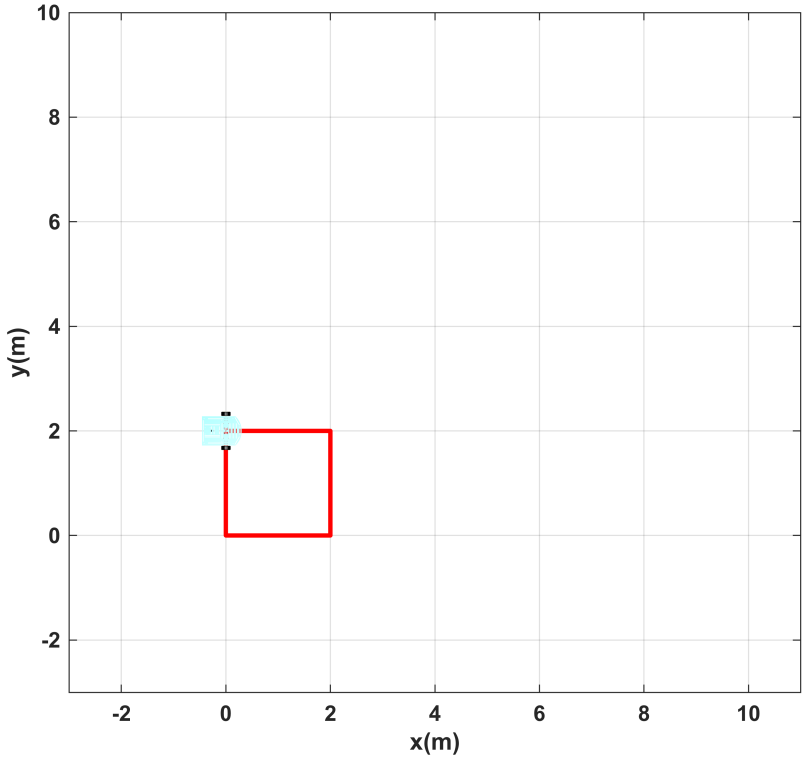
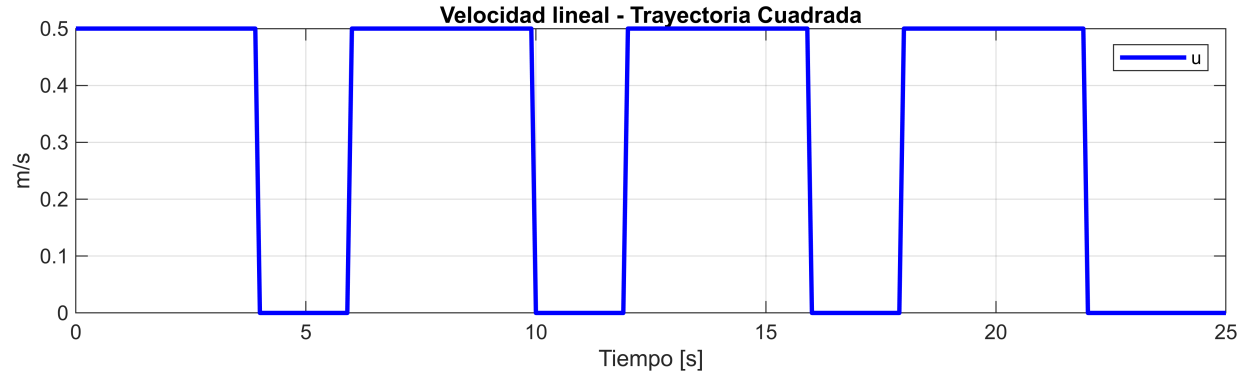
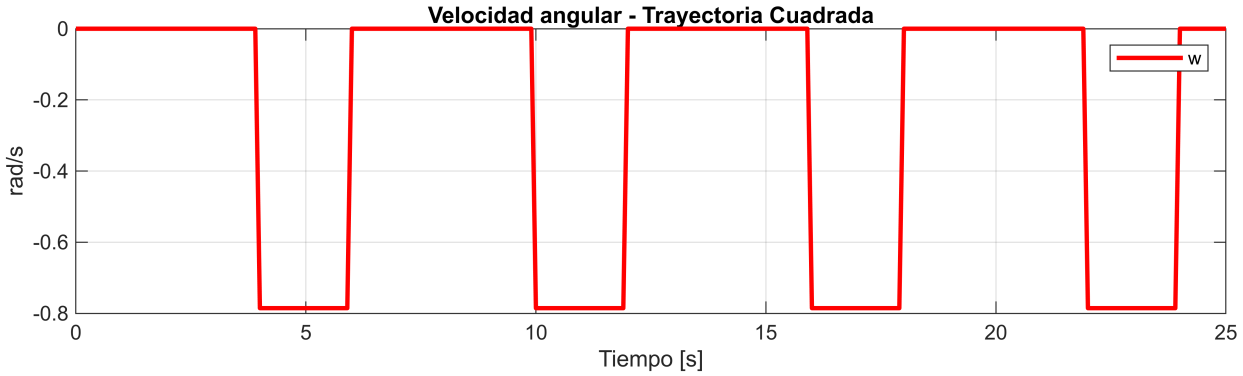
```
clear
close all
clc

%% ==================== SIMULACIÓN CUADRADA ========================

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% TIEMPO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tf = 25;           % Tiempo total de simulación
ts = 0.1;          % Paso de muestreo
t_square = 0:ts:tf;    % Vector de tiempo
N_square = length(t_square);    % Número de iteraciones

%%%%%%%%%%%%%%%%%%%%%%%% CONDICIONES INICIALES %%%%%%%%%%%%%%%%%%%%%%
x1_square = zeros(1,N_square+1);   % Posición x
y1_square = zeros(1,N_square+1);   % Posición y
phi_square = zeros(1,N_square+1);  % Orientación

x1_square(1) = 0;   % Posición inicial x
y1_square(1) = 2;   % Posición inicial y
phi_square(1) = 0;  % Orientación inicial

hx_square = zeros(1,N_square+1);   % Historial x
hy_square = zeros(1,N_square+1);   % Historial y
hx_square(1) = x1_square(1);
hy_square(1) = y1_square(1);

%%%%%%%%%%%%%%%%%%%%%%% VELOCIDADES DE REFERENCIA %%%%%%%%%%%%%%%%%%%%
u_square = zeros(1,N_square);  % Velocidad lineal
w_square = zeros(1,N_square); % Velocidad angular

v = 0.5;        % Velocidad lineal constante
L = 2;          % Longitud del lado del cuadrado
t_lado = L / v; % Tiempo para recorrer un lado
w_val = -pi/4;  % Velocidad angular constante para girar
t_giro = (pi/2) / abs(w_val);   % Tiempo necesario para girar 90°

pasos_lado = round(t_lado / ts);    % Pasos para un lado
pasos_giro = round(t_giro / ts);    % Pasos para un giro

idx = 1;
for i = 1:4
    % Avance recto
    if idx + pasos_lado - 1 <= N_square
        u_square(idx : idx + pasos_lado - 1) = v;
        idx = idx + pasos_lado;
    end
    % Giro
    if idx + pasos_giro - 1 <= N_square
        w_square(idx : idx + pasos_giro - 1) = w_val;
        idx = idx + pasos_giro;
```

```
    end
end

%%%%%%%%%%%%%%%%%%%%%%% BUCLE DE SIMULACIÓN %%%%%%%%%%%%%%%%%%%%%%%
for k = 1:N_square
    % Actualización de orientación
    phi_square(k+1) = phi_square(k) + w_square(k)*ts;

    % Cálculo de desplazamientos
    xp = u_square(k)*cos(phi_square(k+1));
    yp = u_square(k)*sin(phi_square(k+1));

    % Actualización de posiciones
    x1_square(k+1) = x1_square(k) + xp*ts;
    y1_square(k+1) = y1_square(k) + yp*ts;

    % Almacenamiento del historial
    hx_square(k+1) = x1_square(k+1);
    hy_square(k+1) = y1_square(k+1);
end

%%%%%%%%%%%%%%%%%%%%%%% SIMULACIÓN GRÁFICA %%%%%%%%%%%%%%%%%%%%%%%
scene1 = figure;
set(scene1,'Color','white');
set(gca,'FontWeight','bold');
sizeScreen = get(0,'ScreenSize');
set(scene1,'position',sizeScreen);
camlight('headlight');
axis equal; grid on; box on;
xlabel('x(m)'); ylabel('y(m)'); zlabel('z(m)');
view([2]);
axis([-3 11 -3 10 0 2]);

scale = 4;  % Tamaño del robot
MobileRobot_5;
H1 = MobilePlot_4(x1_square(1), y1_square(1), phi_square(1), scale); hold on;
H2 = plot3(hx_square(1), hy_square(1), 0, 'r', 'lineWidth', 2);

% Animación del movimiento
for k = 1:N_square
    delete(H1); delete(H2);
    H1 = MobilePlot_4(x1_square(k), y1_square(k), phi_square(k), scale);
    H2 = plot3(hx_square(1:k), hy_square(1:k), zeros(1,k), 'r', 'lineWidth', 2);
    pause(ts);
end
```




```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% GRÁFICAS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
graph1 = figure;
set(graph1,'position',sizeScreen);
subplot(211)
plot(t_square(1:N_square), u_square, 'b', 'LineWidth', 2);
grid on; xlabel('Tiempo [s]'); ylabel('m/s'); legend('u');
title('Velocidad lineal - Trayectoria Cuadrada');

subplot(212)
plot(t_square(1:N_square), w_square, 'r', 'LineWidth', 2);
grid on; xlabel('Tiempo [s]'); ylabel('rad/s'); legend('w');
title('Velocidad angular - Trayectoria Cuadrada');
```

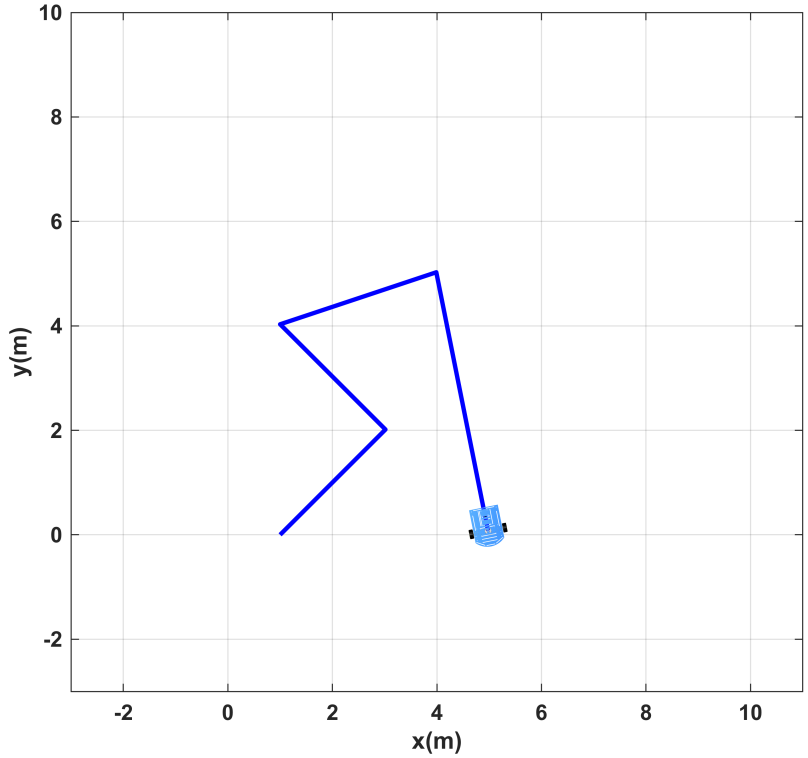




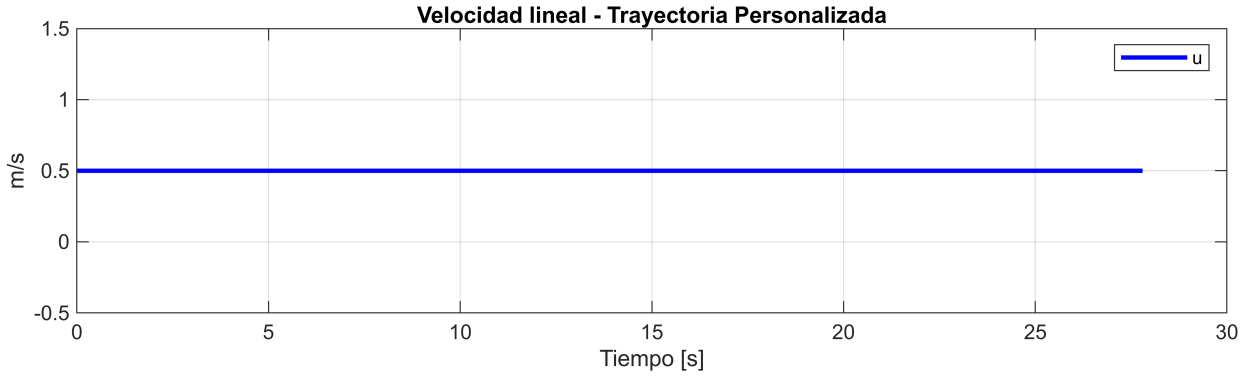

```
%% ===================== SIMULACIÓN PERSONALIZADA ==========================

%%%%%%%%%%%%%%%%%%%%%%%%%%%% PUNTOS DE TRAYECTORIA %%%%%%%%%%%%%%%%%%%%%%%%
% Se definen los puntos por los que el robot debe pasar
p1 = [1, 0];
p2 = [3, 2];
p3 = [1, 4];
p4 = [4, 5];
p5 = [5, 0];
puntos = [p1; p2; p3; p4; p5];

% Inicialización de vectores de trayectoria
x1_custom = []; y1_custom = []; phi_custom = [];
hx_custom = []; hy_custom = [];

% Posición inicial
x_actual = p1(1);
y_actual = p1(2);
phi_actual = 0;

% Guardar el primer punto
x1_custom(1) = x_actual;
y1_custom(1) = y_actual;
phi_custom(1) = phi_actual;
```

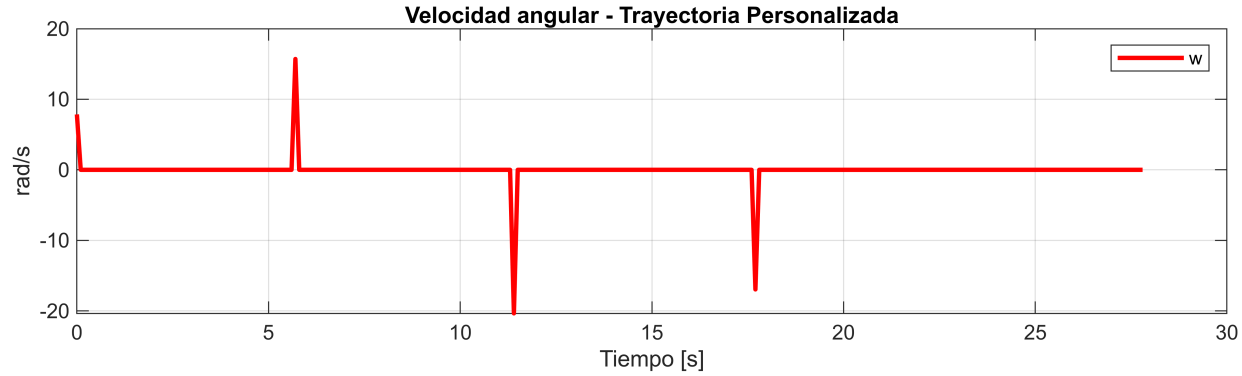
```
hx_custom(1) = x_actual;
hy_custom(1) = y_actual;

%%%%%%%%%%%%%%%%%%%%%%%% GENERACIÓN DEL RECORRIDO %%%%%%%%%%%%%%%%%%%%%%%%
for i = 1:(size(puntos,1)-1)
    p_origen = puntos(i,:);
    p_destino = puntos(i+1,:);

    dx = p_destino(1) - p_origen(1);
    dy = p_destino(2) - p_origen(2);

    distancia = sqrt(dx^2 + dy^2);     % Longitud del tramo
    tiempo = distancia / v;            % Tiempo para recorrerlo
    pasos = round(tiempo / ts);        % Número de pasos

    phi_actual = atan2(dy, dx);        % Orientación hacia el destino

    % Movimiento en línea recta hacia el siguiente punto
    for k = 1:pasos
        x_actual = x_actual + v * cos(phi_actual) * ts;
        y_actual = y_actual + v * sin(phi_actual) * ts;

        x1_custom(end+1) = x_actual;
        y1_custom(end+1) = y_actual;
        phi_custom(end+1) = phi_actual;
        hx_custom(end+1) = x_actual;
        hy_custom(end+1) = y_actual;
    end
end

% Longitud del recorrido personalizado
N_custom = length(x1_custom) - 1;
t_custom = 0:ts:(N_custom-1)*ts;
u_custom = v * ones(1, N_custom);   % Velocidad constante
w_custom = zeros(1, N_custom);      % Inicialización de w

% Cálculo de velocidad angular como derivada de la orientación
for k = 2:length(phi_custom)
    w_custom(k-1) = (phi_custom(k) - phi_custom(k-1)) / ts;
end

%%%%%%%%%%%%%%%%%%%%%%%% SIMULACIÓN GRÁFICA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
scene2 = figure;
set(scene2,'Color','white');
set(gca,'FontWeight','bold');
set(scene2,'position',sizeScreen);
camlight('headlight');
axis equal; grid on; box on;
xlabel('x(m)'); ylabel('y(m)'); zlabel('z(m)');
view([2]);
```

```
axis([-3 11 -3 10 0 2]);

% Primer frame del robot
H3 = MobilePlot_4(x1_custom(1), y1_custom(1), phi_custom(1), scale); hold on;
H4 = plot3(hx_custom(1), hy_custom(1), 0, 'b', 'lineWidth', 2);

% Animación del recorrido personalizado
for k = 1:N_custom
    delete(H3); delete(H4);
    H3 = MobilePlot_4(x1_custom(k), y1_custom(k), phi_custom(k), scale);
    H4 = plot3(hx_custom(1:k), hy_custom(1:k), zeros(1,k), 'b', 'lineWidth', 2);
    pause(ts);
end
```




```
%%%%%%%%%%%%%%%%%%%%%%%%%%%% GRÁFICAS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
graph2 = figure;
set(graph2,'position',sizeScreen);
subplot(211)
plot(t_custom, u_custom, 'b', 'LineWidth', 2);
grid on; xlabel('Tiempo [s]'); ylabel('m/s'); legend('u');
title('Velocidad lineal - Trayectoria Personalizada');

subplot(212)
plot(t_custom, w_custom, 'r', 'LineWidth', 2);
grid on; xlabel('Tiempo [s]'); ylabel('rad/s'); legend('w');
title('Velocidad angular - Trayectoria Personalizada');
```


Velocidad lineal - Trayectoria Personalizada
u
m/s
1.5
1
0.5
0
-0.5
0
5
10
15
20
25
30
Tiempo [s]


Velocidad angular - Trayectoria Personalizada
w
rad/s
20
10
0
-10
-20
0
5
10
15
20
25
30
Tiempo [s]