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

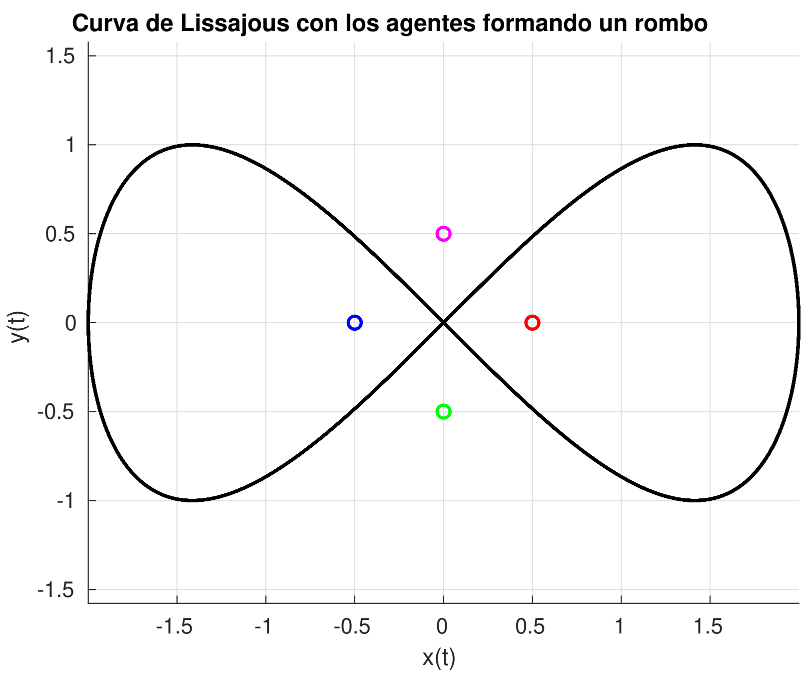
Guadalajara, Jalisco — 19 de Junio del 2024

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

$$\begin{aligned} A &= 2 \\ B &= 1 \\ a &= 1 \\ b &= 2 \\ delta &= 0 \end{aligned}$$

y parametrizada en el tiempo como $x = Asin(at + delta)$ y $y = Bsin(bt)$. Se desea lograr un seguimiento en formación en diamante como se muestra en la siguiente figura:

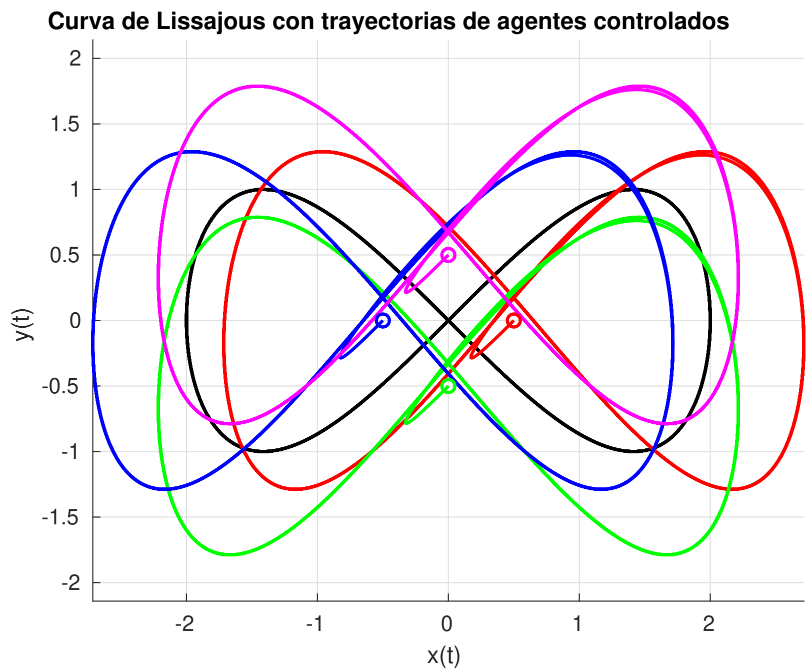


Pregunta 2.-

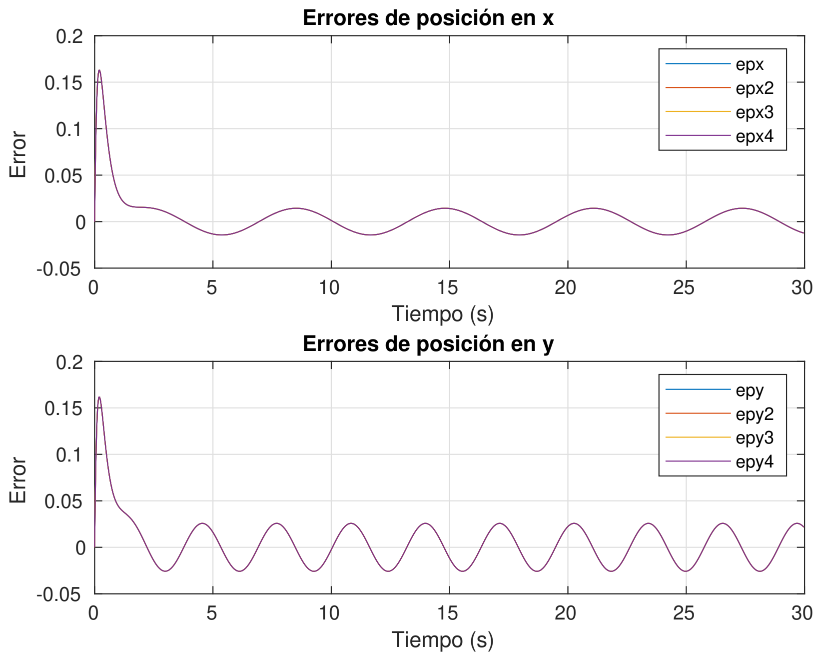
Las posiciones de cada agente está definida con respecto a la curva a una unidad de separación en sentido horario, es decir, el primer agente (rojo en la figura) está en $x + d, y$ donde x, y son las coordenadas de la curva de Lissajous, el segundo agente (el verde) estaría definido como $x, y - d$,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

Pregunta 3.-

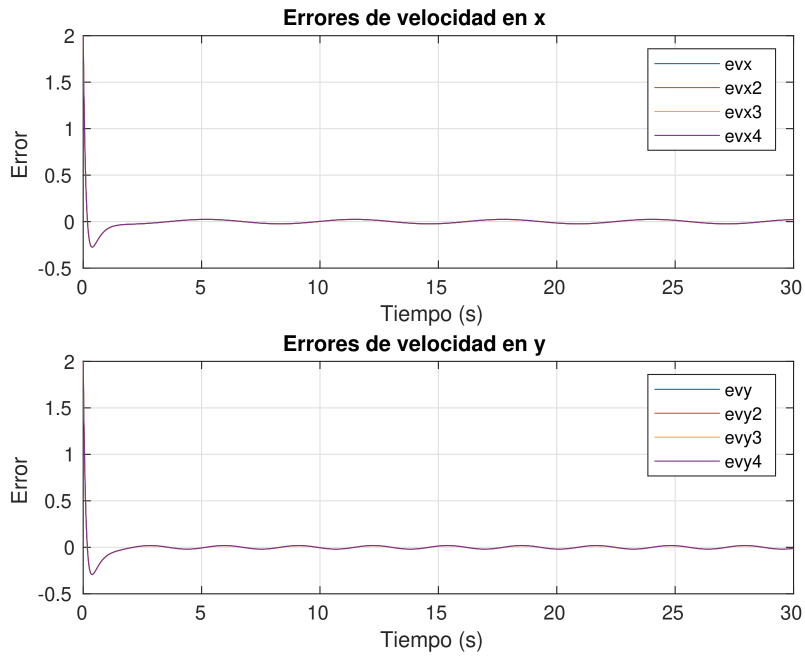
Utilizando la siguiente ley de control: $u = a_{ref} + k_a(k_v e_v + k_p e_p)$, donde $k_a, k_v, k_p > 0$ son ganancias de retroalimentación, $e_p = p^* - p$, es la posición deseada menos la real, a_{ref} es la segunda derivada de la función que define nuestra curva de Lissajous, y $e_v = v^* - v$ es 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:



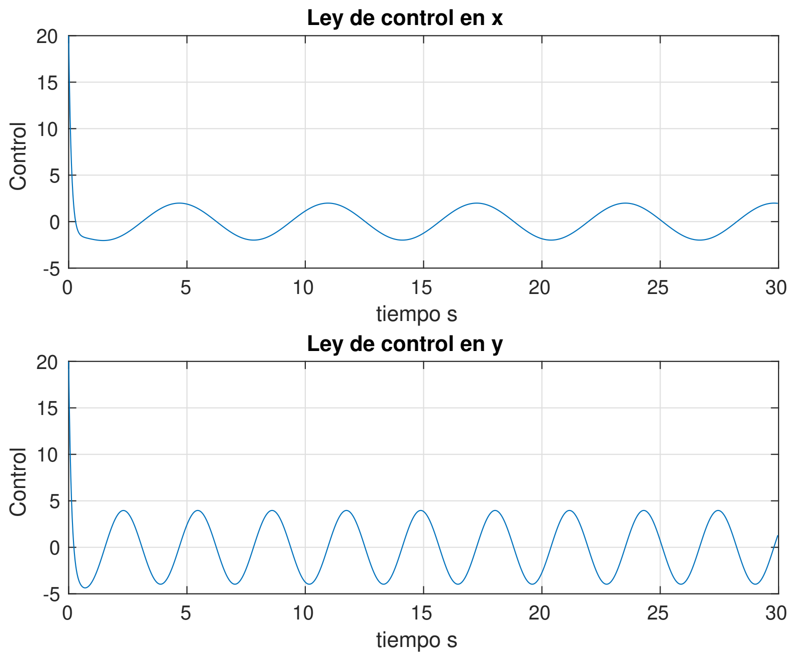
Los errores de posición en x y y, para todos los agentes:



Los errores de velocidad en x y y, para todos los agentes:

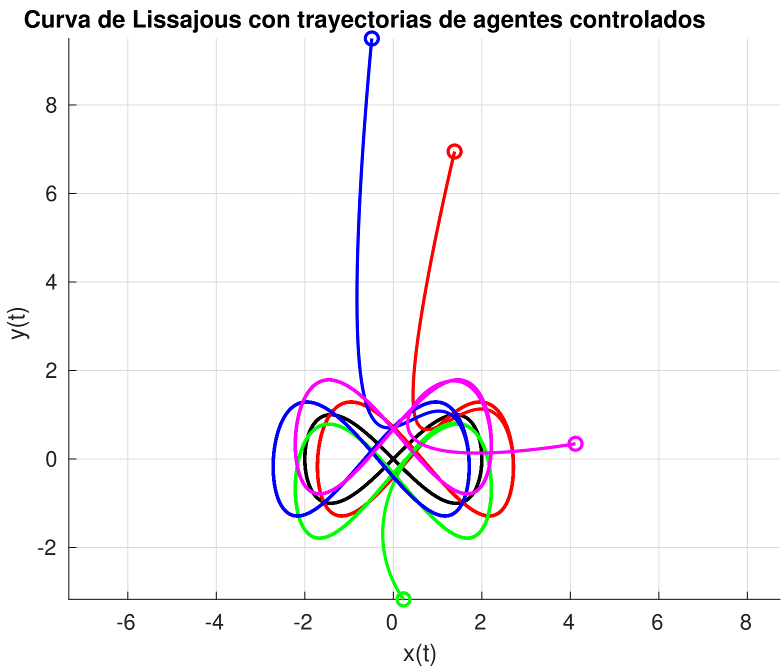


y finalmente las leyes de control para ambas coordenadas:

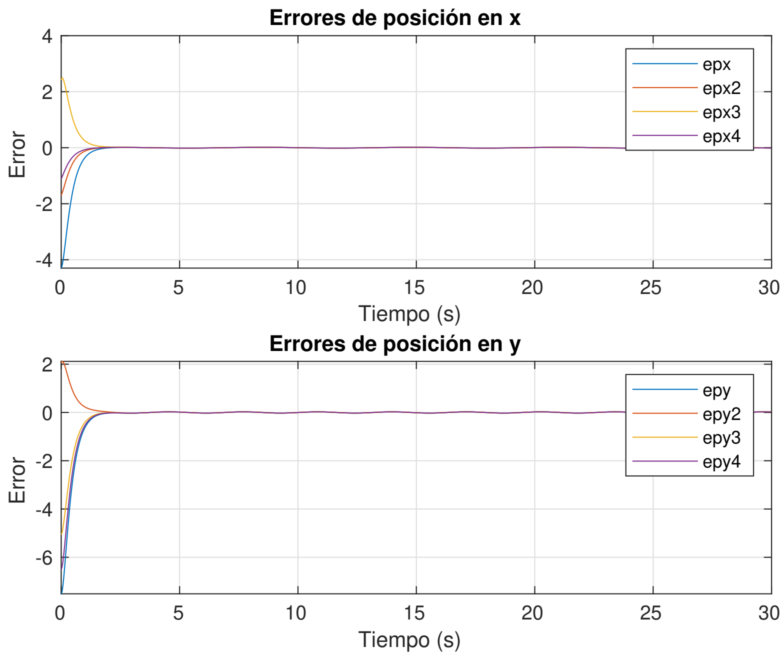


Pregunta 4.-

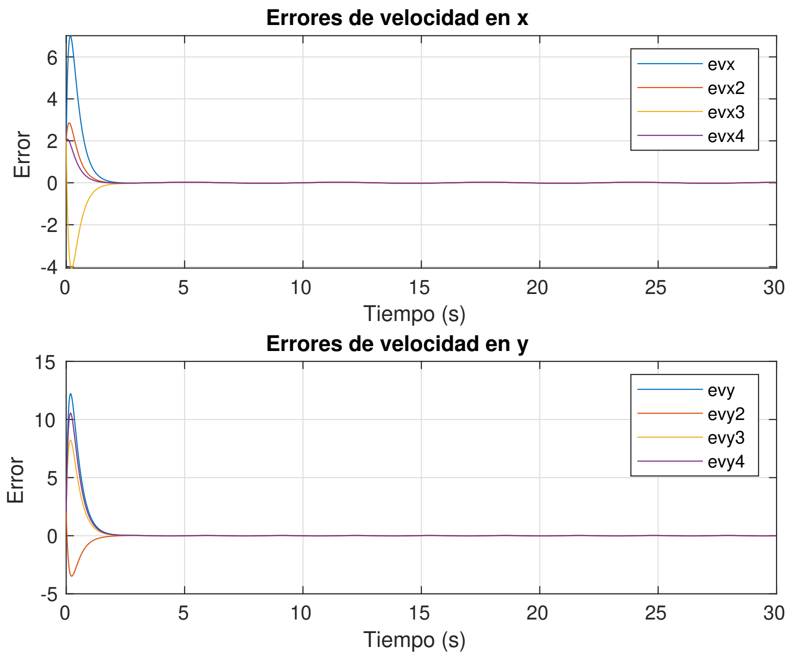
Repitiendo la simulación, pero para condiciones iniciales aleatorias y no partiendo en posición, obtenemos los siguientes resultados:



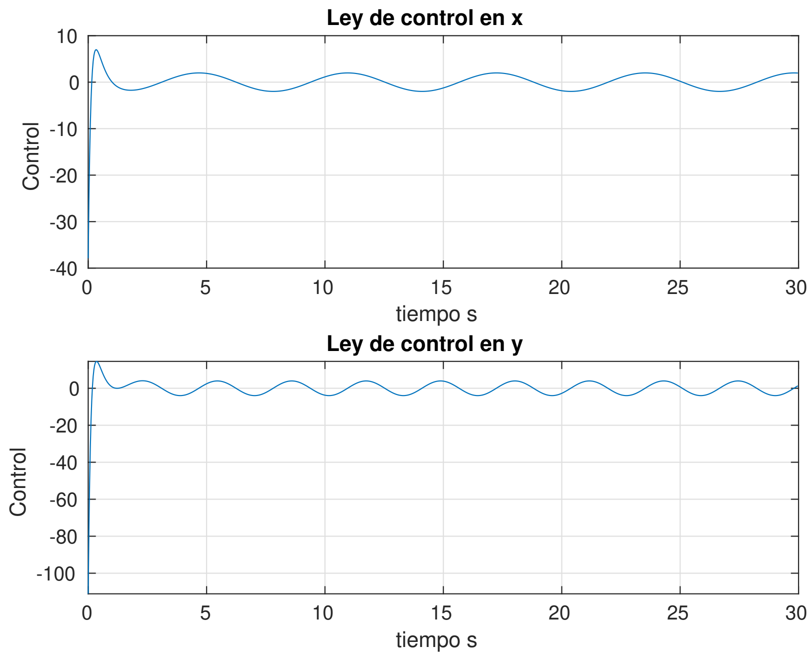
Los errores de posición en x y y, para todos los agentes:



Los errores de velocidad en x y y, para todos los agentes:



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)

```
clc
clearvars
close all
```

```

% Parametros de la curva de Lissajous
A = 2;          % Amplitud en x
B = 1;          % Amplitud en y
a = 1;          % Frecuencia en x
b = 2;          % Frecuencia en y
delta = 0;      % Diferencia de fase

AA=[0 0 1 0;
    0 0 0 1;
    0 0 0 0;
    0 0 0 0];

BB=[0 0;
    0 0;
    1 0;
    0 1];

% Tiempo
tiempo=30; %tiempo en segundos
t = linspace(0, tiempo, 1000); % Aumentar el rango de tiempo para la simulacion

% Ecuaciones parametricas de la curva de Lissajous
x_curve = A * sin(a * t + delta);
y_curve = B * sin(b * t);
%Derivada de la curva de Lissajous
vx_curve= A*a*cos(a*t+delta);
vy_curve= B*b*cos(b*t);
%Segunda derivada
ax_curve= - A* a^2 * sin(a * t +delta);
ay_curve= - B * b^2 * sin(b * t);
% Distancia desde la curva para los agentes
d = .5;

% Posiciones relativas de los agentes con respecto a la curva
%Primer punto
% x_offset = [d, 0, -d, 0];
% y_offset = [0, -d, 0, d];
%Segundo punto
x_offset = 5*[rand(), rand(), -rand(), rand()];
y_offset = 10*[rand(), -rand(), rand(), rand()];

% Parametros de control
k_a = 1;
k_v = 10;
k_p = 20;

% Inicializar posiciones y velocidades de los agentes
p = zeros(4, 1);
p2 = zeros(4, 1);
p3 = zeros(4, 1);
p4 = zeros(4, 1);

% Inicializar estados
%Primer punto
p(:,1)=[x_curve(1) + x_offset(1);y_curve(1) + y_offset(1);0;0];
```

```
p2(:,1)=[x_curve(1) + x_offset(2);y_curve(1) + y_offset(2);0;0];
p3(:,1)=[x_curve(1) + x_offset(3);y_curve(1) + y_offset(3);0;0];
p4(:,1)=[x_curve(1) + x_offset(4);y_curve(1) + y_offset(4);0;0];

% Simulacion
dt = t(2) - t(1);
for k = 1:length(t)-1
    % Posiciones deseadas en el siguiente instante de tiempo
    p_des = [x_curve(k)+d;y_curve(k)];
    p_des2 = [x_curve(k) ;y_curve(k)-d];
    p_des3 = [x_curve(k) - d;y_curve(k)];
    p_des4 = [x_curve(k) ;y_curve(k)+d];
    % Posicion actual
    p_real= p(1:2,k);
    p_real2= p2(1:2,k);
    p_real3= p3(1:2,k);
    p_real4= p4(1:2,k);
    %Error de posicion
    e_p = p_des-p_real;
    e_p2 = p_des2-p_real2;
    e_p3 = p_des3-p_real3;
    e_p4 = p_des4-p_real4;

    %Velocidad deseada
    v_des=[vx_curve(k+1);vy_curve(k+1)];
    v_des2=[vx_curve(k+1);vy_curve(k+1)];
    v_des3=[vx_curve(k+1);vy_curve(k+1)];
    v_des4=[vx_curve(k+1);vy_curve(k+1)];
    %Velocidad actual
    v_real=p(3:4,k);
    v_real2=p2(3:4,k);
    v_real3=p3(3:4,k);
    v_real4=p4(3:4,k);
    %Error de velocidad
    e_v = v_des-v_real;
    e_v2 = v_des2-v_real2;
    e_v3 = v_des3-v_real3;
    e_v4 = v_des4-v_real4;
    % e_v=-k_p*e_p;

    % Control de aceleracion
    u = [ax_curve(k);ay_curve(k)]+k_a * (k_v * e_v + k_p * e_p);
    u2 =[ax_curve(k);ay_curve(k)]+k_a * (k_v * e_v2 + k_p * e_p2);
    u3 =[ax_curve(k);ay_curve(k)]+k_a * (k_v * e_v3 + k_p * e_p3);
    u4 =[ax_curve(k);ay_curve(k)]+k_a * (k_v * e_v4 + k_p * e_p4);
    %Sistema
    p(:,k+1)=p(:,k)+(AA*p(:,k)+BB*u)*dt;
    p2(:,k+1)=p2(:,k)+(AA*p2(:,k)+BB*u2)*dt;
    p3(:,k+1)=p3(:,k)+(AA*p3(:,k)+BB*u3)*dt;
    p4(:,k+1)=p4(:,k)+(AA*p4(:,k)+BB*u4)*dt;

    % Control
    ux(:,k)=u(1);
    uy(:,k)=u(2);

    %Errores en posiciones
    epx(:,k)=e_p(1);
```

```
    epy(:,k)=e_p(2);
    epx2(:,k)=e_p2(1);
    epy2(:,k)=e_p2(2);
    epx3(:,k)=e_p3(1);
    epy3(:,k)=e_p3(2);
    epx4(:,k)=e_p4(1);
    epy4(:,k)=e_p4(2);

    %Errores en velocidad
    evx(:,k)=e_v(1);
    evy(:,k)=e_v(2);
    evx2(:,k)=e_v2(1);
    evy2(:,k)=e_v2(2);
    evx3(:,k)=e_v3(1);
    evy3(:,k)=e_v3(2);
    evx4(:,k)=e_v4(1);
    evy4(:,k)=e_v4(2);

end

% Graficar la curva de Lissajous y las trayectorias de los agentes
figure;
hold on;

% Graficar la curva de Lissajous
plot(x_curve, y_curve, 'k', 'LineWidth', 1.5);

% Colores para los agentes
colors = ['r', 'g', 'b', 'm'];

plot(p(1, :), p(2, :), 'Color', colors(1), 'LineWidth', 1.5);
plot(p2(1, :), p2(2, :), 'Color', colors(2), 'LineWidth', 1.5);
plot(p3(1, :), p3(2, :), 'Color', colors(3), 'LineWidth', 1.5);
plot(p4(1, :), p4(2, :), 'Color', colors(4), 'LineWidth', 1.5);

plot(p(1, 1), p(2, 1),'o', 'Color', colors(1), 'LineWidth', 1.5);
plot(p2(1, 1), p2(2, 1),'o', 'Color', colors(2), 'LineWidth', 1.5);
plot(p3(1, 1), p3(2, 1),'o', 'Color', colors(3), 'LineWidth', 1.5);
plot(p4(1, 1), p4(2, 1),'o', 'Color', colors(4), 'LineWidth', 1.5);

title('Curva de Lissajous con trayectorias de agentes controlados');
xlabel('x(t)');
ylabel('y(t)');
grid on;
axis equal;
hold off;

figure
subplot(2,1,1)
plot(t(1,1:size(t,2)-1),epx);
title('Errores de posicion en x')
ylabel('Error')
xlabel('tiempo s')
grid on
subplot(2,1,2)
plot(t(1,1:size(t,2)-1),epy);
```

```
title('Errores de posicion en y')
ylabel('Error')
xlabel('tiempo s')
grid on
%
%
figure
subplot(2,1,1)
plot(t(1,1:size(t,2)-1),evx);
title('Errores de velocidad en x')
ylabel('Error')
xlabel('tiempo s')
grid on
subplot(2,1,2)
plot(t(1,1:size(t,2)-1),evy);
title('Errores de velocidad en y')
ylabel('Error')
xlabel('tiempo s')
grid on


figure
subplot(2,1,1)
plot(t(1,1:size(t,2)-1),ux);
title('Ley de control en x')
ylabel('Control')
xlabel('tiempo s')
grid on
subplot(2,1,2)
plot(t(1,1:size(t,2)-1),uy);
title('Ley de control en y')
ylabel('Control')
xlabel('tiempo s')
grid on
```