, observando las capacidades del cómputo evolutivo dentro de la teoría de control lineal, mejorando ambientes industriales de trabajo al reducir tiempos de sintonización evitando que el operador seleccione las ganancias de una manera empírica y poco fiable debido a su desconocimiento sobre este tipo de teorías y a la rapidez con la que el trabajo tiene que ser desarrollado para tener el menor tiempo el proceso detenido, al momento de realizar sintonizaciones no únicamente de motores de corriente directa, sino también de sistemas lineales de n orden. Para lograr esto, se debe conocer la función de transferencia del sistema que es controlado (en este caso un motor de corriente directa) por lo que se aplicarán dos métodos de identificación de sistemas (método de mínimos cuadrados y Ziegler-Nichols). Por lo que será necesario conocer, estudiar y documentarse sobre teoría de control clásica (sistemas lineales) y sobre el propio algoritmo de ED aplicado a búsqueda de valores en funciones de transferencia para desarrollar este proyecto de tesis. En esta etapa se pretende abrir una brecha de investigación a los métodos heurísticos de búsqueda en conjunto con la teoría de control para cumplir con los estándares altos de calidad que actualmente existen en la industria de procesos a lo largo del mundo, sin necesidad de la adquisición de dispositivos de control más costosos o con más funciones, abaratando costos de producción para la empresa.

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

2. Abstract

This thesis project presents the tuning of a PID controller using a heuristic search method called Differential Evolution (usually abbreviated as DE) which is based on the theory of evolution of Darwin and being contrasted with a classic method called Ziegler-Nichols, observing the capabilities of evolutionary computation within linear control theory, improving industrial working environments by reducing tuning times by preventing the operator from selecting the gains in an empirical way and unreliable due to its ignorance about this type of theories and the speed with which the work has to be developed to have the shortest time the process stopped, at the time of tuning not only of direct current motors, but also of linear systems of n order. To achieve this, it must be known the system transfer function that is controlled (in this case a direct current motor) therefore, two methods of system identification (least square and Ziegler-Nichols) could be applied. Then it will be necessary to know, study and document on classical control theory (linear systems) and on the ED algorithm itself applied to search for values in transfer functions to develop this thesis project. At this stage it is intended to open a research gap to heuristic search methods in conjunction with control theory to meet the high quality standards that currently exist in the process industry throughout the world, without the need for the acquisition of more expensive or more functioning control devices, lowering production costs for the company.

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

3. Antecedentes

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.

En la actualidad es muy solicitada la necesidad de aplicar teorías de este tipo, en específico hablando de la identificación de sistemas lineales, lo cual nos permite conocer la ecuación característica (función de transferencia) de dicho sistema que quiere ser controlado, como consecuencia se podrá hacer un control más fino y preciso del sistema, teniendo como resultado un proceso mucho más eficiente, estable y con resultados más preferibles.

El presente proyecto de tesis consta de la obtención de dichas ecuaciones y parámetros a partir de métodos de control para este tipo de aplicaciones. Se utilizará el método de Ziegler-Nichols y para sintonizar el sistema a través de un controlador PID cuyas ganancias serán obtenidas mediante un método heurístico llamado Evolución Diferencial.

3.1. Importancia del Control y de la Instrumentación en la Industria

Es complicado suponer un proceso industrializado donde no se encuentren sensores, controladores o elementos finales de control que permitan realizar el proceso o fabricación de un producto [1]. Por lo que la aplicación de estos elementos en la industria resultan útiles para obtener productos con estándares de calidad deseada y siendo económicamente factibles.

Las materias que permiten este tipo de resolución de problemas son la instrumentación y el control, los cuales pueden ser análogos al sistema nervioso y el cerebro en nuestro cuerpo respectivamente.

Por lo que medir y controlar para tomar acciones correctivas sobre un proceso para llevarlo a condiciones deseadas, es vital dentro de la eficiencia de un proceso. Una vez dicho esto, se puede observar que estas acciones son importantes para la producción en cantidades mayores, para poder satisfacer la demanda de un producto o servicio (alimentos, servicios públicos, etc.), y lograr que, tanto la producción como las ganancias para la empresa o institución, sean las suficientes y permitan el crecimiento de la economía nacional e internacional.

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

Los métodos de optimización a través de heurística, son bastante útiles para problemas con soluciones que no son exactas, y/o existen una n cantidad de soluciones correctas con algún margen de error, donde un método exacto tardaría mucho tiempo (miles de iteraciones) para llegar al resultado, y probablemente la capacidad de cómputo sea insuficiente para resolver el problema. Por lo que los algoritmos heurísticos tratan de basarse en comportamientos de la naturaleza que puedan permitir realizar un “atajo” sobre los métodos exactos, como es el caso de una sintonización de un controlador PID, donde existen n cantidad de soluciones, que permite desarrollar correctamente el control del proceso como es requerido.

4. Introducción

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 buena respuesta en un sistema cerrado. Sin embargo, encontrar la combinación adecuada de ganancias en estos sistemas suele ser a veces complicado. El método de sintonización por Ziegler-Nichols es muy popular debido a su simplicidad. No obstante, es complicado encontrar parámetros que cumplan con requerimientos específicos [2].

Como consecuencia se han desarrollado métodos heurísticos para encontrar estas ganancias del controlador, tomados a partir de algoritmos de optimización que en esta tesis será la Evolución Diferencial el caso de estudio para el desarrollo de las ganancias del controlador.

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. Siendo parte de los algoritmos genéticos en su desarrollo y operación para la convergencia de soluciones óptimas, más no exactas.

El algoritmo de Evolución Diferencial para la resolución de problemas multi-objetivo (abreviado como *MODE* por sus siglas en Inglés), consiste en tener un vector solución y evaluarlo, para posteriormente tener una aproximación de la función objetivo (la solución deseada), y en función de ello es que 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 a base de iteraciones un vector solución con un margen de error al ser evaluada en la función objetivo.

4.1. Planteamiento del Problema

Actualmente en la industria se requieren niveles de calidad altos en los productos que son ofertados, estos productos deben ser competentes para poder sobrevivir a la competencia en el mercado. Para lograrlo requieren que sus procesos de fabricación sean rápidos y eficientes, así como también baratos.

La utilización de motores de corriente directa es bastante común debido a su simplicidad. Sin embargo, estos sufren deterioros continuamente debido a su uso, lo que causa un cambio en su función de transferencia, propiciando que los controladores con los que son sintonizados inicialmente no cumplan con los requerimientos en un futuro, causando una descompensación del sistema, requiriendo un paro en la planta de producción para volver a sintonizar el controlador, lo que se traduce en una pérdida en la producción para la empresa.

Desafortunadamente la mayoría de aplicaciones de ese tipo de técnicas de sintonización son bastante empíricas, debido a que no es conocido el cambio exacto en dicha función de transferencia del motor y no se sabe con precisión cuál será el cambio de ganancias que necesitará el sistema. Además de que el desconocimiento sobre técnicas de control por parte de los operadores pueden afectar, aletargando el proceso de sintonización. Lo que puede causar que el reinicio de la planta pueda atrasarse más. Por lo que conocer cómo ha cambiado la función de transferencia del motor permitirá conocer una referencia sobre el comportamiento del motor y mediante la sintonización a través del método heurístico de Evolución Diferencial permitirá que el lazo pueda ser sintonizado de una manera más sencilla y rápida con los criterios que fueron requeridos.

Actualmente existen sistemas de control con la función integrada de auto tuning, pero esto causa una elevación en los costes debido a que este tipo de sistemas son más caros y existen empresas que no podrían darse el lujo de cambiar todo su sistema de control, además de que el auto tuning no es del todo confiable debido a sus problemas al trabajar en modo automático mientras se encuentra en la etapa transitoria de la respuesta del sistema. Por lo que la opción de utilizar un método heurístico como es la Evolución Diferencial es una opción bastante útil y rentable para las empresas en la actualidad.

4.2. Hipótesis

- Implementar un prototipo para identificar funciones de transferencia y sintonizar las ganancias de un controlador PID a través de Evolución Diferencial (ED) logra la sintonización de un sistema reduciendo el tiempo de sintonización a menos de 1 minuto.

4.3. Objetivo General

- Desarrollar un prototipo para la identificación de funciones de transferencia de motores de corriente directa mediante Ziegler-Nichols y sintonizar dicho sistema usando la técnica de Evolución Diferencial, que permita fortalecer las líneas de investigación en aplicaciones de métodos heurísticos y de control.

4.4. Objetivos específicos

- Montar bastidor de acrílico mediante tornillería para colocar todos los componentes del sistema.
- Diseñar placa PCB para el data-logger mediante un programa de software para albergar los componentes electrónicos del sistema.
- Crear programa(s) para microcontrolador(es) a través de lenguaje C para obtener datos de los sensores (crear el software de un data-logger) .
- Validar mediciones mediante instrumentos de referencia para obtener fiabilidad de los datos.
- Diseñar una estrategia de control para el funcionamiento del motor mediante el uso de la fundamentación teórica de control.
- Desarrollar un programa computacional a través de un lenguaje de programación que pueda analizar y procesar datos recibidos por el data-logger para poder tener comunicación bidireccional entre la computadora y el data-logger.
- Desarrollar programa computacional a través de un lenguaje de programación (Python) para crear un algoritmo de Evolución Diferencial que pueda obtener las ganancias adecuadas para el controlador PID.

4.5. Justificación

El problema de realizar una sintonización correcta es un tema bastante 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 son una excelente herramienta para realizar este tipo de trabajos.

Por lo que el interés principal del presente trabajo de tesis, de nombre “*Sistema de control PID sintonizado por algoritmo de Evolución Diferencial y Ziegler-Nichols aplicado a motores de CD*”, es poder abrir campo en el estudio del control y los métodos heurísticos de manera conjunta, así como también realizar un prototipo que permita sacar provecho para el sector productivo.

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). Todo esto con la finalidad de poder aplicar un algoritmo evolutivo para obtener las ganancias que satisfagan y/o mejoren las condiciones del sistema.

5. Estado del Arte

Los algoritmos heurísticos tienen mucho tiempo de haber sido desarrollados. Sin embargo, fueron mayormente usados para la minimización y maximización de funciones para problemas de búsqueda y optimización, por lo que el uso en sistemas de control a tomado en los últimos años un avance creciente sobre los experimentos realizados aportando nuevas ideas y estrategias en este campo de estudio.

V. Vishal, V. Kumar, K.P.S. Rana y P. Mishra [4] desarrollaron una investigación sobre las distintas técnicas de sintonización por métodos heurísticos aplicadas a motores de corriente directa, donde contrastaron las desventajas de la sintonización por métodos clásicos sobre los heurísticos observando deficiencias en el desempeño del sistema.

Abdelhakim Idir, Madjid Kidouche, Yassine Bensafia, Khatir Khettab y Sid Ahmed Tadjer [5] desarrollaron un algoritmo de sintonización de un controlador PID, ocupando Evolución Diferencial y Optimización por Enjambres de Partículas (PSO). Para demostrar la efectividad de estos métodos en la teoría de control.

B. Monthary and S. Panda y P.K. Hota [6] desarrollaron un controlador PI y PID sintonizado por evolución diferencial y algoritmos genéticos para controlar un sistema de potencia no lineal, demostrando su efectividad en ese tipo de sistemas de control.

K. Ogata [7] presenta un código en Matlab donde se realizaron pruebas de un controlador PID sometido a varias ganancias introducidas ya definidas en sus partes proporcional, integral y derivativa, donde únicamente se muestra la mejor dentro de esa población limitada.

Maria Guadalupe Martínez [8] realizó una investigación sobre el método de Evolución Diferencial, donde explica de una manera detallada el algoritmo que tiene que ser llevado a cabo y el fundamento matemático que envuelve este método para ser posible.

L. Vicente Santana Quintero [9] realizó una tesis a nivel maestría donde se explican las ventajas del uso de los algoritmos genéticos y heurísticos para la resolución de problemas, observando resultados satisfactorios del uso de dichos métodos en comparación con los métodos exactos.

V. Mazzone [10] desarrolló un artículo sobre el controlador PID paralelo junto con sus sintonizaciones por diferentes métodos como Ziegler-Nichols y Curva de Reacción. Obteniendo resultados sobre el rango de controlabilidad de los contro-

ladores PID y sus variaciones (PD y PI).

T. Sibalija [11] realizó una investigación sobre el aprovechamiento de los métodos heurísticos en la industria, comparándolo con los procesos actuales y enfocándose en los algoritmos computacionales para procesos de sintonización y la optimización computacional.

L.C.T. Bezerra, M. López-Ibáñez y T. Stützle [12] desarrollaron un artículo sobre los algoritmos heurísticos multi-objetivo y su importancia en la automatización, así como también en su alto desempeño sobre la búsqueda de datos y la realización de procesos.

M. G. Villarreal-Cervantes, A. Rodríguez-Molina, C. García-Mendoza, O. Peñaloza-Mejía y G. Sepúlveda-Cervantes [13] proponen un controlador adaptativo a través de un controlador multi-objetivo en línea mediante el algoritmo de Evolución Diferencial, para el control de posición del motor de un sistema mecatrónico.

6. Fundamentación Teórica

6.1. Controlador PID

El controlador PID es una estrategia de control ampliamente usada que permite la regulación de sistemas a partir de una entrada o error (generalmente una diferencia algebraica entre el valor deseado o Setpoint (S_p) y el medido (Y) representada por la letra e) y a su salida un esfuerzo de control (generalmente representado por la letra u) [14] [15].

El controlador PID consta de 3 partes esenciales las cuales son su parte Proporcional, Integral y Derivativa, existen muchos arreglos de estas 3 variables, por lo que para este caso se ocupará su arreglo en paralelo como se muestra en (1).

$$u(t) = k_p e(t) + k_i \int e(t) dt + k_d \frac{de(t)}{dt} \quad (1)$$

Aplicando transformada de Laplace \mathcal{L} se obtiene el modelo del PID (2) continuo:

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

Se muestra en la Figura 1 los elementos del controlador PID paralelo dentro de un sistema de control mediante un diagrama de bloques.

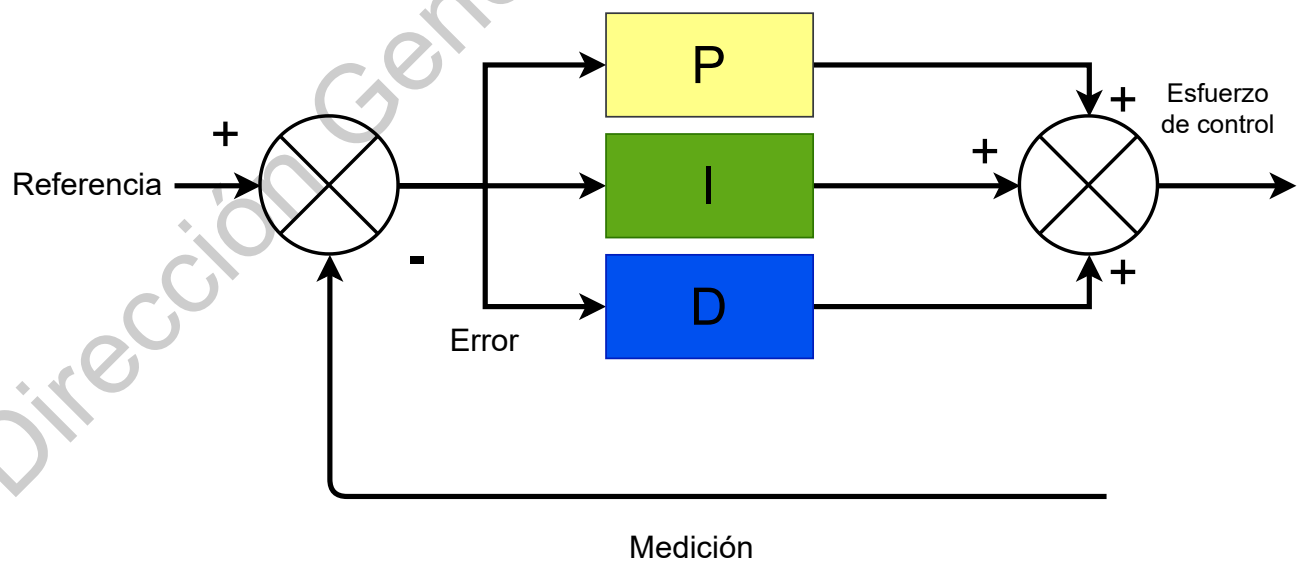


Figura 1: Diagrama de bloques de controlador PID paralelo.

6.2. Método de Ziegler-Nichols

El método de Ziegler-Nichols permite obtener los parámetros de un sistema de primer orden con retardo, a través de una prueba en lazo abierto con la medición de la salida. Con la gráfica del sistema se obtendrán los parámetros correspondientes [7] [16].

En este caso se propone una función de transferencia de la planta (motor de corriente directa) de la siguiente forma (3):

$$\frac{Y(s)}{U(s)} = \frac{Ke^{-Ls}}{\tau s + 1} \quad (3)$$

en la Figura 2 se muestran los componentes de la ecuación (3) de manera gráfica la respuesta al escalón del sistema.

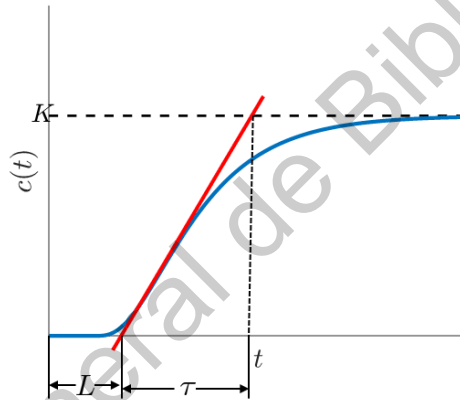


Figura 2: Método gráfico de Ziegler-Nichols.

Este método también ofrece una regla de sintonía de controlador PID (Tabla 1).

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	$1.2 \frac{\tau}{L}$	$2L$	$0.5L$

Tabla 1: Regla de sintonía de Ziegler-Nichols basada en la respuesta escalón de la planta.

Donde se propone un controlador PID [7] de la forma (4):

$$\frac{U(s)}{E(s)} = k_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (4)$$

6.3. Método Heurístico de Evolución Diferencial

La Evolución Diferencial (abreviada como ED) es una estrategia de búsqueda, que está diseñada para localizar un elemento concreto dentro de una estructura de datos.

Este algoritmo se basa en la teoría de evolución de Darwin. Se modela de una manera computacional la selección natural y el entrecruzamiento de especies por medio de la recombinación genética y la mutación [17].

La idea principal detrás de la ED es un nuevo esquema para generar vectores. La ED genera estos nuevos vectores cuando se suma la diferencia de pesos entre dos vectores miembros de la población a un tercer vector miembro. Si la aptitud del vector resultante es menor que el miembro de la población elegido entonces el nuevo vector reemplaza al vector con el cual fue comparado. Este vector a comparar puede ser (aunque no necesariamente lo es) parte del proceso de generación arriba mencionado. Además, el mejor vector $X_{mejor,G}$ se evalúa en cada generación G para no perder de vista el progreso durante el cual se hace la minimización [18]. El comportamiento anteriormente descrito puede ser visualizado en un diagrama de flujo como se muestra a continuación en la Figura 3.

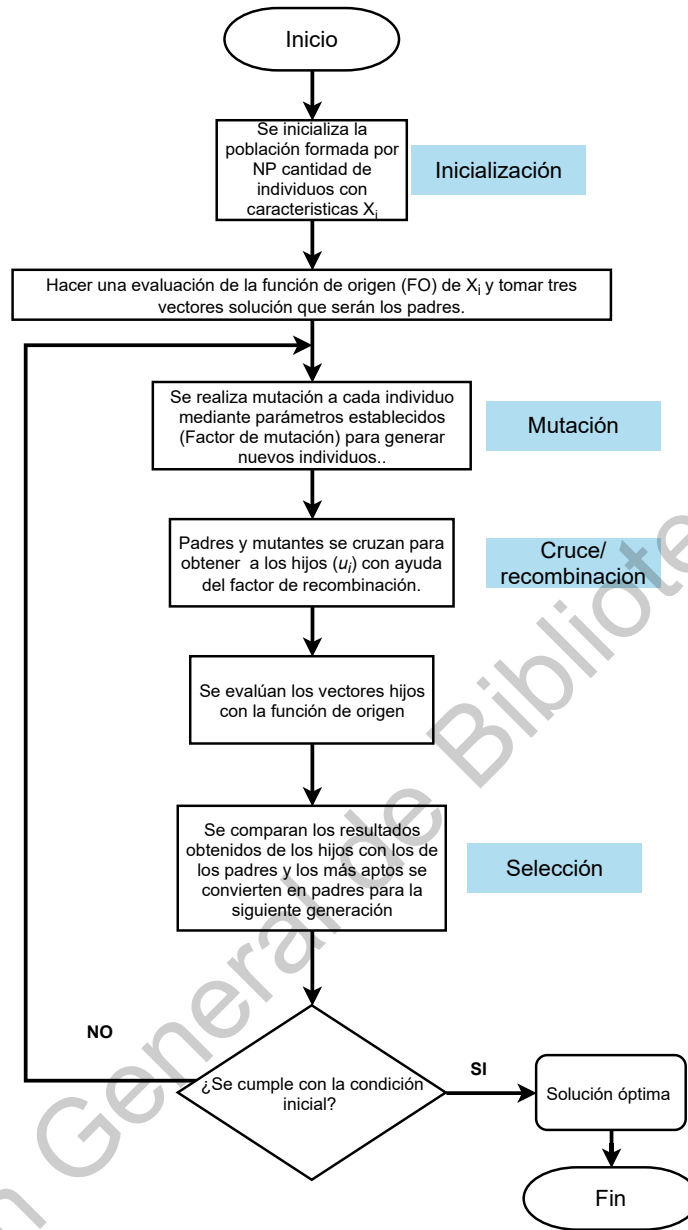


Figura 3: Diagrama de flujo de algoritmo de Evolución Diferencial.

7. Materiales y Métodos

7.1. Equipo de medición

Se utilizó un multímetro marca TRUPER modelo MUT-33 (véase Figura 4) para realizar las mediciones de tensión en un rango de 0 a 20V de corriente directa, corriente con un rango de 0 a 10A de corriente directa. Para poder posteriormente medir la de alimentación de los componentes y su consumo de energía W. Así como también fue de utilidad para medir continuidad y resistencia en un rango de 0 a 20,000,000 Ω al momento de realizar pruebas en el prototipo experimental.



Figura 4: Multímetro MUT-33.

Asimismo se utilizó un osciloscopio de bajo costo de la marca Quimat modelo DSO138 de un canal (véase Figura 5), con la capacidad de medir en un rango de amplitud de 50V pico a pico, un ancho de banda de 0 a 200KHz a una resolución de 12 bits y un rango de tiempo base mínimo de 10 μ s y máximo de 500s. Con el propósito de observar el comportamiento de las señales como son: la señal de entrada de alimentación al motor, la señal de PWM de salida del controlador y la señal de salida del encoder en PPR aplicando la fórmula (5).

$$rev_s = 2b \cdot R \quad (5)$$

donde rev_s son las revoluciones por segundo $\frac{revolucion}{s}$ o R.P.S., b es la frecuencia del osciloscopio en Hz que al multiplicarse por 2 se convierte en flancos de subida por segundo $\frac{pulsos}{s}$ y R es la relación de revolución por pulsos $\frac{revolucion}{pulsos}$.



Figura 5: Osciloscopio DSO138.

7.2. Material Electrónico

Para el proyecto se utilizará un motoreductor de corriente directa con escobillas con una relación de engranajes de 100:1, portando un modelo identificador 625500/C (véase Figura 6). Teniendo como características una velocidad angular máxima de 80 R.P.M. a 10.53V y un encoder magnético (sensor de efecto Hall) de modelo V90-021 de 100 PPR (Pulsos Por Revolución).



Figura 6: Motoreductor modelo 625500/C y encoder magnético V90-021.

Microcontrolador PIC 18F4550 (véase Figura 7) trabajando a 20MHz con un cristal oscilador externo, mediante el cual se harán las operaciones correspondientes que serán enviadas a la PC y recibirá datos para el modo de controlador en un tiempo de muestreo aproximado a $t_m = 0.01s$. Realizando una lectura periódica del número de pulsos contados por el encoder.



Figura 7: PIC 18f4550.

Un modulo USB a serial RS-232 modelo Tlt PL2303 (véase Figura 8) que permitirá la comunicación y transferencia de datos entre el microcontrolador y la PC para el procesamiento correspondiente de cada elemento, logrando una comunicación a 115,200 Baud (**Nota:** cabe destacar que cualquier tipo de convertor de USB a serial funcionaría como por ejemplo los módulos Bluetooth HC-0X y el circuito integrado MAX232).



Figura 8: Modulo Tlt PL2303.

Un modulo de puente H basado en el circuito integrado L298N (véase Figura 9) para lograr amplificar la potencia de la señal de PWM ofrecida por el microcontrolador PIC 18F4550, protegiéndolo contra sobrecargas de corriente.

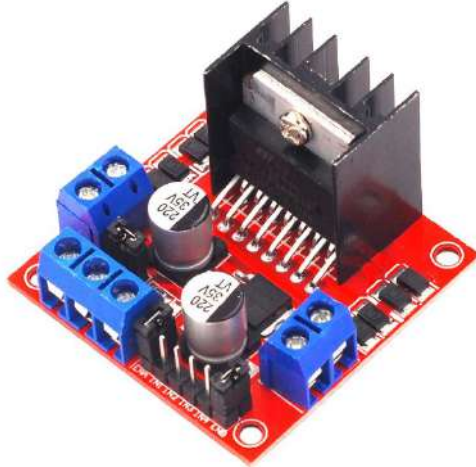


Figura 9: Modulo L298N.

7.3. Componentes adicionales del prototipo experimental

Se realizó una interfaz gráfica realizada en C# de Visual Studio 2019 que permitirá la visualización, configuración y procesamiento de los datos, creando imágenes .PNG y archivos .TXT. Mediante de una computadora acer modelo Aspire 5 A515-51-52BQ (véase Figura 10) con las siguientes características:

- Sistema Operativo Windows 10 de 64 bits.
- Procesador Inter Core i5-7200U a 2.5GHz.
- Tarjeta gráfica Intel HD graphics.
- Memoria RAM 8Gb DDR4.
- Dos puertos USB 2.0 y uno 3.0



Figura 10: Computadora Acer Aspire 5.

Así como también se ocupó software para el procesamiento e interpretación de datos:

- MATLAB professional versión R2017a .
- Proteus versión 8.6 SP2.
- PIC C compiler versión 5.015.
- Spider versión 3.3.6.
- Visual Studio versión 2019.
- App Strobo RPM.

7.4. Métodos

La metodología a seguir para la obtención de los parámetros de la función de transferencia y la sintonización del controlador, una vez construido el data-logger, es la siguiente:

1. Realizar la prueba de escalón al sistema, en caso de ser una respuesta de primer orden (sin sobrepaso o un sobrepaso despreciable), como es el caso de la Figura 2, se encuentran las constantes de tiempo (τ y L) para formar un sistema de ecuaciones (6) y (7) de la siguiente manera:

$$L + \frac{\tau}{3} = t_{0.284} \quad (6)$$

$$L + \tau = t_{0.632} \quad (7)$$

donde L y a τ son las constantes en la ecuación (3), $t_{0.284}$ y $t_{0.632}$ representan al tiempo en el que el valor final de la salida (K) alcanza el 28.4 % y el 63.2 % representado en la ecuación (8):

$$K = \frac{Y_f}{U_e} = \frac{Y_{final} - Y_{inicial}}{U_{final} - U_{inicial}} \quad (8)$$

Donde Y_f es el valor final en estado estacionario en R.P.M. y U_e es la amplitud del escalón a la entrada en V, por lo que para este caso se obtendrá en $\frac{R.P.M.}{V}$. Una vez obtenidos los valores de L , τ y K se sustituyen en la ecuación (3), si se le aplica transformada inversa de Laplace (\mathcal{L}^{-1}) se obtiene la expresión (9) en tiempo continuo:

$$y(t) = KU_e[1 - e^{-\frac{t-L}{\tau}}] \quad (9)$$

En caso de que la función de transferencia tuviera un comportamiento de segundo orden o superior (véase Figura 11) con un sobrepaso y un tiempo de subida.

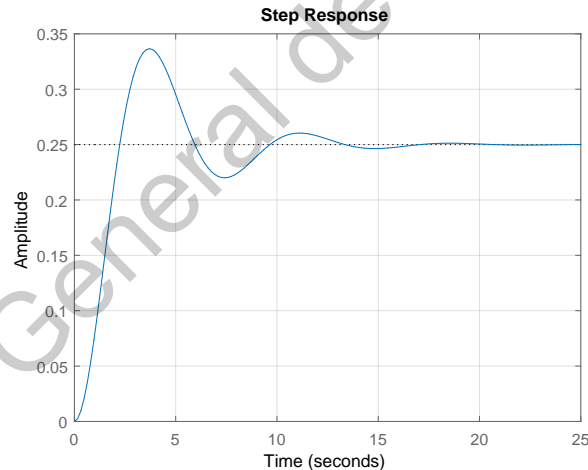


Figura 11: Respuesta de un sistema de 2do orden al escalón.

Se aplicará el método de mínimos cuadrados fuera de línea basado en el modelo ARX expuesto en [16], por lo que resultará necesario acudir a dicha referencia si es necesaria la demostración. Se propone una función de transferencia discreta (10):

$$\frac{Y(z)}{U(z)} = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_nz^{-n}}{1 - a_1z^{-1} - a_2z^{-2} - \dots - a_nz^{-n}} \quad (10)$$

donde las constantes (parámetros) serán obtenidas a partir de operaciones matriciales de los datos adquiridos por el data-logger expresadas en las ecuaciones (11) y (12).

$$P^T = [b_0 \ a_1 \ b_1 \ \cdots \ a_n \ b_n] \quad (11)$$

$$z^T(k) = [u(k) \ y(k-1) \ u(k-1) \ \cdots \ y(k-n) \ u(k-n)] \quad (12)$$

donde P^T y $z^T(k)$ son vectores de dimensión $(2n+1)$ por lo que al aproximar a un segundo orden $n = 2$ y la dimensión será de 5.

Por lo que una vez dicho esto, se puede expresar de una manera generalizada en una entrada entrada escalón (representada con u) y la salida del sistema (representada con y) en un intervalo de tiempo (tiempo de muestreo, representado por t_m) como se muestra en las ecuaciones (13)-(16):

$$Y^T(t_m) = P^T Z^T(t_m) + E_r^T(t_m) \quad (13)$$

$$Y^T(t_m) = [y(1) \ y(2) \ \cdots \ y(t_m)] \quad (14)$$

$$U^T(t_m) = [u(1) \ u(2) \ \cdots \ u(t_m)] \quad (15)$$

$$Z^T(t_m) = \begin{bmatrix} u(1) & u(2) & \cdots & u(t_m) \\ y(0) & y(1) & \cdots & y(t_m - 1) \\ u(0) & u(1) & \cdots & u(t_m - 1) \\ \vdots & \vdots & \ddots & \vdots \\ y(1-n) & y(2-n) & \cdots & y(t_m - n) \\ u(1-n) & u(2-n) & \cdots & u(t_m - n) \end{bmatrix} \quad (16)$$

Conociendo las matrices $Z(t_m)$, $Z^T(t_m)$ y el vector $Y(t_m)$, se puede calcular fácilmente la expresión del estimador mínimo cuadrático (17) que tiene la forma:

$$\hat{P}(t_m) = [Z^T(t_m)Z(t_m)]^{-1}Z^T(t_m)Y(t_m) \quad (17)$$

Para poder cambiar la función de transferencia de tiempo discreto (ecuación (10) a una función de tiempo continuo, transformada de Laplace \mathcal{L}) se aplicará una función de MATLAB llamada d2c, por lo que será necesario consultar a [19] para obtener el método de transformación a tiempo continuo.

- Una vez identificado el orden y los parámetros correspondientes, se realiza la sintonización por Ziegler-Nichols para primer orden (Tabla 1) sustituyendo como se muestra en la ecuación (18), en caso de ser un segundo orden, no aplica este método por lo que se omitirá este paso.

$$\frac{U(s)}{E(s)} = k_p + \frac{k_p}{T_i s} + (k_p \cdot T_d s) \quad (18)$$

por lo que las ganancias de k_i y de k_d serán (19) y (20):

$$k_i = \frac{k_p}{T_i} \quad (19)$$

$$k_d = k_p \cdot T_d \quad (20)$$

- Una vez identificado el orden y los parámetros correspondientes, se realiza la captura de la función de transferencia en el programa creado en Python, con las restricciones deseadas para obtener la combinación de ganancias para el controlador. Para posteriormente ser probadas tanto en simulación como por el prototipo experimental, a continuación se muestra (Figura 12) en un diagrama de bloques, los pasos que realizará el programa para realizar la sintonización adecuada.

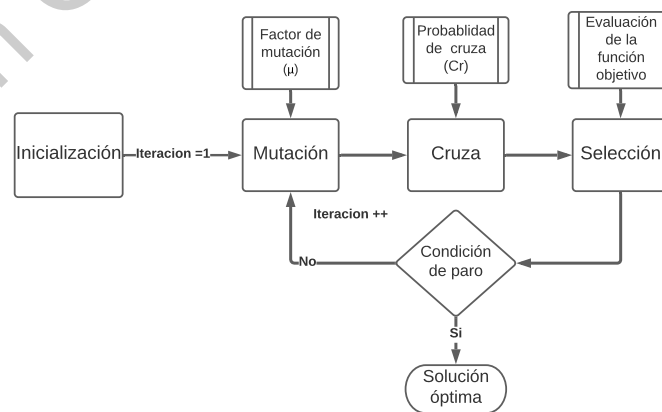


Figura 12: Diagrama de evolución diferencial.

8. Prototipo Experimental

El prototipo experimental consta básicamente de una parte electrónica y una parte de software para el almacenamiento y procesamiento de datos (data-logger), como se muestra en la Figura 13.

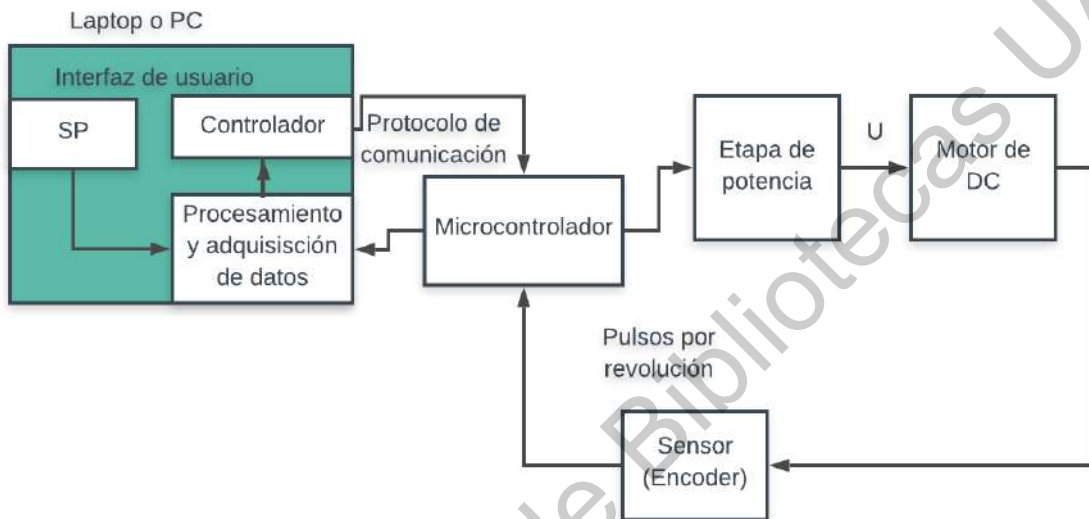


Figura 13: Diagrama de bloques del prototipo experimental.

8.1. Subsistema Electrónico

Este subsistema consta de una placa de control, donde se almacena el PIC 18F4550 que contiene puertos de comunicación con el módulo USB-RS232 y con el programador, así como también consta de LEDs que permiten conocer el estado de la placa (encendido y apagado), y la etapa en la que se encuentre el proceso del prototipo (prueba de escalón, prueba de PID) (véase Figura 14 y 15).

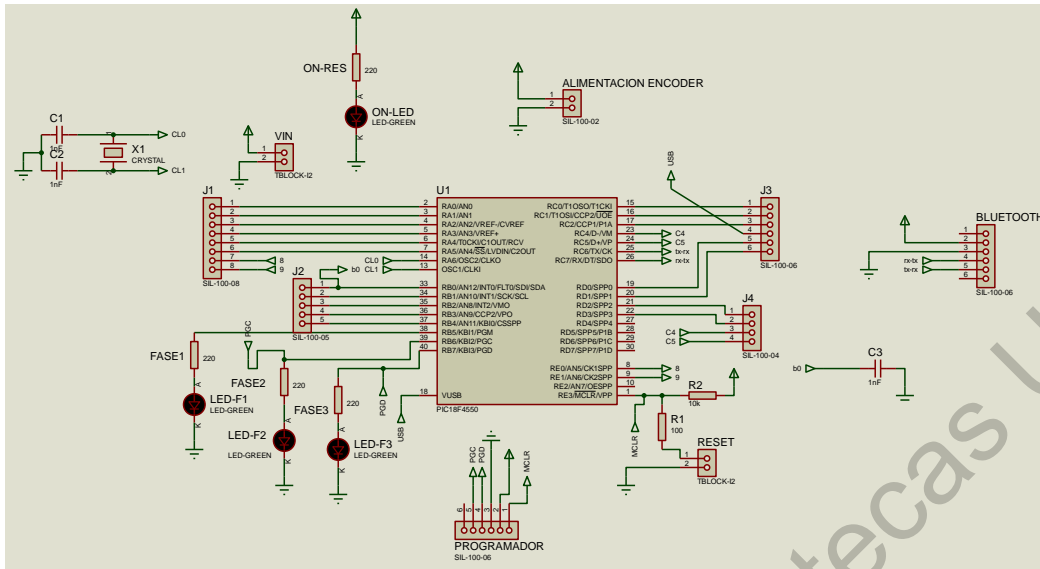


Figura 14: Diagrama esquemático de la placa de control del prototipo experimental.

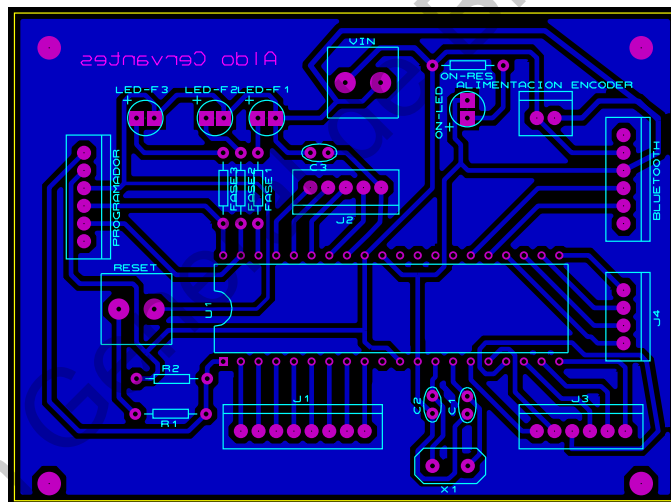


Figura 15: Placa PCB de control y adquisición de datos.

Posteriormente se lleva a una placa de aislamiento donde se separan 2 fuentes a través de un opto-acoplador, por un lado se tiene una fuente de computadora que soporta picos de hasta 20A para la alimentación del puente H. Donde se proporcionará una amplificación de señal hacia el motor, teniendo como consecuencia tensiones muy inestables para el microcontrolador, pudiendo reiniciar el programa y causar fallas en su operación. Por esta razón es que se tiene otra fuente de un eliminador de baterías, la cual es muy estable y al no demandar mucha corriente, es menos probable que existan picos que alteren la operación del microcontro-

lador. Adicional a esto, se tienen 2 compuertas AND, que tienen como función asegurar los estados lógicos que serán mandados al puente H para ser amplificados, evitando ruido e interferencia por parte del circuito de control (Véase Figura 16 y 17).

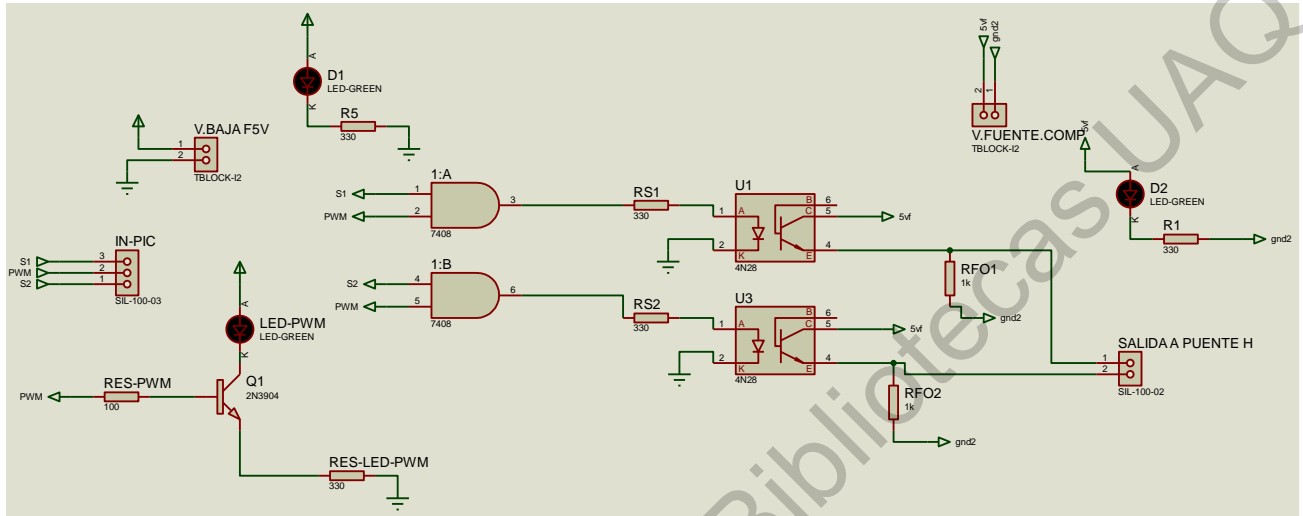


Figura 16: Diagrama esquemático de la etapa de aislamiento.

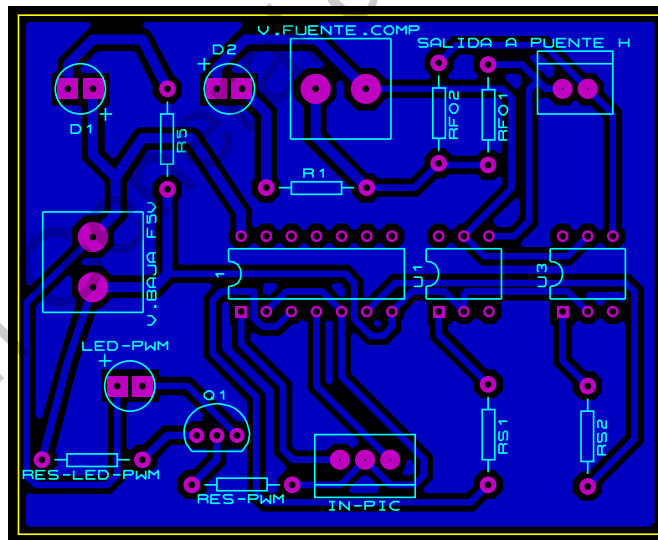


Figura 17: Placa PCB de la etapa de aislamiento.

A continuación, se muestran imágenes referentes al data-logger junto con el software trabajando en conjunto para la adquisición y tratamiento de datos. Primeramente en la Figura 18, se muestra el prototipo experimental con todos sus componentes integrados (motor, placas PCB, fuentes de alimentación, Laptop y software).

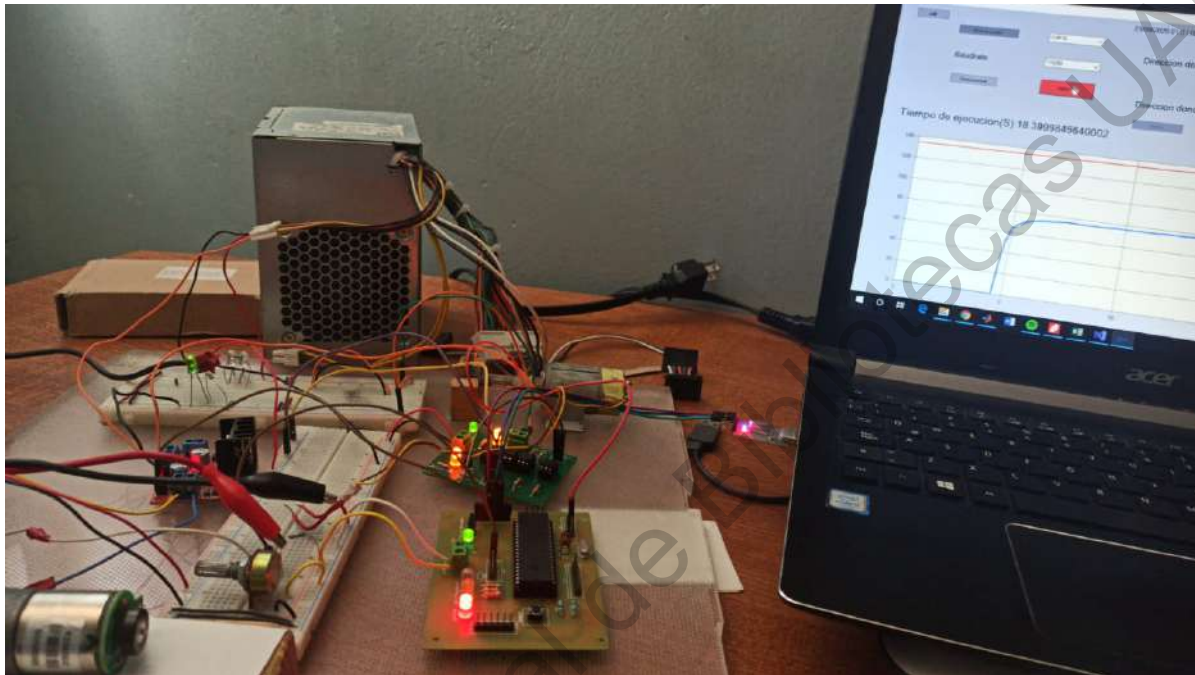


Figura 18: Prototipo experimental.

Por su parte en la figura 19 se muestra a detalle la fuente de alimentación principal, con sus características energéticas que permitirán que el motor logre moverse con facilidad sin tener una limitación considerable de energía eléctrica. Además de proveer de distintas tensiones para los demás componentes que sean requeridos.



Tensión	Pin	Color	Color	Pin	Tensión
+3.3 V	1	Orange	Brown	13	+3.3 V
+3.3 V	2	Orange	Blue	14	-12 V
Tierra	3	Black	Black	15	Tierra
+5 V	4	Red	Green	16	PS_ON
Tierra	5	Black	Black	17	Tierra
+5 V	6	Red	Black	18	Tierra
Tierra	7	Black	Black	19	Tierra
Power OK	8	Black	White	20	-5 V (opcional)
+5 VSB	9	Purple	Red	21	+5 V
+12 V	10	Yellow	Red	22	+5 V
+12 V	11	Yellow	Red	23	+5 V
+3.3 V	12	Orange	Black	24	Tierra

Figura 19: Fuente de computadora.

Adicional a esto, se muestra a detalle las placas PCB con un acabado profesional y soldadas (véase Figura 20). Esto permitirá que no existan falsos contactos y los componentes se encuentren totalmente fijos. También permite tener en un espacio reducido y con un buen aspecto al proyecto.

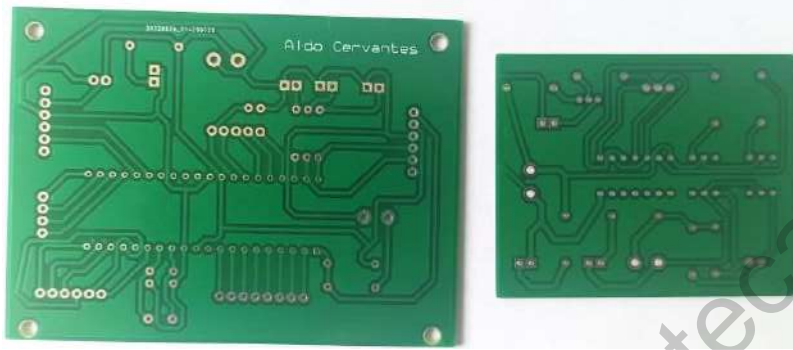


Figura 20: Placas PCB de aislamiento y control.

Dentro de la placa de control (véase Figura 20 y 21) se observan los LEDs que indican la etapa en la que se encuentra el proceso de experimentación, siendo atractivo y lógico de una manera visual. Esto para obtener una experiencia mas amigable y de fácil entendimiento, sin necesidad de conocimientos previos por parte del usuario.

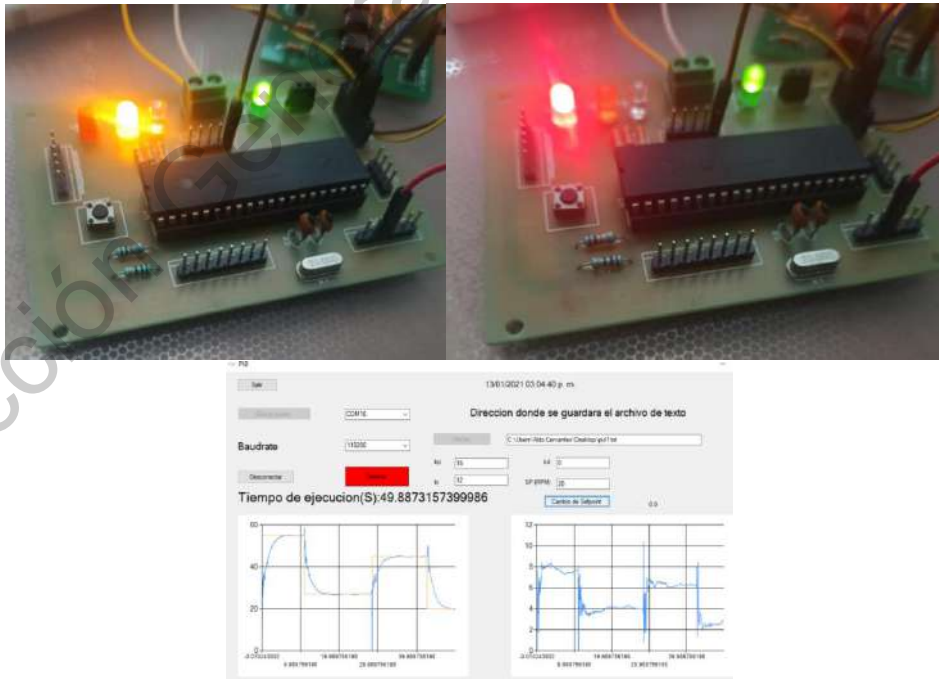


Figura 21: Placa de control con detalle a etapas de funcionamiento.

Con la finalidad de que cualquier persona pueda utilizar esta herramienta, se agregó en el software algunos colores que indican el estado del sistema (véase Figura 22), tanto para el inicio como para el final de cada prueba que sea realizada.

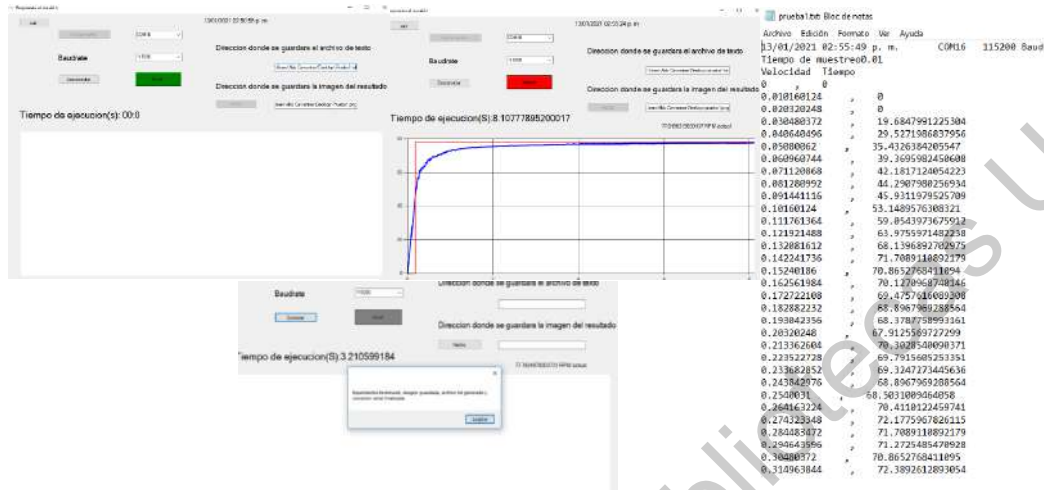


Figura 22: Interfaz gráfica en funcionamiento.

Por parte del software de sintonización, se tiene un programa de software en Python donde se muestra el resultado de cada iteración (véase Figura 23), además de que los datos son almacenados en variables para poder ser gráficas en caso de ser requeridas. También al finalizar las iteraciones, se muestra una gráfica de la respuesta del sistema, y se despliegan los datos más importantes de dicha respuesta (numero de iteraciones, sobrepaso, valor final, ganancias del controlador, etc.).

```
[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
```

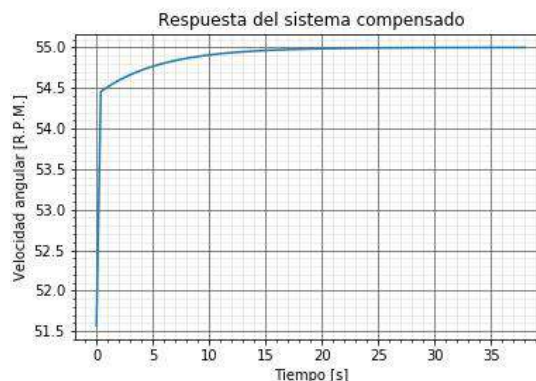


Figura 23: Pantalla de software en Python para sintonización de controlador PID.

8.2. Software de Control

El software de control consta de un programa en C# donde se tiene comunicación con la placa de control, a través de un módulo USB-RS232, hacia una interfaz gráfica (véase Figura 24) dónde se procesarán los datos, al iniciar el programa se tendrá la siguiente ventana.

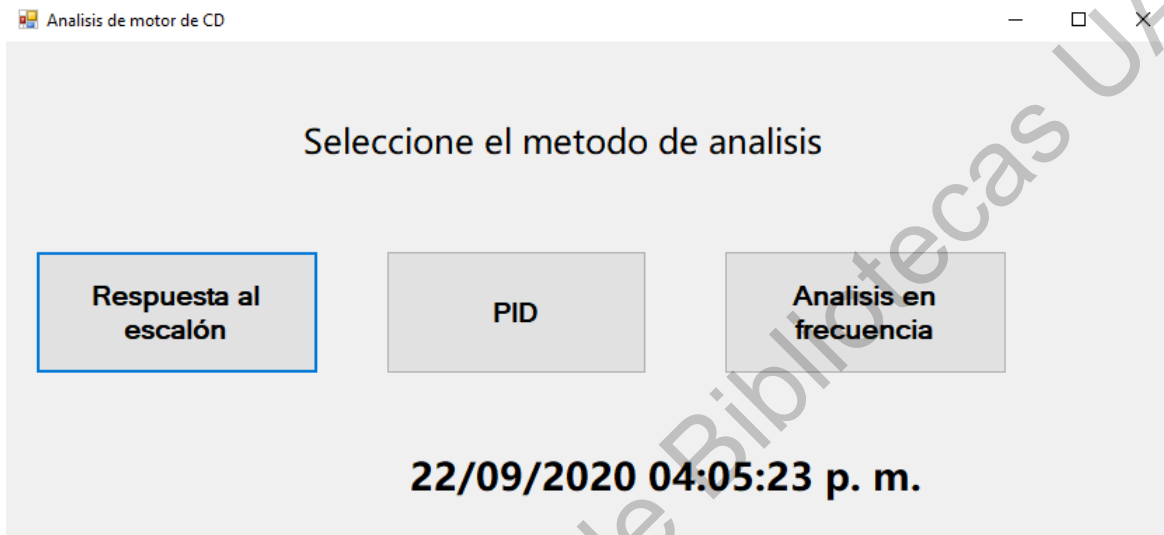


Figura 24: Pantalla de inicio de la interfaz gráfica.

Ahí se podrá seleccionar el modo de operación del microcontrolador, en este caso es de interés el PID y la respuesta a escalón. Al seleccionar el botón de Respuesta al escalón, se abrirá una ventana como la siguiente (véase Figura 25).

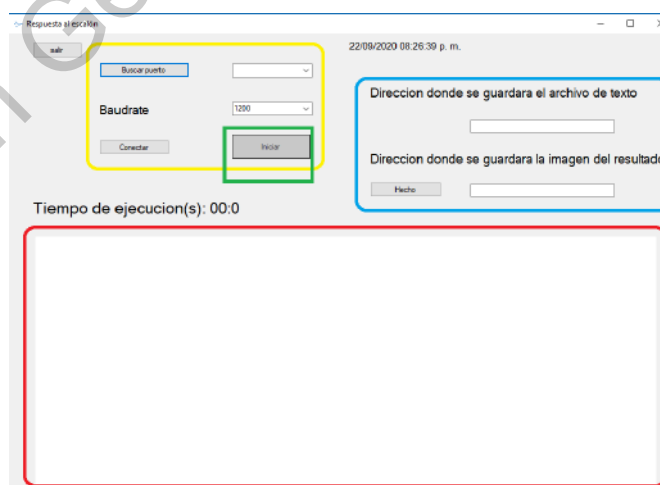


Figura 25: Ventana de respuesta al escalón.

Donde:

- El recuadro amarillo representa la parte de la comunicación serial con el conversor USB-RS232.
- El recuadro azul muestra dos textbox, donde se deberá incluir una ruta de acceso debido a que se creará un archivo con extensión .TXT y uno con extensión .PNG. Con la finalidad de poder importar los datos a algún software externo para procesar los datos.
- El recuadro rojo muestra donde se verán gráficos los datos obtenidos por el data-logger.
- El recuadro verde muestra un botón que, cuando sean llenados los campos del recuadro amarillo y azul, cambiará de color a verde para iniciar la prueba, en ese momento cambiara a rojo para detener la prueba experimental cuando sea requerido.

Por otro lado, la interfaz al seleccionar el botón del PID (véase Figura 26), se mostrará una ventana muy semejante a la Figura 25 con unos pocos cambios que serán explicados.



Figura 26: Ventana de PID.

Donde:

- Los recuadros amarillo, azul y verde tienen la misma función que los de la respuesta al escalón (Figura 25).
- El recuadro morado muestra un grupo de 4 textbox, donde se deben ingresar las ganancias requeridas para el controlador, así como también el Setpoint o valor deseado de velocidad del motor.
- El recuadro rojo muestra que habrán dos gráficas, una de voltaje aplicado (derecha), a partir de datos experimentales (véase Figura 27), se realizó una regresión lineal del voltaje aplicado, y la otra gráfica (izquierda), mostrará el comportamiento de la salida (sistema compensado) en lazo cerrado con las ganancias ingresadas.

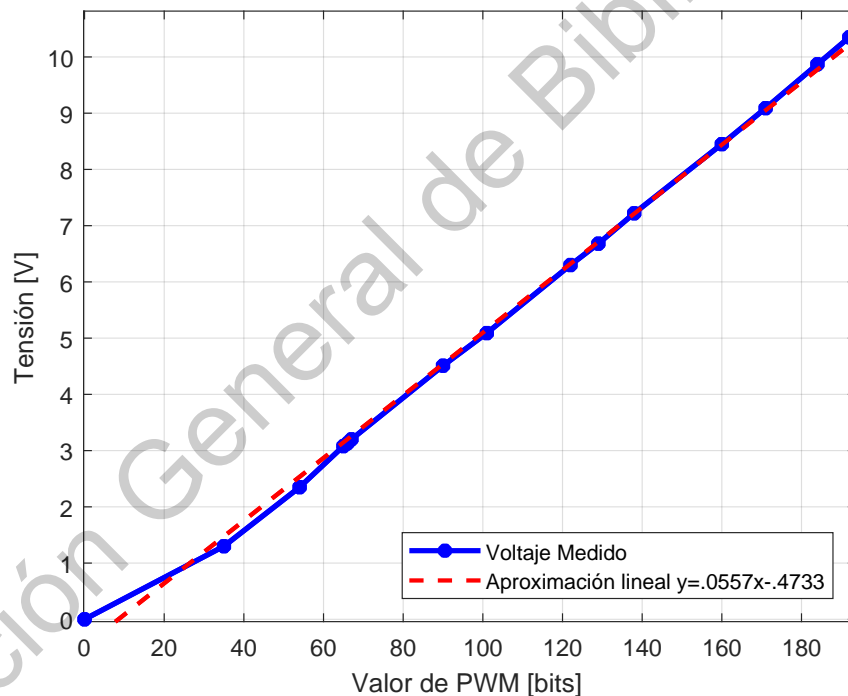


Figura 27: Aproximación lineal de voltaje de salida al motor.

8.3. Operaciones del Software

El programa realizado en C# realiza básicamente 2 operaciones fundamentales, las cuales son:

- cálculo de la velocidad del motor.
- calculo de salida del controlador PID.

Para el caso de la medición de la velocidad, el motor envía un tren de pulsos de frecuencia variable con un ciclo de trabajo del 50%, por lo que la medición del periodo entre pulsos es en donde se basa la medición (21), a través de la fórmula:

$$\dot{\theta} = \frac{\text{pulsos}(k)}{t_m[s]} \cdot \frac{1\text{revolución}}{100\text{pulsos}} = \frac{\theta(k) - \theta(k-1)[rad]}{\Delta t[s]} \quad (21)$$

Por lo que la velocidad es medida a partir del número de pulsos medidos por cada tiempo de muestreo. Posteriormente dividiéndolo por el numero de pulsos que tiene cada revolución, siendo igual a medir una diferencia de posiciones entre la posición en el tiempo de muestreo actual y el tiempo de muestreo anterior, dividido entre la diferencia de tiempos en que fueron medidos. Teniendo como resultado que esa diferencia de tiempos sea igual al tiempo de muestreo t_m . Para que se tengan unidades más comerciales, se hizo la conversión a revoluciones por minuto (R.P.M.), multiplicando por la equivalencia $\frac{60[s]}{1[min]}$. Logrando eliminar los segundos del tiempo de muestreo y quedando en el denominador minutos.

Debido al bajo número de pulsos por revolución del motor, además de las bajas velocidades que maneja, se realizó un método de medias aritméticas de las mediciones propuesta en [20], con la finalidad de evitar picos de velocidad y fluctuaciones de la medición, pudiendo afectar más adelante al funcionamiento del controlador PID en especial en su parte derivativa (D). En consecuencia no será necesario incrementar el tiempo de muestreo. Simplificando será la fórmula (22) la que se utilizará.

$$\dot{\theta} \approx \sum_{k=1}^n \frac{\frac{\text{pulsos}(k)}{t_m[s]} \cdot \frac{1\text{revolución}}{100\text{pulsos}} \cdot \frac{60[s]}{1[min]}}{k} \quad (22)$$

Para el caso del cálculo del controlador PID, se hizo uso de métodos numéricos para la representación de la parte integral y derivativa. En el programa de C#, se proponen alguno algoritmos en [14] [21] [22] [23], como se muestra en (23) y (24):

$$k_i \int e(t) dt \approx k_i \sum_{k=1}^n e(k) t_m \quad (23)$$

$$k_d \frac{de(t)}{dt} \approx k_d \frac{e(k) - e(k-1)}{t_m} \quad (24)$$

Para el caso específico de la parte derivativa hay varias alternativas para disminuir picos por parte de la parte derivativa, siendo los siguientes:

- Cambiando el parámetro de la pendiente de la curva, esto quiere decir que en lugar de medir la diferencia de errores, se medirá la diferencia de salidas (y), esto atenúa los picos (25), pero ocurre un gran problema al recalculer el controlador PID siendo poco factible para este proyecto.

$$k_d \frac{dy(t)}{dt} \approx k_d \frac{y(k) - y(k-1)}{t_m} \quad (25)$$

- Agregando un filtro EMA propuesto en [16] para evitar picos de medición (26). Sin embargo, ofrece un desfase de señal, retardando el sistema y alterando de una manera significativa el proceso.

$$A_n = \alpha M + (1 - \alpha) A_{n-1} \quad (26)$$

donde α es el coeficiente de filtrado, el cual varía entre 0 y 1, M es la medición tomada y A_n es la medición nueva o alterada en la iteración actual.

- La manera más usual de atenuar los picos de la derivada, es aplicar un filtro de coeficiente derivativo [24] [25], el cual agrega un coeficiente N a la función de transferencia derivativa (27) de la siguiente manera:

$$k_d \frac{Ns}{1 + N\frac{1}{s}} = k_d \frac{Ns}{s + N} \quad (27)$$

Siendo aproximada por métodos numéricos (28) de la siguiente manera:

$$k_d \frac{Ns}{s + N} = D \approx N(k_d e(k) - t_m \sum_{k=1}^n D) \quad (28)$$

9. Resultados y Discusión

9.1. Parámetros del Prototipo Experimental (prueba de escalón)

Mediante la metodología expuesta, se obtuvieron los siguientes resultados sobre los parámetros del prototipo experimental, a través del data-logger que son mostrados en la Tabla 2 a continuación:

Símbolo	Valor
K	7.613
τ	0.14
L	0.0095

Tabla 2: Resultados de parámetros de la planta.

Como se puede observar, se obtuvo una proporción de la velocidad angular con respecto al voltaje aplicado en el motor. Obteniendo una función de transferencia de primer orden (ecuación (3)), a través de la prueba de escalón al sistema y sustituida en la ecuación (29), con los parámetros de la Tabla 2.

$$\frac{Y(s)}{U(s)} = \frac{7.613e^{-0.0095s}}{0.14s + 1} \quad (29)$$

Al tabular la ecuación (29) se obtuvo la Figura 28 donde se muestra el modelo aproximado al real.

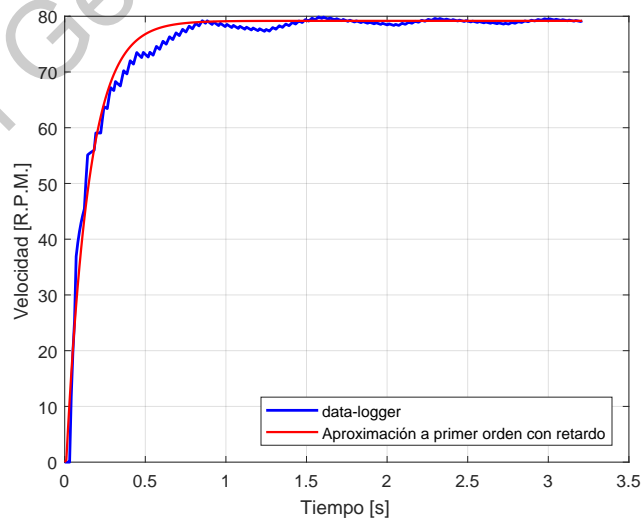


Figura 28: Aproximación del sistema a un primer orden con retardo.

Cabe resaltar que el parámetro de retardo (L) es muy pequeño, por lo que se observó su comportamiento sin dicho retardo y se obtuvo la Figura 29:

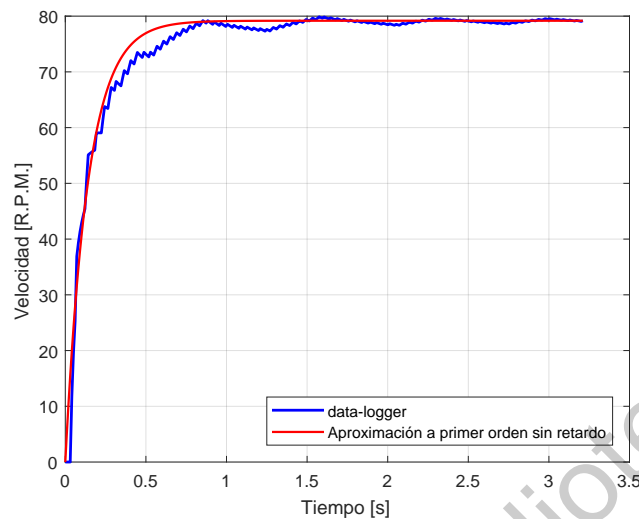


Figura 29: Aproximación del sistema a un primer orden sin retardo.

Por lo que esto permite establecer la siguiente aproximación (30) a través de simple inspección:

$$\frac{7.613}{0.14s + 1} \approx \frac{7.613e^{-0.0095s}}{0.14s + 1} \quad (30)$$

Lo que significa que el motor tiene un tiempo de reacción (retardo) L muy corto debido a la naturaleza del propio sistema, pues en un motor generalmente se debe aplicar un voltaje que logre romper las barreras de fricción estática y fricción dinámica, provocando que reaccione de una manera muy pronta, esto debido a la posibilidad de variar la tensión con mucha rapidez. Así como también una constante de tiempo de subida τ pequeña. Siendo la función de transferencia sin retardo la usada para la sintonización por algoritmo de ED, y usando la función de transferencia con retardo para la sintonización por Ziegler-Nichols debido a que la diferencia entre ambas funciones de transferencia de la ecuación (30) son prácticamente despreciables.

Más adelante se muestra un diagrama de Bode (véase Figura 30) de la respuesta del motor, donde se observa un ángulo de fase β_{ω_1} en el cruce con 0 DB de -82.5° . Esto significa que el sistema por si solo en lazo cerrado es estable ($k_p = 1, k_i = 0, k_d = 0$), para un tiempo $t \rightarrow \infty$ mediante la aplicación del teorema del valor final, siempre y cuando el cruce con 0 DB sea mayor a -180° , esto se puede conocer a través del margen de fase k_f en la ecuación (31).

$$k_f = -180 - \beta_{\omega_1} \quad (31)$$

donde k_f es el margen de fase y β_{ω_1} es el ángulo de fase a la frecuencia de cruce de ganancia (0 DB).

Este desfase es debido a que contiene un polo complejo negativo, lo que genera que la fase sea negativa, puesto que el polo en tiempo continuo representa una integral (integrador, polo en 0) que posee un desfase de -90° . Para este caso, en el motor se tiene un polo más alejado de 0 por lo que el desfase se aproxima mas a 0° , permitiendo un margen de fase $k_f = 97.5^\circ$ más amplio para controlar el sistema.

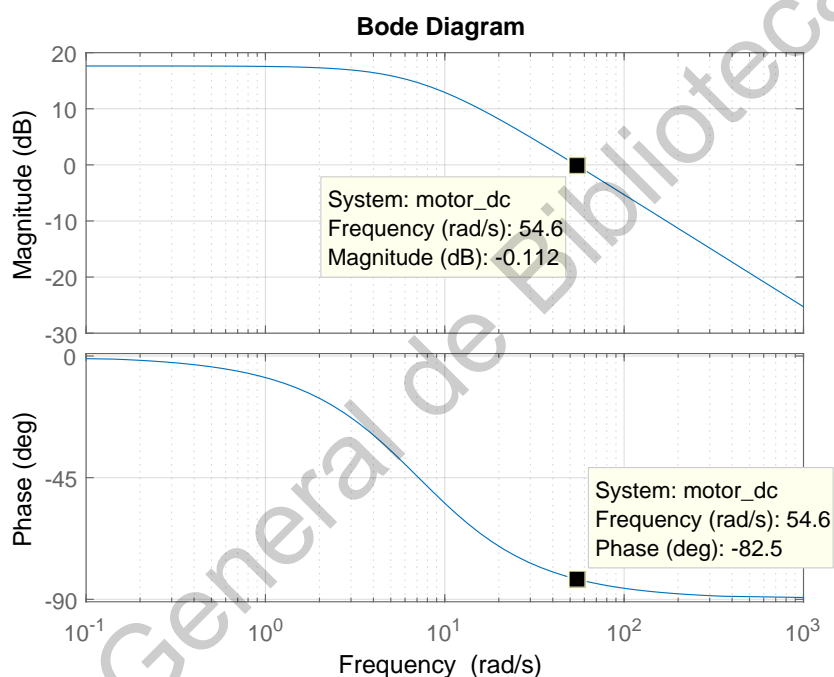


Figura 30: Diagrama de Bode de respuesta en frecuencia del motor 625500/C.

9.2. Parámetros y Respuesta del Controlador PID por Ziegler-Nichols

Los parámetros obtenidos para el controlador PID sintonizado por Ziegler-Nichols se muestran en la Tabla 3, donde se hace uso de la Tabla 1 como referencia para calcular los valores de cada constante.

Se realizó mediante el data-logger en conjunto con la interfaz gráfica, una recopilación de datos a partir de un archivo .TXT, para posteriormente ser expuestos en la Figura 31.

Símbolo	Valor
T_i	0.018
T_d	0.0045
k_p	18.667
k_i	1037.0374
k_d	0.084

Tabla 3: Ganancias del controlador PID por Ziegler-Nichols.

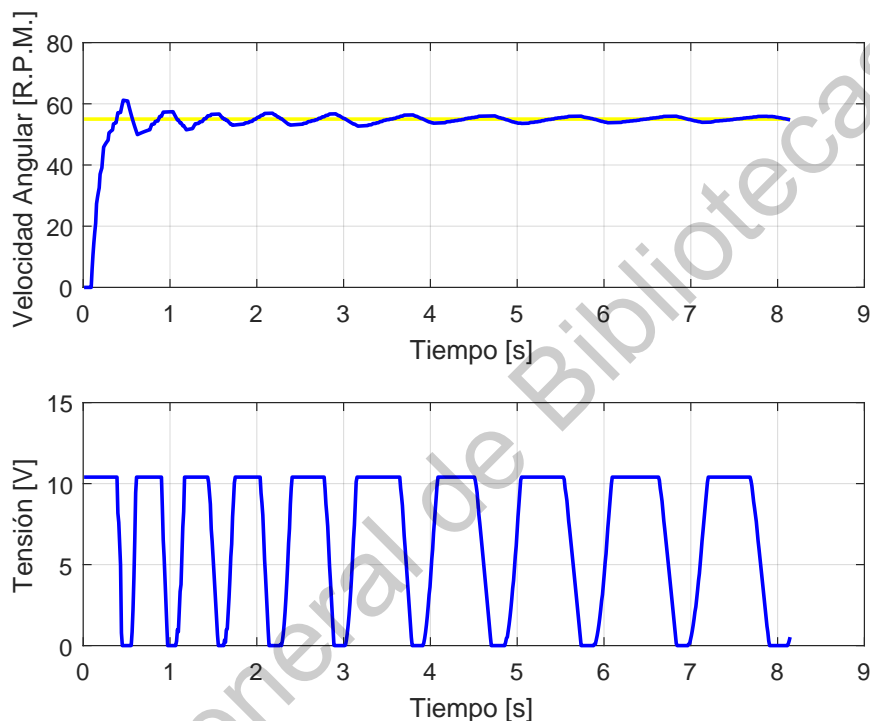


Figura 31: Respuesta del controlador con ganancias obtenidas por Ziegler-Nichols.

Como se observa en la Tabla 3, se calcularon las ganancias para un controlador PID a partir de los tiempos integrales y derivativos (ecuaciones (18)-(20)), donde se obtuvieron ganancias muy altas afectando el comportamiento del motor, pues al ser un sistema muy rápido (L muy pequeña, poco retardo). Esto permite que en el método se calcule una ganancia proporcional alta, amplificando el ruido y las oscilaciones de la señal de salida, que se ven aún más afectadas por algún error o ruido en la medición del sensor. Además de poseer una ganancia integral muy alta, generando inestabilidad y cambios muy bruscos del actuador (puente H), pudiendo dañarlo y causar picos no deseados dentro del sistema (Figura 31). Por su parte, la ganancia derivativa es muy pequeña para lograr causar un efecto

de amortiguamiento del sistema. Por lo que se recomienda siempre al usar este método, realizar ajustes manuales de las ganancias con base en la experiencia del operador, así como también se necesitará de un simulador o en su defecto, de una cantidad de tiempo alta para realizar las pruebas necesarias de manera física en la planta.

A continuación se muestra en la Figura 32, el diagrama de Bode del sistema compensado con parámetros de sintonización obtenidos por el método de Ziegler-Nichols. Donde se observa que $\beta_{\omega_1} = -178^\circ$ y por lo tanto $k_f = -2^\circ$. Lo que implica que se encuentra muy próximo al margen de inestabilidad, sin embargo esto asegura que sea un sistema muy rápido pero con mayores oscilaciones. El mayor problema con esta sintonización es que en caso de que los parámetros del sistema cambien (motor), se podría modificar el margen de fase y llegar a una zona de inestabilidad, lo que significa que no cumpliría con el teorema del valor final para su estabilidad.

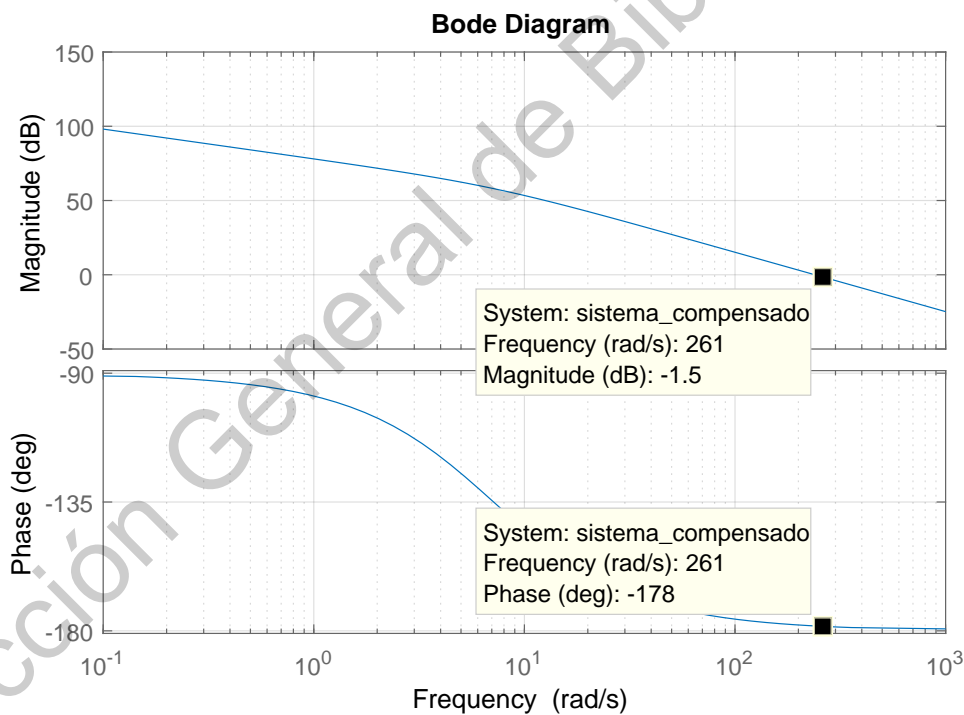


Figura 32: Diagrama de Bode de respuesta en frecuencia del sistema compensado con ganancias obtenidas por Ziegler-Nichols.

9.3. Parámetros y Respuesta del Controlador PID por Evolución Diferencial

Para el programa de Python de algoritmo de ED, se proponen las siguientes restricciones (véase Tabla 4), de modo que el algoritmo pueda funcionar de una manera más correcta.

Símbolo	Descripción	Valor
n	número de iteraciones	200
μ	factor de mutación	0.4
C_r	probabilidad de cruce	0.56
t_s	tiempo de subida	$\leq 7[s]$
m_p	sobrepaso	$\leq 27[\%]$
sp	valor deseado	55[R.P.M.]
k_p	ganancia proporcional	$.01 \leq k_p \leq 20$
k_i	ganancia integral	$.02 \leq k_i \leq 10$
k_d	ganancia derivativa	$0 \leq k_d \leq 1$

Tabla 4: Restricciones de algoritmo de Evolución Diferencial.

Una vez corrido el algoritmo, se obtuvieron los valores tanto de las ganancias (véase Tabla 5), como de simulación de respuesta al escalón del propio programa con base en las restricciones de la Tabla 4.

Como se observa en la Tabla 5, se indica que según la simulación no habrá un sobrepaso y habrá un error porcentual de .002% en el valor final con respecto al valor deseado, siendo muy aproximado e inclusive considerado despreciable por el sensor. Sin embargo, cabe resaltar que el programa al simular las situaciones ideales alcanza un error porcentual menor al 1% antes del tiempo de subida t_s .

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

Tabla 5: Resultados obtenidos por el algoritmo de Evolución Diferencial.

Obtenidos los valores de interés de la Tabla 5 se realizó la prueba del prototipo experimental con las ganancias k_p , k_i y k_d (véase Figura 33) propuestas por el algoritmo de ED.

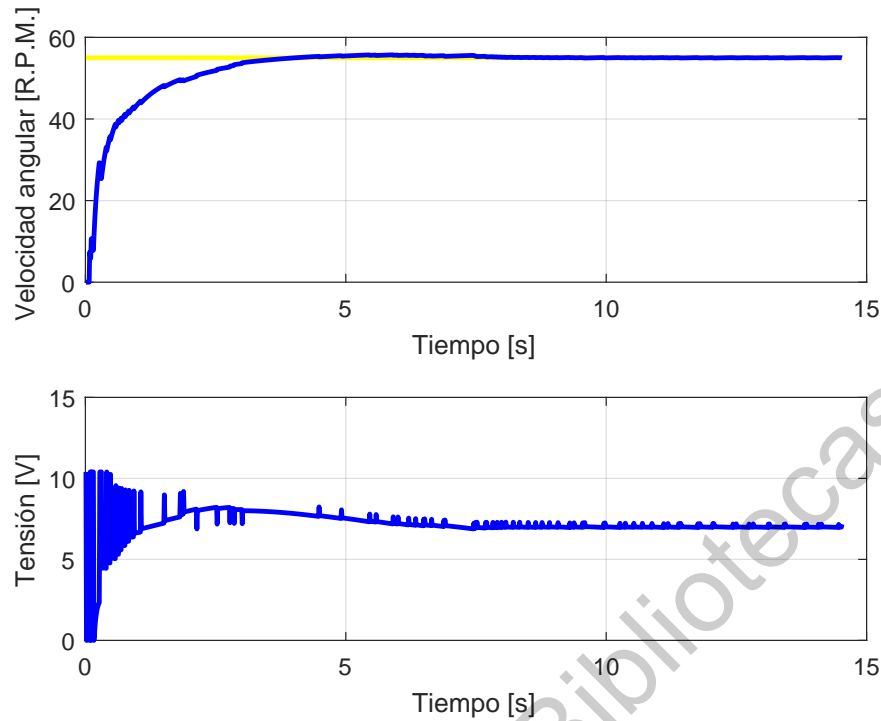


Figura 33: Respuesta del controlador con ganancias obtenidas por el algoritmo.

Como se muestra en la Tabla 6 en comparación con la Tabla 5, se observa que hubo un sobrepaso m_p siendo algo que la simulación no tuvo. Sin embargo, esto se debe a que al no ser un sistema de control totalmente continuo, se puede perder información. También la estrategia de medición permite tener mediciones mas lentas (desfase), debido a la media de medición que se utiliza para evitar picos de velocidad.

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

Tabla 6: Resultados obtenidos por el prototipo experimental.

En la Figura 34 se observa que $\beta_{\omega_1} = -166^\circ$ y por lo tanto $k_f = -14^\circ$, lo que significa un margen de fase más amplio al de la Figura 32, en donde el contro-

lador sintonizado por Evolución Diferencial es más robusto ante cambios en los parámetros del sistema en comparación con el de Ziegler-Nichols. Sin embargo, para lograr un perfecto control de margen de fase, es necesario utilizar un controlador autoajustable, que pueda variar los parámetros del controlador ante cambios más bruscos de la planta.

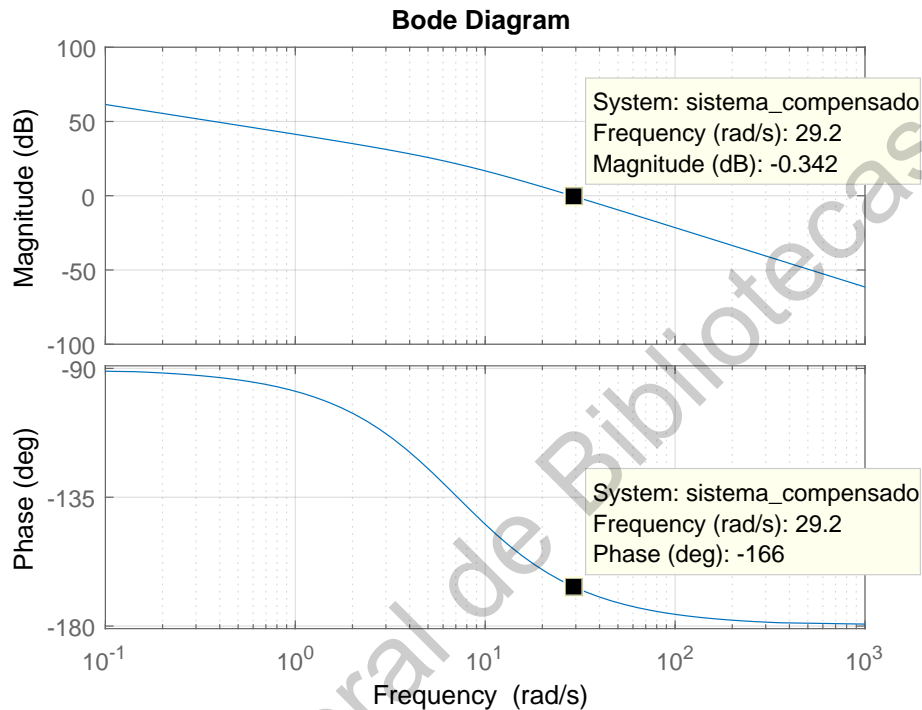


Figura 34: Diagrama de Bode de respuesta en frecuencia del sistema compensado con ganancias obtenidas por el algoritmo de Evolución Diferencial.

En la Figura 35 se observa que el algoritmo sigue una trayectoria (líneas azules) no definida al avanzar la iteración (triángulo rojo iteración $n = 0$ y estrella roja solución final $n = 200$). Esto debido a que se realiza una búsqueda aleatoria dentro del rango de valores dados, además de que al tener un factor de mutación μ y un factor de cruce C_r , para evitar estancamientos en algún valor ya sea máximo o mínimo local. En consecuencia, provoca que al volver a correr el programa se obtendrá una secuencia diferente y un conjunto de ganancias resultantes diferentes, las cuales cumplen con los requerimientos dados con anterioridad (Tabla 4). Una vez dicho esto, se comprueba que existe una gran variedad de valores combinatorios que satisfacen a las restricciones (Tabla 4).

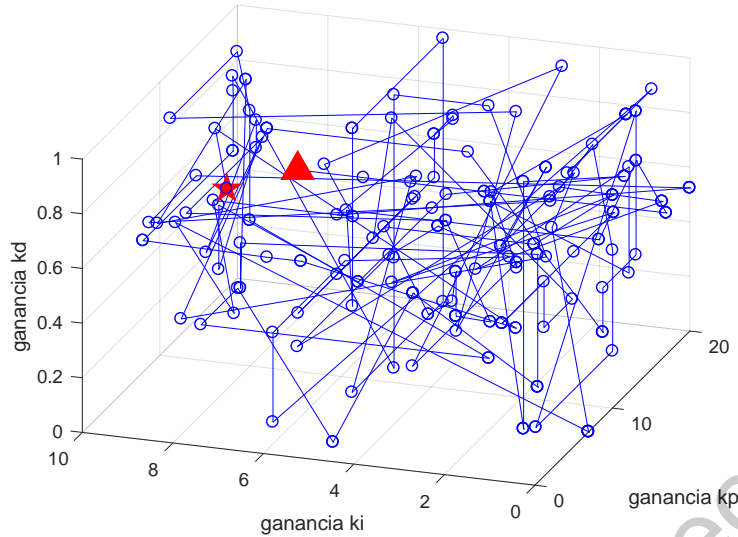


Figura 35: Dispersión y dirección de las iteraciones.

9.4. Saturación y no linealidades en el sistema de control

Haciendo uso de software especializado para simulación de sistemas de control dentro del entorno de MATLAB (Simulink), se realizó una simulación del sistema compensado tomando en cuenta la saturación del sistema. Por lo que el sistema de control mediante el prototipo experimental (Figura 18) no puede proporcionar más de 10.3V en su etapa de potencia. Permitiendo una limitación de tensión para proteger el motor de picos de corriente y tensión. Todo esto, con la finalidad de proteger a los componentes que sufren estrés durante la prueba del sistema de control compensado [23].

Una vez dicho esto, se observa en la Figura 36 la respuesta del sistema con esta limitación, como se observa en la segunda gráfica, en ningún momento se exceden los 10.3V y conforme el tiempo transcurre, la tensión llega a un estado estacionario.

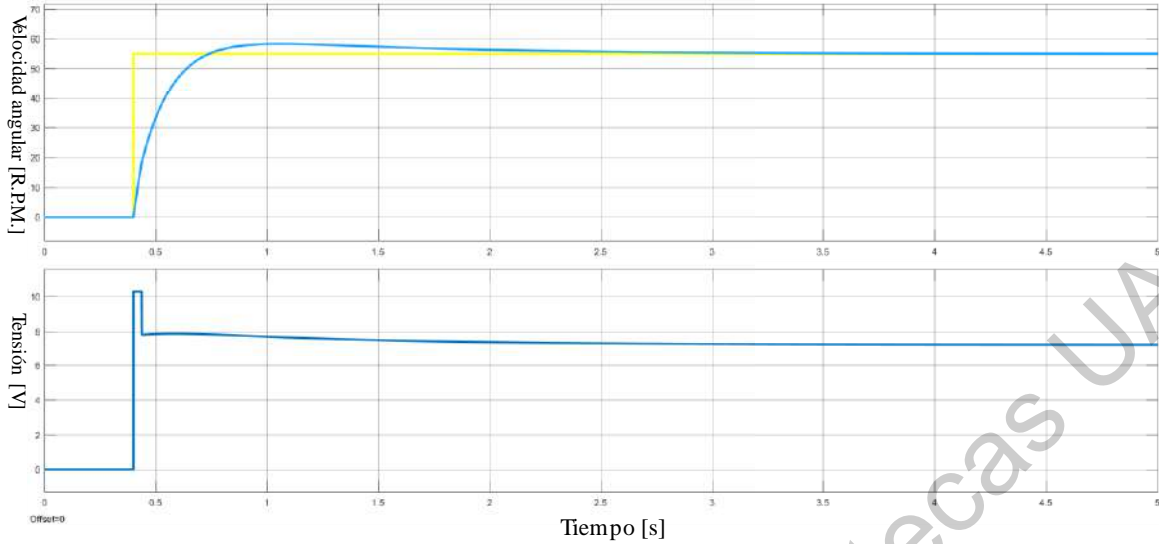


Figura 36: Respuesta del sistema compensado con saturación de tensión.

Por su parte la gráfica de velocidad angular contra tiempo muestra un pequeño sobrepaso como se obtuvo en la Tabla 6 y en la Figura 33, aproximándose de una manera mas cercana a la realidad. Esto provoca que la saturación se transforme en un problema en la práctica para la estimación de otras referencias (Setpoints), modificando las respuestas según este cambio. En consecuencia, el sistema compensado es un sistema no lineal que variará su respuesta según el Setpoint empleado. Sin embargo, en algunos rangos de operación es posible conservar la linealidad del sistema por la naturaleza del propio motor [26].

Siendo realizadas varias pruebas para la sintonización por Evolución diferencial para obtener distintos parámetros, utilizando de manera aleatoria la 5ta prueba del controlador, observando pocos cambios en la respuesta. Sin embargo para la prueba de Ziegler-Nichols únicamente se realizó una prueba, puesto que los parámetros del controlador no se podían cambiar según el cálculo obtenido.

10. Conclusiones

Se construyó un prototipo experimental (data-logger) para obtener las mediciones de un motor de corriente directa con encoder, con material reciclado de proyectos de materias de semestres cursados con anterioridad en la carrera, así como también el software embebido que permitirá todas las operaciones necesarias para hacer funcionar el sistema completo.

Los problemas de programación fueron debidos al uso de nuevas funciones y a la integración con el data-logger, causando problemas en la sincronización de tiempos, todo esto para que en ambas partes se trabajara de una manera síncrona. Posteriormente se tuvieron problemas con los falsos contactos y ruido externo en los cables, siendo solucionado con la fabricación de las placas PCB.

En este trabajo, se optó por un controlador PID para controlar la velocidad del motor donde al ser aplicadas ambas técnicas de sintonización, se obtuvieron resultados fructíferos con sus respectivas características de desempeño. Por un lado se tuvo el sistema compensado con el controlador sintonizado con el método de Ziegler-Nichols, que obtuvo un resultado satisfactorio para las restricciones solicitadas. Sin embargo, se observó en el voltaje proporcionado al motor una variación alta, pudiendo dañar el actuador (puente H) y el motor por las fluctuaciones tan altas de voltaje en estado estacionario. Por otro lado, se realizó la sintonización mediante un programa en Python, que únicamente requería de las restricciones para poder operar. Para después obtener una combinación de ganancias con sus respectivos datos de salida, siendo este último método más cómodo y rápido para el operador o el encargado del proceso, en comparación a métodos tradicionales como Ziegler-Nichols y otros más.

Cabe destacar que el programa de sintonización en Python fue realizado con un enfoque en los motores de corriente directa, aunque puede ser utilizado para cualquier sistema de control lineal de n orden que se desee controlar (e.g. sistema térmico, sistema ball and beam, etc.).

Para futuros trabajos se podría mejorar el sistema de medición, el microcontrolador usado y en cuestiones de software, mejorar el algoritmo de evolución diferencial mediante optimización de código, o también uso de inteligencia artificial para resolución de este tipo de problemas.

Bibliografía

- [1] G.A. Leal. Importancia de la Instrumentación y el Control en el Sector Industrial de la Actualidad. <https://www.lymcapacitacion.com/blog/16158/instcontrol>, 2018.
- [2] I. Chiha, H. Liouane y N.Liouane . 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 (IREMOS)*, 5(2), 2012.
- [3] Rainer Storn y Kenneth Price . Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, 1997.
- [4] V. Vishal, V. Kumar, K. P. S. Rana, and P. Mishra. Comparative study of some optimization techniques applied to dc motor control. pages 1342–1347, 2014.
- [5] Abdelhakim Idir and Madjid Kidouche and Yassine Bensafia and Khair Khettab y Sid Ahmed Tadjer . Speed Control of DC Motor Using PID and FOPID Controllers Based on Differential Evolution and PSO). *International Journal of Intelligent Engineering Systems*, 2018.
- [6] B. Monthary and S. Panda y P.K. Hota. Differential evolution algorithm based automatic generation control for interconnected power systems with non-linearity. *Alexandria Engineering Journal*, 2014.
- [7] K. Ogata. *Ingeniería de control moderna*. Pearson Educación, 2010.
- [8] Ma. Guadalupe Martínez. Cómputo evolutivo. <https://ccc.inaoep.mx/~a.morales/EC/pdf/ClaseED.pdf>, 2019.
- [9] L.V. Santana. *Un Algoritmo Basado en Evolución Diferencial para Resolver Problemas Multiobjetivo*. CINVESTAV, 2004.
- [10] V. Mazzone. Controladores PID. <https://www-eng.newcastle.edu.au/~jhb519/teaching/caut1/Apuntes/PID.pdf>, 2002.
- [11] T. Sibalija. Metaheuristic algorithms in industrial process optimisation: Performance, comparison and recommendations. *Communications in Computer and Information Science*, 1198:270–283, 2020.

- [12] L.C.T. Bezerra, M. López-Ibáñez, and T. Stützle. Automatic configuration of multi-objective optimizers and multi-objective configuration. *Studies in Computational Intelligence*, 833:69–92, 2020.
- [13] M. G. Villarreal-Cervantes, A. Rodríguez-Molina, C. García-Mendoza, O. Peñaloza-Mejía, and G. Sepúlveda-Cervantes. Multi-objective on-line optimization approach for the dc motor controller tuning using differential evolution. *IEEE Access*, 5:20393–20407, 2017.
- [14] V.M. Hernández Guzmán y R. Silva Ortigoza. *Automatic Control with Experiments*. Springer, 2019.
- [15] S. Moya. Controlador PID. *ISA InTech México*, 2017.
- [16] A. Aguado y M. Martinez. *Identificación y control adaptativo*. Prentice Hall, 2003.
- [17] ECURED. Algoritmo de búsqueda. <https://www.ecured.cu/Algoritmodebusqueda>, 2012.
- [18] Z. Adak, M.Ö. Arıoğlu Akan, and S. Bulkan. Multiprocessor open shop problem: literature review and future directions. *Journal of Combinatorial Optimization*, 2020.
- [19] *Matlab*. Convert model from discrete to continuous time - MATLAB d2c. <https://www.mathworks.com/help/control/ref/d2c.html>, 2020.
- [20] Angel Romero, Antonio J. Muñoz Ramírez y J. Manuel Gómez de Gabriel. Realimentación de velocidad con encoders de baja resolución en Simulink. *Comité Español de Automática de la IFAC (CEA-IFAC)*, 2015.
- [21] Murray R. Spiegel, Seymour Lipschutz y John Liu. *FÓRMULAS Y TABLAS DE MATEMÁTICA APLICADA (Español) 5ta edición*. McGraw-Hill, 2020.
- [22] V.M. Hernández Guzmán, R. Silva Ortigoza y R.V. Carrillo Serrano. *Control Automático: Teoría de diseño, Construcción de Prototipos, Modelado, Identificación y Pruebas Experimentales*. Colección CIDETEC del Instituto Politécnico Nacional, 2013.
- [23] M. Nasreldin, A. Abdalghar, A. Alfatih y A. Abdelrahman. IMPLEMENTATION OF A PID CONTROL SYSTEM ON MICROCONTROLLER (DC

MOTOR CASE STUDY). *International Conference on Communication, Control, Computing and Electronics Engineering*, 2017.

- [24] *Matlab*. Filter coefficient (N) in PID block? <https://www.mathworks.com/matlabcentral/answers/63762-filter-coefficient-n-in-pid-block>, 2013.
- [25] *Matlab*. PID Controller. <https://www.mathworks.com/help/simulink/slref/pidcontroller.html#br9ejg0-18>, 2020.
- [26] Karl J. Åström y Tore Hägglund. *Control PID avanzado*. Prentice Hall, 2009.

Dirección General de Bibliotecas UAQ