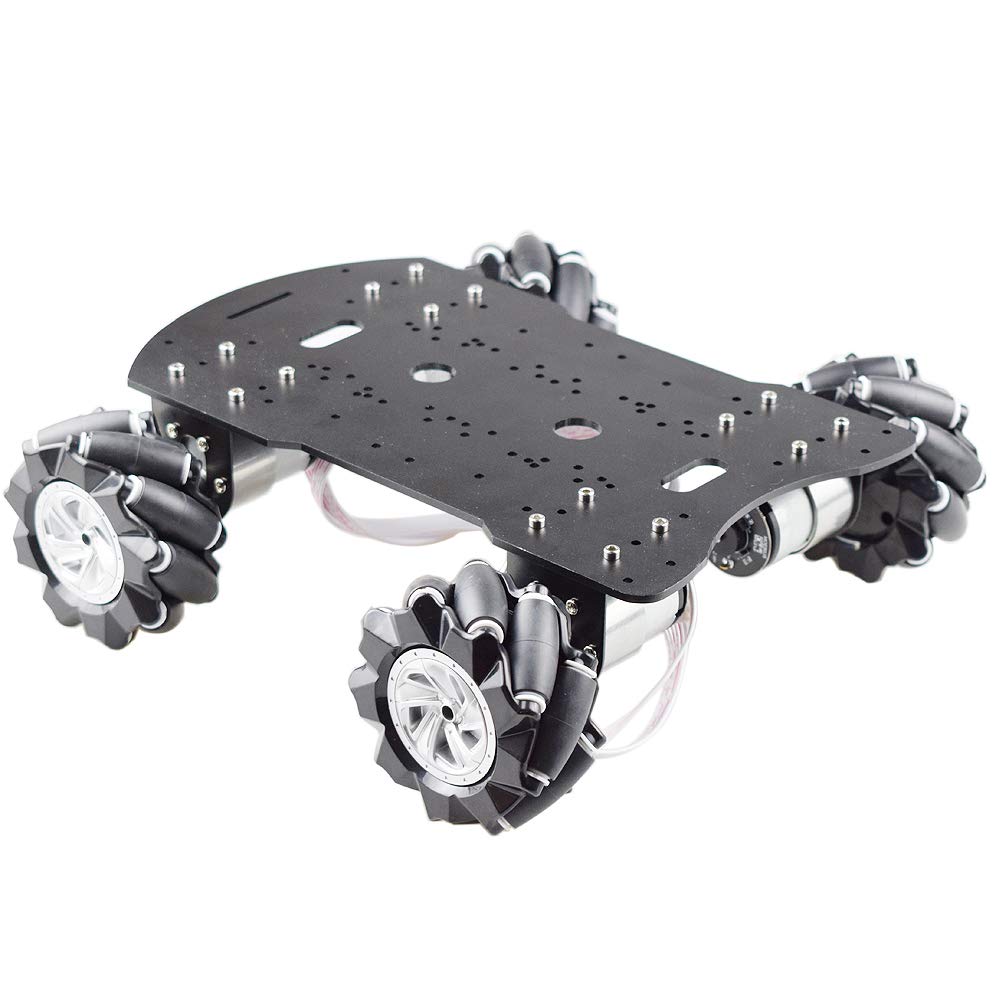
# Sistema de control para Plataforma robótica

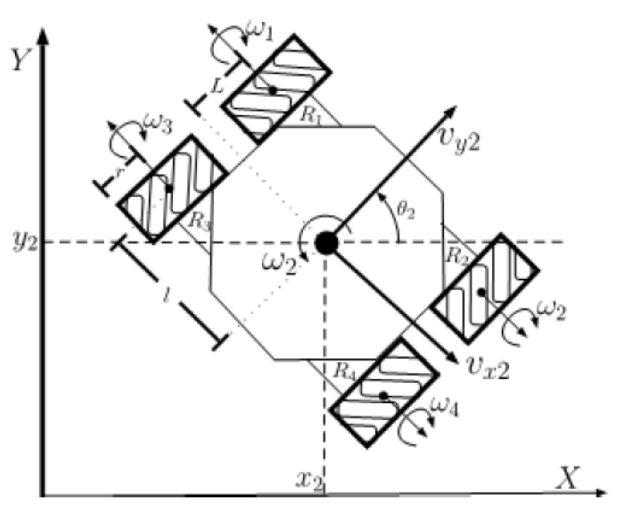
**NOTA: Hay elementos por revisar, modificar y organizar**

Se tiene un robot de 4 ruedas omnidireccional [Moebius](https://www.amazon.com/-/es/Moebius-22-0-lbs-codificador-Raspberry-proyecto/dp/B07VSW8VTB/ref=pd_cart_vw_crc_2_2/131-6708501-8014833?_encoding=UTF8&pd_rd_i=B07VSW8VTB&pd_rd_r=a0f9daf6-0b14-42e4-bfb1-8138329c5db4&pd_rd_w=W0F5Y&pd_rd_wg=OqBov&pf_rd_p=01004c92-8f40-4f1a-bee8-08cb36dccac2&pf_rd_r=1QE1SKMF00YNA90R6SEF&psc=1&refRID=1QE1SKMF00YNA90R6SEF)

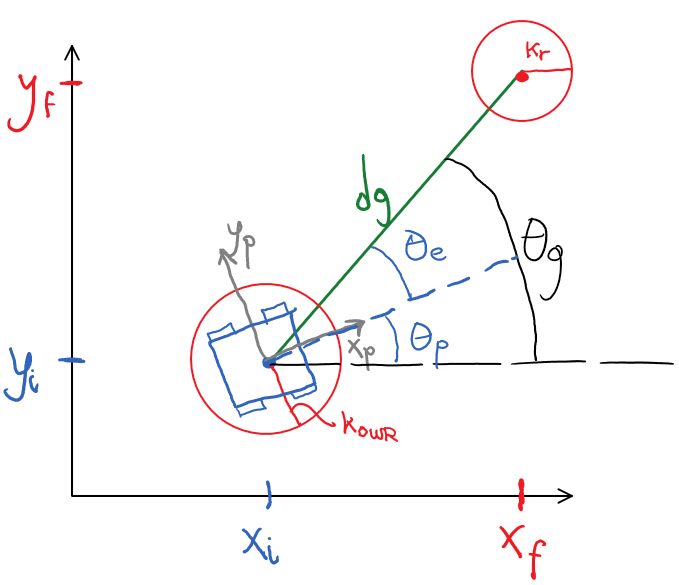


Con un modelo cinemático

Basado en:



Basado en que las entradas para el modelo cinemático son se consideran las ecuaciones para obtener estas variables



Se tiene el robot en con un radio de seguridad que quiere llegar al punto con radio de seguridad

, magnitud de la distancia entre el robot y el punto deseado.

, ángulo del punto deseado desde el robot.

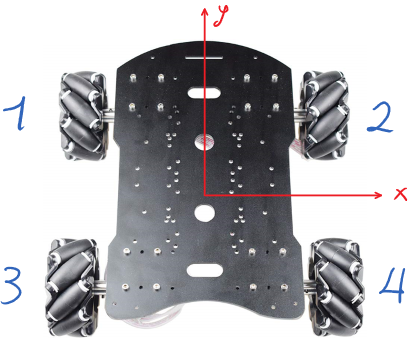
, ángulo de error, la diferencia entre el ángulo del punto de llegada y el ángulo actual del robot.

, velocidad máxima del robot en X

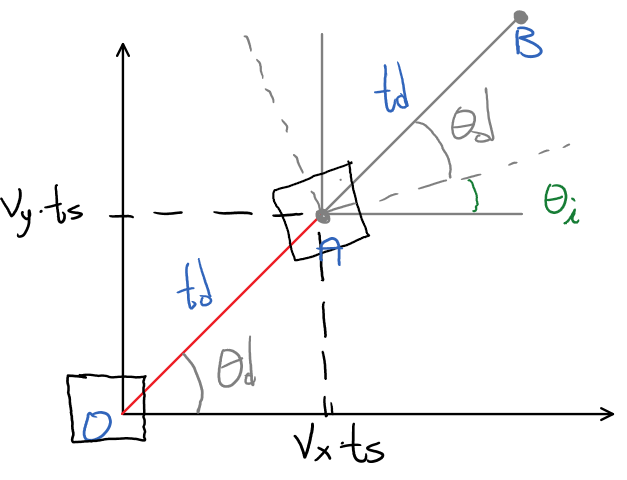
, velocidad máxima del robot en Y

, velocidad angular máxima del robot

Teniendo, , se puede usar el modelo cinemático donde:



Para realizar la simulación y calcular la nueva posición luego cada paso en un tiempo de simulación determinado



, desplazamiento en X

, desplazamiento en Y

, distancia total recorrida

, ángulo de desplazamiento

, ángulo del robot con respecto a la horizontal

Posición inicial O:

Posición final A:

Simulación en Matlab

% Posiciones

posicion\_inicial = [0 0]; % [m]

posicion\_final = [2 1]; % [m]

t\_i = pi/1.5; % Rotación inicial del robot [rad]

% Se obtienen velocidades iniciales a partir de los puntos dados

[Vxp, Vyp, w, t\_e] = desplazamiento(xi, yi, t\_i, xf, yf);

% Se inicia ciclo

for t=0:ts:3000

% Obtener las velocidades angulares de cada rueda

[w1, w2, w3, w4] = fcn(Vxp, Vyp, w);

% Calcular nueva posición

dx = Vxp\*ts; % Distancia recorrida en X

dy = Vyp\*ts; % Distancia recorrida en Y

d = sqrt((dx^2)+(dy^2)); % Distancia total recorrida

td = atan(dy/dx); % Angulo del vector de desplazamiento

xi = xi + d\*cos(td+t\_i); % Nueva posicion X

yi = yi + d\*sin(td+t\_i); % Nueva posicion Y

t\_i = t\_i + w\*ts; % Nueva posicion angular

end

Simulación sin introducir grados