

# Tarea 1

Tema a tratar

Integrantes: Integrante 1  
Integrante 2  
Profesor: Profesor 1  
Auxiliar: Auxiliar 1  
Ayudantes: Ayudante 1  
Ayudante 2  
Ayudante de laboratorio: Ayudante 1

Fecha de realización: 11 de noviembre de 2024  
Fecha de entrega: 11 de noviembre de 2024  
Santiago de Chile

# Resumen

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

# Índice de Contenidos

1. Pregunta 2	1
---------------	---

## Índice de Figuras

1. Implementación de la planta no lineal en <i>Simulink</i> , junto con el código implementado para la planta, además de las diferentes entradas y salidas. . . . .	1
2. Implementación del ruido blanco gaussiano considerando una media de 0, una varianza de $\sigma^2 = (0.01)^2$ , y un tiempo de muestreo de $T_s = 1$ [s], usando el bloque <i>band limited white noise</i> . . . . .	2
3. Implementación del controlador PI en <i>Simulink</i> junto con la señal de referencia $r(t) = 2$ , además de las diferentes entradas y salidas del controlador. . . . .	2
4. Implementación del controlador PI en <i>Simulink</i> con sus valores de ganancia proporcional $K_p = 0.01$ y ganancia integral $K_i = 0.06$ . . . . .	3
5. Salida de la planta controlada, la cual cumple con el tiempo de establecimiento menor a 100 [s] además de no superar el límite del 10 % de sobrepaso. . . . .	3
6. Implementación del controlador difuso en <i>Simulink</i> donde se considera el bloque <i>Fuzzy Logic</i> , además de los ponderadores o ganancias $GE$ , $GDE$ y $GU$ . . . . .	4
7. Definición de los conjuntos difusos tanto para el error ( <b>e</b> ), la derivada del error ( <b>de</b> ) y por último la salida del controlador difuso ( <b>u</b> ). . . . .	5
8. Salida de la planta controlada donde se observa una respuesta suficientemente aceptable. . . . .	6
9. Implementación de la señal PRBS en <i>Simulink</i> para la obtención del modelo ARX. . . . .	7
10. Señal PRBS y la respuesta de la planta para la obtención del modelo ARX. . . . .	7
11. Implementación del modelo ARX con sus parámetros óptimos y comparación de datos. . . . .	8
12. Implementación del controlador MPC en <i>Simulink</i> con los coeficientes del modelo ARX. . . . .	9
13. Implementación de la señal de referencia $r(t)$ en <i>Simulink</i> para la comparación de los controladores. . . . .	10
14. Respuesta del controlador PI clásico para la señal de referencia $r(t)$ . . . . .	11
15. Respuesta del controlador PI difuso para la señal de referencia $r(t)$ . . . . .	12
16. Respuesta del controlador MPC para la señal de referencia $r(t)$ . . . . .	13
17. Comparación de las respuestas de los controladores para la señal de referencia $r(t)$ . . . . .	14

## Índice de Tablas

1. Parámetros de las Funciones de Membresía para las Variables de Entrada y Salida	5
--	---

---

2.	Reglas de Inferencia para el Controlador Difuso PI . . . . .	6
3.	Coeficientes del Modelo ARX . . . . .	8

# 1. Pregunta 2

En este problema se diseñarán distintos tipos de controladores para una planta no lineal, que obedece a la siguiente dinámica:

$$y(k) = \left(0.9 - \frac{0.4}{\sqrt{1 + [y(k-2)]^2}}\right) y(k-1) - \left(0.5 + \frac{0.7}{\sqrt{1 + [y(k-1)]^2}}\right) y(k-2) \quad (1)$$

$$+ u(k-1) - 0.3u(k-2) + 0.2u(k-1)u(k-2) + \epsilon(k) \quad (2)$$

donde  $\epsilon(k)$  es un ruido blanco gaussiano con media 0 y varianza  $\sigma^2 = (0.01)^2$ . Además, la planta tiene un tiempo de muestreo de  $T_s = 1$  [s]. Se requiere diseñar diferentes tipos de controladores, por lo que:

1. Diseñe un controlador PI clásico utilizando como referencia la señal  $r(t) = 2$ . Los requisitos de diseño del controlador son los siguientes:

- Sobrespaso menor al 10 %.
- Tiempo de estabilización menor a 100 [s].

Para la simulación del sistema se utilizará la herramienta *Simulink*. Primero, es necesario implementar la planta no lineal del sistema mediante el bloque *MATLAB Function*, que se muestra en la Figura 1.

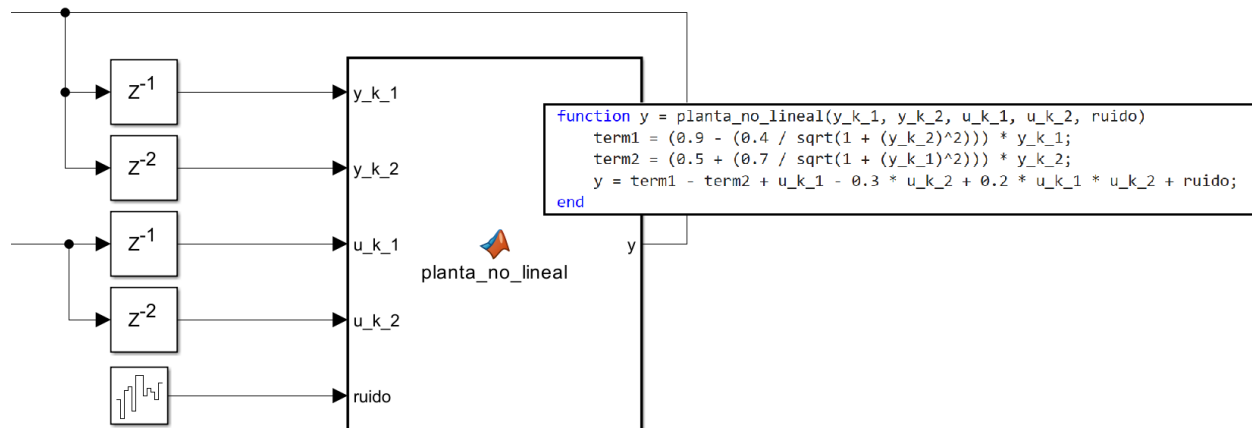


Figura 1: Implementación de la planta no lineal en *Simulink*, junto con el código implementado para la planta, además de las diferentes entradas y salidas.

Con base en la función implementada en el bloque *MATLAB Function*, se observa que existen 5 entradas: dos de ellas corresponden al retardo de la señal de control  $u(k-1)$  y  $u(k-2)$ ; otras dos están asociadas a la señal de salida de la planta  $y(k-1)$  y  $y(k-2)$ ; y

finalmente, la señal de ruido blanco  $\epsilon(k)$ , implementada mediante una función de ruido blanco gaussiano con media 0 y varianza  $\sigma^2 = (0.01)^2$ .

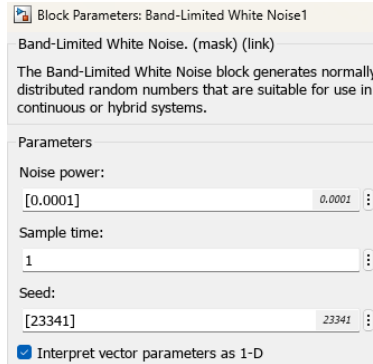


Figura 2: Implementación del ruido blanco gaussiano considerando una media de 0, una varianza de  $\sigma^2 = (0.01)^2$ , y un tiempo de muestreo de  $T_s = 1$  [s], usando el bloque *band limited white noise*.

A continuación, se plantea el siguiente esquema general para la implementación del controlador PI, utilizando el bloque *PID Controller* de *Simulink*. Además, la entrada de referencia corresponde a un escalón *Step* con la forma  $r(t) = 2$ , como se muestra en la Figura 3.

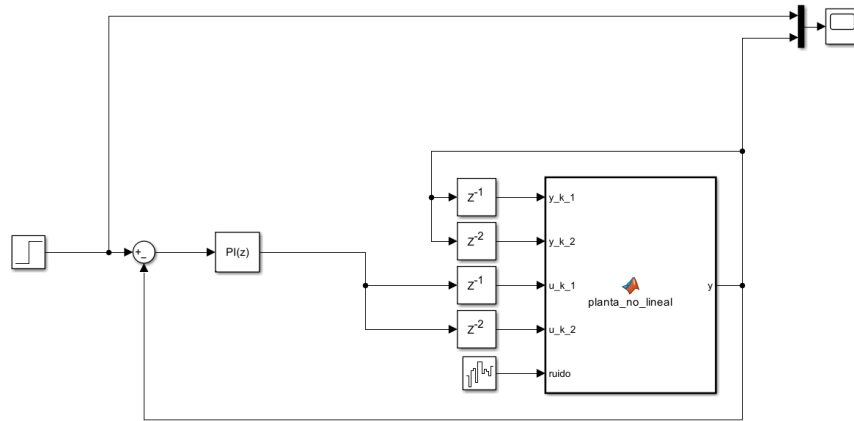


Figura 3: Implementación del controlador PI en *Simulink* junto con la señal de referencia  $r(t) = 2$ , además de las diferentes entradas y salidas del controlador.

Con el bloque del controlador PI discreto, se ajustan los parámetros asociados a la ganancia proporcional  $K_p$  y la ganancia integral  $K_i$ , obteniéndose los siguientes valores:

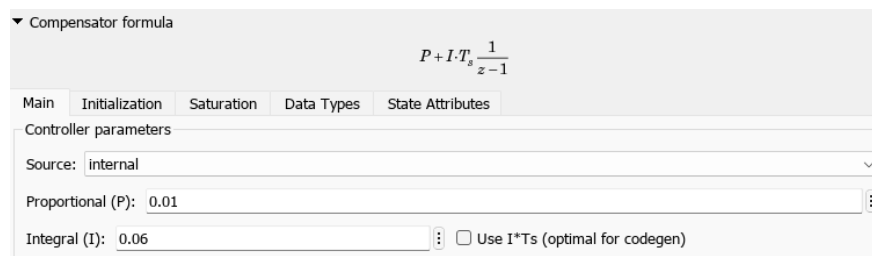


Figura 4: Implementación del controlador PI en *Simulink* con sus valores de ganancia proporcional  $K_p = 0.01$  y ganancia integral  $K_i = 0.06$ .

Finalmente, se obtiene la salida de la planta controlada, que se muestra a continuación:

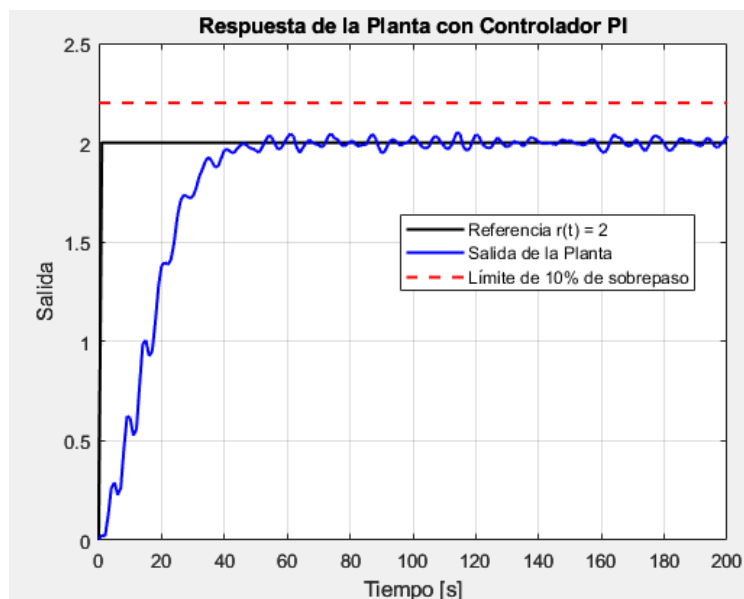


Figura 5: Salida de la planta controlada, la cual cumple con el tiempo de establecimiento menor a 100 [s] además de no superar el límite del 10 % de sobrepaso.

En conclusión, el controlador PI diseñado cumple con los requisitos especificados en cuanto al tiempo de estabilización y el límite de sobrepaso, y su implementación es simple y eficiente.

**2. Diseñe un controlador PI difuso considerando 3 conjuntos difusos para cada premisa y para cada consecuencia en la construcción de las reglas.**

Se busca realizar un controlador PI difuso, para lo cual se considerará la misma estructura de la planta mostrada anteriormente en la Figura 1. Para la implementación del controlador difuso se utilizará el bloque *Fuzzy Logic* de *Simulink*. A continuación, se muestra la implementación del controlador difuso en *Simulink* junto con la señal de referencia  $r(t) = 2$ , además de las diferentes entradas y salidas del controlador.

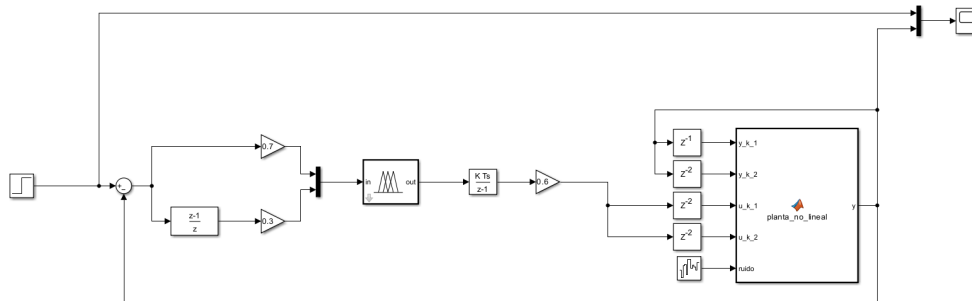


Figura 6: Implementación del controlador difuso en *Simulink* donde se considera el bloque *Fuzzy Logic*, además de los ponderadores o ganancias  $GE$ ,  $GDE$  y  $GU$ .

Luego se deben definir tanto las reglas asociadas al controlador difuso como los conjuntos difusos para cada premisa y para cada consecuencia en la construcción de las reglas. Se definen los siguientes conjuntos difusos para cada premisa asociada al error, la derivada del error y por último la salida del controlador difuso.



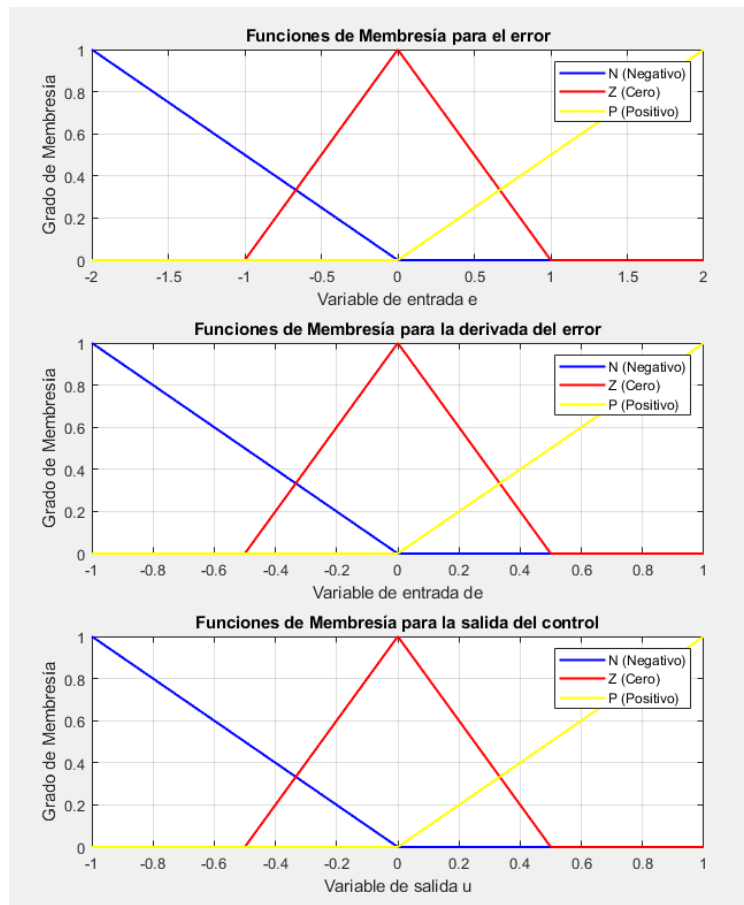


Figura 7: Definición de los conjuntos difusos tanto para el error (**e**), la derivada del error (**de**) y por último la salida del controlador difuso (**u**).

Los rangos se visualizan de mejor manera en la siguiente tabla:

Tabla 1: Parámetros de las Funciones de Membresía para las Variables de Entrada y Salida

Variable	Conjunto Difuso	Tipo	Parámetros
Error ( $e$ )	N (Negativo)	Triangular	$[-3, -2, 0]$
	Z (Cero)	Triangular	$[-1, 0, 1]$
	P (Positivo)	Triangular	$[0, 2, 3.667]$
Derivada del Error ( $de$ )	N (Negativo)	Triangular	$[-1, -1, 0]$
	Z (Cero)	Triangular	$[-0.5, 0, 0.5]$
	P (Positivo)	Triangular	$[0, 1, 1]$
Salida ( $u$ )	N (Negativo)	Triangular	$[-1, -1, 0]$
	Z (Cero)	Triangular	$[-0.5, 0, 0.5]$
	P (Positivo)	Triangular	$[0, 1, 1.833]$

Luego, se deben definir las reglas difusas asociadas al controlador difuso, las cuales se muestran en la siguiente tabla:

Tabla 2: Reglas de Inferencia para el Controlador Difuso PI

Regla	Condición	Resultado	Salida (u)
1	Si $e$ es N y $de$ es N	Entonces $u$ es	N
2	Si $e$ es N y $de$ es Z	Entonces $u$ es	N
3	Si $e$ es N y $de$ es P	Entonces $u$ es	Z
4	Si $e$ es Z y $de$ es N	Entonces $u$ es	P
5	Si $e$ es Z y $de$ es Z	Entonces $u$ es	Z
6	Si $e$ es Z y $de$ es P	Entonces $u$ es	N
7	Si $e$ es P y $de$ es N	Entonces $u$ es	Z
8	Si $e$ es P y $de$ es Z	Entonces $u$ es	P
9	Si $e$ es P y $de$ es P	Entonces $u$ es	P

Una vez implementado todo lo anterior asociado al controlador difuso, se obtiene la siguiente salida de la planta controlada.

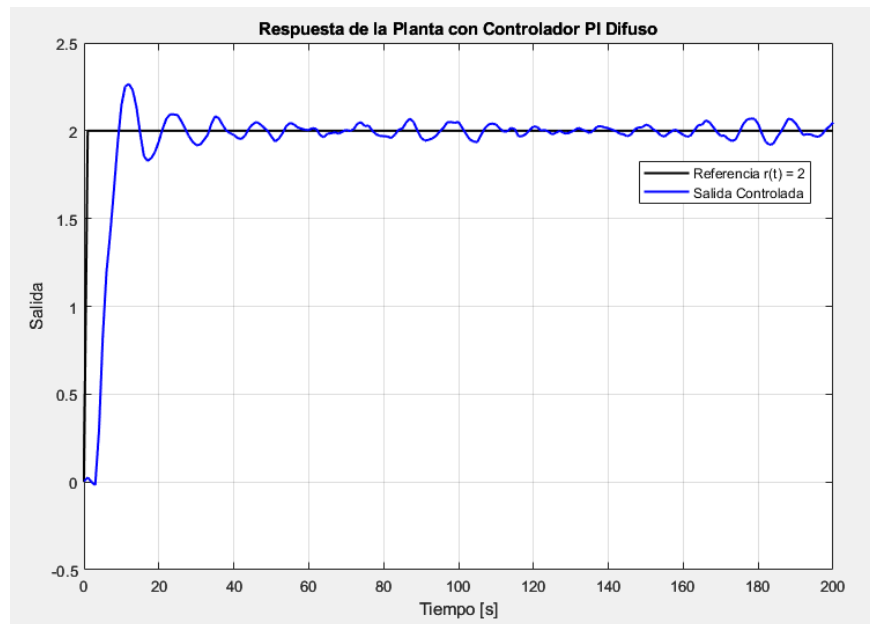


Figura 8: Salida de la planta controlada donde se observa una respuesta suficientemente aceptable.

Finalmente, se puede concluir que el controlador PI difuso cumple con los requisitos de diseño y muestra una respuesta aceptable.

3. Diseñe e implemente un controlador MPC basado en un modelo ARX, indicando su estructura completa y probando con distintos parámetros de sintonización.

Ahora se procederá a diseñar un controlador MPC basado en un modelo ARX. Para obtener los parámetros del modelo, se utiliza una señal PRBS en la entrada del sistema. La estructura básica se muestra en la Figura 9.

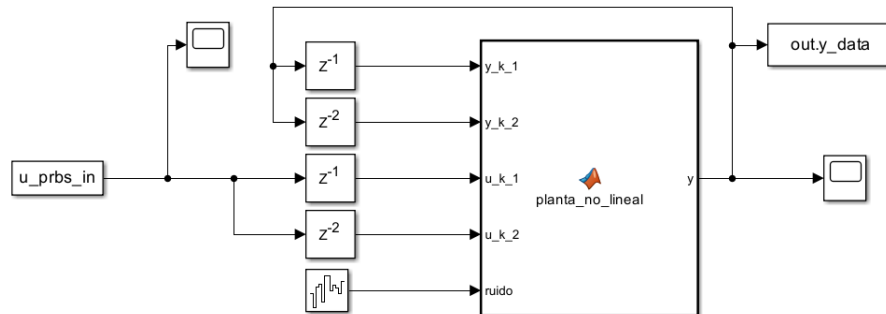


Figura 9: Implementación de la señal PRBS en *Simulink* para la obtención del modelo ARX.

Luego, se registran las respuestas de la planta a la señal PRBS, como se ilustra en la Figura 10.

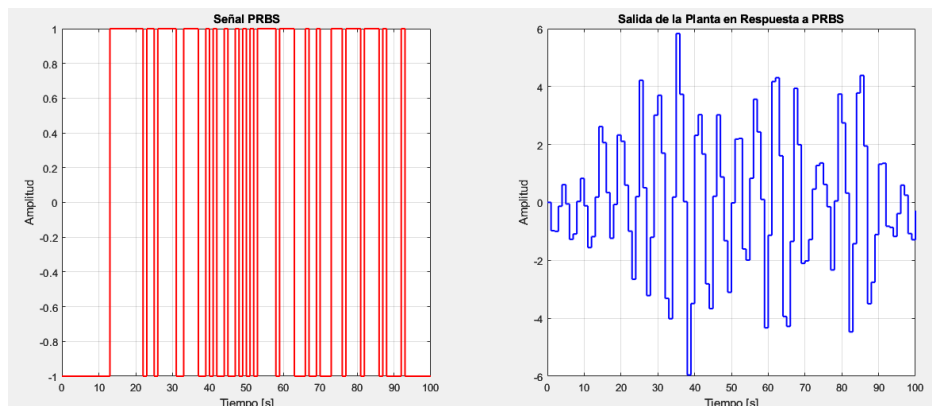


Figura 10: Señal PRBS y la respuesta de la planta para la obtención del modelo ARX.

Mediante un script en *MATLAB*, se realiza la identificación del modelo ARX, obteniendo los siguientes coeficientes para el modelo:

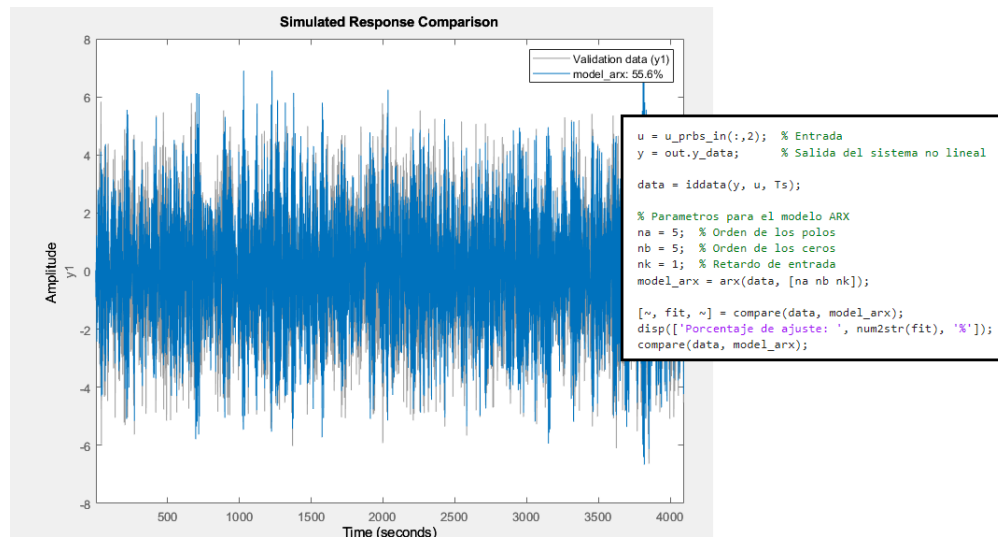


Figura 11: Implementación del modelo ARX con sus parámetros óptimos y comparación de datos.

Los coeficientes obtenidos son los siguientes:

Tabla 3: Coeficientes del Modelo ARX

Orden	Coeficientes A (Salida)	Coeficientes B (Entrada)
1	-0.2229	0.9953
2	0.6442	0.2711
3	0.3981	-0.0153
4	0.0495	0.0596
5	0.0470	-0.0211

Finalmente, se implementa el controlador MPC en *Simulink* utilizando los coeficientes del modelo ARX, como se muestra en la Figura 12.

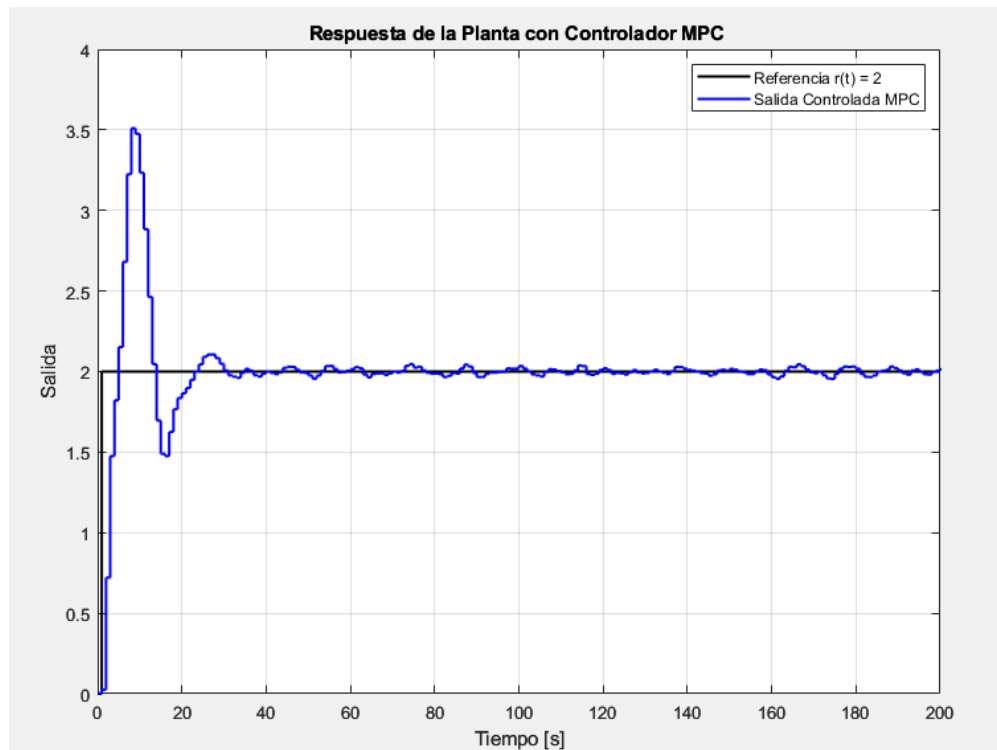


Figura 12: Implementación del controlador MPC en *Simulink* con los coeficientes del modelo ARX.

Se concluye que el controlador MPC proporciona una respuesta aceptable, aunque presenta algunos sobressaltos que deben considerarse para mejorar la estabilidad y precisión del sistema.

4. Comparar el desempeño de los 3 controladores anteriores cuando ocurre un cambio en la referencia. Haga referencia a las métricas que considere pertinente. En particular, utilice la siguiente referencia:

$$r(t) = \begin{cases} 2 & t \leq 200 \text{ [s]} \\ 5 & 200 \leq t < 400 \text{ [s]} \\ 8 & 400 \leq t \leq 600 \text{ [s]} \end{cases}$$

Dado que se busca comparar el desempeño de los controladores, se considerara diferentes escalones, por lo que implementara de la siguiente manera:

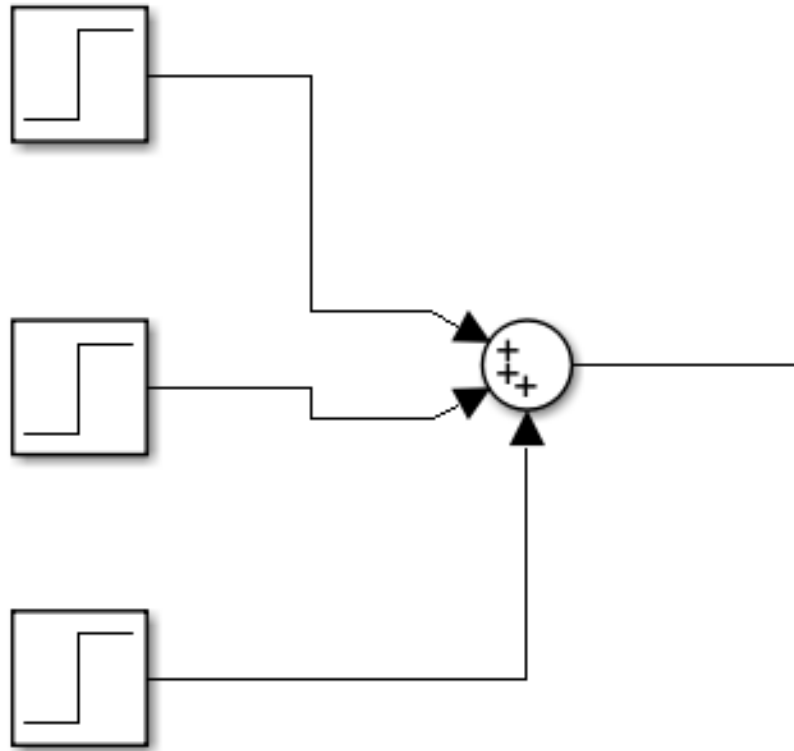


Figura 13: Implementación de la señal de referencia  $r(t)$  en *Simulink* para la comparación de los controladores.

Luego tendremos que la respuesta para el controlador PI clásico es la siguiente:

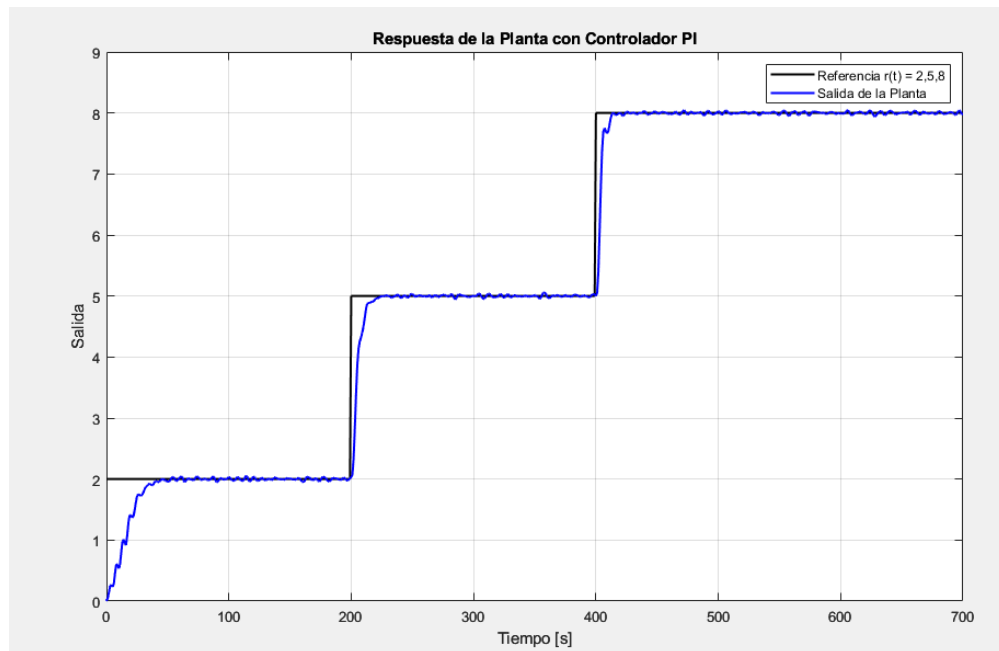


Figura 14: Respuesta del controlador PI clásico para la señal de referencia  $r(t)$ .

Para el controlador PI difuso, se obtiene la siguiente respuesta:

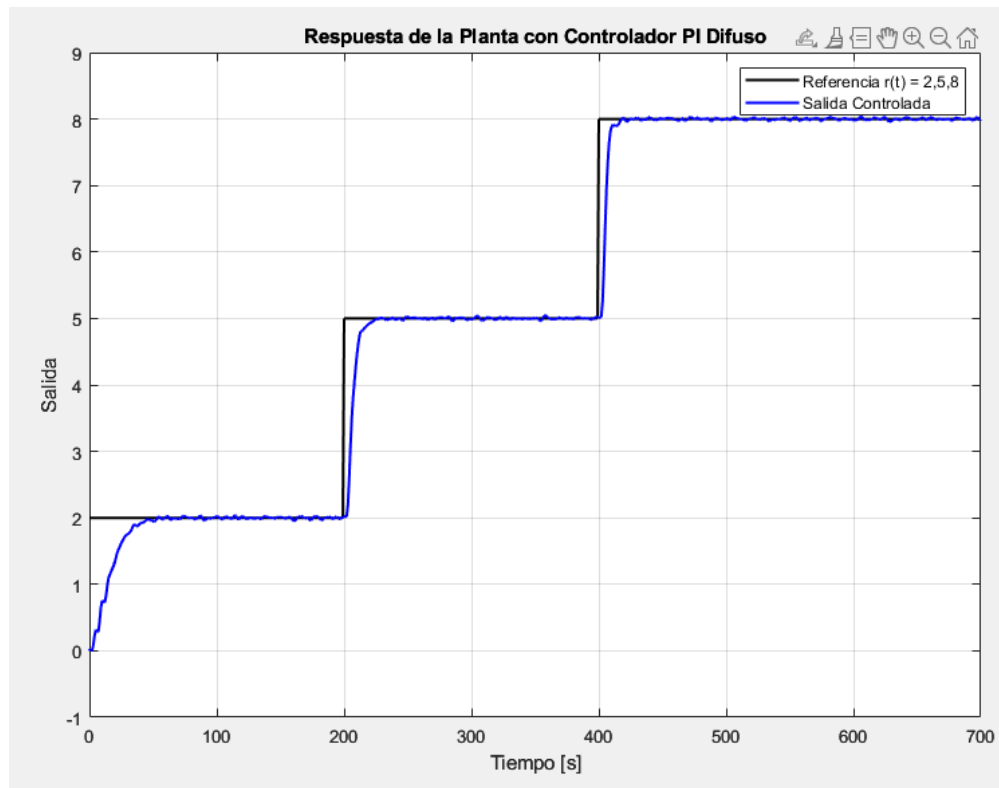


Figura 15: Respuesta del controlador PI difuso para la señal de referencia  $r(t)$ .

Finalmente tenemos que para el controlador MPC, se obtiene la siguiente respuesta:



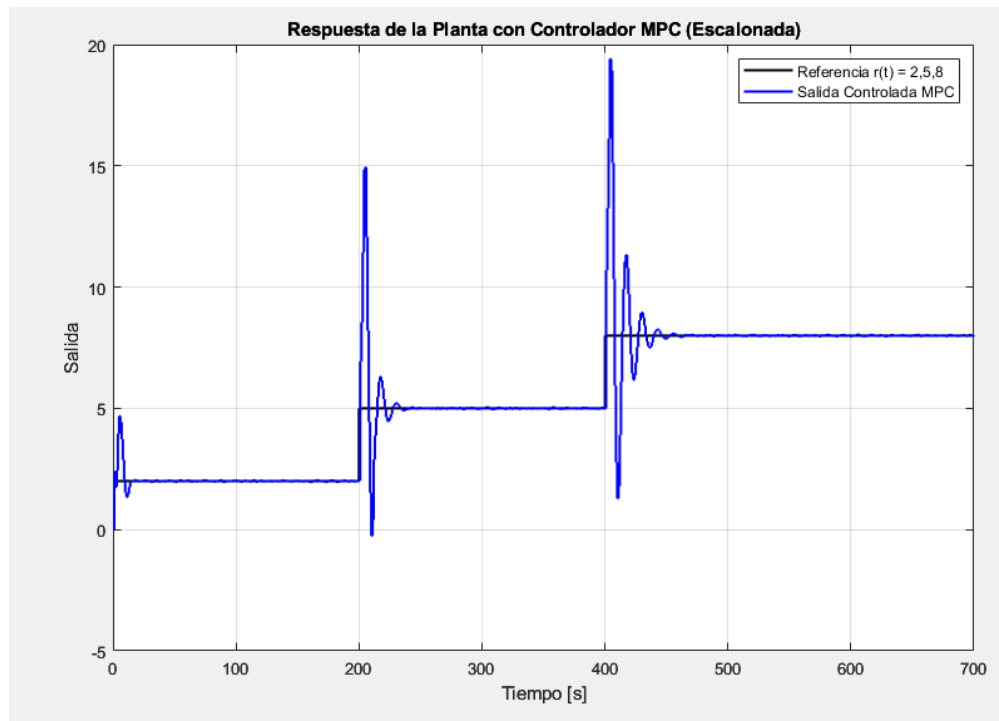


Figura 16: Respuesta del controlador MPC para la señal de referencia  $r(t)$ .

Podemos comparar los graficos dando como resultado:

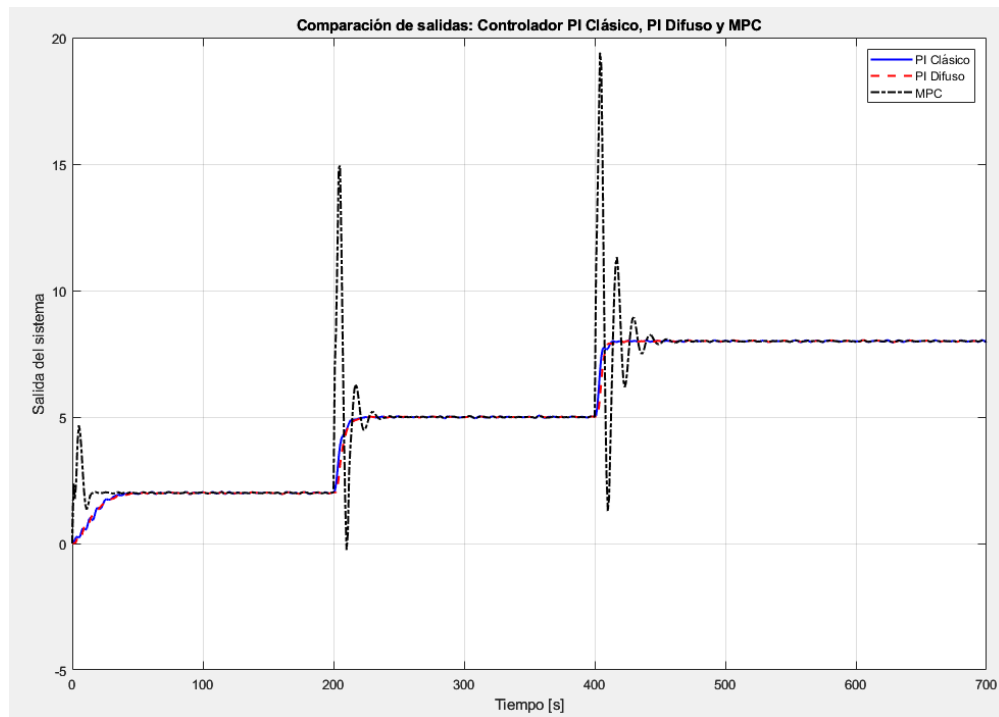


Figura 17: Comparación de las respuestas de los controladores para la señal de referencia  $r(t)$ .

Se puede concluir que