**Taller 1 (punto 4)**

**Presentado por:**

* Camila Lozano Jiménez - código 20201020161.
* Juan Felipe Rodríguez Galindo - código 20181020158.

**Enunciado:**

**IV. SISTEMAS DE CONTROL BASADOS EN RELACIONES BOOLEANAS**

**APLICACIÓN II**

Para el diseño del punto II proponer conjuntos difusos partiendo de los conjuntos booleanos y determinar:

* Diseño de los conjuntos difusos.
* Simulación en MATLAB.

***Requerimientos de diseño:***

* Entrada de referencia escalón unitario µ(t).
* Sobre un pico inferior al 10%.
* Error (oscilación) en estado estable inferior al ±5%.

***Configuraciones:***

* k: 2.
* b1: 2.
* b0: 2.

**Desarrollo:**

**Características del sistema:**

* Ung = -1.0
* Unp = -0.5
* Uz = 0
* Upp = 0.5
* Upg = 1.0

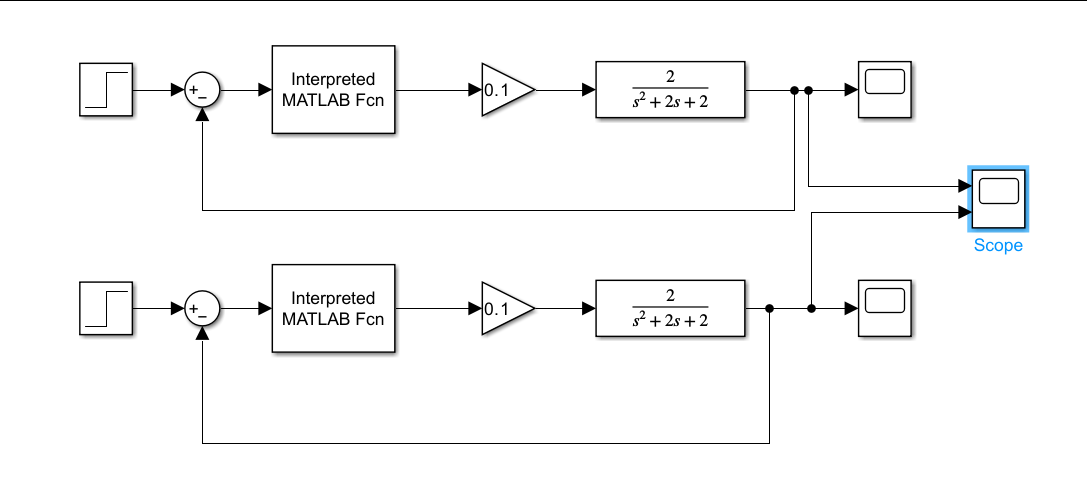
**Función de transferencia:**

**Análisis:**

* Ung, Unp, Uz, Upp, Upg son ajustables
* La salida es la acción suministrada a la planta

**Configuración Matlab:**

Para el presente laboratorio se tiene la configuración (K,B0,B1), la cual representa k=2, b0=2 y k1= 2 respectivamente.



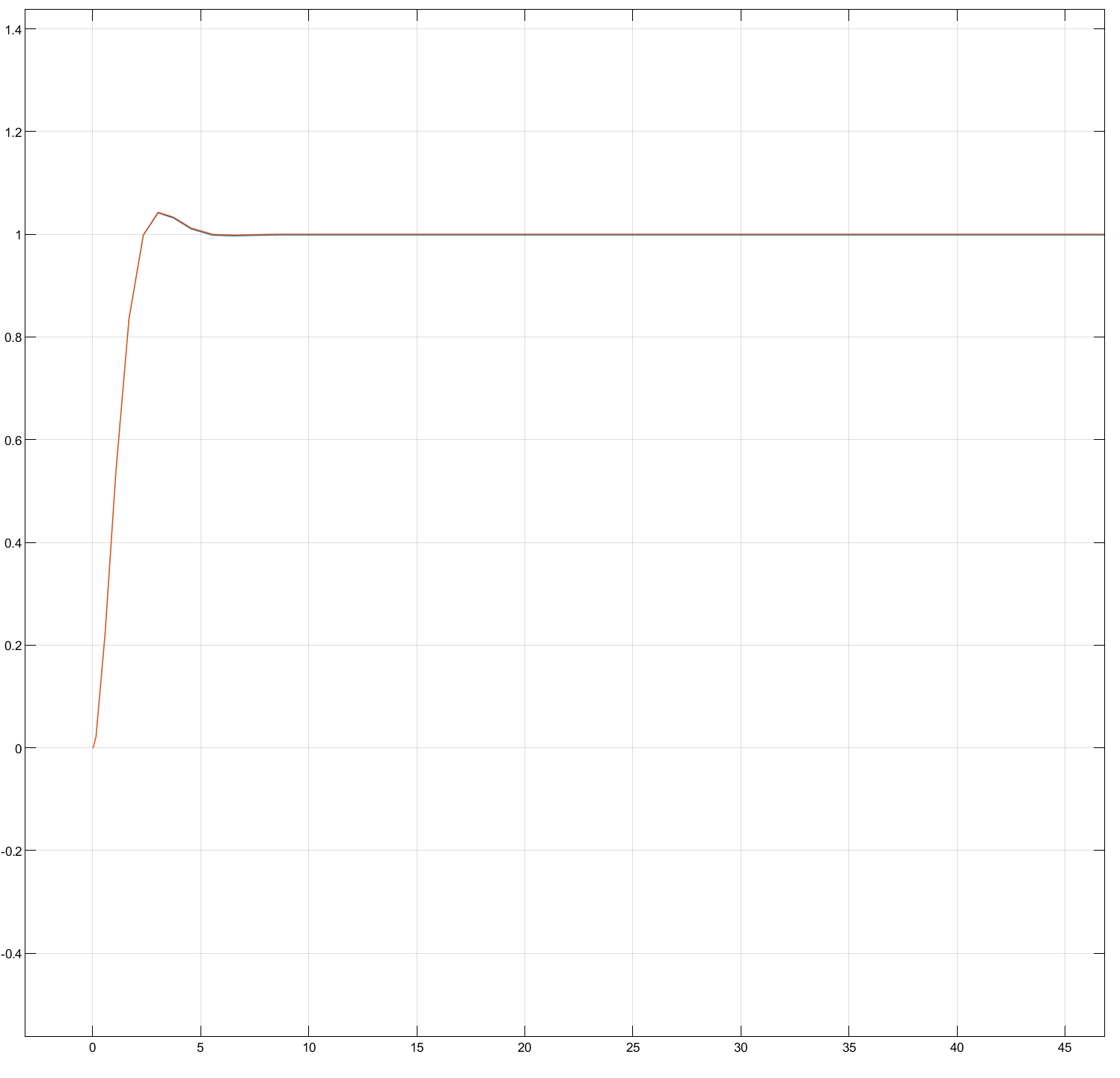
*Imagen 2, Simulink,cambio de parametros.*

Configuramos los diferentes scopes, para realizar el control de la similitud de las dos configuraciones creadas

**Requerimientos del diseño:**

* Entrada de referencia escalón unitario μ(t)
* Sobrepico inferior al 20%.
* Oscilación en estado estable inferior al ±10%.

**Cumplimiento de los requerimientos:**

****

*Imagen 3, Scope Simulink, cumplimiento requerimientos.*

**Código:**

**Controlador Matlab:**

function f = ControladorD(e)

% Función que implementa el controlador Booleano

% Taller 1 - Punto 4

% Controlador booleano con retroalimentacion del error

% Definición de e para la graficación

%e = (0:0.1:7)';

% Error positivo A (epa)

erPosA = smf(e, [-500 0.3]);

% Error positivo B (epb)

erPosB = dsigmf(e, [3 -2 4 36.6]);

% Error negativo B (enb)

erNegB = smf(e, [-500 0.3]);

% Error negativo A (ena)

erNegA = dsigmf(e, [0.309 1 2.19 36.6]);

% Llamada a la función de graficación

%graficarFuncionesPertenencia(e, erPosA, erPosB, erNegB, erNegA);

% Ecuaciones

Y1=erNegA;

Y2=erNegB;

Y3=0;

Y4=erPosB;

Y5=erPosA;

% Actuadores

Ung = 0;

Unp = 5;

Uz = 0;

Upp = 5;

Upg = 0;

% Salida

f = Ung\*Y1 + Unp\*Y2 + Uz\*Y3 + Upp\*Y4 + Upg\*Y5;

end

**Graficador:**

function graficarFuncionesPertenencia(e, erPosA, erPosB, erNegB, erNegA)

plot(e, erPosA, 'r'); % Añadido color para diferenciar

hold on;

plot(e, erPosB, 'g'); % Añadido color para diferenciar

hold on;

plot(e, erNegB, 'b'); % Añadido color para diferenciar

hold on;

plot(e, erNegA, 'm'); % Añadido color para diferenciar

hold on;

xlabel('Error');

ylabel('Grado de pertenencia');

title('Funciones de pertenencia del error');

legend('erPosA', 'erPosB', 'erNegB', 'erNegA');

hold off;

end