



Cinvestav

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

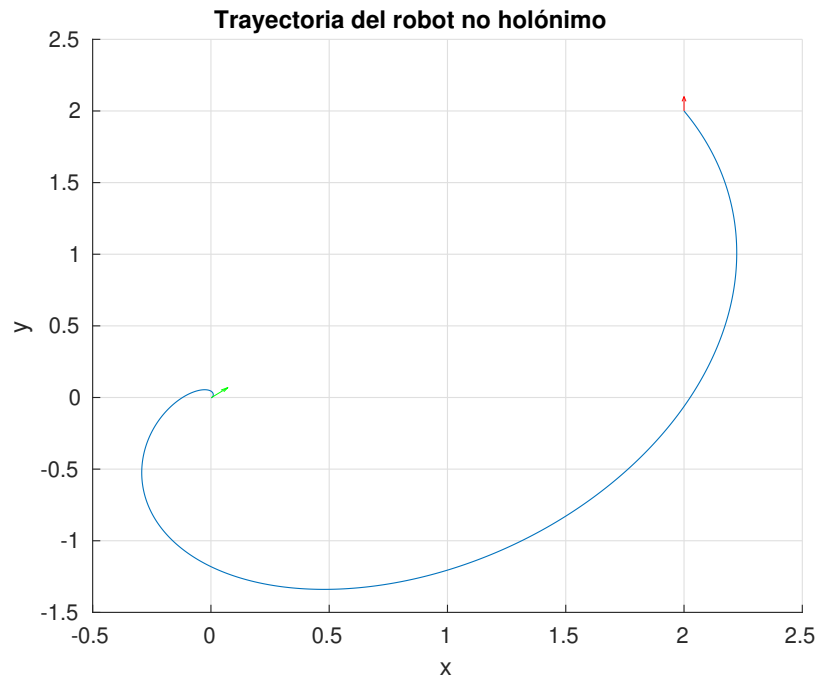
15 julio 2024

Pregunta 1.-

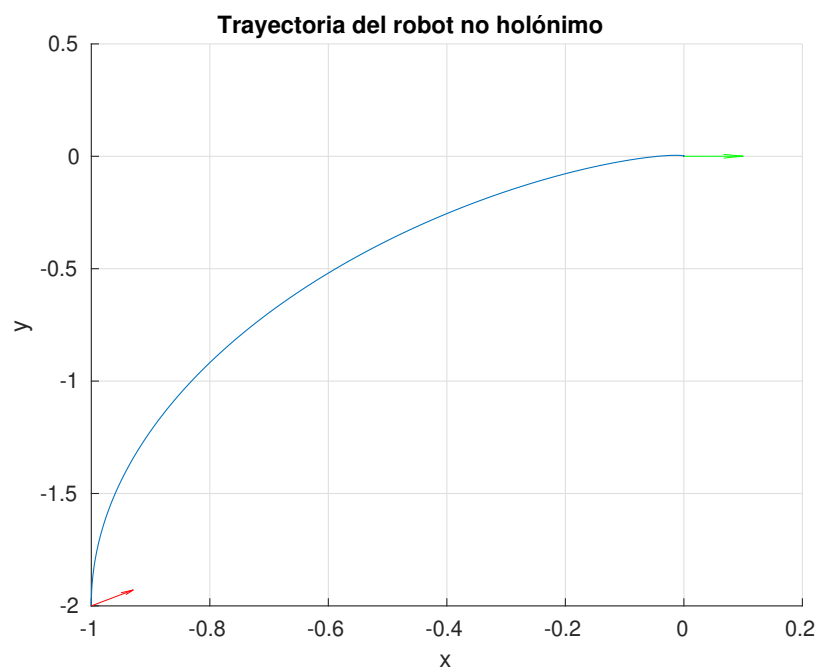
Considerando un solo robot de manejo diferencial y utilizando el planteamiento del punto de control.

Inciso a)

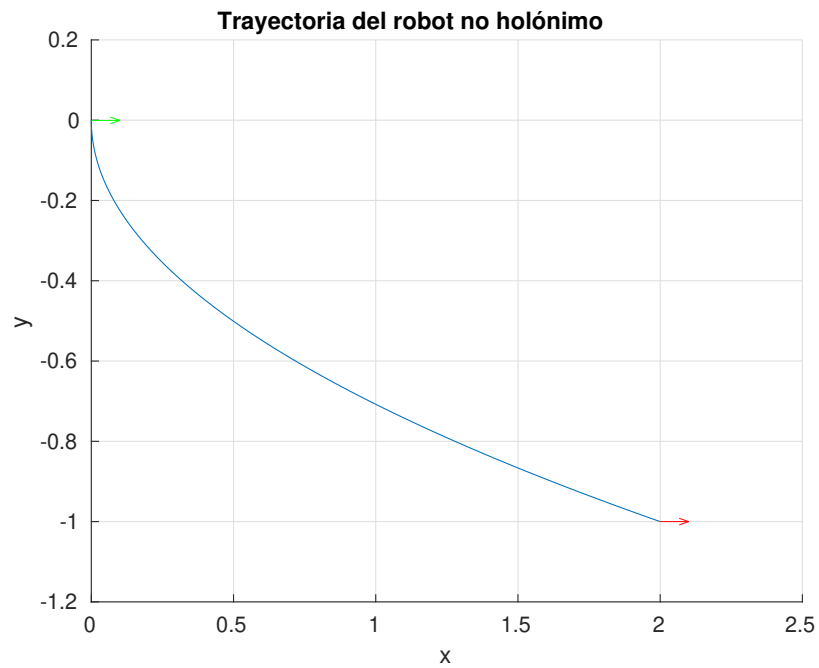
Probando con tres condiciones iniciales, la primera en $z = [2 \ 2], \theta = 90$, obtenemos el siguiente resultado (la flecha roja indica el inicio de la trayectoria y su ángulo, la flecha verde el final, lo mismo para las siguientes condiciones iniciales)



Con condiciones iniciales $z = [-1 \ -2], \theta = 45$:



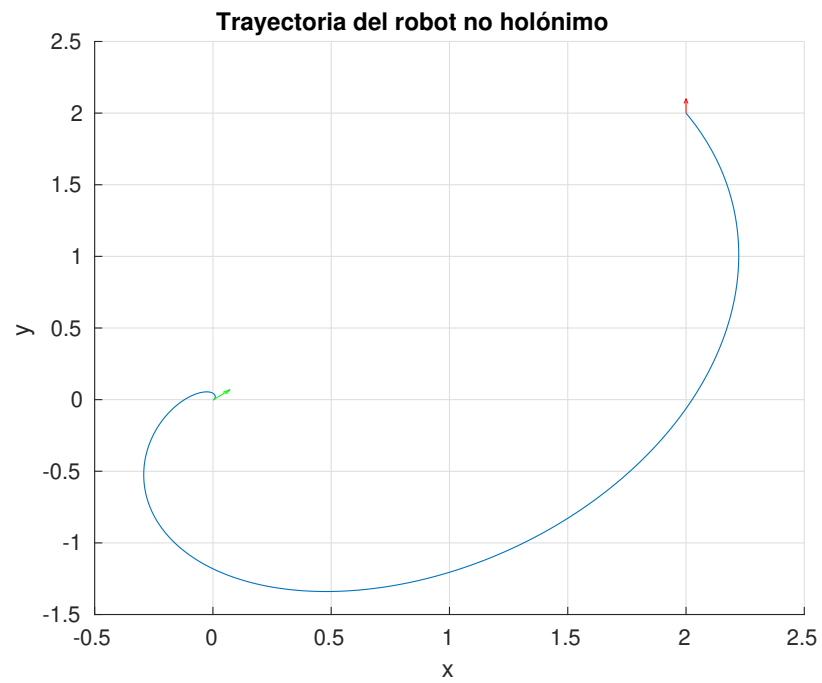
Con condiciones iniciales $z = \begin{bmatrix} -1 & -2 \end{bmatrix}, \theta = 0$:



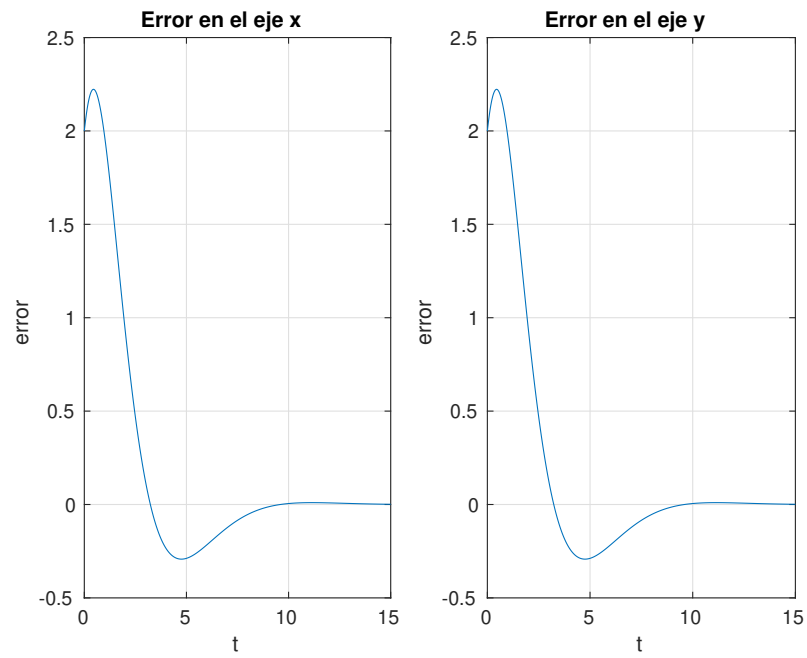
Nota: Las ganancias de control son iguales tanto en velocidad lineal como rotacional, podrían ser distintas, pero es más fácil programarlo así.

Inciso b)

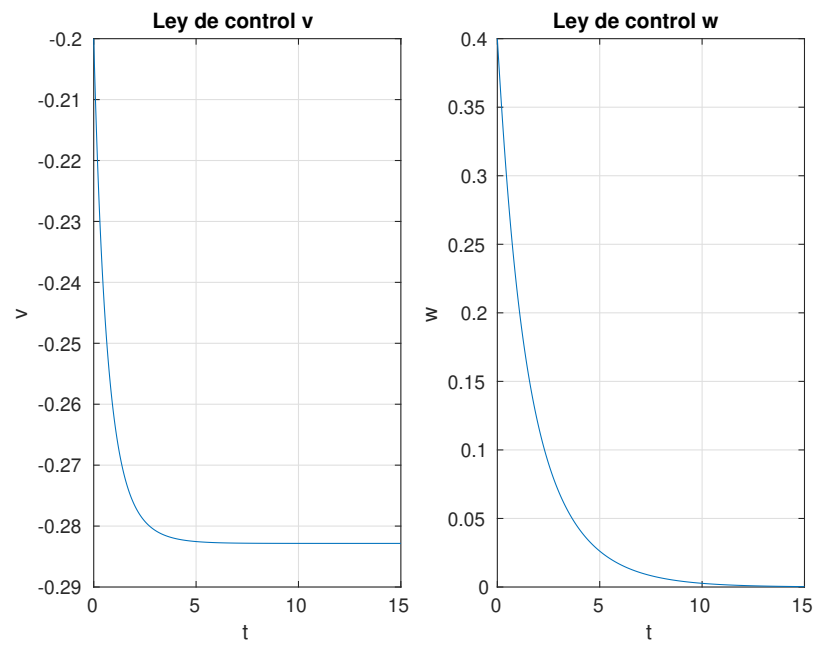
Utilizando las primeras condiciones iniciales, es decir:



La evolución del error en cada coordenada es la siguiente

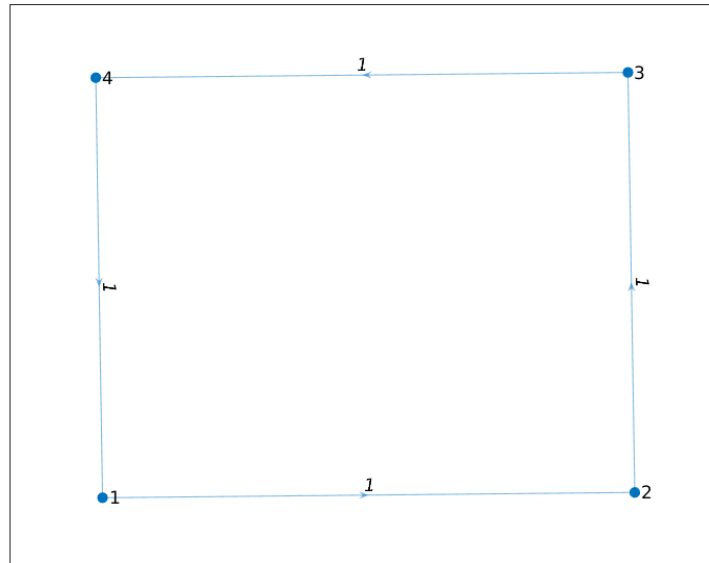


Y la evolución de las leyes de control tanto lineal como angular son:



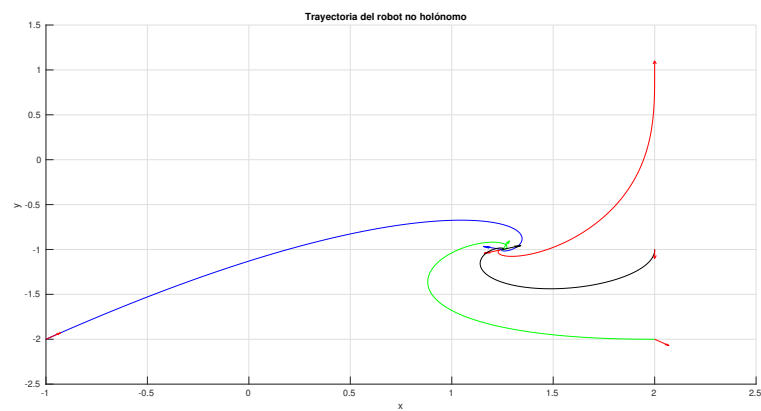
Pregunta 2.-

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



Inciso a)

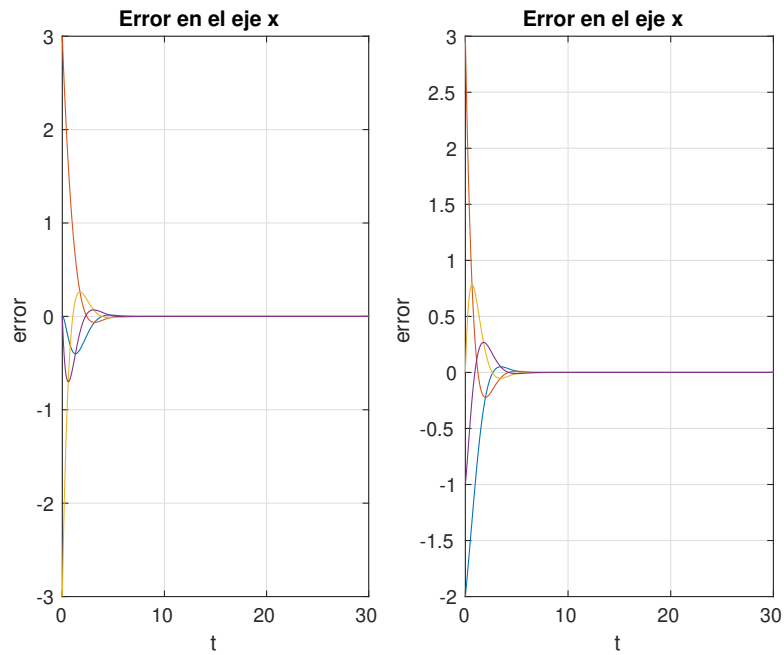
Obtenemos las siguientes trayectorias



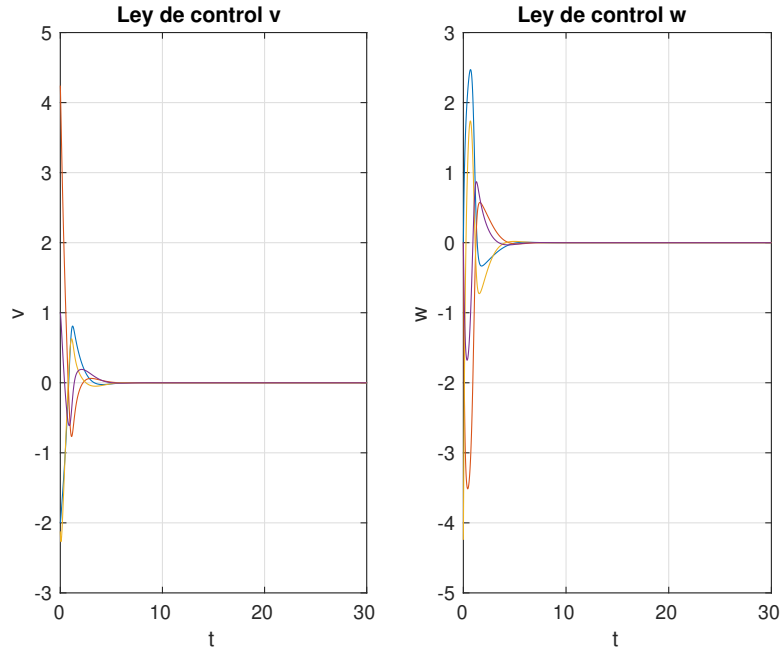
Las flechas rojas indican el inicio de las trayectorias, y para el final hay una flecha de cada color del robot.

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

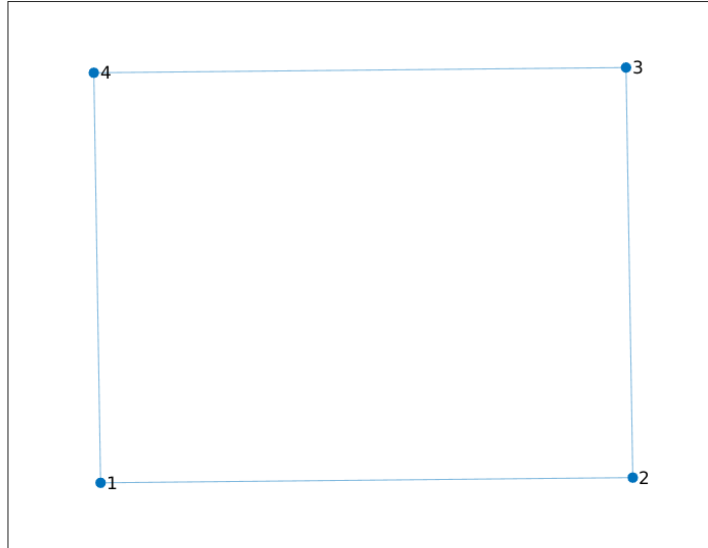


Inciso b)

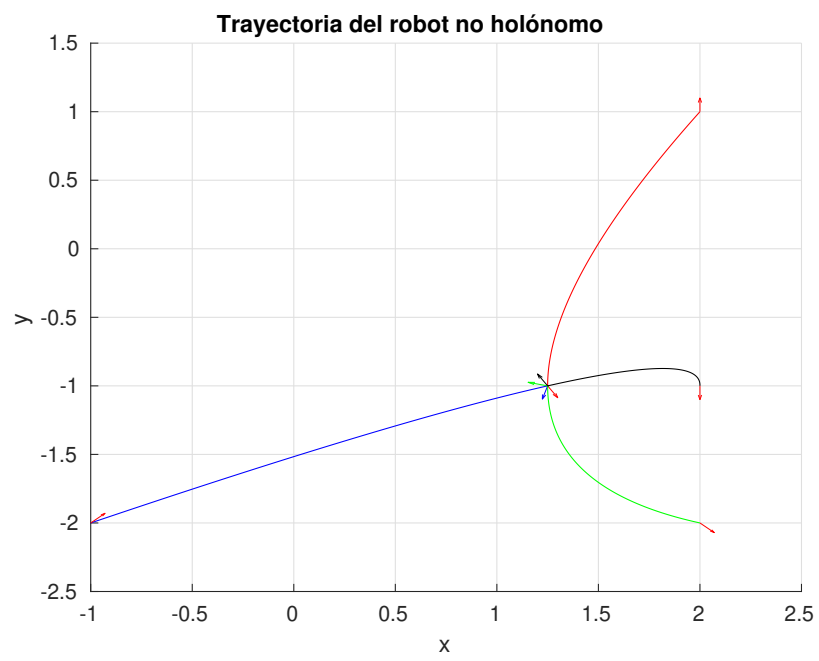
Si calculamos el vector nulo izquierdo normalizado de la matriz Laplaciana, el cual es $q = [1/4 \ 1/4 \ 1/4 \ 1/4]$, y tomamos como referencia que los valores iniciales son $x_0 = [2 \ -1 \ 2 \ 2]^T$ y $y_0 = [1 \ -2 \ -2 \ -1]^T$. Los robots convergen a $x = 5/4$ y $y = 1$, lo que corresponde con los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

Pregunta 3.-

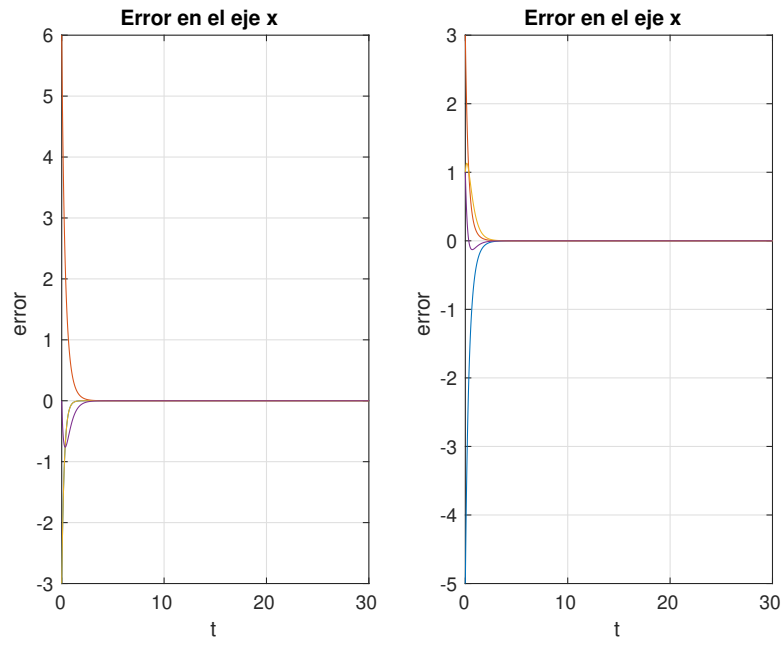
Ahora, considerando un grafo no dirigido, pero igualmente un ciclo, como el que se muestra a continuación:



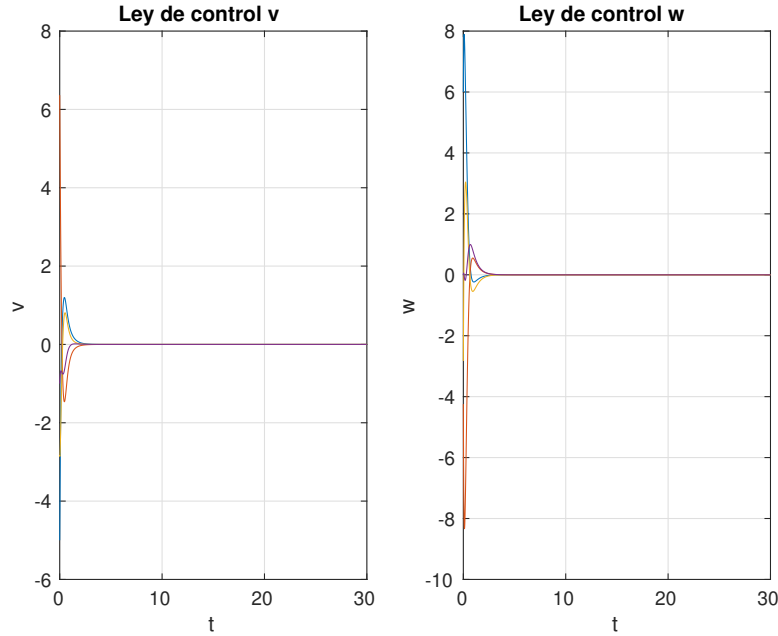
obtenemos la siguiente evolución de trayectorias



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.-

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.

Las trayectorias seguidas por los 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, es posible obtener trayectorias rectilíneas alineando adecuadamente los agentes en la dirección del origen desde el inicio. Esta alineación inicial puede simplificar la dinámica y resultar en trayectorias más directas hacia el objetivo.

Anexo (código usado)

Pregunta 1

```

1  clc
2  clearvars
3  close all
4
5  % Parametros iniciales
6  l = .5; % Distancia del centro del robot al punto de control
7  z = [-1; -2]; % Condiciones iniciales punto 1
8  Angulo = 45; % Angulo inicial ( en grados) Condicion uno
9  K = [0.1 0; 0 0.1]; % Ganancia del controlador
10
11
12 % Datos de la simulacion
13 Dt = 0.01; % Periodo de muestreo
14 tiempo = 15; % Duracion de la simulacion en segundos
15 iteraciones = tiempo / Dt;
16
17 %Inicializar el angulo theta
18 theta=zeros(1,iteraciones+1);
19 theta(1,1)=deg2rad(Angulo);
20 % Inicializar trayectorias
21 z_hist = zeros(2, iteraciones+1);
22 z_hist(:, 1) = z;
23
24 % Inicializar trayectorias
25 v_hist = zeros(1,iteraciones);
26 % Inicializar trayectorias
27 w_hist = zeros(1,iteraciones);
28 % Simulacion
29 for k = 1:iteraciones
30     % Calcular la matriz M en cada iteracion
31     M = [cos(theta(k)) -l*sin(theta(k));
32         sin(theta(k)) l*cos(theta(k))];
33
34     % Controlador
35     u = M \ (-K * z); %  $u = M^{-1}(-Kz)$ 
36
37     % Actualizar el estado del sistema
38     v = u(1);
39     w = u(2);
40
41     % Dinamica del sistema
42     z(:,k+1) = z(:,k)-Dt*(M*z(:,k));
43     theta(k+1) = theta(k)-Dt*(w*theta(k)) ; % Actualizacion del
        angulo
44
45     %Errores
46     ex(k)=z(1,k);
47     ey(k)=z(2,k);
48
49     % Guardar el historial de valores para graficar
50     z_hist(:, k+1) = z(:,k+1);
51     v_hist(k)=v;
52     w_hist(k)=w;
53 end
54
55 t=linspace(0,tiempo,iteraciones);
56

```

```

57 % Grafica de resultados
58 figure;
59 hold on
60 plot(z_hist(1, :), z_hist(2, :));
61 quiver(z_hist(1, 1), z_hist(2, 1), cos(theta(1)), sin(theta(1)),
        0.1, 'r', 'LineWidth', .1, 'MaxHeadSize', 1 );
62 quiver(z_hist(1, end), z_hist(2, end), cos(theta(end)), sin(theta(
        end)), 0.1, 'g', 'LineWidth', .1, 'MaxHeadSize', 2);
63 xlabel('x');
64 ylabel('y');
65 title('Trayectoria del robot no holonimo');
66 hold off
67 grid on;
68
69 figure
70 subplot(1,2,1)
71 plot(t,ex)
72 xlabel('t');
73 ylabel('error');
74 title('Error en el eje x');
75 grid on
76 subplot(1,2,2)
77 plot(t,ey)
78 plot(t,ex)
79 xlabel('t');
80 ylabel('error');
81 title('Error en el eje y');
82 grid on
83
84
85 figure
86 subplot(1,2,1)
87 plot(t,v_hist)
88 xlabel('t');
89 ylabel('v');
90 title('Ley de control v');
91 grid on
92 subplot(1,2,2)
93 plot(t,w_hist)
94 xlabel('t');
95 ylabel('w');
96 title('Ley de control w');
97 grid on

```

Pregunta 2 y 3

```

1 clc
2 clearvars
3 close all
4
5 % Parametros iniciales
6 l = 0.5; % Distancia del centro del robot al punto de control
7 K = 1*eye(8); % Ganancia del controlador )

```

```

8
9 % Inicializar condiciones iniciales para 4 robots
10 z = [2; 1; -1; -2; 2; -2; 2; -1]; % Cada columna es [x; y] de un
    robot
11 Angulo = [90; 45; -45; -90]; % Angulos dados en grados
12
13 % Matriz Laplaciana de un grafo dirigido en ciclo para 4 robots
14 % L = [ 1  0  0 -1;
15 %      -1  1  0  0;
16 %        0 -1  1  0;
17 %        0  0 -1  1];
18 %Laplaciano grafo no dirigido
19 L=[2 -1  0 -1;
20     -1  2 -1  0;
21     0 -1  2 -1;
22     -1  0 -1  2];
23 I=eye(2);
24
25 % Datos de la simulacion
26 Dt = 0.01; % Periodo de muestreo
27 tiempo = 30; % Duracion de la simulacion en segundos
28 iteraciones = tiempo / Dt;
29
30 % Inicializar trayectorias
31 z_hist = zeros(8, iteraciones+1);
32 z_hist(:, 1) = z;
33
34 theta = zeros(4, iteraciones+1);
35 theta(:, 1) = deg2rad(Angulo);
36
37
38 v_hist = zeros(4, iteraciones);
39 w_hist = zeros(4, iteraciones);
40
41 evx_hist = zeros(4, iteraciones);
42 evy_hist = zeros(4, iteraciones);
43
44 % Simulacion
45 for k = 1:iteraciones
46     % Calcular la matriz M para cada robot en cada iteracion
47     M1 = [cos(theta(1,k)) -1*sin(theta(1,k));
48           sin(theta(1,k))  1*cos(theta(1,k))];
49     M2 = [cos(theta(2,k)) -1*sin(theta(2,k));
50           sin(theta(2,k))  1*cos(theta(2,k))];
51     M3 = [cos(theta(3,k)) -1*sin(theta(3,k));
52           sin(theta(3,k))  1*cos(theta(3,k))];
53     M4 = [cos(theta(4,k)) -1*sin(theta(4,k));
54           sin(theta(4,k))  1*cos(theta(4,k))];
55     M = blkdiag(M1, M2, M3, M4);
56
57     ev=-kron(L,I)*z(:,k);
58
59     u= M \ (K * ev);
60
61     %Separo la ley de control en lineal y rotacional
62     v=[u(1),u(3),u(5),u(7)];
63     w=[u(2),u(4),u(6),u(8)];

```

```

64 % Dinamica del sistema
65 z(:,k+1) = z(:,k)+Dt*(M*u);
66 theta(:,k+1) = theta(:,k)+Dt*(w*theta(:,k)) ; % Actualizacion
    del angulo
67
68 % Guardar el historial de valores para grafica
69 z_hist(:, k+1) = z(:,k+1);
70 v_hist(:,k)=v';
71 w_hist(:,k)=w';
72 evx_hist(:,k)=[ev(1);ev(3);ev(5);ev(7)];
73 evy_hist(:,k)=[ev(2);ev(4);ev(6);ev(8)];
74 end
75
76 t = linspace(0, tiempo, iteraciones);
77
78 % Grafica de resultados
79 figure;
80 hold on
81 plot(z_hist(1, :), z_hist(2, :),'r');
82 quiver(z_hist(1, 1), z_hist(2, 1), cos(theta(1,1)), sin(theta(1,1))
    , 0.1, 'r', 'LineWidth', .1, 'MaxHeadSize', 1 );
83 quiver(z_hist(1, end), z_hist(2, end), cos(theta(1,end)), sin(theta
    (1,end)), 0.1, 'r', 'LineWidth', .1, 'MaxHeadSize', 2);
84 plot(z_hist(3, :), z_hist(4, :),'b');
85 quiver(z_hist(3, 1), z_hist(4, 1), cos(theta(2,1)), sin(theta(2,1))
    , 0.1, 'r', 'LineWidth', .1, 'MaxHeadSize', 1 );
86 quiver(z_hist(3, end), z_hist(4, end), cos(theta(2,end)), sin(theta
    (2,end)), 0.1, 'b', 'LineWidth', .1, 'MaxHeadSize', 2);
87 plot(z_hist(5, :), z_hist(6, :),'g');
88 quiver(z_hist(5, 1), z_hist(6, 1), cos(theta(3,1)), sin(theta(3,1))
    , 0.1, 'r', 'LineWidth', .1, 'MaxHeadSize', 1 );
89 quiver(z_hist(5, end), z_hist(6, end), cos(theta(3,end)), sin(theta
    (3,end)), 0.1, 'g', 'LineWidth', .1, 'MaxHeadSize', 2);
90 plot(z_hist(7, :), z_hist(8, :),'k');
91 quiver(z_hist(7, 1), z_hist(8, 1), cos(theta(4,1)), sin(theta(4,1))
    , 0.1, 'r', 'LineWidth', .1, 'MaxHeadSize', 1 );
92 quiver(z_hist(7, end), z_hist(8, end), cos(theta(4,end)), sin(theta
    (4,end)), 0.1, 'k', 'LineWidth', .1, 'MaxHeadSize', 2);
93 xlabel('x');
94 ylabel('y');
95 title('Trayectoria del robot no holonomo');
96 hold off
97 grid on;
98
99
100 figure
101 subplot(1,2,1)
102 plot(t,evx_hist)
103 xlabel('t');
104 ylabel('error');
105 title('Error en el eje x');
106 grid on
107 subplot(1,2,2)
108 plot(t,evy_hist)
109 xlabel('t');
110 ylabel('error');
111 title('Error en el eje y');

```



```

112 grid on
113
114
115 figure
116 hold on
117 subplot(1,2,1)
118 plot(t,v_hist)
119 xlabel('t');
120 ylabel('v');
121 title('Ley de control v');
122 grid on
123 subplot(1,2,2)
124 plot(t,w_hist)
125 xlabel('t');
126 ylabel('w');
127 title('Ley de control w');
128 grid on
129 hold off
130
131 %Esto es para tener referencia de los grafos
132 An=[0 0 0 1;
133     1 0 0 0;
134     0 1 0 0;
135     0 0 1 0];
136 Gn=digraph(An');
137
138 figure
139 plot(Gn, 'Layout', 'force', 'EdgeLabel', Gn.Edges.Weight);
140
141 A=[0 1 0 1;
142    1 0 1 0;
143    0 1 0 1;
144    1 0 1 0];
145
146 G=graph(A);
147
148 figure
149 plot(G);

```