

Diseño e implementación de filtros y controladores PID para estabilización de sistemas dinámicos

Design and implementation of filters and PID controllers for dynamic system stabilization

Sebastián Villada García¹

Universidad Nacional Abierta y a Distancia, Colombia

Resumen

En este trabajo, se aborda el diseño y la implementación de filtros digitales y controladores PID discretos para estabilizar una función de transferencia dada. Utilizando la herramienta c2d de Matlab, se discretiza la función de transferencia original, y se observa que la planta no alcanza el valor deseado de 1. Se implementa un filtro digital multiplicando por un factor de escala. Posteriormente, se emplea el método del lugar geométrico de las raíces para analizar el comportamiento de la planta realimentada, revelando una respuesta lenta al escalón. Luego, se aplica el método de Ziegler-Nichols para sintonizar un controlador PID. Los parámetros resultantes generan una respuesta con un ligero sobre impulso. Se utiliza la herramienta PID Tuner para ajustar los parámetros PID, logrando una respuesta más suavizada y rápida estabilización. El documento aborda cada fase del diseño, desde la discretización inicial hasta la sintonización y validación del controlador PID.

Palabras clave: filtros digitales, controladores PID, diseño de sistemas, Matlab, lugar geométrico de las raíces, sintonización PID.

Abstract

In this work, the design and implementation of digital filters and discrete PID controllers to stabilize a given transfer function is addressed. Using Matlab's c2d tool, the original transfer function is discretized, and it is observed that the plant does not reach the desired value of 1. A digital filter is implemented by multiplying by a scale factor. Subsequently, the root locus method is used to analyze the behavior of the refeed plant, revealing a slow response to stepping. Then, the Ziegler-Nichols method is applied to tune a PID controller. The resulting parameters generate a response with a slight overshoot. The PID Tuner tool is used to adjust the PID parameters, achieving a smoother response and faster stabilization.

¹ svilladaga@unadvirtual.edu.co

The paper addresses each phase of the design, from initial discretization to PID controller tuning and validation.

Keywords: Digital filters, PID controllers, system design, Matlab, root locus, PID tuning.

1. Introducción

El diseño de sistemas de control para funciones de transferencia discreta es esencial para garantizar un rendimiento óptimo. Este trabajo aborda el proceso de desratización de una función de transferencia, la implementación de un filtro digital y la sintonización de un controlador PID para mejorar la respuesta del sistema.

2. Metodología

La metodología implica la discretización de la función de transferencia, la implementación de un filtro digital y la sintonización de un controlador PID mediante el método de Ziegler-Nichols y la herramienta PID Tuner de Matlab.

2.1 Método 123c

Hallar el 75 y el 25 % con regla de 3:

$$15 \rightarrow 100\%$$

$$x \rightarrow 25\%$$

$$x_1 = \frac{15 * 25\%}{100\%}$$

$$x_1 = 3.75$$

Realizamos el mismo procedimiento para el 75 %:

$$15 \rightarrow 100\%$$

$$x \rightarrow 75\%$$

$$x_1 = \frac{15 * 75\%}{100\%}$$

$$x_1 = 11.25$$

Realizamos el cálculo para k_p

$$k_p = \frac{\Delta y}{\Delta u} = \frac{15}{0.15} = 100.$$

$$k_p = 100$$

$$\tau = 0.9102(\tau_{75} - \tau_{25})$$

$$\tau = 0.9102(0.72 - 0.25)$$

$$\tau = 0.427$$

$$t_m = 1.2620t_{25} - 0.2620t_{75}$$

$$t_m = 1.2620(0.25) - 0.2620(0.72)$$

$$t_m = 0.3155 - 0.1886$$

$$t_m = 0.1269$$

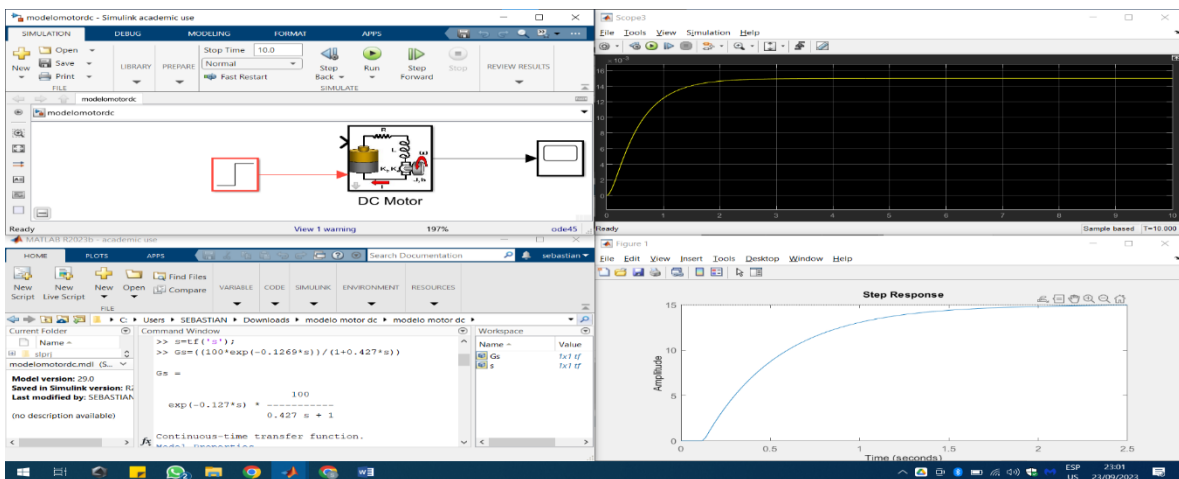
Teniendo estos datos podemos reemplazar valores en la ecuación del modelo de primer orden:

$$G_{p1}(s) = \frac{k_p * e^{-t_m s}}{\tau s + 1}$$

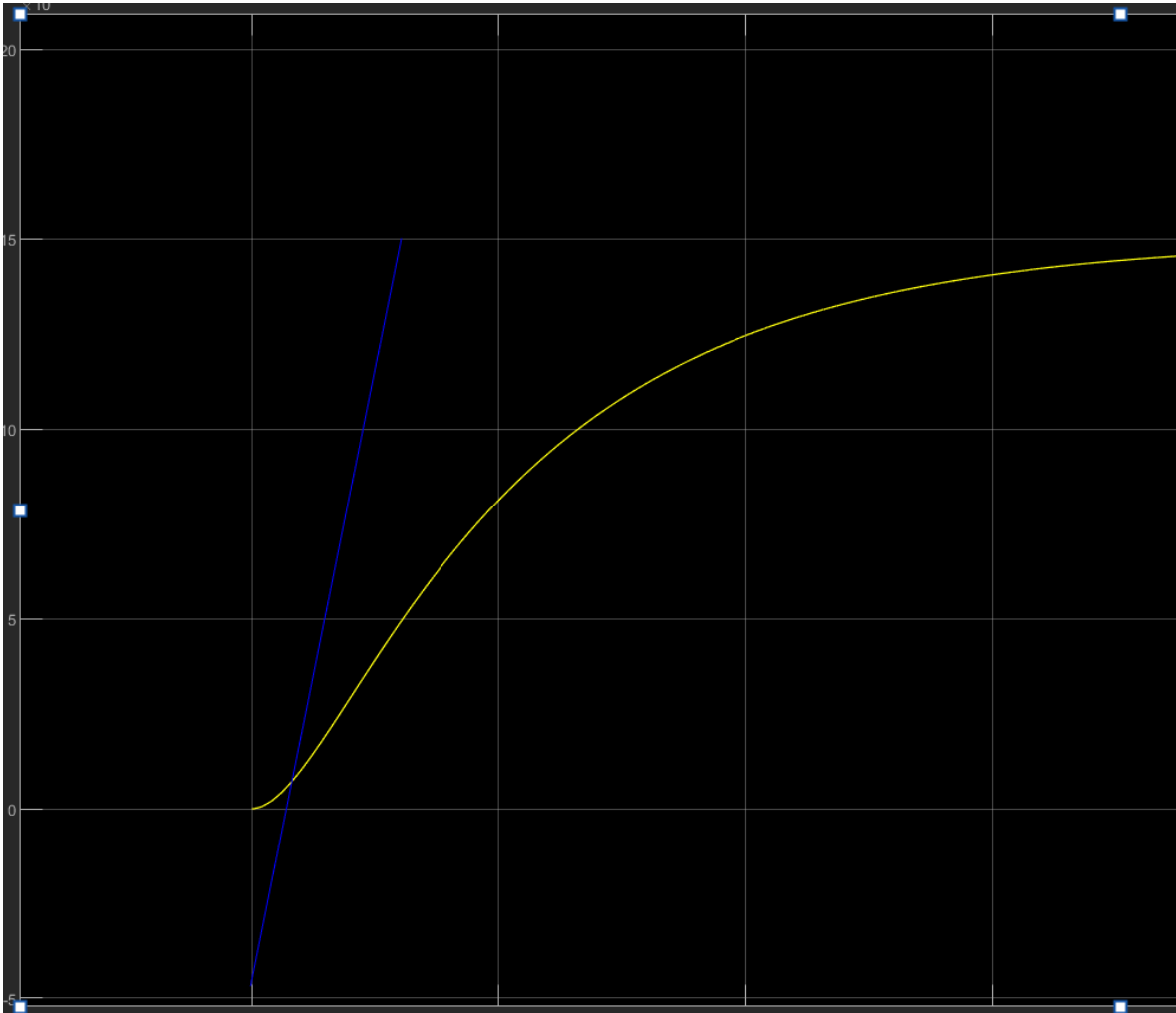
Reemplazamos términos:

$$G_{p1}(s) = \frac{100 * e^{-0.1269s}}{0.427s + 1}$$

Teniendo esta ecuación podemos realizar la simulación en Matlab:



Identificación de la recta tangente de Zingler y Nichols:



En el software de Matlab, realizamos una recta tangente sobre la gráfica del motor, esto de forma visual donde se genera el punto de inflexión de la misma. Una vez terminado esto, continuamos con los siguientes cálculos:

Calculamos $k_p = \frac{\Delta y}{\Delta u}$

$$\Delta y = (15E - 3) - 0 = 15$$

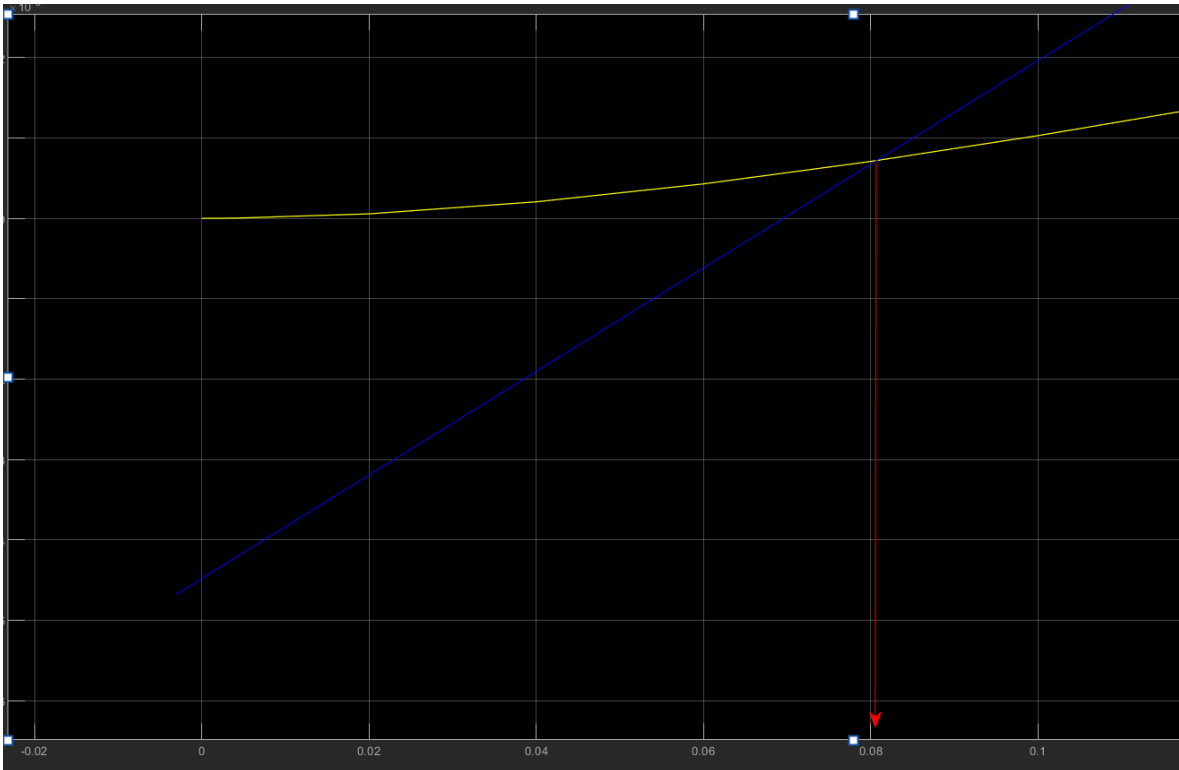
$$\Delta u = 0.15 - 0 = 0.15$$

$$k_p = \frac{15}{0.15}$$

$$k_p = 100$$

Realizamos el cálculo de t_m , el cual obtenemos los datos aproximados según la recta tangente trazada sobre la gráfica de respuesta del motor DC:

$$t_m = 0.08 \approx$$



Realizamos el cálculo aproximado de forma visual del tiempo en el que la respuesta del motor se estabiliza justo después del punto que interseca la tangente:

$$\tau = 0.3 - 0.08 = 0.22$$

teniendo estos datos, reemplazamos sobre la ecuación:

$$G_{p1}(s) = \frac{kp * e^{-tms}}{\tau s + 1}$$

$$G_{p1}(s) = \frac{100 * e^{-0.08s}}{0.22s + 1}$$

Graficamos en el software Matlab:

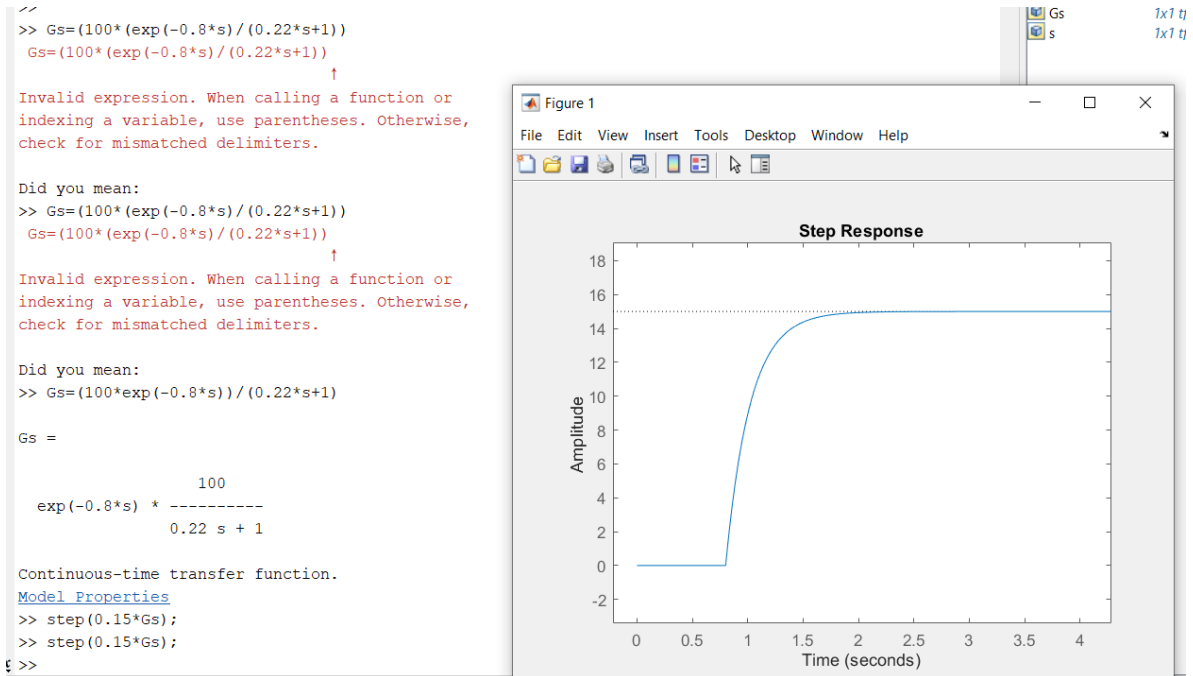


Gráfico 1. Método identificación de la recta tangente de Zingler y Nichols.

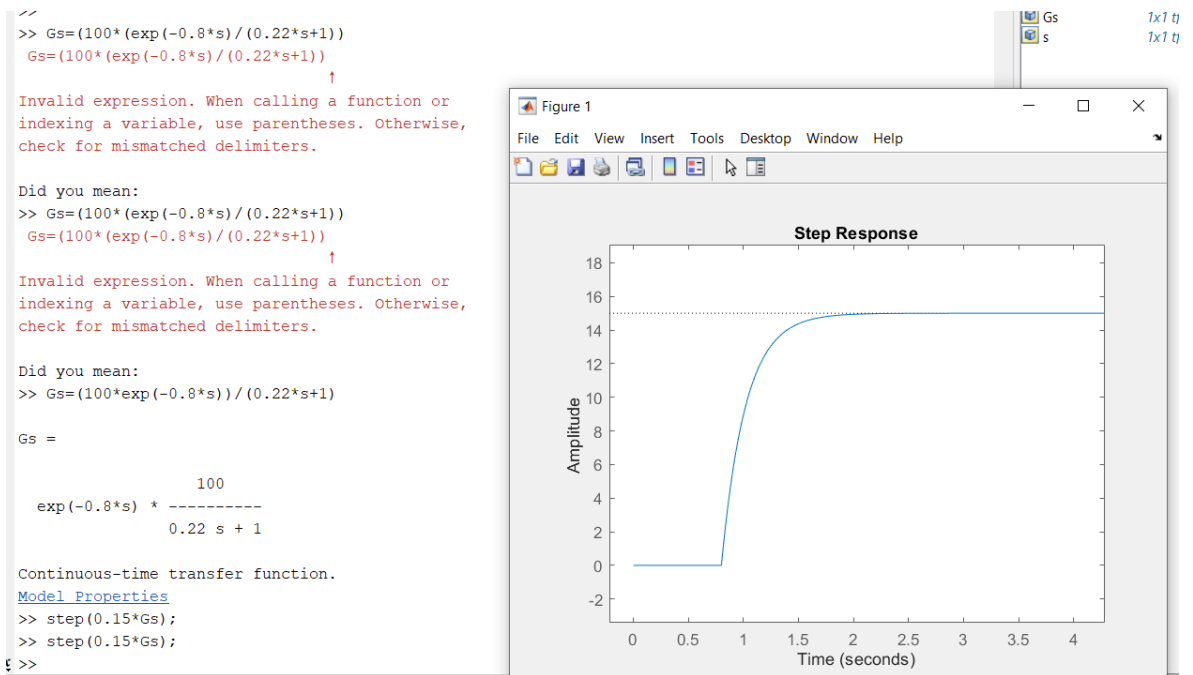
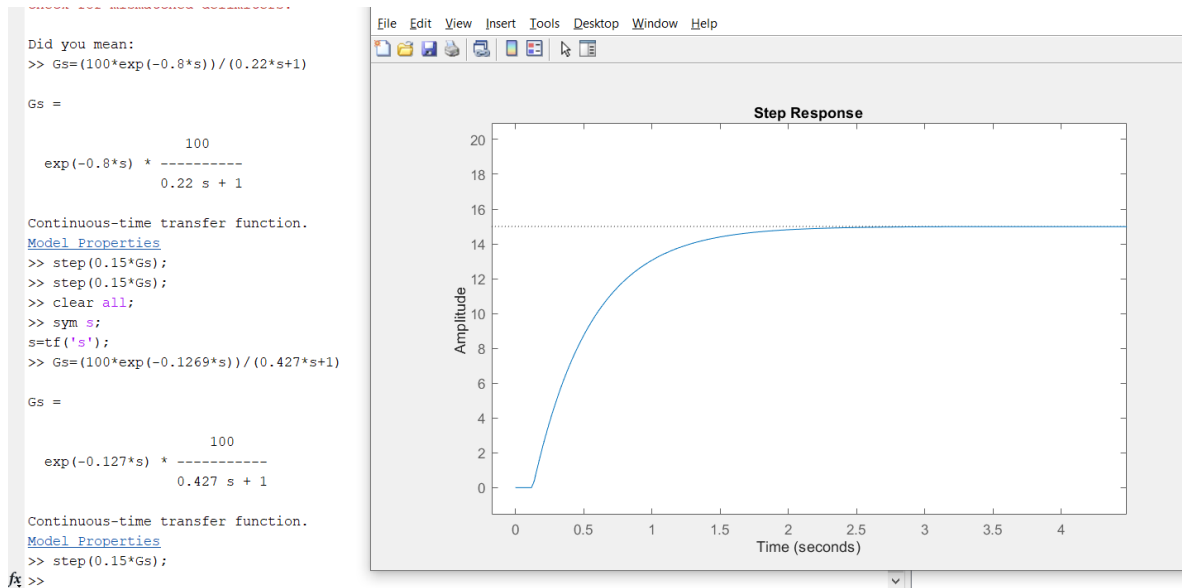


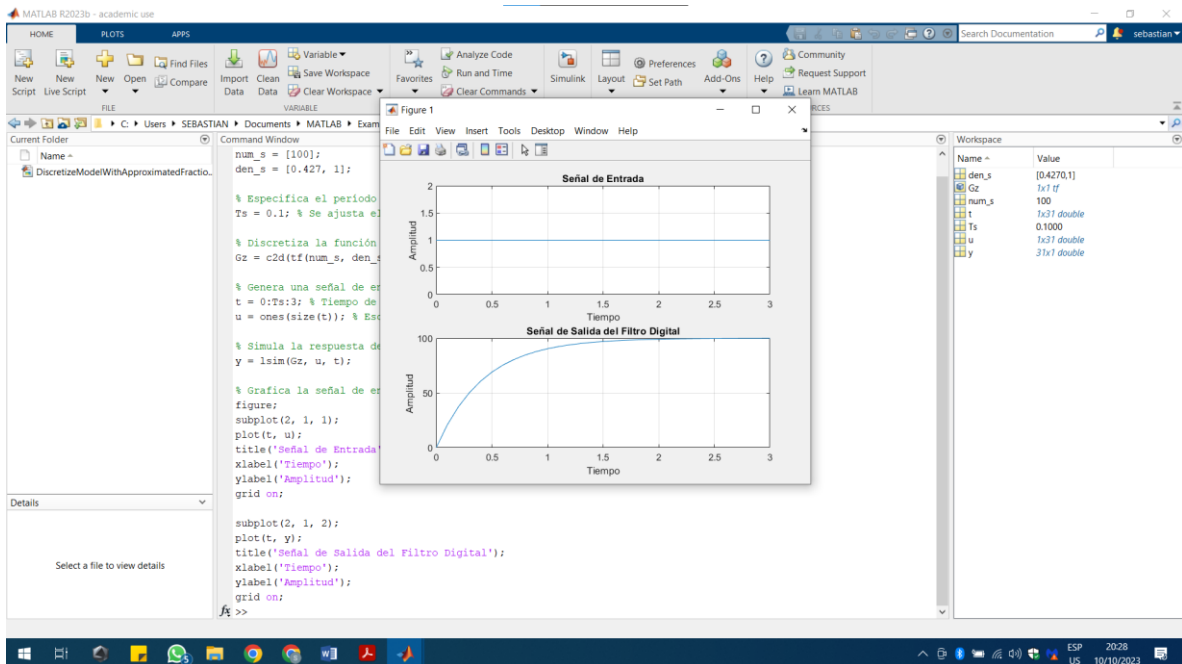
Gráfico 2. Método identificación 123c.



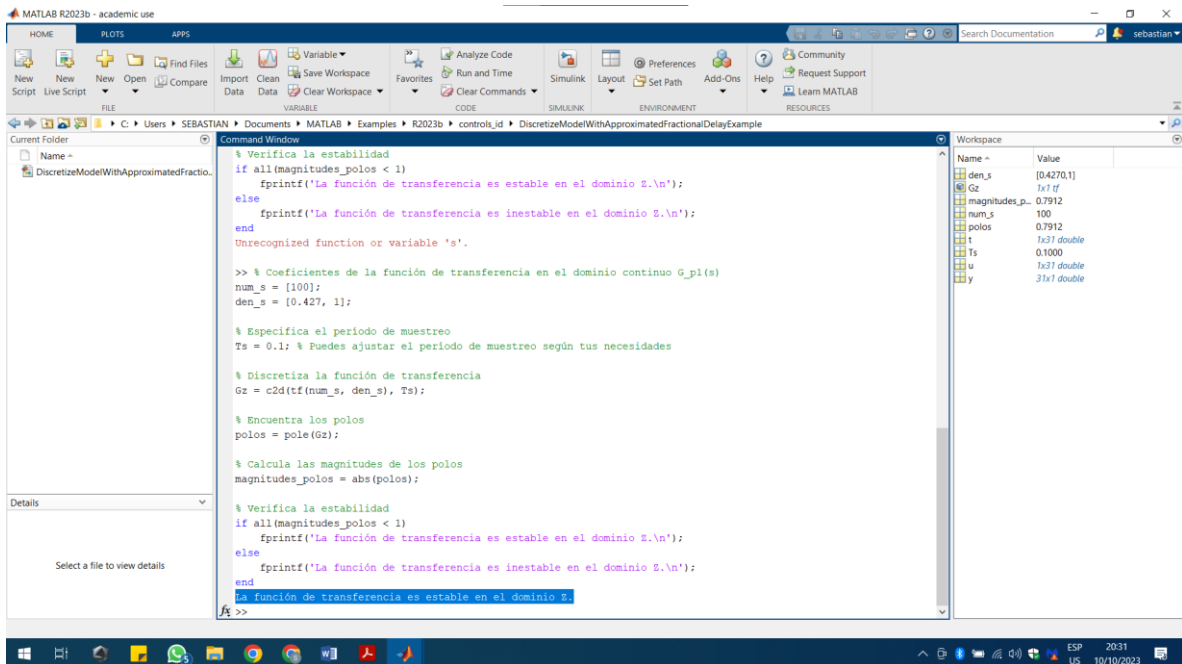
- Realizar el análisis de respuesta transitoria de manera analítica y computacional. Comparar resultados.

Teniendo en cuenta los resultados de los cálculos matemáticos y las simulaciones en el programa Matlab, podemos ver que el modelo más adecuado es aquel obtenido mediante el método de identificación "123c". Este modelo muestra una respuesta transitoria más suave, lo que significa que, al implementarlo en motores reales, logramos un rendimiento más eficiente del motor.

Período de muestreo (T_s): el período de muestreo T_s se establece en 0.1 segundos en este ejemplo, pero puede ajustarse según las necesidades del sistema. La elección de T_s afecta la precisión de la simulación y el rendimiento del filtro digital. Un T_s más pequeño



Dentro de las funciones de Matlab, podemos aplicar un código, el cual nos determina si el filtro es estable o no:



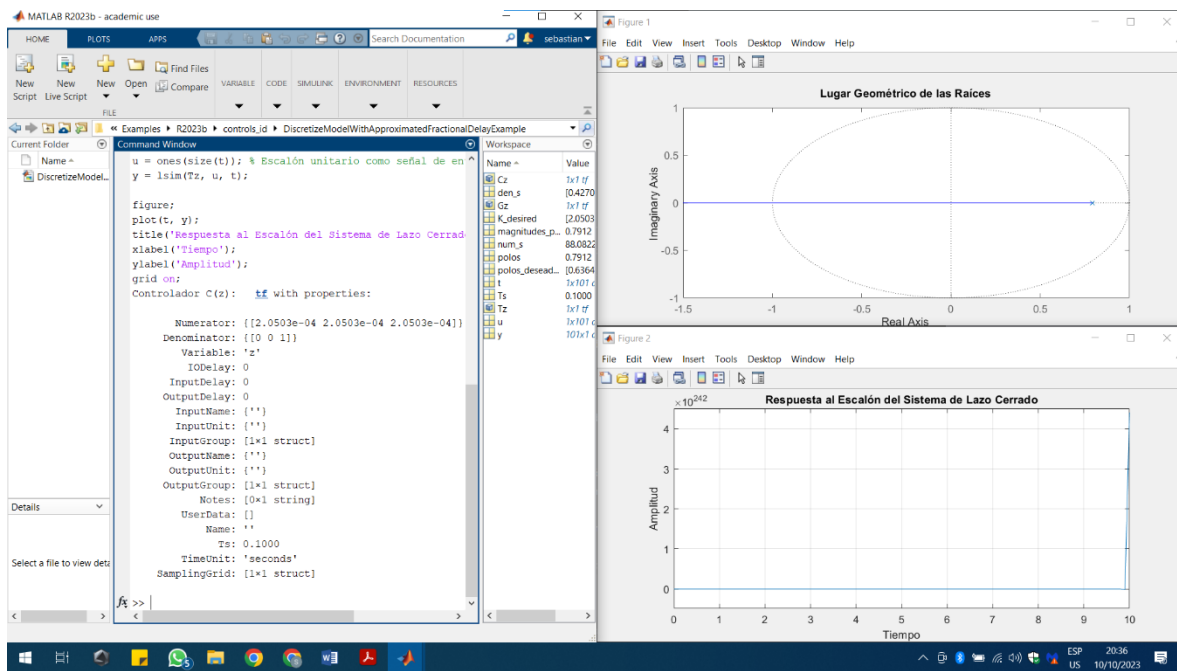
1. Diseño basado en el método de lugar geométrico de las raíces.

Función de transferencia original: inicialmente, se define la función de transferencia continua $G_{p1}(s)$ con los coeficientes num_s y den_s . Esto representa el sistema que deseas controlar.

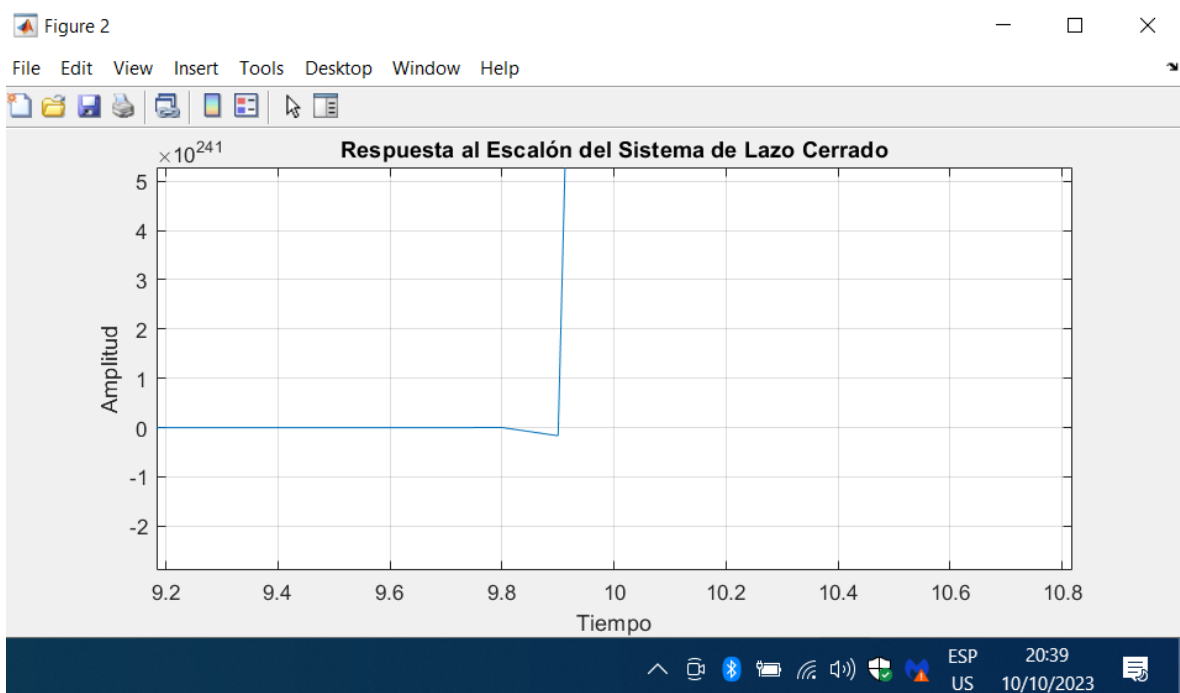
Aplicamos el siguiente código en Matlab, teniendo en cuenta la función de transferencia:

```
% Coeficientes de la función de transferencia en el dominio continuo  
G_p1(s)
```

Discretización: se discretiza la función de transferencia continua $G_p1(s)$ con un período de muestreo T_s para obtener la función de transferencia discreta G_z . Este paso es fundamental para trabajar con sistemas en tiempo discreto.



Lugar geométrico de las raíces (LGR): se traza el LGR de la función de transferencia discreta G_z . Esto permite visualizar cómo se mueven los polos a medida que se varía una ganancia en el controlador.



2. Diseño e implementación de controladores PID discretos.

En primer lugar, se definen los coeficientes del controlador PID discreto, que consta de una ganancia proporcional (K_p), una ganancia integral (K_i) y una ganancia derivativa (K_d). Estos coeficientes son críticos para el rendimiento del controlador. Se define una función de transferencia continua ($G_{p1}(s)$) en términos de su numerador y denominador. Esta función representa el sistema en el dominio continuo antes de la discretización.

Se simula la respuesta del sistema en un período de tiempo definido (t) con una señal de referencia (r). La respuesta del sistema en lazo cerrado se almacena en " y ". Se genera una gráfica que muestra tanto la señal de referencia como la respuesta del sistema en función del tiempo. Esto permite visualizar cómo el sistema se comporta bajo la influencia del controlador PID discreto.

MATLAB R2023b - academic use

HOME PLOTS APPS

File Edit View Insert Tools Desktop Window Help

Current Folder: C:\Users\SEBASTIAN\Documents\MATLAB\Examples\R2023b\controls_id

Command Window

```

>> clear all
>> % Definición de los coeficientes del controlador
Kp = 1.0; % Ganancia proporcional
Ki = 0.2; % Ganancia integral
Kd = 0.1; % Ganancia derivativa
>> % Coeficientes de la función de transferencia
num_s = [100 * exp(-0.1269)];
den_s = [0.427, 1];

% Especifica el periodo de muestreo
Ts = 0.1; % Se ajusta el periodo de muestreo

% Discretiza la función de transferencia
Gz = c2d(tf(num_s, den_s), Ts);

% Creación del controlador PID discreto
Cz = pid(Kp, Ki, Kd);

% Ajusta el periodo de muestreo del controlador
Cz.Ts = Ts;

% Creación del sistema en lazo cerrado
SistemaLazoCerrado = feedback(Cz * Gz, 1);

% Tiempo de simulación
t = 0:Ts:10;

% Señal de referencia (por ejemplo, un escalón)
r = ones(size(t));

% Simulación de la respuesta del sistema en lazo cerrado
[y, ~] = lsim(SistemaLazoCerrado, r, t);

```

Figure 2: Respuesta del Sistema en Lazo Cerrado con Controlador PID

Workspace

Name	Value
Cz	1x1 pid
den_s	[0.4270, 1]
Gz	1x1 tf
Kd	0.1000
Ki	0.2000
Kp	1
num_s	88.0822
r	1x101 double
SistemaLazoC...	1x1 tf
t	1x101 double
Ts	0.1000
y	101x1 double

ESP 2021
LAA 12/10/2023

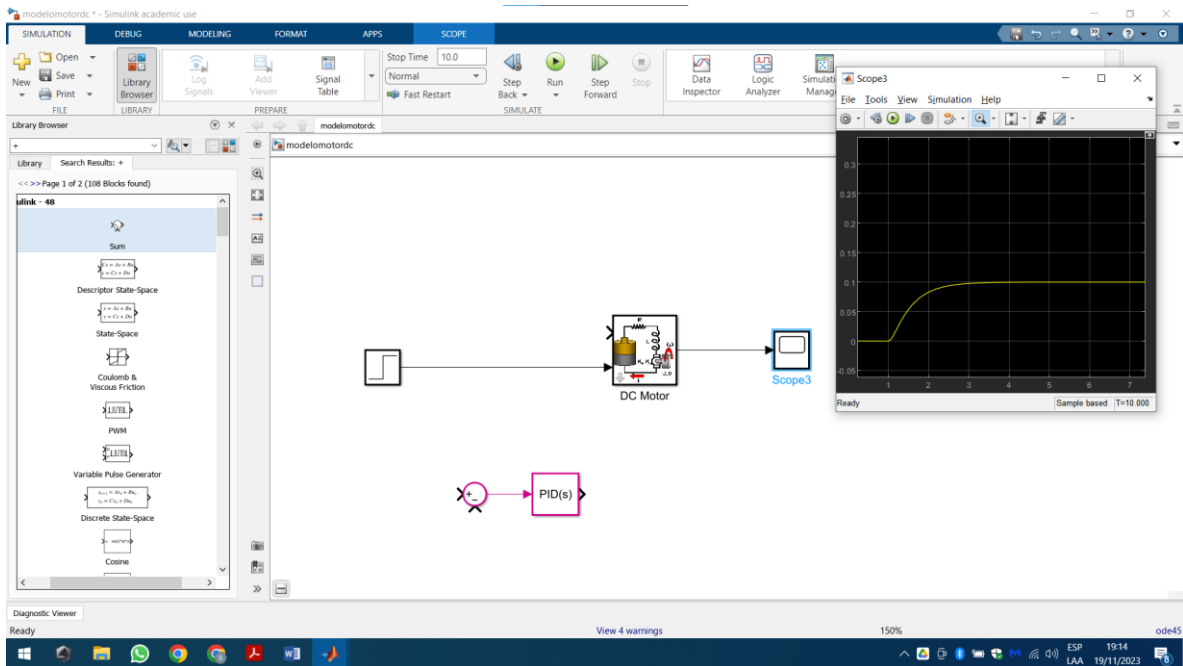


Ilustración 1. controlador sin PID.

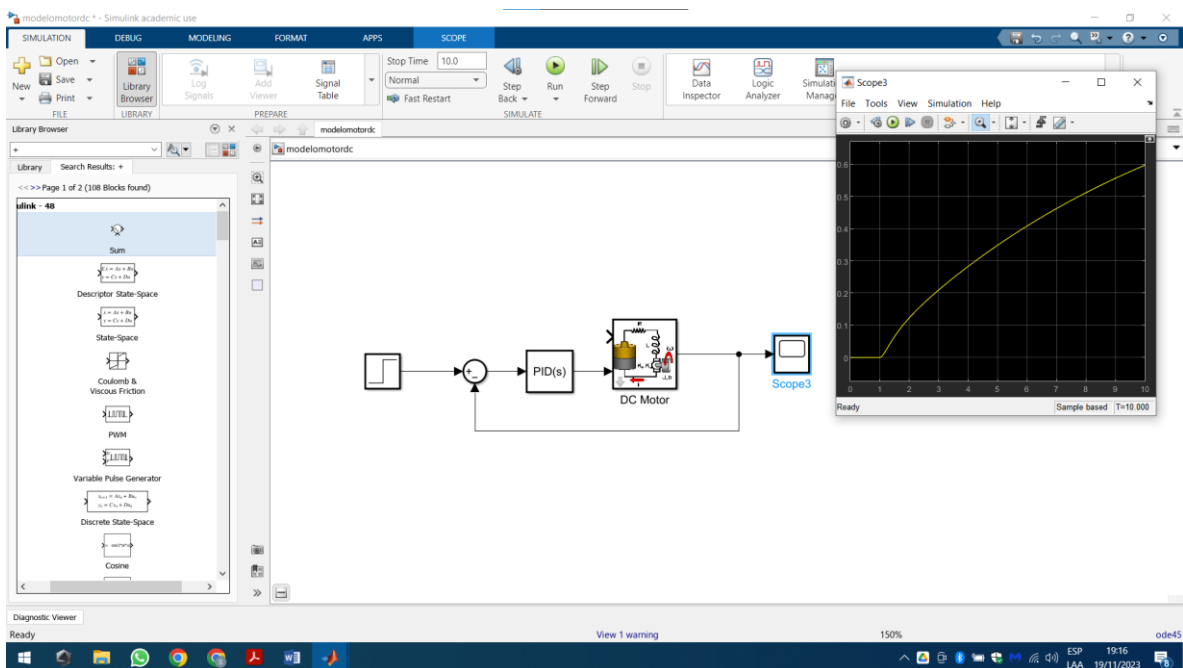


Ilustración 2. controlador con PID sin sintonizar.

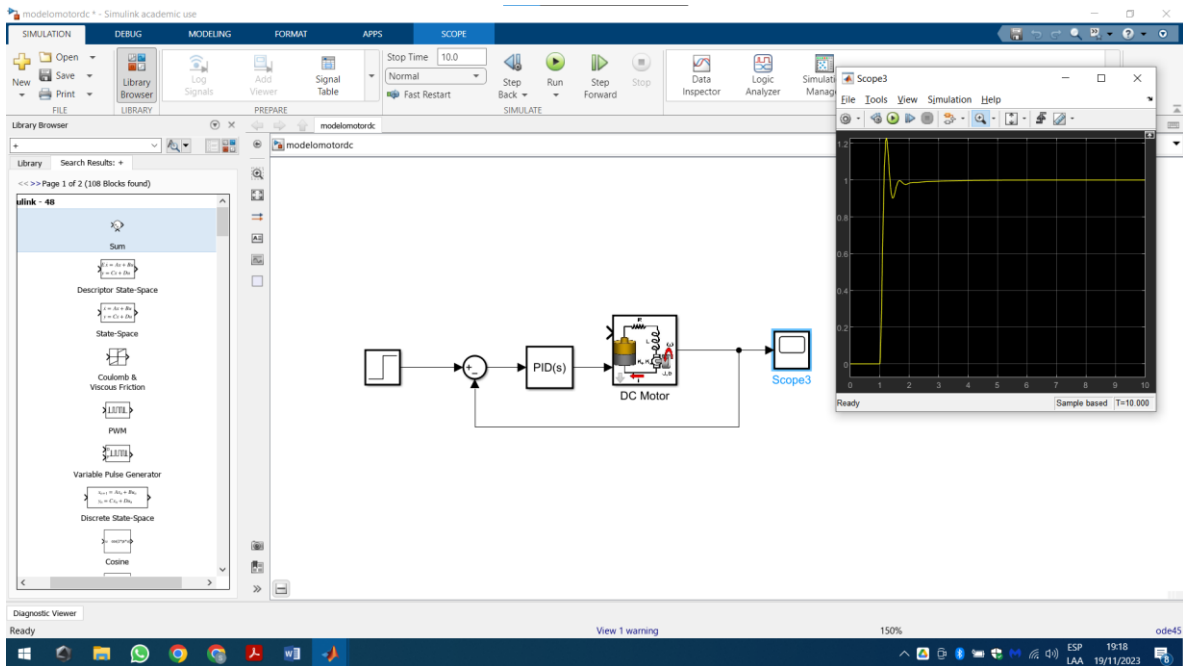


Ilustración 3. controlador PID sintonizado.

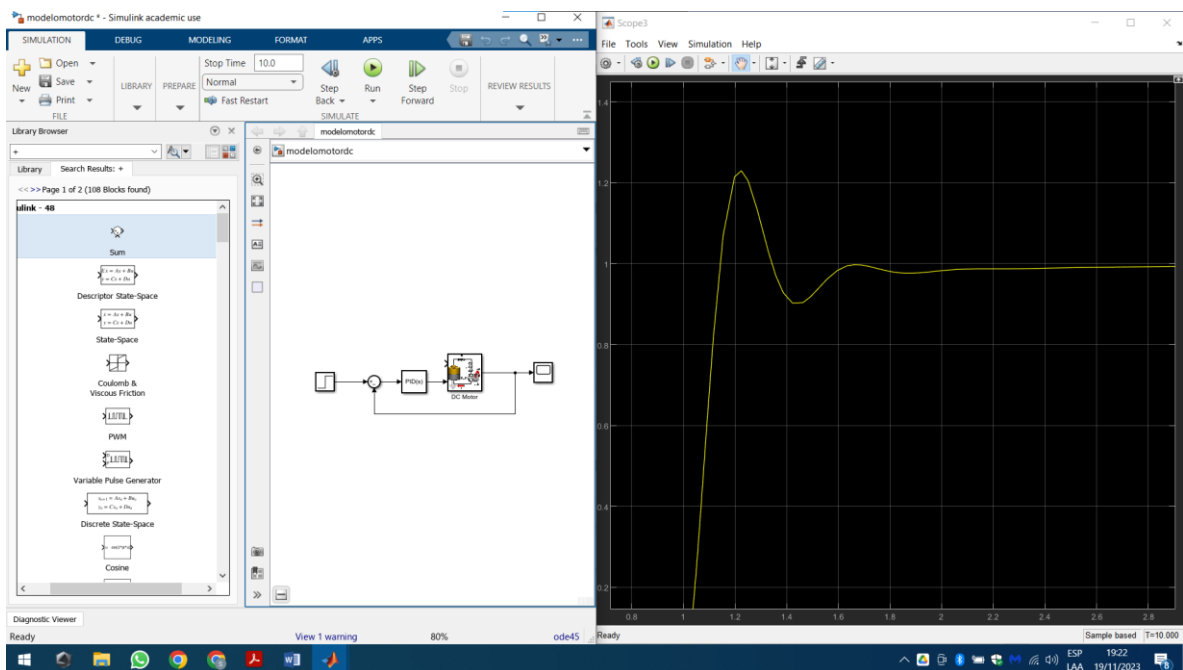


Ilustración 4. ampliación del gráfico del controlador PID.

Para este controlador vemos que tiene unos picos los cuales llevados a la realidad pueden ser un problema, es por esto que se va a realizar una modificación en los parámetros del PID, esto con el fin de que la señal de control sea más estable, permitiendo de esta manera que el motor tenga una velocidad más controlada en la vida real.

En la figura se aprecia que la planta es susceptible de ser regulada mediante un controlador proporcional-integral-derivativo (PID). Además, se dispone de la capacidad de ajustar los parámetros utilizando un sintonizador automático. Es posible modificar los valores asociados al pico de respuesta, aunque es importante señalar que también es factible incrementar los tiempos relacionados con la estabilidad estacionaria y otros aspectos relevantes.

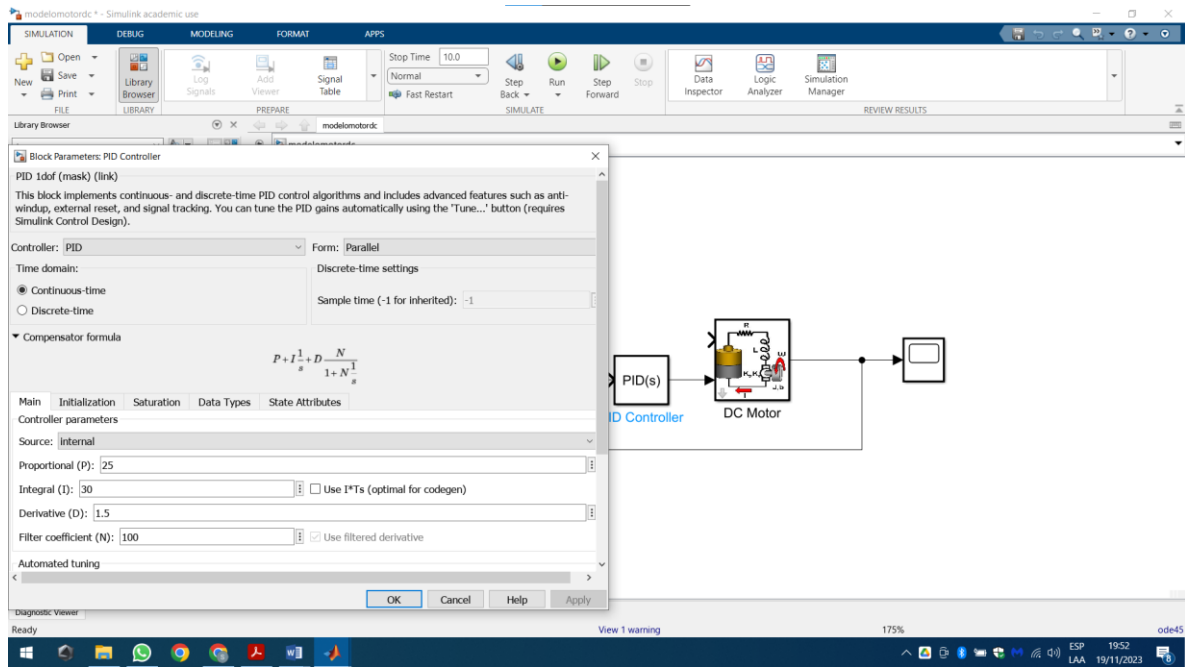


Ilustración 5. parámetros nuevos controlador PID.

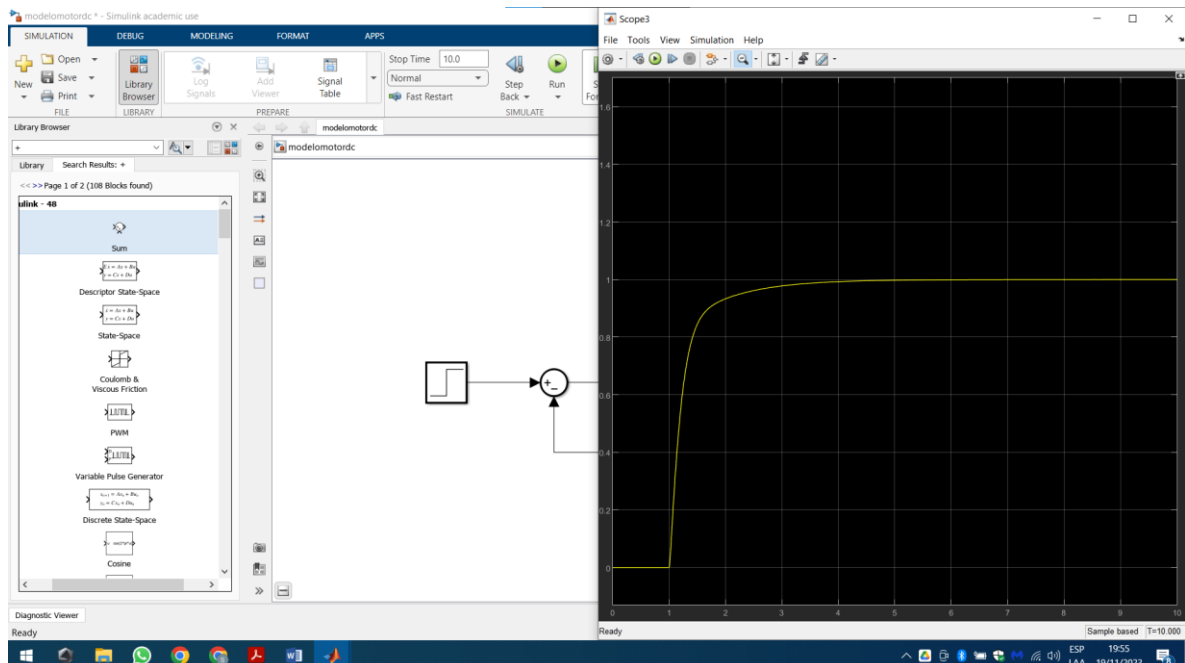


Ilustración 6. gráfico de señal controlador PID parámetros nuevos.

Como podemos ver en la Ilustración 6, el controlador PID tiene un comportamiento más estable al momento de llegar a su punto máximo, esto hace que el comportamiento del motor tenga más rendimiento y no afecte los demás sistemas por ruido o perturbaciones en sensores, frecuencias resonantes no deseadas, retrasos en los sistemas, entre otros.

En la Ilustración 6, se evidencia que el controlador proporcional-integral-derivativo (PID) exhibe una respuesta más robusta al alcanzar su valor máximo. Este fenómeno contribuye a un rendimiento óptimo del motor, minimizando la influencia de factores externos como ruido, perturbaciones en los sensores, frecuencias resonantes indeseadas y retardos en los sistemas. En términos técnicos, el PID muestra una mayor estabilidad en condiciones desafiantes, lo que se traduce en un comportamiento más eficiente y menos propenso a interferencias en la operación del sistema.

3. Discusión

Se discuten los resultados obtenidos en cada etapa del diseño, destacando los desafíos encontrados y las soluciones aplicadas. Se analiza la eficacia del filtro digital y se compara con la respuesta realimentada utilizando el método del lugar geométrico de las raíces.

4. Conclusiones

El trabajo demuestra que el diseño e implementación de filtros digitales y controladores PID son fundamentales para mejorar la estabilidad y el rendimiento de sistemas dinámicos discretos. La combinación de métodos analíticos y herramientas de software, como Matlab, facilita un enfoque integral para abordar estos desafíos.

Este documento proporciona una guía detallada que puede ser de utilidad para ingenieros y científicos que trabajan en el diseño y control de sistemas dinámicos discretos.

Referencias

Bertran Albertí, E. (2006). *Procesado digital de señales: fundamentos para comunicaciones y control*. Ediciones UPC.

Bustos, J. (2018). *Controlador PID en un microcontrolador*.
<http://hdl.handle.net/10596/22985>

Quintero Monroy, C., Oñate López, J. & Jiménez Mares, J. (2014).
Control automático aplicado. Universidad del Norte.

Sauchelli, V. H. (2020). *Introducción a los sistemas de control digital*.
Ariel