

---

## Planificación de movimiento y control de robots móviles con ruedas

---

### Objetivos

- Implementar y comparar algoritmos de control de punto a punto para un robot móvil Pioneer P3-DX dentro del entorno de simulación Webots.
- Implementar algoritmos de planificación de movimiento basados en búsqueda en grafos y muestreo aleatorio, para resolver un laberinto sencillo dentro del entorno de simulación Webots.

### Procedimiento

En esta práctica usted deberá efectuar tres tareas específicas: lograr mover a un robot móvil de un punto a otro, generar una ruta válida para el mismo entre un punto inicial y una meta dentro de un mapa y finalmente visualizar la combinación de ambas tareas dentro de Webots. Para lograr esto, realice lo siguiente:

#### Control del robot Pioneer P3-DX

1. Descargue de Canvas el archivo `mt3005lab9.zip` y extraiga sus contenidos dentro de una carpeta en una ubicación de su preferencia. Dentro de esta carpeta se encuentra un directorio de proyecto de Webots, el cual contiene un archivo de mundo con un robot móvil Pioneer P3-DX (se adjunta a la guía la ficha técnica del robot) en el borde inferior de un piso de 10x10m en donde cada cuarto se encuentra delimitado por paredes, como se muestra en la Figura 1.
2. Ubique al robot Pioneer P3-DX en el explorador de escena a la izquierda y verifique que en el campo de `extensionSlot` tenga conectados dos sensores, un GPS y una brújula, los cuales le permitirán encontrar la posición y orientación del robot dentro del código de control. Emplee los sensores para encontrar la posición absoluta y el ángulo de *bearing* del robot (se le da un ejemplo de como obtener el mismo en la página de referencia del sensor). Recuerde que puede verificar que está obteniendo los datos adecuados imprimiendo las mediciones otorgadas por los sensores a la consola, cambiando la posición y orientación del robot manualmente y verificando que las mediciones coincidan.
3. Utilice la información de los sensores y la posición contenida en el vector `goal1` para implementar un controlador PID con acercamiento exponencial y un LQI (por aparte, puede

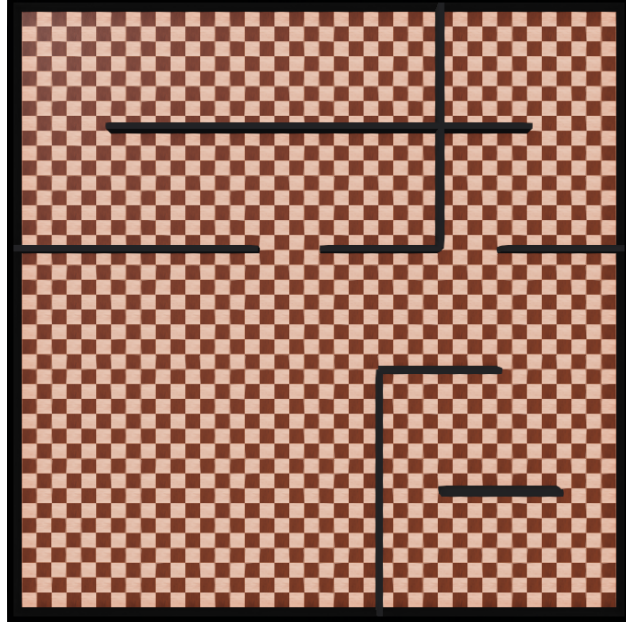


Figura 1: Mapa del mundo de Webots suministrado.

emplear ifs para seleccionar uno u otro) que hagan que el robot llegue a la meta `goal1` de forma adecuada (sin espirales amplias y evitando colisionar con alguna de las paredes). Recuerde que la estrategia de control para un robot diferencial es primero encontrar las velocidades  $(v, \omega)$  del unicycle y luego mapearlas a las velocidades de cada una de las ruedas mediante

$$\dot{\phi}_R = \frac{v + \omega \ell}{r}, \quad \dot{\phi}_L = \frac{v - \omega \ell}{r},$$

en donde  $r$  y  $\ell$  corresponden al radio de las ruedas del robot y a la distancia del centro del robot a cada rueda respectivamente (estos parámetros ya se encuentran definidos dentro del código e igualmente se encuentran en la hoja de especificaciones del robot Pioneer P3-DX adjunta a la guía del laboratorio en Canvas).

**Muestre los resultados de esta sección al catedrático del laboratorio antes de continuar.**

### Planificación de movimiento empleando la Robotics Toolbox

Ya con el control de punto a punto funcionando adecuadamente para el robot, se procederá a determinar rutas válidas entre el robot y dos de las metas restantes del array `goal_points`, las cuales no pueden alcanzarse sólo con el control punto a punto ya que presentan obstáculos, mediante planificación de movimiento. Para ello, ejecute los siguientes pasos:

1. La variable `map` dentro del script contiene la *occupancy grid* para el mapa (formada desde una imagen del mismo), la cual corresponde a un array en donde cada casilla puede tomar sólo 2 valores, 1 si la casilla contiene un obstáculo y 0 en caso contrario (estos valores se presentan en tipo de dato `double` como requerimiento de la *Robotics Toolbox*). Esta se visualiza cada vez que se corre la simulación y será una de las entradas requeridas para los algoritmos de planificación de movimiento. Tome nota que algunos de los algoritmos de planificación le realizan un “flip” al mapa, por lo que puede que los comandos `flip` y/o `imrotate` le sean de utilidad.

2. Emplee el documento de referencia `Navigation-PC.pdf` para implementar un algoritmo de búsqueda D\* que determine una ruta válida entre la posición inicial del robot y la `goal3`. Para hacer esto tome en consideración lo siguiente:

- Al momento de emplear los métodos constructores (`Dstar` y `PRM`) puede colocarles como opción (para el caso D\* por ejemplo) `Dstar(map, 'inflate', radius_in_cells)`; lo cual efectúa una inflación/aumento del mapa asumiendo que el robot cabe dentro de un círculo con radio dado en número de celdas. Puede determinar este radio empleando las dimensiones del robot descritas en su hoja de especificaciones y sabiendo que las dimensiones del piso en Webots (10x10m) corresponden a las dimensiones del mapa (500x500 celdas).
- Es necesario que efectúe una conversión de metros a celdas para las posiciones de las metas y la posición inicial del robot. Tome en consideración que el origen del mapa dentro de Webots se encuentra exactamente en el centro de la escena, lo cual quiere decir que toda posición se encuentra en el intervalo  $[-5, 5]$ . Es necesario entonces hacer una conversión de distancia en este intervalo al intervalo  $[1, 500]$ .
- Al momento de hacer una consulta al algoritmo de planificación, puede emplear (para el caso D\* por ejemplo) `path = ds.query(goal, 'animate')`; para visualizar la animación del recorrido encontrado al igual que guardar los puntos del mismo dentro del array `path`. El resultado deberá verse (cualitativamente) similar a la Figura 2. Guarde el recorrido encontrado ya que se empleará para la última sección de la práctica.

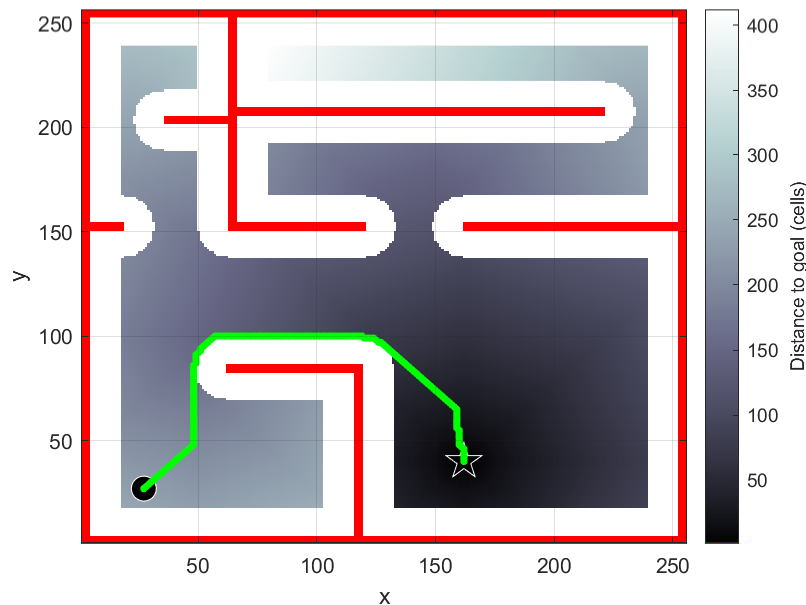


Figura 2: Ejemplo de resultado para búsqueda D\*.

3. Repita el procedimiento del inciso anterior para la `goal2` pero empleando Probabilistic Roadmaps (PRMs). De nuevo guarde el recorrido encontrado ya que se empleará para la última sección de la práctica.

## Resolviendo el laberinto en Webots

Ya con los recorridos encontrados, prosiga a hacer que el robot móvil en Webots recorra estas rutas mediante el mejor control punto a punto que haya determinado en la primera sección de la práctica, en donde el punto a donde se desea llegar debe desplazarse según los puntos del recorrido. Para lograr esto se le recomienda seguir los siguientes pasos:

1. Dentro del ciclo de simulación, extraiga la posición  $(x_g, y_g)$  para los controladores desde los arrays de los recorridos creados durante la planificación para cada una de las metas. Modifique su código tal que no se tome un nuevo punto del recorrido mientras que no se haya llegado lo suficientemente cerca del punto actual. Esto puede codificarse como una condición para el error de posición  $e_p$ . Tome en consideración que el array del recorrido encontrado está en coordenadas en celdas, por lo cual es necesario regresarlo a metros antes que el robot trate de ejecutar su control de punto a punto.
2. En caso de ser necesario, ajuste los valores de las constantes del controlador hasta lograr que el robot llegue a cada una de las metas, a través de los recorridos, adecuadamente. Muestre la ejecución de ambos recorridos al catedrático del laboratorio.

Cuando esté satisfecho con sus resultados, presente los mismos al catedrático del laboratorio y responda a las preguntas que se le hagan. Recuerde que entregas tardías representan una penalización del 25 % por semana.