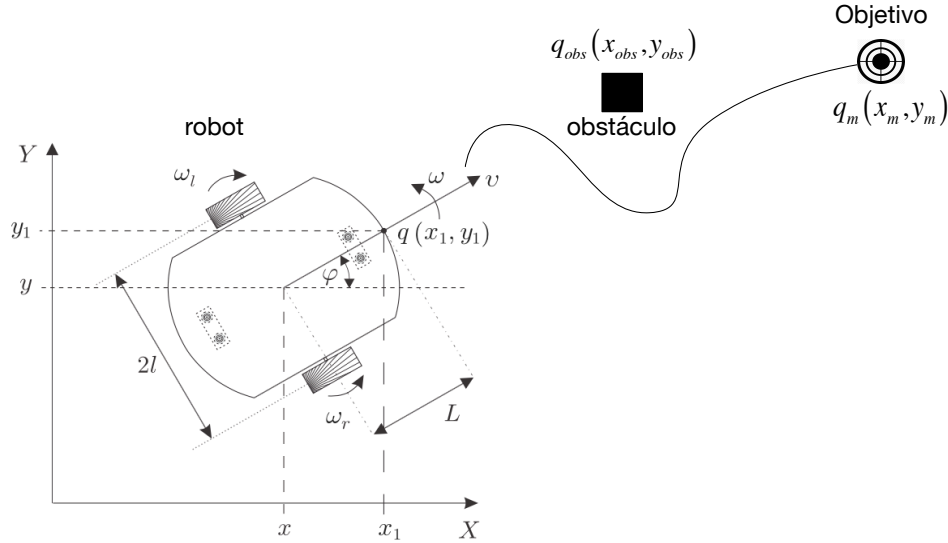


ROBÓTICA MÓVIL

Maestría en Ingeniería Electrónica
Pontificia Universidad Javeriana, Bogotá Colombia



TALLER: Modelado y Control de un Robot Móvil para evasión de obstáculos mediante planificación de trayectorias y campos potenciales.



Este Taller plantea el reto de desarrollo de un simulador en Matlab que permita a un robot móvil alcanzar un punto objetivo q_m de manera autónoma. Para ello, hay dos (2) posibles alternativas de solución:

- I. Definir una trayectoria del C-space libre de colisiones. Se asume conocida la posición de los obstáculos en el mundo de navegación. **NOTA:** Se requiere de la implementación de un planificador de trayectorias de bajo nivel.
- II. Definir una arquitectura de navegación basado en la teoría de campos potenciales, es decir, definir a lo largo del C-space fuerzas virtuales de atracción del robot hacia el objetivo y fuerzas de repulsión con respecto a los obstáculos. **NOTA:** No se requiere planificador de trayectorias; la navegación se realiza gracias a las fuerzas virtuales de los campos potenciales definidos.

Este Taller explora las dos alternativas de diseño previamente descritas. Los parámetros del robot móvil tipo unicycle (diferencial) son:

Constant	Definition	Value
l	Distance from (x, y) to the wheels	0.220 m
r	Wheel ratio	0.075 m
L	Distance from (x, y) to q	0.250 m
v_d	Desired constant velocity	0.5 m/s
ε	Constant value close to zero	0.1
η	Positive scale factor (repulsive)	2
ζ	Positive scale factor (attractive)	1
ρ_0	Distance of the obstacle's influence	0.5 m

ROBÓTICA MÓVIL

Maestría en Ingeniería Electrónica
Pontificia Universidad Javeriana, Bogotá Colombia



Parte 1. Planificador de trayectorias de bajo nivel: Implemente en Matlab el siguiente prototipo de función:

function [traj] = traj_planner(arg1, arg2, arg3, arg4)

Entrada argumentos:

- arg1: punto inicial y final de la recta $[X_0 \ Y_0 \ Z_0 ; X_m \ Y_m \ Z_m]$ dado en metros.
- arg2: velocidad máxima V_k del perfil trapezoidal dado en metros/segundo
- arg3: % aceleración/desaceleración.
- arg4: tiempo de muestreo dado en segundos.

Salida:

- traj: $[X \ Y \ Z \ V_x \ V_y \ V_z \ A_x \ A_y \ A_z \ \text{tiempo}]$; posiciones, velocidades, aceleraciones, tiempo.

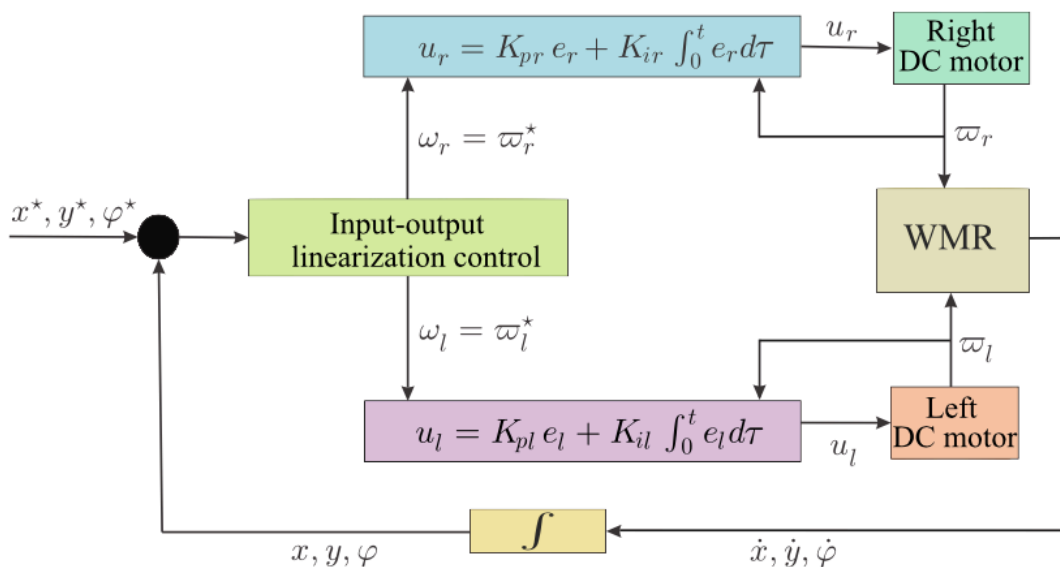
Protocolo de Prueba:

- Calcule una trayectoria en línea recta en el plano con perfil de velocidad trapezoidal con las siguientes características: velocidad máxima $V_k=1\text{m/s}$, %aceleración=20%, tiempo de muestreo = 0.05 segundos.

traj = traj_planner([0 0 0; 3 2 0],1,0.2,0.05);

- Grafique la trayectoria cartesiana X vs Y. Así mismo, grafique la velocidad en X-Y vs tiempo y la normal de la velocidad vs tiempo. Compruebe que se cumple la velocidad máxima establecida, así como los porcentajes de aceleración/desaceleración.

Parte 2. Arquitectura de Navegación mediante campos potenciales



ROBÓTICA MÓVIL

Maestría en Ingeniería Electrónica
Pontificia Universidad Javeriana, Bogotá Colombia



Descripción de módulos:

- **WMR:** modelo cinemático del robot móvil. El modelo está descrito por las siguientes ecuaciones geométricas:

$$\dot{x} = \frac{(\omega_r + \omega_l) r}{2} \cos \varphi,$$

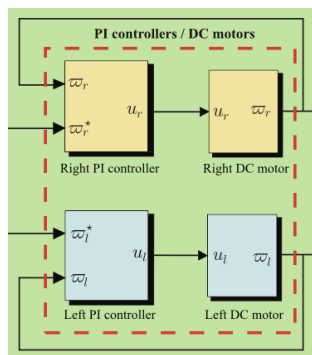
$$\dot{y} = \frac{(\omega_r + \omega_l) r}{2} \sin \varphi,$$

$$\dot{\varphi} = \frac{(\omega_r - \omega_l) r}{2l},$$

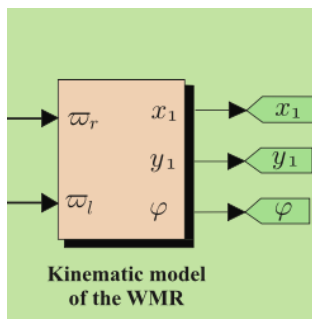
siendo ω_r, ω_l las velocidades angulares de cada rueda, r el radio de las llantas, φ el ángulo de orientación del robot con respecto al eje X del sistema de referencia, y l es la distancia entre el centro de masa del robot y el eje de cada rueda.

- **Right-left DC motor:** El modelo dinámico de cada motor en el dominio de Laplace es: $u(s) = (s+1)^{-1} \omega(s)$, siendo $u(s)$ el voltaje aplicado y $\omega(s)$ la velocidad de giro.
- **Control PI:** las señales de entrada para cada motor u_r, u_l son generadas por controladores tipo PI. Sintonizar las ganancias del PI según la planta del motor DC, un tiempo de establecimiento de 2 segundos y factor de amortiguamiento de 1.
- **Input-Output Linearization Control:** Genera los comandos deseados de velocidad para cada motor ω_r^*, ω_l^* en función de la posición del robot deseada por el usuario: x^*, y^*, φ^* . Este método requiere de la definición de un campo potencial para la navegación del robot.

Desarrollos:



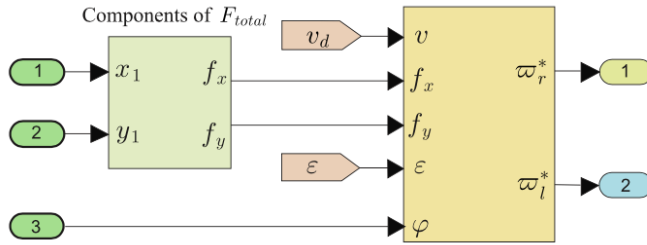
1. En simulink, implemente el control PI para el control de cada motor. Verifique el correcto seguimiento de la referencia de velocidad para $\omega^* = 2 \text{ rad/s}$ y $\omega^* = 4 \sin(2\pi t)$. Introduzca una perturbación constante de $u^p = 2v$. Grafique las señales de seguimiento, error y acción de control.



2. En simulink, implemente el modelo cinemático del robot móvil (WMR) usando el bloque *subsystem*. Las entradas al modelo son las velocidades angulares de cada motor y las salidas corresponden a las tres variables de estado del robot: x, y, φ

ROBÓTICA MÓVIL

Maestría en Ingeniería Electrónica
Pontificia Universidad Javeriana, Bogotá Colombia



3. En simulink, implemente el controlado de navegación para el robot (**Input-Output Linearization Control**). Este control requiere el calculo de las fuerzas virtuales f_x, f_y las cuales guiarán al robot a la meta, y a su vez, permitirán la

evasión de obstáculos en el camino. La acción de control que genera las referencias de velocidad a los bucles internos de control es:

$$\begin{pmatrix} \omega_r \\ \omega_l \end{pmatrix} = \frac{v_d}{\sqrt{f_x^2 + f_y^2 + \epsilon}} \frac{1}{Lr} \begin{pmatrix} L \cos \varphi - l \sin \varphi & l \cos \varphi + L \sin \varphi \\ L \cos \varphi + l \sin \varphi & - (l \cos \varphi - L \sin \varphi) \end{pmatrix} \begin{pmatrix} f_x \\ f_y \end{pmatrix} \quad (1)$$

siendo v_d la velocidad máxima deseada ($v_d = 1 \text{ ms}^{-1}$) y $\epsilon = 0.1$ (ver tabla de parámetros). f_x, f_y son los componentes de la fuerza potencial total F_{total} :

$$\begin{pmatrix} f_x \\ f_y \end{pmatrix} = \begin{pmatrix} f_{xat} + f_{x1rep} \\ f_{yat} + f_{y1rep} \end{pmatrix} \quad (2)$$

donde f_{xat}, f_{yat} son las fuerzas de atracción a lo largo del eje X-Y y f_{x1rep}, f_{y1rep} las fuerzas de repulsión generadas por (1) obstáculo en el C-space. En el caso que existiesen múltiples obstáculos distribuidos por el C-space, los componentes de fuerza virtual serían:

$$\begin{pmatrix} f_x \\ f_y \end{pmatrix} = \begin{pmatrix} f_{xat} + f_{x1rep} + f_{x2rep} + \dots + f_{xnrep} \\ f_{yat} + f_{y1rep} + f_{y2rep} + \dots + f_{ynrep} \end{pmatrix}$$

En este taller, solo se empleará el uso de 1 obstáculo, por lo tanto, los componentes de fuerzas son aquellos definidos en la Eq. (2). Por otro lado, las fuerzas de atracción del robot hacia el objetivo/meta se definen como:

$$\begin{aligned} f_{xat} &= -\xi(x_1 - x_m), \\ f_{yat} &= -\xi(y_1 - y_m). \end{aligned} \quad (3)$$

donde $\xi=1$ es la constante de atracción, x_1, y_1 es la posición actual del robot y x_m, y_m la posición del objetivo a alcanzar. Por otro lado, las fuerzas de repulsión encargadas de desviar el robot del obstáculo, se calculan como dos funciones a trozos, de la siguiente manera:

ROBÓTICA MÓVIL

Maestría en Ingeniería Electrónica
Pontificia Universidad Javeriana, Bogotá Colombia

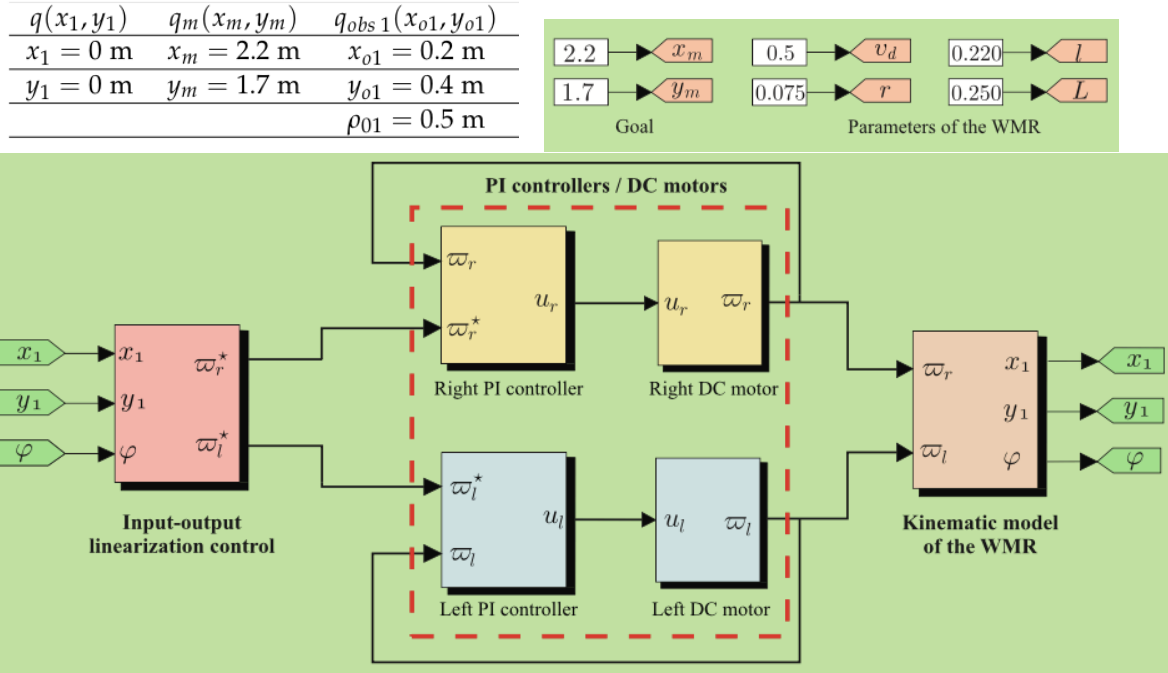


$$f_{x1rep} = \begin{cases} \eta \left[\frac{1}{\rho(q, q_{obs1})} - \frac{1}{\rho_{01}} \right] \left[\frac{1}{\rho^3(q, q_{obs1})} \right] (x_1 - x_{o1}), & \text{for } \rho(q, q_{obs1}) \leq \rho_{01}, \\ 0, & \text{for } \rho(q, q_{obs1}) > \rho_{01}. \end{cases}$$

$$f_{y1rep} = \begin{cases} \eta \left[\frac{1}{\rho(q, q_{obs1})} - \frac{1}{\rho_{01}} \right] \left[\frac{1}{\rho^3(q, q_{obs1})} \right] (y_1 - y_{o1}), & \text{for } \rho(q, q_{obs1}) \leq \rho_{01}, \\ 0, & \text{for } \rho(q, q_{obs1}) > \rho_{01}. \end{cases} \quad (4)$$

siendo: $\eta = 2$ la constante de repulsión. La expresión $\rho(q, q_{obs1}) = \|q - q_{obs1}\|$ hace referencia a la norma vectorial de la resta de los vectores de posición del robot q y la posición del obstáculo q_{obs1} . Por otro lado, x_1 es la posición en X del robot y x_{o1} es la posición en X del obstáculo. Finalmente, $\rho_{o1} = 0.5m$ es la distancia de influencia del obstáculo en el C-space.

- En Simulink, implemente la arquitectura final de navegación del robot. En la siguiente figura se detalla cómo debe ser la integración total del sistema:



- Grafique la trayectoria X vs Y del robot, incluya la posición del obstáculo y el radio de influencia.
- Grafique el error de seguimiento en X y Y, así como el error de seguimiento de ω_r, ω_l . Incluya también las acciones de control respectivas.

NOTA: para mayor información sobre campos potenciales y el control de navegación, estudiar el siguiente artículo (descargar en *Uvirtual*): **Obstacle Avoidance Task for a Wheeled Mobile Robot – A Matlab-Simulink-Based Didactic Application** By R. Silva-Ortigoza, C. Márquez-Sánchez, F. Carrizosa-Corral, V. M. Hernández-Guzmán, J. R. García-Sánchez, H. Taud, M. Marciano-Melchor and J. A. Álvarez-Cedillo.