

Fundamentos de Robótica Móvil

Tarea 3: Navegación por Campo Potencial

Presentado por:

Julián Andres González Reina

Emily Angelica Villanueva Serna

Elvin Andres Corredor Torres

Julio 1 2025

Solución Matlab

Modelo cinemático

Para realizar el modelo cinemático se toma el radio de la rueda como 0.02122 m y el espacio entre las ruedas, es decir la trocha como 0,053 m. El modelo cinemático para el robot e-puck es el siguiente

Modelo cinemático del e-puck

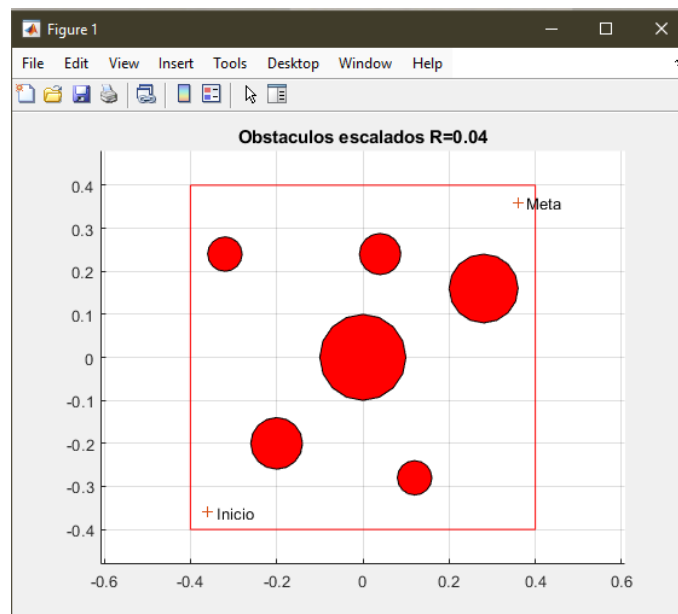
- $v = (r/2) * (v_r + v_l)$
- $\omega = (r/L) * (v_r - v_l)$

Modelo Cinemático Inverso

- $v_r = (2v + \omega L) / (2r)$
- $v_l = (2v - \omega L) / (2r)$

Mapas

Usando el algoritmo para redimensionar los obstáculos suministrado en la guía del laboratorio mapa.m se multiplica el valor de K por 0,04 que sería el radio que abarca toda la zona del robot e-puck incluyendo sus ruedas, dando como resultado la siguiente imagen.



Navegación por Campo Potencial

Para la navegación por campo potencial se tienen en cuenta las ecuaciones de repulsión y atracción.

Fuerza de atracción

$$U_{att}(q) = 1/2 \zeta |q - q_f|^2$$

$$F_{att}(q) = -\nabla U_{att}(q)$$

Fuerza de repulsión

$$1/2 \eta (1/(\rho(q)) - 1/Q^*)^2$$

cuando

$$\rho(q) \leq Q^*$$

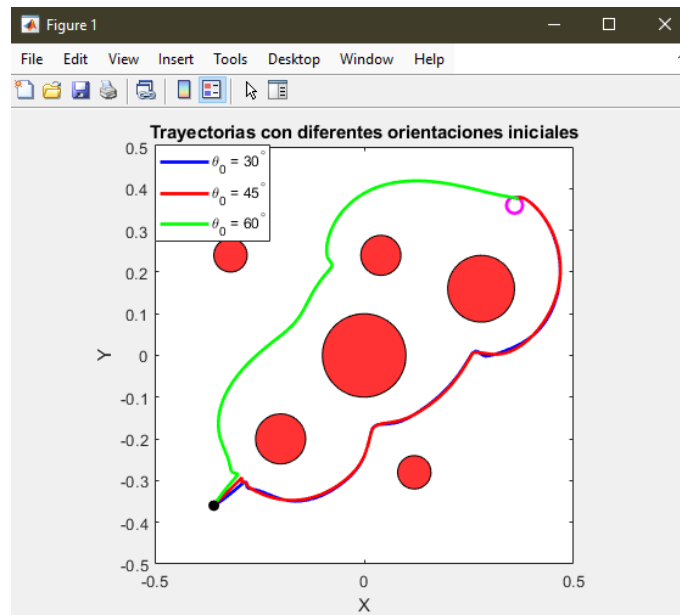
$$0, \rho(q) > Q^*$$

$$F_{rep}(q) = -\nabla U_{rep}(q)$$

Gradiente total

$$F_{total} = F_{att} + \Sigma F_{rep}$$

En la siguiente gráfica se muestran las trayectorias resultantes a partir de la suma de gradientes utilizando el código [Trayectorias.m](https://github.com/RobertoGarcia/Trayectorias.m)



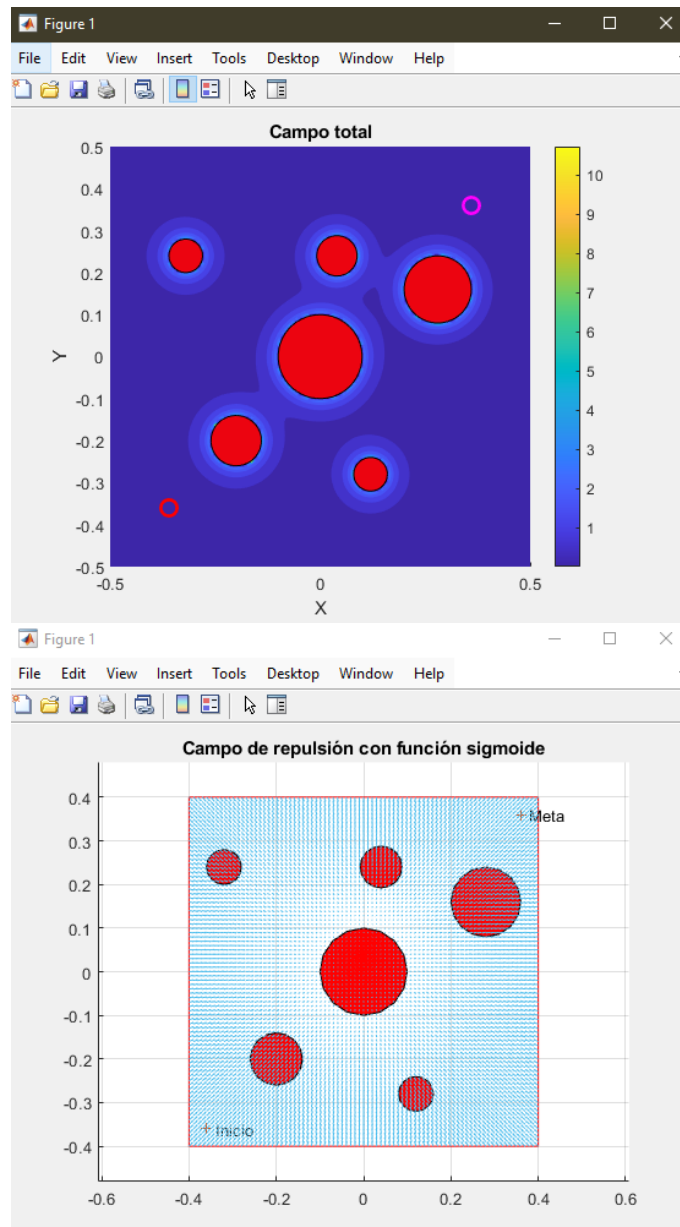
Se ajusta el algoritmo mencionado anteriormente para lograr que el robot cumpla su objetivo y realice la trayectoria hasta la meta

Parámetro	Valor	Descripción
Zeta	50	Atracción a la meta
eta	0.01	Repulsión de obstáculos
dsStar	0.3	Umbral atracción
Qstar	0.22	Alcance repulsión

Gradiente del Campo Potencial

Las siguientes imágenes están elaborados con el código [Campo Potencial.m](#) y [Gradiente.m](#) , la imagen del lado izquierdo es la representación en escala logarítmica de la atracción y repulsión que ejerce la meta y los obstáculos respectivamente, mientras que la imagen de la derecha intenta mostrar los vectores de repulsión y atracción usando la función sigmoideal. Función sigmoideal

$$1/(1+e^{-x})$$



De igual manera se realiza una simulación del sistema, pero en este caso con cinemática directa.

Para la resolución de esta tarea se buscaron las medidas del robot asignado para poder crear el mapa de obstáculos en Matlab, siendo los siguientes datos los correspondientes al robot e-puck.

Diámetro de Ruedas = 4 mm
 Distancia entre Ruedas = 53 mm
 Diámetro = 70 mm
 Altura = 55 mm

Se determinan las ecuaciones de cinemática directa del robot de acuerdo a las velocidades angulares del robot.

$$v = \frac{R}{2}(\omega_R + \omega_L)$$

$$\omega = \frac{R}{L}(\omega_R - \omega_L)$$

Y para resolver la posición y orientación del robot, usamos el método de integración de Euler, el cual fue el siguiente

$$x_{t+1} = x_t + v \cdot \cos(\theta) \cdot dt$$

$$y_{t+1} = y_t + v \cdot \sin(\theta) \cdot dt$$

$$\theta_{t+1} = \theta_t + \omega \cdot dt$$

Siendo

R: radio de las ruedas

L: distancia entre ruedas

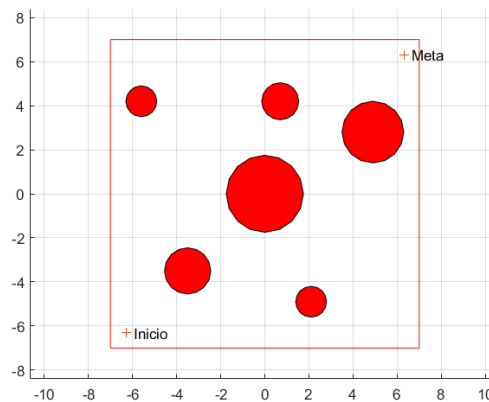
ω_R , ω_L : velocidades angulares de la rueda derecha e izquierda

x, y: posición del robot

θ : orientación

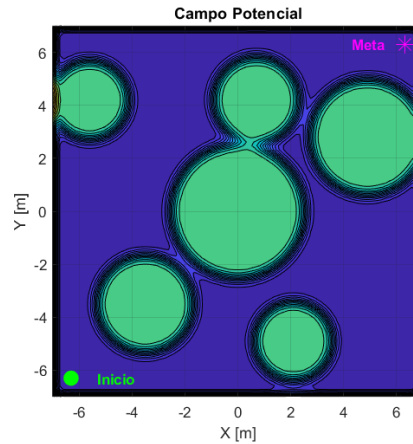
dt: paso de tiempo

Posterior a esto se aplica el algoritmo arena2025.m para obtener el mapa de obstáculos como se ve a continuación.



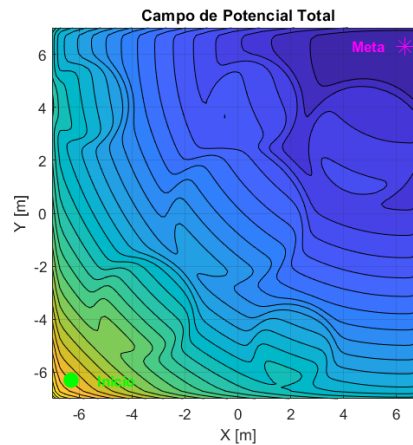
Mapa de obstáculos generado por Arena2025.m

Usando la función sigmoideal transformamos este mapa en un campo potencial repulsivo, también se agrega una repulsión en los bordes pensada para que el robot conozca los límites de movimiento sobre el que puede avanzar, en la visualización de este mapa se usan colores para identificar las zonas de mayor repulsión de campo en Matlab.



Campo potencial repulsivo

A continuación, se crea el campo potencial atractivo de la meta y se suman ambos campos para obtener el campo de potencial total.



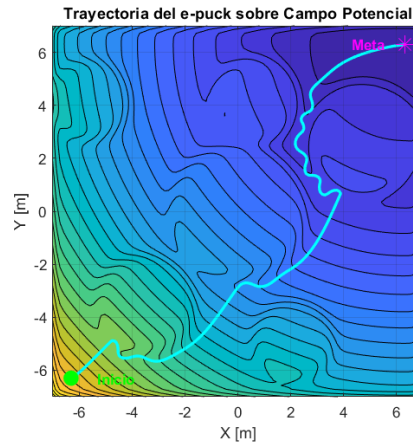
Este será el campo el cual el robot deberá utilizar para llegar a la meta. Luego de pruebas de simulación se logra llegar a las tres soluciones de salida, con los siguientes parámetros.

Distancia de Repulsión	Factor de Repulsión	Intensidad Campo Atractivo
D = 0,8 m	Lambda = 8	Alpha = 0,08

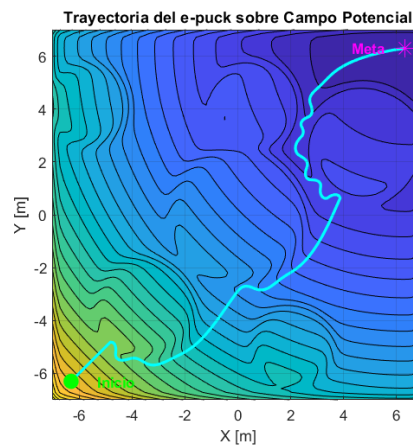
Estos valores fueron elegidos debido a varias características:

- Un valor inicial de Lambda = 15 generaba un campo de repulsión más fuerte a medida que se acercaba el robot, generando que en la condición de 30° el robot quedara girando sin poder llegar a la meta.
- La distancia de repulsión se redujo para que el robot pudiera acercarse más a los obstáculos antes de ser desviado de su trayectoria.
- Se aumentó la intensidad del campo atractivo, inicialmente en un valor Alpha = 0,01 en condiciones que nuevamente quedaba girando el robot sin llegar a la meta.

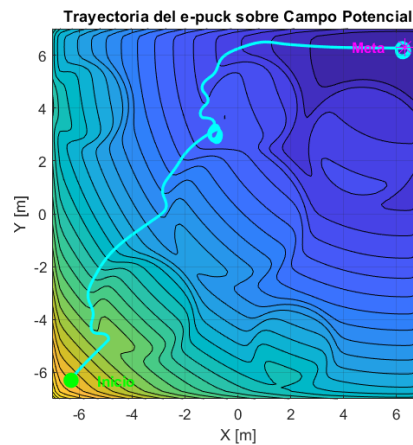
Con estas características se muestra a continuación los resultados obtenidos con los valores de ángulo inicial.



Trayectoria a 30°



Trayectoria a 45°

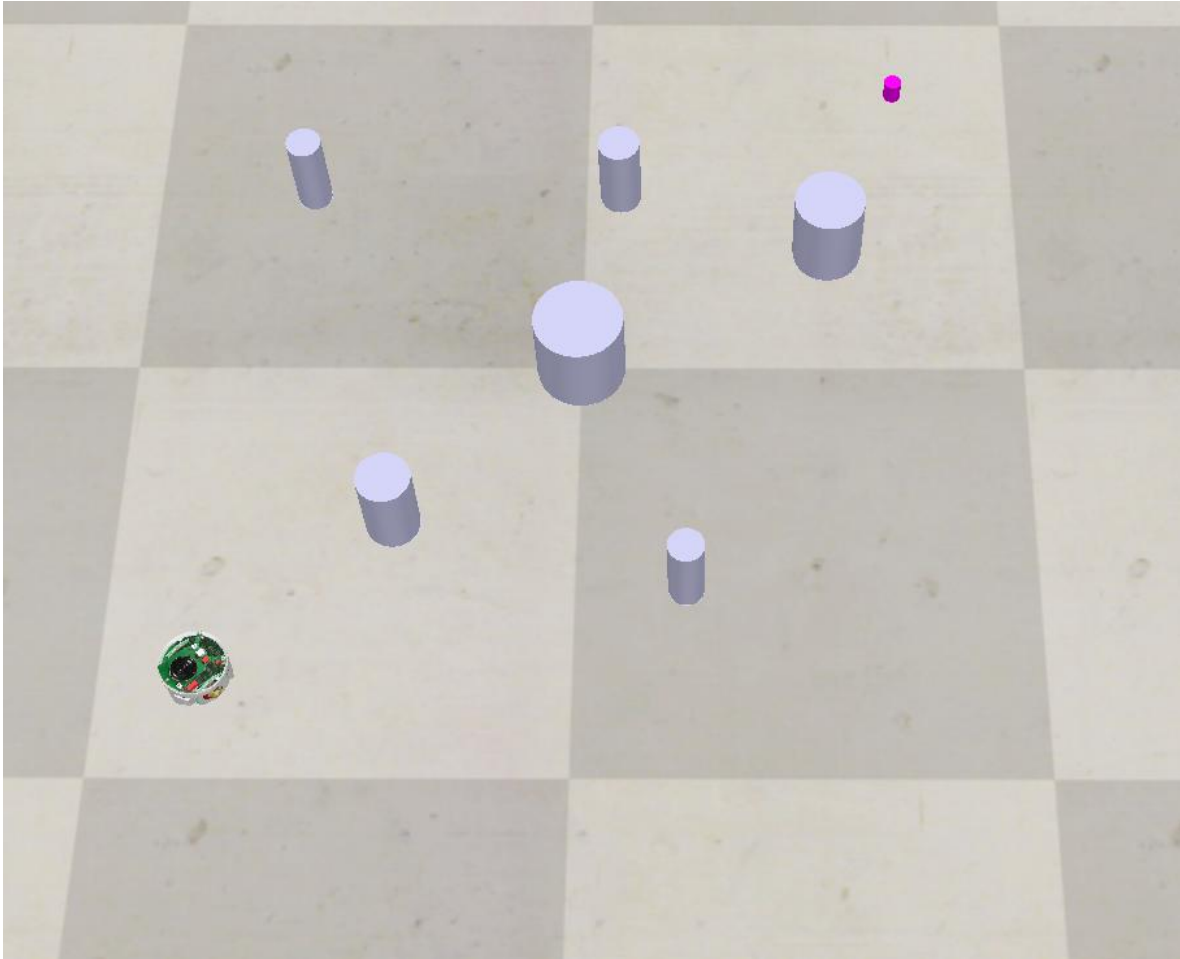


Trayectoria a 60°

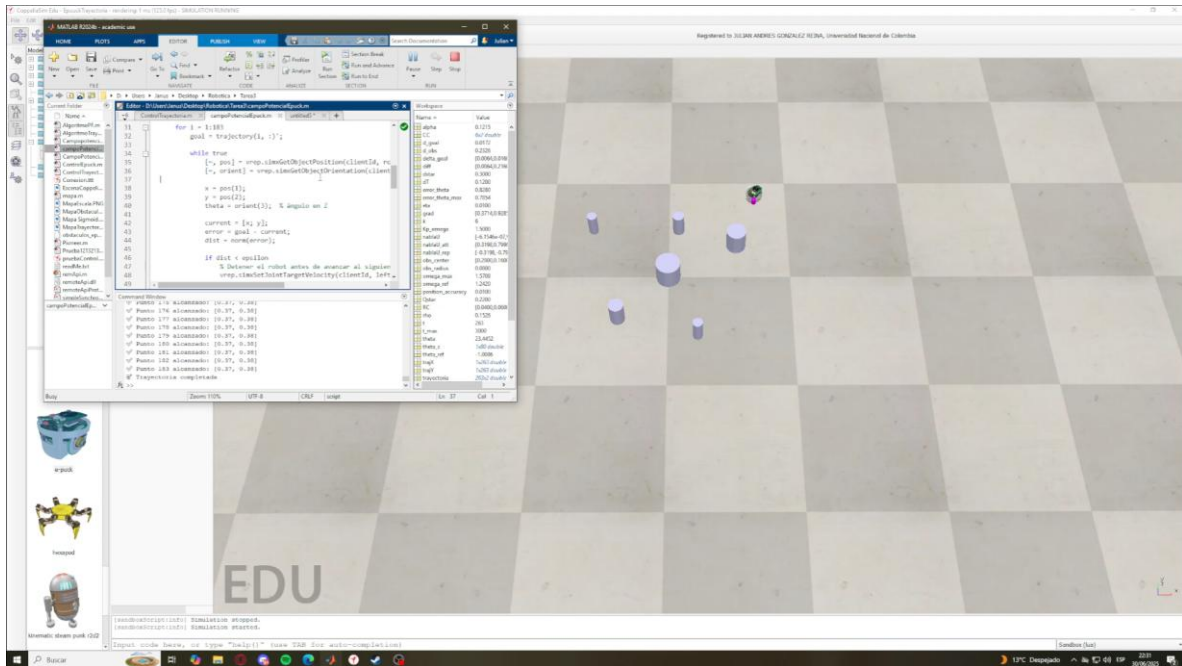
Aunque no es muy visible la diferencia entre 30° y 45° esta se nota justamente al inicio, luego de esto el campo de repulsión de ambas toma la misma trayectoria, mientras que en la trayectoria a 60° si es visible la diferencia además de que en un punto queda girando el robot, esto conocido como un mínimo local, solucionado de manera que se le da un giro aleatorio para poder salir del mínimo local.

Simulación CoppeliaSim

Para la simulación en CoppeliaSim se agregan cilindros correspondientes a los obstáculos y se fijan con un force sensor, la ubicación y radio de los obstáculos están dados por el código mapa.m, la demostración se encuentra en la siguiente imagen donde el robot tiene la siguiente pose $[-0.36, -0.36, 45]$



Finalmente, en la siguiente imagen está el robot llegando al objetivo para una orientación de 45 grados.



Conclusiones

- Una de las dificultades más notorias para realizar esta navegación fue el hecho de tener que cuadrar parámetros a prueba y error para lograr que el robot lograra cumplir con su misión. Al inicio si la repulsión era muy grande el robot solo avanzaba unos pocos milímetros y quedaba en un mínimo local donde dejaba de moverse asumiendo haber llegado al objetivo, por otra parte, en caso de tener una atracción muy grande la trayectoria pasaba por encima de los obstáculos.
- Durante la simulación en Coppeliasim a diferencia del código en Matlab, el robot hacia giros extraños como rebotando con los obstáculos por el efecto de repulsión, es una de las desventajas de utilizar este método de navegación, es que los parámetros son muy sensibles y pueden causar dinámicas no deseadas al momento de la implementación.
- Aunque el concepto es sencillo de entender es posible que en algunos casos pueda existir la posibilidad de no poder llegar o encontrar la meta así exista un camino disponible.
- En ese caso para el robot e-puck fue necesario colocar un parámetro de atracción muy grande respecto al de repulsión ya que con nuestro algoritmo siempre se quedaba en mínimos locales antes de llegar al objetivo, asumiendo que el problema estaba en los parámetros decidimos agrandar la atracción realizada por la meta para lograr cumplir con la misión.

Este informe y sus códigos de programación pueden ser consultados en el repositorio

[JulianGonzalezR1999/NavegacionCampoPotencial](https://github.com/JulianGonzalezR1999/NavegacionCampoPotencial)