Tarea 04
Planeación de rutas mediante RRT*

Robots Móviles (TSM I, TSM II, TSCR), FI, UNAM, 2025-1

Nombre:_____

1. Actividades

1. Abra el archivo `catkin_ws/src/navigation/path_planner/scripts/rrt.py` y pegue el siguiente código en la línea 89 para implementar el algoritmo de árboles aleatorios de exploración rápida:

```
89 while goal_node.parent is None and max_attempts > 0:
90     [x,y] = get_random_q(grid_map)
91     nearest_node = get_nearest_node(tree, x, y)
92     new_node     = get_new_node(nearest_node, x, y, epsilon)
93     if not check_collision(nearest_node, new_node, grid_map):
94         nearest_node.children.append(new_node)
95         if not check_collision(new_node, goal_node, grid_map):
96             new_node.children.append(goal_node)
97             goal_node.parent = new_node
98     max_attempts -= 1
99
```

2. Abra una terminal y corra la simulación con el comando (revise las instrucciones en el README del repositorio):

```
1     roslaunch surge_et_ambula movement_planning.launch
2
```

3. En otra terminal, corra el inflado de mapas con el comando:

```
1     rosrun map_augmenter map_inflater.py _inflation_radius:=0.2
2
```

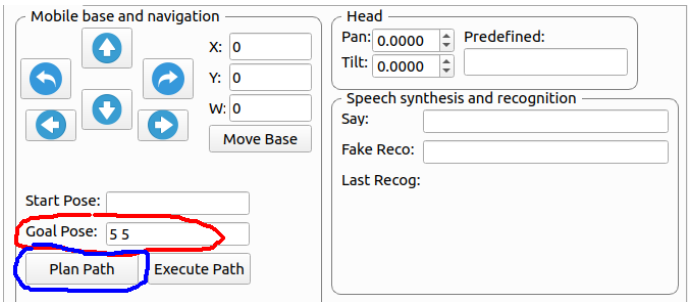
4. En otra terminal (sí, otra), corra el algoritmo *RRT* con el comando:

```
1     rosrun path_planner rrt.py _epsilon:=0.5 _max_n:=200
2
```

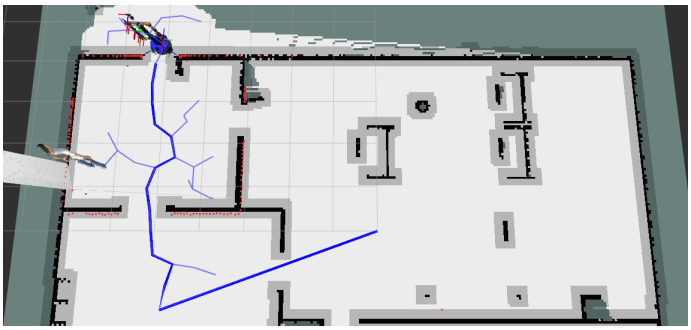
   Este comando ejecuta el algoritmo RRT con $\epsilon = 0.5$ y $N = 200$.

5. Calcule una ruta al punto con coordenadas $P = (5, 5)$. Para ello, en la GUI escriba la coordenada del punto meta y presione el botón *Plan Path* como se muestra en la figura:

*Material elaborado con apoyo del proyecto PAPIME PE105524



Se debería desplegar, en el visualizador RViz, el arbol de exploración y la ruta planeada como se muestra en la figura:



Recuerde que es un algoritmo de exploración aleatoria por lo que el resultado puede no ser el mismo. Revise en la terminal donde se ejecutó el algoritmo RRT si se pudo o no calcular la ruta y el tiempo que le tomó al algoritmo calcularla.

6. Pruebe el algoritmo para diferentes puntos meta, se sugieren: $(2.0, 10.0)$, $(10.5, 9, 75)$ y $(2.0, 7.0)$. Para cada posición pruebe diferentes valores de ϵ y N y registre si se pudo calcular o no la ruta, y el tiempo de ejecución.

7. Realice una o varias tablas donde se registren los resultados de los experimentos (puntos meta, número de éxitos, parámetros usados y tiempos de ejecución).

2. Entregables

- Código modificado en la rama correspondiente
- Documento impreso con los siguientes puntos:
  - Capturas de pantalla de al menos dos experimentos (se pueden incluir más) con los resultados más significativos.
  - Tabla de datos de la actividad 7.
  - Discusión de los datos presentados.

No es necesario pegar el código modificado en el documento escrito. Para eso está el repositorio en línea.

3. Evaluación

Para la evaluación se utilizará la siguiente lista de cotejo:

- [**0 puntos**] Los cambios al código se subieron a la rama correspondiente con las credenciales de cada alumno. Este punto es requisito para evaluar los demás.
- [**2 puntos**] El programa para el cálculo de rutas funciona correctamente.

- [**2 puntos**] Se incluyen al menos dos capturas de pantalla de dos rutas calculadas con parámetros de sintonización diferentes.
- [**2 puntos**] Se incluyen más de dos capturas de pantalla con rutas calculadas para diferentes puntos meta y con diferentes parámetros de sintonización.
- [**2 puntos**] Se incluye(n) la(s) tabla(s) de la actividad 7.
- [**2 puntos**] Se incluye una discusión sobre el efecto de los parámetros ϵ y N así como el punto meta.