

### TAREA 2.3: Retroalimentación de Error

Considere el siguiente sistema dinámico:

$$\dot{X} = \begin{bmatrix} -1 & 0 & -4 & 3 \\ 3 & -1 & 2 & -1 \\ 0 & -2 & 0 & -1 \\ 2 & 0 & 1 & -2 \end{bmatrix} X + \begin{bmatrix} 0 & 0 \\ 3 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} u$$
$$Y = [1 \ 0 \ 0 \ 0]X$$

Diseñe un controlador que haga la salida del sistema siga la referencia

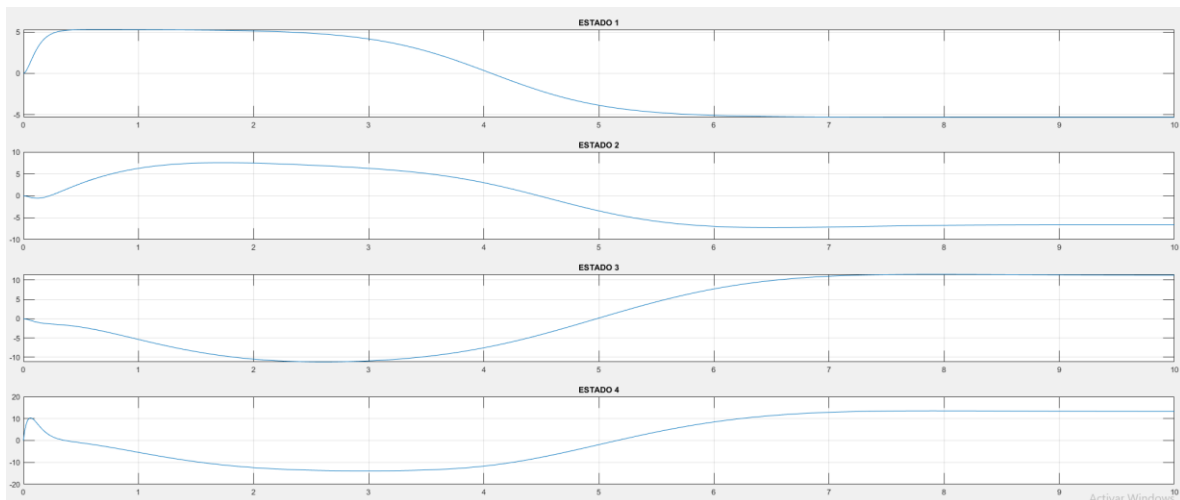
$$Y_{ref} = 5 \tanh(4 - t)$$

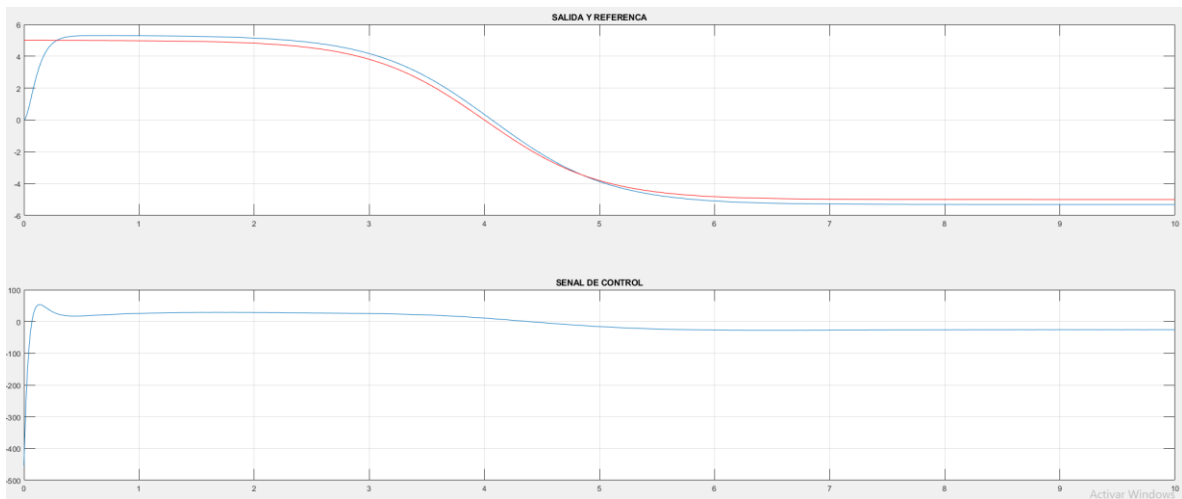
con un tiempo de estabilización no mayor a 1/2 segundo.

Para este problema, dado que tenemos dos entradas y una sola salida, se puede suponer que la matriz resultante no será una matriz cuadrada. Por lo que el siguiente paso es corroborar la controlabilidad de  $M_c = C^*B$ ; podemos ver que obtenemos que esta no es controlable. Por lo que a paso consiguiente iremos a un sistema de control por bloques en el que omitiremos una entrada (como se llegaron a hacer en los repases del examen) . De misma manera retomando la explicación que dio el maestro en clase, tomamos un camino con solo dos bloques, en el que hacemos que:

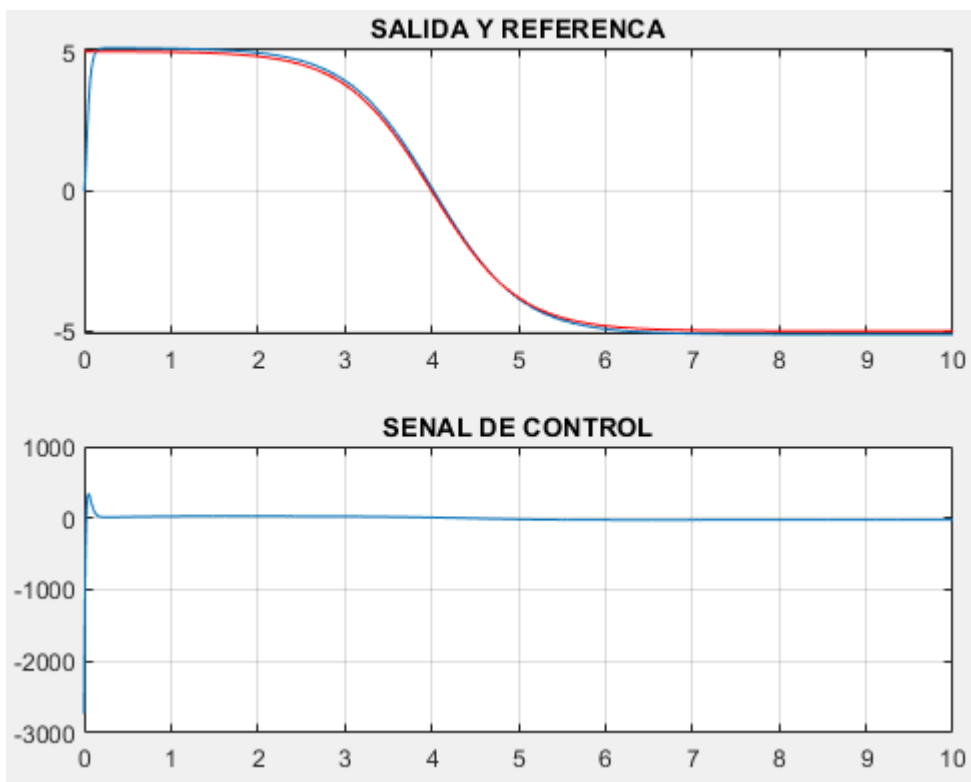
$$u_2 \rightarrow x_4 \rightarrow x_1$$

Dicho desarrollo lo hacemos en Matlab como se mostrara en el código (realmente esto fue por terquedad). Sin embargo primero mostraremos los resultados obtenidos.





Sin embargo los resultados no son como los queremos, así que al momento le ingresamos un polo mas agresivo al sistema.



Obteniendo un mejor resultado a costo de la señal de control.

## CODIGO:

```
%% T 2.3
close all; clc; clear;

%t < 0.5 s
% t = 0.25s
k = -4/0.1;

tspan = [0,10];
x0 = [0;0;0;0];
Pd = [k,k];

controlador_bloques_plot(tspan, x0, Pd)

%% Functions
function controlador_bloques_plot(tspan, x0, Pd)

global A B C k1 k2

A = [-1,0,-4,3;
      3,-1,2,-1;
      0,-2,0,-1;
      2,0,1,-2];
B = [0,0;
      3,0;
      0,0;
      0,-1];
C = [1,0,0,0];

k1 = Pd(1);
k2 = Pd(2);

[t,X] = ode45(@controlador_bloques_sys,tspan,x0);

%y_ref
ref = 5*tanh(4-t);
%dy_ref
dref = -5*sech(4-t).^2;
%ddy_ref
ddref = -10*tanh(4-t).*sech(4-t).^2;

%Bloques
%Op1 2 bloques
% A x1 lo controlamos con x4
% A x4 lo controlamos con U
dx1_op1 = -X(:,1) + 0*X(:,2) - 4*X(:,3) + 3*X(:,4);
dx2_op1 = 3*X(:,1) - 1*X(:,2) + 2*X(:,3) - 1*X(:,4);
dx3_op1 = 0*X(:,1) - 2*X(:,2) + 0*X(:,3) - 1*X(:,4);
%dx4_op1 = 2*X(:,1) + 0*X(:,2) + 1*X(:,3) - 2*X(:,4) - U;
%Op2 2 bloques
% A x1 lo controlamos con x2
% A x2 lo controlamos con U
% dx1_op2 = -X(:,1) + 0*X(:,2) - 4*X(:,3) + 3*X(:,4);
% dx2_op2 = 3*X(:,1) - 1*X(:,2) + 2*X(:,3) - 1*X(:,4) + 3*U;
% dx3_op2 = 0*X(:,1) - 2*X(:,2) + 0*X(:,3) - 1*X(:,4);
% dx4_op2 = 2*X(:,1) + 0*X(:,2) + 1*X(:,3) - 2*X(:,4);

%Señales de error
e1 = ref - X(:,1);
d_e1 = ref - dx1_op1;
x4_ref = (1/3)*(dref + X(:,1) + 4*X(:,3) - k1*e1);
dx4_ref = (1/3)*(ddref + dx1_op1 + 4*dx3_op1 - k1*d_e1);
```

```

e2 = x4_ref - X(:,4);

%Ley de Control
U = [zeros(numel(X(:,1)), 1), -dx4_ref + 2*X(:,1) + 0*X(:,2) + 1*X(:,3) -
2*X(:,4) + k2*e2];

figure;
subplot(4,1,1); plot(t, X(:,1)); title('ESTADO 1'); grid;
subplot(4,1,2); plot(t, X(:,2)); title('ESTADO 2'); grid;
subplot(4,1,3); plot(t, X(:,3)); title('ESTADO 3'); grid;
subplot(4,1,4); plot(t, X(:,4)); title('ESTADO 4'); grid;

figure;
subplot(2,1,1); plot(t, X(:,1), t, ref, 'red'); title('SALIDA Y REFERENCIA');
grid;
subplot(2,1,2); plot(t, U(:,2)); title('SEÑAL DE CONTROL');
grid;
end

function dX = controlador_bloques_sys(t,X)

global A B C k1 k2

%y_ref
ref = 5*tanh(4-t);
%dy_ref
dref = -5*sech(4-t).^2;
%ddy_ref
ddref = -10*tanh(4-t).*sech(4-t).^2;

%Bloques
    %Op1 2 bloques
    % A x1 lo controlamos con x4
    % A x4 lo controlamos con U
dx1_op1 = -X(1) + 0*X(2) - 4*X(3) + 3*X(4);
dx2_op1 = 3*X(1) - 1*X(2) + 2*X(3) - 1*X(4);
dx3_op1 = 0*X(1) - 2*X(2) + 0*X(3) - 1*X(4);
%dx4_op1 = 2*X(1) + 0*X(2) + 1*X(3) - 2*X(4) - U;

%Señales de error
e1 = ref - X(1);
d_e1 = ref - dx1_op1;
x4_ref = (1/3)*(dref + X(1) + 4*X(3) - k1*e1);
dx4_ref = (1/3)*(ddref + dx1_op1 + 4*dx3_op1 - k1*d_e1);

e2 = x4_ref - X(4);

%Ley de Control
U = [0; -dx4_ref + 2*X(1) + 0*X(2) + 1*X(3) - 2*X(4) + k2*e2];

dX = A*X + B*U;
end

```