Centro de Investigación y de Estudios
Avanzados del Instituto Politécnico Nacional
Unidad Guadalajara

Tarea 7. Control de formación para robots no holónomos

Presentado por

Jesús Alejandro Díaz Hernández

Presentado para el curso de

Tópicos avanzados de control 2

Curso impartido por: Héctor Manuel Becerra Fermín
Profesor

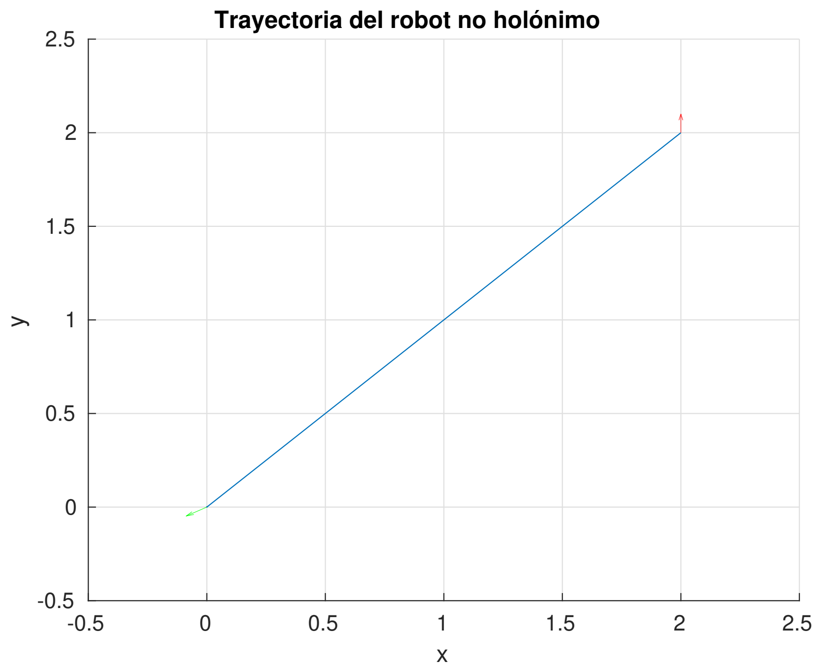
Guadalajara, Jalisco 15 julio 2024

Pregunta 1.-

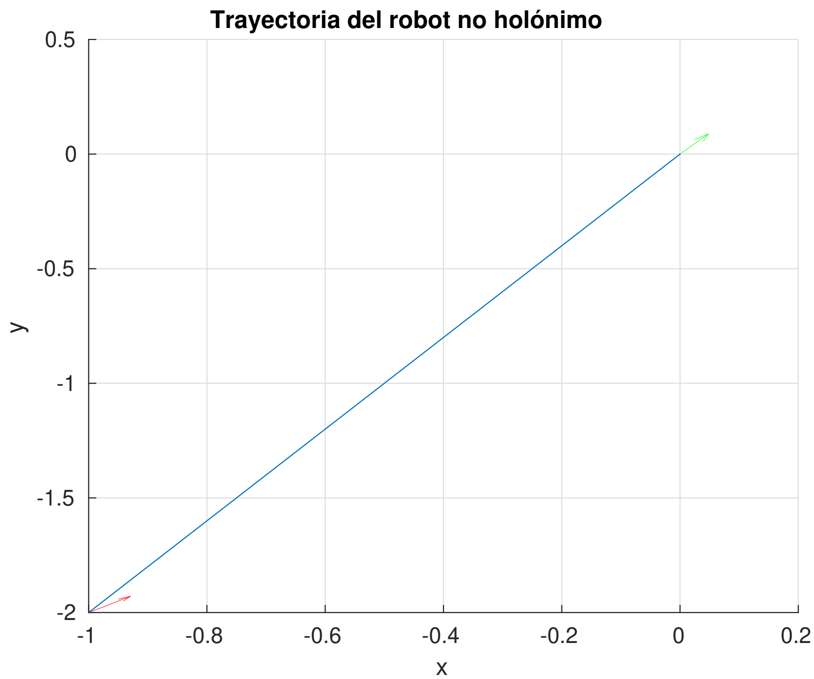
Considerando un solo robot de manejo diferencial y utilizando el planteamiento del punto de control. Se grafica la trayectoria del punto de control y no del vehículo como tal.

Inciso a)

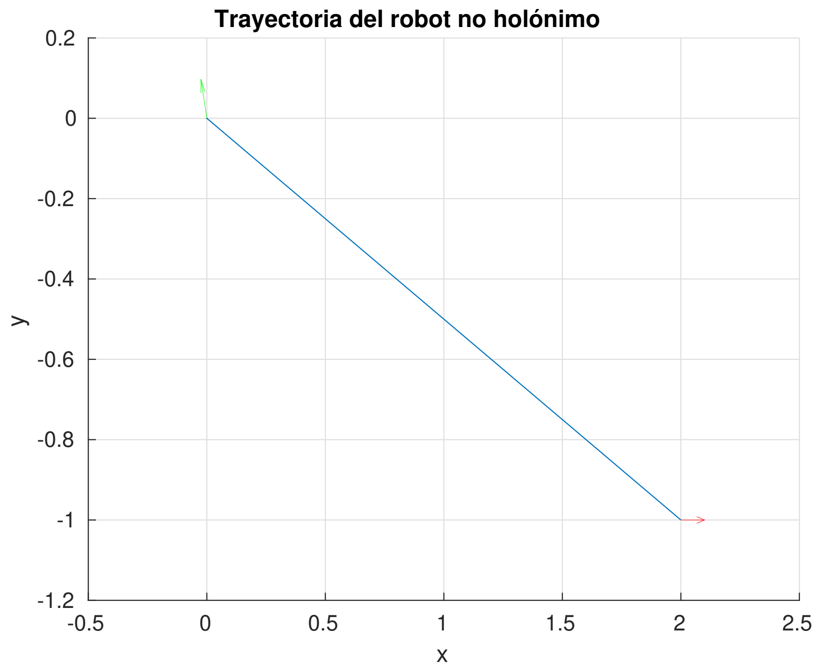
Probando con tres condiciones iniciales, la primera en $z = \begin{bmatrix} 2 & 2 \end{bmatrix}, \theta = 90$, obtenemos el siguiente resultado (la flecha roja indica el inicio de la trayectoria y su ángulo, la flecha verde el final, lo mismo para las siguientes condiciones iniciales)



Con condiciones iniciales $z = \begin{bmatrix} -1 & -2 \end{bmatrix}, \theta = 45$:



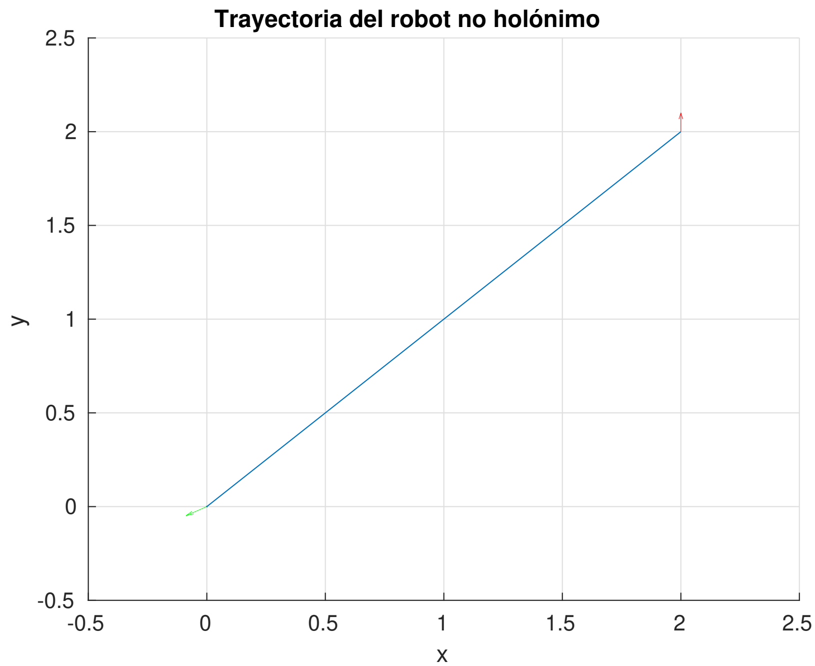
Con condiciones iniciales $z = \begin{bmatrix} -1 & 2 \end{bmatrix}, \theta = 0$:



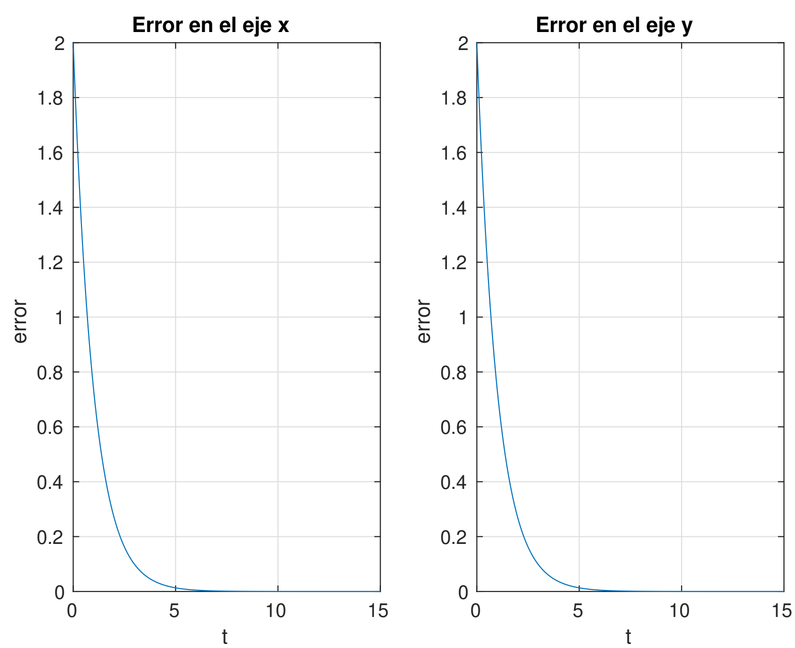
Nota: Las ganancias de control son iguales tanto en velocidad lineal como rotacional, podrían ser distintas, pero es más fácil programarlo así.

Inciso b)

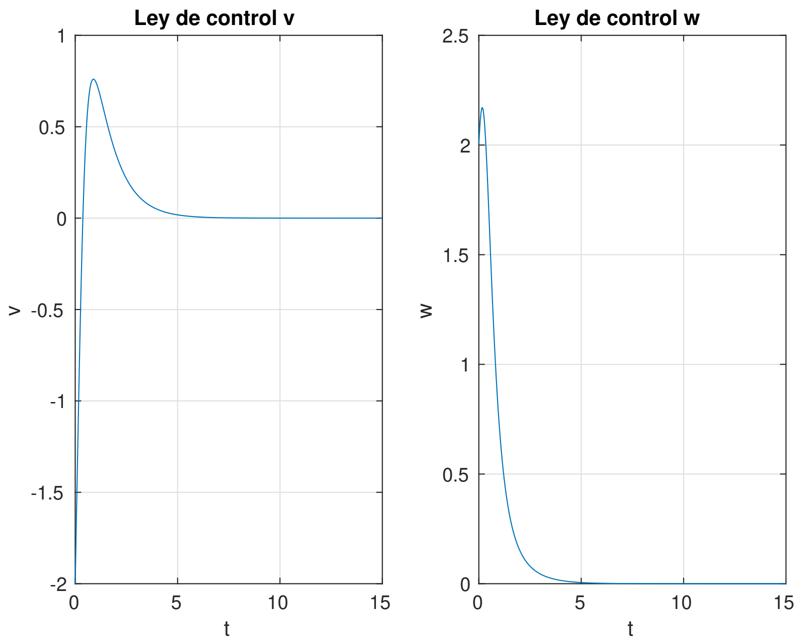
Utilizando las primeras condiciones iniciales, es decir:



La evolución del error en cada coordenada es la siguiente

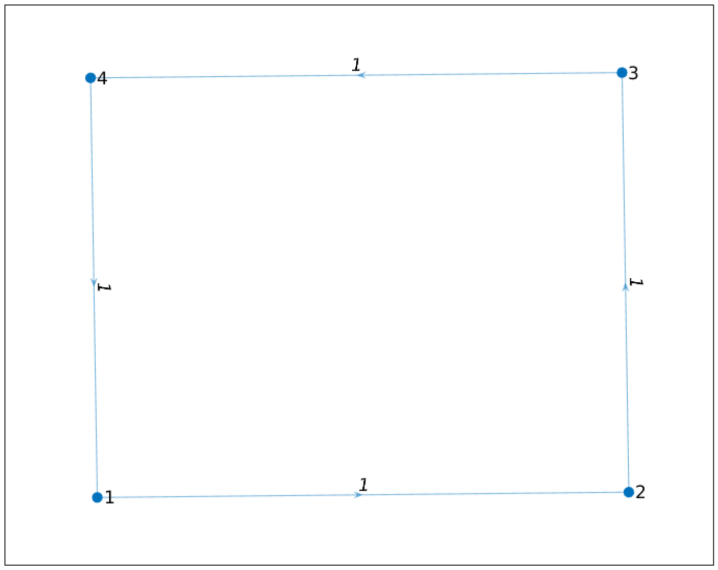


Y la evolución de las leyes de control tanto lineal como angular son:



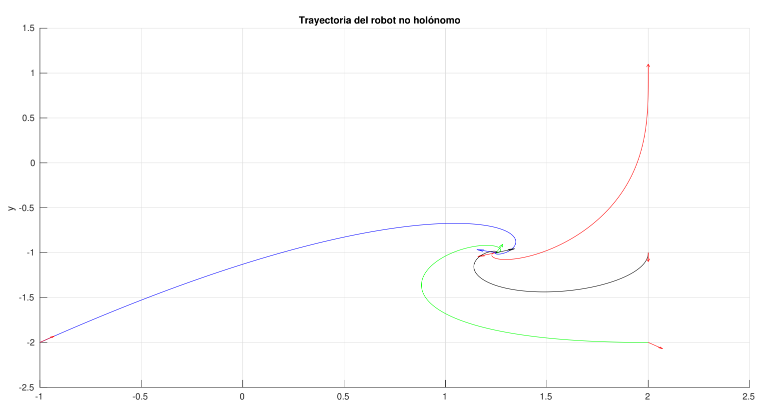
Pregunta 2.-

Extendiendo el control para cuatro robots y con la conectividad dirigida de un ciclo como en la siguiente imagen



Inciso a)

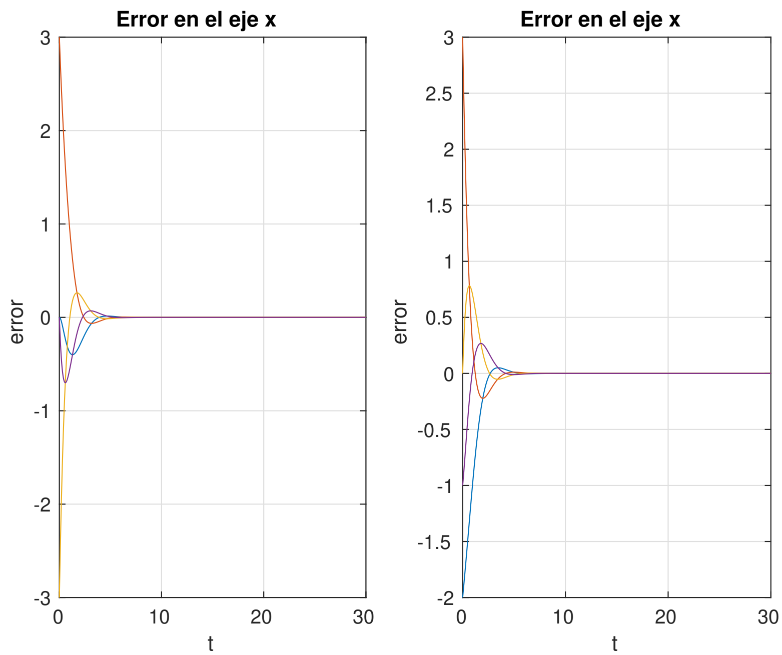
Obtenemos las siguientes trayectorias



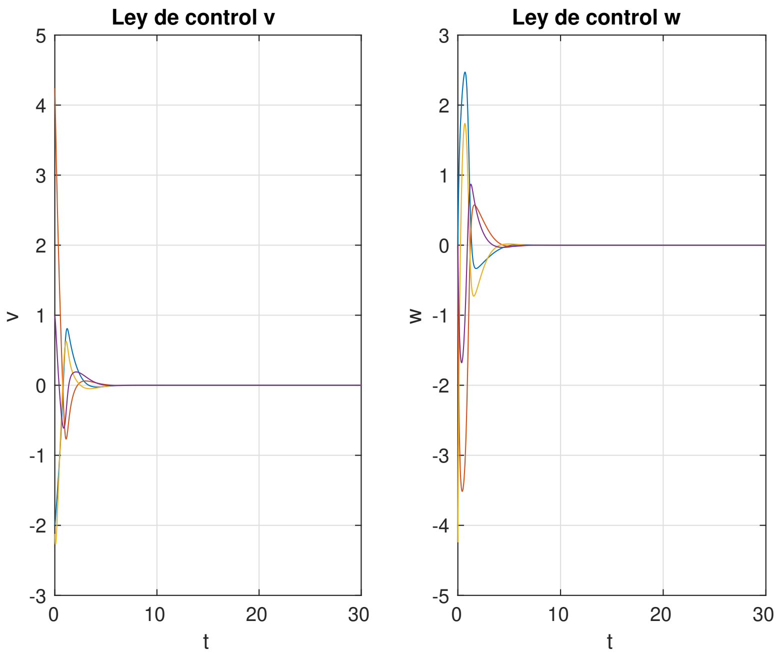
Las flechas rojas indican el inicio de las trayectorias, y para el final hay una flecha de cada color del robot.

Nota: Las ganancias de control son iguales tanto en velocidad lineal como rotacional para todos los robots, podrían ser distintas, pero es más fácil programarlo así.

La evolución de error de consenso para cada coordenada es



Y las entradas de control para velocidad lineal y rotacional son

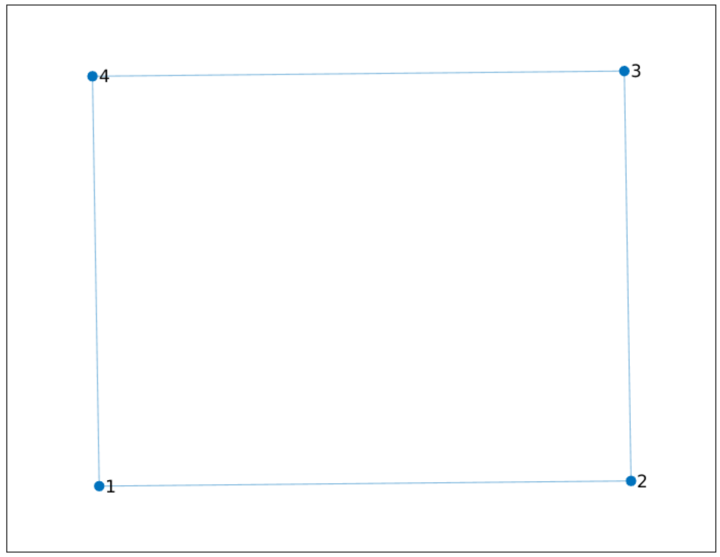


Inciso b)

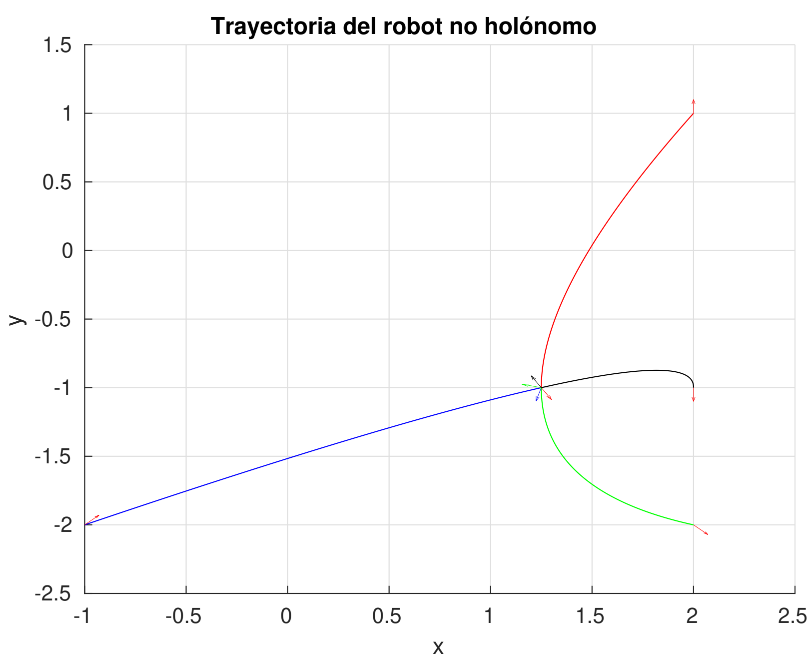
Si calculamos el vector nulo izquierdo normalizado de la matriz Laplaciana, el cual es $q = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix}$, y tomamos como referencia que los valores iniciales son $x_0 = \begin{bmatrix} 2 & -1 & 2 & 2 \end{bmatrix}^T$ y $y_0 = \begin{bmatrix} 1 & -2 & -2 & -1 \end{bmatrix}^T$. Los robots convergen a $x = 5/4$ y $y = 1$, lo que corresponde con los valores a los que llega la simulación.

Pregunta 3.-

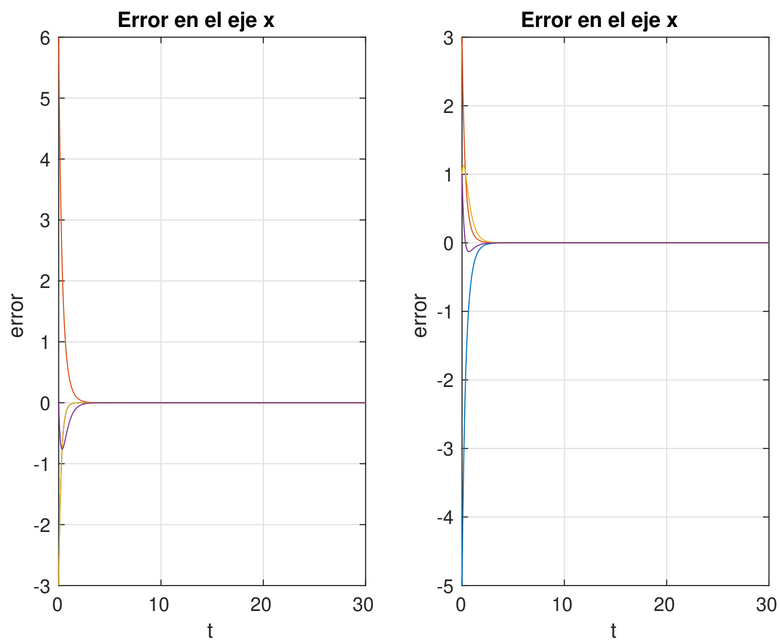
Ahora, considerando un grafo no dirigido, pero igualmente un ciclo, como el que se muestra a continuación:



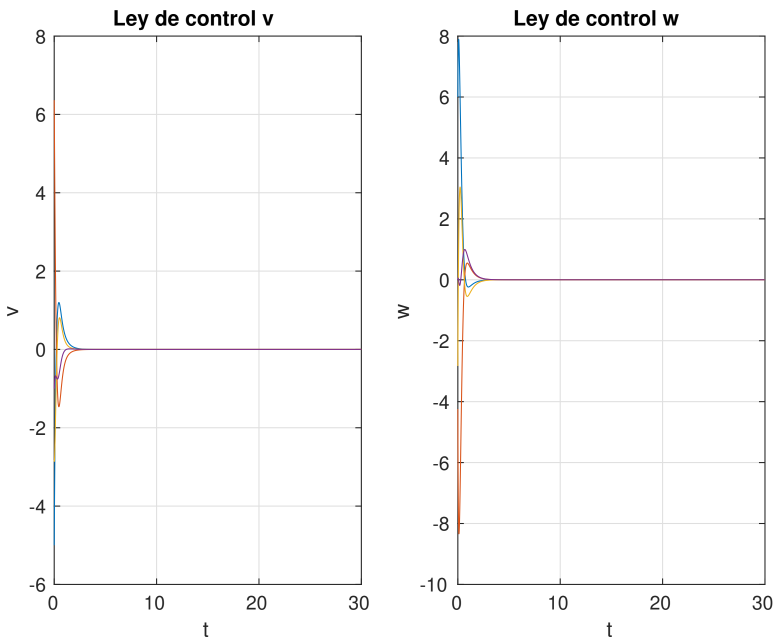
obtenemos la siguiente evolución de trayectorias



Los errores en cada coordenada



y las leyes de control como



En este caso los valores de consenso coinciden con los del grafo dirigido, es decir, $x = 5/4$ y $y = 1$, que son los promedios de los valores iniciales.

Pregunta 4.-

Las trayectorias seguidas cuando se usan múltiples agentes no son líneas rectas, sino curvas. Este comportamiento curvilíneo se debe a la dinámica inherente de los robots no holónomos y a las interacciones entre ellos definidas por la matriz Laplaciana. No obstante, el robot solo también gira, pero hay que tomar en cuenta que se grafica el punto de control y no el agente en sí. Como se puede apreciar en la ley de control, gira rápidamente al inicio para acomodarse en la dirección al punto de origen.

Se observa que la ley de control aplicada en el grafo no dirigido logra una convergencia más rápida en comparación con el grafo dirigido. Esta diferencia en la velocidad de convergencia puede atribuirse a la estructura simétrica del grafo no dirigido, que facilita una comunicación y coordinación más eficiente entre los agentes.

Anexo (código usado)

Pregunta 1

```
clc
clearvars
close all

% Parametros iniciales
l = 1; % Distancia del centro del robot al punto de control
z = [2; -1]; % Condiciones iniciales punto 1
Angulo = 0; % Angulo inicial ( en grados) Condicion uno
K = [1 0; 0 1]; % Ganancia del controlador


% Datos de la simulacion
Dt = 0.01; % Periodo de muestreo
tiempo = 15; % Duracion de la simulacion en segundos
iteraciones = tiempo / Dt;

%Inicializar el angulo theta
theta=zeros(1,iteraciones+1);
theta(1,1)=deg2rad(Angulo);
% Inicializar trayectorias
z_hist = zeros(2, iteraciones+1);
z_hist(:, 1) = z;

% Inicializar trayectorias
v_hist = zeros(1,iteraciones);
% Inicializar trayectorias
w_hist = zeros(1,iteraciones);
% Simulacion
for k = 1:iteraciones
    % Calcular la matriz M en cada iteracion
    M = [cos(theta(k)) -l*sin(theta(k));
         sin(theta(k)) l*cos(theta(k))];

    % Controlador
    u = M \ (-K * z(:,k)); % u = M^{-1}(-kz)

    % Actualizar el estado del sistema
    v = u(1);
    w = u(2);

    % Dinamica del sistema
    z(:,k+1) = z(:,k)+Dt*(M*u);
    theta(k+1) = theta(k)+Dt*(w) ; % Actualizacion del angulo

    %Errores
    ex(k)=z(1,k);
    ey(k)=z(2,k);

    % Guardar el historial de valore para graficar
    z_hist(:, k+1) = z(:,k+1);
    v_hist(k)=v;
    w_hist(k)=w;
end

t=linspace(0,tiempo,iteraciones);

% Grafica de resultados
```

```
figure;
hold on
plot(z_hist(1, :), z_hist(2, :));
quiver(z_hist(1, 1), z_hist(2, 1), cos(theta(1)), sin(theta(1)),
    0.1, 'r', 'LineWidth', .1, 'MaxHeadSize', 1 );
quiver(z_hist(1, end), z_hist(2, end), cos(theta(end)), sin(theta(
    end)), 0.1, 'g', 'LineWidth', .1, 'MaxHeadSize', 2);
xlabel('x');
ylabel('y');
title('Trayectoria del robot no holonimo');
hold off
grid on;

figure
subplot(1,2,1)
plot(t,ex)
xlabel('t');
ylabel('error');
title('Error en el eje x');
grid on
subplot(1,2,2)
plot(t,ey)
plot(t,ex)
xlabel('t');
ylabel('error');
title('Error en el eje y');
grid on


figure
subplot(1,2,1)
plot(t,v_hist)
xlabel('t');
ylabel('v');
title('Ley de control v');
grid on
subplot(1,2,2)
plot(t,w_hist)
xlabel('t');
ylabel('w');
title('Ley de control w');
grid on
```

Pregunta 2 y 3

```
clc
clearvars
close all

% Parametros iniciales
l = 0.5; % Distancia del centro del robot al punto de control
K = 1*eye(8); % Ganancia del controlador )

```

```
% Inicializar condiciones iniciales para 4 robots
z = [2; 1; -1; -2; 2; -2; 2; -1]; % Cada columna es [x; y] de un robot
Angulo = [90; 45; -45; -90]; % Angulos dados en grados

% Matriz Laplaciana de un grafo dirigido en ciclo para 4 robots
% L = [ 1  0  0 -1;
%      -1  1  0  0;
%       0 -1  1  0;
%       0  0 -1  1];
%Laplaciano grafo no dirigido
L=[2 -1  0 -1;
  -1  2 -1  0;
   0 -1  2 -1;
  -1  0 -1  2];
I=eye(2);

% Datos de la simulacion
Dt = 0.01; % Periodo de muestreo
tiempo = 30; % Duracion de la simulacion en segundos
iteraciones = tiempo / Dt;

% Inicializar trayectorias
z_hist = zeros(8, iteraciones+1);
z_hist(:, 1) = z;

theta = zeros(4, iteraciones+1);
theta(:, 1) = deg2rad(Angulo);


v_hist = zeros(4, iteraciones);
w_hist = zeros(4, iteraciones);

evx_hist = zeros(4,iteraciones);
evy_hist = zeros(4,iteraciones);

% Simulacion
for k = 1:iteraciones
    % Calcular la matriz M para cada robot en cada iteracion
    M1 = [cos(theta(1,k)) -l*sin(theta(1,k));
          sin(theta(1,k)) l*cos(theta(1,k))];
    M2 = [cos(theta(2,k)) -l*sin(theta(2,k));
          sin(theta(2,k)) l*cos(theta(2,k))];
    M3 = [cos(theta(3,k)) -l*sin(theta(3,k));
          sin(theta(3,k)) l*cos(theta(3,k))];
    M4 = [cos(theta(4,k)) -l*sin(theta(4,k));
          sin(theta(4,k)) l*cos(theta(4,k))];
    M = blkdiag(M1, M2, M3, M4);

    ev=-kron(L,I)*z(:,k);

    u= M \ (K * ev);

    %Separo la ley de control en lineal y rotacional
    v=[u(1),u(3),u(5),u(7)];
    w=[u(2),u(4),u(6),u(8)];
    % Dinamica del sistema
```

```
    z(:,k+1) = z(:,k)+Dt*(M*u);
    theta(:,k+1) = theta(:,k)+Dt*(w') ; % Actualizacion del angulo

    % Guardar el historial de valores para grafica
    z_hist(:, k+1) = z(:,k+1);
    v_hist(:,k)=v';
    w_hist(:,k)=w';
    evx_hist(:,k)=[ev(1);ev(3);ev(5);ev(7)];
    evy_hist(:,k)=[ev(2);ev(4);ev(6);ev(8)];
end

t = linspace(0, tiempo, iteraciones);

% Grafica de resultados
figure;
hold on
plot(z_hist(1, :), z_hist(2, :),'r');
quiver(z_hist(1, 1), z_hist(2, 1), cos(theta(1,1)), sin(theta(1,1)), 0.1, 'r', 'LineWidth', .1, 'MaxHeadSize', 1 );
quiver(z_hist(1, end), z_hist(2, end), cos(theta(1,end)), sin(theta(1,end)), 0.1, 'r', 'LineWidth', .1, 'MaxHeadSize', 2);
plot(z_hist(3, :), z_hist(4, :),'b');
quiver(z_hist(3, 1), z_hist(4, 1), cos(theta(2,1)), sin(theta(2,1)), 0.1, 'r', 'LineWidth', .1, 'MaxHeadSize', 1 );
quiver(z_hist(3, end), z_hist(4, end), cos(theta(2,end)), sin(theta(2,end)), 0.1, 'b', 'LineWidth', .1, 'MaxHeadSize', 2);
plot(z_hist(5, :), z_hist(6, :),'g');
quiver(z_hist(5, 1), z_hist(6, 1), cos(theta(3,1)), sin(theta(3,1)), 0.1, 'r', 'LineWidth', .1, 'MaxHeadSize', 1 );
quiver(z_hist(5, end), z_hist(6, end), cos(theta(3,end)), sin(theta(3,end)), 0.1, 'g', 'LineWidth', .1, 'MaxHeadSize', 2);
plot(z_hist(7, :), z_hist(8, :),'k');
quiver(z_hist(7, 1), z_hist(8, 1), cos(theta(4,1)), sin(theta(4,1)), 0.1, 'r', 'LineWidth', .1, 'MaxHeadSize', 1 );
quiver(z_hist(7, end), z_hist(8, end), cos(theta(4,end)), sin(theta(4,end)), 0.1, 'k', 'LineWidth', .1, 'MaxHeadSize', 2);
xlabel('x');
ylabel('y');
title('Trayectoria del robot no holonomo');
hold off
grid on;


figure
subplot(1,2,1)
plot(t,evx_hist)
xlabel('t');
ylabel('error');
title('Error en el eje x');
grid on
subplot(1,2,2)
plot(t,evy_hist)
xlabel('t');
ylabel('error');
title('Error en el eje x');
grid on
```

```
figure
hold on
subplot(1,2,1)
plot(t,v_hist)
xlabel('t');
ylabel('v');
title('Ley de control v');
grid on
subplot(1,2,2)
plot(t,w_hist)
xlabel('t');
ylabel('w');
title('Ley de control w');
grid on
hold off

%Esto es para tener referencia de los grafos
An=[0 0 0 1;
    1 0 0 0;
    0 1 0 0;
    0 0 1 0];
Gn=digraph(An');

figure
plot(Gn, 'Layout', 'force', 'EdgeLabel', Gn.Edges.Weight);

A=[0 1 0 1;
   1 0 1 0;
   0 1 0 1;
   1 0 1 0];

G=graph(A);

figure
plot(G);
```