

Planificación de movimientos basada en sensores para robots móviles en ambientes desconocidos

Valerio Galiote T., Abraham Sánchez L., José F. Texcucano D.,
Alfredo Toriz P., Rogelio González V., Elberfeld Pérez G.

Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación,
Laboratorio MOVIS, Puebla, Pue., México
{val.gali, faustinetexcucano, alfredot, enrique5001}@hotmail.com
abraham.sanchez@correo.buap.mx, rgonzalez@cs.buap.mx

Resumen. La navegación basada en sensores permite a un robot explorar un ambiente desconocido y construir un mapa de él, utilizando su información del sensor. Este es uno de los temas de investigación tradicionales para robots móviles y se han propuesto muchos algoritmos a lo largo de los últimos años. En este trabajo, desarrollamos algunas estrategias para la planificación de movimientos basada en sensores en entornos desconocidos, basados en el enfoque de árboles aleatorios de exploración rápida (RRT). Estas estrategias exploran el área desconocida de manera similar al método del árbol aleatorio basado en sensores (SRT) y adaptan el enfoque RRT a la planificación del camino local en el espacio de estados libre asociado con la región segura. Nombramos nuestras estrategias Local SRRT y Local B SRRT. Se presentan resultados experimentales para mostrar el desempeño de las estrategias propuestas.

Palabras clave: RRT, SRT, planificación de movimientos, robots móviles.

Sensor-Based Motion Planning for Mobile Robots in Unknown Environments

Abstract. Sensor based navigation enables a robot to explore an unknown environment and build a map of it, using its sensor information. This is one of the traditional research subjects for mobile robots and many algorithms have been proposed. In this work, we developed some strategies for sensor-based motion planning in unknown environments, based in the rapidly-exploring random trees (RRT) approach. These strategies explore the unknown area in a similar way than the Sensor-based Random Tree (SRT) method and adapt the RRT approach to the local path planning in the free space state associated with the safe region. We named our strategies Local SRRT and Local B SRRT. Experimental results are presented, in order to show the performance of the proposed strategies.

Keywords: RRT, SRT, motion planning, mobile robots.

1. Introducción

La navegación ambiental está comprendida de dos comportamientos principales, llamados exploración y planificación de caminos. Explorar un ambiente desconocido y buscar un punto objetivo está más allá de las tareas básicas de un robot móvil autónomo. Ambos problemas han recibido una atención considerable en geometría computacional y robótica; ver [2,3].

El enfoque RRT, presentado en [1], se ha convertido en el referente más popular en los últimos años. Los algoritmos basados en RRT fueron primero desarrollados para problemas de planificación no holonómica y cinemática [4], donde el espacio a ser explorado es el espacio de estados. Un método para exploración basada en sensores de ambientes desconocidos fue presentado en [5], este método procede de la construcción de una estructura de datos llamada SRT. El SRT representa un roadmap del área explorada asociada a su región segura y estima el espacio libre percibido por el robot durante la exploración. El método SRT original trata con robots holonómicos, Espinoza et al., han presentado algunas alternativas para tratar con robots móviles no holonómicos [6,7]. En este trabajo se emplearon algunas estrategias, basadas en el enfoque RRT para la navegación basada en sensores en ambientes desconocidos.

Las estrategias de exploración denominadas Local SRRT y Local B SRRT, con dos variantes diferentes cada una de ellas, adaptan a los planificadores de RRT para que se ejecuten localmente en el área segura construida con la información proporcionada por los sensores. De manera incremental, estas estrategias construyen un RRT a medida que el sensor informa el espacio libre del entorno.

El documento está organizado de la siguiente manera. La Sección II explica los detalles de las estrategias propuestas para la exploración y la planificación de movimientos. La Sección III analiza el desempeño de los dos planificadores propuestos, Local SRRT Local y Local B SRRT. Finalmente, las conclusiones y el trabajo futuro se presentan en la Sección IV.

2. Exploración y planificador de movimientos

El objetivo principal del presente trabajo es tomar ventaja del potencial de un método como RRT [1,4] en temas de exploración y el manejo de restricciones no holonómicas y expandirlo a exploración de ambientes desconocidos usando robots móviles. La idea es construir un árbol aleatorio local (del inglés, LRT, Local Random Tree) que representa la conectividad de un espacio de estado libre asociado con una región segura detectada por el sensor. El árbol puede ser expandido incrementalmente con las nuevas percepciones del ambiente obtenidas con el sensor, antes de que el estado meta sea un nodo en la LRT, cuando el procedimiento encuentra un camino factible libre o un máximo número de iteraciones se alcanza y el proceso concluye.

Se representa la construcción de una estructura llamada LRT, donde cada nodo contiene un estado libre de colisiones, una entrada de control y una descripción de la región segura local asociada con el estado. Cada arista en el árbol significa que el robot puede moverse de un estado a otro usando el control de entrada. Los nodos del árbol pueden ser de dos tipos: 1) Intermedio, un nodo utilizado para el movimiento del robot

dentro de la región segura local (LSR, Local Safe Region) y 2) Terminal, un nodo donde se puede realizar un nuevo proceso de percepción del entorno.

```

CONSTRUIR( $x_{inic}$ ,  $K_{max}$ ,  $I_{max}$ )
1   $x_{act} = x_{inic}$ ;
2   $k \leftarrow 0$ ;
3  mientras ( NO(META_EN_RSL( $x_{act}$ )) o  $NO(x_{act} = NULL)$  o  $k < K_{max}$ )
4     $S \leftarrow$  PERCEPCION( $x_{act}$ );
5    AGREGA( $T$ , ( $x_{act}$ ,  $S$ ));
6     $i \leftarrow 0$ ;
7    repetir
8       $x_{cand} \leftarrow$  EDO_CANDIDATO( $x_{act}$ );
9       $ExisteCamino \leftarrow$  PLANIFICADOR_RRT( $x_{act}$ ,  $x_{cand}$ );
10      $i \leftarrow i + 1$ ;
11     hasta que ( $ExisteCamino$  o  $i = I_{max}$ )
12     si  $ExisteCamino$  entonces
13       MOVER_A( $x_{cand}$ );
14        $x_{act} \leftarrow x_{cand}$ ;
15     sino
16       MOVER_A( $x_{act}$ ,  $padre$ );
17        $x_{act} \leftarrow x_{act}$ . $padre$ ;
18      $k \leftarrow k + 1$ ;

19 si (META_EN_RSL( $x_{act}$ )) entonces
20    $ExisteCamino \leftarrow$  PLANIFICADOR_RRT( $x_{act}$ ,  $x_{meta}$ );
21   si  $ExisteCamino$  entonces
22     Regresa EXITO;
23 si ( $x_{act}=NULL$ ) entonces
24   "Robot en la posición inicial ";
25 Regresa FALLO;

```

Fig. 1. Algoritmo básico del SRT.

Opcionalmente, este nodo puede incluir la referencia a una lista de estados candidatos para un nuevo proceso de percepción del entorno. Esta lista es generalmente referenciada en los nodos terminales. La descripción del robot, los obstáculos y el LSR son poligonales [7]. El procedimiento comienza en el estado inicial con un proceso de percepción del entorno, para determinar la región segura local asociada al estado. Después de eso, verifica si el estado del objetivo está en el LSR; de lo contrario, un planificador RRT alcanza un estado candidato y el ciclo se repite, expandiendo el LRT y aumentando el entorno explorado. Cuando se agota la lista de candidatos, el robot volverá al estado anterior. Este proceso de movimiento hacia atrás puede mover el robot al estado inicial; En este caso el algoritmo termina con fallo. El procedimiento también terminará con fallo después de un número máximo de iteraciones. El algoritmo finaliza con éxito cuando el estado objetivo está en la región segura local actual y hay una ruta que conecta los estados inicial y final.

El algoritmo explora el área desconocida de manera similar al método SRT y adapta el enfoque RRT a la planificación del camino local en el espacio de estados libre asociado con la región segura local. En cada iteración k del algoritmo, ver la Figura 1, se realiza un proceso de percepción para obtener la región segura local circundante S del robot, en el estado actual, X_{act} . Se crea un nuevo nodo que contiene el estado y la descripción de la LSR y se agrega al árbol como un nodo terminal. El algoritmo verifica

si el estado objetivo X_{goal} está en la región segura local de X_{act} , si no, un procedimiento determina un estado candidato válido dentro de la región segura para llevar a cabo un nuevo proceso de percepción.

Las restricciones diferenciales del robot tipo carro se manipulan con el planificador RRT que se ejecuta localmente en el espacio de estados libre para encontrar un camino factible entre los estados actual y candidato. Cada nodo creado por el planificador local se agrega al árbol como un nodo intermedio. Si el proceso de planificación local es exitoso, el nodo candidato se agrega al árbol como un nodo terminal y el ciclo se repite; de lo contrario, el algoritmo busca otro estado candidato válido de X_{act} , que el planificador pueda vincular con éxito con el estado actual, a través de un camino factible, o hasta que se alcance un número de iteraciones máximas I_{max} .

Si este es el caso, el robot regresa al nodo principal de X_{act} y el ciclo comienza de nuevo. El proceso hacia atrás puede llevar al robot hasta el estado inicial, en este caso el algoritmo termina con un error, es decir, no es posible ejecutar más percepciones. De este algoritmo surgen diferentes estrategias de exploración, que están relacionadas con la elección de los estados candidatos y la construcción del árbol de exploración. En las siguientes subsecciones se describirán las estrategias Local SRRT (RRT basado en sensores) y Local B SRRT. Hay dos variantes del Local SRRT que adaptan la ejecución local del planificador RRT en el proceso de exploración basado en sensores. Las variantes difieren en la forma en que construyen el árbol de exploración.

EDO_CANDIDATO(X_{act}). Los estados candidatos para un nuevo proceso de percepción del ambiente son generados con la rutina EDO_CANDIDATO(X_{act}). En el SRT local, la elección de candidatos es simple y sigue los siguientes pasos:

1. Una dirección aleatoria se selecciona para el primer estado candidato, X_{cand} y debe de estar dentro de la LSR dejando un margen en la frontera para asegurar la movilidad del robot, como en el algoritmo presentado en [7] (Figura 2).
2. El estado es validado si está localizado a una distancia mayor de un mínimo establecido y no está localizado dentro de la LSR de un nodo.
3. Los estados candidatos siguientes pueden ser generados en direcciones aleatorias o se puede usar un método más determinista, localizarlos en direcciones equidistantes de X_{act} . También deber ser validados de acuerdo al paso 2.

Los estados candidatos validos son almacenados en una lista de nodos, donde se toma la percepción del ambiente. La descripción de la LSR se debe calcular y opcionalmente almacenarse. Cuando el robot se mueve de X_{act} a X_{cand} , puede incrementar la información del espacio libre añadiendo o calculando la descripción de las LSRs de los nodos previos. El número de candidatos previos fue seleccionado experimentalmente con buenos resultados.

En este punto se realizaron dos maneras de generar nodos candidatos, ambas formas fueron mencionadas previamente, ver las Figuras 2 y 3. La primera forma consiste en generar cinco nodos candidatos distribuidos en una dirección aleatoria a la misma distancia de X_{act} entre sí en un área desconocida. La segunda forma consiste en generar los cinco nodos candidatos para que sean distribuidos en una distancia equidistante de cada nodo cercano a la frontera de la LSR en un área desconocida. Con ambos

planteamientos se busca que los candidatos cerca de la frontera de áreas no conocidas sean válidos, y con la distribución homogénea en cinco direcciones se busca eliminar la posibilidad de no visitar una zona no explorada. Una vez seleccionado, el estado candidato es marcado como visitado para no volver y hacer una exploración por segunda vez en esa zona.

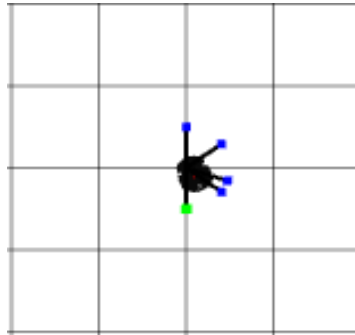


Fig. 2. Generación de nodos candidatos en direcciones aleatorias.

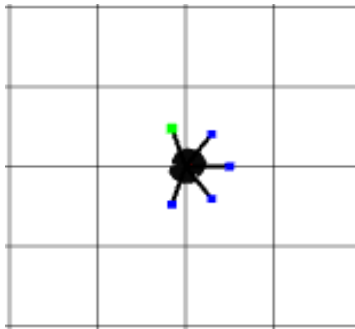


Fig. 3. Generación de nodos candidatos en direcciones equidistantes.

PLANIFICADOR_RRT. El enfoque del algoritmo RRT esta formulado en termino de 6 componentes:

1. Espacio de estados: Un espacio topológico, X .
2. Valores límite: $X_{ini} \in X$ y $X_{meta} \subset X$.
3. Detector de colisión: Una función, $D: X \rightarrow \{\text{verdadero, falso}\}$, que determina si las restricciones globales son satisfechas desde el estado X_{act} . Esta podría ser una función binario o real.
4. Entradas: Un conjunto, U , que especifica el conjunto de controles o acciones que pueden afectar el estado.
5. Simulador incremental: Dado X_{act} y las entradas aplicadas sobre un intervalo de tiempo, $\{u(t') \mid t \leq t' \leq t + \Delta t\}$, calcular $X(t + \Delta t)$.

6. Métrica: Una función real, $\rho: X \times X \rightarrow [0, \infty)$, la cual especifica la distancia entre pares de puntos en X .

La planificación de caminos generalmente es vista como una búsqueda en el espacio de estados, X , para un camino continuo desde un estado inicial X_{ini} a un estado final X_{meta} . Se asume que se tiene un conjunto complicado de restricciones diferenciales sobre X y cualquier solución debe mantener al estado dentro de este conjunto. Un detector de colisión reporta si un estado dado, X , satisface las restricciones globales. Generalmente, se utiliza la notación, X_{libre} para referirse al conjunto de estados que satisfacen las restricciones globales. Se asignan restricciones locales y diferenciales a través de un conjunto de entradas (o controles) y de un simulador incremental. Estos dos componentes especifican los posibles cambios en el estado. El simulador incremental puede definirse por integración numérica de una ecuación de transición de estado. Finalmente, se define una métrica para indicar la cercanía de pares de puntos en el espacio de estados.

El planificador se desarrolló de acuerdo con el modelo de movimiento de velocidad (ver figura 4) el cual asume que podemos controlar al robot mediante 2 velocidades, una velocidad rotacional y una velocidad traslacional, donde denotamos la velocidad traslacional en tiempo t por v_t y la velocidad rotacional por ω_t .

$$u_t = \begin{pmatrix} v_t \\ \omega_t \end{pmatrix}$$

Fig. 4. Modelo de movimiento de velocidad.

META_EN_LSR(X_{act}). Esta función regresa verdadero si X_{meta} se encuentra dentro de la LSR de X_{act} . Esto se puede validar calculando la distancia entre la posición de X_{act} con la posición de X_{meta} y comparándola con la longitud del rayo que une la posición actual con un punto en la frontera de la LSR de X_{meta} .

MOVER_A(X_{cand}). Esta rutina simplemente sigue las entradas de control generadas por el planificador local para llegar el estado X_{cand} desde el estado X_{act} . Opcionalmente en cada movimiento se actualiza la región segura con nuevas percepciones del ambiente.

EXISTE_CAMINO(). Esta función determina si existen nodos X_{cand} pertenecientes al nodo X_{act} y que no hayan sido visitados, en caso de existir nodos que aún no han sido visitados el nodo X_{act} se convierte en el nodo X_{cand} seleccionado, en caso contrario al ya no haber nodos por visitar el nodo X_{act} se convierte en el nodo $X_{act.padre}$ generando así un encadenamiento hacia atrás.

PERCEPCIÓN(X_{act}). El ambiente es percibido mediante un sensor láser con un campo de visión de 240° , como ya se ha mencionado el número máximo de nodos X_{cand} es 5, se dividió el campo de visión en 5 regiones iguales, dando lugar a las 2 posibles maneras de generar un EDO_CANDIDATO(X_{act}), ver la figura 5.

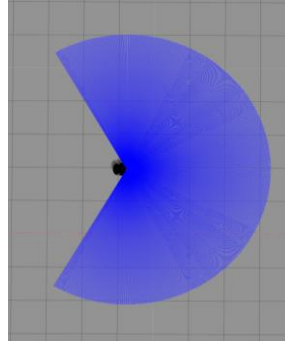


Fig. 5. Vista aérea del robot Pioneer 3DX con el Laser Hokuyo en el simulador Gazebo.

3. Resultados experimentales

Esta sección presenta resultados experimentales del robot Pioneer 3DX. SRT fue implementado en la distribución de ROS Kinetic Kame, ROS agrega valor a muchos proyectos y aplicaciones en robótica. ROS fue diseñado para ser distribuido tan modular como sea posible, permitiendo al usuario utilizar mucho o poco de ROS como lo deseen o necesiten. La modularidad de ROS permite seleccionar que partes usar y que partes del proyecto desarrollar e implementar.

Las pruebas fueron desarrolladas en un procesador Intel CORE i7 7th Gen con 2.7 GHz con 16 GB RAM. Las variantes de estrategias de exploración son referenciadas como variante 1 y variante 2 de Local SRRT y Local B SRRT. Se utilizaron diferentes ambientes para poner a prueba la eficiencia de las variantes, ver las Figuras 6, 7, 8, 9, 10, y 11. Se utilizaron diferentes ambientes (Ambiente 1, Ambiente 2, y Ambiente 3) para poner a prueba la eficiencia de ambas variantes de Local SRRT, en las tablas 1 y 2 se muestran los resultados obtenidos respecto al número de nodos y el tiempo de ejecución. El tiempo de ejecución proporcionado por las simulaciones corresponde al tiempo total, incluyendo el tiempo de percepción del sensor.

Los caminos planificados con la variante uno de la estrategia de exploración Local SRRT, son factibles para los movimientos del robot móvil de tipo carro, pero son difíciles de seguir por que requieren muchas maniobras de conducción por parte del robot.

Esto sucede por la unión de pequeños arboles bidireccionales y la elección aleatoria del siguiente estado para un nuevo sensado, ambas características obligan al algoritmo a encontrar un camino factible en un espacio reducido que se traduce en muchas maniobras de conducción. La segunda variante de Local SRRT surgió motivada por este problema, ¿cómo obtener caminos que cumplan con las restricciones globales del ambiente y del robot y que además sean un poco menos complicados de seguir?



Fig. 6. Estrategia Local SRRT en el escenario 1.

Tabla 1. Resultados de Local SRRT.

	Ambiente 1	Ambiente 2	Ambiente 3
Nodos (mín.)	40	31	34
Nodos (máx.)	51	58	46
Tiempo (mín.)	119.12 segs.	77.12 segs	103.95 segs.
Tiempo (máx.)	240.01 segs.	371.56 segs.	331.78 segs.

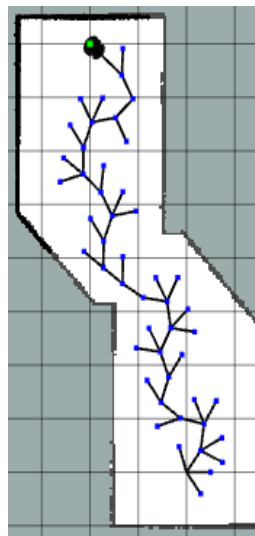


Fig. 7. Estrategia Local B SRSRT en el escenario 1.

Tabla 2. Resultados de Local B SRRT.

	Ambiente 1	Ambiente 2	Ambiente 3
Nodos (mín.)	38	27	31
Nodos (máx.)	51	44	49
Tiempo (mín.)	110.79 segs.	71.26 segs.	92.7 segs.
Tiempo (máx.)	238.8 segs.	262.2 segs.	348.3 segs.

El hecho de que la variante dos utilice al árbol aleatorio de exploración como árbol de planificación local da mayores oportunidades al planificador local de encontrar un nodo que conecte al estado candidato, sin obligarlo a construir caminos factibles en espacios reducidos, sino de aprovechar caminos ya construidos a lo largo de la región segura. En ocasiones el algoritmo ocupa un porcentaje de tiempo ligeramente mayor que la variante uno, en su búsqueda del nodo más cercano al cual conectar el nodo aleatorio producido por el planificador local.

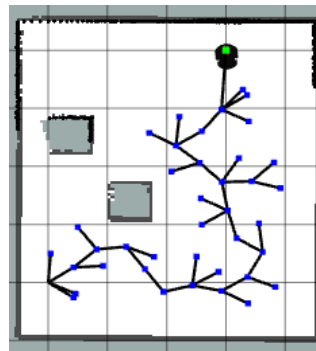


Fig. 8. Aplicación de la variante Local SRRT en el escenario 2 registrando un tiempo de 234.6 segundos.

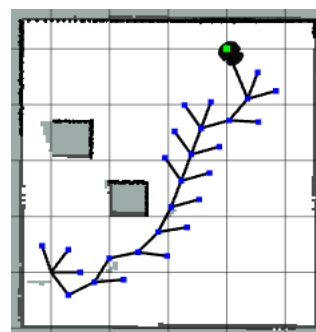


Fig. 9. Aplicación de la variante Local B SRRT en el escenario 2 registrando un tiempo de 75.98 segundos.

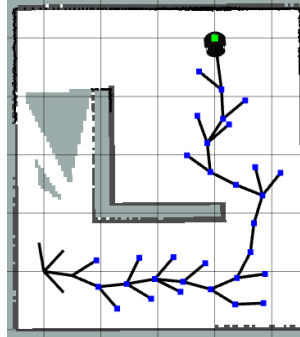


Fig. 10. Aplicación de Local SRRT en el escenario 3 registrando un tiempo de 120.6 seg.

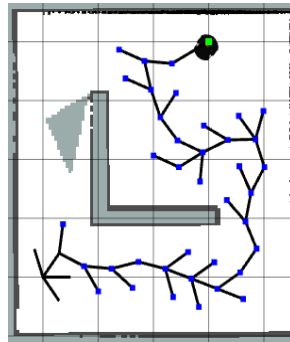


Fig. 11. Aplicación de Local B SRRT en el escenario 3 registrando un tiempo de 184.74 seg.

En la tabla número 3, se describen los resultados y puntos meta de las figuras 6, 7, 8, 9, 10, y 11; se visualizan los árboles aleatorios de exploración usando sensores (SRT), mediante la percepción del sensor se pueden generar y validar estados candidatos a través de espacios reducidos o explorados.

Tabla 3. Descripción general de las figuras.

	Tiempo segs.	Nodos	Meta (X, Y)
Figura 6	198.2	49	(-2, 8)
Figura 7	183.5	50	(-2, 8)
Figura 8	234.6	40	(3, 4)
Figura 9	75.98	28	(3, 4)
Figura 10	120.6	36	(3, 4)
Figura 11	184.74	43	(3, 4)

La ventaja que presentan las variantes de la estrategia Local B SRRT sobre la estrategia Local SRRT es que no arrastran con el problema original de los planificadores RRT–bidireccionales, que necesitan conectar varios vértices para unir un árbol al otro. En Local B SRRT la expansión del árbol de exploración se da al

construir un nuevo árbol local a partir de un nodo ya existente en el árbol de exploración y de conectar los nodos aleatorios que produce el planificador local al árbol de exploración con raíz en el estado meta o al árbol local con raíz en el estado actual.

Las estrategias de exploración conservan la habilidad de escapar de mínimos locales al realizar el proceso de retroceso hacia el estado inicial, al excederse el número máximo de intentos para encontrar un nodo válido o después de evaluar todos los nodos en la lista de estados candidatos válidos para el siguiente proceso de percepción del ambiente. De igual forma, los algoritmos mantienen la característica de terminar con fallo después de un tiempo máximo, e implementan otra forma de conclusión derivada del proceso de retroceso. El algoritmo termina cuando el robot regresa al estado inicial, asumiendo que no es posible incrementar la información del espacio libre a través de los sensores.

La siguiente Figura muestra unas instantáneas que ilustran la ejecución en tiempo real con el robot Pioneer 3DX.

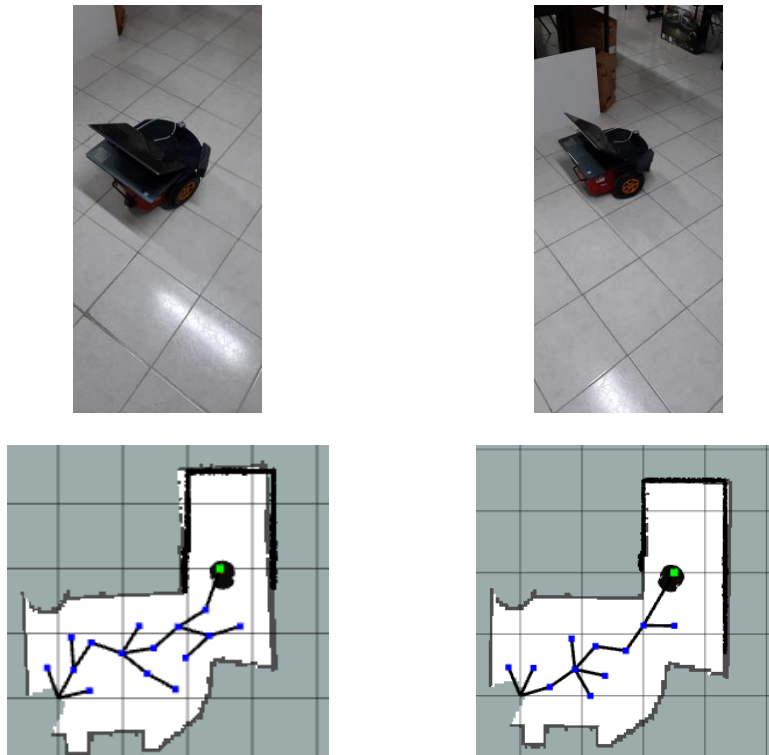


Fig. 12. Ejecución en tiempo real con el robot Pioneer 3DX.

4. Conclusiones y trabajo a futuro

En este trabajo se desarrollaron dos estrategias basadas en el enfoque RRT aplicados en ambientes desconocidos. Las estrategias son llamadas Local SRRT y Local B SRRT

con su determinada variante para generar nodos candidatos generando así una región segura construida a partir de las lecturas del sensor laser.

Los resultados obtenidos a través de las simulaciones muestran que las estrategias aplicadas resuelven el problema principal que es encontrar una meta, usando como apoyo la información sensorial, encontrando así un camino factible del estado inicial al estado meta (considerando las restricciones globales del ambiente y la información obtenida por los sensores del robot) en ambientes simples y complejos. Adaptamos al método, la estrategia de percepción SRT Radial [7], para aprovechar lo más posible la información proporcionada por los sensores.

Como ya hemos visto en la sección de resultados, ambas estrategias consiguen llegar a la meta, no obstante, en términos computacionales el método Local B SRRT arroja mejores resultados que el Local SRRT. Estas estrategias de exploración adaptan los planificadores RRT para ejecutarse localmente en el área segura construida por los sensores.

Las estrategias construyen incrementalmente un árbol aleatorio de exploración como el sensor reporta el espacio libre del ambiente. El árbol refleja la conectividad del espacio libre conocido donde el robot móvil se mueve para reconocer mayores porciones del espacio físico. Para explorar el ambiente, las estrategias utilizan una simple elección de estados candidatos para un nuevo proceso de sensado, dependiendo de la estrategia, guiados con una heurística sencilla sobre la distancia euclidiana que separa al estado candidato del estado meta. En el árbol, los nodos se distinguen por dos clases, nodos intermedios, aquellos donde se lleva a cabo un proceso de percepción del ambiente; y nodos terminales, nodos que sirven para darle movilidad al robot que opcionalmente tienen una descripción de la región segura local y principalmente contienen la entrada de control.

Como trabajo futuro, sería probar las estrategias en robots reales, situación que estamos concretando.

Referencias

1. LaValle, S.M., Kuffner, J.: Rapidly-exploring random trees. In: Progress and prospects. Workshop on Algorithmic Foundations of Robotics (2000)
2. Deng, X., Kameda, T., Papadimitriou, C.: How to learn an unknown environment I: The rectilinear case. *Journal of the ACM* 45(2), 215–245 (1998)
3. Mitchell, J.S.: Shortest paths and networks. In: Goodman, J.E., O'Rourke, J. (eds.): *Handbook of Discrete and Computational Geometry*, Chapter 24, pp. 445–466 (1997)
4. LaValle, S.M., Kuffner, J.: Randomized kinodynamic planning. *International Journal of Robotics Research* 20(5), 378–400 (2001)
5. Oriolo, G., Vendittelli, M., Freda, L., Troso, G.: The SRT method: Randomized strategies for exploration. In: *IEEE Int. Conf. On Robotics and Automation*, pp. 4688–4694 (2004)
6. Espinoza Leon, J.: Estrategias para la exploración de ambientes desconocidos en robótica móvil. Tesis de Maestría, FCC-BUAP (2006)
7. Espinoza, L.J., Sanchez, L.A., Osorio, L.M.: Exploring unknown environments with mobile robots using SRT-Radial. In: *IEEE Int. Conf. On Intelligent Robots and Systems*, pp. 2089–2094 (2007)