

EJERCICIO 1

```
%Limpieza de pantalla
clear all
close all
clc
```

```
%1 TIEMPO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tf=31.4;           % Tiempo de simulación en segundos (s)
ts=0.008;          % Tiempo de muestreo en segundos (s)
t=0:ts:tf;         % Vector de tiempo
N= length(t);      % Muestras
```

```
%2 CONDICIONES INICIALES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%Damos valores a nuestro punto inicial de posición y orientación
x1(1)=0; %Posición inicial eje x
y1(1)=0; %Posición inicial eje y
phi(1)=0; %Orientación inicial del robot
```

```
%Igualamos el punto de control con las proyecciones X1 y Y1 por su
%coincidencia
hx(1)= x1(1);      % Posición del punto de control en el eje (X) metros (m)
hy(1)= y1(1);      % Posición del punto de control en el eje (Y) metros (m)
```

```
%3 TRAYECTORIA DESEADA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%Ecuaciones paramétricas de la trayectoria deseada
%a)Circulo
hxd=2*cos(0.2*t);
hyd= 2*sin(0.4*t);
```

```
%Velocidades de la trayectoria deseada
hxdp=-8*0.2*sin(0.2*t);
hydp=8*0.4*cos(0.4*t);
```

```
%4 CONTROL, BUCLE DE SIMULACION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
for k=1:N
```

```
    %a)Errores de control (Aqui la posición deseada ya no es constante,
    % varia con el tiempo)
    hxe(k)=hxd(k)-hx(k);
    hye(k)=hyd(k)-hy(k);
```

```
    %Matriz de error
    he= [hxe(k);hye(k)];
    %Magnitud del error de posición
    Error(k)= sqrt(hxe(k)^2 +hye(k)^2);
```

```
    %b)Matriz Jacobiana
```

```

J=[cos(phi(k)) -sin(phi(k));... %Matriz de rotación en 2D
    sin(phi(k)) cos(phi(k))];

%c)Matriz de Ganancias
K=[14.7 0;...
    0 18.2];

%d)Velocidades deseadas
hdp=[hxdp(k);hydp(k)];

%e)Ley de Control:Agregamos las velocidades deseadas
qpRef= pinv(J)*(hdp + K*he);

v(k)= qpRef(1);    %Velocidad lineal de entrada al robot
w(k)= qpRef(2);    %Velocidad angular de entrada al robot

%5 APLICACIÓN DE CONTROL AL ROBOT %%%%%%%%%%

    %Aplico la integral a la velocidad angular para obtener el angulo "phi" de la
orientación
    phi(k+1)=phi(k)+w(k)*ts; % Integral numérica (método de Euler)

%%%%%%%%% MODELO CINEMATICO %%%%%%%%%%

    xp1=v(k)*cos(phi(k));
    yp1=v(k)*sin(phi(k));

    %Aplico la integral a la velocidad lineal para obtener las cordenadas
    %"x1" y "y1" de la posición
    x1(k+1)=x1(k)+ ts*xp1; % Integral numérica (método de Euler)
    y1(k+1)=y1(k)+ ts*yp1; % Integral numérica (método de Euler)

    % Posicion del robot con respecto al punto de control
    hx(k+1)=x1(k+1);
    hy(k+1)=y1(k+1);

end

%%%%%%%%% SIMULACION VIRTUAL 3D %%%%%%%%%%

% a) Configuracion de escena

scene=figure; % Crear figura (Escena)
set(scene,'Color','white'); % Color del fondo de la escena
set(gca,'FontWeight','bold') ;% Negrilla en los ejes y etiquetas
sizeScreen=get(0,'ScreenSize'); % Retorna el tamaño de la pantalla del computador
set(scene,'position',sizeScreen); % Configurar tamaño de la figura
camlight('headlight'); % Luz para la escena
axis equal; % Establece la relación de aspecto para que las unidades de datos sean
las mismas en todas las direcciones.
grid on; % Mostrar líneas de cuadrícula en los ejes
box on; % Mostrar contorno de ejes
xlabel('x(m)'); ylabel('y(m)'); zlabel('z(m)'); % Etiqueta de los eje

```

```

view([-0.1 90]); % Orientacion de la figura
axis([-10 10 -10 10 0 1]); % Ingresar limites minimos y maximos en los ejes x y z
[minX maxX minY maxY minZ maxZ]

% b) Graficar robots en la posicion inicial
scale = 4;
MobileRobot_5;
H1=MobilePlot_4(x1(1),y1(1),phi(1),scale);hold on;

% c) Graficar Trayectorias
H2=plot3(hx(1),hy(1),0,'r','lineWidth',2);
H3=plot3(hxd,hyd,zeros(1,N),'g','lineWidth',2); %Grafico circulo en posición deseada
%H4=plot3(hx(1),hy(1),0,'go','lineWidth',2);%Grafico circulo en posición inicial
% d) Bucle de simulacion de movimiento del robot

step=1; % pasos para simulacion

for k=1:step:N

    delete(H1);
    delete(H2);

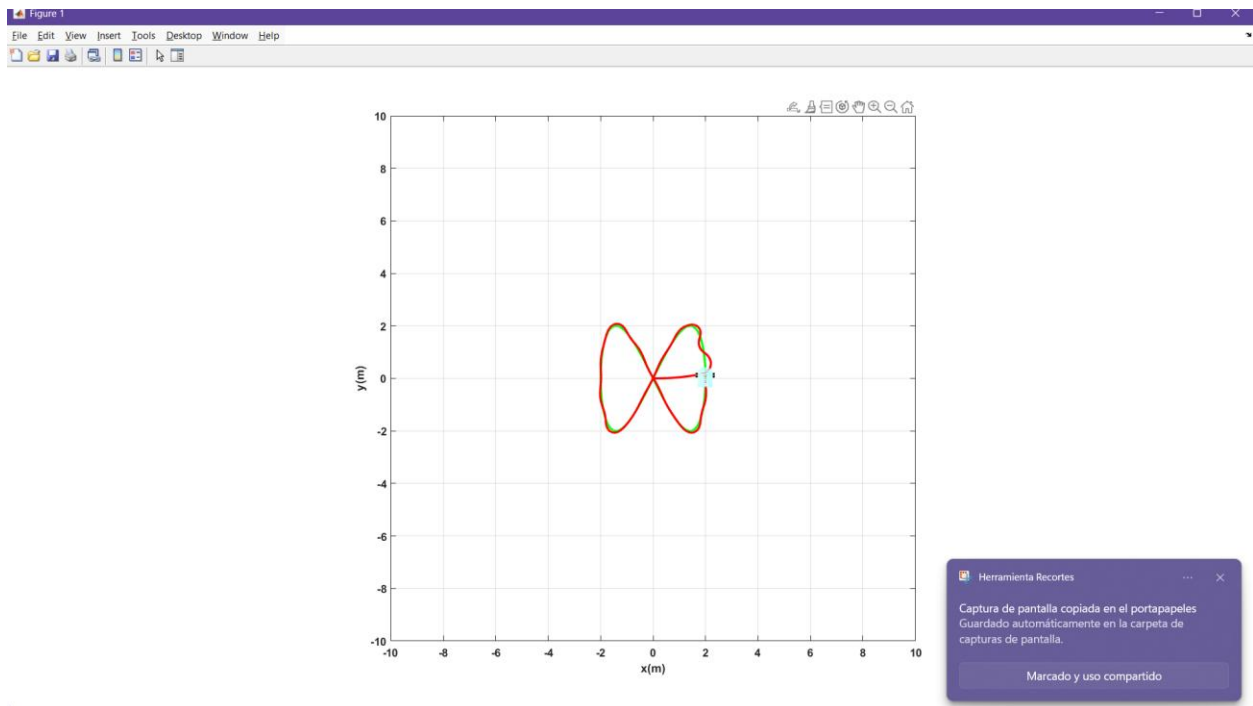
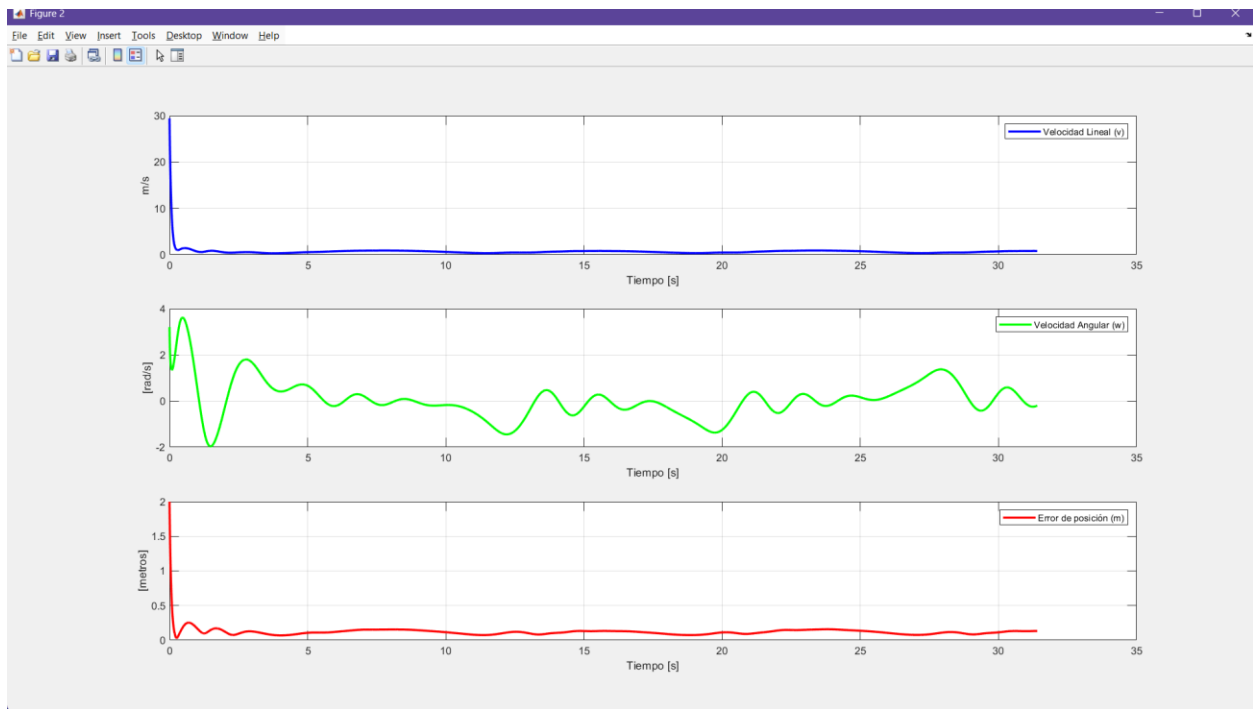
    H1=MobilePlot_4(x1(k),y1(k),phi(k),scale);
    H2=plot3(hx(1:k),hy(1:k),zeros(1,k),'r','lineWidth',2);

    pause(ts);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Graficas %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
graph=figure; % Crear figura (Escena)
set(graph,'position',sizeScreen); % Congigurar tamaño de la figura
subplot(311)
plot(t,v,'b','LineWidth',2),grid('on'),xlabel('Tiempo
[s]'),ylabel('m/s'),legend('Velocidad Lineal (v)');
subplot(312)
plot(t,w,'g','LineWidth',2),grid('on'),xlabel('Tiempo
[s]'),ylabel('rad/s'),legend('Velocidad Angular (w)');
subplot(313)
plot(t>Error,'r','LineWidth',2),grid('on'),xlabel('Tiempo
[s]'),ylabel('metros'),legend('Error de posición (m)');

```



EJERCICIO 2

```
%Limpieza de pantalla
clear all
close all
clc
```

```
%1 TIEMPO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tf=20;          % Tiempo de simulación en segundos (s)
ts=0.004;       % Tiempo de muestreo en segundos (s)
t=0:ts:tf;      % Vector de tiempo
N= length(t);   % Muestras
```

```
%2 CONDICIONES INICIALES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%Damos valores a nuestro punto inicial de posición y orientación
x1(1)=0; %Posición inicial eje x
y1(1)=0; %Posición inicial eje y
phi(1)=0; %Orientación inicial del robot
```

```
%Igualamos el punto de control con las proyecciones X1 y Y1 por su
%coincidencia
hx(1)= x1(1); % Posición del punto de control en el eje (X) metros (m)
hy(1)= y1(1); % Posición del punto de control en el eje (Y) metros (m)
```

```
%3 TRAYECTORIA DESEADA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%Ecuaciones paramétricas de la trayectoria deseada
%a)Circulo
hxd=t-(3*sin(t));
hyd= 4-3*cos(t);
```

```
%Velocidades de la trayectoria deseada
hxdp=-8*0.2*sin(0.2*t);
hydp=8*0.4*cos(0.4*t);
```

```
%4 CONTROL, BUCLE DE SIMULACION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
for k=1:N
```

```
    %a)Errores de control (Aqui la posición deseada ya no es constante,
    % varia con el tiempo)
    hxe(k)=hxd(k)-hx(k);
    hye(k)=hyd(k)-hy(k);
```

```
    %Matriz de error
    he= [hxe(k);hye(k)];
    %Magnitud del error de posición
    Error(k)= sqrt(hxe(k)^2 +hye(k)^2);
```

```
    %b)Matriz Jacobiana
```

```

J=[cos(phi(k)) -sin(phi(k));... %Matriz de rotación en 2D
    sin(phi(k)) cos(phi(k))];

%c)Matriz de Ganancias
K=[55 0;...
    0 47];

%d)Velocidades deseadas
hdp=[hxdp(k);hydp(k)];

%e)Ley de Control:Agregamos las velocidades deseadas
qpRef= pinv(J)*(hdp + K*he);

v(k)= qpRef(1);    %Velocidad lineal de entrada al robot
w(k)= qpRef(2);    %Velocidad angular de entrada al robot

%5 APLICACIÓN DE CONTROL AL ROBOT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %Aplico la integral a la velocidad angular para obtener el angulo "phi" de la
orientación
    phi(k+1)=phi(k)+w(k)*ts; % Integral numérica (método de Euler)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MODELO CINEMATICO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    xp1=v(k)*cos(phi(k));
    yp1=v(k)*sin(phi(k));

    %Aplico la integral a la velocidad lineal para obtener las cordenadas
    %"x1" y "y1" de la posición
    x1(k+1)=x1(k)+ ts*xp1; % Integral numérica (método de Euler)
    y1(k+1)=y1(k)+ ts*yp1; % Integral numérica (método de Euler)

    % Posicion del robot con respecto al punto de control
    hx(k+1)=x1(k+1);
    hy(k+1)=y1(k+1);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SIMULACION VIRTUAL 3D %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% a) Configuracion de escena

scene=figure; % Crear figura (Escena)
set(scene,'Color','white'); % Color del fondo de la escena
set(gca,'FontWeight','bold') ;% Negrilla en los ejes y etiquetas
sizeScreen=get(0,'ScreenSize'); % Retorna el tamaño de la pantalla del computador
set(scene,'position',sizeScreen); % Configurar tamaño de la figura
camlight('headlight'); % Luz para la escena
axis equal; % Establece la relación de aspecto para que las unidades de datos sean
las mismas en todas las direcciones.
grid on; % Mostrar líneas de cuadrícula en los ejes
box on; % Mostrar contorno de ejes
xlabel('x(m)'); ylabel('y(m)'); zlabel('z(m)'); % Etiqueta de los eje

```

```

view([-0.1 90]); % Orientacion de la figura
axis([-5 22 -10 10 0 1]); % Ingresar limites minimos y maximos en los ejes x y z
[minX maxX minY maxY minZ maxZ]

% b) Graficar robots en la posicion inicial
scale = 4;
MobileRobot_5;
H1=MobilePlot_4(x1(1),y1(1),phi(1),scale);hold on;

% c) Graficar Trayectorias
H2=plot3(hx(1),hy(1),0,'r','lineWidth',2);
H3=plot3(hxd,hyd,zeros(1,N),'g','lineWidth',2); %Grafico circulo en posición deseada
%H4=plot3(hx(1),hy(1),0,'go','lineWidth',2);%Grafico circulo en posición inicial
% d) Bucle de simulacion de movimiento del robot

step=50; % pasos para simulacion

for k=1:step:N

    delete(H1);
    delete(H2);

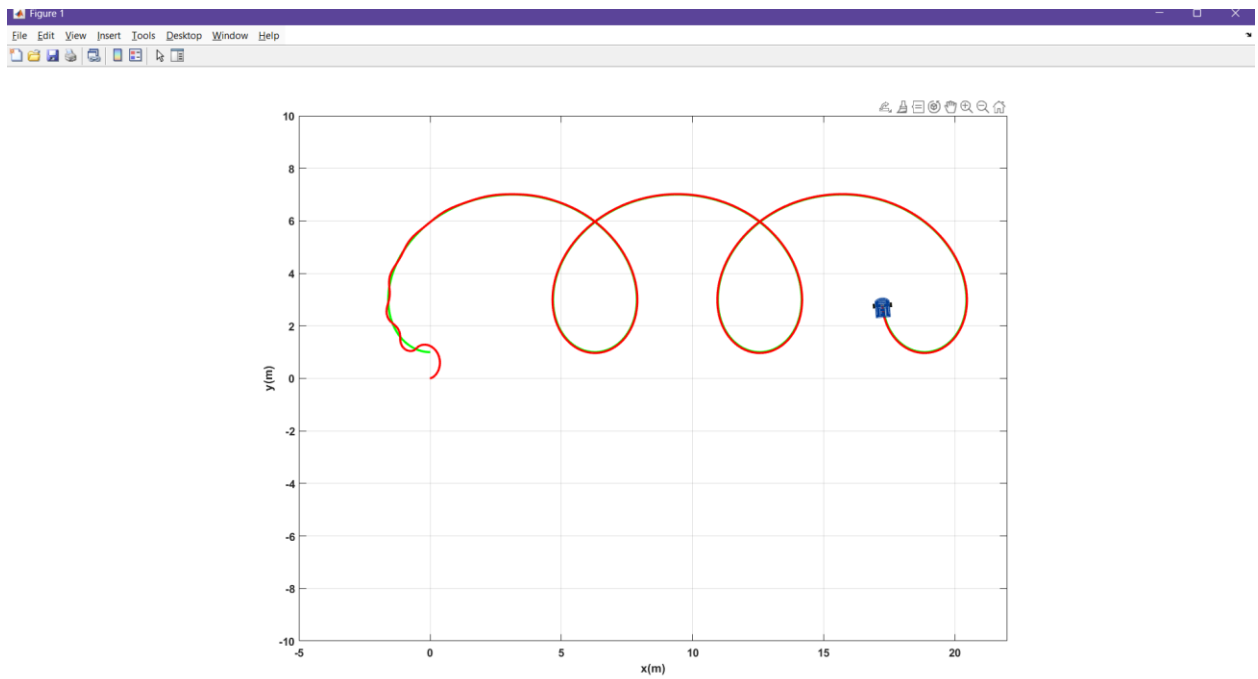
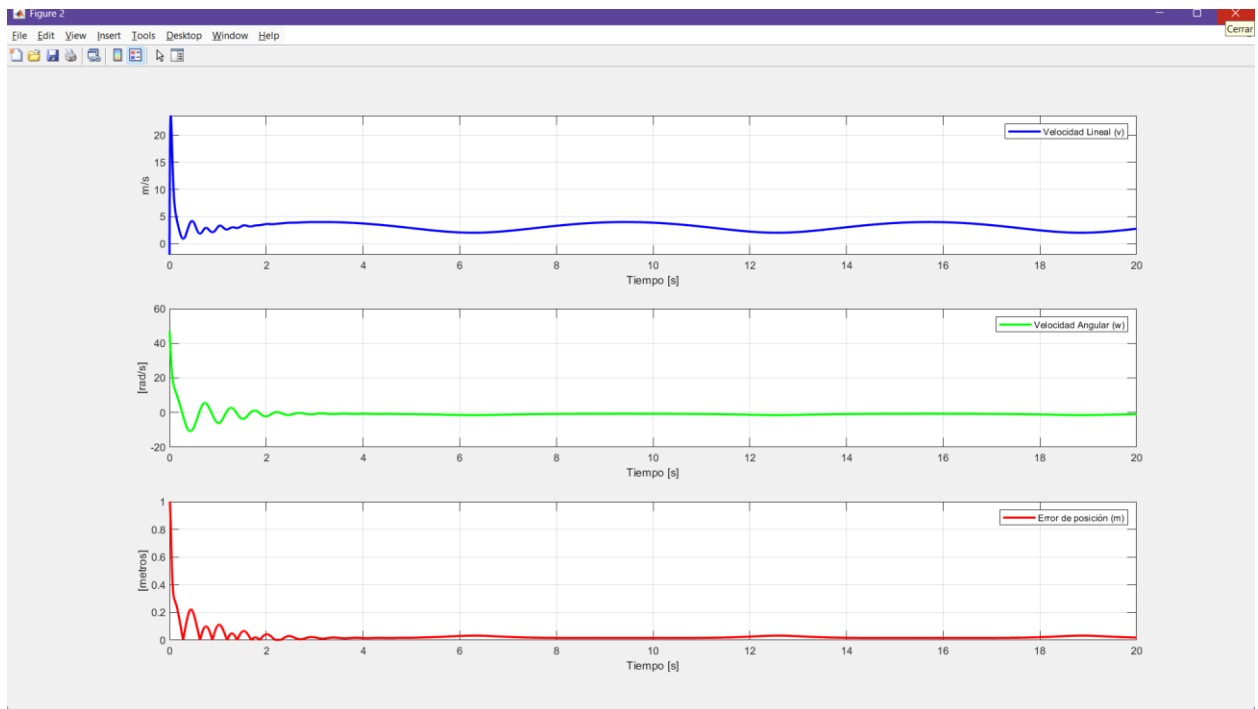
    H1=MobilePlot_4(x1(k),y1(k),phi(k),scale);
    H2=plot3(hx(1:k),hy(1:k),zeros(1,k),'r','lineWidth',2);

    pause(ts);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Graficas %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
graph=figure; % Crear figura (Escena)
set(graph,'position',sizeScreen); % Congigurar tamaño de la figura
subplot(311)
plot(t,v,'b','LineWidth',2),grid('on'),xlabel('Tiempo
[s]'),ylabel('m/s'),legend('Velocidad Lineal (v)');
subplot(312)
plot(t,w,'g','LineWidth',2),grid('on'),xlabel('Tiempo
[s]'),ylabel('rad/s'),legend('Velocidad Angular (w)');
subplot(313)
plot(t>Error,'r','LineWidth',2),grid('on'),xlabel('Tiempo
[s]'),ylabel('metros'),legend('Error de posición (m)');

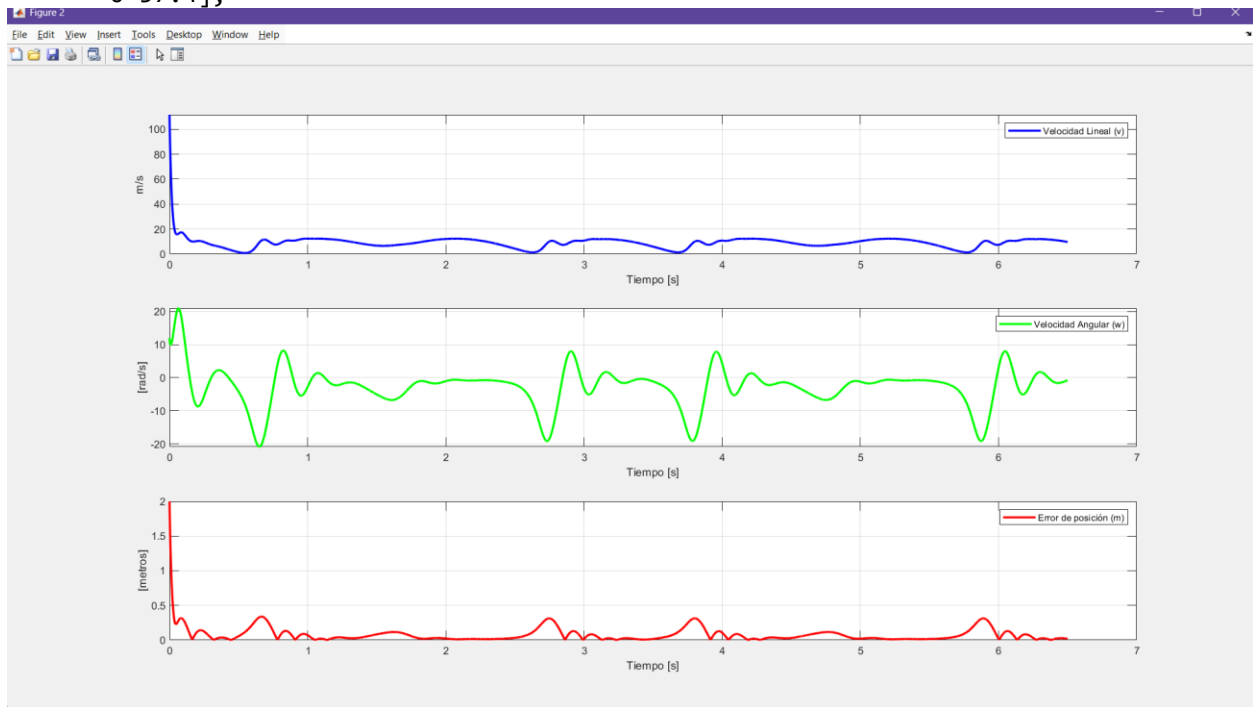
```

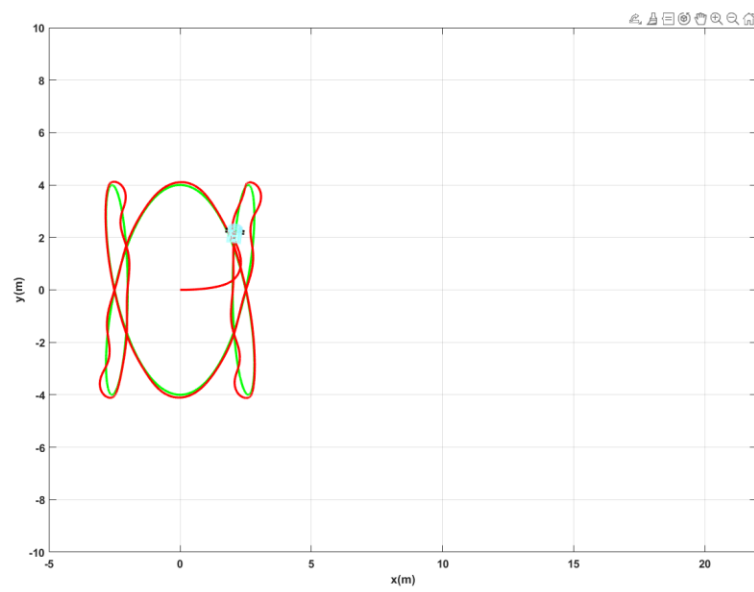
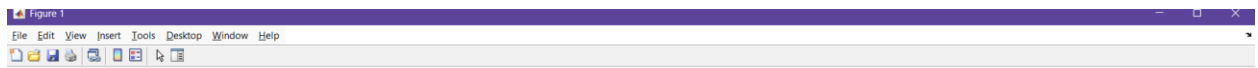


EJERCICIO 3

A partir de aquí solo cambiamos t_s , el tiempo de simulación y las ganancias. Entonces solo estarán presentes dichos cambios para no hacer agobiante el reporte.

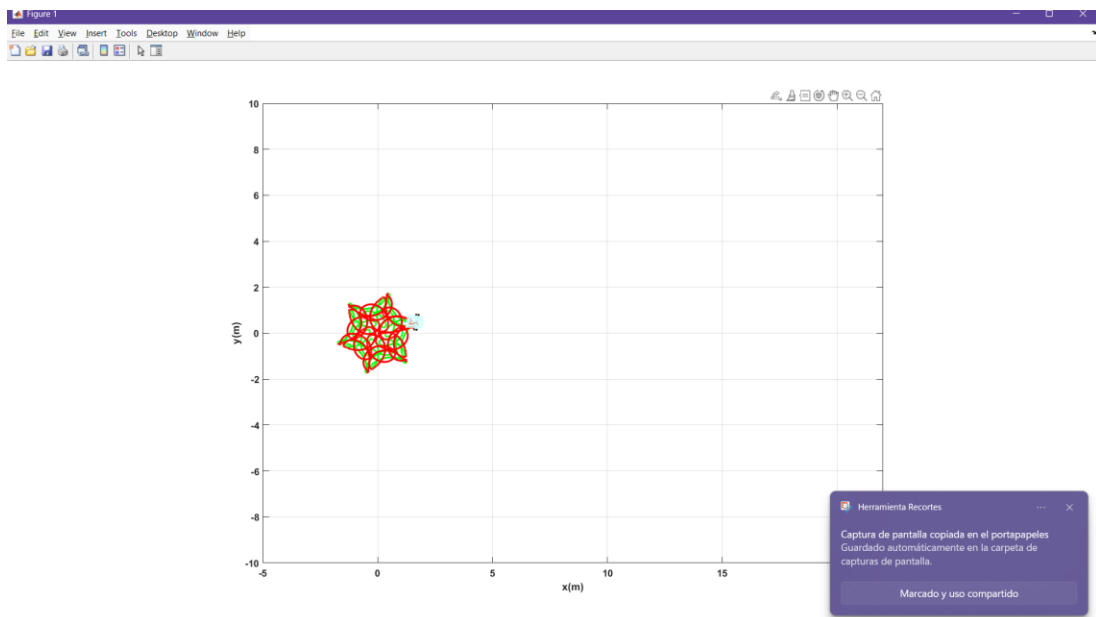
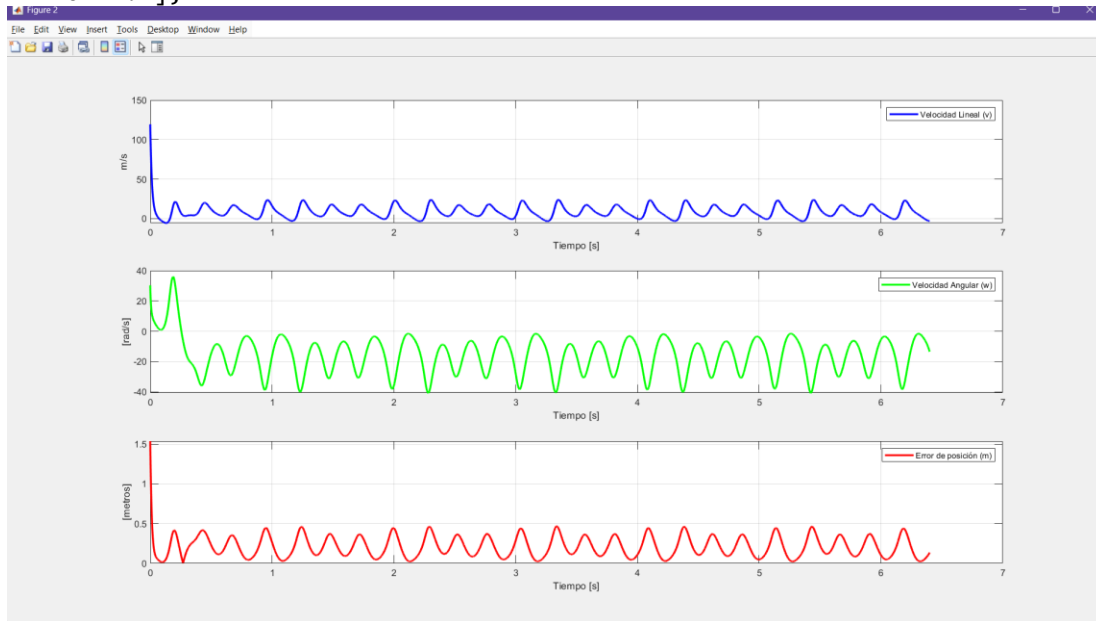
```
%1 TIEMPO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tf=6.5;           % Tiempo de simulación en segundos (s)
ts=0.0006;        % Tiempo de muestreo en segundos (s)
t=0:ts:tf;         % Vector de tiempo
N= length(t);      % Mues
K=[55.8 0;...
  0 57.4];
```





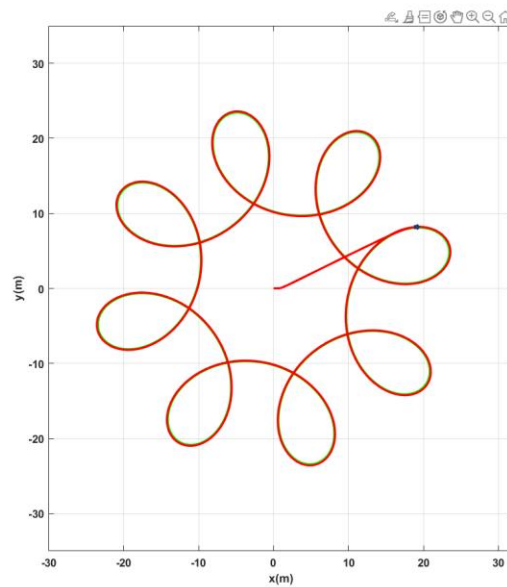
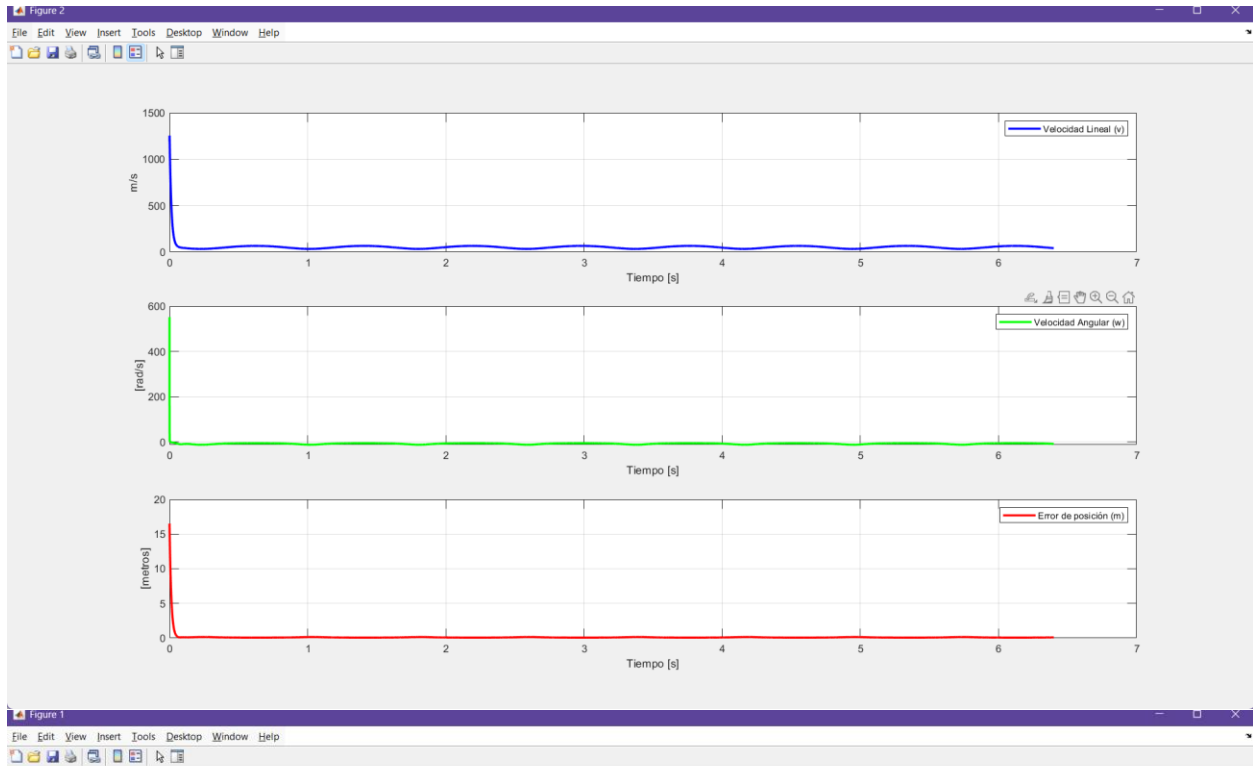
EJERCICIO 4

```
tf=6.4;           % Tiempo de simulación en segundos (s)
ts=0.0003;        % Tiempo de muestreo en segundos (s)
t=0:ts:tf;         % Vector de tiempo
N= length(t);      % Muestras
%c)Matriz de Ganancias
K=[75.8 0;...
  0 77.4];
```



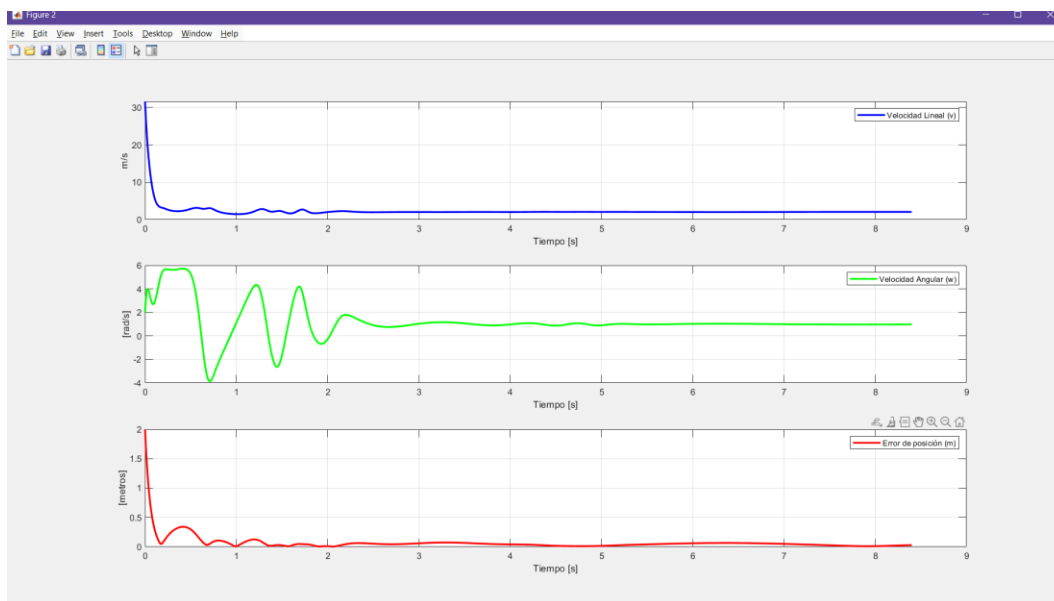
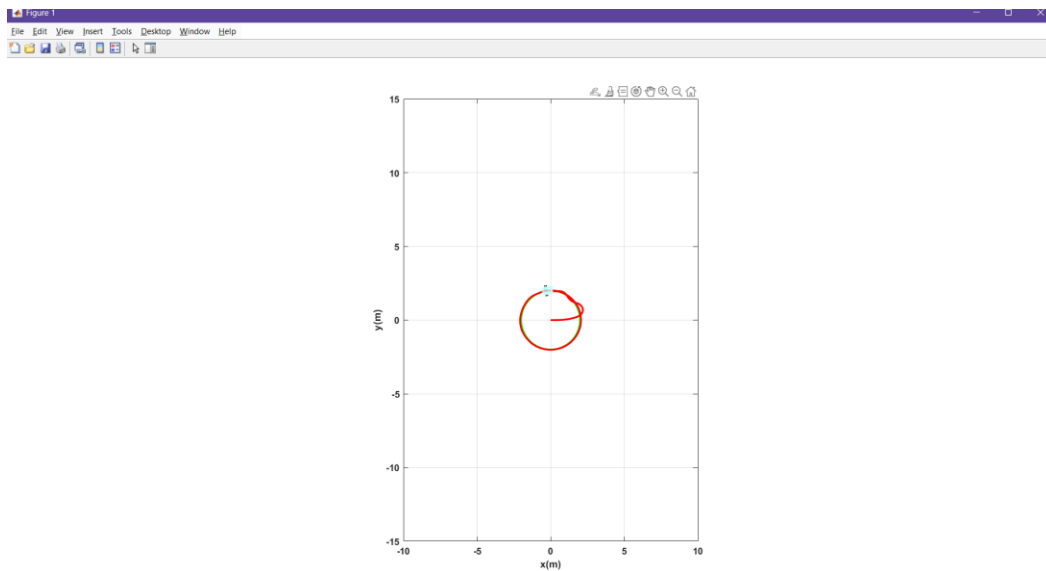
EJERCICIO 5

```
tf=6.4;           % Tiempo de simulación en segundos (s)
ts=0.0007;        % Tiempo de muestreo en segundos (s)
t=0:ts:tf;         % Vector de tiempo
N= length(t);      % Muestras
K=[75.8 0;...
  0 77.4];
```



EJERCICIO 6

```
%1 TIEMPO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tf=8.4;           % Tiempo de simulación en segundos (s)
ts=0.01;          % Tiempo de muestreo en segundos (s)
t=0:ts:tf;        % Vector de tiempo
N= length(t);     % Muestras
%c)Matriz de Ganancias
K=[15.8 0;...
  0 87.4];
Error(k)= sqrt(hxe(k)^2 +hye(k)^2);
```

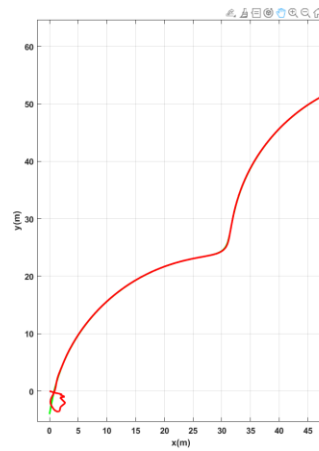
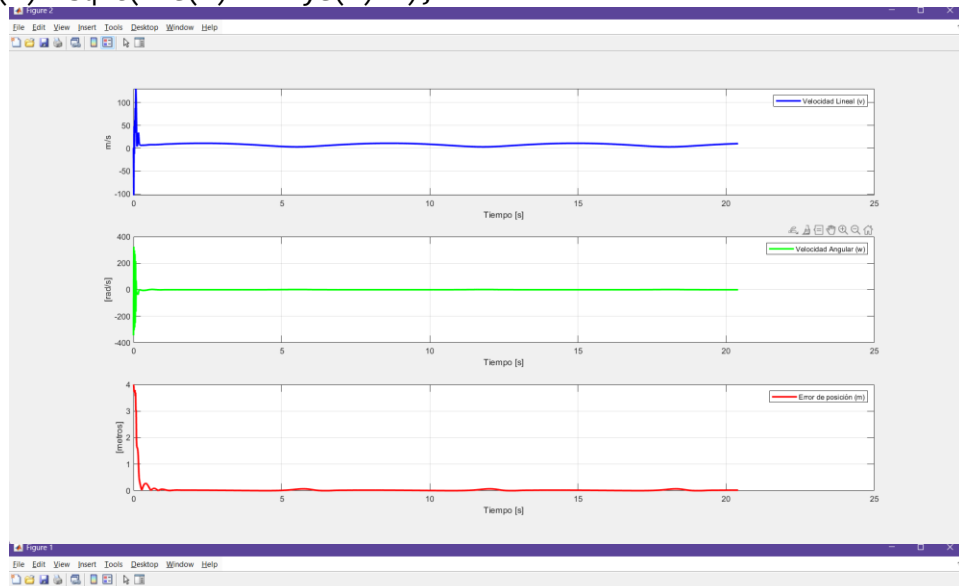


EJERCICIO 7

```

tf=20.4;           % Tiempo de simulación en segundos (s)
ts=0.01;           % Tiempo de muestreo en segundos (s)
t=0:ts:tf;         % Vector de tiempo
N= length(t);      % Muestras
%c)Matriz de Ganancias
K=[15.8 0;...
  0 87.4];
Error(k)= sqrt(hxe(k)^2 +hye(k)^2);

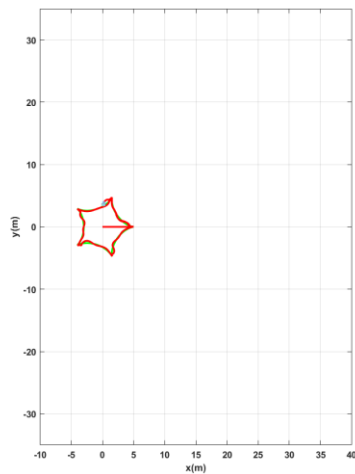
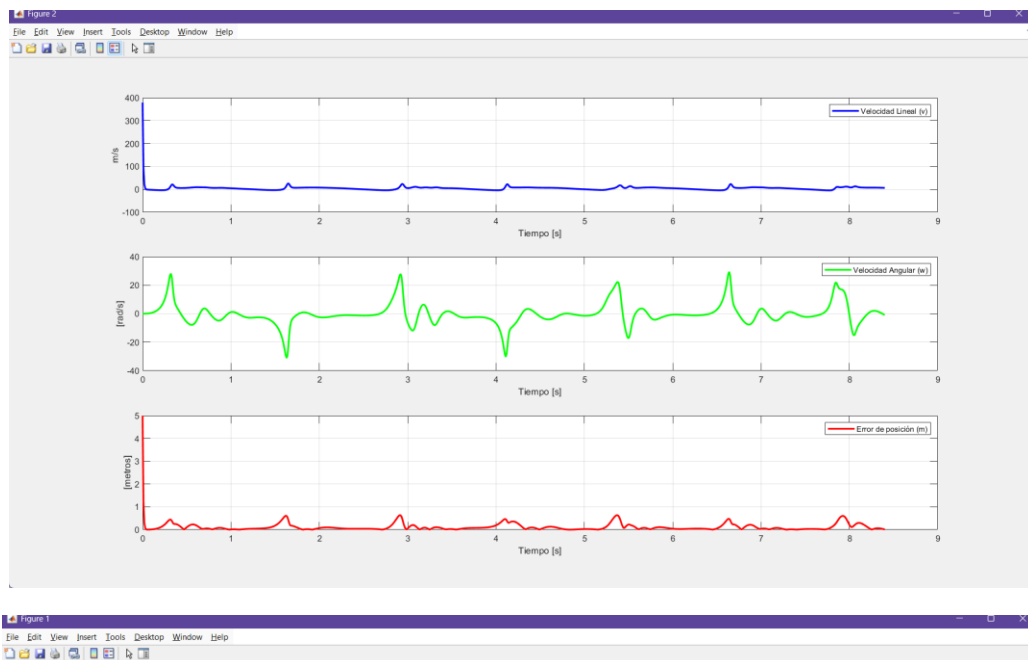
```



EJERCICIO 8

```
%1 TIEMPO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tf=8.4;           % Tiempo de simulación en segundos (s)
ts=0.01;          % Tiempo de muestreo en segundos (s)
t=0:ts:tf;        % Vector de tiempo
N= length(t);     % Muestras

%c)Matriz de Ganancias
K=[75.8 0;...
  0 25.4];
```



EJERCICIO 9

Para este ejercicio ya que estamos realizando un hexágono con puntos específicos no podemos simplemente trazar una trayectoria para x e y, entonces establecemos el numero de lados, el radio y el angulo de los vértices.

```
% 3. TRAYECTORIA: HEXÁGONO REGULAR
n = 6; % Lados del polígono
r = 1; % Radio del hexágono
theta = linspace(0, 2*pi, n+1); % Ángulos de vértices

% Coordenadas de los vértices
xv = r * cos(theta);
yv = r * sin(theta);

% Muestras por lado
samples_per_side = floor(N / n);

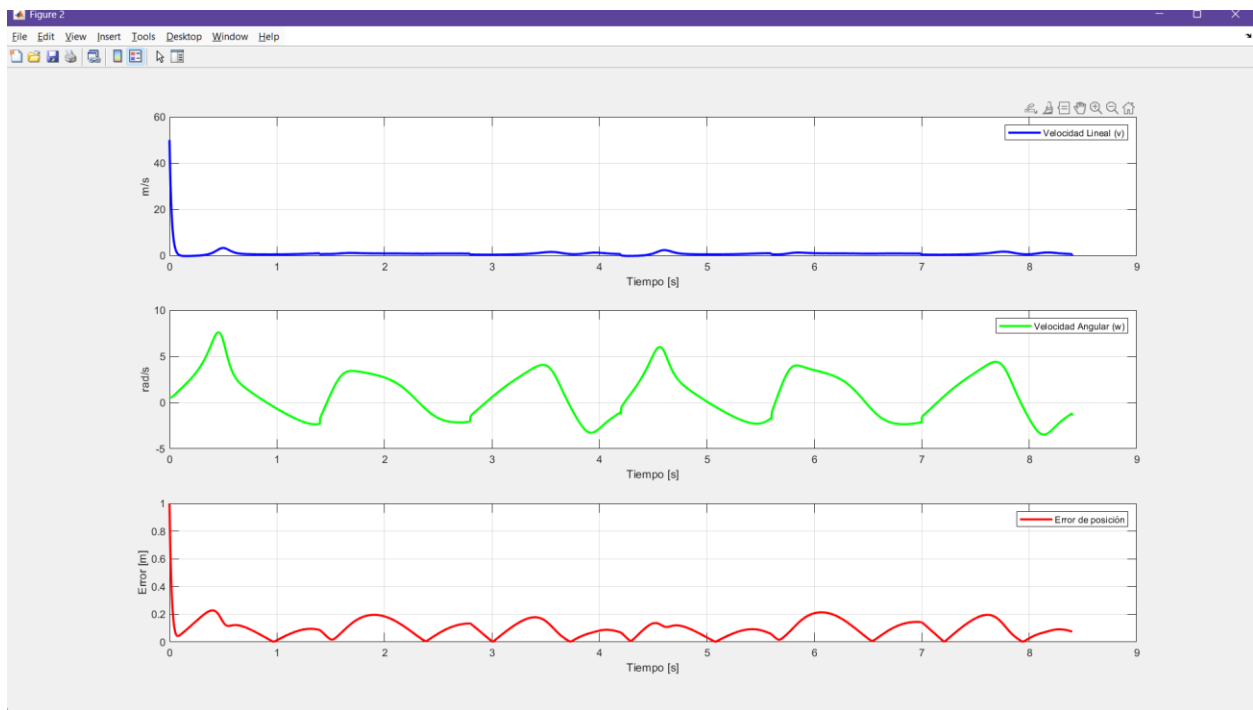
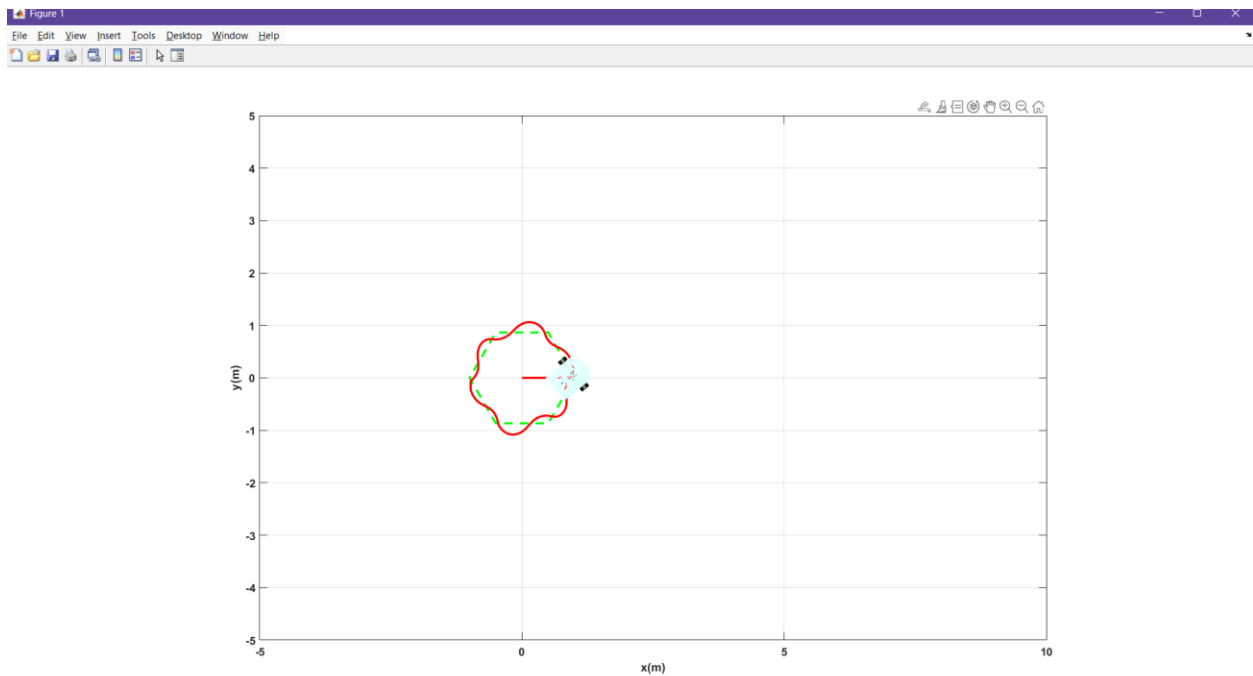
% Inicialización
hxd = [];
hyd = [];

% Interpolación de tramos
for i = 1:n
    hxd = [hxd, linspace(xv(i), xv(i+1), samples_per_side)];
    hyd = [hyd, linspace(yv(i), yv(i+1), samples_per_side)];
end

% Ajuste a longitud N
hxd = [hxd, repmat(hxd(end), 1, N - length(hxd))];
hyd = [hyd, repmat(hyd(end), 1, N - length(hyd))];

% Derivadas numéricas (velocidades deseadas)
hxdp = [diff(hxd), 0] / ts;
hydp = [diff(hyd), 0] / ts;

% 4. BUCLE DE CONTROL
K = [50.8 0; 0 15.4];
```

EJERCICIO 10

```
% --- Trayectoria tipo FLOR POLAR ---
```

```
A = 50;
```

```
k = 5;
```

```
% Trayectoria
```

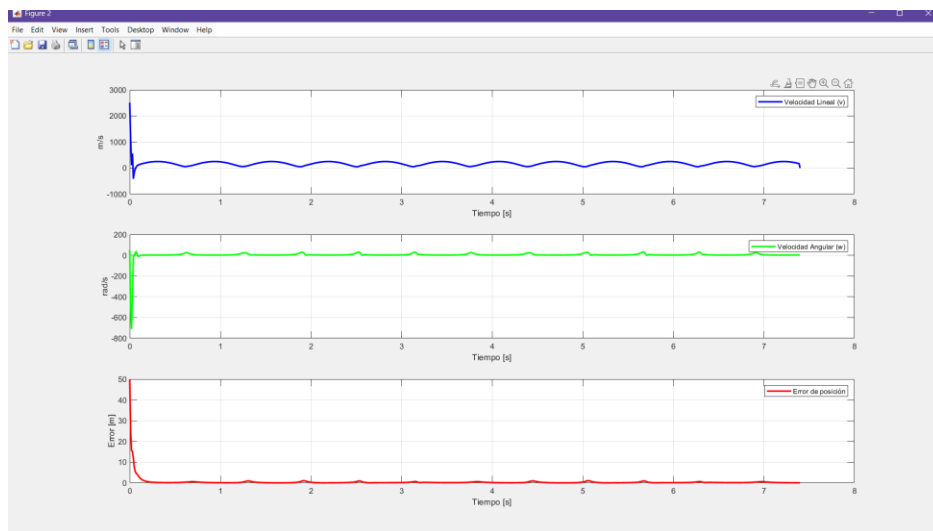
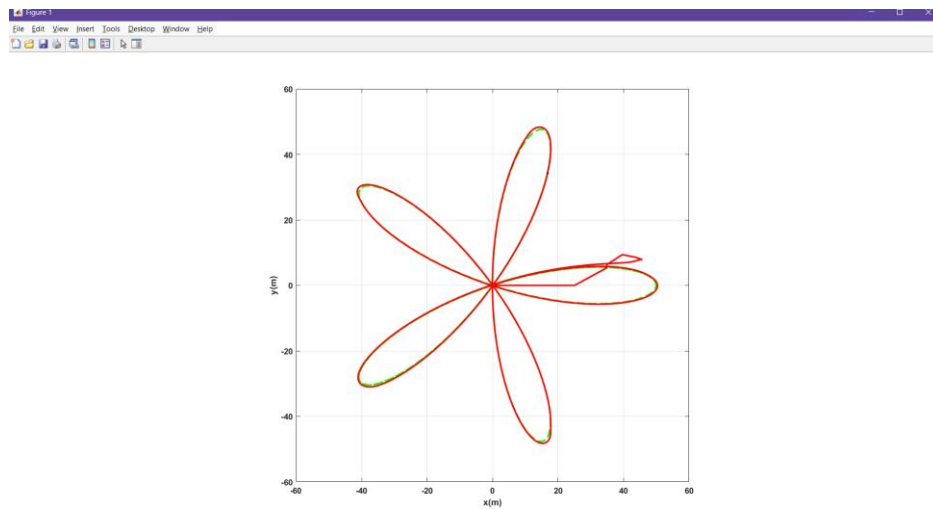
```
hxd = A * cos(k * t) .* cos(t);
```

```
hyd = A * cos(k * t) .* sin(t);
```

```
% Derivadas numéricas (velocidades deseadas)
```

```
hxdp = [diff(hxd), 0] / ts;
```

```
hydp = [diff(hyd), 0] / ts;
```



EJERCICIO 11

```
% Redefinir vector de tiempo solo para 1 ciclo de 0 a 2pi  
t = linspace(0, 2*pi, N);
```

```
% Ecuaciones paramétricas del corazón
```

```
hxd = 16 * sin(t).^3;
```

```
hyd = 13*cos(t) - 5*cos(2*t) - 2*cos(3*t) - cos(4*t);
```

```
% Derivadas numéricas (velocidades deseadas)
```

```
hxdp = [diff(hxd), 0] / ts;
```

```
hydp = [diff(hyd), 0] / ts;
```

```
K = [50.3 0; 0 15.4];
```

