Estudio comparativo de algoritmos de control PID clásico para el control angular de un brazo electromecánico

Comparative study of classical PID control algorithms for the angular control of an electromechanical arm

Hernán Astudillo Roblero¹

José Gallardo Arancibia^{1*}

Claudio Ayala Bravo²

Recibido 15 de julio de 2020, aceptado 1 de septiembre de 2020

Received: July 15, 2020

Accepted: September 1, 2020

RESUMEN

De acuerdo a diversos reportes, se estima que alrededor del 70% de los lazos de control industrial utilizan los clásicos algoritmos de control PID (Proporcional Integral Derivativo), el más utilizado de ellos es el PID Paralelo, pero existen variantes tales como los PID seriales, I-PD y PI-D, cada uno de ellos posee diferentes propiedades. En este estudio se evalúa el desempeño de estas diversas implementaciones en una planta de control angular que consta de un brazo manipulado por un motor Brushless. Para el control angular del brazo, se ensayaron las diversas implementaciones de controladores PID reportados en la literatura, obteniendo el valor de sus parámetros en función del modelo de la planta. Para lograr diseñar los algoritmos fue necesario discretizar las ecuaciones diferenciales que definen a los controladores PID, transformando sus ecuaciones diferenciales a ecuaciones de diferencias las que posteriormente se programaron en C++. En una primera instancia los esquemas de control se implementaron en modo simulación para posteriormente implementarlos en tiempo real. Los resultados logrados fueron bastante similares y la posterior evaluación de las diversas implementaciones ensayadas demostró que el algoritmo PID paralelo cumplió más eficazmente los objetivos de control.

Palabras clave: Control automático, algoritmos de control, control angular, control PID.

ABSTRACT

According to various reports, it is estimated that around 70% of industrial control loops use the classical PID control algorithms (Proportional Integral Derivative), the most used of them is the Parallel PID, but there are variants such as serial PID, I-PD and PI-D, each of them has different properties. In this study, the performance of these diverse implementations is evaluated in an angular control plant that consists of an arm manipulated by a Brushless motor. For the angular control of the arm, the various implementations of PID controllers reported in the literature were tested, obtaining the value of their parameters according to the model of the plant. In order to design the algorithms it was necessary to discretize the differential equations that define the PID controllers, transforming their differential equations to equations of differences which were later programmed in C++. In a first instance, the control schemes were implemented in simulation mode to subsequently implement them in real time. The results obtained were quite similar and the subsequent evaluation of the various implementations tested showed that the parallel PID algorithm fulfilled the control objectives more effectively.

Keywords: Automatic control, control algorithms, angular control, PID control.

¹ Universidad Católica del Norte. Departamento de Ingeniería de Sistemas y Computación. Antofagasta, Chile. E-mail: har004@alumnos.ucn.cl; jgallardo@ucn.cl

² Universidad de Antofagasta. Departamento de Ingeniería Eléctrica. Antofagasta, Chile. E-mail: claudio.ayala@uantof.cl

* Autor de correspondencia: jgallardo@ucn.cl

INTRODUCCIÓN

Es posible encontrar diversos reportes que mencionan que cerca 70% de los procesos industriales aun utilizan para el control de sus procesos, los clásicos algoritmos PID (Proporcional, Integral, Derivativo) [1], así como también la existencia y aplicación de diversas variantes del mencionado algoritmo. En consideración a lo ya mencionado, el trabajo que se presenta en este reporte es la implementación y evaluación de las diversas variantes del controlador PID en el control de una planta que presenta características que dificultan su control como por ejemplo un elevado tiempo muerto.

La planta a controlar posee un brazo electromecánico, en uno de sus extremos sostiene una hélice acoplada a un motor Brushless, además la planta posee un controlador electrónico de velocidad, que en conjunto con una placa Arduino Uno, le permite controlar el ángulo que forma el brazo con la barra vertical que se conecta con el extremo superior del brazo. Para elegir el rango de control, se modeló la planta y se comporta de manera lineal entre los 0° y 45°.

En las siguientes secciones se describen: un algoritmo de control PID, sus diversas variantes, el modelado de la planta a controlar, la programación de las diversas implementaciones, su experimentación en modo de simulación, su experimentación en tiempo real y finalmente se realiza una evaluación del comportamiento de las diversas implementaciones.

Figura 1. Planta de control angular.

CONTROLADOR PID

Un controlador PID [2] es un controlador, basado en la combinación de las acciones proporcional, integral y derivativa, de tal forma que se logren características dinámicas estables en la variable que se requiera controlar, cada una de las acciones por sí solas, definen una determinada respuesta en la salida de un proceso cuando éste es sometido a una excitación.

La mayor parte de los controladores que se utilizan en la actualidad corresponde a los controladores PID, y se advierte que existen muchas formas de implementarlo, entre las cuales se encuentran los interactivos y los no interactivos, de los cuales el 95% son solamente PI [5]. La ecuación 1, define la representación de un controlador PID.

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

Donde:

- $e(t)$: Es la señal de error entre la consigna y la salida de la planta.
- $u(t)$: Es la salida del controlador
- K_p : La ganancia proporcional.
- T_i : La constante de tiempo integral.
- T_d : La constante de tiempo derivativa.

CONTROLADOR PID 2DOF (INDUSTRIAL)

Un esquema de control PID 2DOF¹³ (Two-Degree-of-Freedom) (ec. 2), también denominado PID industrial, es un algoritmo de control que consiste en un controlador que puede configurarse según se introduzcan en él, un par de parámetros (β y γ) cuyas combinaciones en sus valores originan implementaciones diferentes (controladores PI-D e I-PD) los que se utilizan para mitigar la influencia nociva de cambios frecuentes en la señal de consigna [10].

$$u(s) = k_p (\beta r - y) + \frac{k_i}{s} (r - y) + k_d s (\gamma r - \gamma) \quad (2)$$

Donde:

- K_p : ganancia proporcional.
- K_i : ganancia integral.
- K_d : ganancia derivativa.

- β : factor de peso para consigna en término proporcional.
- $\beta = 1$ implica acción proporcional sobre el error.
- $\beta = 0$ implica acción proporcional sobre la realimentación del controlador.
- γ : factor de peso para consigna en término derivativo.
- $\gamma = 1$ implica acción derivativa sobre el error.
- $\gamma = 0$ implica acción derivativa sobre la realimentación del controlador.
- Si $\beta = 1$ y $\gamma = 1$, entonces se obtiene un PID en paralelo.
- Si $\beta = 1$ y $\gamma = 0$, entonces se obtiene un PI-D.
- Si $\beta = 0$ y $\gamma = 0$, entonces se obtiene un I-PD.

DISCRETIZACIÓN

Para diseñar reguladores digitales existen dos métodos, el primero es trabajar directamente en el dominio de la transformada “z”, y el segundo consiste en diseñar en “s” (transformada de Laplace), y luego pasar a “z”. En este trabajo se utilizó el segundo método [9]. Por otro lado, la versión discreta de un sistema continuo dispone de dos opciones, la primera es el método exacto, que tiene en cuenta la relación que existe entre “z” y “s” de la siguiente forma $z = e^{sT}$, el segundo, la de los métodos aproximados, como por ejemplo el de la derivada, el cual es utilizado más adelante para obtener la versión discreta de los algoritmos PID. Este método consiste en aproximar la derivada por la pendiente de la recta que pasa por dos muestras consecutivas, con lo cual se tiene la ecuación 3.

$$\frac{dy(t)}{dt} \rightarrow \frac{y_k - y_{k-1}}{T} \quad (3)$$

Aplicando Laplace y transformada Z, la ecuación 3 se convierte a la ecuación 4:

$$sY(s) \rightarrow \frac{1-z^{-1}}{T} Y(Z) \quad (4)$$

De acuerdo con lo anterior, se puede obtener un PID discreto a partir de la versión continua (ec. 5),

asumiendo de la ec. 4, que $s = \frac{1-z^{-1}}{T}$.

$$\frac{U(s)}{E(s)} = k_p + \frac{k_i}{s} + K_d s \quad (5)$$

Reemplazando en la ec. 5, $s = \frac{1-z^{-1}}{T}$ y simplificando la ecuación, finalmente se obtiene la función de transferencia de un PID discreto, donde T es el tiempo de muestreo (ver ecuación 6).

$$\frac{U(z)}{E(z)} = \frac{k_p + k_i T + \frac{k_d}{T} - \left(k_p + 2\frac{k_d}{T}\right)z^{-1} + \frac{k_d}{T}z^{-2}}{1-z^{-1}} \quad (6)$$

Si se aplica la transformada inversa a la ecuación 6, se obtiene su ecuación de diferencias (ec. 7), que finalmente es la que se programa en un computador digital.

$$u_k = b_0 e_k + b_1 e_{k-1} + b_2 e_{k-2} + u_{k-1} \quad (7)$$

Donde:

$$b_0 = k_p + k_i T + \frac{k_d}{T}; b_1 = -\left(k_p + 2\frac{k_d}{T}\right); b_2 = \frac{k_d}{T} \quad (8)$$

En los siguientes párrafos se describirán las implementaciones más comunes [5].

PID PARALELO (NO INTERACTIVO)

Se conoce con el nombre de algoritmo no interactivo de control PID, al algoritmo clásico en el que las acciones de control integral y derivativo son independientes, aunque exista un parámetro del controlador (la ganancia proporcional K_p) que afecte a las tres acciones (proporcional, integral y derivativa). Está considerado como el estándar por la ISA (Instrument Society of America), es el más citado en la bibliografía y el más utilizado actualmente [3], este algoritmo se presenta a nivel de diagrama de bloques en la Figura 2, y es representado por la ecuación diferencial 9 [4].

$$u(t) = k_p e(t) + k_i \int e(t) dt + k_d \frac{de(t)}{dt} \quad (9)$$

$$\frac{U(s)}{E(s)} = k_p + \frac{k_i}{s} + k_d s \quad (10)$$

La ecuación 10 representa la función de transferencia del controlador en estudio, y la ecuación 11 la función de transferencia en términos de la transformada Z.

Figura 2. Diagrama de bloques de un PID discreto.

$$\frac{U(z)}{E(z)} = \frac{k_p + k_i T + \frac{k_d}{T} - \left(k_p + 2\frac{k_d}{T}\right)z^{-1} + \frac{k_d}{T}z^{-2}}{1 - z^{-1}} \quad (11)$$

$$u_k = b_0 e_k + b_1 e_{k-1} + b_2 e_{k-2} + u_{k-1} \quad (12)$$

$$b_0 = k_p + k_i T + \frac{k_d}{T}; b_1 = -\left(k_p + 2\frac{k_d}{T}\right); b_2 = \frac{k_d}{T} \quad (13)$$

Finalmente, la ecuación 11 es expresada como una ecuación de diferencias (ecuación 12) expresión que es posible programar. En el resto de las diversas implementaciones en estudio, se sigue un procedimiento similar para obtener una expresión matemática que pueda ser directamente programada en un computador digital.

PID INTERACTIVO (ESTÁNDAR)

El algoritmo interactivo o algoritmo serie (Figura 3 y ecuaciones 14 a 18) fue empleado generalmente

en los antiguos reguladores analógicos [6]. Con el calificativo de interactivo se quiere recalcar que la modificación de cualquiera de los parámetros del controlador K_i o K_d afecta a las tres acciones (proporcional, integral y derivativa), tal como se entienden dichas acciones en el algoritmo clásico. Este algoritmo surgió como una posibilidad de realizar control PID analógico con dos amplificadores operacionales. A diferencia del no interactivo que requiere el uso de tres, es por ello que, en la gran mayoría de los reguladores analógicos, buscando el ahorro económico se usaba el algoritmo interactivo.

En la actualidad, aunque ya no existen inconvenientes en la realización digital del control PID no interactivo, algunos fabricantes siguen ofreciéndole [6]. De esta forma, se cubre la demanda de quienes desean mantener la validez de las técnicas de ajuste habituales en controladores analógicos, y sacar el máximo provecho a la experiencia anterior de los operadores de planta.

Figura 3. Diagrama de bloques de un PID interactivo.

$$u(t) = k_p e(t) + k_p k_i \int e(t) dt + k_p k_d \frac{de(t)}{dt} \quad (14)$$

$$\frac{U(s)}{E(s)} = K_p + k_p \frac{K_i}{s} + k_p K_d s \quad (15)$$

$$\frac{U(z)}{E(z)} = \frac{k_p + k_p k_i T + \frac{k_p k_d}{T} - \left(k_p + 2 \frac{k_p k_d}{T}\right) z^{-1} + k_p \frac{k_d}{T} z^{-2}}{1 - z^{-1}} \quad (16)$$

$$u_k = b_0 e_k + b_1 e_{k-1} + b_2 e_{k-2} + u_{k-1} \quad (17)$$

$$b_0 = k_p + k_p k_i T + \frac{k_p k_d}{T}; b_1 = -\left(k_p + 2 \frac{k_p k_d}{T}\right); b_2 = \frac{k_p k_d}{T} \quad (18)$$

PI-D

Este tipo de controladores, con la acción derivativa en el lazo de realimentación, son comúnmente utilizados para evitar los overshoot producidos por errores bruscos, al aplicar la derivación en la realimentación se evita derivar la referencia. En la Figura 4 se representa el diagrama de bloques de un controlador PI-D, y su dinámica en las ecuaciones 19 a 23.

$$u(t) = k_p e(t) + k_i \int e(t) dt - k_d \frac{dy(t)}{dt} \quad (19)$$

$$u(t) = -k_p y(t) + \frac{k_i}{s} e(t) - k_d y(t) s \quad (20)$$

$$\frac{U(z)}{E(z)} = (k_p + K_i T) y_i(t) - k_p y_i(t) z^{-1} + \left(-k_p - K_i T - \frac{k_d}{T}\right) y(t) + \left(k_p + \frac{2k_d}{T}\right) y(t) z^{-1} - \frac{k_d}{T} y(t) z^{-2} \quad (21)$$

$$u_k = b_0 y_{ik} + b_1 y_{ik-1} + b_2 y_k + b_3 y_{k-1} + b_4 y_{k-2} + u_{k-1} \quad (22)$$

$$b_0 = (k_p + K_i T); b_1 = (-k_p); b_2 = \left(-k_p - k_i T - \frac{k_d}{T}\right); b_3 = \left(k_p + \frac{2k_d}{T}\right); b_4 = -\frac{k_d}{T} \quad (23)$$

I-PD

Esta variante se utiliza cuando no es necesario seguir los cambios de setpoint de manera rápida. Pero sí se necesita obtener un control estable de la variable controlada. La Figura 5 representa el diagrama de bloques de un controlador tipo I-PD, y las ecuaciones 24 a 28 describen las relaciones entre la señal de error y la señal de salida del controlador.

$$u(t) = -K_p y(t) + k_i \int e(t) dt - k_d \frac{dy(t)}{dt} \quad (24)$$

$$u(t) = -k_p y(t) + \frac{k_i}{s} e(t) - k_d y(t) s \quad (25)$$

Figura 4. Diagrama de bloques de un PI-D.

Figura 5. Diagrama de bloques de un I-PD.

$$\frac{U(z)}{E(z)} = (k_i T) Y_i(t) + \left(-k_p - k_i T + \frac{k_d}{T} \right) \quad (26)$$

$$y(t) + \left(k_p + \frac{2k_d}{T} \right) y(t)z^{-1} - \frac{k_d}{T} y(t)z^{-2}$$

$$u_k = b_0 y_{ik} + b_1 y_k + b_2 y_{k-1} + b_3 y_{k-2} + u_{k-1} \quad (27)$$

$$b_0 = (k_i T); b_1 = \left(-k_p - K_i T + \frac{k_d}{T} \right); \quad (28)$$

$$b_2 = \left(k_p + \frac{2k_d}{T} \right); b_3 = -\frac{k_d}{T}$$

de la misma, sobre esta base, y pendiendo de un soporte vertical se ubica un brazo de aluminio que posee un motor Brushless en su extremo inferior. En el extremo superior del brazo está instalado un sensor que se encarga de medir la posición del brazo con respecto al soporte vertical.

Al interior de la base se encuentra el ESC (controlador electrónico de velocidad), además de un sistema de procesamiento de propósitos específicos denominado Arduino, y la fuente de poder que alimenta los diversos componentes. El motor Brushless está conectado al ESC y al Arduino. La Figura 6, ilustra el diagrama de conexiones.

DESCRIPCIÓN DE LA PLANTA A EXPERIMENTAR

La planta denominada brazo electromecánico de control angular está montada en una base, en la cual se encuentran ubicados los componentes eléctricos

RESULTADOS OBTENIDOS

Lo primero que se hizo, fue verificar el correcto funcionamiento del instrumento de medición, en un principio una IMU, sin embargo como se advierte

Figura 6. Esquemático de la planta de control angular.

en la Figura 7, el ruido generado por el motor no permitía obtener una buena medición por lo que la IMU, fue reemplazada por un potenciómetro lineal.

Una vez reemplazada la IMU, se pudo obtener una señal más limpia como se observa en la Figura 8, en este caso ambas señales, consigna y la señal proveniente del sensor, prácticamente se encuentran superpuestas observándose que la vibración del motor no afecta negativamente en la medición que éste realiza sobre el ángulo de inclinación del brazo electromecánico.

OBTENCIÓN DEL MODELO DE LA DINÁMICA DE LA PLANTA

Posteriormente, fue necesario conocer el comportamiento de la planta respecto de la

alimentación que consiste en una señal PWM proveniente de la placa Arduino. Al variar la señal de alimentación al motor, el desplazamiento angular del brazo es el que se observa en la Figura 9. Se puede ver que la planta posee un comportamiento lineal entre los 0° y los 45° , y posteriormente se observa lineal por tramos, lo que permite definir el rango de control en una determinada zona.

Para obtener el modelo de su dinámica, fue necesario alimentar la planta mediante una entrada de tipo escalón en lazo abierto. Realizando un muestreo de todos los datos que se generan. La curva que obtiene se muestra en la Figura 10. Los datos fueron tomados con un programa escrito en el lenguaje C#, se guardan los datos en un archivo de texto. Observando la respuesta que se obtiene se puede

Figura 7. Medición del desplazamiento angular mediante una IMU.

Figura 8. Medición del desplazamiento angular mediante un potenciómetro.

Figura 9. Comportamiento de la planta a lazo abierto.

Figura 10. Comportamiento de la planta ante una entrada escalón.

considerar que corresponde a la respuesta de un sistema de segundo orden.

Luego los datos se alimentaron a una opción del software Control Station V3.7, que permite en función de los datos ingresados aproximar el modelo de la planta, cuya función de transferencia se ilustra en la ec. 29, más un tiempo muerto significativo utilizando una regresión para un modelo de segundo orden con tiempo muerto.

$$H(s) = \frac{0,1335}{8539.6s^2 + 184.82s + 1} \quad (29)$$

SIMULACIÓN EN MATLAB

Inicialmente y de manera previa a los ensayos en la planta real, se acordó ensayar los algoritmos en modo simulación, considerando el modelo de la planta obtenido. Estos ensayos de simulación

se realizaron mediante la plataforma *simulink* de Matlab. La Figura 11 ilustra el diagrama en bloques del sistema implementado.

Luego de haber determinado los parámetros óptimos para el control en simulación [12], se procede a hacer la comparativa entre los algoritmos (PID estándar, PID paralelo, PI-D e I-PD). Los resultados obtenidos, se muestran en la Figura 12, en la cual se puede observar que las mejores respuestas, son las de los controladores PID paralelo y PI-D, que presentan una mejor respuesta en velocidad de respuesta más un mínimo sobre impulso. El algoritmo I-PD presenta una respuesta demasiado lenta.

COMPARATIVA EN LA PLANTA REAL

Las pruebas de los 4 algoritmos se realizaron, utilizando los parámetros obtenidos mediante la utilidad PID Tuning de Matlab ($K_p = 4.27$,

Figura 11. Controlador PID con tiempo muerto.

Figura 12. Respuestas de las diversas implementaciones PID a una entrada escalón unitario.

$K_i = 0,011763085$ y $K_d = 386.862$). La primera respuesta ante entrada escalón, para la configuración PID paralelo es la realizada inicialmente, la respuesta obtenida es la que se ilustra en la Figura 13, que evidencia una respuesta oscilatoria (inestable).

Al constatar el magro resultado logrado, se procedió a realizar la sintonía [11] fina de los parámetros del controlador, una vez que se obtuvieron los parámetros óptimos para el control se procede a trabajar en la comparativa utilizando los mismos parámetros en cada uno de los controladores.

PID PARALELO

El diagrama de bloques de la configuración PID paralelo implementado, se ilustra en la Figura 14 y la ecuación que lo caracteriza es la 30.

$$u(t) = k_p e(t) + k_i \int e(t) dt + k_d \frac{de(t)}{dt} \quad (30)$$

La Figura 15 representa la respuesta de la planta controlada por una configuración PID paralelo, ante una entrada escalón, como puede observarse el tiempo de establecimiento es de aproximadamente 7,1 segundos y cumple con los objetivos de control (error de estado estacionario nulo, no sobre impulsos y mínimo tiempo de establecimiento) ya que no presenta oscilaciones.

La segunda prueba efectuada, fue introducir cambios en la consigna y el controlador responde de manera adecuada (ver Figura 16), y la tercera prueba fue observar la respuesta (Figura 17) ante perturbaciones introducidas mediante cargas (pesos) en el brazo.

Figura 13. Respuesta a los parámetros obtenidos en PID tuning.

Figura 14. PID paralelo.

Figura 15. Respuesta de PID Paralelo ante entrada escalón.

Figura 16. Respuesta de PID Paralelo ante cambios de consigna.

Figura 17. Respuesta de PID Paralelo ante perturbaciones introducidas en la planta.

PID INTERACTIVO (ESTÁNDAR)

El controlador PID estándar se especifica mediante el diagrama de bloques que se muestra en la Figura 18 y está descrito por la ecuación 31.

$$u(t) = k_p e(t) + k_p k_i \int e(t) dt + k_p k_d \frac{de(t)}{dt} \quad (31)$$

Al igual que el controlador PID paralelo, este controlador fue sometido a una excitación del tipo escalón. El tiempo de establecimiento que se puede observar en la respuesta, presentada en la Figura 19, es de 21,75 segundos muy superior a la de la configuración PID paralelo y acercándose asintóticamente a la consigna.

La Figura 20 muestra su respuesta ante cambios de la señal de consigna, observándose que si los cambios son poco distanciados en el tiempo la salida no alcanza la nueva consigna producto de su elevado tiempo de establecimiento.

La respuesta ante perturbaciones tampoco es satisfactoria, el tiempo de establecimiento después de la perturbación es muy superior a 20 segundos (ver Figura 21).

PI-D

El controlador PI-D está construido como se muestra en la Figura 22, y la ecuación que lo describe es la 32.

$$u(t) = k_p e(t) + k_i \int e(t) dt - k_d \frac{dy(t)}{dt} \quad (32)$$

El controlador PI-D ante una entrada escalón posee un tiempo de establecimiento de 7,15 segundos, como se puede observar en la Figura 23.

La respuesta del controlador PI-D ante los cambios de la señal de consigna (Figura 24) también es bastante buena (bajo tiempo de establecimiento y del tipo sobre amortiguada).

Figura 18. Diagrama de bloques del controlador interactivo implementado.

Figura 19. Respuesta de un controlador PID interactivo estándar.

Figura 20. Respuesta de un controlador estándar.

Figura 21. Respuesta ante perturbaciones de un controlador estándar.

Figura 22. Diagramas de bloque de un controlador PI-D.

Figura 23. Respuesta de un controlador PI-D.

Figura 24. Respuesta de un controlador PI-D ante cambios en la señal de consigna.

La próxima prueba es evaluar su respuesta cuando el sistema es sometido a perturbaciones observándose (Figura 25) que el tiempo de establecimiento es de aproximadamente 8 segundos no presentando prácticamente oscilaciones.

I-PD

El diagrama de bloques del controlador I-PD se ilustra en la Figura 26 y la ecuación 33 representa su ecuación característica.

$$u(t) = -k_p y(t) + k_i \int e(t) dt - k_d \frac{dy(t)}{dt} \quad (33)$$

Realizando las mismas pruebas que en los casos anteriores, se puede observar (Figura 27) una respuesta sobre amortiguada con un tiempo de establecimiento de 7,19 segundos.

La respuesta del controlador ante cambios en la señal de consigna, es algo más lenta pues no se alcanza a determinar en la escala de tiempo, su convergencia hacia las nuevas consignas (Figura 28) sin embargo se puede anticipar que no se presentarán sobre impulsos.

Finalmente la última prueba es someter al sistema a perturbaciones obteniéndose una respuesta satisfactoria con tiempos de establecimiento cercanos a 9 segundos una vez perturbado el sistema (Figura 29).

DISCUSIÓN DE RESULTADOS

Al momento de diseñar e implementar un sistema de control, es importante definir cuáles son los objetivos de control. Lo esperado normalmente, es que el sistema diseñado principalmente sea estable.

Figura 25. Respuesta de un controlador PI-D ante perturbaciones.

Figura 26. Diagrama de bloques de un controlador I-PD.

Figura 27. Respuesta de un controlador I-PD.

Figura 28. Respuesta ante cambios de setpoint de un controlador I-PD.

Figura 29. Respuesta ante perturbaciones de un controlador I-PD.

Objetivos adicionales y muy relevantes, son que la respuesta del sistema ante cambios de consigna en el intervalo de control definido, converja a la nueva consigna y no se presenten oscilaciones, que la respuesta no presente máximos sobre

impulsos que dañen los elementos de control o los actuadores que deben ser alimentados por la señal de control. Otro parámetro importante, es que los tiempos de establecimiento sean los mínimos tal que no se produzcan pérdidas significativas en el

sistema de producción (materias primas, productos defectuosos, etc.).

Aclarados estos conceptos previos, los resultados logrados en el trabajo realizado son los siguientes: como se puede observar en la Figura 30, el controlador que posee el mejor desempeño es el controlador PID paralelo seguido del controlador PI-D que posee un comportamiento similar, en tercer lugar, se ubica el controlador PI-D que es apenas 0,14 segundos más lento que los mencionados anteriormente.

En cuanto al controlador PID serial, este último no fue sintonizado con los mismos parámetros ya que si se utilizaban los mismos, el controlador no ofrecía una respuesta en el rango de valores admitidos por el controlador del motor brushless, por lo que fue necesario hacer una nueva sintonía para encontrar los parámetros adecuados. Su tiempo de establecimiento fue de 21,75 segundos, por lo que no cumple con los objetivos de control.

Tabla 1. Comparativa de tiempos de establecimiento.

Algoritmo	Tiempo de establecimiento [s]
PID Paralelo	7,1
PID Serial	21,75
PI-D	7,15
I-PD	7,29

CONCLUSIONES

El establecer la estructura digital de control para la planta, se hizo un trabajo tedioso al tener que

trabajar con las transformadas de Laplace y la transformada z, aun así, representar el algoritmo de control como una ecuación de diferencias, facilita el representar posteriormente las diversas variantes de controladores PID, pues es solo calcular para cada uno de los correspondientes nuevos parámetros en la ecuación de diferencias.

Existen muchos algoritmos de control, los cuales se pueden clasificar en algoritmos de control clásico y algoritmos de control avanzado, los clásicos son los más utilizados y los más fáciles de implementar, pero tienen ciertas desventajas, una de ellas es la dificultad que poseen para controlar procesos complejos como por ejemplo no lineales, de tipo MIMO o con un elevado tiempo muerto, es por ello por lo que existen controladores predictivos o los que se basan en el uso del predictor de Smith.

En cuanto a la comparativa es importante mencionar que todo depende de los parámetros que se ingresen a los controladores, en este caso el único algoritmo que se sintoniza de manera distinta fue el algoritmo serial, ya que éste no respondía al ser sintonizado con los parámetros de los otros controladores. La respuesta de todos los controladores (excepto el serial) fue bastante aceptable desde el punto de vista de los objetivos de control (respuesta estable, mínimos sobre impulsos, errores de estado estacionario nulos y bajos tiempos de establecimiento).

Lo otro relevante fue observar que los resultados de las implementaciones en tiempo real, fueron bastante similares a las obtenidas en tiempo de simulación, en cuanto al tipo de respuestas y de calidad de las diferentes implementaciones.

Figura 30. Comparativa de algoritmos PID.

REFERENCIAS

- [1] L.F. Lozano-Valencia, L.F. Rodríguez-García y D. Giraldo-Buitrago. "Diseño, Implementación y validación de un Controlador PID Autosintonizado". *Tecnológicas* N° 28, pp. 33-53. 2012. ISSN 0123-7799.
- [2] K. Ogata. "Ingeniería de Control Moderno". Pearson Education S.A. Madrid. 2010.
- [3] K. Ogata. "Sistemas de Control de Tiempo Discreto". Prentice Hall Hispanoamericano S.A. México. 1996.
- [4] E. Tacconi, R. Mantz, J. Solsona y P. Puleston. "Controladores Basados en Estrategias PID". 2005.
- [5] F. Morilla García. "Controladores PID". ETSI, UNED. Madrid. 2007.
- [6] D. Rodríguez Ramírez y C. Bordóns Alba. "Apuntes de Ingeniería de Control". 2007.
- [7] J. Cosco Grimaney. "Estrategias de Control". Universidad Nacional de Ingeniería. 2011.
- [8] V.B. Sreenivasappa and R.Y. Udaykumar. "Analysis and implementation of Discrete Time PID Controller using FPGA". *International Journal of Electrical and Computer Engineering*. Vol. 2, Issue 1, pp. 71-82. 2010. ISSN 0974-2190.
- [9] S.E. Toledo Chojolán. "Diseño de Controladores PID en Tiempo Discreto y Análisis de Respuesta Utilizando Herramientas Computacionales". Universidad de San Carlos de Guatemala, Escuela de Ingeniería Mecánica. Guatemala. 2007.
- [10] F.S. González Muñoz y A.F. Blabin Vallejo. "Diseño de un Controlador Digital Universal PID con Características de Tipo Industrial". Universidad Tecnológica de Pereira. 2015.
- [11] F. Morilla García. "Control PID, Ajuste Empírico". Dpto. de Informática y Automática, ETSI de informática, UNED. Madrid. 2016. URL: http://www.dia.uned.es/~fmorilla/MaterialDidactico/ajuste_empirico.pdf
- [12] C. Álvarez, A. Soto y F. Watkins. "Simulación de controladores digitales". *Ingeniare. Revista chilena de ingeniería*, Vol. 17 N° 3 pp. 309-316. Septiembre-Diciembre 2009.
- [13] M. Sánchez y O Llanes. "Una propuesta de reglas de sintonía robustas para controladores PI-2D". *Ingeniare. Revista chilena de ingeniería*. Vol. 25 N° 1, pp. 28-38. Mayo 2016.