UNIVERSIDAD CATÓLICA BOLIVIANA “SAN PABLO” 08/11/2025 TÉCNICAS DE CONTROL – DRM-503

**Clase 6 – Control Óptimo y difuso para distintos plantas**

Para este laboratorio se hará uso del Helicopter y del Crane

Archivos para el Helicoptero:

https://quanserinc.box.com/shared/static/zztk0rpahb06fg6ybj32jafjno9m4zlm.zip Archivos para el Crane:

https://quanserinc.box.com/shared/static/jta1y17m8wp2i073dn4pcllj20nw8m4y.zip Archivo Fuzzy para el Helicopter:

https://drive.google.com/drive/folders/19oh06eoreAngOFuQE52uBLzyWPP\_YfHl?usp=sharing

**Objetivos del Laboratorio:**

• Comprender el controlador actual utilizado por cada planta no lineal (Helicoptero y Crane). • Modificar la ganancia LQR del Helicoptero 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:**

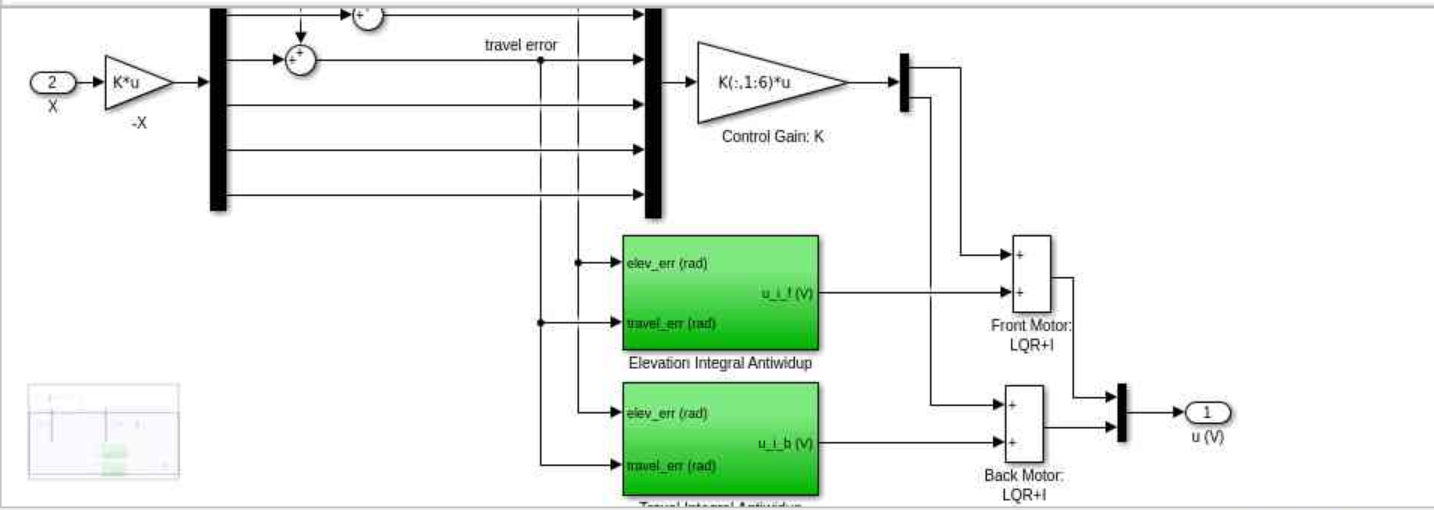
**Helicóptero – LQR**

**Complete las siguientes instrucciones para el Helicóptero**

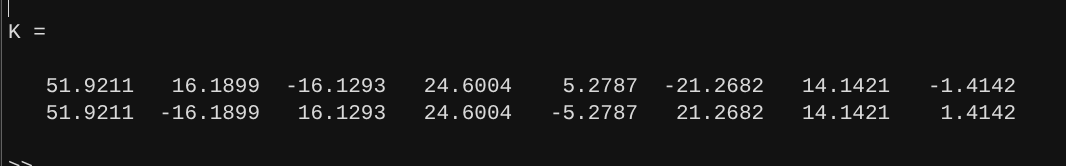
• Identificar el controlador original

o Abrir el archivo del modelo del helicóptero.

o Localizar el bloque donde aparece la ley de control típica:



o Anotar la matriz de Ganancias original:



• Modificación del LQR (Reducción de overshoot)

o Abrir el archivo donde se definen **Q** y **R** del LQR.

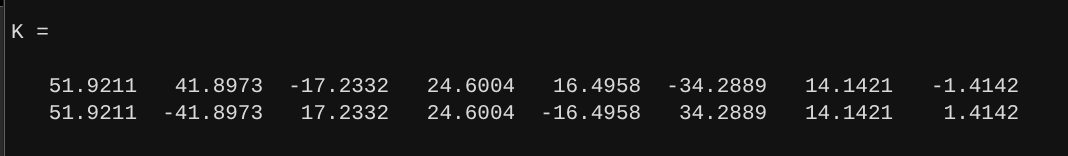
o Aumentar selectivamente los pesos asociados a:

▪ error del ángulo pitch

▪ velocidad angular pitch

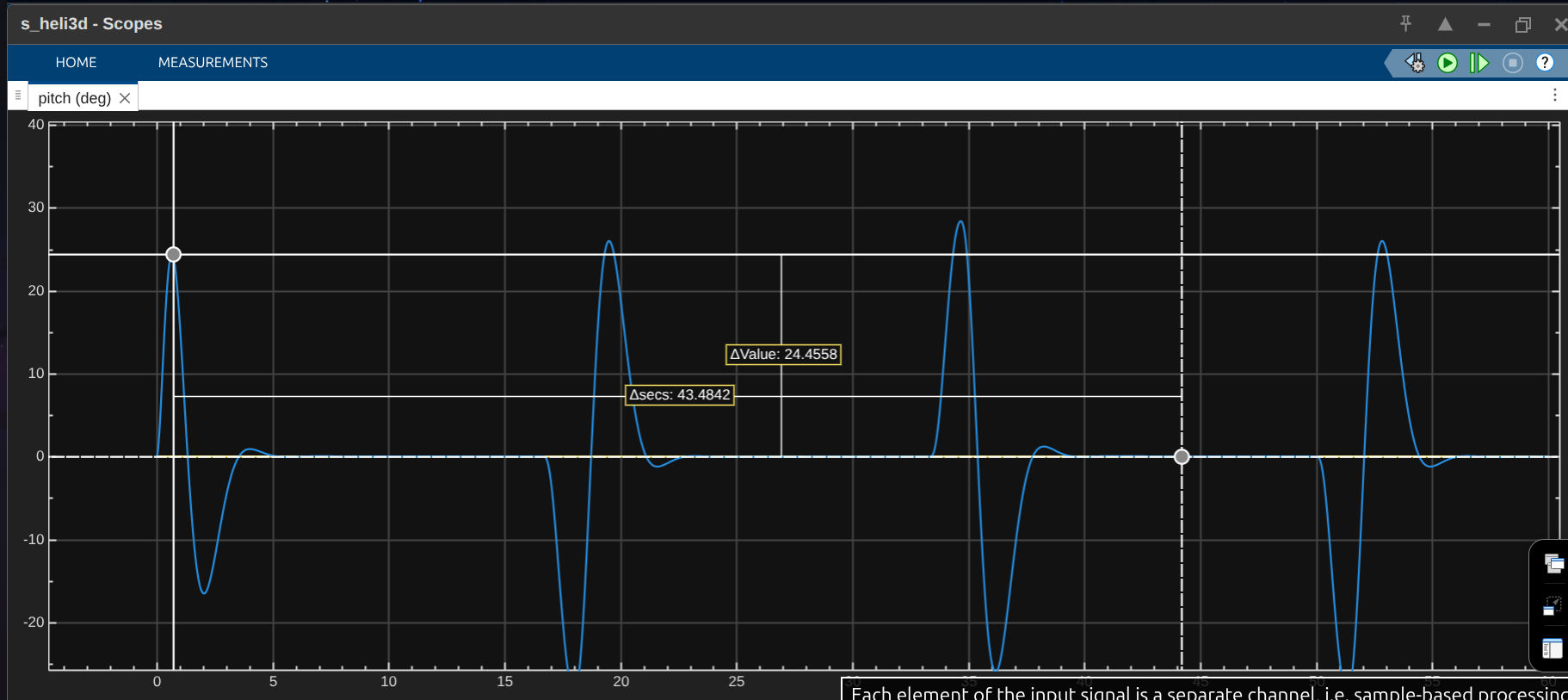
o Generar la nueva ganancia:

K\_new = lqr(A, B, Q, R);

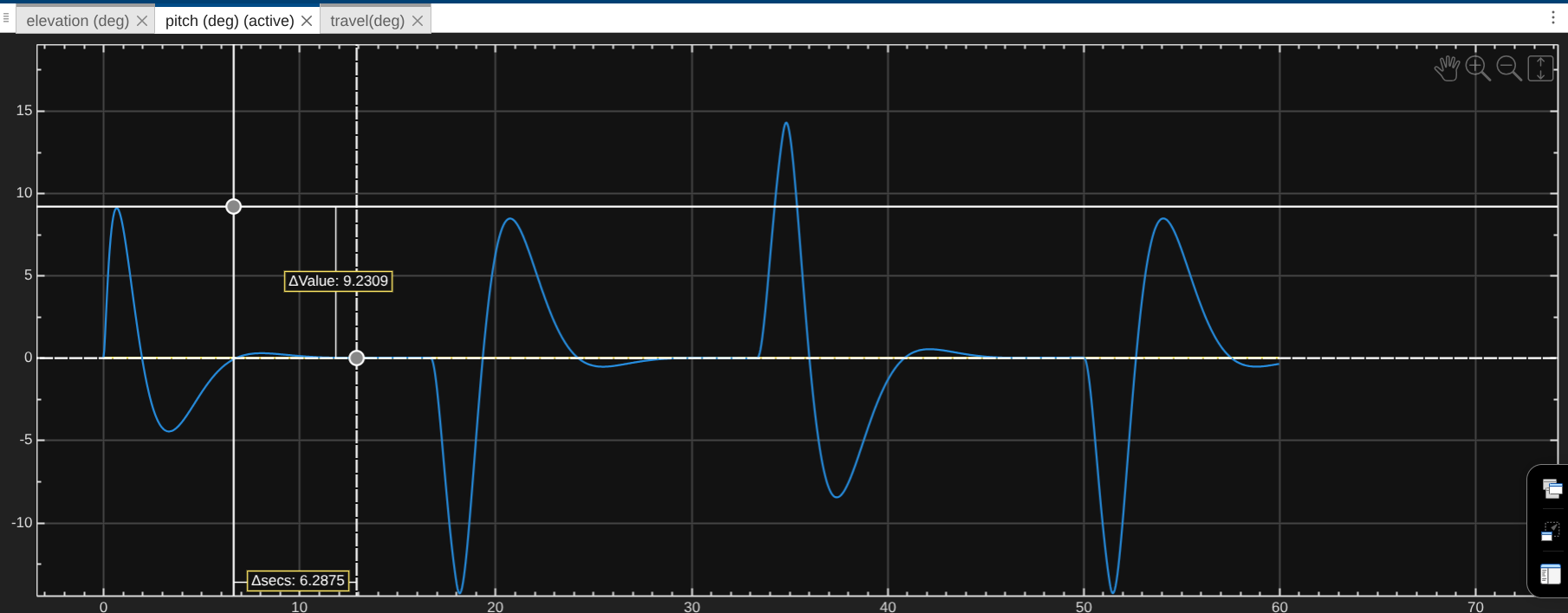


o Reemplazar la ganancia en el modelo Simulink o Python como usted desee. o Ejecutar la simulación (añadir capturas de pantallas) y anotar:

▪ Overshoot original: 24.4558



▪ Overshoot modificado: 9.2309 (A disminuido un 37.754% con respecto al original)



▪ Tiempo de establecimiento

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

**Helicóptero – Control Difuso**

• Revisar el documento Fuzzy\_heli.pdf

o Comprender las variables de entrada e y de, la base de reglas y la función de

defuzzificación utilizada.

**VARIABLES**

*Error (e):* Diferencia entre el ángulo deseado y el actual.

*Cambio del Error (de):* Derivada del error.

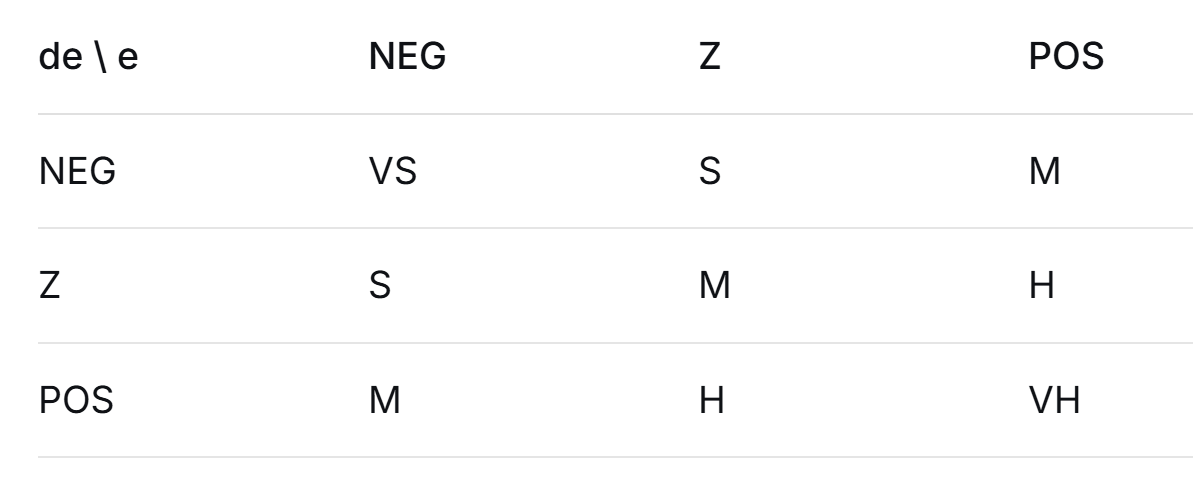
*Respuesta del sistema:* Señal de control que ajusta los motores.

**CONJUNTOS DIFUSOS**

**De entrada:** {NEG, Z, POS} / (negativo, cero, positivo)

**De salida:** {VS, S, M, H, VH} / (Muy lento, lento, medio, alto, muy alto)

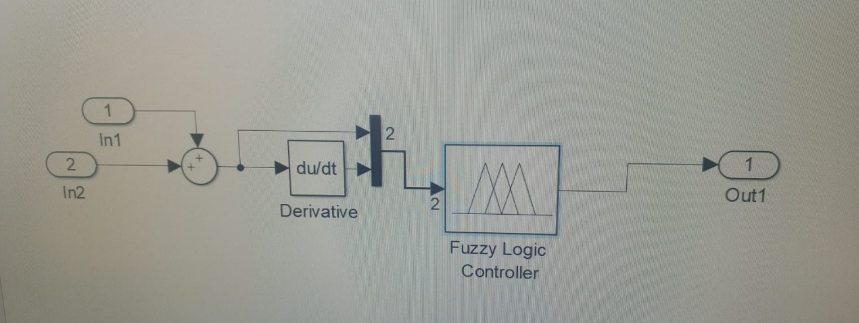
**REGLAS**

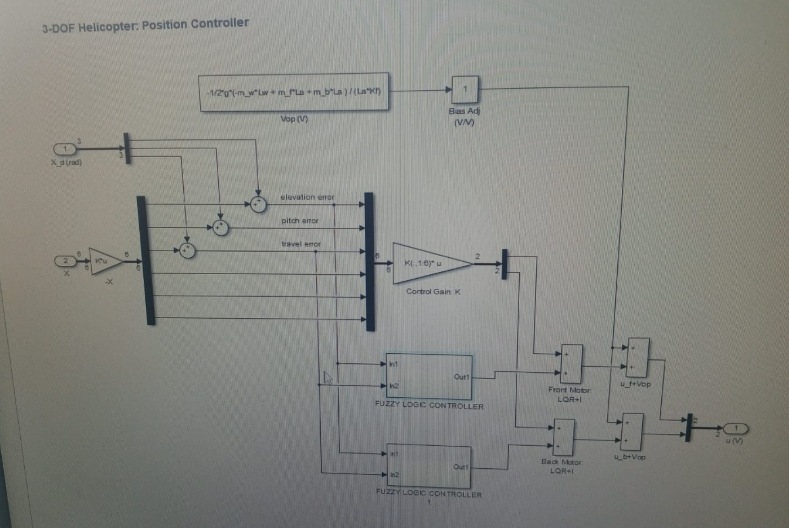
****

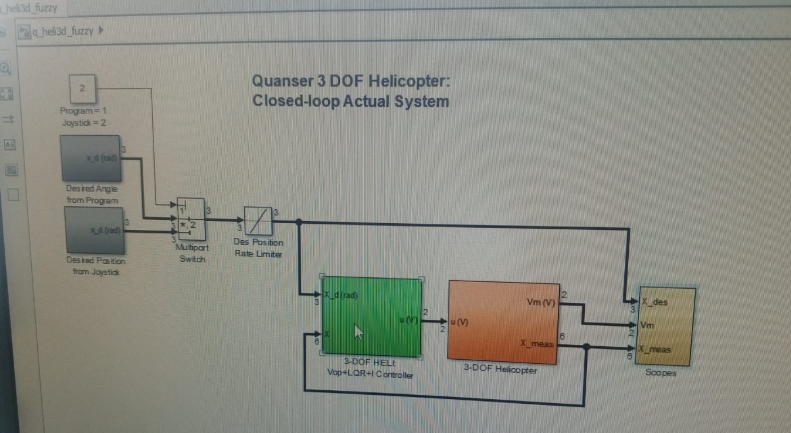
Se usa el método centroide

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

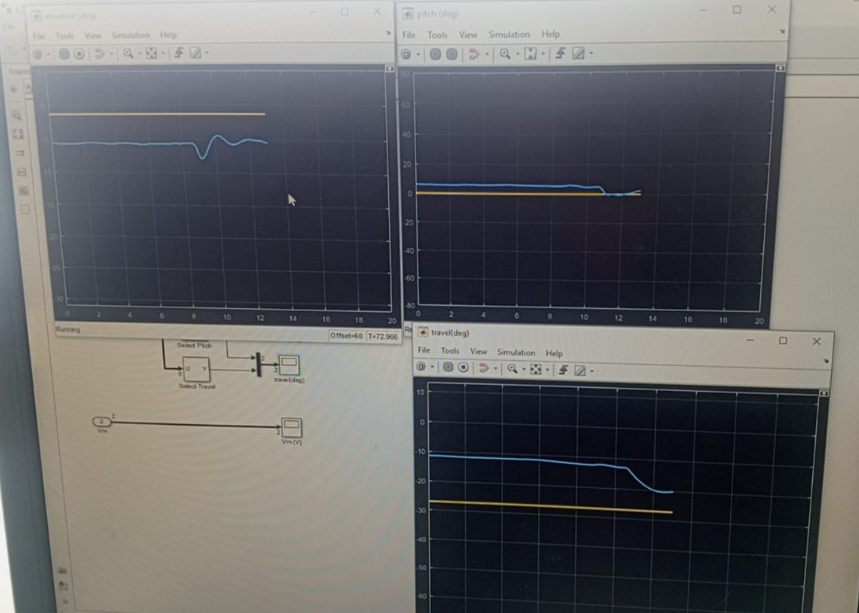
▪ Anote el OS, ST y el RT de la implementación real.

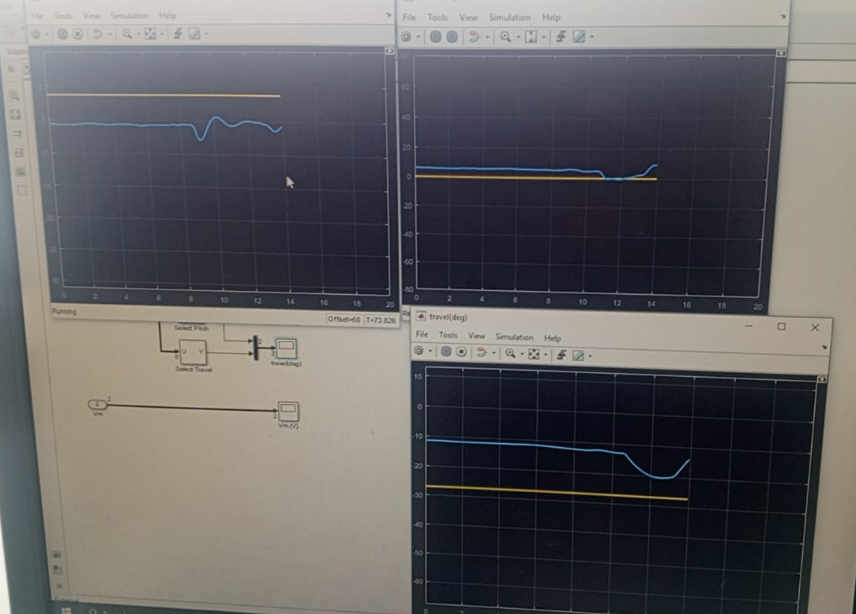
****

****

****

****

****

****

****

**codigo:**

**%% ==========================**

**% Fuzzy Helicopter Controller (.fis)**

**% Compatible con MATLAB R2019b en adelante**

**% ==========================**

**% Crear nuevo sistema Mamdani**

**fis = mamfis('Name','Fuzzy\_Helicopter','AndMethod','min','DefuzzificationMethod','centroid');**

**%% ==== ENTRADA 1: ERROR (e) ====**

**fis = addInput(fis,[-0.2 0.2],'Name','Error');**

**fis = addMF(fis,'Error','trimf',[-0.2 -0.2 0],'Name','NEG');**

**fis = addMF(fis,'Error','trimf',[-0.1 0 0.1],'Name','Z');**

**fis = addMF(fis,'Error','trimf',[0 0.2 0.2],'Name','POS');**

**%% ==== ENTRADA 2: CAMBIO DE ERROR (de) ====**

**fis = addInput(fis,[-0.3 0.3],'Name','Change\_in\_Error');**

**fis = addMF(fis,'Change\_in\_Error','trimf',[-0.3 -0.3 0],'Name','NEG');**

**fis = addMF(fis,'Change\_in\_Error','trimf',[-0.15 0 0.15],'Name','Z');**

**fis = addMF(fis,'Change\_in\_Error','trimf',[0 0.3 0.3],'Name','POS');**

**%% ==== SALIDA: RESPUESTA (w) ====**

**fis = addOutput(fis,[-1.6 1.6],'Name','Response');**

**fis = addMF(fis,'Response','trimf',[-1.6 -1.6 -0.8],'Name','VS');**

**fis = addMF(fis,'Response','trimf',[-1.2 -0.8 0],'Name','S');**

**fis = addMF(fis,'Response','trimf',[-0.4 0 0.4],'Name','M');**

**fis = addMF(fis,'Response','trimf',[0 0.8 1.2],'Name','H');**

**fis = addMF(fis,'Response','trimf',[0.8 1.6 1.6],'Name','VH');**

**%% ==== BASE DE REGLAS (del paper) ====**

**rules = [...**

**1 1 1 1 1; % If e=NEG & de=NEG => w=VS**

**1 2 2 1 1; % If e=NEG & de=Z => w=S**

**1 3 3 1 1; % If e=NEG & de=POS => w=M**

**2 1 2 1 1; % If e=Z & de=NEG => w=S**

**2 2 3 1 1; % If e=Z & de=Z => w=M**

**2 3 4 1 1; % If e=Z & de=POS => w=H**

**3 1 3 1 1; % If e=POS & de=NEG => w=M**

**3 2 4 1 1; % If e=POS & de=Z => w=H**

**3 3 5 1 1]; % If e=POS & de=POS => w=VH**

**fis = addRule(fis,rules);**

**%% ==== GUARDAR EL ARCHIVO ====**

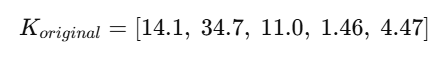
**writeFIS(fis,'Fuzzy\_Helicopter');**

**Crane – LQR Agresivo**

• Identificar sistema y controlador base para el Trolley, Cart o JIB

o Localizar matrices A, B, matrices de pesado Q y R originales y la ganancia K.

o Anotar la matriz de Ganancias original: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



• Ajuste para comportamiento agresivo

o Reducir R (penaliza menos el esfuerzo → control más rápido)

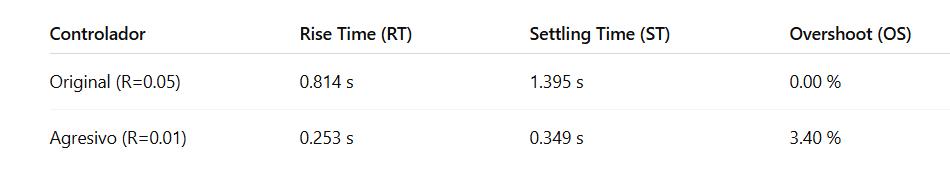
o Aumentar pesos en Q para la posición del trolley y velocidad



• Simule el sistema con las nuevas ganancias

o Adjunte capturas de pantalla del sistema controlado con LQR modificado.

▪ Anote el OS, ST y el RT.



**Rúbrica de Calificación**

| **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:**

|  |  |
| --- | --- |
|  |  |
| **Sahori Claribel Tapia Pari** |  |
|  |  |
|  |  |