

Clase 6 – Control Óptimo y difuso para distintos plantas

Para este laboratorio se hará uso del Helicopter y del Crane

Archivos para el Helicóptero:

<https://quanserinc.box.com/shared/static/zztk0rpahb06fg6ybj32jafjno9m4zlm.zip>

Archivos para el Crane:

<https://quanserinc.box.com/shared/static/jta1y17m8wp2i073dn4pcli20nw8m4y.zip>

Archivo Fuzzy para el Helicopter:

https://drive.google.com/drive/folders/19oh06eoreAngOFuQE52uBLzyWPP_YfHI?usp=sharing

Objetivos del Laboratorio:

Comprender el controlador actual utilizado por cada planta no lineal (Helicóptero y Crane).

Modificar la ganancia LQR del Helicóptero para reducir el overshoot del ángulo pitch en al menos 15%.

Implementar un controlador difuso para el helicóptero siguiendo el documento Fuzzy_heli.pdf.

Ajustar el "LQR" del Crane para obtener un comportamiento más agresivo del Trolley/JIB.

Ejecutar la simulación y visualizar la respuesta en tiempo real, documentando resultados.

Instrucciones:

Helicóptero – LQR

Complete las siguientes instrucciones para el Helicóptero

- **Identificar el controlador original**
 - Abrir el archivo del modelo del helicóptero.
 - Localizar el bloque donde aparece la ley de control típica:
$$u = -kx$$
 - Anotar la matriz de Ganancias original:

Matriz de ganancias originales:

Calculated LQR controller gain elements:

K =

51.9211	16.1899	-16.1293	24.6004	5.2787	-21.2682	14.1421	-1.4142
51.9211	-16.1899	16.1293	24.6004	-5.2787	21.2682	14.1421	1.4142

- **Modificación del LQR (Reducción de overshoot)**
 - Abrir el archivo donde se definen Q y R del LQR.
 - **Aumentar selectivamente los pesos asociados a:**
 - Error del ángulo pitch
 - Velocidad angular pitch

```
elseif strcmp(WITH_ADS, 'YES')
    Q = diag([100 20 10 0 5 2 10 0.1]); % 15
    R = 0.025*diag([1 1]);
end
```

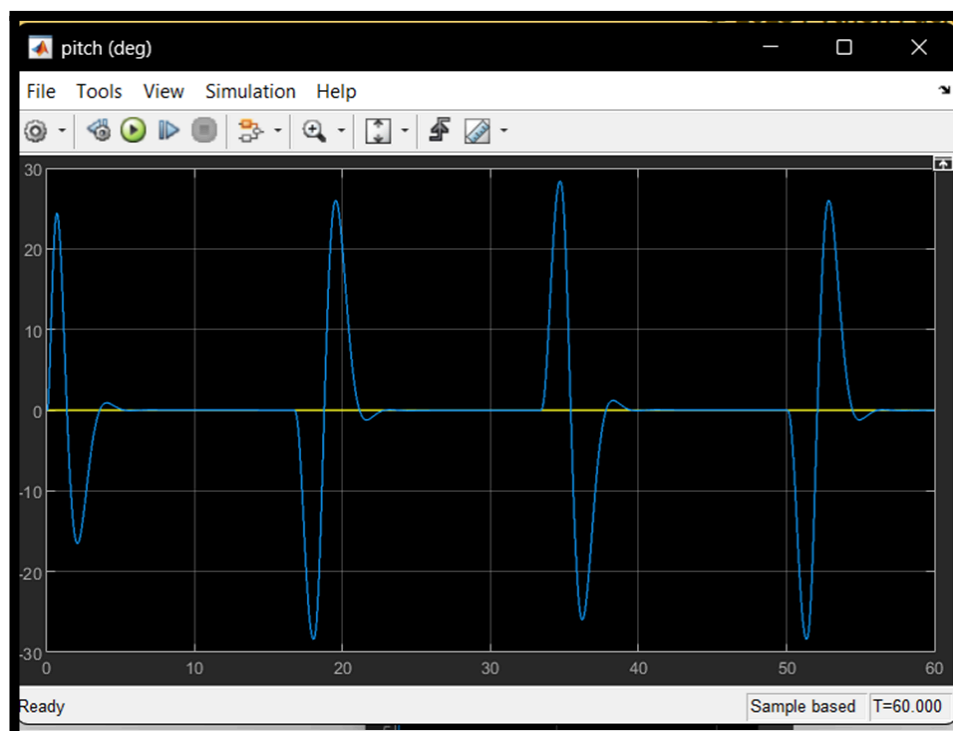
- **Generar la nueva ganancia:**
 $K_{new} = lqr(A, B, Q, R);$
Matriz obtenida tras modificar Q y R.

```
K_new =

51.9211    37.1573   -17.0668    24.6004    12.8044   -32.2704    14.1421   -1.4142
51.9211   -37.1573    17.0668    24.6004   -12.8044    32.2704    14.1421    1.4142
```

- **Reemplazar la ganancia en el modelo Simulink o Python como usted desee.**
- **Ejecutar la simulación (añadir capturas de pantallas) y anotar:**
 - Overshoot original
 - Overshoot modificado
 - Tiempo de establecimiento

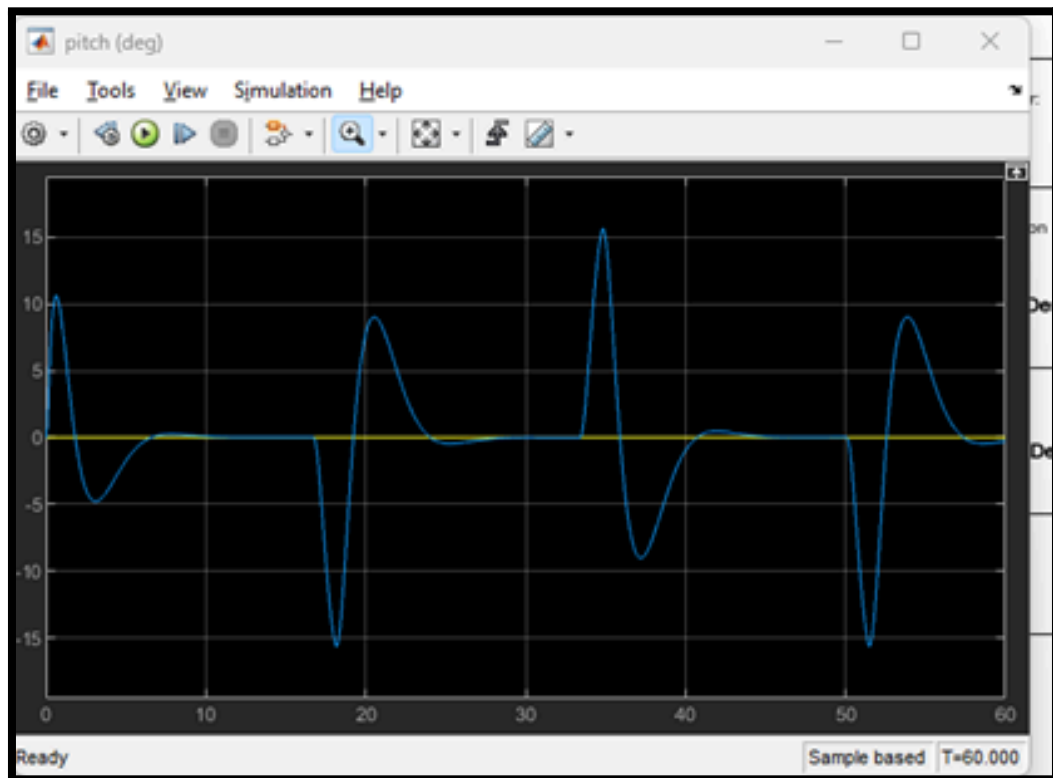
SIMULACIÓN ORIGINAL:



Overshoot: 24.42

Tiempo de establecimiento: 6.3s

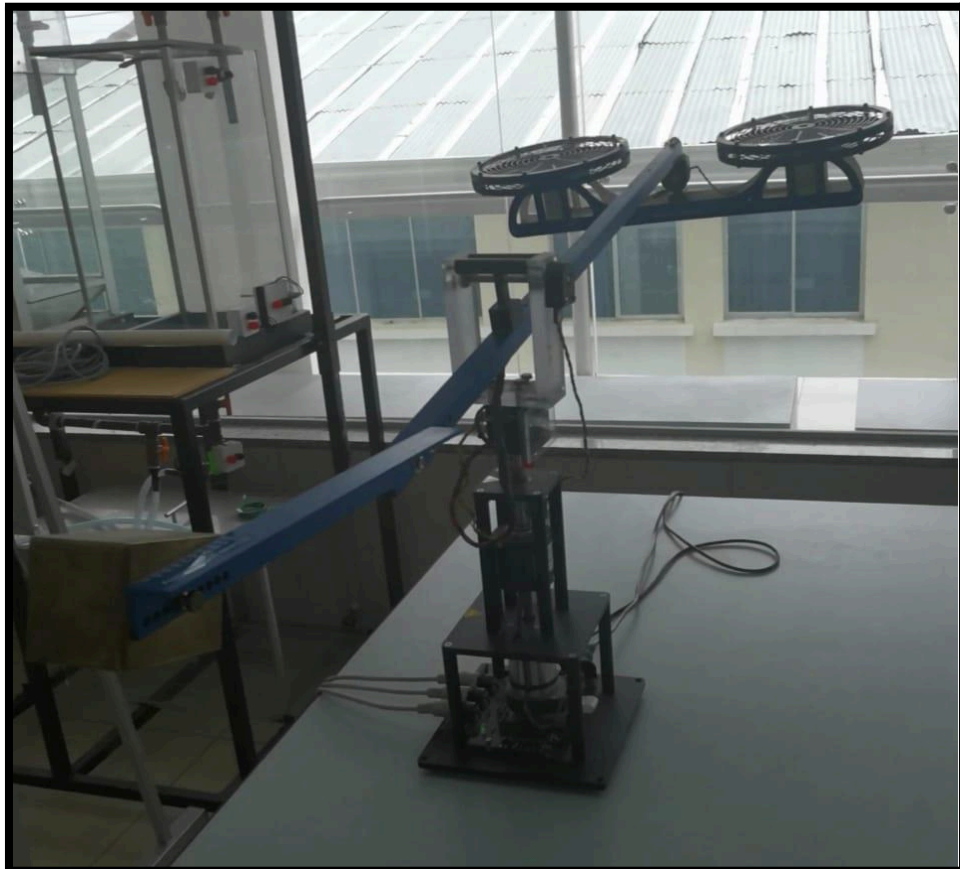
SIMULACIÓN DE LOS NUEVOS VALORES:



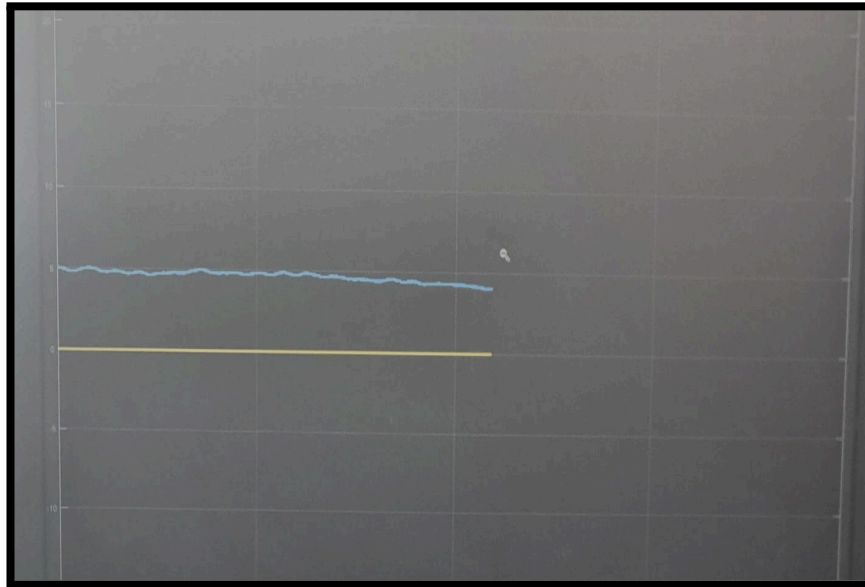
Overshoot: 15

Tiempo de establecimiento: 12s

- Implemente las nuevas ganancias y adjunte capturas de pantalla del pitch.



IMPLEMENTACIÓN DE LOS NUEVOS VALORES



Se logró reducir el overshoot a un 15%, aunque se pudo observar leves oscilaciones durante el tiempo de establecimiento.

Helicóptero – Control Difuso

- Revisar el documento Fuzzy_heli.pdf
 - Comprender las variables de entrada “e” y “de”, la base de reglas y la función de defuzzificación utilizada.

En el bloque del controlador Fuzzy se rige mediante la siguiente matriz de reglas difusas:

DE	E		
	NEG	Z	POS
NEG	VS	S	M
Z	S	M	H
POS	M	H	VH

El diseño del controlador de lógica difusa utiliza dos entradas y una salida.

Elemento	Variable	Símbolo	Subconjuntos Lingüísticos	Rango Diseñado
Entrada 1	Error	e	NEG, Z, POS	[-0.2, 0.2]
Entrada 2	Cambio en el error	de	NEG, Z, POS	[-0.3, 0.3]
Salida	Respuesta del Sistema	Output	VS, S, M, H, VH	[-1.6, 1.6]

Error (e): La diferencia entre el ángulo de elevación deseado y el ángulo de elevación actual. Las variables lingüísticas para la entrada e y de son: NEG, Z, POS.

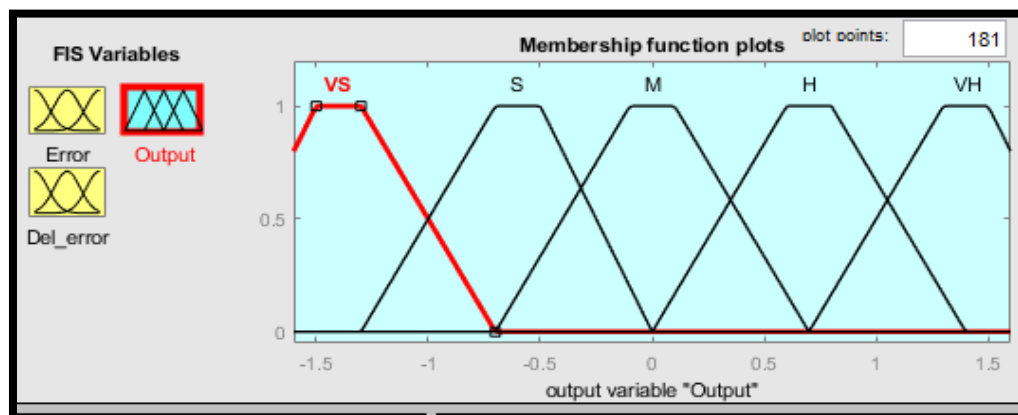
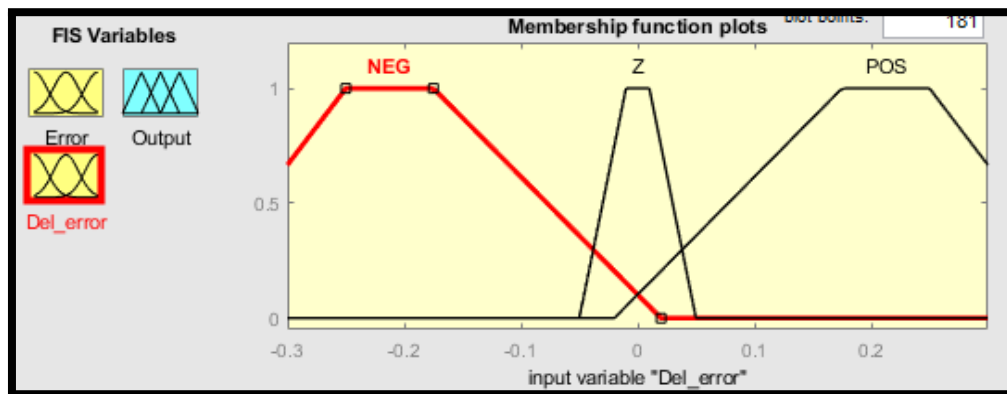
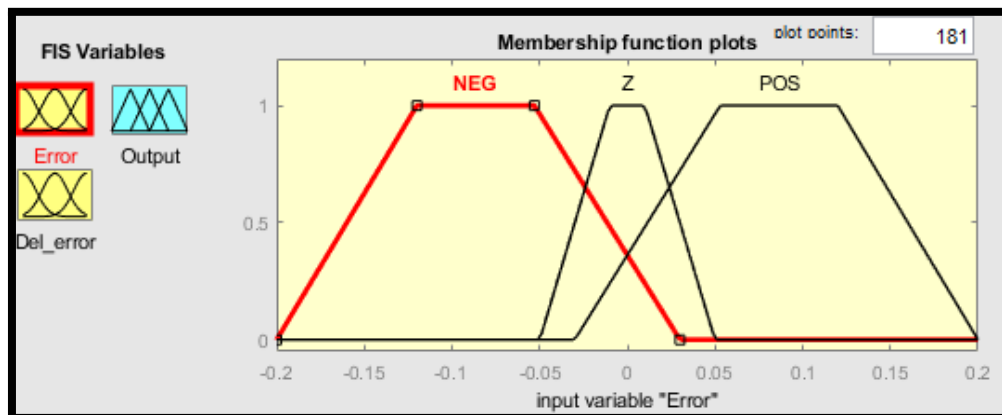
Cambio en el Error (de): La tasa de cambio del error de elevación. Las variables lingüísticas para la salida son: VS, S, M, H, VH.

La defuzzificación funciona en base a nueve reglas difusas que relacionan las entradas (e y de) con la salida. Estas reglas se implementan en el editor FIS de Matlab.

La defuzzificación es el proceso final para convertir el resultado del motor de inferencia (una variable lingüística) en un valor numérico que el sistema de control puede utilizar.

El sistema utiliza un motor de inferencia de tipo Mamdani y el método de defuzzificación utilizado es el Centroide.

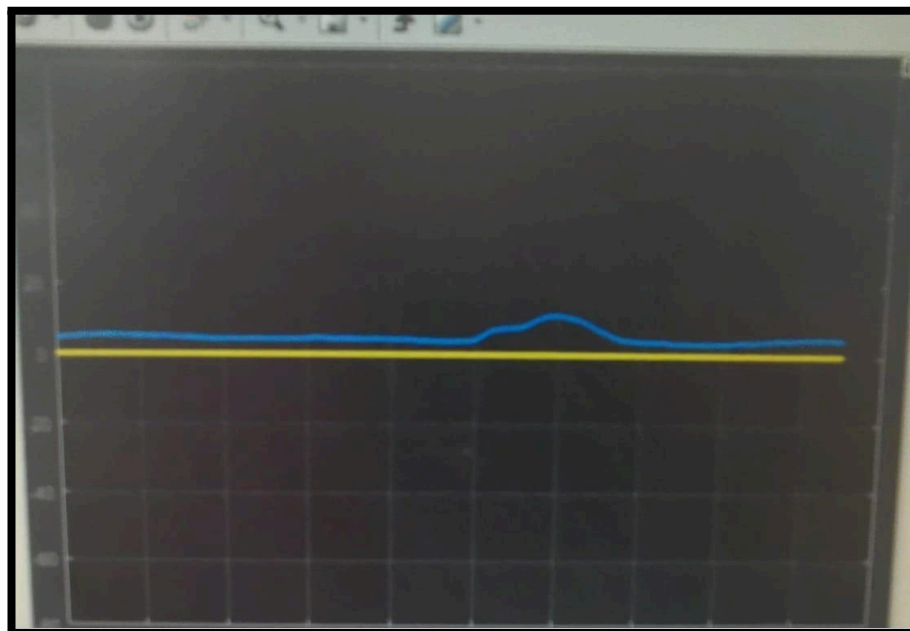
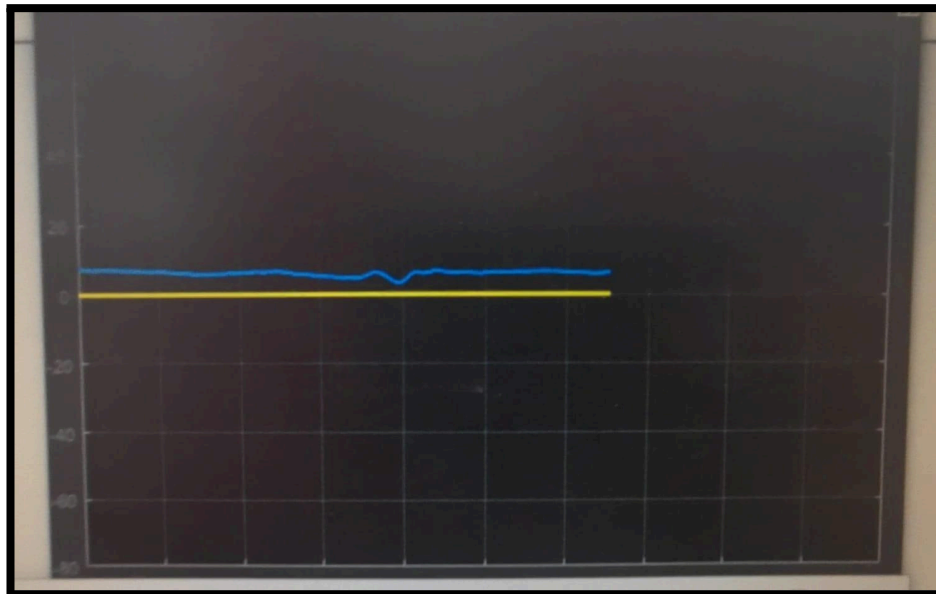
- **Realizar la modificación en el modelo del helicóptero real considerando el modelo utilizado en el artículo. Guarde únicamente el archivo .fis**



Fuzzy Rule Base		
1. If (Error is NEG) and (Del_error is NEG) then (Output is VS) (1)		
2. If (Error is NEG) and (Del_error is Z) then (Output is S) (1)		
3. If (Error is NEG) and (Del_error is POS) then (Output is M) (1)		
4. If (Error is Z) and (Del_error is NEG) then (Output is S) (1)		
5. If (Error is Z) and (Del_error is Z) then (Output is M) (1)		
6. If (Error is Z) and (Del_error is POS) then (Output is H) (1)		
7. If (Error is POS) and (Del_error is NEG) then (Output is M) (1)		
8. If (Error is POS) and (Del_error is Z) then (Output is H) (1)		
9. If (Error is POS) and (Del_error is POS) then (Output is VH) (1)		

If	and	Then
Error is	Del_error is	Output is
NEG	NEG	VS
Z	Z	S
POS	POS	M
none	none	H
		VH
		none

- Anote el OS, ST y el RT de la implementación real



Crane – LQR Agresivo

- Identificar sistema y controlador base para el Trolley, Cart o JIB
 - Localizar matrices A, B, matrices de pesado Q y R originales y la ganancia K.

A:

```
A_J =  
  
      0      0      1.0000      0      0  
      0      0      0      1.0000      0  
      0     -1.7019      0      0      0  
      0    -13.3301      0      0      0  
  1.0000      0      0      0      0
```

B:

```
B_J =  
  
      0  
      0  
  18.2478  
  21.1299  
      0
```

Q y R:

```
Q_J =  
  
      5      0      0      0      0  
      0      5      0      0      0  
      0      0      1      0      0  
      0      0      0      5      0  
      0      0      0      0      1  
  
>> R_J  
  
R_J =  
  
      0.0500
```

- Anotar la matriz de Ganancias original:

```
K_J =  
  
  14.0715   34.7159   10.9577   1.4618   4.4721
```

- Ajuste para comportamiento agresivo

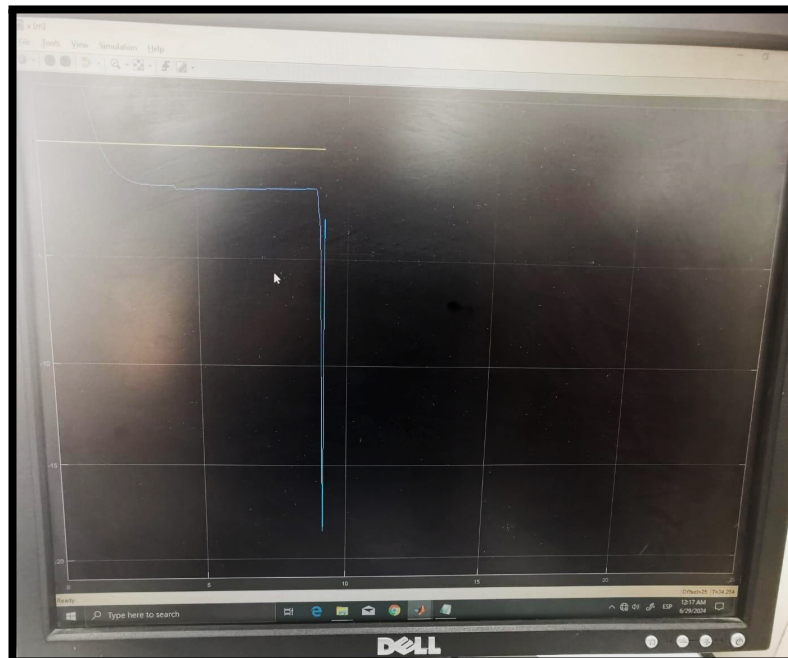
- Reducir R (penaliza menos el esfuerzo → control más rápido)
- Aumentar pesos en Q para la posición del trolley y velocidad

```
86 Q_J = diag([100 50 1 5 1]);
87 R_J = 0.01;
```

- Simule el sistema con las nuevas ganancias



- Adjunte capturas de pantalla del sistema controlado con LQR modificado.
 - Anote el OS, ST y el RT.



Criterio	Peso	Indicador	Nota
Implementación funcional	30 %	Simulación ejecuta y responde correctamente	
Análisis experimental	30 %	Tablas, tiempos y observaciones	
Comprensión teórica	20 %	Respuestas coherentes a preguntas	
Presentación	20 %	Claridad, organización y conclusiones	

Nombre y firma:

Butron Garay Pablo Sandro	
Colque Ruiz Paolo Andrés	