

ROBÓTICA

GRUPO 2:

735976 - Alberto Lardiés

780131 - Devid Dokash

801108 - Félix Ozcoz

LISTADO DE OPCIONALES

P2. Generación de trayectorias:

- Centrar robot mediante ultrasonido
- Actualización de odometría en todo momento
- Generación de trayectorias mediante interpolación
- Recorrido de trayectorias por muestreo
- Rotación del robot hacia la trayectoria si está rotado inicialmente
- Generación de trayectorias mediante un mapa

P3. Tracking de un objeto mediante visión:

- Corrección de trayectoria cuando el robot va a por la pelota
- Seguimiento fluido
- Marcha atrás si tenía la pelota visible en vez de girar
- Rotar hacia el último lado donde estaba la pelota
- Bajar/subir la cesta dependiendo de dónde esté la pelota
- Elegir sentido de la rotación inicial
- Añadir un filtro en Y para blobs que estén muy por encima en la imagen
- Velocidades lineal y angular dependientes de la distancia al blob (distancia en píxeles en la imagen)

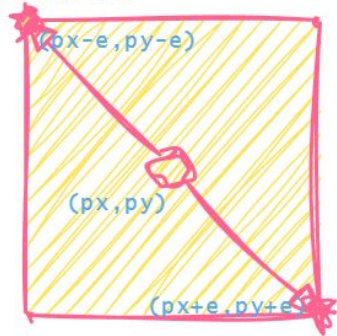
P4. Planificación y navegación de mapa:

- Corrección de trayectoria de una celda a otra mediante odometría
- Detección de obstáculos y re-planificación de camino
- “Backtracking”: Si me habían cortado un camino lo abro y vuelvo a comprobar
- Backtracking “inteligente”. Si el mapa solo tiene un camino y se lo tapan se queda quieto
- Encontrar camino mediante A*
- Posicionamiento y localización mediante ultrasonido
- Interpolación de trayectoria a partir de un path 8N

¿CÓMO SE DETERMINA LA POSICIÓN Y ORIENTACIÓN DEL ROBOT?

1. Mediante odometría calculada con el giroscopio, activada constantemente.
2. Uso de geometría para los desvíos de odometría:

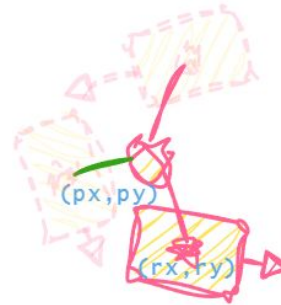
a) Área



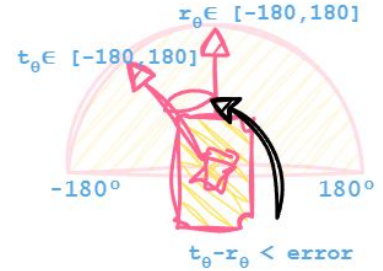
b) Distancia



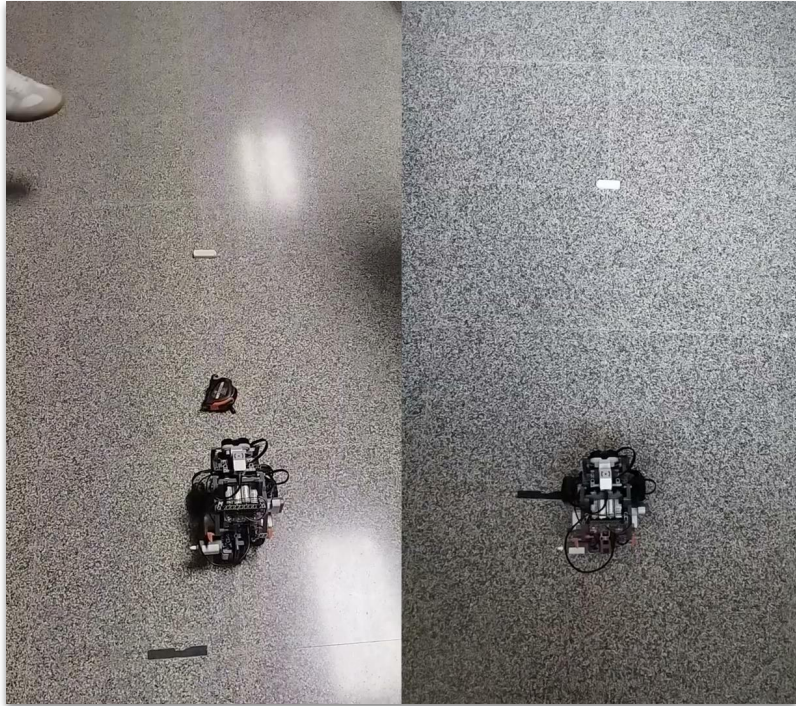
c) Mínimo local



d) Ángulo



¿CÓMO “GENERABA” UNA TRAYECTORIA?

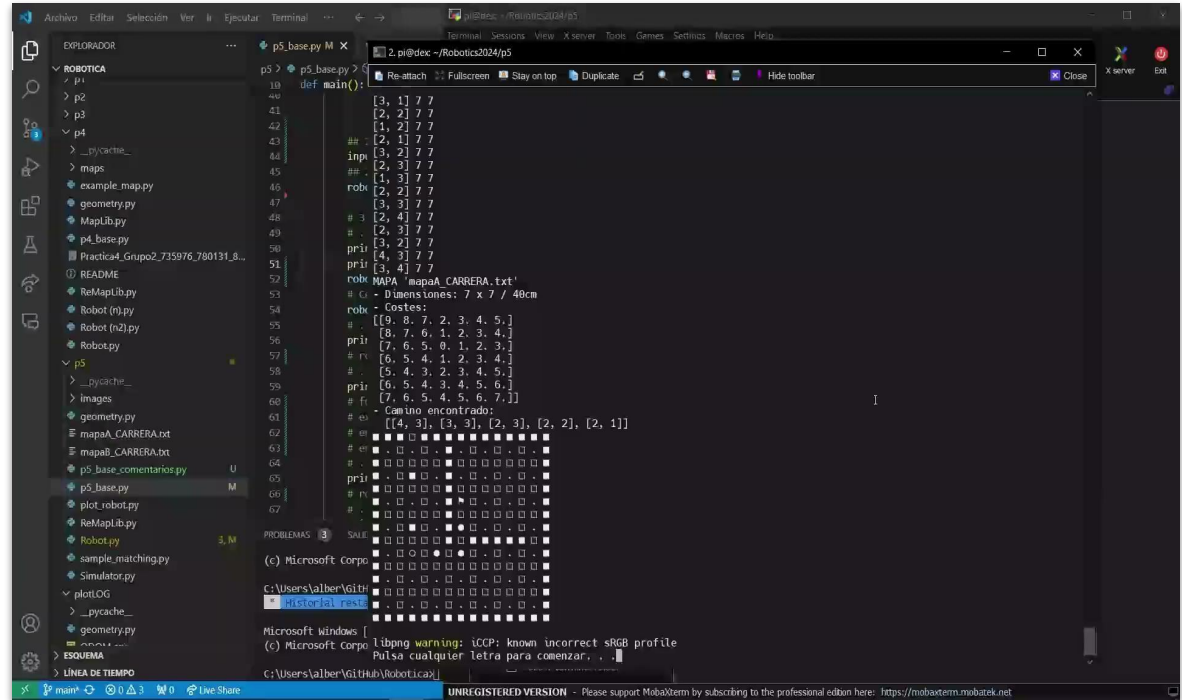


Realizada a mano:

- Se determina la posición y la rotación de cada punto de la trayectoria.
- Se determina la velocidad lineal y angular en cada punto de la trayectoria.
- Sin el uso de la geometría.

¿CÓMO GENERA UNA TRAYECTORIA AHORA?

1. Se **interpola la trayectoria** en tiempo real mediante una **serie de puntos en el espacio**.
2. Se recorre mediante **muestreo de segmentos**:
 - a. La trayectoria se divide en n segmentos.
 - b. Un segmento se determina por un punto en el espacio y una orientación.
3. Respecto al **control de movimiento**:
 - a. Se define una velocidad lineal constante.
 - b. La velocidad angular se define mediante la variación de ángulo entre el segmento anterior y el siguiente por un escalar s .
 - c. Rotación inicial del robot para alinearlo al primer segmento.



```
p5 > p5_base.py M X
def main():
    # Robot
    rob = Robot(0, 0, 0)
    # Mapa
    mapa = Mapa(7, 7)
    # Trayectoria
    traj = [
        [3, 1], [7, 7],
        [2, 2], [7, 7],
        [1, 2], [7, 7],
        [2, 1], [7, 7],
        [3, 2], [7, 7],
        [2, 3], [7, 7],
        [1, 3], [7, 7],
        [2, 2], [7, 7],
        [3, 3], [7, 7],
        [2, 4], [7, 7],
        [2, 3], [7, 7],
        [3, 2], [7, 7],
        [4, 3], [7, 7],
        [3, 4], [7, 7]
    ]
    # Mapa
    mapa = Mapa(7, 7)
    # Costes
    costs = [
        [9, 8, 7, 2, 3, 4, 5],
        [8, 7, 6, 1, 2, 3, 4],
        [7, 6, 5, 0, 1, 2, 3],
        [6, 5, 4, 1, 2, 3, 4],
        [5, 4, 3, 2, 3, 4, 5],
        [6, 5, 4, 3, 4, 5, 6],
        [7, 6, 5, 4, 5, 6, 7]
    ]
    # Camino encontrado
    path = [[4, 3], [3, 3], [2, 3], [2, 2], [2, 1]]
    # Grid
    grid = [
        [0, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 0]
    ]
    # Visualización
    mapa.plot_robot(rob)
    mapa.plot_traj(traj)
    mapa.plot_path(path)
    mapa.show()
    # Historial de movimientos
    print(traj)
    print(path)
    print(grid)
    # Fin del programa
    return 0
```



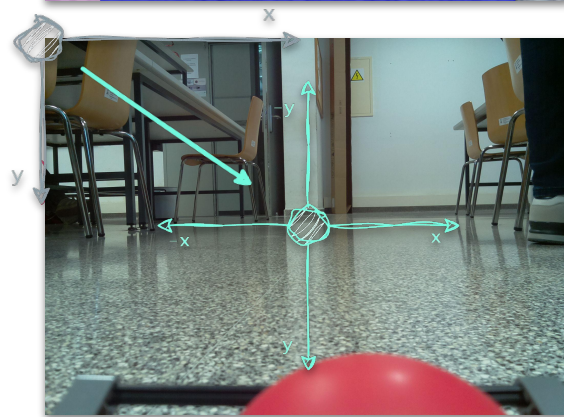
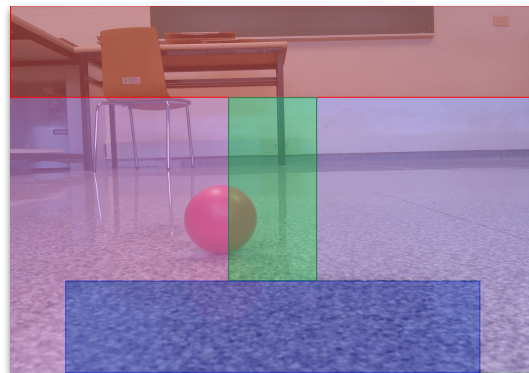

TRACKING DE OBJETOS

1. Propiedades de la imagen:

- Zona roja:** Los blobs en esa altura en Y son ignorados.
- Zona verde:** Representa el área en la que blob debe estar. Si se sale y sigue detectando, añadirá una pequeña velocidad angular adicional para corregir la trayectoria, si no, girará sobre sí mismo.
- Zona azul:** Si el blob está en esta zona, el robot baja la cesta. Si deja de verla y estaba en esta zona, dará marcha atrás. Si la bola se aleja, el robot volverá a subir la cesta.
- El sistema de coordenadas se traslada al centro de la imagen de la esquina superior izquierda.

2. Control de movimiento:

- La velocidad lineal depende de la altura en Y del blob, y la angular de la posición en X.
- El robot girará hacia el último lado que ha visto la pelota.
- (NO DEPURADO) Si el robot detecta un obstáculo cuya distancia varía con el ultrasonido tras capturar la pelota, se detiene.

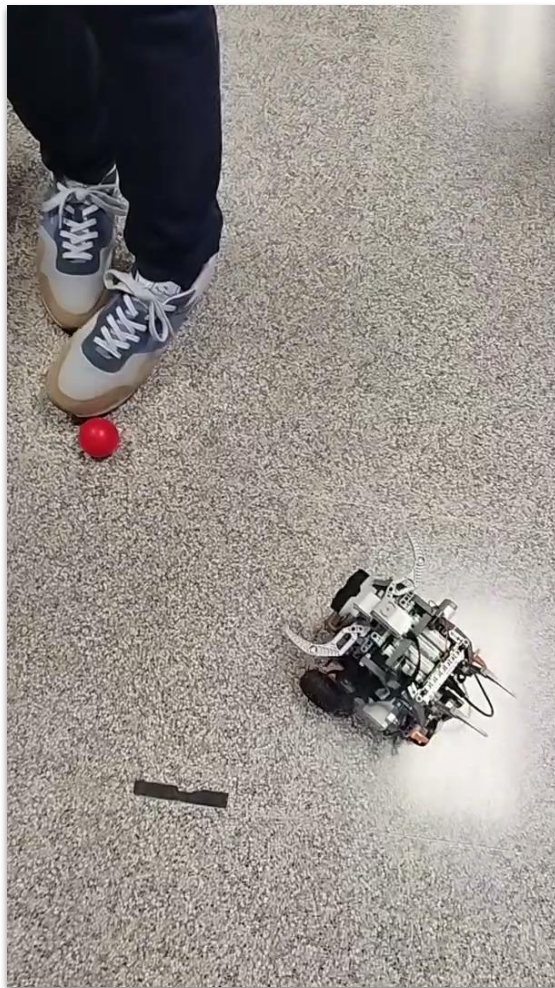




Tracking de la pelota en vivo y márgenes de imagen



Marcha atrás



Tracking y corrección

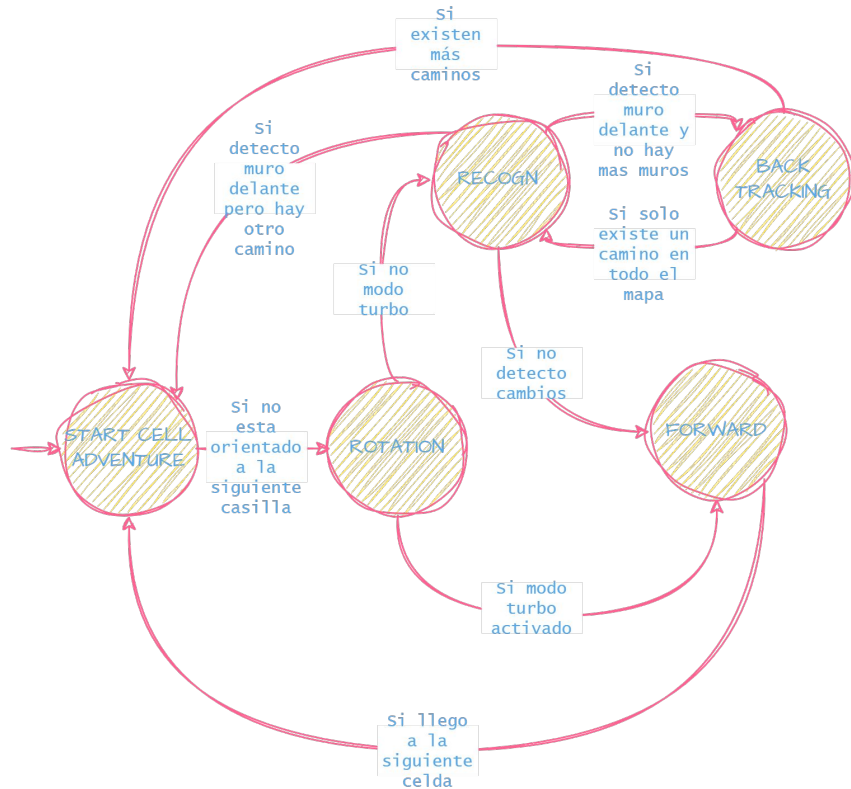


Giro al último lado visto



Bajada y subida de cesta

PLANIFICACIÓN Y NAVEGACIÓN DE MAPAS



1. Búsqueda de path con **A***.
2. “**Backtracking**” para búsqueda continua de paths.
1. Corrección de trayectoria de una celda a otra mediante geometría y ultrasonido:
 - a. Si el ultrasonido detecta distancias menores a 3 baldosas, se usa.
 - b. Si es mayor, solo va por odometría y geometría.

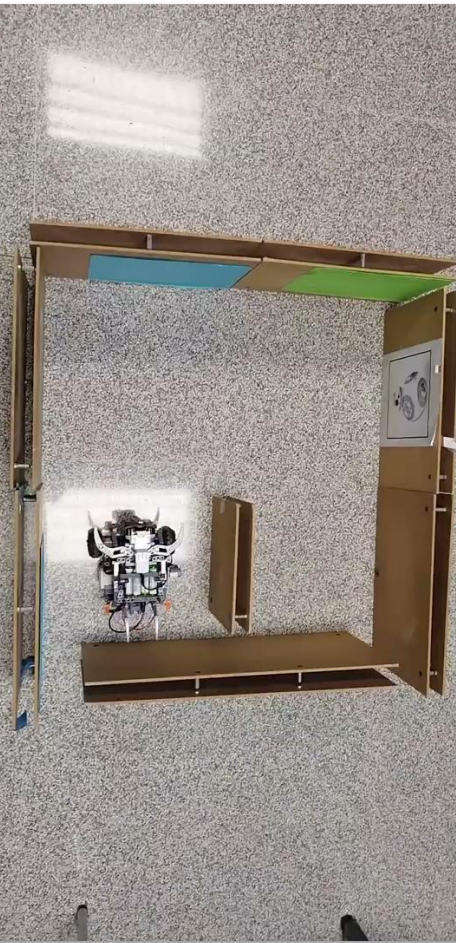


Backtracking en vivo:

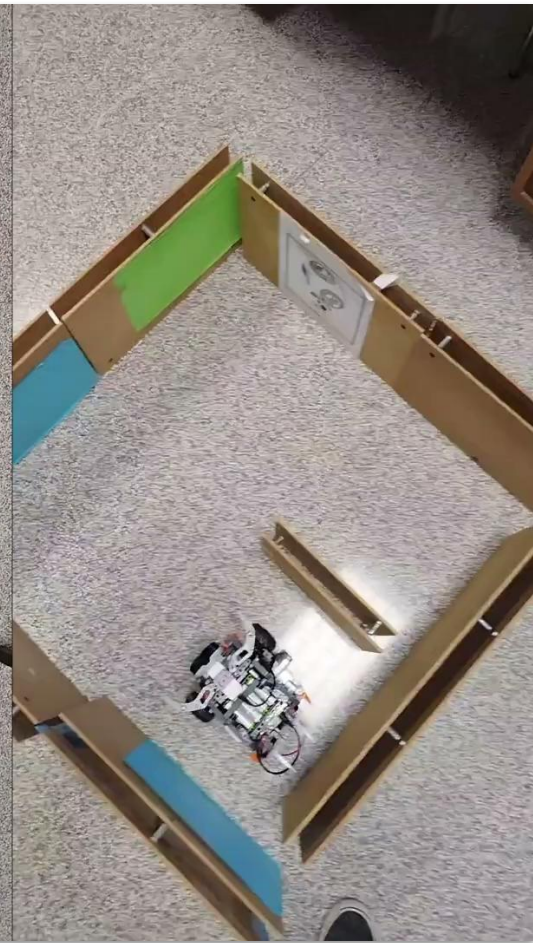
- 1) Cuando solo hay un camino
- 2) Cuando hay varios



Sin corrección



Corrección geométrica



Corrección geométrica
y ultrasonido



Navegación con corrección

INTERPOLACIÓN DE TRAYECTORIA A PARTIR DE UN PATH EN 8 VECINDAD

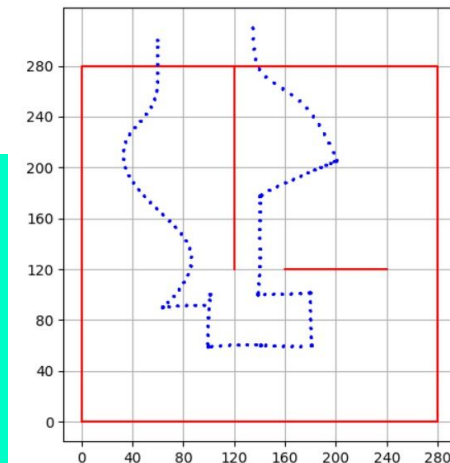
1. Se calcula la trayectoria más corta en 8-vecindad desde la posición del robot a la casilla que debe salir



2. Y se calcula la trayectoria en base al camino encontrado



MEJORES EJECUCIONES









FIN