Actividad 1.5 (Evaluación)

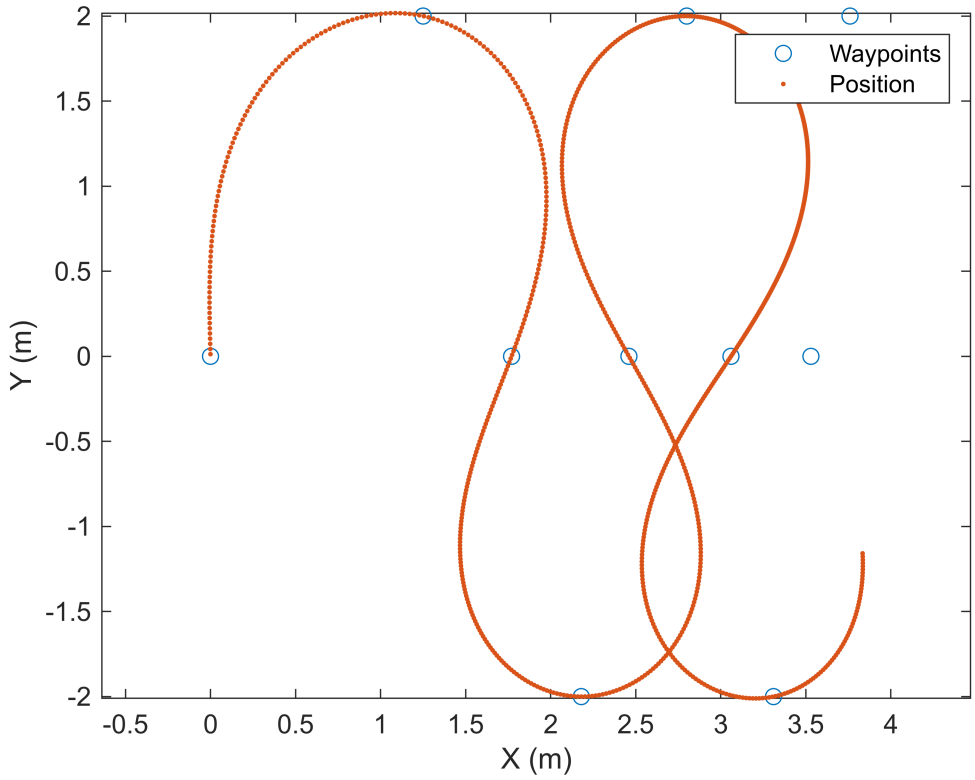
Alan Iván Flores Juárez | A01736001

Ejercicio 1

No se logró obtener una trayectoria que cumpliera con f(x)=2sin(x^2)

Mejor aproximación de la trayectoria solicitada

```
waypoints1 = [0     0  0
              1.25  2  0
              1.77  0  0
              2.18 -2  0
              2.46  0  0
              2.8   2  0
              3.06  0  0
              3.31 -2  0
              3.53  0  0
              3.76  2  0];
timeOfArrival1 = [2.6; 5.4; 8;11.5; 15;18.5; 24.3; 28.3; 32; 36];

traj1 = waypointTrajectory(waypoints1,timeOfArrival1);
sampleTimes1 = linspace(0,30,1000);
[position1, orientation1, velocity1] = lookupPose(traj1,sampleTimes1);
plot(waypoints1(:,1),waypoints1(:,2),"o", ...
     position1(:,1), position1(:,2),".");
xlabel("X (m)");
ylabel("Y (m)");
axis equal
legend({"Waypoints","Position"})
```

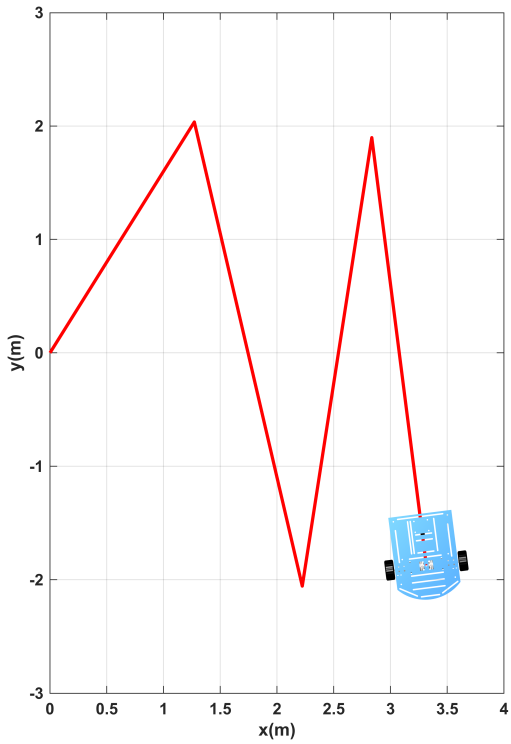



```
clear
close all
clc
load Uniciclo_6.mat

%%%%%%%%%%%%%%%%%%%%%%%%%%%%% COORDENADAS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
coordenadas = [0,0 ; 1.25,2 ; 2.18,-2 ; 2.8,2 ; 3.31,-2]; %Trazo de un cuadrado

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% TIEMPO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ts = 0.1;            % Tiempo de muestreo en segundos (s)
velocidad_lineal = 2; %Velocidad de 2m/s

r = zeros(1,size(coordenadas, 1));
radLocal = zeros(1,size(coordenadas, 1));
radGlobal = zeros(1,size(coordenadas, 1));
side_length_local = zeros(1,size(coordenadas, 1));
side_length_global = zeros(1,size(coordenadas, 1));

for j = 1:size(coordenadas, 1)-1 %Para cada coordenada - 1
    punto1 = coordenadas(j, :);
    punto2 = coordenadas(j+1, :);
    r(j) = sqrt((punto2(1)-punto1(1))^2 + (punto2(2)-punto1(2))^2);%Se calcula la
distancia
    radLocal(j) = atan2(punto2(2)-punto1(2), punto2(1)-punto1(1)); % Calcula el
ángulo en radianes
```

```
    side_length_local(j) = round(r(j)/ts/velocidad_lineal) + 1; %Se calcula la
longitud local
    if j==1
        radGlobal(j) = atan2(punto2(2)-punto1(2), punto2(1)-punto1(1)); % Calcula
el ángulo en radianes
        side_length_global(j) = side_length_local(j);
    else
        radGlobal(j) = atan2(punto2(2)-punto1(2), punto2(1)-punto1(1))-
radLocal(j-1); % Calcula el ángulo en radianes
        side_length_global(j) = side_length_local(j)+side_length_global(j-1);
    end

end


distancia_total = 0;
for j=1:size(coordenadas, 1)-1
    distancia_total = r(j) + distancia_total;
end

tiempo_recorrido = distancia_total/velocidad_lineal+((size(coordenadas, 1)-1)*ts);
%Tiempo de recorrido

tf = tiempo_recorrido;             % Tiempo de simulacion en segundos (s)
t = 0: ts: tf;       % Vector de tiempo
N = length(t);       % Muestras

%%%%%%%%%%%%%%%%%%%%%%%% CONDICIONES INICIALES %%%%%%%%%%%%%%%%%%%%%%%%%%

x1 = zeros (1,N+1);  % Posición en el centro del eje que une las ruedas (eje x) en
metros (m)
y1 = zeros (1,N+1);  % Posición en el centro del eje que une las ruedas (eje y) en
metros (m)
phi = zeros(1, N+1); % Orientacion del robot en radianes (rad)

x1(1) = 0;    % Posicion inicial eje x
y1(1) = 0;   % Posicion inicial eje y
phi(1) = 0;   % Orientacion inicial del robot

%%%%%%%%%%%%%%%%%%%%%%%%%%%% PUNTO DE CONTROL %%%%%%%%%%%%%%%%%%%%%%%%%%%%

hx = zeros(1, N+1);  % Posicion en el punto de control (eje x) en metros (m)
hy = zeros(1, N+1);  % Posicion en el punto de control (eje y) en metros (m)

hx(1) = x1(1); % Posicion en el punto de control del robot en el eje x
hy(1) = y1(1); % Posicion en el punto de control del robot en el eje y


%%%%%%%%%%%%%%%%%%%%%% VELOCIDADES DE REFERENCIA %%%%%%%%%%%%%%%%%%%%%%%%%
```

```
u = velocidad_lineal*ones(1, N); % Velocidad lineal de referencia (m/s)
w = zeros(1, N); % Velocidad angular de referencia (rad/s)

for i=1:N
    if i == 1
        w(i) = radGlobal(1)/ts;
        u(i) = 0;
    end
    if i == 1+side_length_global(1)
        w(i) = (radGlobal(2))/ts;
        u(i) = 0;

    end
    if i == 1+side_length_global(2)
        w(i) = radGlobal(3)/ts;
        u(i) = 0;
    end
    if i == 1+side_length_global(3)
        w(i) = radGlobal(4)/ts;
        u(i) = 0;
    end

end

%%%%%%%%%%%%%%%%%%%%%%%% BUCLE DE SIMULACION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for k=1:N

    phi(k+1)=phi(k)+w(k)*ts; % Integral numérica (método de Euler)

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

    xp1=u(k)*cos(phi(k+1));
    yp1=u(k)*sin(phi(k+1));

    x1(k+1)=x1(k) + xp1*ts ; % Integral numérica (método de Euler)
    y1(k+1)=y1(k) + yp1*ts ; % 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); % Congigurar 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([135 35]); % Orientacion de la figura
axis([-3 11 -3 10 0 2]); % 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);

% d) Bucle de simulacion de movimiento del robot

step=1; % pasos para simulacion

for k=1:step:N

    delete(H1);
    delete(H2);

    xlim([0 4])
    ylim([-3 3])
    view([-0.21 90.00])
    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
```

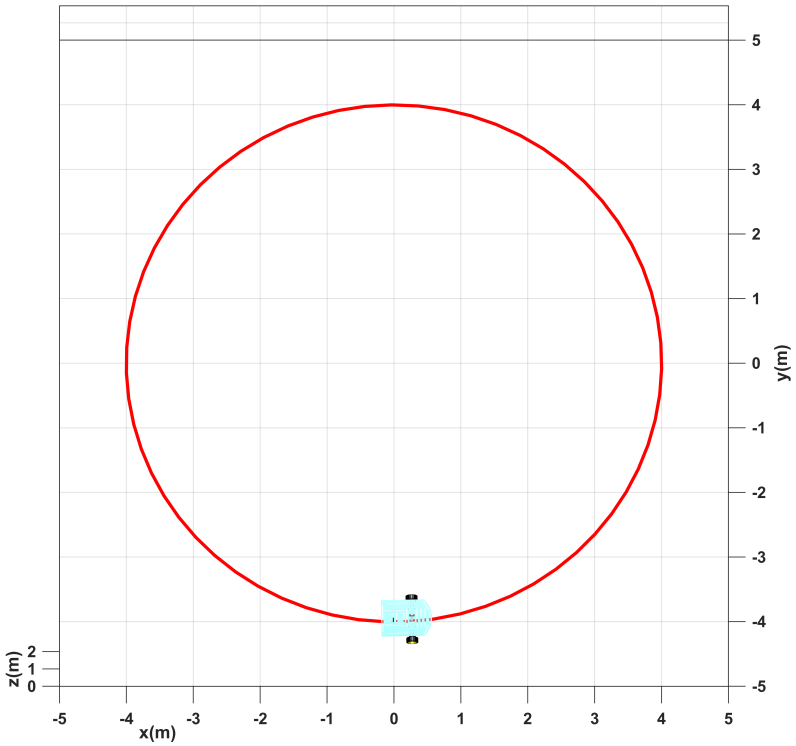


Ejercicio 2

```
clear
close all
clc

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% TIEMPO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

tf = 6.35;              % Tiempo de simulacion en segundos (s)
ts = 0.1;             % Tiempo de muestreo en segundos (s)
t = 0: ts: tf;        % Vector de tiempo
N = length(t);        % Muestras

%%%%%%%%%%%%%%%%%%%%%%%% CONDICIONES INICIALES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x1 = zeros (1,N+1);  % Posición en el centro del eje que une las ruedas (eje x) en
metros (m)
y1 = zeros (1,N+1);  % Posición en el centro del eje que une las ruedas (eje y) en
metros (m)
phi = zeros(1, N+1); % Orientacion del robot en radianes (rad)

x1(1) = 0.2;    % Posicion inicial eje x
y1(1) = -4;   % Posicion inicial eje y
phi(1) = 0;   % Orientacion inicial del robot

%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PUNTO DE CONTROL %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
hx = zeros(1, N+1);  % Posicion en el punto de control (eje x) en metros (m)
hy = zeros(1, N+1);  % Posicion en el punto de control (eje y) en metros (m)

hx(1) = x1(1); % Posicion en el punto de control del robot en el eje x
hy(1) = y1(1); % Posicion en el punto de control del robot en el eje y


%%%%%%%%%%%%%%%%%%%%%% VELOCIDADES DE REFERENCIA %%%%%%%%%%%%%%%%%%%%%%%%%

u = 4*ones(1,N); % Velocidad lineal de referencia (m/s)
w = 1*ones(1,N); % Velocidad angular de referencia (rad/s)


%%%%%%%%%%%%%%%%%%%%%%%% BUCLE DE SIMULACION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for k=1:N

    phi(k+1)=phi(k)+w(k)*ts; % Integral numérica (método de Euler)

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

    xp1=u(k)*cos(phi(k+1));
    yp1=u(k)*sin(phi(k+1));

    x1(k+1)=x1(k) + xp1*ts ; % Integral numérica (método de Euler)
    y1(k+1)=y1(k) + yp1*ts ; % 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); % Congigurar 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([135 35]); % Orientacion de la figura
axis([-3 11 -3 10 0 2]); % 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);

% d) Bucle de simulacion de movimiento del robot

step=1; % pasos para simulacion

for k=1:step:N

    delete(H1);
    delete(H2);
    xlim([-5 5])
    ylim([-5 5])
    view([0 75])
    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
```



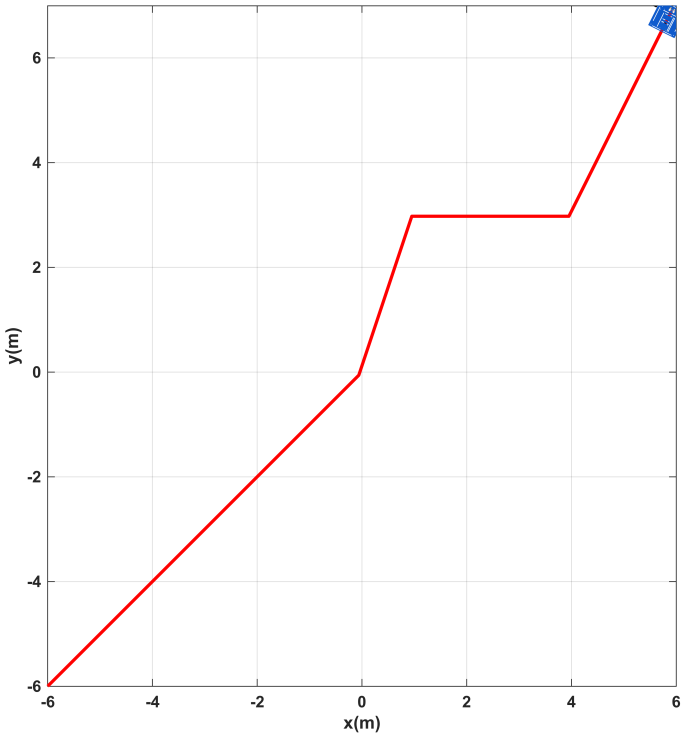
Ejercicio 3

```
clear
close all
clc
load Uniciclo_6.mat

%%%%%%%%%%%%%%%%%%%%%%%%%%%% COORDENADAS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
coordenadas = [-6,-6 ; 0,0 ; 1,3 ; 4,3 ; 6,7]; %Trazo de un cuadrado

%%%%%%%%%%%%%%%%%%%%%%%%%%%% TIEMPO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ts = 0.1;            % Tiempo de muestreo en segundos (s)
velocidad_lineal = 2; %Velocidad de 2m/s

r = zeros(1,size(coordenadas, 1));
radLocal = zeros(1,size(coordenadas, 1));
radGlobal = zeros(1,size(coordenadas, 1));
side_length_local = zeros(1,size(coordenadas, 1));
side_length_global = zeros(1,size(coordenadas, 1));

for j = 1:size(coordenadas, 1)-1 %Para cada coordenada - 1
    punto1 = coordenadas(j, :);
    punto2 = coordenadas(j+1, :);
    r(j) = sqrt((punto2(1)-punto1(1))^2 + (punto2(2)-punto1(2))^2);%Se calcula la
distancia
```

```
    radLocal(j) = atan2(punto2(2)-punto1(2), punto2(1)-punto1(1)); % Calcula el
ángulo en radianes
    side_length_local(j) = round(r(j)/ts/velocidad_lineal) + 1; %Se calcula la
longitud local
    if j==1
        radGlobal(j) = atan2(punto2(2)-punto1(2), punto2(1)-punto1(1)); % Calcula
el ángulo en radianes
        side_length_global(j) = side_length_local(j);
    else
        radGlobal(j) = atan2(punto2(2)-punto1(2), punto2(1)-punto1(1))-
radLocal(j-1); % Calcula el ángulo en radianes
        side_length_global(j) = side_length_local(j)+side_length_global(j-1);
    end

end


distancia_total = 0;
for j=1:size(coordenadas, 1)-1
    distancia_total = r(j) + distancia_total;
end

tiempo_recorrido = distancia_total/velocidad_lineal+((size(coordenadas, 1)-1)*ts);
%Tiempo de recorrido

tf = tiempo_recorrido;             % Tiempo de simulacion en segundos (s)
t = 0: ts: tf;       % Vector de tiempo
N = length(t);       % Muestras

%%%%%%%%%%%%%%%%%%%%%%%% CONDICIONES INICIALES %%%%%%%%%%%%%%%%%%%%%%%%%%%%

x1 = zeros (1,N+1);  % Posición en el centro del eje que une las ruedas (eje x) en
metros (m)
y1 = zeros (1,N+1);  % Posición en el centro del eje que une las ruedas (eje y) en
metros (m)
phi = zeros(1, N+1); % Orientacion del robot en radianes (rad)

x1(1) = -6;    % Posicion inicial eje x
y1(1) = -6;   % Posicion inicial eje y
phi(1) = 0;   % Orientacion inicial del robot

%%%%%%%%%%%%%%%%%%%%%%%%%%%% PUNTO DE CONTROL %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

hx = zeros(1, N+1);  % Posicion en el punto de control (eje x) en metros (m)
hy = zeros(1, N+1);  % Posicion en el punto de control (eje y) en metros (m)

hx(1) = x1(1); % Posicion en el punto de control del robot en el eje x
hy(1) = y1(1); % Posicion en el punto de control del robot en el eje y
```

```
%%%%%%%%%%%%%%%%%%%%% VELOCIDADES DE REFERENCIA %%%%%%%%%%%%%%%%%%%%%%%%


u = velocidad_lineal*ones(1, N); % Velocidad lineal de referencia (m/s)
w = zeros(1, N); % Velocidad angular de referencia (rad/s)

for i=1:N
    if i == 1
        w(i) = radGlobal(1)/ts;
        u(i) = 0;
    end
    if i == 1+side_length_global(1)
        w(i) = (radGlobal(2))/ts;
        u(i) = 0;

    end
    if i == 1+side_length_global(2)
        w(i) = radGlobal(3)/ts;
        u(i) = 0;
    end
    if i == 1+side_length_global(3)
        w(i) = radGlobal(4)/ts;
        u(i) = 0;
    end

end

%%%%%%%%%%%%%%%%%%%%%%% BUCLE DE SIMULACION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for k=1:N

    phi(k+1)=phi(k)+w(k)*ts; % Integral numérica (método de Euler)

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

    xp1=u(k)*cos(phi(k+1));
    yp1=u(k)*sin(phi(k+1));

    x1(k+1)=x1(k) + xp1*ts ; % Integral numérica (método de Euler)
    y1(k+1)=y1(k) + yp1*ts ; % 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); % Congigurar 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([135 35]); % Orientacion de la figura
axis([-3 11 -3 10 0 2]); % 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);

% d) Bucle de simulacion de movimiento del robot

step=1; % pasos para simulacion

for k=1:step:N

    delete(H1);
    delete(H2);

    xlim([-6 6])
    ylim([-6 7])
    view([-0.21 90.00])

    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
```


6
4
2
0
-2
-4
-6
y(m)
-6
-4
-2
0
2
4
6
x(m)