Identificación y Control PI

Control Digital Avanzado

Camilo Andres Vera Ruiz

caverar@unal.edu.co

1. Características de la Planta

   El sistema consiste en un motor DC de 12v, que cuenta con una caja reductora de 20:1, es decir que el eje de salida da una vuelta por cada 20 vueltas del motor. Además el sistema cuenta con un encoder de cuadratura el cual alcanza una resolución de 360 cuentas por revolución, que multiplicado por los cuatro flancos de subida y bajada de las dos señales de cuadratura, se tiene como resultado 1440 cuentas por revolución para el eje del motor, para un total de 28800 cuentas por revolución para el eje de salida de la caja reductora.

   El sistema se controla mediante un microcontrolador STM32 F411 ajustado a una frecuencia de 80Mhz y programado mediante el software PlatformIO. Adicionalmente se configuro el periférico UART a 115200 baudios con el fin de observar las señales de control, velocidad, y referencia, en tiempo real, así como para ajustar los parámetros del control, tales como el tipo de señal de referencia, su periodo, su amplitud, el tipo de controlador, etc.

   La velocidad del motor es medida a partir de las cuentas del encoder, en revoluciones por minuto, mientras que la posición se mide en grados, contando con un filtro de velocidad el cual mitiga el overflow del contador del encoder, eliminando mediciones que superen valores de velocidad que no son posibles, en este caso 50 rpm.

   Finalmente se tiene configuro un periodo de muestreo de 5ms mediante interrupciones.

2. Identificación de Sistema

   Tal y como se especificó en la guía de laboratorio, se identificó el sistema como uno de primer orden para la velocidad del motor, utilizando la herramienta de identificación de sistemas de MATLAB. Para ello se configuro en el microcontrolador una señal cuadrada con un periodo de 500ms y un PWM de 0.6 a 0.8 es decir una variación de voltaje de 7.2v a 9.6v, con el fin de evitar la región no lineal del sistema, en la que la fricción de coulomb, evita que el motor se mueva hasta alcanzar un voltaje determinado, es decir que se seleccionó la señal, buscando que el motor nunca se detuviese.

   A partir de la configuración de la señal PWM, se capturaron aproximadamente 2 segundos de la señal con un tiempo de muestreo de 5ms, por medio de la salida UART y un conversor USB serial.

En la Figura 1, se puede observar la gráfica de salida de la herramienta de identificación de sistemas de MATLAB, donde se comparan los datos capturados, con una simulación del sistema de primer orden que MATLAB aproximo, evidenciando una precisión de 80.72%. Es importante recalcar que el modelo no iguala los valores de estado estacionario del sistema real, en contraparte a la dinámica que si que está muy bien modelada (aunque con una clara asimetría en el comportamiento descendiente vs el ascendente), pero dado que se pretende desarrollar control realimentado con cero error de estado estacionario, el error de estado estacionario que presenta el modelo de aproximadamente 12.5%, no debería suponer un problema.

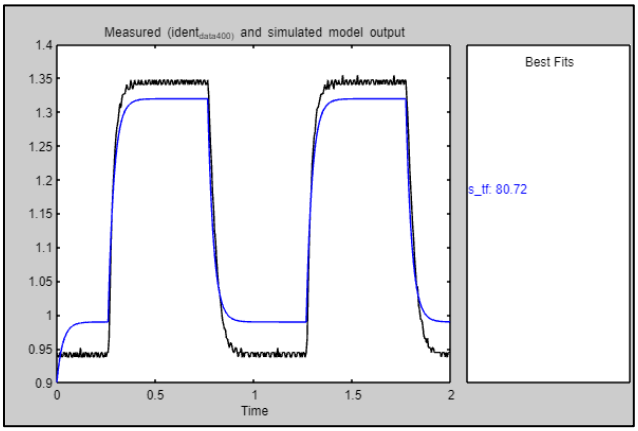

Figura 1: Identificación del Sistema.

El modelo del sistema se muestra en la siguiente ecuación.

$$G_p(s) = \frac{55.99}{s + 33.95}$$

En la figura 2 se muestra la respuesta al escalón unitario, donde se evidencia un tiempo de subida de 64.7ms.

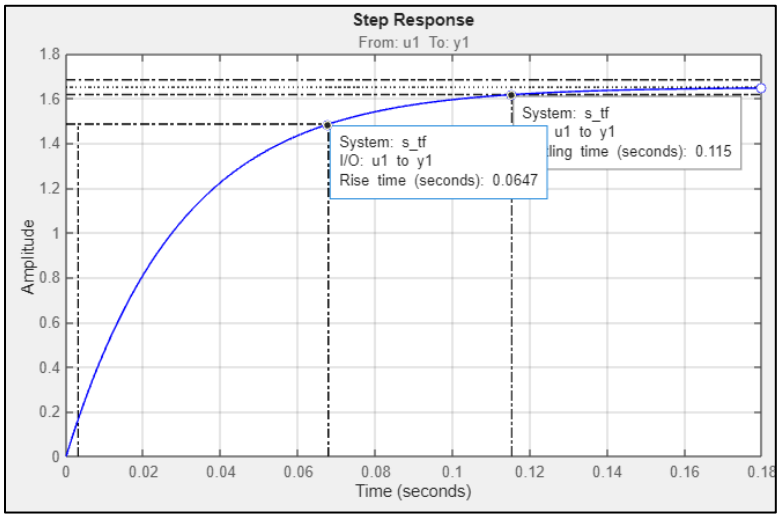

Figura 2: Respuesta al paso del modelo de primer orden de la planta.

3. Diseño de Controlador

Se requiere un controlador PI con cerro error de estado estacionario y un tiempo de subida menor al 70% del tiempo de subida de la planta sin realimentar, es decir 45.3ms, y teniendo en cuenta que el componente integral del controlador debería eliminar el error de estado estacionario, solo es necesario preocuparse por el tiempo de subida y la saturación del actuador.

El diseño se realizó en tiempo continuo por medio de la herramienta PID Tuner de MATLAB, la cual calcula las constantes del controlador a partir de dos parámetros que controlan el tiempo de respuesta y la robustes. En la figura 3 se muestra cómo se ajustó el parámetro de tiempo de respuesta con un valor muy cercano a los 45.3ms, en este caso se seleccionó el valor de 43.6ms, luego se modificó el segundo parámetro buscando reducir el esfuerzo de control al menor posible, con el fin reducir el riesgo de saturar el actuador, en este caso buscando una amplitud en la señal de control, menor a 1, teniendo en cuenta que la señal de PWM va desde -1 a 1. Finalmente se obtiene un controlador con un tiempo de subida de 41.6ms el cual es menor al límite máximo requerido, así como un repuesta paso prácticamente de primer orden, con un sobre pico muy pequeño de 1.09% y un tiempo de estabilización de 62.3 ms. En la figura 3 se muestra una captura de la herramienta PID Tuner con gráficas de la respuesta al paso y el esfuerzo de control, así como los parámetros de desempeño y las constantes obtenidas para el componente proporcional y el componente integral del controlador, utilizando una topología de control PI paralelo.

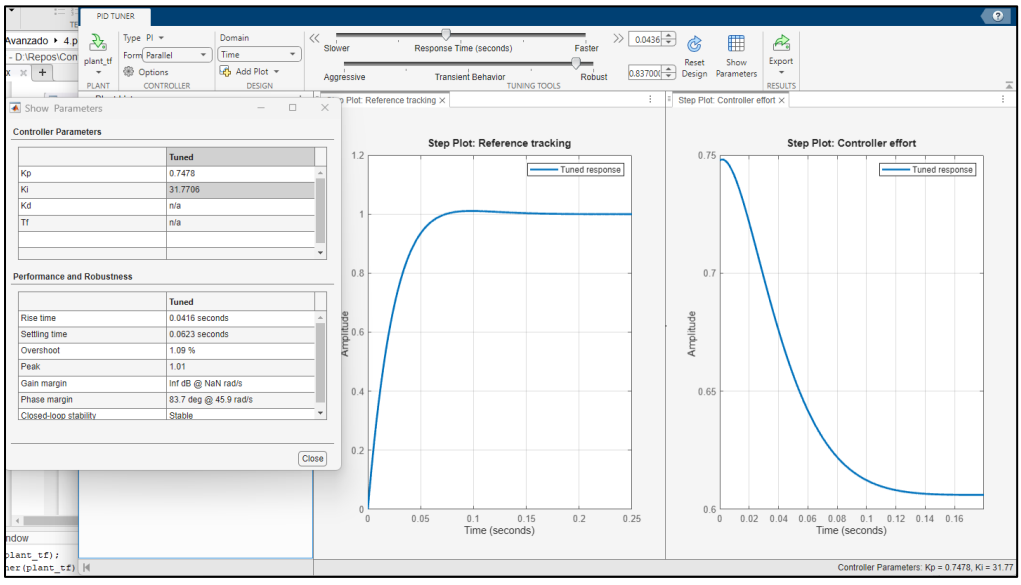

Figura 3: Diseño de PI

El controlador puede ser expresado mediante la siguiente expresión de función de transferencia:

$$G_c(s) = Kp + \frac{Ki}{s} = 0.74773 + \frac{31.7677}{s}$$

4. Discretizaciones

Como lo especifica la guía de laboratorio, se implementaron 6 métodos de discretización: invarianza del impulso, invarianza del paso (ZOH), aproximación Backward, aproximación Forward, aproximación Tustin, y mapeo de polos y ceros. 4 de estas se realizaron mediante la función de Matlab "c2d", mientras que las aproximaciones Forward y Backward se realizaron con ayuda del toolbox de matemática simbólica de MATLAB, realizando un cambio de variable $s = (z-1)/T_s$ para la aproximación Forward y un cambio de variable $s = (z-1)/(zT_s)$ para la aproximación Backward. El periodo de muestreo de todas las discretizaciones, fue de 5ms.

En la tabla 1 se presentan las funciones de transferencia resultantes para cada método de discretización, adicionalmente se coloca el ancho de banda como método de verificación de que el periodo de muestreo es lo suficientemente pequeño para que el controlador funcione apropiadamente. En esta caso dado que la frecuencia de muestreo es de 200Hz, y por lo tanto la frecuencia Nyquist es de 100Hz, y además se requiere que esta última sea un orden de magnitud mayor al ancho de banda de los controladores, se concluye que los controladores no deben superar un ancho de banda de 10Hz, cumpliendo con el criterio para todas las discretizaciones.

Tabla 1: Funciones de Transferencia de las discretizaciones

Técnica de discretización	Función de Transferencia	Ancho de banda
Invarianza del Impulso	$\frac{0.1588z - 4.988 \cdot 10^{-17}}{z-1}$	9.2564 Hz
Invarianza del paso	$\frac{0.7477z - 0.5889}{z-1}$	8.9244 Hz
Aproximación Forward	$\frac{0.7477z - 0.5889}{z-1}$	8.9244 Hz
Aproximación Backward	$\frac{0.9066z - 0.7477}{z-1}$	9.305 Hz
Aproximación Tustin	$\frac{0.8271z - 0.6683}{z-1}$	9.0849 Hz
Mapeo de polos y ceros	$\frac{0.83z - 0.6711}{z-1}$	9.0916 Hz

En la figura 4 se muestra una captura con la respuesta al paso de todas las discretizaciones, donde se evidencia que todas tienen un comportamiento muy similar al controlador continuo, con prácticamente el mismo tiempo subida (cumpliendo con el requerido de menor a 45.3ms), a excepción de la discretización obtenida mediante el método de la invarianza del impulso, donde el comportamiento cambia drásticamente a un sistema de segundo orden, con un sobre pico cercano al 30% y un tiempo de establecimiento considerablemente más grande, de alrededor de 250ms, en contraposición a los 62.3ms del controlador continuo, que prevalece aproximadamente en el resto de discretizaciones.

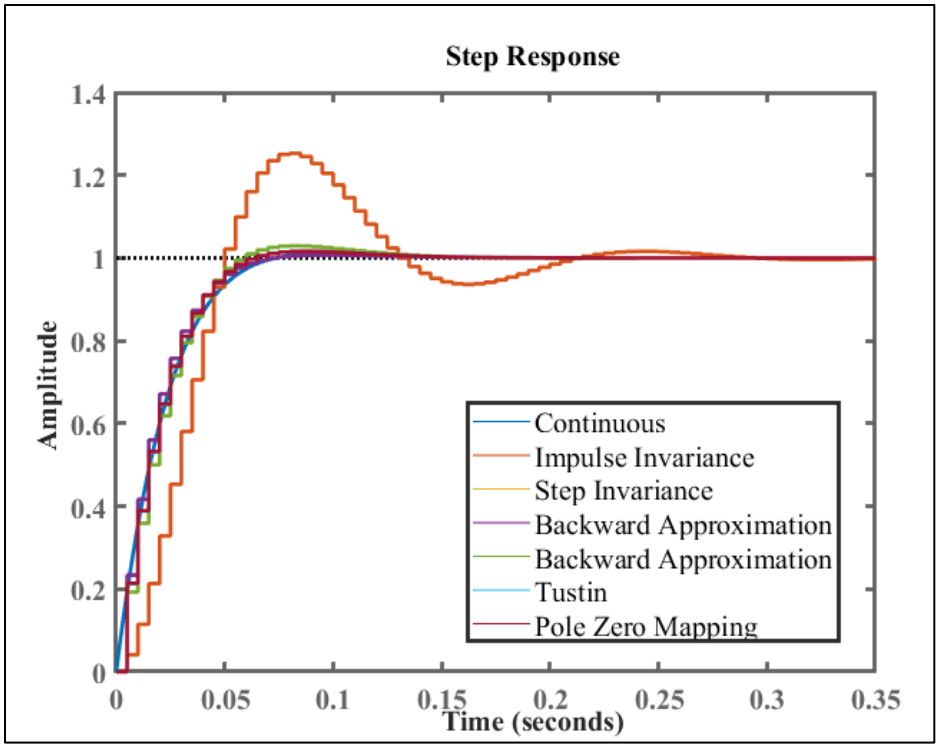

Figura 4: Respuesta al paso de cada método de discretización.

5. Simulación:

Dado que se requiere implementar algunos de estos controladores en un microcontrolador, se tomaron las funciones de transferencia y se convirtieron en su forma de potencias negativas mediante MATLAB, de esta forma cualquier función de transferencia puede ser implementada en el microcontrolador de la siguiente forma, donde e_{-n} representa el valor de la señal de entrada n muestras atrás, y u_{-m} representa el valor de la señal de salida, m muestras atrás.

$$G(z) = \frac{U(z)}{E(z)} = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}}{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_m z^{-m}} \rightarrow$$

$$u = \frac{(a_0 e + a_1 e_{-1} + a_2 e_{-2} + \cdots a_n e_{-n}) - (b_1 u_{-1} + b_2 u_{-2} + \cdots b_m u_{-m})}{b_0}$$

Las 6 discretizaciones fueron simuladas mediante Simulink, usando el diagrama de bloques que se muestra en la figura 5, donde se busca reflejar de la forma más precisa posible el sistema real, para ello se utiliza la planta continua, junto con retenedores de orden cero en la señal de realimentación, y en la salida del controlador, simulando el efecto del muestreo del encoder y la aplicación del voltaje al motor mediante PWM, también se incluye un bloque saturación para reflejar los límites físicos de la señal de control que alimenta el motor, y un retenedor de orden cero adicional para reflejar el comportamiento discreto de la señal de referencia. El controlador fue implementado como una función de transferencia en z.

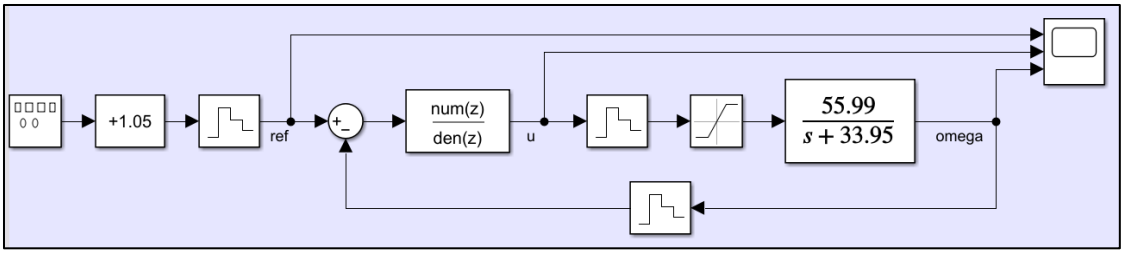

Figura 5: Diagrama de bloques de la simulación

De la Figura 6 a la figura 12 se muestra la respuesta a un señal de referencia cuadrada de 0.25Hz, que varía entre 0.8 y 1.3 rpm para el controlador continuo y para todas las discretizaciones, usando Simulink con el diagrama de bloques que se muestra en la figura 5, graficando además la señal de control, con el fin de verificar que no haya saturación. La señal de control se muestra en color azul, la referencia en amarillo, y la salida en naranja.

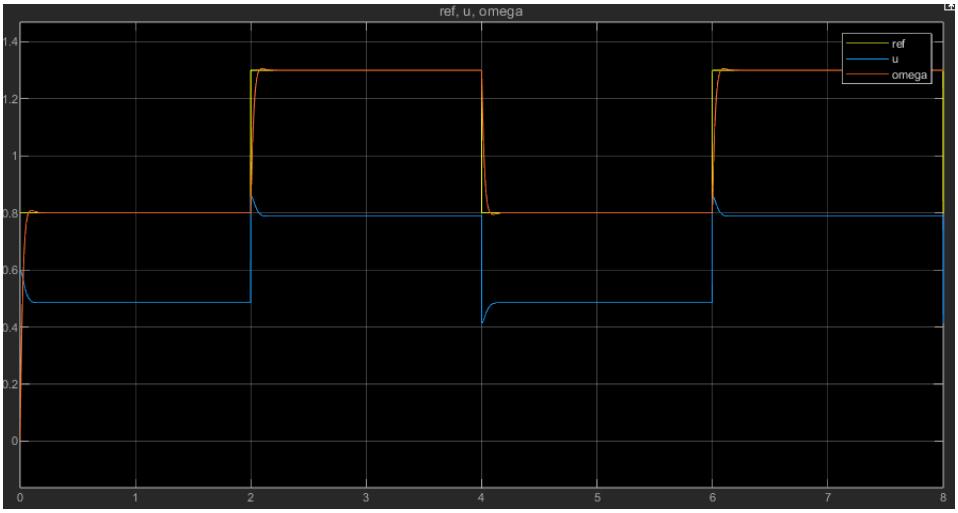

Figura 6: Respuesta del controlador continuo.

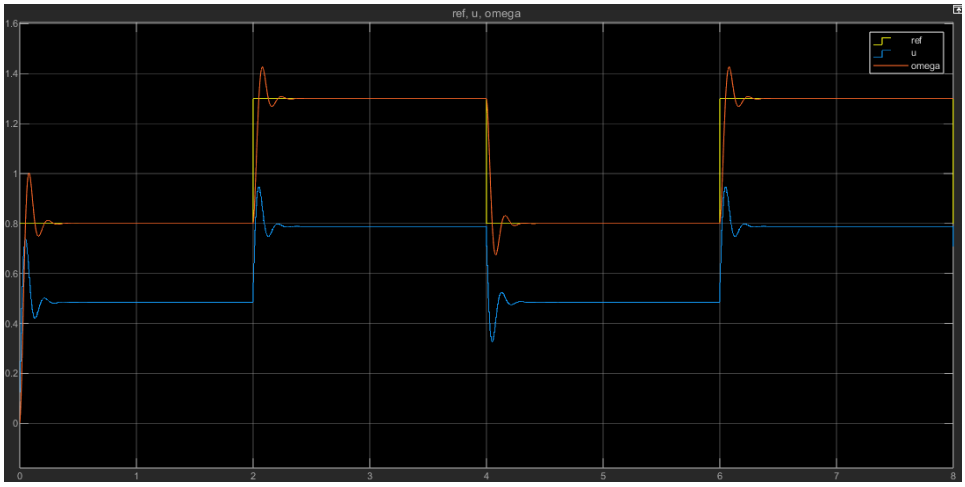

Figura 7: Respuesta de la discretización por invarianza del impulso.

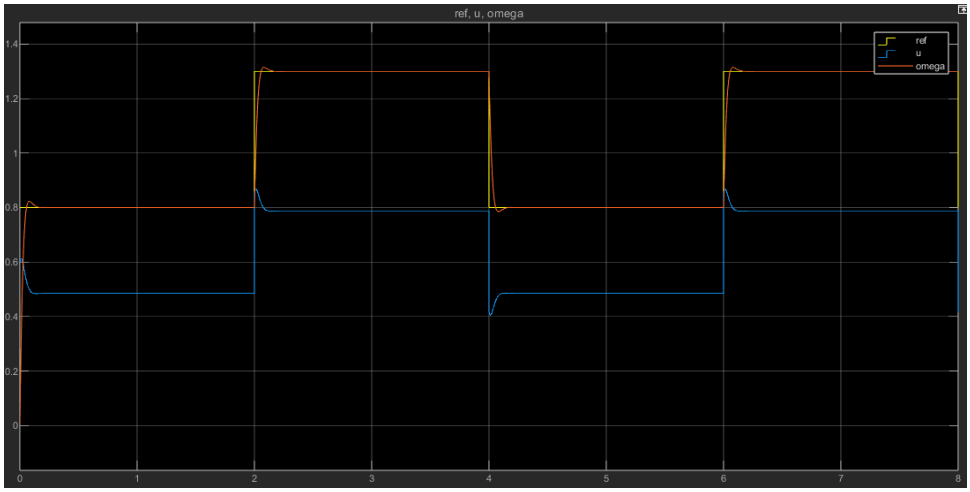

Figura 8: Respuesta de la discretización por invarianza del paso (ZOH).

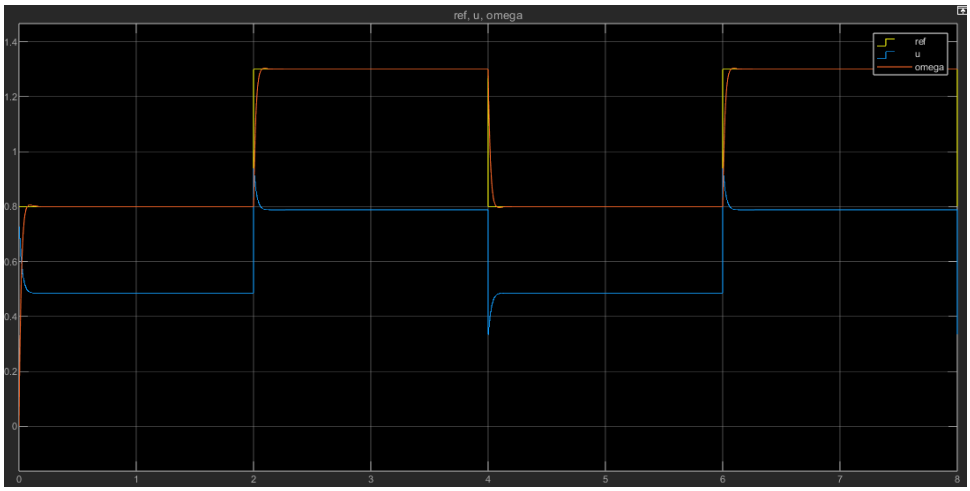

Figura 9: Respuesta de la discretización por aproximación Backward.

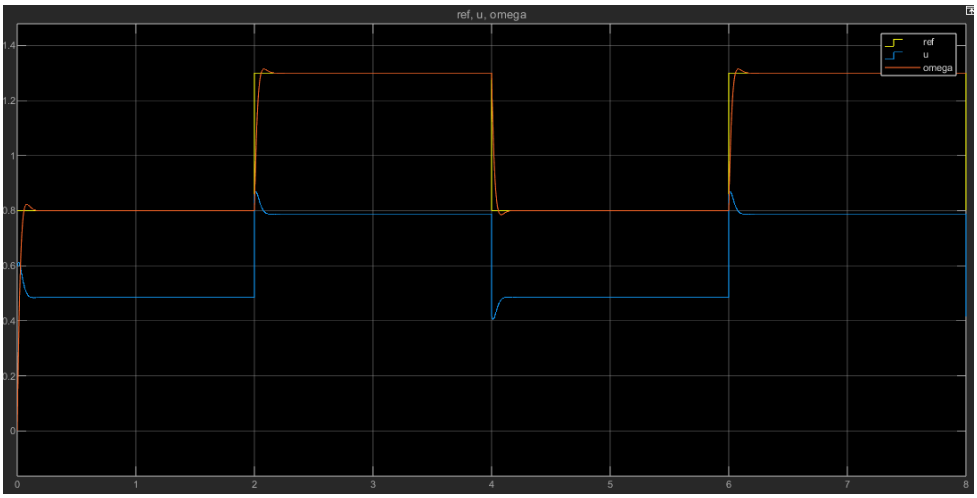

Figura 10: Respuesta de la discretización por aproximación Forward.

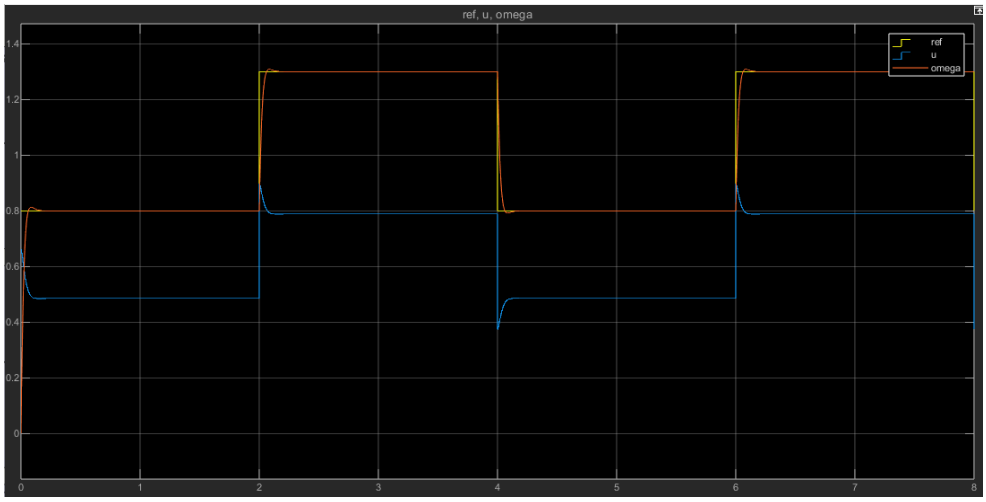

Figura 11: Respuesta de la discretización por aproximación Tustin.

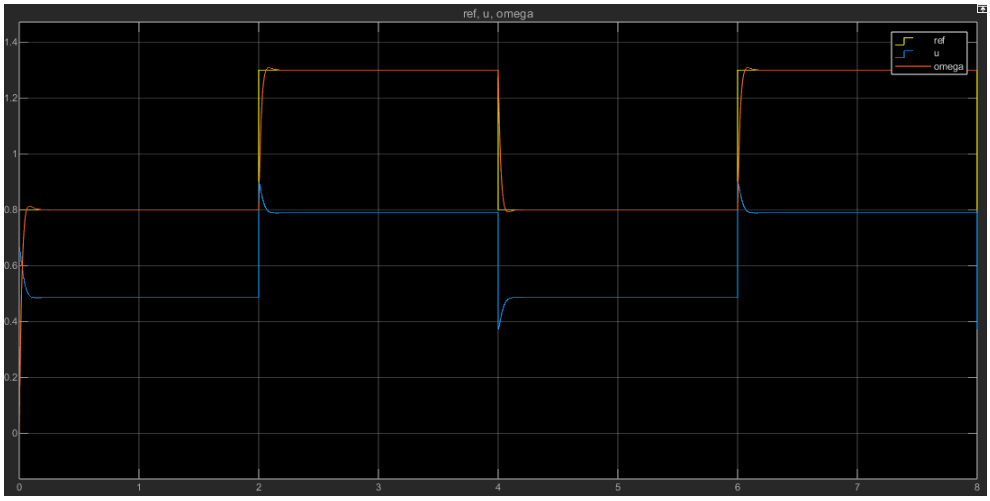

Figura 12: Respuesta de la discretización por mapeo de polos y ceros.

En las anteriores figuras, se evidencia que nunca se da la saturación, es decir la señal de control nunca alcanza el límite de 1, por lo cual se procede a realizar la implementación de los controladores.

6. Implementación y Resultados

Se seleccionaron 3 discretizaciones para implementar en el microcontrolador: invarianza del impulso, invarianza del paso y aproximación Tustin. Para comparar los resultados de la implementación con simulaciones, se seleccionó una señal cuadrada de 0.25Hz que va desde 0.8 a 1.3 rpm, se capturaron los datos de Simulink y del controlador implementado en el microcontrolador, para cada discretización, luego se exportaron los datos a MATLAB, donde se alinearon de tal forma que las señales de referencia coincidiesen en el tiempo, para finalmente imprimir los datos en una única grafica para cada discretización, incluyendo las señales de control. Los resultados se muestran en las Figura 13, 14 y 15.

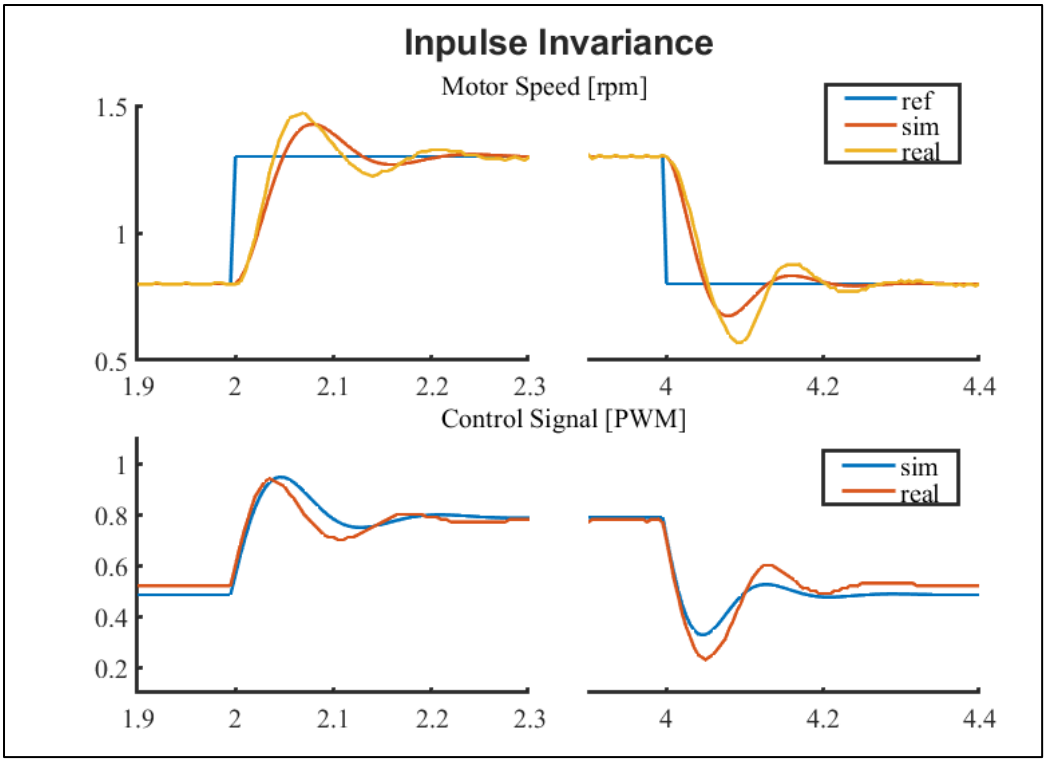

Figura 13: Resultados del controlador de invarianza del impulso

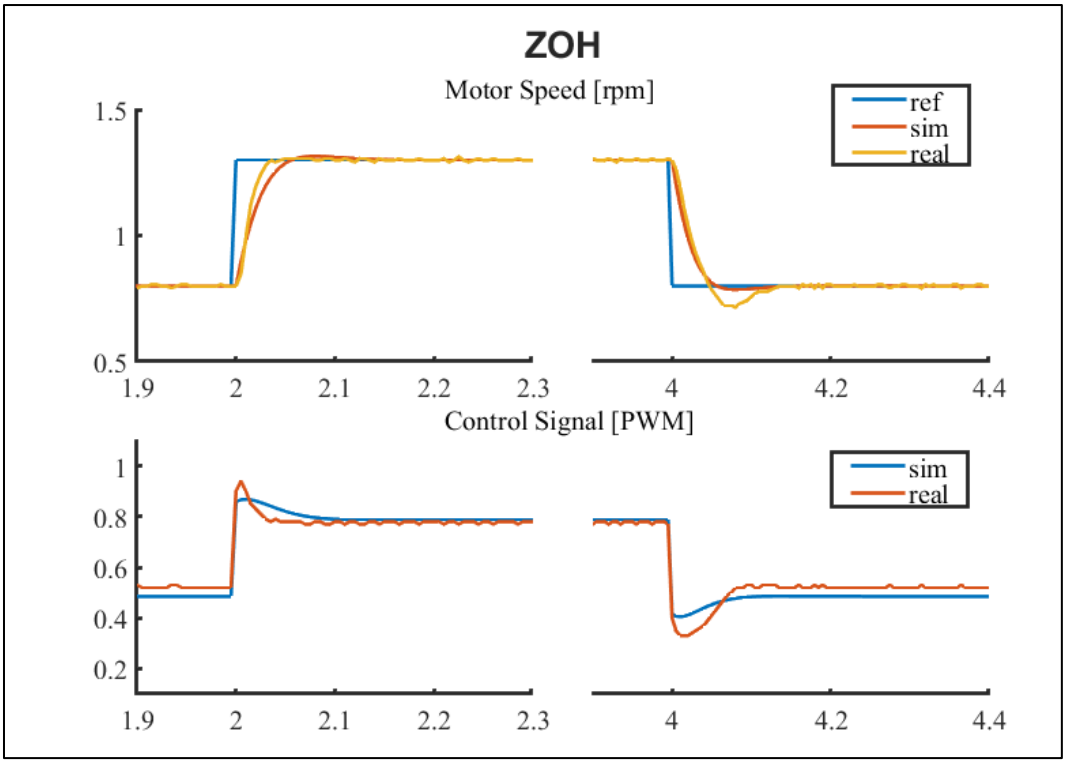

Figura 14: Resultados del controlador de invarianza del paso.

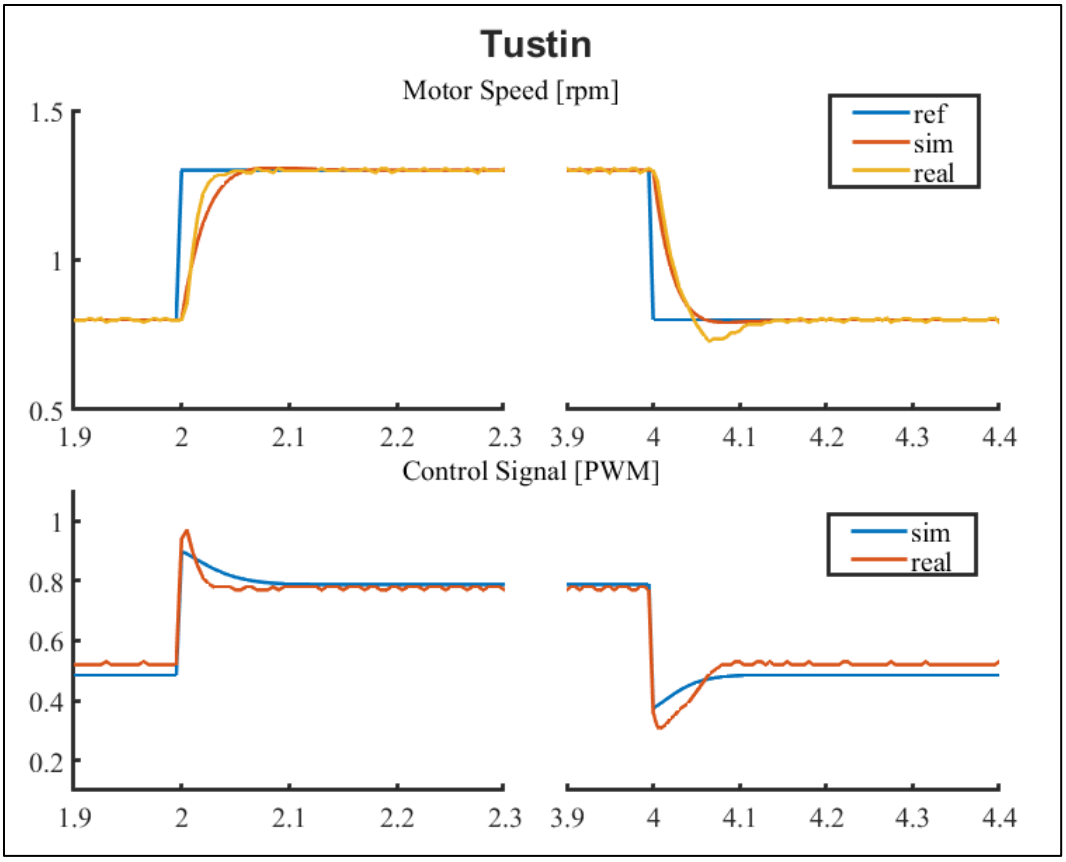

Figura 14: Resultados del controlador por aproximación Tustin.

7. Análisis y Conclusiones

Al analizar los resultados es posible concluir que:

- El modelo del motor de primer orden obtenido mediante MATLAB, modela muy bien la dinámica real del sistema, dado que el comportamiento entre los controladores implementados y las simulaciones es bastante similar, sin embargo el modelo no es lo suficientemente bueno, como para modelar apropiadamente el comportamiento decreciente del sistema, cuya dinámica en la realidad, presenta mayores sobre picos a los esperados por la simulación.
- Tal y como se esperaba, existe un error estacionario en la señales de control, entre el comportamiento real y el simulado, dado que el modelo hallado tuvo un error estacionario de alrededor del 12.5%, sin embargo esto no supone un problema ya que la componente integral del controlador, es capaz de lograr una respuesta al paso con un error de estado estacionario nulo.
- La aproximación por invariancia al impulso es sin lugar a duda, la que tuvo un peor desempeño en aproximar la respuesta del controlador continuo al tiempo discreto, de hecho presenta un sobre pico cercano al 50%, y aunque el tiempo de subida es similar, el tiempo de establecimiento es considerablemente mayor. Por otro lado las

aproximaciones por invariancia del paso y la aproximación Tustin, sí que tuvieron un comportamiento muy similar al controlador continuo, tanto para la simulación, como para la implementación real del sistema, logrando incluso un comportamiento más rápido en la implementación real, es decir que se cumple satisfactoriamente con los requerimientos especificados,