



Cinvestav

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

Tarea 5. Control en formación

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

19 de Junio del 2024

Pregunta 1.

Consideremos agentes modelados como dobles integradores, es decir $\ddot{p} = u_i$. Consideremos, además, la referencia como una curva de Lissajous modelado con los siguientes parámetros

$$A = 2$$

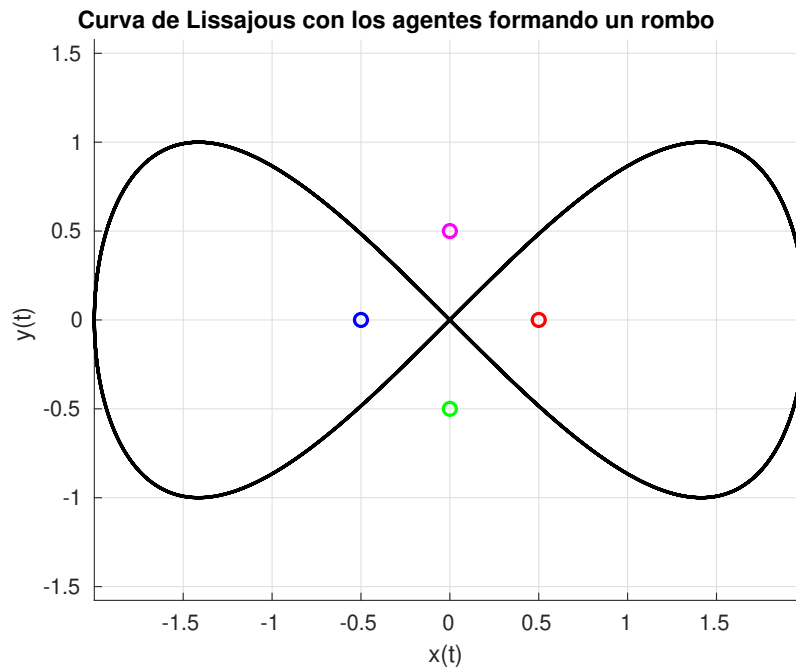
$$B = 1$$

$$a = 1$$

$$b = 2$$

$$\text{delta} = 0$$

y parametrizada en el tiempo como $x = A\sin(at + \text{delta})$ y $y = B\sin(bt)$. Se desea lograr un seguimiento en formación en diamante como se muestra en la siguiente figura:

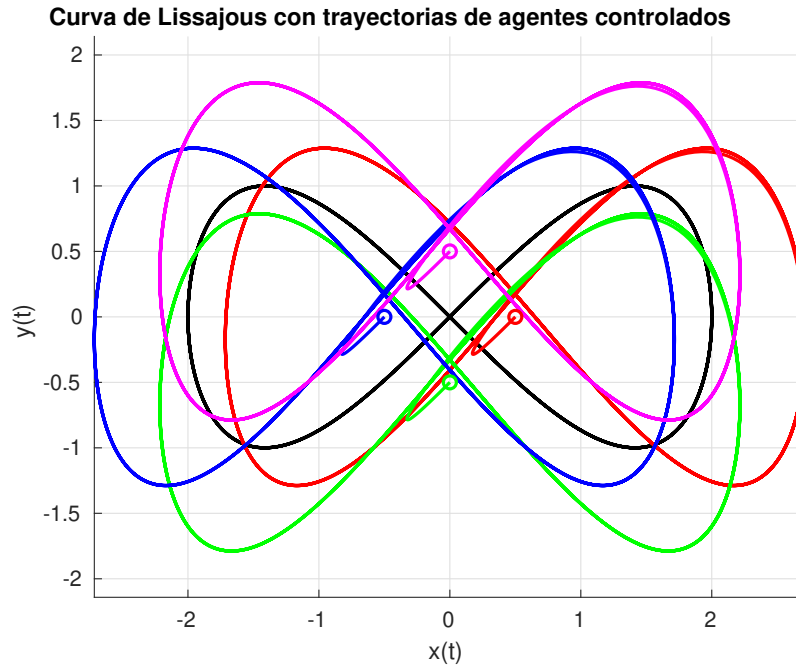


Pregunta 2.-

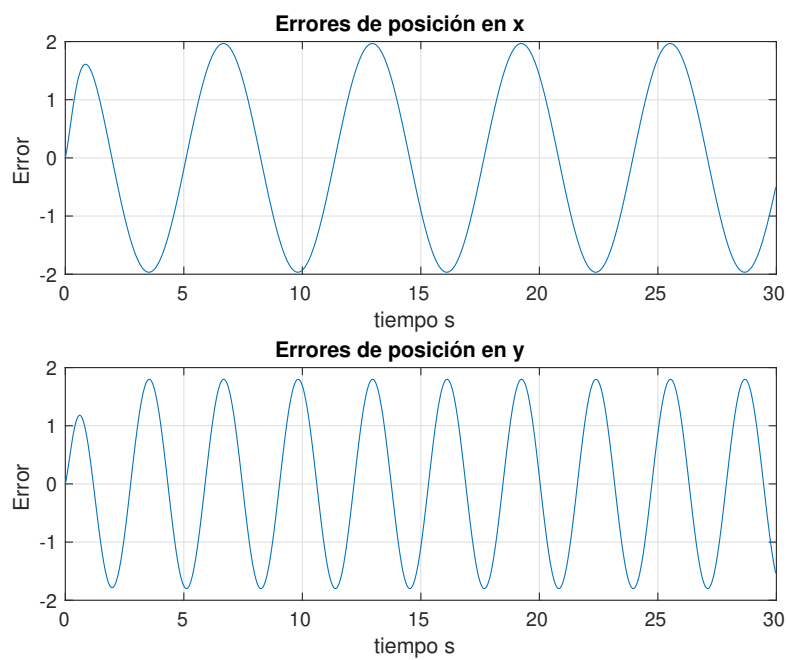
Las posiciones de cada agente está definida con respecto a la curva a una unidad de separación en sentido horario, es decir, el primer agente (rojo en la figura) está en $x + d, y$ donde x, y son las coordenadas de la curva de Lissajous, el segundo agente (el verde) estaría definido como $x, y - d$, el tercero (azul) $x - d, y$ y el cuarto (rosa) $x, y + d$. Las coordenadas irán cambiando con el tiempo.

Pregunta 3.-

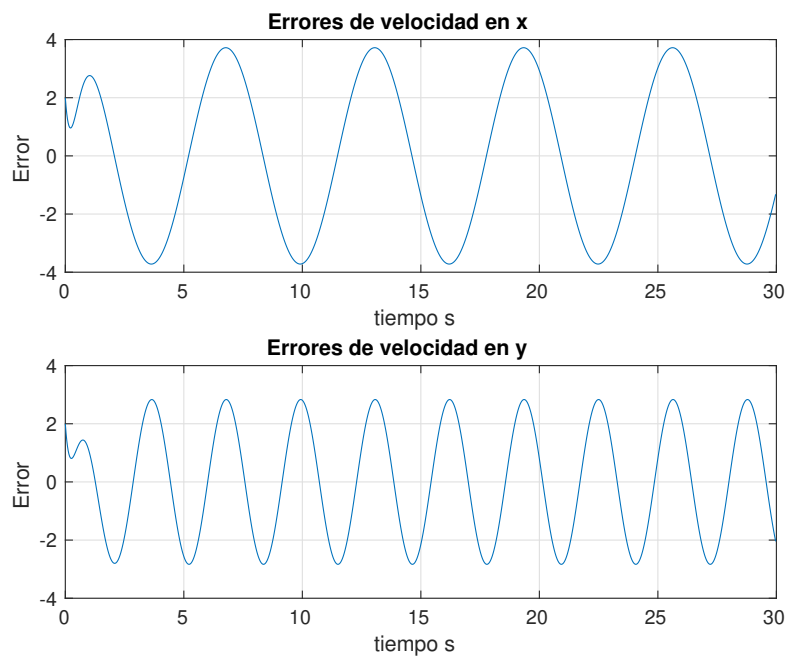
Utilizando la siguiente ley de control: $u = -k_a(k_v e_v - k_p e_p)$, donde $k_a, k_v, k_p > 0$ son ganancias de retroalimentación, $e_p = p^* - p$, es la posición deseada menos la real, y $e_v = v^* - v$ es la velocidad deseada menos la real. Al aplicarla a 4 agentes cuya posición inicial está dada ya en forma de rombo, como en la figura el punto 1, obtenemos los siguientes resultados. Las trayectorias:



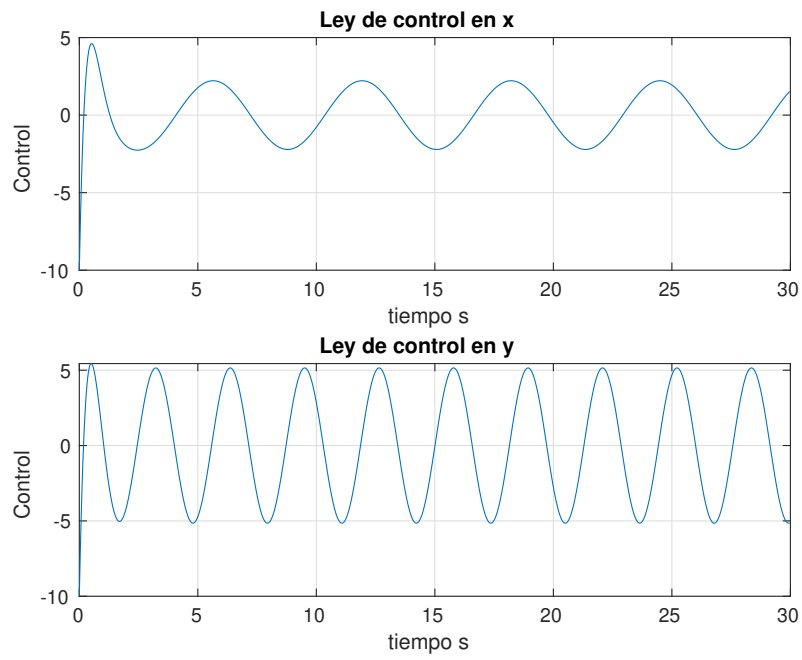
Los errores de posición en x y y , para un solo agente:



Los errores de velocidad en x y y , para un solo agente:

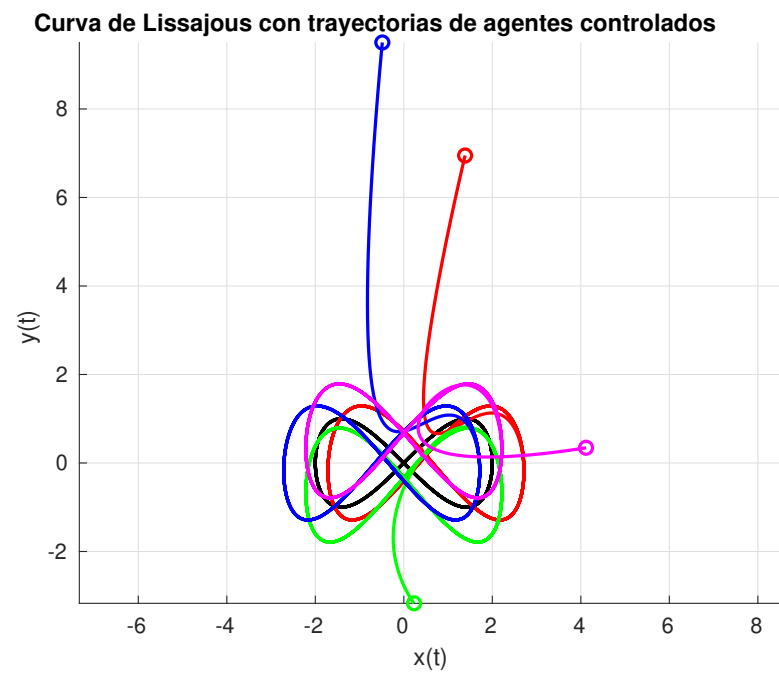


y finalmente las leyes de control para ambas coordenadas:

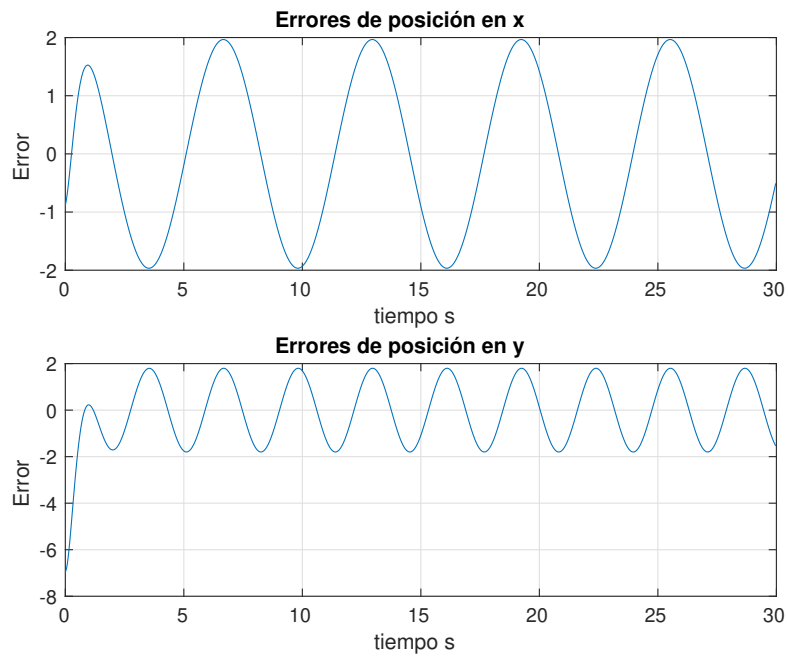


Pregunta 4.-

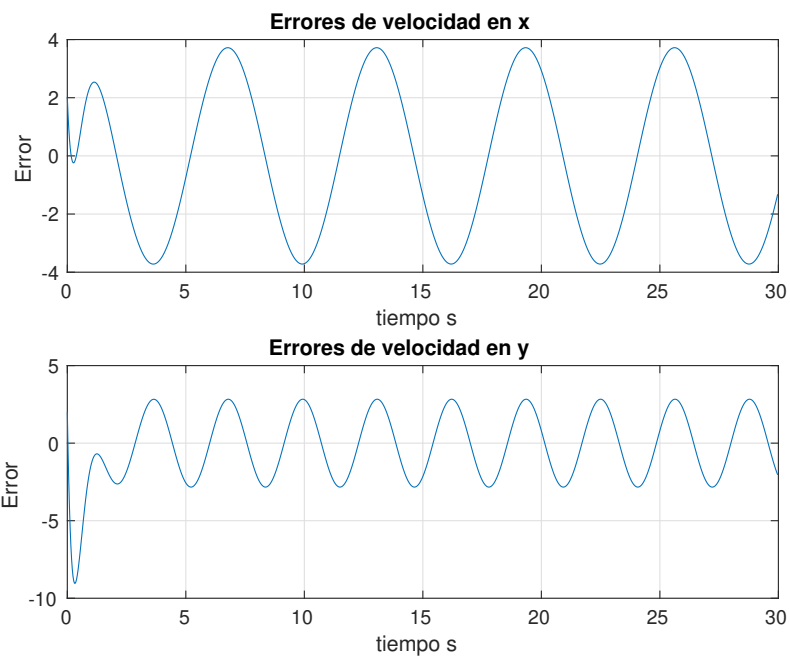
Repetiendo la simulación, pero para condiciones iniciales aleatorias y no partiendo en posición, obtenemos los siguientes resultados:



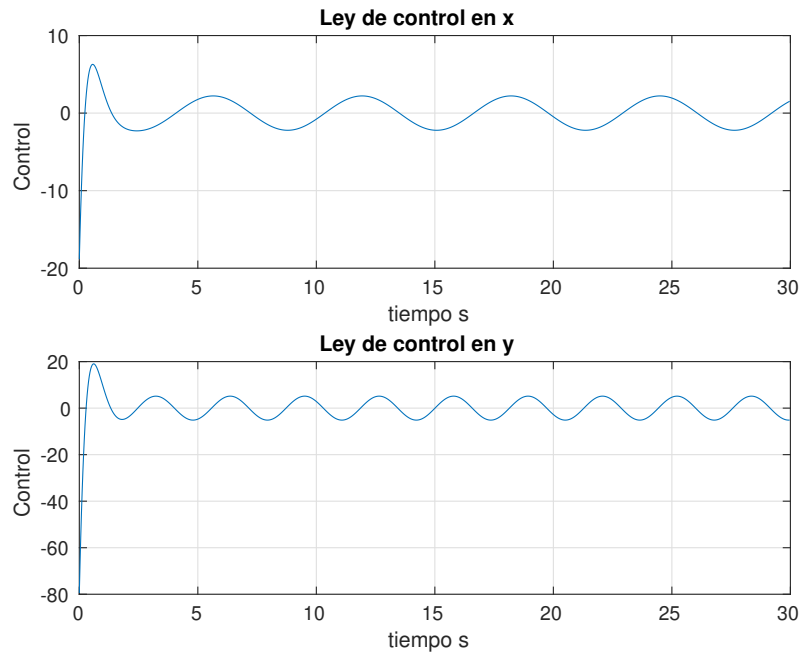
Los errores de posición en x y y , para un solo agente:



Los errores de velocidad en x y y , para un solo agente:



y finalmente las leyes de control para ambas coordenadas:



Pregunta 5.-

El control propuesto fue probado para varios parámetros de la curva de Lissajous, así como para diferentes valores de las ganancias k_i . Los resultados mostraron que al aumentar las ganancias, la convergencia del error se acerca más rápidamente a cero. No obstante, se observó que para lograr una convergencia efectiva, es necesario incrementar las ganancias de manera significativa, lo cual puede no ser práctico en todas las situaciones.

Esta observación sugiere que el sistema de control podría beneficiarse de un enfoque más sofisticado, como la incorporación de referencias de los agentes circundantes. Con un sistema de referencia relativo entre los agentes, es probable que se puedan obtener resultados de convergencia más precisos sin necesidad de utilizar ganancias tan altas. Esto no solo mejoraría la eficiencia del sistema de control, sino que también podría reducir el esfuerzo de control y los posibles efectos adversos de grandes ganancias, como la saturación del actuador o la inestabilidad del sistema.

Anexo (código usado)

```
1 clc
2 clearvars
3 close all
```

```

4
5 % Parametros de la curva de Lissajous
6 A = 2;           % Amplitud en x
7 B = 1;           % Amplitud en y
8 a = 1;           % Frecuencia en x
9 b = 2;           % Frecuencia en y
10 delta = 0;       % Diferencia de fase
11
12 AA=[0 0 1 0;
13      0 0 0 1;
14      0 0 0 0;
15      0 0 0 0];
16
17 BB=[0 0;
18      0 0;
19      1 0;
20      0 1];
21
22 % Tiempo
23 tiempo=30; %tiempo en segundos
24 t = linspace(0, tiempo, 1000); % Aumentar el rango de tiempo para
    la simulacion
25
26 % Ecuaciones parametricas de la curva de Lissajous
27 x_curve = A * sin(a * t + delta);
28 y_curve = B * sin(b * t);
29
30 vx_curve= A*a*cos(a*t);
31 vy_curve= B*b*cos(b*t);
32
33 % Distancia desde la curva para los agentes
34 d = .5;
35
36 % Posiciones relativas de los agentes con respecto a la curva
37 %Primer punto
38 % x_offset = [d, 0, -d, 0];
39 % y_offset = [0, -d, 0, d];
40 %Segundo punto
41 x_offset = 5*[rand(), rand(), -rand(), rand()];
42 y_offset = 10*[rand(), -rand(), rand(), rand()];
43
44 % Parametros de control
45 k_a = 1;
46 k_v = 5;
47 k_p = 10;
48
49 % Inicializar posiciones y velocidades de los agentes
50 p = zeros(4, 1);
51 p2 = zeros(4, 1);
52 p3 = zeros(4, 1);
53 p4 = zeros(4, 1);
54
55 % Inicializar estados
56 %Primer punto
57 p(:,1)=[x_curve(1) + x_offset(1);y_curve(1) + y_offset(1);0;0];
58 p2(:,1)=[x_curve(1) + x_offset(2);y_curve(1) + y_offset(2);0;0];
59 p3(:,1)=[x_curve(1) + x_offset(3);y_curve(1) + y_offset(3);0;0];

```

```

60 p4(:,1)=[x_curve(1) + x_offset(4);y_curve(1) + y_offset(4);0;0];
61
62 % Simulacion
63 dt = t(2) - t(1);
64 for k = 1:length(t)-1
65     % Posiciones deseadas en el siguiente instante de tiempo
66     p_des = [x_curve(k)+d;y_curve(k)];
67     p_des2 = [x_curve(k) ;y_curve(k)-d];
68     p_des3 = [x_curve(k) - d;y_curve(k)];
69     p_des4 = [x_curve(k) ;y_curve(k)+d];
70     % Posicion actual
71     p_real= p(1:2,k);
72     p_real2= p2(1:2,k);
73     p_real3= p3(1:2,k);
74     p_real4= p4(1:2,k);
75     %Error de posicion
76     e_p = p_des-p_real;
77     e_p2 = p_des2-p_real2;
78     e_p3 = p_des3-p_real3;
79     e_p4 = p_des4-p_real4;
80
81     %Velocidad deseada
82     v_des=[vx_curve(k+1);vy_curve(k+1)];
83     v_des2=[vx_curve(k+1);vy_curve(k+1)];
84     v_des3=[vx_curve(k+1);vy_curve(k+1)];
85     v_des4=[vx_curve(k+1);vy_curve(k+1)];
86     %Velocidad actual
87     v_real=p(3:4,k);
88     v_real2=p2(3:4,k);
89     v_real3=p3(3:4,k);
90     v_real4=p4(3:4,k);
91     %Error de velocidad
92     e_v = v_des-v_real;
93     e_v2 = v_des2-v_real2;
94     e_v3 = v_des3-v_real3;
95     e_v4 = v_des4-v_real4;
96     % e_v=-k_p*e_p;
97
98     % Control de aceleracion
99     u = -k_a * (k_v * e_v - k_p * e_p);
100    u2 = -k_a * (k_v * e_v2 - k_p * e_p2);
101    u3 = -k_a * (k_v * e_v3 - k_p * e_p3);
102    u4 = -k_a * (k_v * e_v4 - k_p * e_p4);
103    %Sistema
104    p(:,k+1)=p(:,k)-(AA*p(:,k)+BB*u)*dt;
105    p2(:,k+1)=p2(:,k)-(AA*p2(:,k)+BB*u2)*dt;
106    p3(:,k+1)=p3(:,k)-(AA*p3(:,k)+BB*u3)*dt;
107    p4(:,k+1)=p4(:,k)-(AA*p4(:,k)+BB*u4)*dt;
108
109    % Control
110    ux(:,k)=u(1);
111    uy(:,k)=u(2);
112
113    %Errores en posiciones
114    epx(:,k)=e_p(1);
115    epy(:,k)=e_p(2);
116    epx2(:,k)=e_p2(1);

```

```

117     epy2(:,k)=e_p2(2);
118     epx3(:,k)=e_p3(1);
119     epy3(:,k)=e_p3(2);
120     epx4(:,k)=e_p4(1);
121     epy4(:,k)=e_p4(2);
122
123     %Errores en velocidad
124     evx(:,k)=e_v(1);
125     evy(:,k)=e_v(2);
126     evx2(:,k)=e_v2(1);
127     evy2(:,k)=e_v2(2);
128     evx3(:,k)=e_v3(1);
129     evy3(:,k)=e_v3(2);
130     evx4(:,k)=e_v4(1);
131     evy4(:,k)=e_v4(2);
132
133 end
134
135 % Graficar la curva de Lissajous y las trayectorias de los agentes
136 figure;
137 hold on;
138
139 % Graficar la curva de Lissajous
140 plot(x_curve, y_curve, 'k', 'LineWidth', 1.5);
141
142 % Colores para los agentes
143 colors = ['r', 'g', 'b', 'm'];
144
145 plot(p(1, :), p(2, :), 'Color', colors(1), 'LineWidth', 1.5);
146 plot(p2(1, :), p2(2, :), 'Color', colors(2), 'LineWidth', 1.5);
147 plot(p3(1, :), p3(2, :), 'Color', colors(3), 'LineWidth', 1.5);
148 plot(p4(1, :), p4(2, :), 'Color', colors(4), 'LineWidth', 1.5);
149
150 plot(p(1, 1), p(2, 1), 'o', 'Color', colors(1), 'LineWidth', 1.5);
151 plot(p2(1, 1), p2(2, 1), 'o', 'Color', colors(2), 'LineWidth', 1.5);
152 plot(p3(1, 1), p3(2, 1), 'o', 'Color', colors(3), 'LineWidth', 1.5);
153 plot(p4(1, 1), p4(2, 1), 'o', 'Color', colors(4), 'LineWidth', 1.5);
154
155 title('Curva de Lissajous con trayectorias de agentes controlados')
156 ;
157 xlabel('x(t)');
158 ylabel('y(t)');
159 grid on;
160 axis equal;
161 hold off;
162
163 figure
164 subplot(2,1,1)
165 plot(t(1,1:size(t,2)-1),epx);
166 title('Errores de posicion en x')
167 ylabel('Error')
168 xlabel('tiempo s')
169 grid on
170 subplot(2,1,2)
171 plot(t(1,1:size(t,2)-1),epy);
172 title('Errores de posicion en y')
173 ylabel('Error')

```

```

173 xlabel('tiempo s')
174 grid on
175 %
176 %
177 figure
178 subplot(2,1,1)
179 plot(t(1,1:size(t,2)-1),evx);
180 title('Errores de velocidad en x')
181 ylabel('Error')
182 xlabel('tiempo s')
183 grid on
184 subplot(2,1,2)
185 plot(t(1,1:size(t,2)-1),evy);
186 title('Errores de velocidad en y')
187 ylabel('Error')
188 xlabel('tiempo s')
189 grid on
190
191
192 figure
193 subplot(2,1,1)
194 plot(t(1,1:size(t,2)-1),ux);
195 title('Ley de control en x')
196 ylabel('Control')
197 xlabel('tiempo s')
198 grid on
199 subplot(2,1,2)
200 plot(t(1,1:size(t,2)-1),uy);
201 title('Ley de control en y')
202 ylabel('Control')
203 xlabel('tiempo s')
204 grid on

```