Instituto Tecnológico de Monterrey

Pose y Velocidades Angulares

Daniel Castillo López - A01737357
Emmanuel Lechuga Arreola - A01736241
Paola Rojas Domínguez - A01737136

Robot movil diferencial

A partir del tópico /cmd_velse publican las velocidades lineal V y angular W, las cuales son parámetros de entrada del nodo Puzzlebot Sim. A la salida del nodo Puzzlebot Sim obtenemos la pose y las velocidades angulares de las ruedas WR y WL

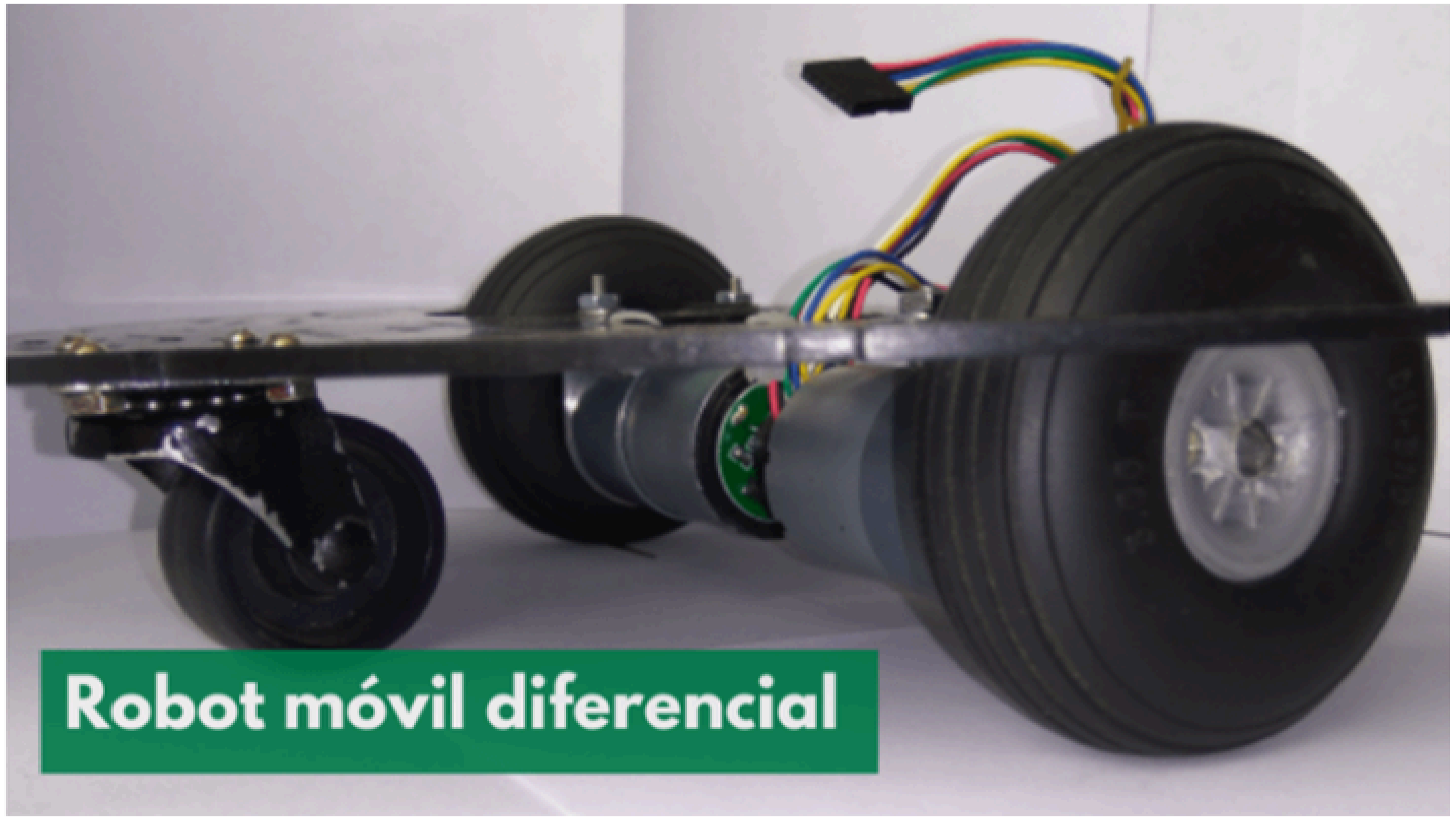
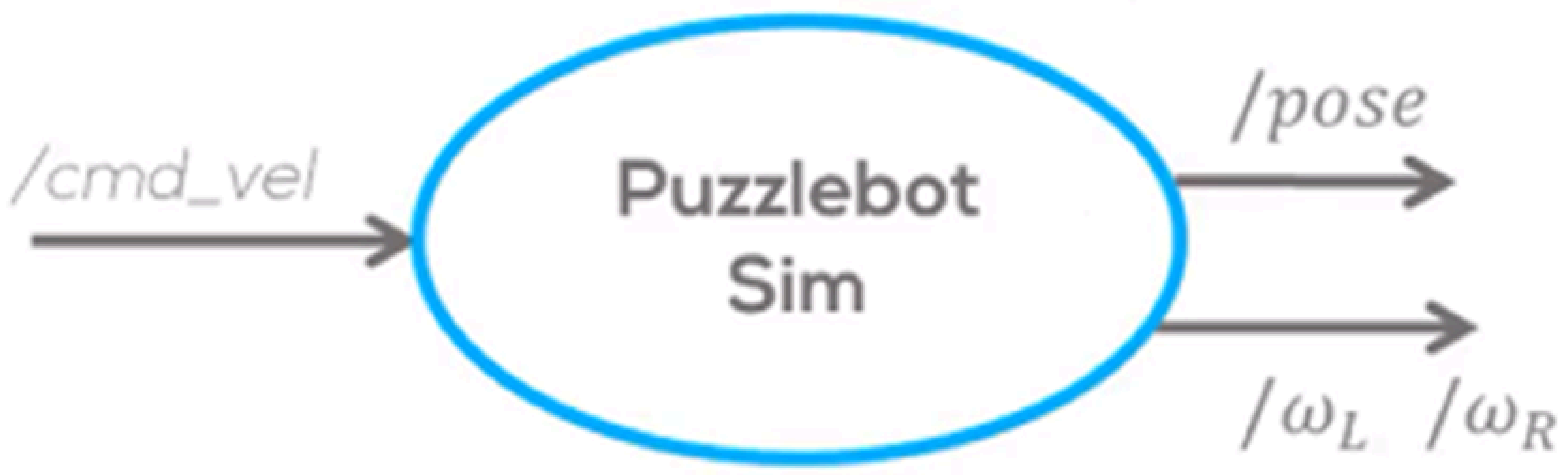
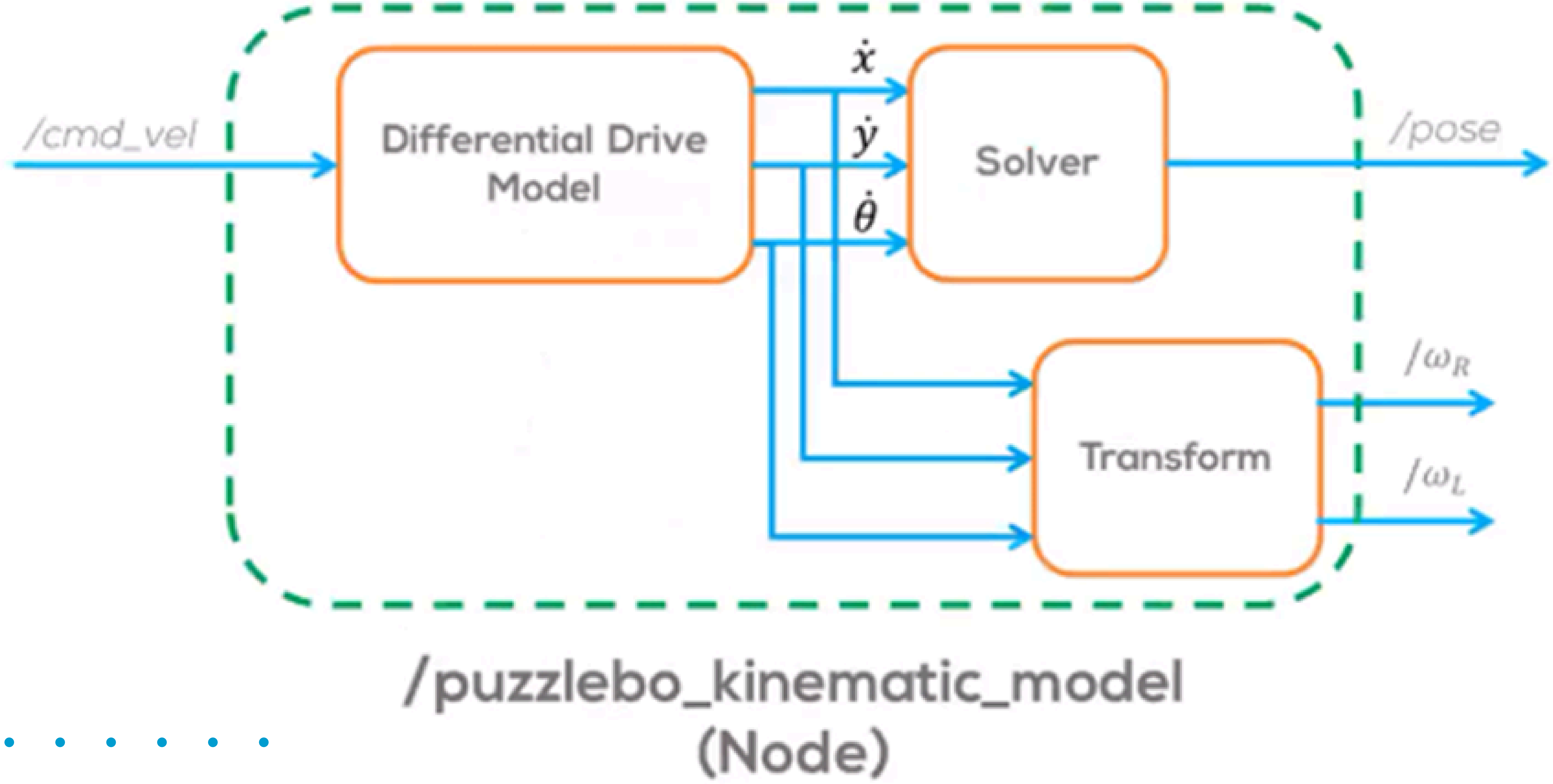


Diagrama de entrada y salidas



Transformación interna del nodo



Tiempo y condiciones iniciales

Establece los parámetros de tiempo, incluyendo el número de muestras, la duración total de la simulación y el intervalo de muestreo.

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% TIEMPO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

N = length(u);       % Muestras
ts = 0.1;            % Tiempo de muestreo en segundos (s)
t = linspace(0, ts, N);  % Vector de tiempo


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

x1 = zeros (1,N+1);  % Posición en el centro del eje que une las ruedas (eje x)
y1 = zeros (1,N+1);  % Posición en el centro del eje que une las ruedas (eje y)
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

Asigna la posición del punto de control del robot, que es una referencia utilizada para describir y seguir su trayectoria.
Además de utilizarse para la simulación o representación en 3D (coordenada 0,0).

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%% 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
```

Bucle de simulación

```
%%%%%%%%%%%%%%%%%%%%%%%%% 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
```

Lógica del bucle.

- Actualiza la orientación (1-l).
- Calcula velocidad en X (2-l).
- Calcula velocidad en Y (3-l).
- Actualiza la posición de X (4-l).
- Actualiza la posición de Y (5-l).
- Actualiza la trayectoria X y Y(6,7-l).

Simulación visual.

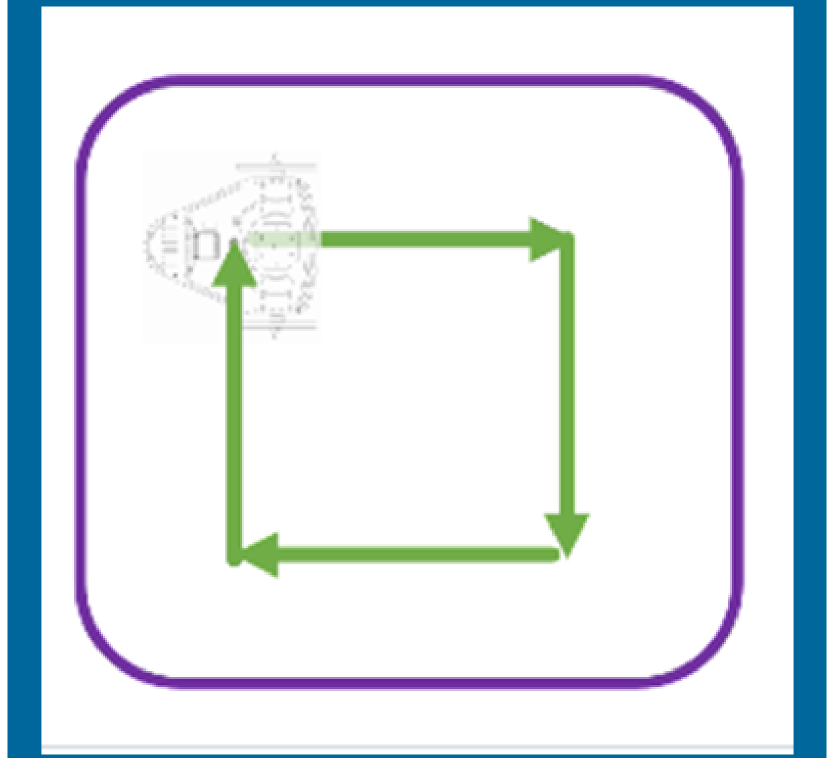
```
% d) Bucle de simulacion de movimiento del robot

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
```

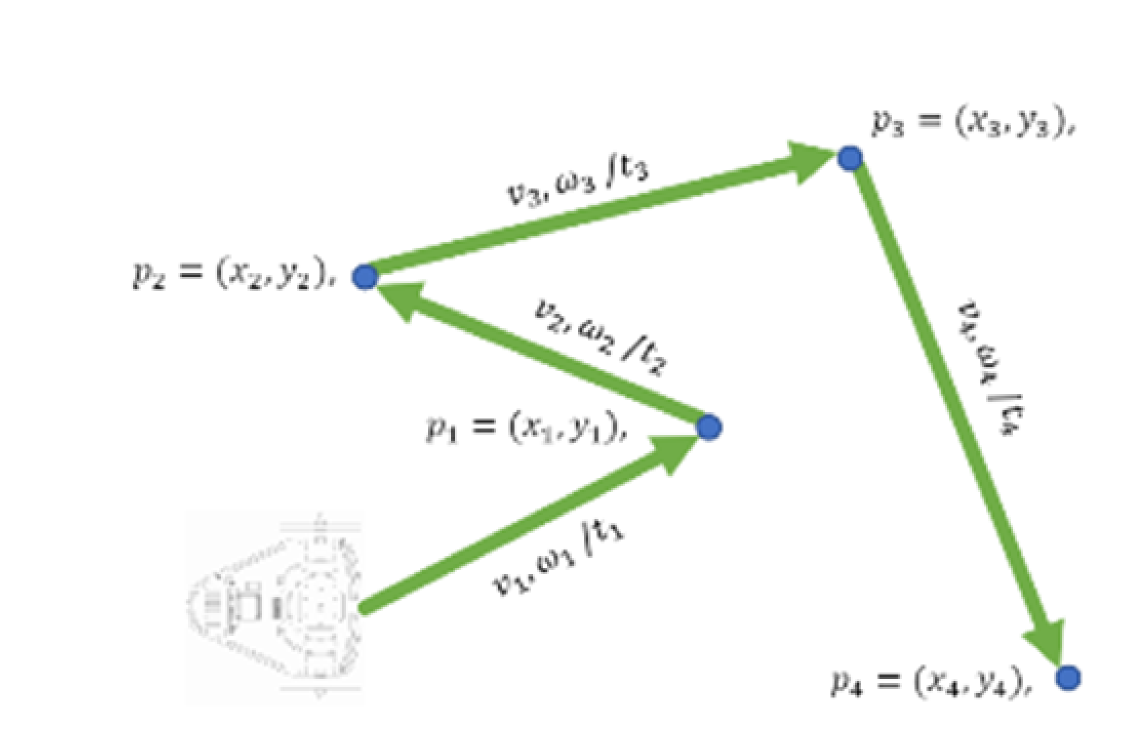
Lógica del bucle.

- Actualización de posición del robot (1-l).
- Borramos el robot anterior y su trayectoria (2,3-l).
- Redibujamos el robot y trayectoria (4,5-l).
- Usamos una pausa para agregar dramatismo de movimiento (6-l).

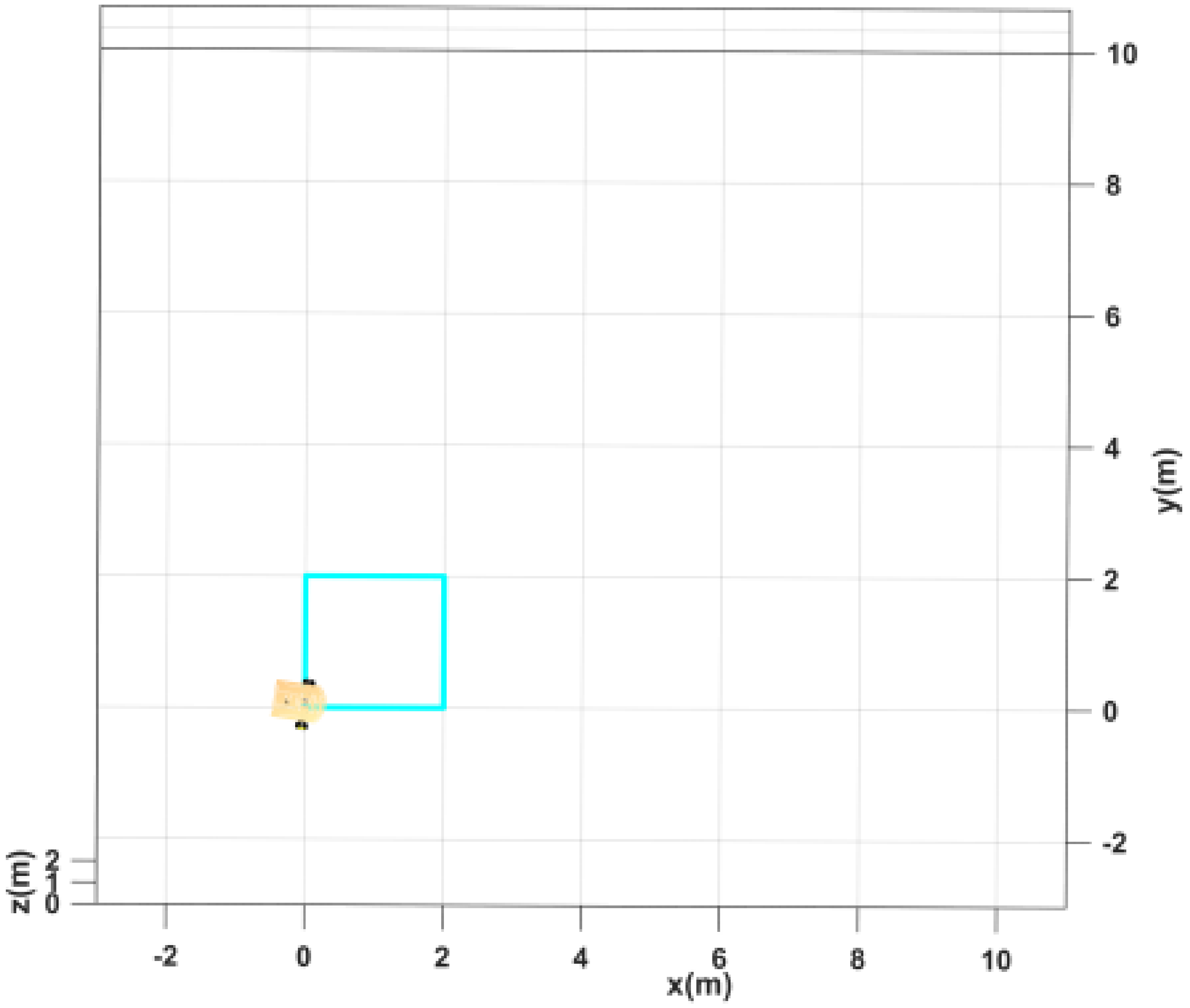
Velocidad referencia

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

u = [2*ones(1, 10), zeros(1, 10), 2*ones(1, 10), zeros(1, 10),...
    2*ones(1,10), zeros(1, 10), 2*ones(1,10), zeros(1, 10)];

w = [zeros(1, 10), pi/2.*ones(1, 10),zeros(1,10), pi/2.*ones(1, 10),...
    zeros(1,10), pi/2.*ones(1, 10),zeros(1,10), pi/2.*ones(1, 10)];  % Velo
```

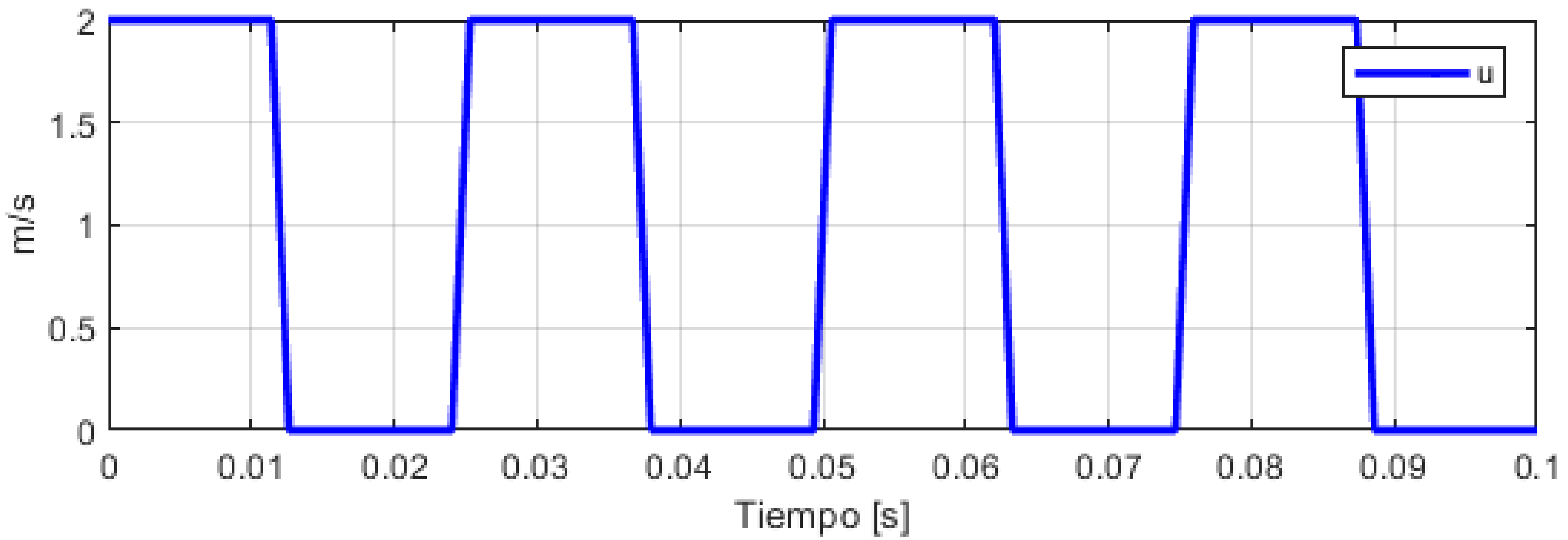




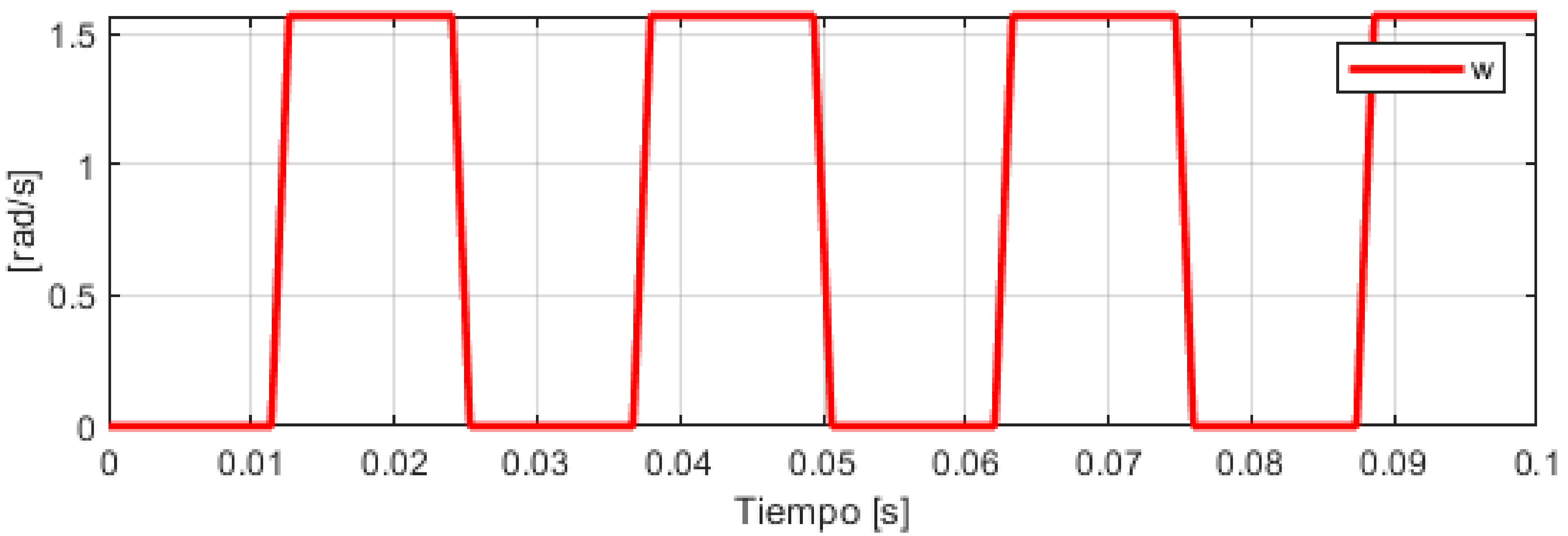

```
%%%%%%%%%%%%%%%%%%%%%% VELOCIDADES DE REFERENCIA %%%%%%%%%%%%%%%%%%%%%%%%%%

u = [zeros(1,10),2*ones(1, 10), zeros(1, 10), 1*ones(1, 10), zeros(1, 10),...
    2*ones(1,10), zeros(1, 10), 4*ones(1,10)];

w = [deg2rad(45).*ones(1, 10),zeros(1,10), deg2rad(100).*ones(1, 10),...
    zeros(1,10), deg2rad(-120).*ones(1, 10),zeros(1,10), deg2rad(-100).*ones(1, 10),zeros(1,10)];
```



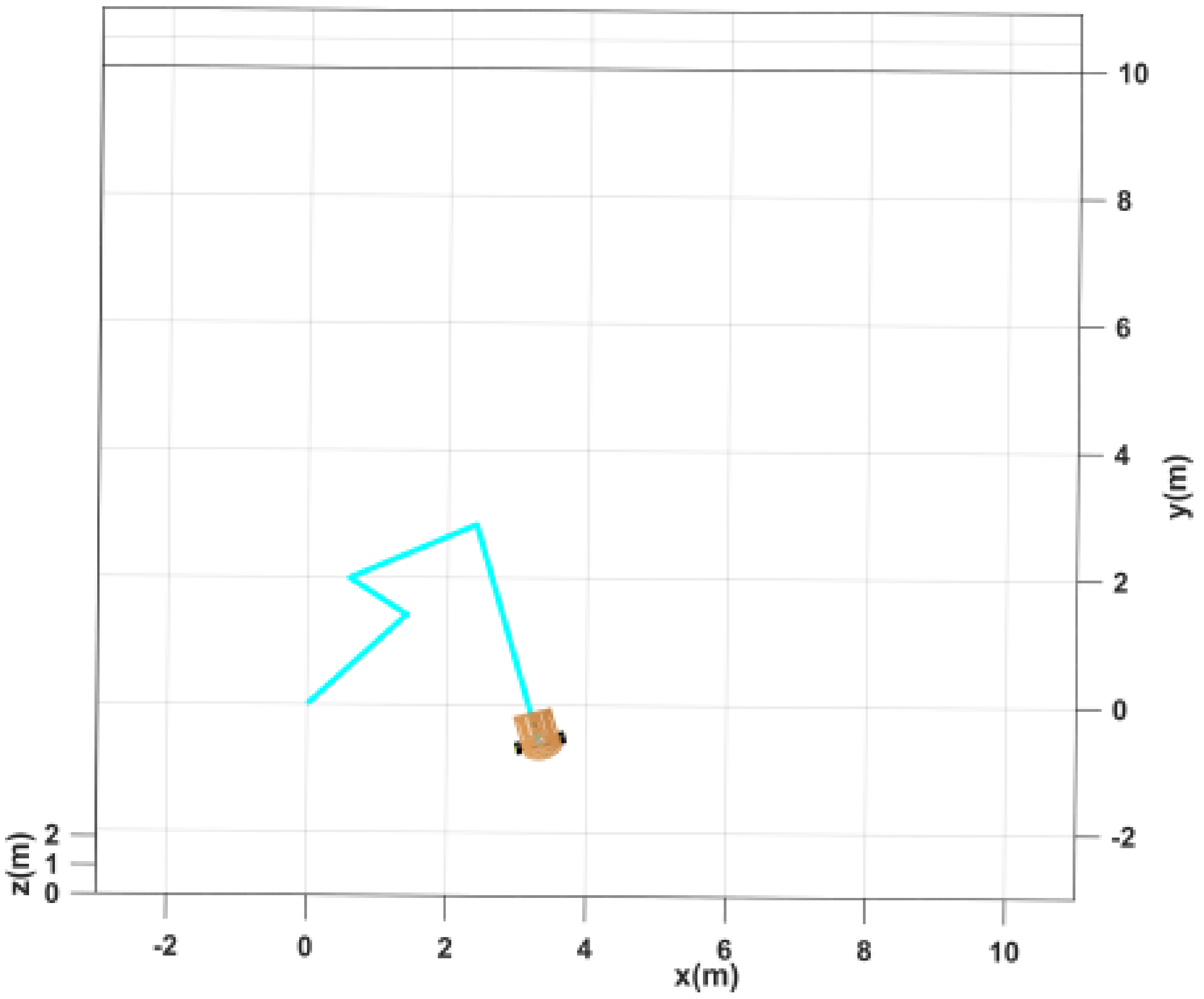
x(m)
y(m)
z(m)



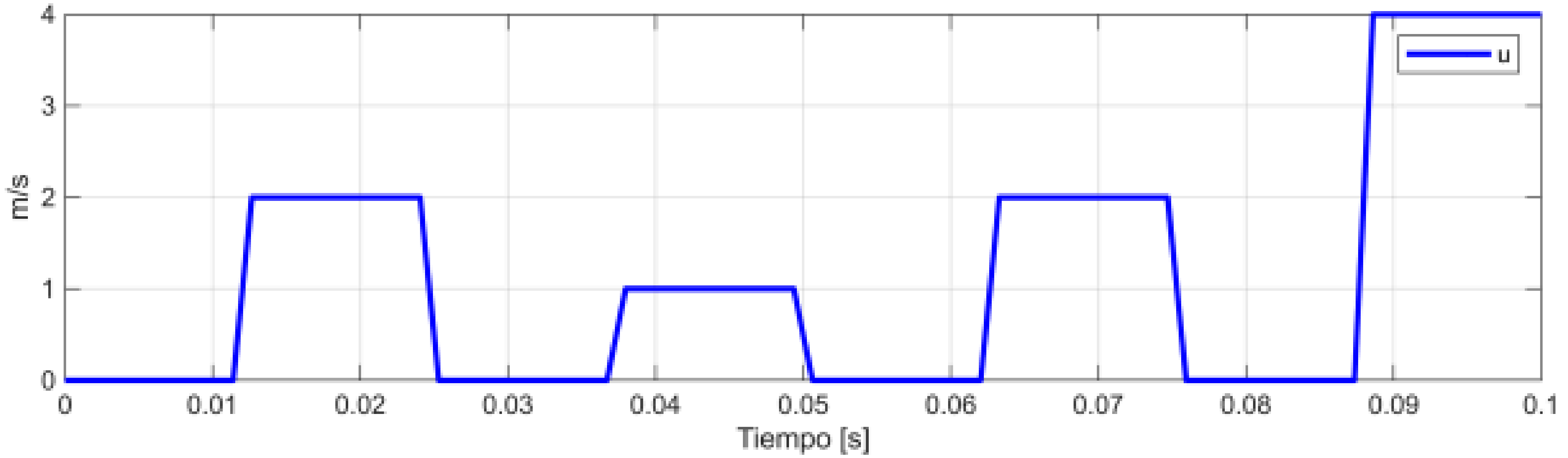
u
m/s
Tiempo [s]



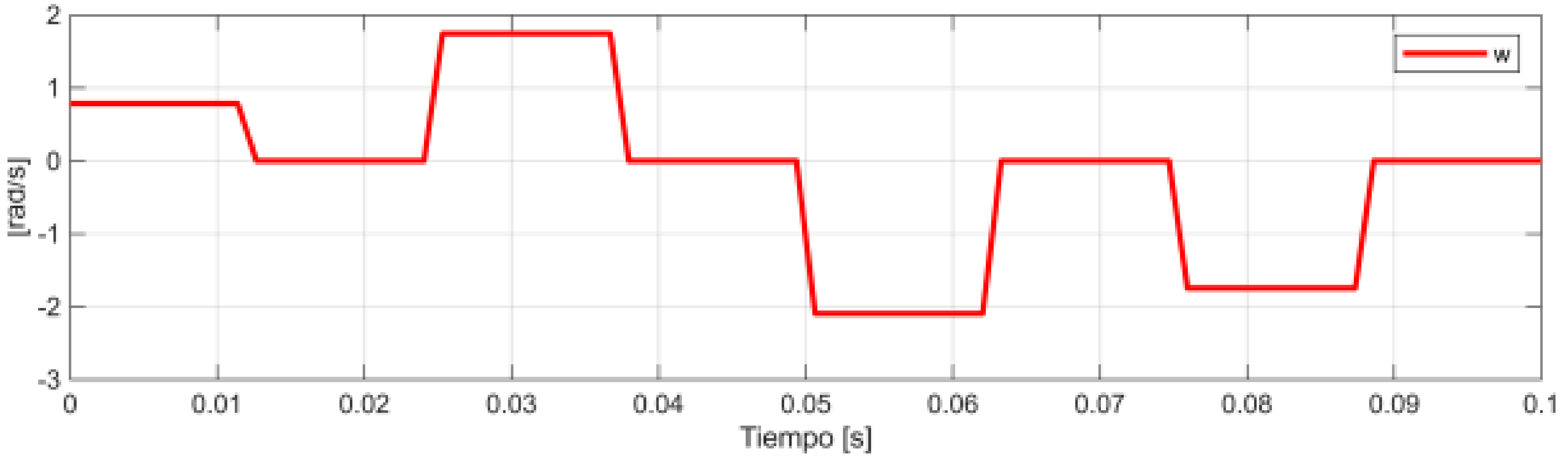
w
[rad/s]
Tiempo [s]



x(m)
y(m)
z(m)



u
m/s
Tiempo [s]



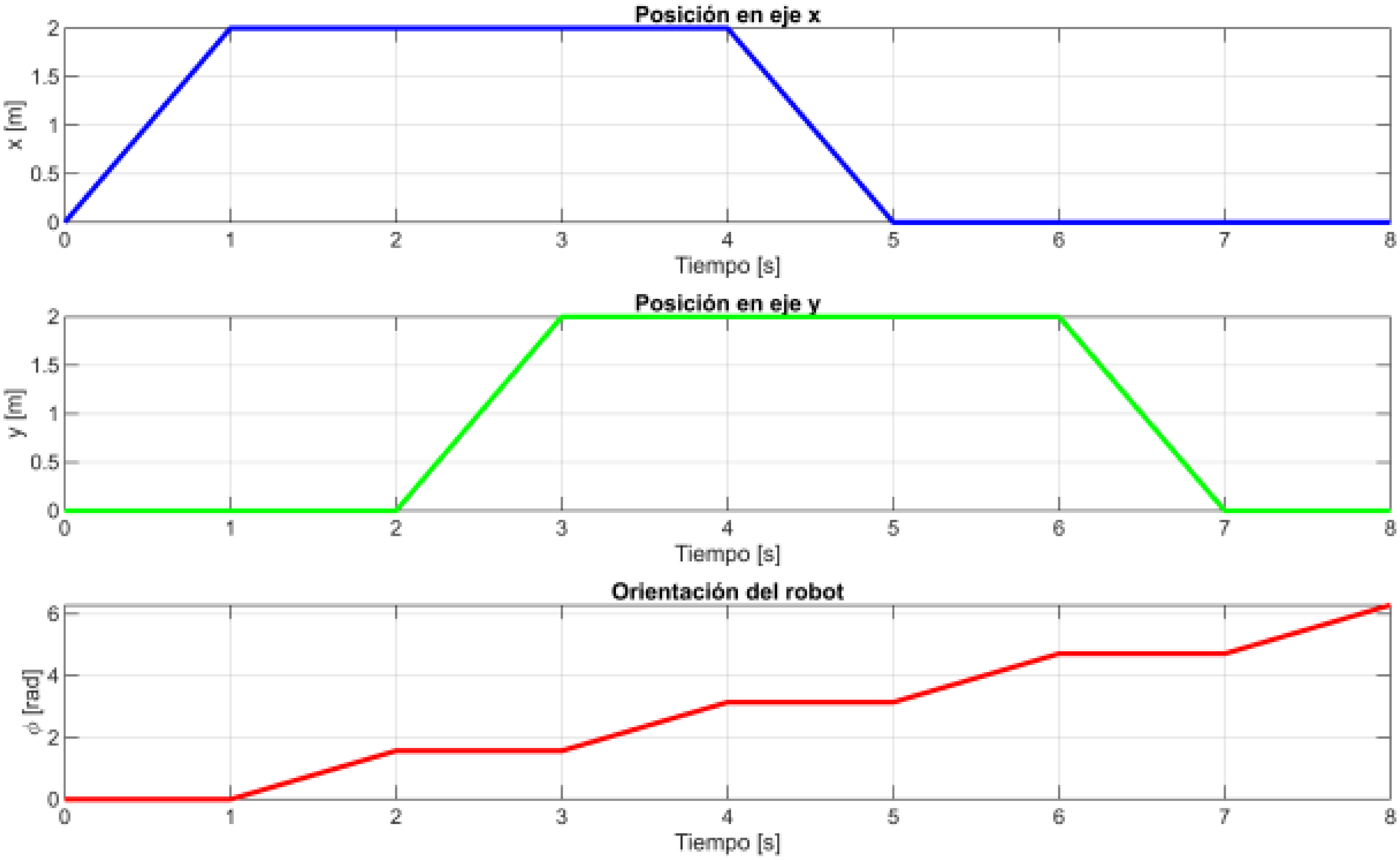
w
[rad/s]
Tiempo [s]

Solver

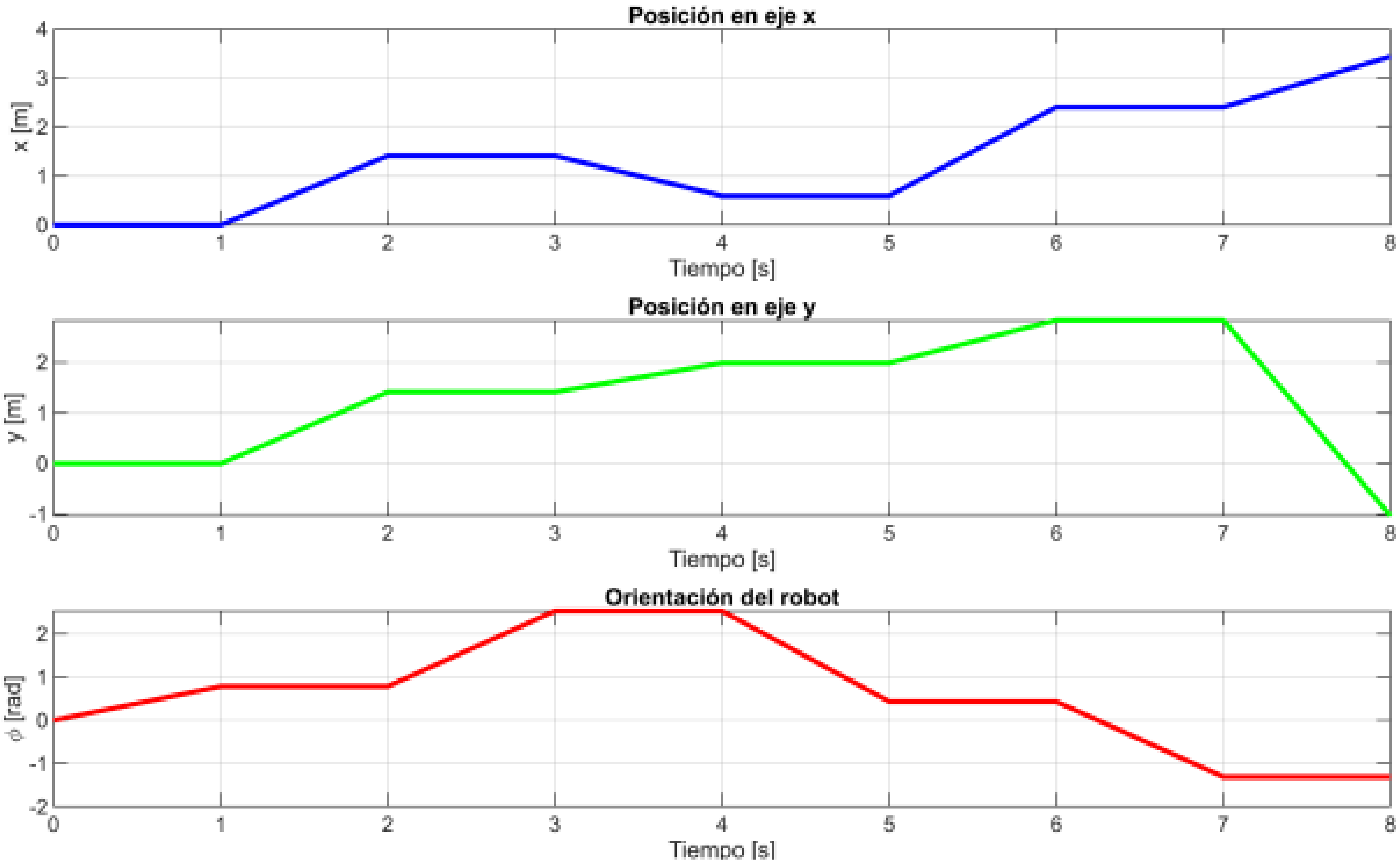
$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} V * \cos(\theta) \\ V * \mathrm{sen}(\theta) \\ W \end{bmatrix}$$

```
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));
%phip = w(t);

x1(k+1)=x1(k) + xp1*ts ; % Integral numérica (método de Euler)
y1(k+1)=y1(k) + yp1*ts ; % Integral numérica (método de Euler)
```

Cuadrado


Posición en eje x
x [m]
Tiempo [s]
Posición en eje y
y [m]
Tiempo [s]
Orientación del robot
ϕ [rad]
Tiempo [s]

Zig zag



Posición en eje x
x [m]
Tiempo [s]
Posición en eje y
y [m]
Tiempo [s]
Orientación del robot
ϕ [rad]
Tiempo [s]

Transform

$$V = r\frac{W_R + W_L}{2}$$

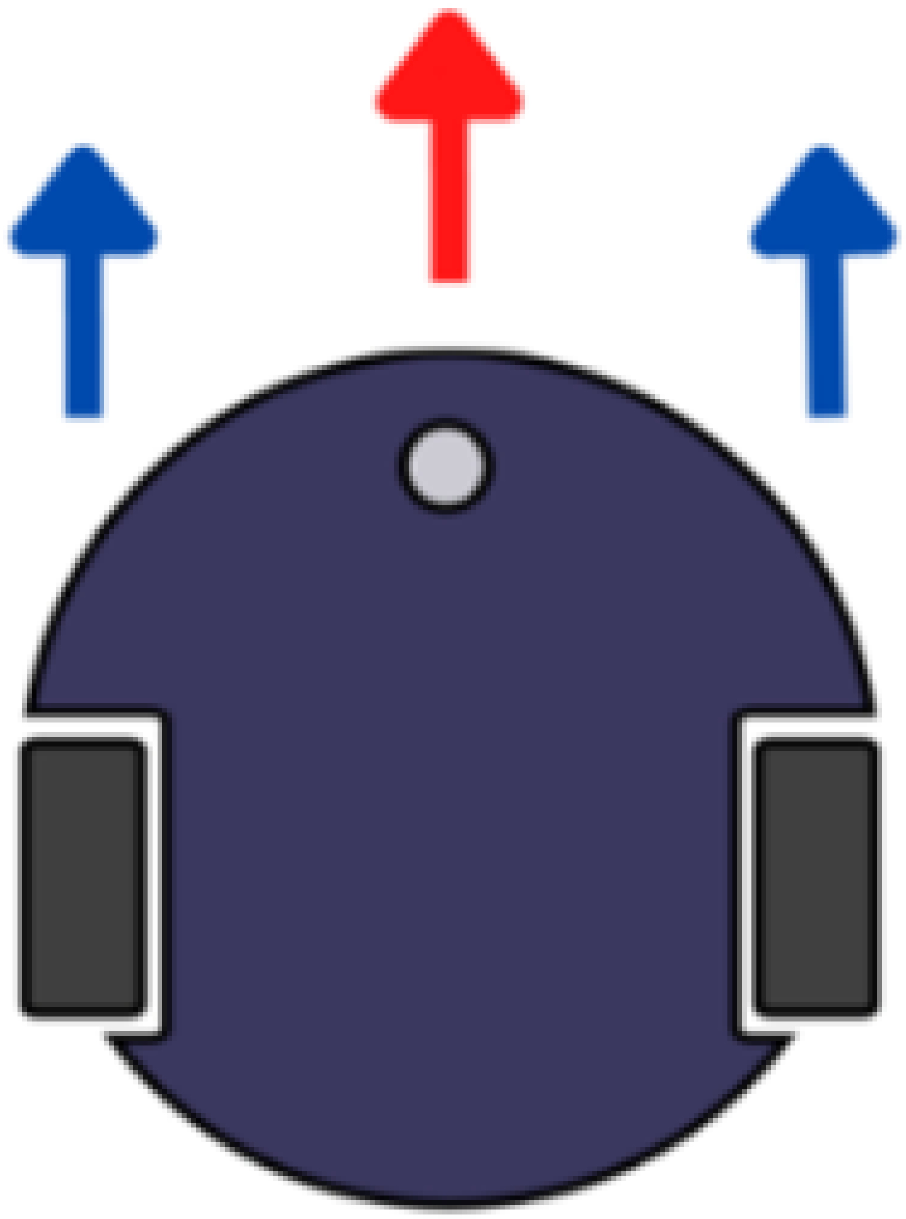
$$W = r\frac{W_R - W_L}{l}$$

Donde:

- W_R = velocidad angular de la rueda derecha
- W_L = velocidad angular de la rueda izquierda
- V = velocidad lineal del robot
- W = velocidad angular del robot
- l = distancia entre ruedas
- r = radio de las ruedas

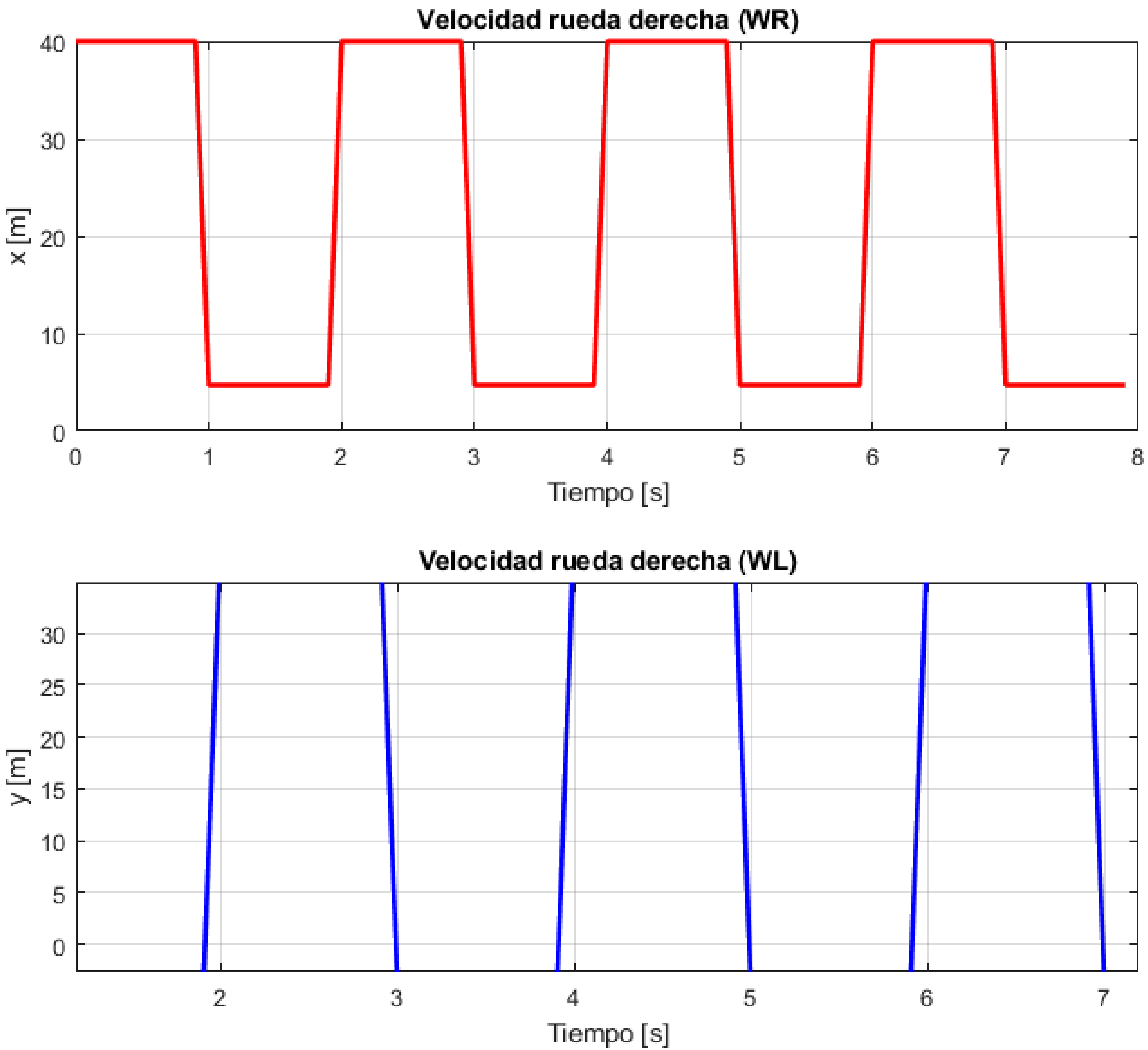
Transform

$$W_R = \frac{2V+Wl}{2r}$$



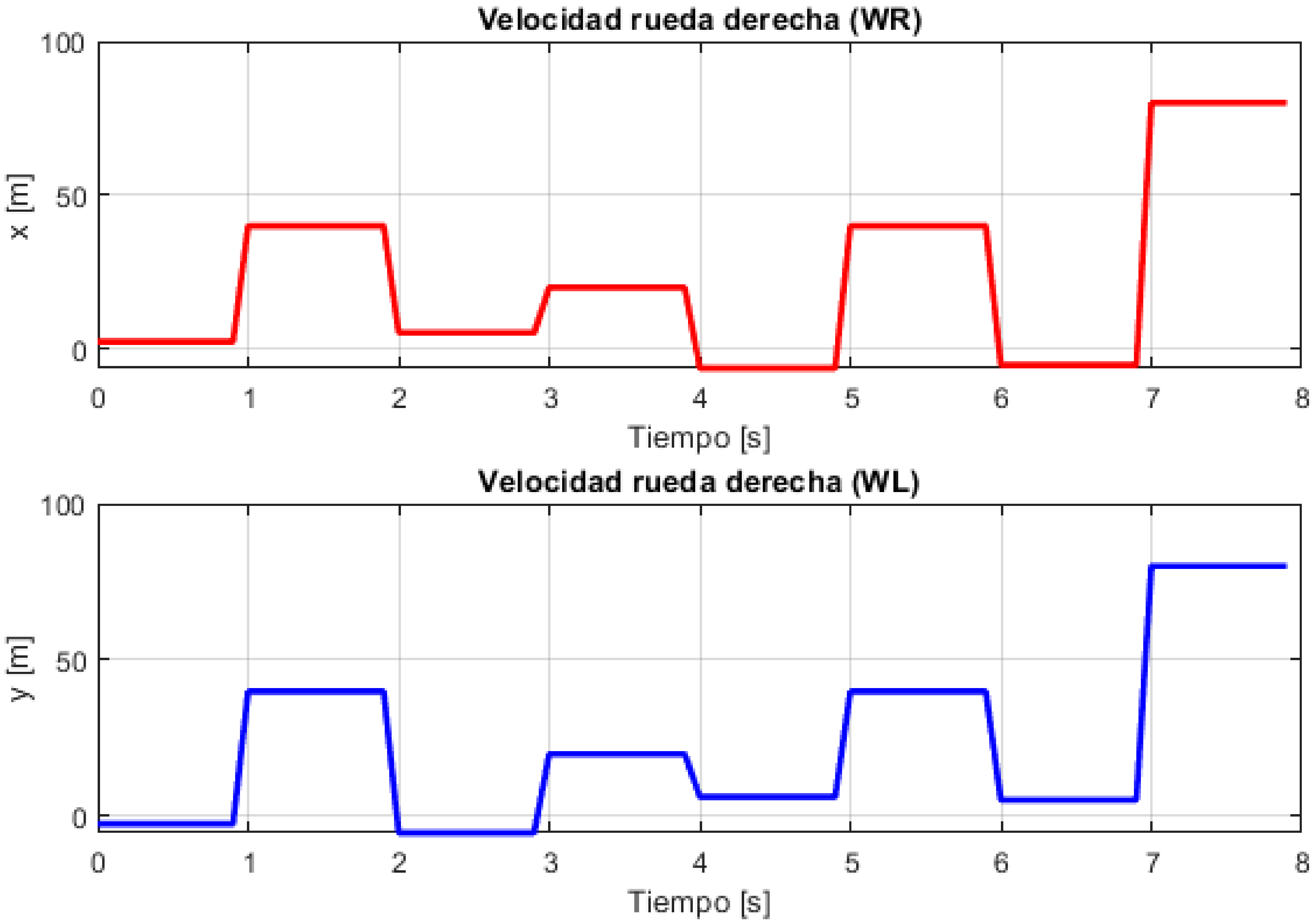
$$W_L = \frac{2V-Wl}{2r}$$

Cuadrado



Velocidad rueda derecha (WR)
x [m]
40
30
20
10
0
0
1
2
3
4
5
6
7
8
Tiempo [s]
Velocidad rueda derecha (WL)
y [m]
30
25
20
15
10
5
0
2
3
4
5
6
7
Tiempo [s]

Zig zag


Velocidad rueda derecha (WR)
x [m]
100
50
0
0 1 2 3 4 5 6 7 8
Tiempo [s]
Velocidad rueda derecha (WL)
y [m]
100
50
0
0 1 2 3 4 5 6 7 8
Tiempo [s]