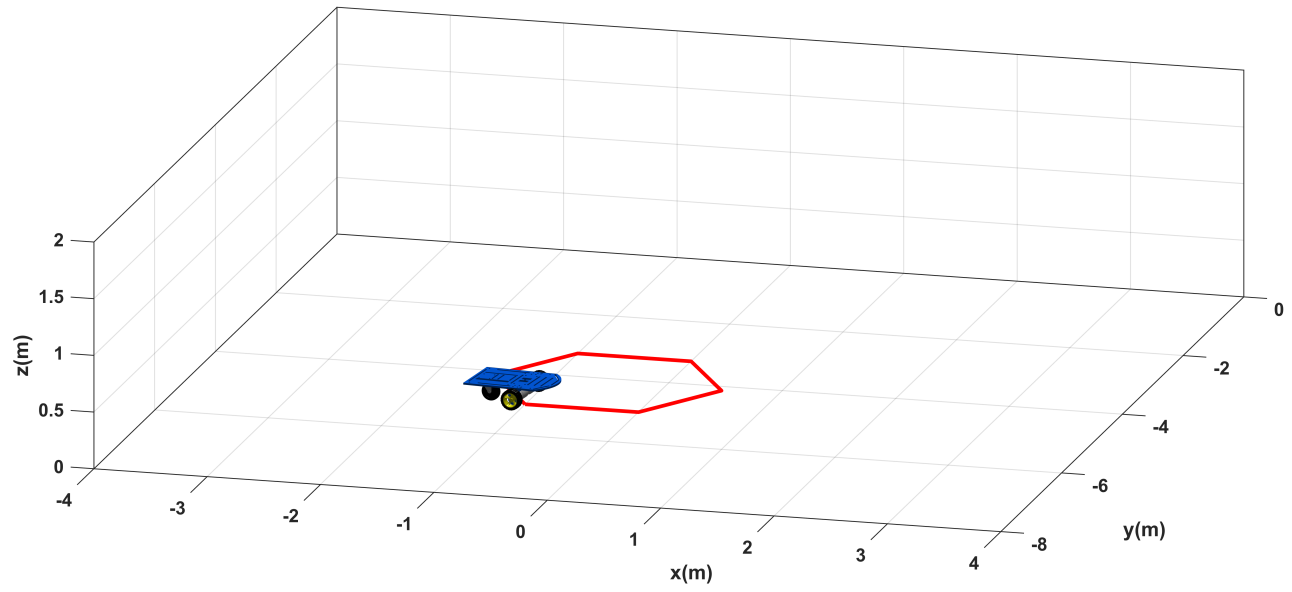
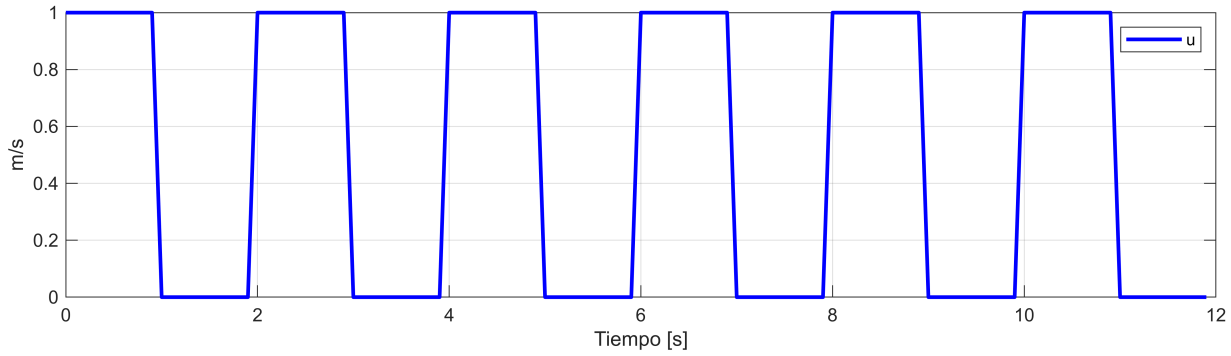
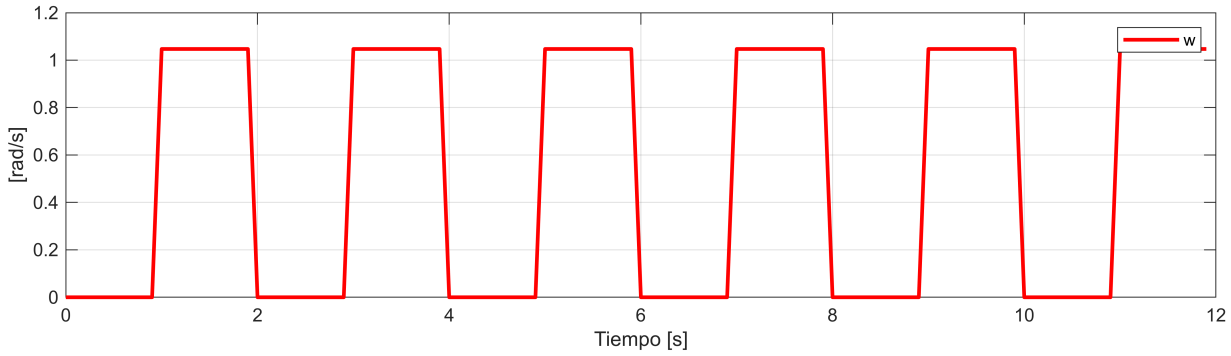
```
clear
close all
clc

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

dt = 0.1;               % Tiempo de muestreo
num_intervalos = 12;    % 12 pasos de la tabla
muestras_por_intervalo = 1 / dt; % 10 muestras por paso
total_muestras = num_intervalos * muestras_por_intervalo; % Total de muestras = 120
tiempo_total = total_muestras * dt;       % Tiempo total
tiempo_vector = 0:dt:tiempo_total;        % Vector de tiempo

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

posX = zeros(1, total_muestras+1);
posY = zeros(1, total_muestras+1);
orientacion = zeros(1, total_muestras+1);

posX(1) = -1;
posY(1) = -5;
orientacion(1) = 0;

trayectoriaX = zeros(1, total_muestras+1);
trayectoriaY = zeros(1, total_muestras+1);
trayectoriaX(1) = posX(1);
trayectoriaY(1) = posY(1);

%%%%%%%%%%%%%%%%%%%%%% VELOCIDADES SEGÚN TABLA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

vel_lineal_tabla = [1 0 1 0 1 0 1 0 1 0 1 0];          % Velocidades lineales
vel_angular_tabla = [0 pi/3 0 pi/3 0 pi/3 0 pi/3 0 pi/3 0 pi/3]; % Velocidades
angulares

vel_lineal = zeros(1, total_muestras);
vel_angular = zeros(1, total_muestras);

for i = 1:num_intervalos
    inicio_idx = (i-1)*muestras_por_intervalo + 1;
    fin_idx = i*muestras_por_intervalo;
    vel_lineal(inicio_idx:fin_idx) = vel_lineal_tabla(i);
    vel_angular(inicio_idx:fin_idx) = vel_angular_tabla(i);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%% SIMULACIÓN %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for k = 1:total_muestras
    orientacion(k+1) = orientacion(k) + vel_angular(k) * dt;
    delta_x = vel_lineal(k) * cos(orientacion(k+1));
    delta_y = vel_lineal(k) * sin(orientacion(k+1));
```

```
    posX(k+1) = posX(k) + delta_x * dt;
    posY(k+1) = posY(k) + delta_y * dt;
    trayectoriaX(k+1) = posX(k+1);
    trayectoriaY(k+1) = posY(k+1);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%% SIMULACIÓN 3D %%%%%%%%%%%%%%%%%%%%%%%%%%%%

fig_3d = figure;
set(fig_3d,'Color','white');
set(gca,'FontWeight','bold');
pantalla = get(0,'ScreenSize');
set(fig_3d,'position',pantalla);
camlight('headlight');
axis equal;
grid on;
box on;
xlabel('x(m)'); ylabel('y(m)'); zlabel('z(m)');
view([15 15]);
axis([-4 4 -8 0 0 2]);

escala = 4;
MobileRobot_5;
robot_plot = MobilePlot_4(posX(1), posY(1), orientacion(1), escala); hold on;
trayectoria_plot = plot3(trayectoriaX(1), trayectoriaY(1), 0, 'r', 'lineWidth', 2);

paso_plot = 1;
for k = 1:paso_plot:total_muestras
    delete(robot_plot);
    delete(trayectoria_plot);
    robot_plot = MobilePlot_4(posX(k), posY(k), orientacion(k), escala);
    trayectoria_plot = plot3(trayectoriaX(1:k), trayectoriaY(1:k), zeros(1,k), 'r',
'lineWidth', 2);
    pause(dt);
end
```

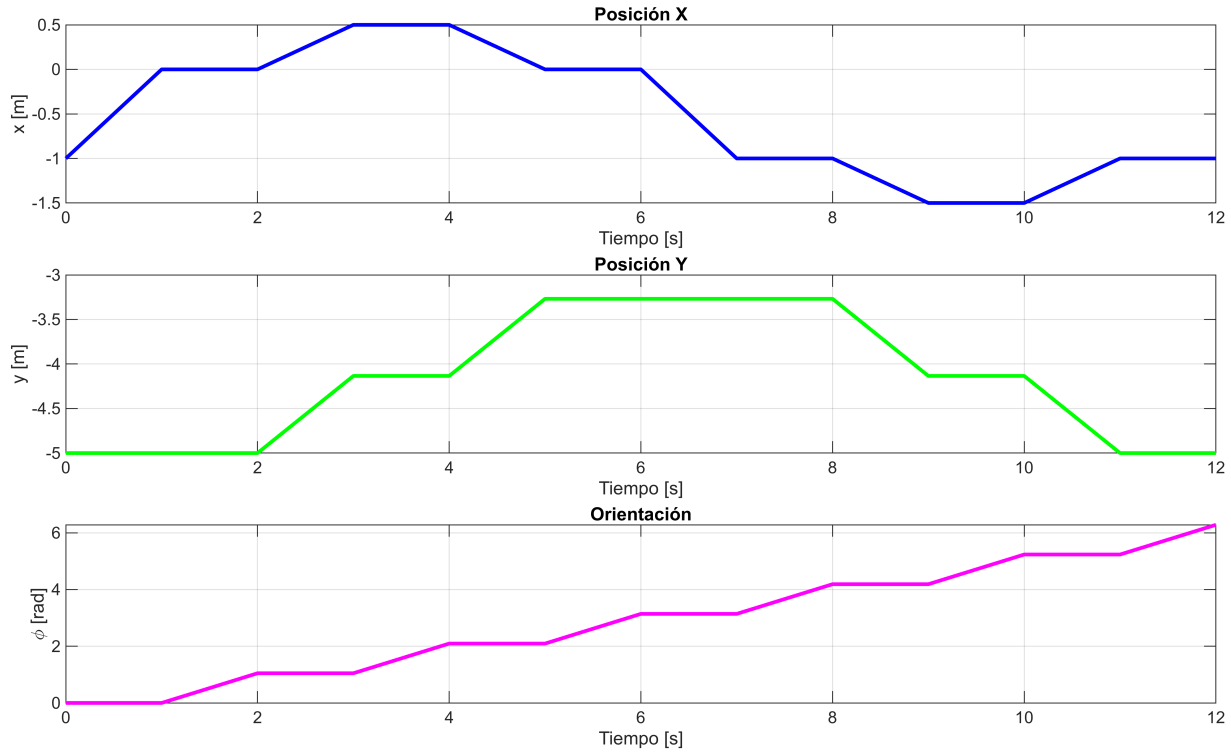



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Graficas %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fig_vel = figure;
set(fig_vel, 'position', pantalla);

subplot(211)
plot(tiempo_vector(1:total_muestras), vel_lineal, 'b', 'LineWidth', 2), grid on
xlabel('Tiempo [s]')
ylabel('m/s')
legend('u')

subplot(212)
plot(tiempo_vector(1:total_muestras), vel_angular, 'r', 'LineWidth', 2), grid on
xlabel('Tiempo [s]')
ylabel('[rad/s]')
legend('w')
```

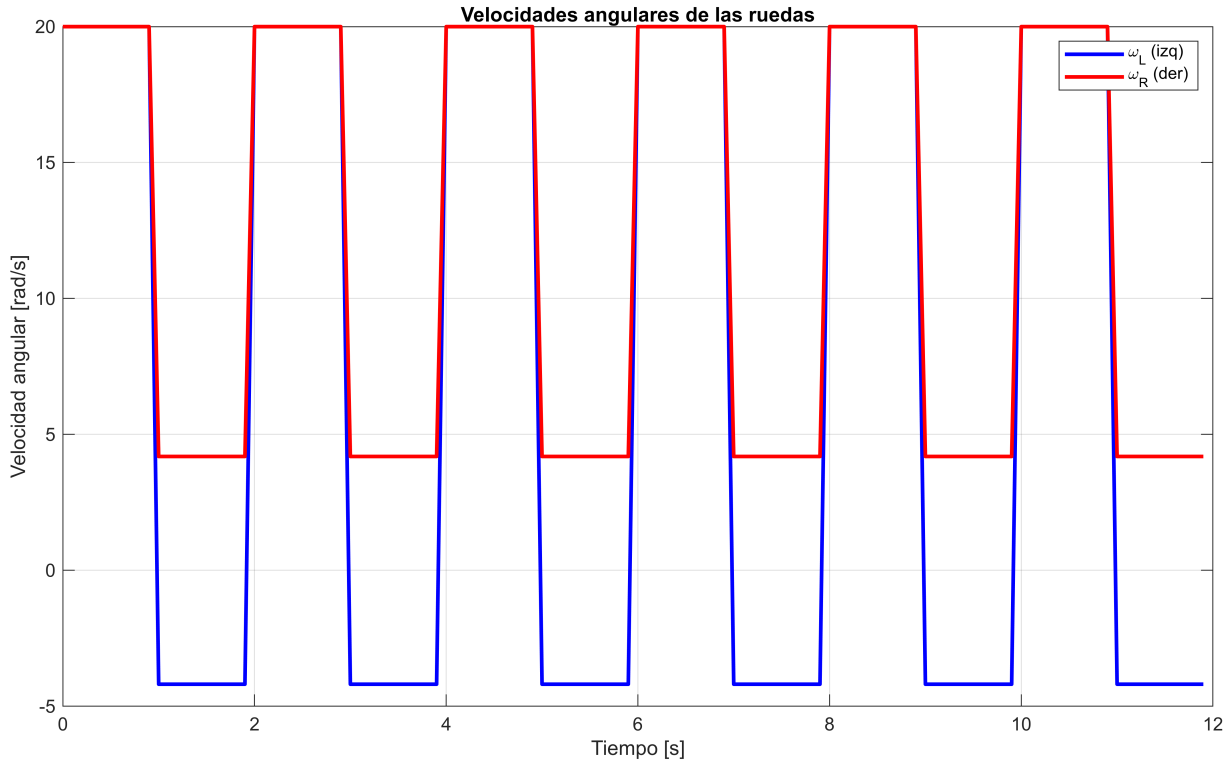





```
%%%%%%%%%%%%%%%%%%%%%% GRAFICAR POSE %%%%%%%%%%%%%%%%%%%

fig_pose = figure;
set(fig_pose, 'position', pantalla);

subplot(311)
plot(tiempo_vector(1:total_muestras+1), posX, 'b', 'LineWidth', 2), grid on
xlabel('Tiempo [s]')
ylabel('x [m]')
title('Posición X');

subplot(312)
plot(tiempo_vector(1:total_muestras+1), posY, 'g', 'LineWidth', 2), grid on
xlabel('Tiempo [s]')
ylabel('y [m]')
title('Posición Y');

subplot(313)
plot(tiempo_vector(1:total_muestras+1), orientacion, 'm', 'LineWidth', 2), grid on
xlabel('Tiempo [s]')
ylabel('\phi [rad]')
title('Orientación');
```




```
%%%%%%%%%%%%%%%%%% CÁLCULO DE VELOCIDADES ANGULARES DE RUEDAS %%%%%%%%%%%%%%%%%%

radio_rueda = 0.05;
dist_ruedas = 0.4;

vel_izq = (2*vel_lineal - vel_angular*dist_ruedas) / (2*radio_rueda);
vel_der = (2*vel_lineal + vel_angular*dist_ruedas) / (2*radio_rueda);

fig_ruedas = figure;
set(fig_ruedas, 'position', pantalla);
plot(tiempo_vector(1:total_muestras), vel_izq, 'b', 'LineWidth', 2), hold on
plot(tiempo_vector(1:total_muestras), vel_der, 'r', 'LineWidth', 2), grid on
xlabel('Tiempo [s]')
ylabel('Velocidad angular [rad/s]')
legend('\omega_L (izq)', '\omega_R (der)')
title('Velocidades angulares de las ruedas')
```




```
%%%%%%%%%%%%%%%%%%%%%%%%% POSE CADA SEGUNDO (a) %%%%%%%%%%%%%%%%%%%%%%%%%

instantes = 1:muestras_por_intervalo:(total_muestras+1-muestras_por_intervalo);  %
12 pasos

tabla_pose = table((0:num_intervalos-1)', ...
    posX(instantes)', ...
    posY(instantes)', ...
    orientacion(instantes)', ...
    'VariableNames', {'Tiempo_s', 'X_m', 'Y_m', 'Theta_rad'});

disp('POSE DEL ROBOT CADA SEGUNDO:')
```

```
POSE DEL ROBOT CADA SEGUNDO:
```

```
disp(tabla_pose)
```

Tiempo_s	X_m	Y_m	Theta_rad
0	-1	-5	0
1	-1.3878e-16	-5	0
2	-1.3878e-16	-5	1.0472
3	0.5	-4.134	1.0472
4	0.5	-4.134	2.0944
5	4.0246e-16	-3.2679	2.0944
6	4.0246e-16	-3.2679	3.1416
7	-1	-3.2679	3.1416

```
        8          -1    -3.2679     4.1888
        9        -1.5     -4.134     4.1888
       10        -1.5     -4.134      5.236
       11          -1         -5      5.236
```

```
fig_tabla = figure;

%%%%%%%%%%%%%%%%%%%%%%%%% MOSTRAR TABLA COMO TEXTO (a) %%%%%%%%%%%%%%%%%%%%%%%%%

texto_pose = sprintf(' PASO |    X (m)   |    Y (m)   |  Theta (rad)\n');
texto_pose = [texto_pose, sprintf('----------------------------------------\n')];

for i = 1:num_intervalos
    texto_pose = [texto_pose, sprintf('  %2d   |  %8.3f |  %8.3f |  %8.3f\n', ...
        i, posX(instantes(i)), posY(instantes(i)), orientacion(instantes(i)))];
end

%%%%%%%%%%%%%%%%%%%%%%%% POSE FINAL (b) %%%%%%%%%%%%%%%%%%%%%%%%

fprintf('\nPOSE FINAL DEL ROBOT (tras 12 pasos):\n');
```

```
POSE FINAL DEL ROBOT (tras 12 pasos):
```

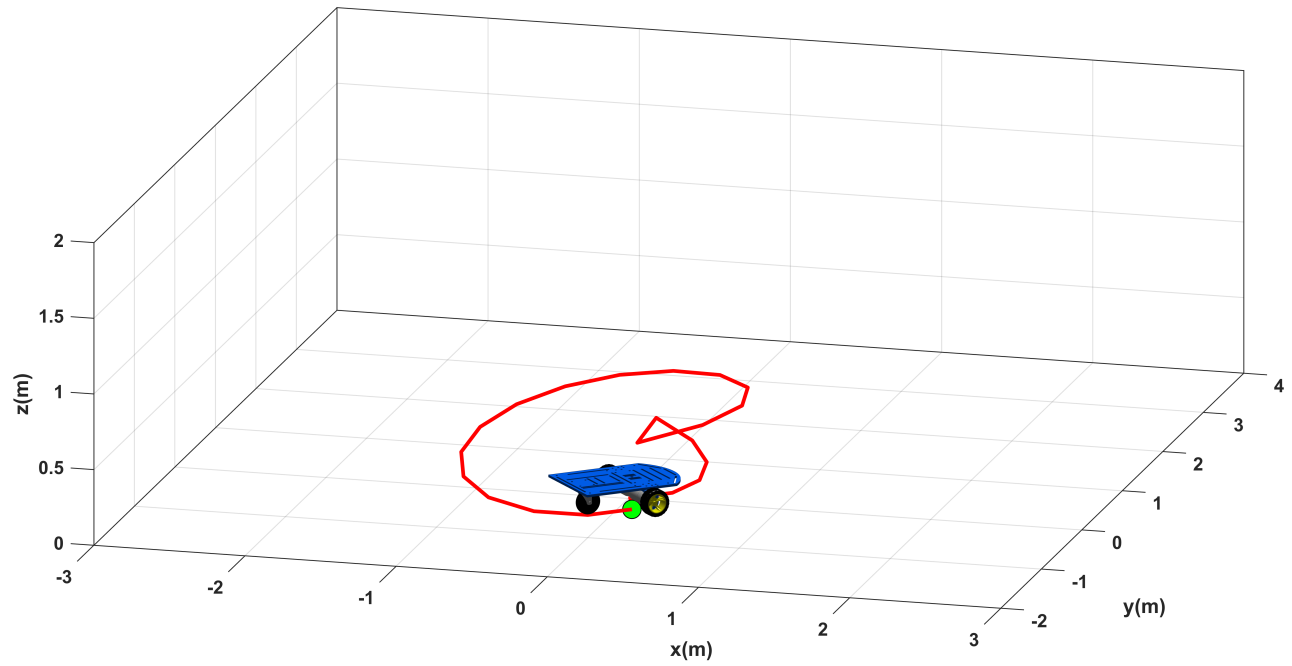
```
fprintf('x = %.4f m\ny = %.4f m\nphi = %.4f rad\n', posX(end), posY(end),
orientacion(end));
```

```
x = -1.0000 m
y = -5.0000 m
phi = 6.2832 rad
```

```
%%%%%%%%%%%%%%%%%%%%% PARÁMETROS DEL ROBOT %%%%%%%%%%%%%%%%%%%%%

radioRueda = 0.1;    % Radio de las ruedas (m)
distanciaRuedas = 0.4; % Distancia entre ruedas (m)
deltaT = 1.0;        % Tiempo por paso (s)

% Entradas de velocidades angulares por rueda (ωR y ωL)
omegaDer = [4.582, 4.773, 5.291, 5.960, 6.490, -1.168, -1.364, 5.960, 5.291, 4.773,
...
            4.582, 4.773, 5.291, 5.960, 6.490, 6.686, 6.490, 5.960, 5.291, 4.773,
4.582];
omegaIzq = [1.701, 2.353, 3.676, 4.856, 5.618, 13.735, 13.472, 4.856, 3.676, 2.353,
...
            1.701, 2.353, 3.676, 4.856, 5.618, 5.881, 5.618, 4.856, 3.676, 2.353,
1.701];

numPasos = length(omegaDer);  % 21 muestras

% Inicialización de vectores de pose
posX = zeros(1, numPasos+1);        % Posición en x
```

```
posY = zeros(1, numPasos+1);        % Posición en y
anguloTheta = zeros(1, numPasos+1); % Orientación en radianes

% Pose inicial
posX(1) = 0;
posY(1) = 0;
anguloTheta(1) = 0;

%%%%%%%%%%%%%%%%%%% CÁLCULO DE VELOCIDADES Y POSE %%%%%%%%%%%%%%%%%%%

velLineal = (radioRueda/2) * (omegaDer + omegaIzq);         % Velocidad lineal v(k)
velAngular = (radioRueda/distanciaRuedas) * (omegaDer - omegaIzq); % Velocidad
angular w(k)

for k = 1:numPasos
    anguloTheta(k+1) = anguloTheta(k) + velAngular(k) * deltaT;
    posX(k+1) = posX(k) + velLineal(k) * cos(anguloTheta(k+1)) * deltaT;
    posY(k+1) = posY(k) + velLineal(k) * sin(anguloTheta(k+1)) * deltaT;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%% SIMULACIÓN 3D %%%%%%%%%%%%%%%%%%%%%%%%%%%%
trayX = posX;
trayY = posY;
trayTheta = anguloTheta;
histX = posX;
histY = posY;

figura3D = figure;
set(figura3D,'Color','white');
set(gca,'FontWeight','bold');
tamPantalla = get(0,'ScreenSize');
set(figura3D,'position',tamPantalla);
camlight('headlight');
axis equal;
grid on;
box on;
xlabel('x(m)'); ylabel('y(m)'); zlabel('z(m)');
view([15 15]);
axis([-3 3 -2 4 0 2]);

escalaRobot = 4;
MobileRobot_5;
grafRobot = MobilePlot_4(trayX(1), trayY(1), trayTheta(1), escalaRobot); hold on;
grafHistorial = plot3(histX(1), histY(1), 0, 'r', 'lineWidth', 2);
plot3(posX(end), posY(end), 0, 'ko', 'MarkerFaceColor', 'g', 'MarkerSize', 10);  %
punto final

pasoAnimacion = 1;
for k = 1:pasoAnimacion:numPasos+1
    delete(grafRobot);
```

```
    delete(grafHistorial);
    grafRobot = MobilePlot_4(trayX(k), trayY(k), trayTheta(k), escalaRobot);
    grafHistorial = plot3(histX(1:k), histY(1:k), zeros(1,k), 'r', 'lineWidth', 2);
    pause(deltaT);
end
```



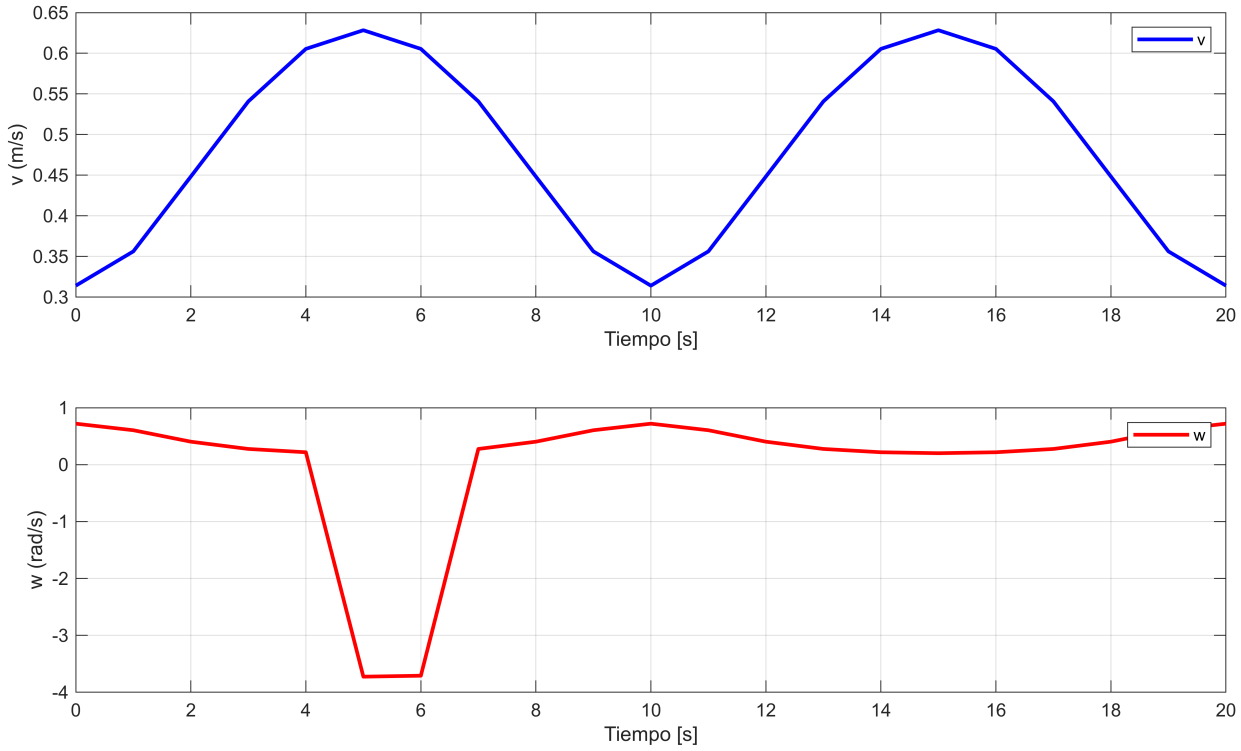

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% TABLA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

tiempoSim = (0:numPasos)';                    % Ahora de 0 a 21
thetaGrados = rad2deg(anguloTheta(:));        % Convertir orientación a grados
velLinealFull = [velLineal, NaN];             % Añadir NaN al final
velAngularFull = [velAngular, NaN];
omegaDerFull = [omegaDer, NaN];
omegaIzqFull = [omegaIzq, NaN];

tablaDatos = table( ...
    tiempoSim, velLinealFull(:), velAngularFull(:), omegaDerFull(:),
omegaIzqFull(:), ...
    posX(:), posY(:), thetaGrados, ...
    'VariableNames', {'t_s', 'v_m_s', 'w_rad_s', 'wR_rad_s', 'wL_rad_s', 'x_m',
'y_m', 'theta_deg'});

disp('TABLA DE POSE FINAL DESPUÉS DE CADA VELOCIDAD:')
```

```
TABLA DE POSE FINAL DESPUÉS DE CADA VELOCIDAD:
```

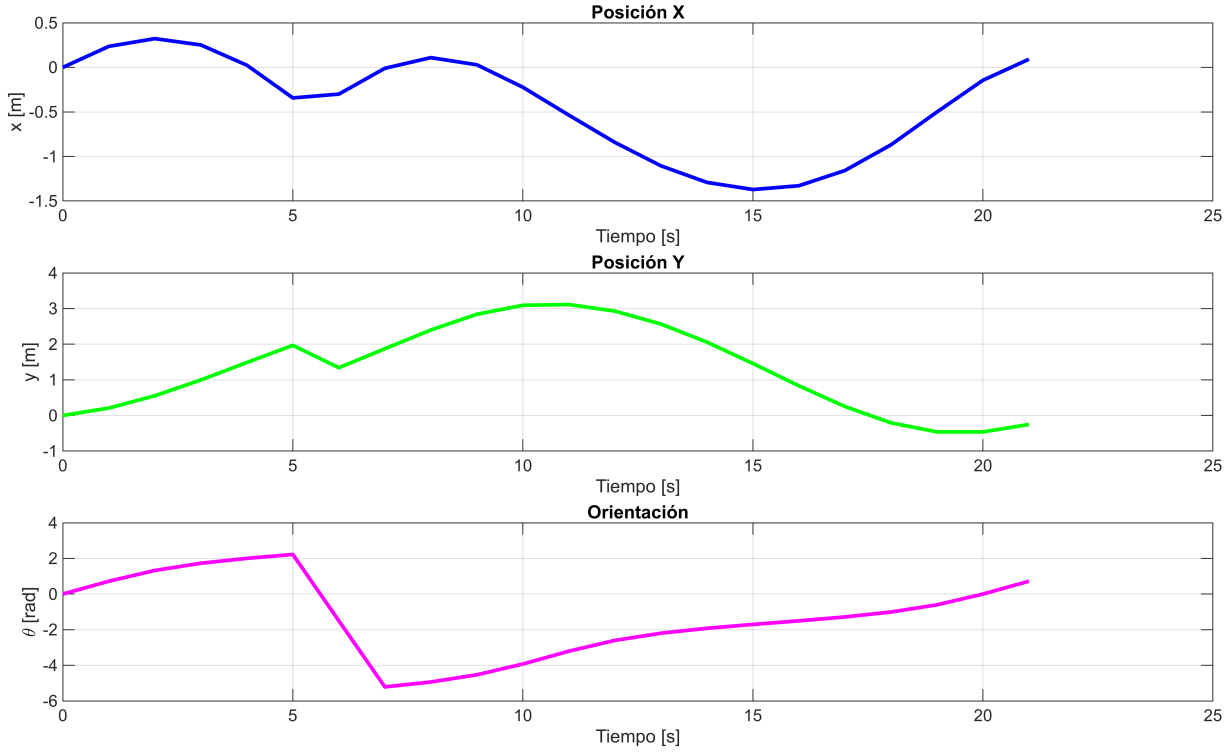
```
disp(tablaDatos)
```

t_s	v_m_s	w_rad_s	wR_rad_s	wL_rad_s	x_m	y_m	theta_deg
0	0.31415	0.72025	4.582	1.701	0	0	0
1	0.3563	0.605	4.773	2.353	0.23613	0.2072	41.267
2	0.44835	0.40375	5.291	3.676	0.32274	0.55282	75.931
3	0.5408	0.276	5.96	4.856	0.2521	0.99557	99.064
4	0.6054	0.218	6.49	5.618	0.024596	1.4862	114.88
5	0.62835	-3.7257	-1.168	13.735	-0.34284	1.9673	127.37
6	0.6054	-3.709	-1.364	13.472	-0.30012	1.3404	-86.101
7	0.5408	0.276	5.96	4.856	-0.010216	1.8719	-298.61
8	0.44835	0.40375	5.291	3.676	0.10958	2.3993	-282.8
9	0.3563	0.605	4.773	2.353	0.029136	2.8403	-259.66
10	0.31415	0.72025	4.582	1.701	-0.2228	3.0923	-225
11	0.3563	0.605	4.773	2.353	-0.53629	3.1127	-183.73
12	0.44835	0.40375	5.291	3.676	-0.84192	2.9296	-149.07
13	0.5408	0.276	5.96	4.856	-1.105	2.5666	-125.94
14	0.6054	0.218	6.49	5.618	-1.2911	2.0588	-110.12
15	0.62835	0.20125	6.686	5.881	-1.3715	1.4588	-97.632
16	0.6054	0.218	6.49	5.618	-1.3288	0.83187	-86.101
17	0.5408	0.276	5.96	4.856	-1.158	0.25107	-73.611
18	0.44835	0.40375	5.291	3.676	-0.86975	-0.20654	-57.797
19	0.3563	0.605	4.773	2.353	-0.50098	-0.46155	-34.664
20	0.31415	0.72025	4.582	1.701	-0.14468	-0.46155	-6.3611e-15
21	NaN	NaN	NaN	NaN	0.091444	-0.25434	41.267

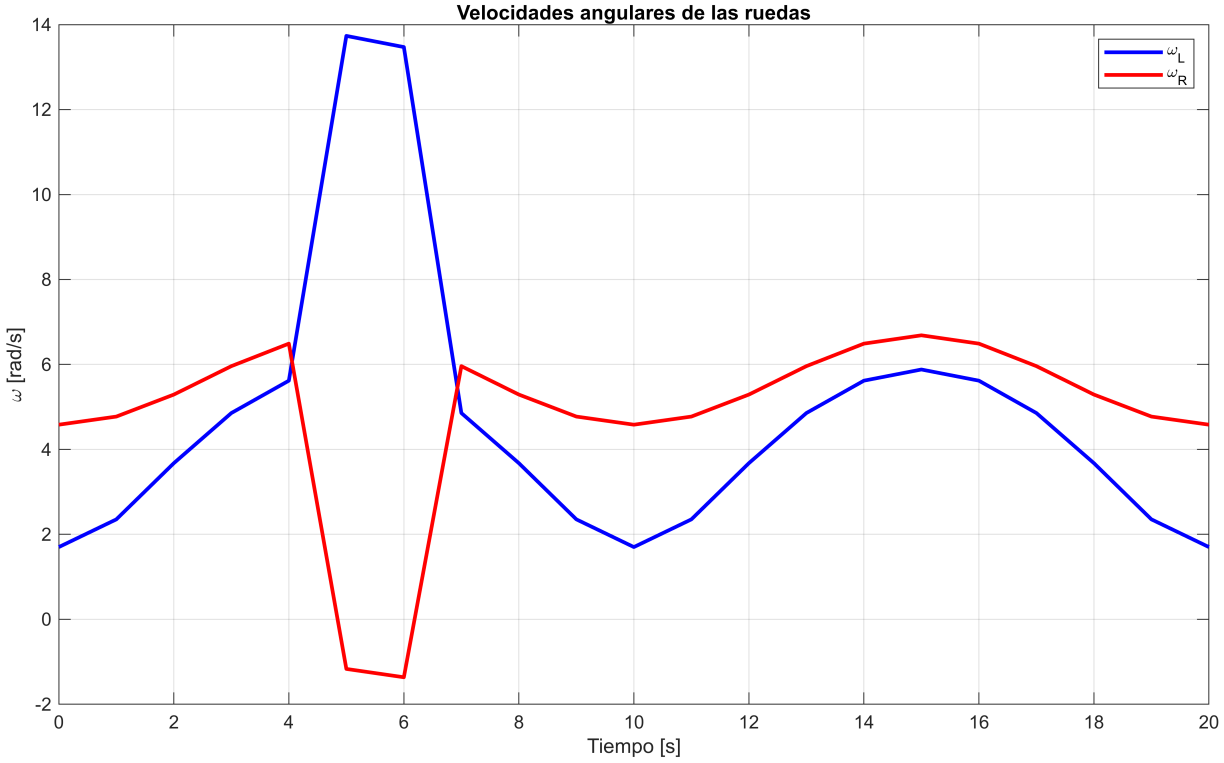
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% GRAFICAS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

tamPantalla = get(0, 'ScreenSize');
tiempoVec = tiempoSim;

% --------- v(t) y w(t) ---------
figVelocidades = figure;
set(figVelocidades, 'position', tamPantalla);

subplot(211)
plot(tiempoVec, velLinealFull, 'b', 'LineWidth', 2), grid on
xlabel('Tiempo [s]')
ylabel('v (m/s)')
legend('v')

subplot(212)
plot(tiempoVec, velAngularFull, 'r', 'LineWidth', 2), grid on
xlabel('Tiempo [s]')
ylabel('w (rad/s)')
legend('w')
```




```
% --------- Pose del robot: x(t), y(t), theta(t) ---------
figPose = figure;
set(figPose, 'position', tamPantalla);

subplot(311)
plot(tiempoVec, posX, 'b', 'LineWidth', 2), grid on
xlabel('Tiempo [s]')
ylabel('x [m]')
title('Posición X')

subplot(312)
plot(tiempoVec, posY, 'g', 'LineWidth', 2), grid on
xlabel('Tiempo [s]')
ylabel('y [m]')
title('Posición Y')

subplot(313)
plot(tiempoVec, anguloTheta, 'm', 'LineWidth', 2), grid on
xlabel('Tiempo [s]')
ylabel('\theta [rad]')
title('Orientación')
```

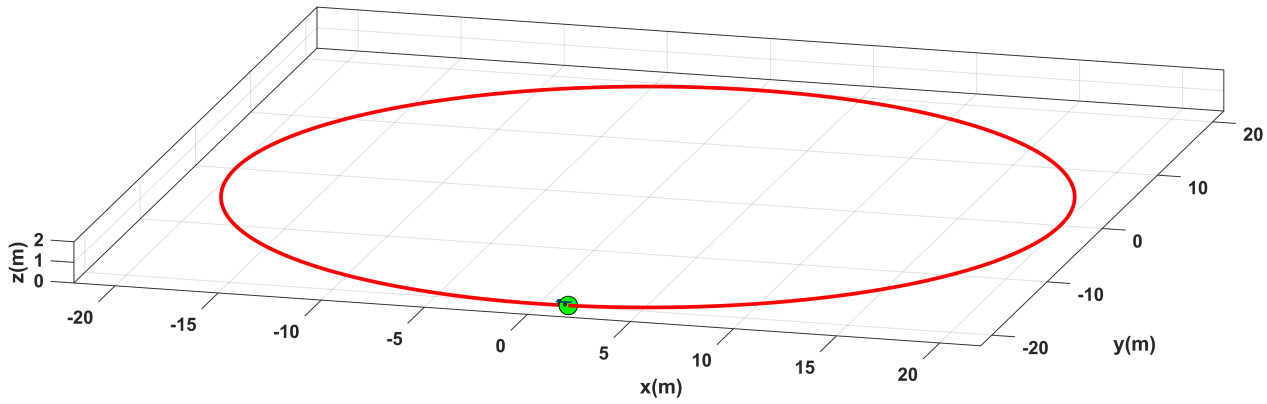



```
% --------- Velocidades angulares de ruedas ---------
figRuedas = figure;
set(figRuedas, 'position', tamPantalla);

plot(tiempoVec, omegaIzqFull, 'b', 'LineWidth', 2), hold on
plot(tiempoVec, omegaDerFull, 'r', 'LineWidth', 2), grid on
xlabel('Tiempo [s]')
ylabel('\omega [rad/s]')
legend('\omega_L', '\omega_R')
title('Velocidades angulares de las ruedas')
```



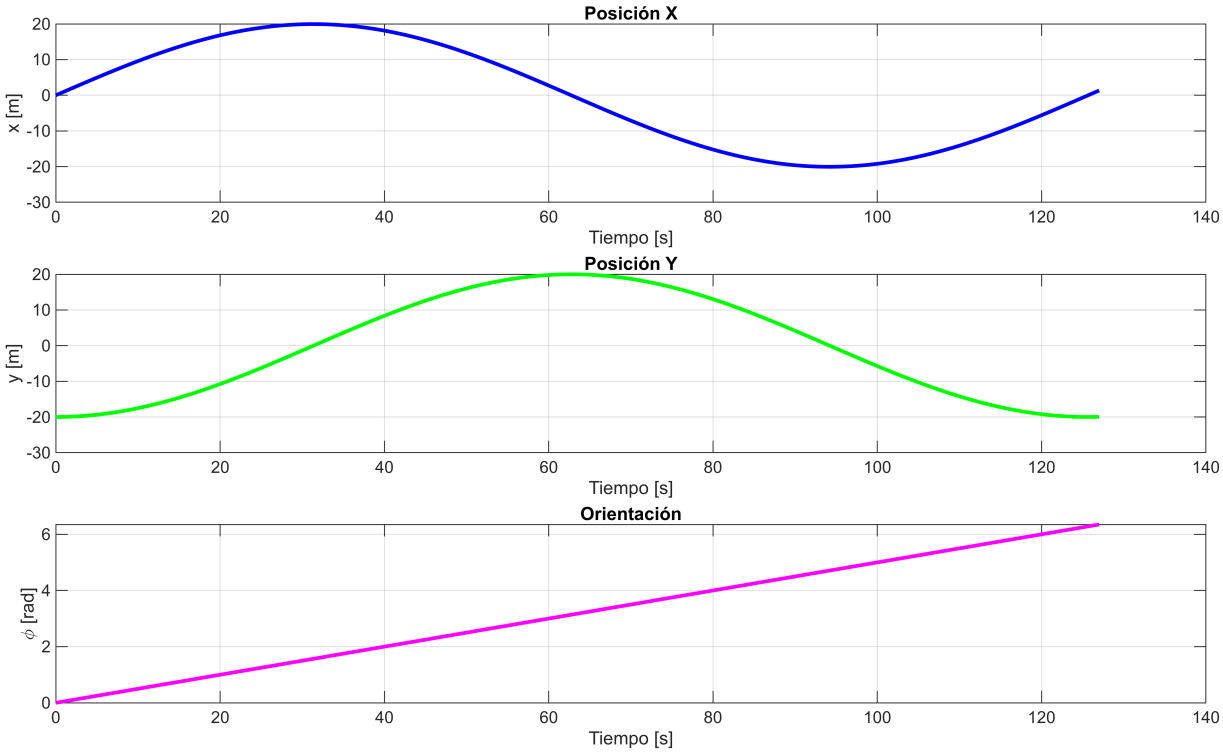

```
clear; clc;

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

tiempo_final = 127;            % Tiempo total de simulación en segundos
delta_tiempo = 0.1;            % Tiempo de muestreo
vector_tiempo = 0:delta_tiempo:tiempo_final;  % Vector de tiempo
num_muestras = length(vector_tiempo);         % Total de muestras

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

posX = zeros(1, num_muestras+1);   % Posición x
posY = zeros(1, num_muestras+1);   % Posición y
orientacion = zeros(1, num_muestras+1); % Ángulo de orientación

posX(1) = 0;
posY(1) = -20;
orientacion(1) = 0;

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

trayX = zeros(1, num_muestras+1);
trayY = zeros(1, num_muestras+1);
trayX(1) = posX(1);
trayY(1) = posY(1);
```

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

vel_lineal_ref = ones(1, num_muestras);          % Velocidad lineal constante
vel_angular_ref = ones(1, num_muestras) / 20;    % Velocidad angular constante

%%%%%%%%%%%%%%%%%%%%%%%% BUCLE DE SIMULACIÓN %%%%%%%%%%%%%%%%%%%%%%%%%%

for k = 1:num_muestras
    orientacion(k+1) = orientacion(k) + vel_angular_ref(k) * delta_tiempo;

    dx = vel_lineal_ref(k) * cos(orientacion(k+1));
    dy = vel_lineal_ref(k) * sin(orientacion(k+1));

    posX(k+1) = posX(k) + dx * delta_tiempo;
    posY(k+1) = posY(k) + dy * delta_tiempo;

    trayX(k+1) = posX(k+1);
    trayY(k+1) = posY(k+1);
end

%%%%%%%%%%%%%%%%%%%%%%%%%% SIMULACIÓN 3D %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figura_sim = figure;
set(figura_sim, 'Color', 'white');
set(gca, 'FontWeight', 'bold');
pantalla = get(0, 'ScreenSize');
set(figura_sim, 'Position', pantalla);
camlight('headlight');
axis equal; grid on; box on;
xlabel('x(m)'); ylabel('y(m)'); zlabel('z(m)');
view([15 15]);
axis([-22 22 -22 22 0 2]);

escala_robot = 4;
MobileRobot_5;
robot_plot = MobilePlot_4(posX(1), posY(1), orientacion(1), escala_robot); hold on;
trayectoria_plot = plot3(trayX(1), trayY(1), 0, 'r', 'lineWidth', 2);
plot3(posX(end), posY(end), 0, 'ko', 'MarkerFaceColor', 'g', 'MarkerSize', 10);

paso_anim = 1;
for k = 1:paso_anim:num_muestras
    delete(robot_plot);
    delete(trayectoria_plot);
    robot_plot = MobilePlot_4(posX(k), posY(k), orientacion(k), escala_robot);
    trayectoria_plot = plot3(trayX(1:k), trayY(1:k), zeros(1,k), 'r', 'lineWidth',
2);
    pause(delta_tiempo);
end
```

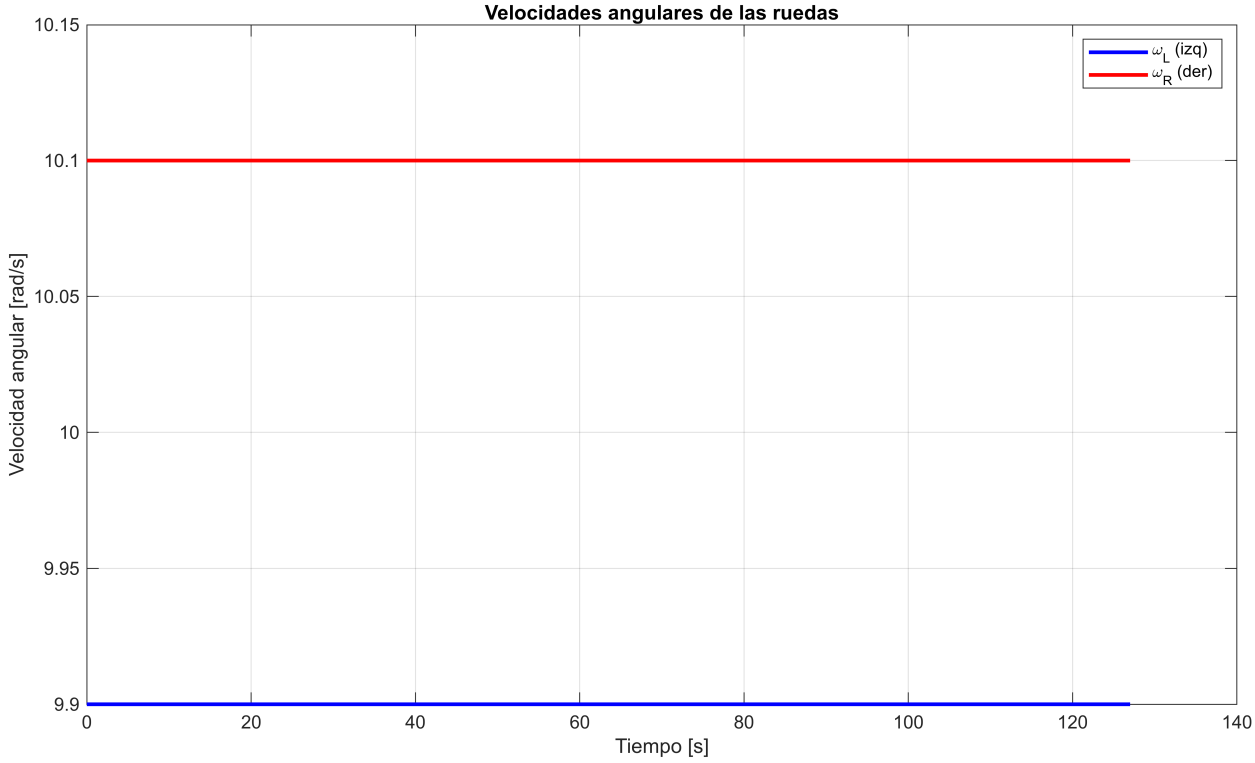



```
%%%%%%%%%%%%%%%%%%% CÁLCULO DE VELOCIDADES ANGULARES %%%%%%%%%%%%%%%%%%%

radio_rueda = 0.1; % Radio de las ruedas
dist_ruedas = 0.4; % Distancia entre ruedas

omega_izq = zeros(1, num_muestras);
omega_der = zeros(1, num_muestras);

for k = 1:num_muestras
    omega_izq(k) = (2 * vel_lineal_ref(k) - vel_angular_ref(k) * dist_ruedas) / (2
* radio_rueda);
    omega_der(k) = (2 * vel_lineal_ref(k) + vel_angular_ref(k) * dist_ruedas) / (2
* radio_rueda);
end

%%%%%%%%%%%%%%%%%%% GRAFICAR POSE Y VELOCIDADES ANGULARES %%%%%%%%%%%%%%%%%%%

figura_pose = figure;
set(figura_pose, 'Position', pantalla);

subplot(311)
plot(vector_tiempo, posX(1:num_muestras), 'b', 'LineWidth', 2); grid on;
xlabel('Tiempo [s]'); ylabel('x [m]'); title('Posición X');

subplot(312)
```

```
plot(vector_tiempo, posY(1:num_muestras), 'g', 'LineWidth', 2); grid on;
xlabel('Tiempo [s]'); ylabel('y [m]'); title('Posición Y');

subplot(313)
plot(vector_tiempo, orientacion(1:num_muestras), 'm', 'LineWidth', 2); grid on;
xlabel('Tiempo [s]'); ylabel('\phi [rad]'); title('Orientación');
```




```
figura_ruedas = figure;
set(figura_ruedas, 'Position', pantalla);
plot(vector_tiempo, omega_izq, 'b', 'LineWidth', 2); hold on;
plot(vector_tiempo, omega_der, 'r', 'LineWidth', 2); grid on;
xlabel('Tiempo [s]');
ylabel('Velocidad angular [rad/s]');
legend('\omega_L (izq)', '\omega_R (der)');
title('Velocidades angulares de las ruedas');
```




```
%%%%%%%%%%%%%%%%%% TABLA MUESTREADA CADA 5 SEGUNDOS %%%%%%%%%%%%%%%%%%

intervalo_muestreo = 5;
idx_muestreo = 1 : intervalo_muestreo/delta_tiempo : num_muestras;
tiempos_muestreo = vector_tiempo(idx_muestreo)';
omegaR_muestreado = omega_der(idx_muestreo)';
omegaL_muestreado = omega_izq(idx_muestreo)';

tabla_velocidades = table(tiempos_muestreo, omegaR_muestreado, omegaL_muestreado,
...
    'VariableNames', {'Tiempo_s', 'omega_R_rad_s', 'omega_L_rad_s'});

disp('Tabla de velocidades angulares muestreadas cada 5 segundos:')
```

```
Tabla de velocidades angulares muestreadas cada 5 segundos:
```

```
disp(tabla_velocidades)
```

Tiempo_s	omega_R_rad_s	omega_L_rad_s
0	10.1	9.9
5	10.1	9.9
10	10.1	9.9
15	10.1	9.9
20	10.1	9.9
25	10.1	9.9
30	10.1	9.9

```
 35          10.1            9.9
 40          10.1            9.9
 45          10.1            9.9
 50          10.1            9.9
 55          10.1            9.9
 60          10.1            9.9
 65          10.1            9.9
 70          10.1            9.9
 75          10.1            9.9
 80          10.1            9.9
 85          10.1            9.9
 90          10.1            9.9
 95          10.1            9.9
100          10.1            9.9
105          10.1            9.9
110          10.1            9.9
115          10.1            9.9
120          10.1            9.9
125          10.1            9.9
```