

Sobre la mejora esperada de la estimación de la odometría en exploración integrada

Toriz Palacios, A.^{a,*}, Sánchez López, A.^b

^a *Departamento de Posgrados en Ingeniería Mecatrónica, Universidad Popular Autónoma del Estado de Puebla, Calle 17 Sur 711, Barrio de Santiago, 72410 Puebla, México.*

^b *Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, Av San Claudio 14 Sur, Cd Universitaria, 72592 Puebla México.*

To cite this article: Toriz Palacios, A., Sánchez López, A. 2020. On the expected improvement of odometry estimation in integrated exploration. *Revista Iberoamericana de Automática e Informática Industrial* 17, 229-238. <https://doi.org/10.4995/riai.2020.11828>

Resumen

El problema de exploración integrada es la nueva tendencia en la construcción de mapas de ambientes desconocidos; en ella, se integra el viejo paradigma de la localización y mapeo simultáneos (SLAM) con la planificación de movimientos necesarios, para que esta tarea sea realizada de forma autónoma. Sin embargo, aunque el control de movimientos es una parte esencial de este paradigma, los trabajos encontrados en la literatura se han limitado a desarrollar estrategias que mejoren los tiempos de recorridos y la cobertura del ambiente, dejando de lado el impacto que estos pueden tener sobre la odometría del robot y, en consecuencia, sobre los requerimientos de los algoritmos de localización. De lo anterior, en este documento se presenta una nueva forma eficiente de exploración de ambientes para el problema de SLAM, que tiene como objetivo mejorar los tiempos de exploración y maximizar la cobertura del área de trabajo, pero además el de minimizar el error odométrico acumulado para simplificar el proceso de localización.

Palabras clave: Robot móvil autónomo, planificación de rutas, estimación de movimiento, errores de posición odométrica, tasa de error.

On the expected improvement of odometry estimation in integrated exploration

Abstract

The problem of Integrated Exploration is the new trend in the construction of maps of unknown environments; in it, the old paradigm of Simultaneous Localization and Mapping (SLAM) is integrated with the planning of movements necessary for this task to be performed autonomously. However, although motion control is an essential part of this new paradigm, the existing literature has been limited to developing strategies that improve travel times and environmental coverage, leaving aside the impact that these can have on robot odometry and, consequently, on the requirements of localization algorithms. Accordingly, this document presents a new efficient way of exploring environments for the SLAM problem, which aims to improve exploration times and maximize coverage of the work area, as well as minimize the accumulated odometric error to simplify the localization process.

Keywords: Autonomous mobile robot, path planning, motion estimation, position errors, error rates.

1. Introducción

La obtención de mapas de ambientes desconocidos de forma autónoma, utilizando mecanismos eficientes para explorar y modelar entornos cada vez más complejos, es uno de los retos fundamentales de la robótica actual; su importancia radica en que estas representaciones constituyen uno de los requisitos principales para casi todas las tareas en robótica

móvil. En general, la tarea de adquirir modelos de entornos desconocidos requiere la solución de tres sub tareas, que son mapeo, localización y control de movimiento, también conocidos como el problema de exploración integrada. Aunque la localización y mapeo simultáneo (SLAM) ha logrado grandes progresos en los últimos años, el control de movimientos, también conocido como exploración, es un tema que ha quedado relegado, ya que los métodos desarrollados

hasta el momento, en el mejor de los casos, solo se centran en realizar la tarea en el menor tiempo posible, maximizando la superficie de exploración cubierta.

La exploración de ambientes puede ser definida como el análisis y toma de decisiones que permiten dirigir al robot hacia áreas prometedoras para recopilar información, lo cual puede ser una tarea difícil de resolver, dado que su finalidad es operar en ambientes desconocidos. De esta forma, muchos trabajos pueden ser encontrados en la literatura que abordan el problema de exploración de ambientes desconocidos (Burgard et al., 2005), (Franchi et al., 2007), (Franchi et al., 2009), (Freda et al., 2006), (Juliá et al., 2008), (Campos et al., 2017), (Prieto et al., 2013), (Chen et al., 2017), siendo los más difundidos los basados en el análisis de fronteras.

Así, (Franchi et al., 2007) presentan en su trabajo el método de exploración del Árbol Aleatorio Basado en Sensor (SRT); el cual, se basa en la construcción incremental de una estructura de datos tipo árbol que representa el roadmap del área explorada, y donde los nodos de la estructura representan una posición dentro del mapa alcanzada por el robot, y una zona de seguridad local (RSL) percibida por los sensores. En este método, la exploración es guiada desde un nodo actual donde se encuentra el robot, lanzando un número finito de configuraciones aleatorias dentro de la RSL para tratar de encontrar una frontera libre donde continuar la exploración. Si el robot se encuentra en una zona donde no es posible encontrar fronteras libres, este retorna automáticamente al nodo padre de la posición actual para buscar nuevas zonas inexploradas. El método termina cuando el proceso de retroceso automático lleve al robot a la raíz de la estructura.

Franchi presenta una extensión del método SRT para el caso multi-robot llamado Grafo Aleatorio de Exploración (Franchi et al., 2009), que consiste en modificar la estructura de exploración tipo árbol a una estructura tipo grafo. En este método, la siguiente posición a explorar está determinada por el uso de una probabilidad proporcional a la longitud del arco de las fronteras libres encontradas en el nodo en el que actualmente se encuentra el robot. Además, la forma de revisitar zonas no exploradas dejadas atrás consiste en generar un árbol de expansión mínima con todos los nodos adyacentes del grafo al nodo actual, y el robot elige el árbol del nodo adyacente con el mayor peso con respecto a longitud de fronteras libres.

En su trabajo, Toriz presenta el método del grafo aleatorio de exploración (REG) (Toriz et al., 2017); el cual, toma como base el método SRT y lo optimiza a través de una estructura de grafo de exploración. En este método, la siguiente posición a explorar está determinada a través de una selección aleatoria de una de las fronteras libres capturadas por el sensor; sin embargo, el punto de aproximación a ella se obtiene de tal forma que maximice la ganancia de información evitando dejar arcos disjuntos de frontera. Además, en este método se introduce un nuevo concepto conocido como el control de fronteras, el cual permite acumular conocimiento sobre las zonas no completamente exploradas que se dejaron atrás y que necesitan ser revisitadas para cubrir completamente el área de trabajo. Con este concepto y la estructura de grafo generada, es posible generar trayectorias óptimas que permitan continuar la exploración.

Más recientemente, NingboYu propuso un método de exploración, creado, como un proceso de decisión de Markov parcialmente observable (Yu and Wang, 2019). Se combinaron métodos de simulación de mapas parciales y de búsqueda de fronteras globales para la exploración autónoma, y se utilizaron restricciones de acción dinámica en el control de movimiento. Para esto, se utilizaron sensores de profundidad (RGB-D) que proporcionan información tanto de color como de profundidad sobre el entorno.

Como puede observarse, los métodos presentados conservan un carácter aleatorio para definir la siguiente posición a explorar, propiciando un excesivo cambio de direcciones del robot, lo que conduce a la acumulación de errores de orientación en el sistema odométrico y, en consecuencia, a grandes errores de posición (Borenstein, 1998), que aumentan proporcionalmente con la distancia recorrida por el robot.

La mayoría de los investigadores coinciden en que la odometría es una parte importante del sistema de navegación de un robot y que estas tareas se simplificarán si se puede mejorar la precisión odométrica, ya que tener una buena estimación de la odometría es necesario para establecer la selección de las landmarks que serán utilizadas en el proceso de localización del robot. Aunque es posible encontrar trabajos en la literatura que hacen referencia a técnicas, métodos, e incluso sensores más precisos, que permiten reducir la acumulación del error odométrico en el robot para el problema de SLAM (Romero et al., 2002), (Cadena et al., 2016), (Hidalgo-Carrió et al., 2017), (Jin and Chung, 2019), (Lou et al., 2019), (Maddahi et al., 2012), (Maddahi, 2018), estos omiten completamente las implicaciones del control de movimiento en el tema de la exploración de ambientes desconocido en su más amplia definición (Cadena et al., 2016).

De lo anterior, es necesario considerar que la forma de explorar ambientes tiene un efecto directo sobre la odometría del robot y en consecuencia sobre la localización y el mapeo de los ambientes. De hecho, una mala odometría podría impedir que los algoritmos de SLAM operen de forma correcta (Torres-González et al., 2014), ya que podría traer como consecuencia un problema catastrófico de asociación de datos, que, aunque es un tópico que ha sido ampliamente tratado en la literatura (Hähnel et al., 2005), aun depende de una buena estimación de la posición del robot (Hanif et al., 2018).

Así, en este trabajo de investigación se presenta una nueva forma de exploración de ambientes desconocidos, el cual tiene como objetivos minimizar el tiempo requerido para realizar la tarea, garantizar una cobertura total del ambiente, y minimiza el impacto del error odométrico acumulado sobre el sistema del robot.

2. Grafo Determinístico de Exploración

El problema de SLAM usa métodos de optimización en combinación con puntos de referencia para corregir el estado del robot inicialmente predicho por su odometría; así, desde los inicios de la robótica, la mejora en el rendimiento de las técnicas de odometría ha sido una constante para los robots móviles (Brossard and Bonnabel, 2018), ya que una buena estimación requerirá de un proceso de localización que consumirá menos recursos para encontrar la ubicación real del robot.

Al tratar de medir y reducir errores de odometría, es importante entender la distinción entre errores de odometría "sistemáticos" y "no sistemáticos" (Lamon et al., 2007). Los errores sistemáticos están caracterizados en la literatura (Abbas et al., 2006), (Ojeda and Borenstein, 2004) como aquellos que son intrínsecamente parte de las propiedades de la cinemática o el controlador, independientemente del entorno y control del robot, es decir, aquellos causados por el propio hardware del robot. Por otro lado, los errores no sistemáticos son complejos y difíciles de predecir, dependen del entorno del robot y de los movimientos realizados por este, y son precisamente estos los que forman el centro de atención de esta propuesta. Es necesario mencionar que los errores no sistemáticos tienen una alta correlación con los errores sistemáticos y, por lo tanto, no se pueden eliminar totalmente a nivel de software; sin embargo, sí pueden minimizarse a través de un control de movimientos eficiente.

El Grafo determinístico de Exploración (DEG) presentado en este artículo, es un método basado en fronteras que toma como base el concepto de tasa de crecimiento de errores (Borenstein, 1998), el cual distingue entre "Crecimiento Rápido" y "Crecimiento Lento" del error odométrico. La tasa de crecimiento argumenta que, mientras en una operación lineal de un robot el error acumulado en la dirección de desplazamiento del robot crece de manera acelerada, el error en el desplazamiento lateral crece de forma lenta; sin embargo, este error crece de forma ilimitada como consecuencia de los cambios de orientación, y se agrava debido a cambio de orientación demasiado violentos.

De lo anterior, el método DEG genera una estructura de exploración de tipo grafo, la cual se extiende hacia áreas prometedoras a través de un mecanismo determinístico de selección de la siguiente posición a explorar, el cual tratará de minimizar el número de movimientos y cambios de dirección que el robot tiene que realizar. La estructura de grafo, se utiliza para hallar caminos óptimos que permiten al robot navegar dentro de regiones ya exploradas, lo que permitirá, por ejemplo, regresar a posiciones previamente visitadas que no hayan sido completamente exploradas sin desperdiciar tiempo en la navegación.

El carácter determinístico del método DEG radica en la forma en que el robot selecciona la siguiente posición a explorar. Para esto, se establecerá sobre cada frontera libre detectada por el robot en la posición actual un punto de aproximación en el que, las longitudes de los arcos de estas fronteras libres sean cubiertas completamente por los límites establecidos del sensor, o en su defecto, un punto de aproximación que permita abarcar la mayor extensión de frontera libre, evitando dejar fronteras discontinuas (Figura 1b). Lo anterior, tendrá un impacto directo en el tiempo de exploración debido a que, cuando sea necesario un regreso a esta zona, el sistema se encontrará con una sola frontera inexplorada, en lugar de encontrar dos o más en direcciones diferentes (Figura 1a).

Una vez que los puntos de aproximación quedan establecidos en las fronteras libres, se elige como posición objetivo aquella que requiera menos esfuerzo para ser alcanzada, y sobre la cual, se realizará un proceso de planificación de trayectorias basado en el algoritmo del árbol aleatorio de exploración rápida (RRT) (LaValle,1998) para

encontrar una trayectoria óptima hacia ella que evite giros sobre el eje del robot, o giros demasiado pronunciados (Figura 2). El resto de los puntos de aproximación serán almacenados en una lista de exploración, junto con la información del nodo al que pertenecen. Esta lista será utilizada para planificar un camino a estos nodos con posibilidad de exploración (Figura 3b) cuando sea necesario un retroceso hacia zonas inexploradas. Así, los puntos objetivo almacenados en la lista de exploración permanecerán en ella hasta que sean visitados o hasta que la RSL de otro nodo cubra la frontera inexplorada que fue dejada atrás.

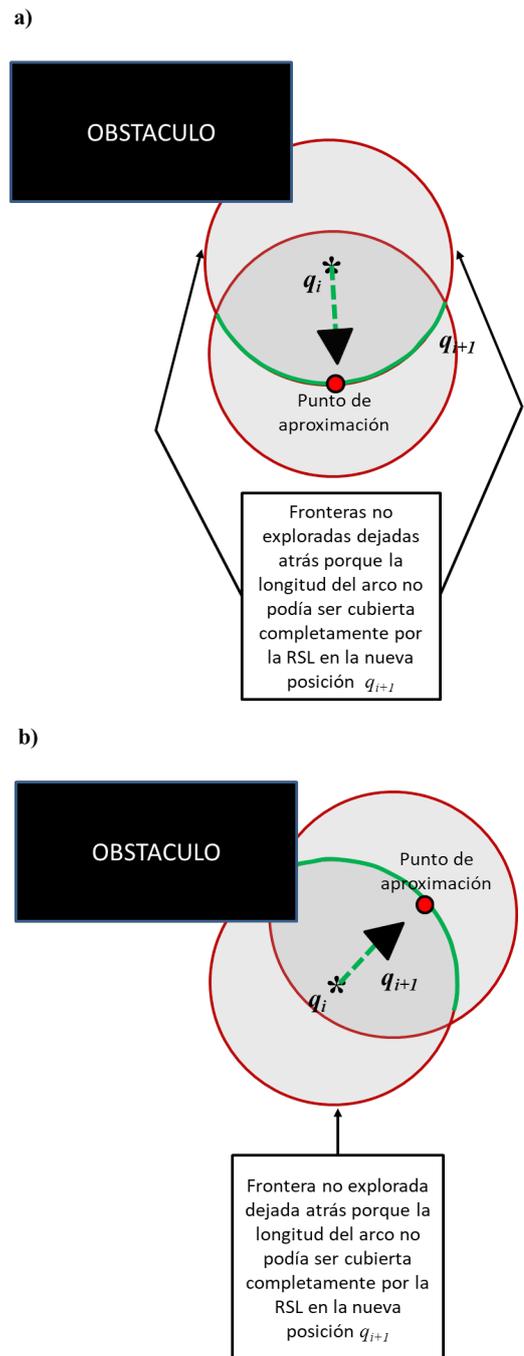


Figura 1. Criterio utilizado por el método DEG para establecer los puntos de aproximación sobre las fronteras libres detectadas por el robot en la posición q_i . a) Selección inadecuada del punto de aproximación, ya que este deja dos nuevas fronteras libres inexploradas segmentadas en direcciones opuestas. b)

Selección correcta del punto de aproximación, el cual no dejen fronteras segmentadas.

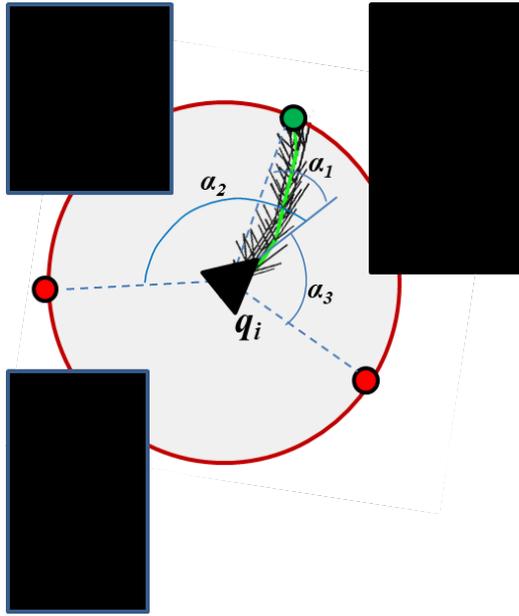
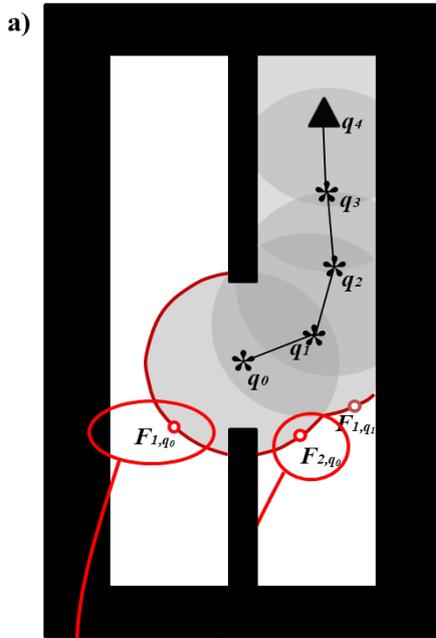


Figura 2. Selección de la siguiente posición a explorar y camino encontrado mediante el método RRT para llegar a ella. El círculo verde representa la posición destino elegida por el método DEG.



b)

| Lista de nodos no completamente explorados | | | |
|--|----------------------------|--|--|
| Nodos | Fronteras libres | Puntos de aproximación | Longitud de los arcos |
| q_0 | F_{1,q_0} F_{2,q_0} | (X_{1,q_0}, Y_{1,q_0}) (X_{2,q_0}, Y_{2,q_0}) | $Longitud(F_{1,q_0})$ $Longitud(F_{1,q_0})$ |
| q_1 | F_{1,q_1} | (X_{1,q_1}, Y_{1,q_1}) | $Longitud(F_{1,q_1})$ |

Figura 3. Lista de exploración. a) Ambiente semi explorado donde los arcos F_{i,q_j} representan fronteras libres no exploradas, b) Lista de nodos no explorados completamente.

Finalmente, el proceso de retroceso hacia zonas no completamente exploradas dejadas atrás una vez que el robot alcance una zona que no pueda seguir siendo explorada, será ejecutada usando el método de búsqueda A* en forma bidireccional, utilizando la estructura de grafo creada, planificando la ruta desde el nodo actual a los puntos objetivo contenidos en la lista de exploración y, desde estos puntos objetivo, hacia el nodo actual, finalizando cuando exista un camino entre el nodo actual y cualquiera de los puntos objetivo en la lista (Figura 4).

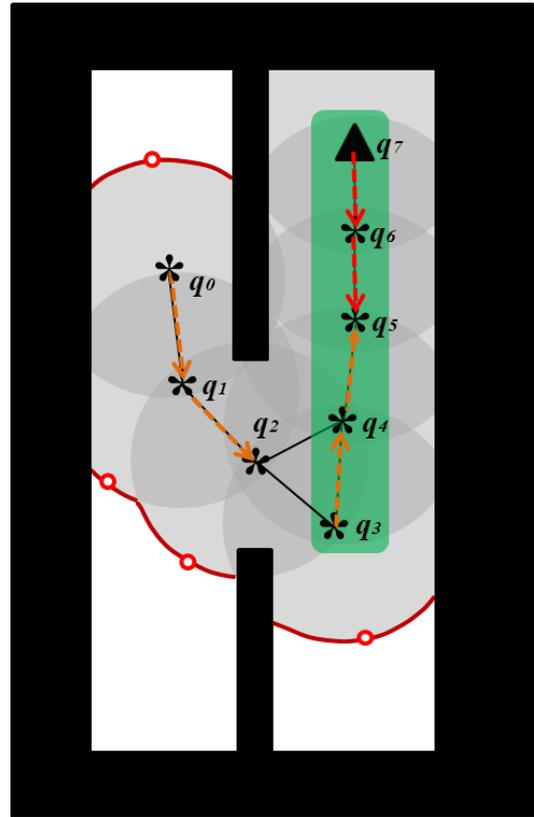


Figura 4. Aplicación bidireccional del método A*, desde la posición actual del robot (Flechas punteadas color rojo) hacia los nodos con posibilidad de exploración almacenados en el control de fronteras, y desde los nodos con posibilidad de exploración (Flechas punteadas color naranja) hacia la posición actual del robot. La ruta encontrada está representada en la región verde

2.1. Algoritmo del Grafo Determinístico de Exploración

Como se ha mencionado, el método DEG se basa en la construcción progresiva de una estructura de datos tipo grafo que representa el roadmap del área explorada. La forma de construcción del grafo consiste en extender la estructura hacia fronteras libres detectadas por los sensores, y dirigir al robot hacia ellas a través de un método determinístico cuya filosofía es la de realizar la menor cantidad de movimientos posible. Cada nodo del grafo de exploración construido por el método DEG, está compuesto de una posición q libre de colisión alcanzada por el robot, y de una descripción S del ambiente (RSL) que rodea a q percibida por el sistema sensorial; además, los arcos de la estructura entre nodos adyacentes, representan un camino libre de colisión por el que el robot puede viajar sin riesgo de colisionar con un obstáculo. La Figura 5 muestra el algoritmo del método DEG.

El algoritmo DEG genera un primer nodo que será considerado como el nodo de salida y de retorno; este contendrá la información del punto de partida q_{init} de la exploración, y de la RSL asociada a él. Una vez que este nodo es construido, el ciclo que controlará la exploración puede empezar.

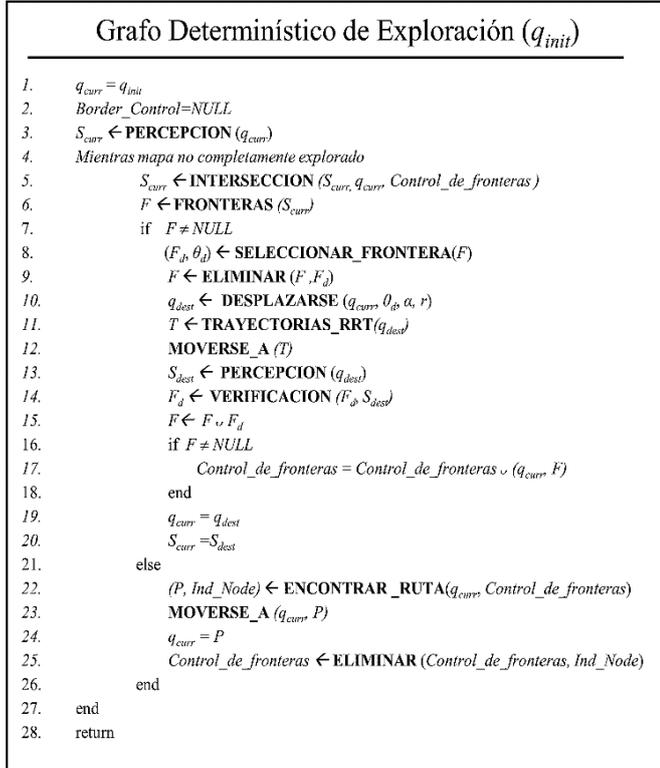


Figura 5. Algoritmo del Grafo Determinístico de Exploración

A continuación, el nuevo nodo será utilizado por la función INTERSECCIÓN para actualizar el estado de las fronteras libres de nodos vecinos cubiertas por la nueva RSL S_{curr} , las cuales pasará de “no visitada” a “visitada”; así mismo, las fronteras libres del nuevo nodo contenidas en nodos vecinos se marcarán como visitadas. Finalmente, la función identificará si existe un camino libre de colisión entre las fronteras que intersectan los nodos vecinos con el nodo actual, en cuyo caso se agregará una nueva arista a la estructura de exploración indicando que los pares de nodos son adyacentes.

Una vez realizada la verificación y actualización de las fronteras, el siguiente paso es identificar cuáles, de entre las que pertenecen al nodo actual, continúan guardando el estatus de “No visitadas” para continuar la exploración. Esto es realizado por la función FRONTERAS. Si la función arroja al menos una frontera F libre no explorada, la función SELECCIONAR_FRONTERA definirá un mecanismo de selección determinístico para continuar la exploración. Así, la función determinará en cada frontera detectada un punto objetivo, el cual se establece en el punto medio de la longitud de los arcos de las fronteras, siempre que esta longitud no exceda un cierto umbral proporcional a la distancia que la nueva RSL podrá cubrir. En caso contrario, el punto objetivo será fijado tomando el punto medio del segmento de frontera proporcional a la longitud del arco que puede ser cubierta por la nueva RSL, empezando en el extremo derecho. Finalmente,

la nueva frontera a explorar será seleccionada considerando el punto objetivo que requiera de un menor cambio de orientación para ser alcanzada y esta será eliminada del grupo de fronteras libres obtenidas con la función FRONTERAS, ya que por ahora ya no es libre. En caso de que la esta función haya detectado más de una frontera, aquellas que no fueron seleccionadas serán agregadas a una lista de exploración.

A continuación, la función DESPLAZARSE obtiene la nueva posición q_{dest} a ser visitada, dando un paso de longitud $\alpha \cdot r$ en la dirección del punto objetivo de la frontera seleccionada; en esta expresión, r representa el radio de la RSL y la constante $\alpha < 1$ asegura que la nueva configuración destino se encuentre dentro de la RSL actual.

Después, la función TRAYECTORIAS_RRT utiliza el algoritmo RRT para definir un camino dentro de la RSL que permitan llevar al robot desde la posición actual hasta la posición destino minimizando los cambios de orientación del robot y evitando giros sobre su propio eje; para esto, se definió empíricamente a partir de una serie de pruebas, la creación de 10 diferentes árboles de exploración rápida orientados en un rango cerrado de $\pm 30^\circ$ a partir de la orientación actual del robot, de los cuales se elegirá aquel cuya trayectoria generada contenga la menor cantidad de cambios de orientación. La función MOVERSE_A será la encargada de llevar al robot desde la posición actual hasta la posición objetivo q_{dest} utilizando la trayectoria obtenida en el paso anterior.

Una vez que el robot ha alcanzado la posición objetivo, se obtiene una descripción S_{dest} para estimar el espacio circundante a la posición q_{dest} . Usando esta información, la función VERIFICACIÓN determinará qué porción de la frontera elegida en el paso anterior fue realmente cubierta por la nueva LSR. Si la frontera no fue cubierta totalmente, esta será agregada en el grupo de fronteras F del nodo previo que deberán ser exploradas. La información acerca de las fronteras libres restantes en el nodo previo lo clasificará como un nodo con posibilidad de exploración. Finalmente, el nuevo nodo creado se agrega a la estructura del grafo y el ciclo comienza nuevamente usando la nueva posición q_{dest} y su RSL S_{dest} como información de la posición actual.

Típicamente, cuando la exploración no puede encontrar una frontera libre hacia donde avanzar, se dice que la zona ha sido completamente explorada y la tarea debe redirigirse a áreas previamente visitadas almacenadas en la lista de exploración, donde fronteras libres “No exploradas” fueron dejadas atrás. La búsqueda de una nueva zona para continuar la exploración es llevada a cabo por la función ENCONTRAR_RUTA, la cual utilizará el algoritmo de búsqueda en grafos A* en forma bidireccional, extendiendo la búsqueda de un camino desde la posición actual en donde se encuentra el robot, y desde los nodos con posibilidad de exploración almacenados en la lista de exploración en el lado opuesto. El algoritmo finalizará cuando se encuentre un camino P .

Una vez que la trayectoria P es obtenida, el método MOVERSE_A llevará al robot desde el nodo actual hasta el nodo con posibilidad de exploración. Finalmente, el índice del nodo elegido es eliminado de la lista de exploración y el método DEG continuará realizando el proceso descrito, para que ningún elemento quede en la lista de exploración y, por lo tanto, ninguna frontera permanezca inexplorada.

3. Resultados Experimentales

En esta sección, se muestran los resultados de numerosos experimentos que se han llevado a cabo para validar el método de exploración propuesto en esta investigación. Para evaluar la efectividad de nuestro método, se realizó un estudio comparativo con el método del “grafo aleatorio de exploración” presentado en (Toriz et al., 2017); el cual, fue sometido a las mismas pruebas y a las mismas condiciones de operación. Esta decisión fue tomada, ya que ambos métodos son clasificados como “basados en fronteras”, mismos que han probado tener una alta eficiencia en la exploración de ambientes para SLAM (Holz et al., 2010); al mismo tiempo, el método REG ha probado tener un mejor desempeño que otros métodos similares (Toriz et al., 2017).

Las pruebas, evaluaron variables clásicas utilizadas para medir el desempeño de métodos de exploración, como los son tiempo y cobertura del mapa; sin embargo, siendo el principal objetivo de nuestro método el de minimizar el impacto que tiene la exploración sobre el error odométrico no sistemático, también se presentan pruebas que muestran el comportamiento de esta variable. Los datos se obtuvieron sobre la base empírica de 30 pruebas, ya que es en este punto cuando se estabilizó la desviación estándar del método comparativo REG.

Para las pruebas, se utilizó una adaptación del ambiente jardín presentado en (Franchi et al., 2007), el cual es mostrado en la Figura 6; además, se simuló un robot diferencial Pioneer P3DX equipado con un sensor laser Hokuyo URG-04lx, el cual tiene un rango de detección de 0.02 a 4 m aproximadamente, con una desviación atípica del 1% de la medida, una resolución angular de 0.36° y un ángulo de escaneo de 240°. Además, el robot tiene un anillo de 16 sensores ultrasónicos, de los cuales 6 de ellos están posicionados en la parte trasera y se usan para obtener información del ambiente en los 120°, donde el sensor láser no tiene cobertura.

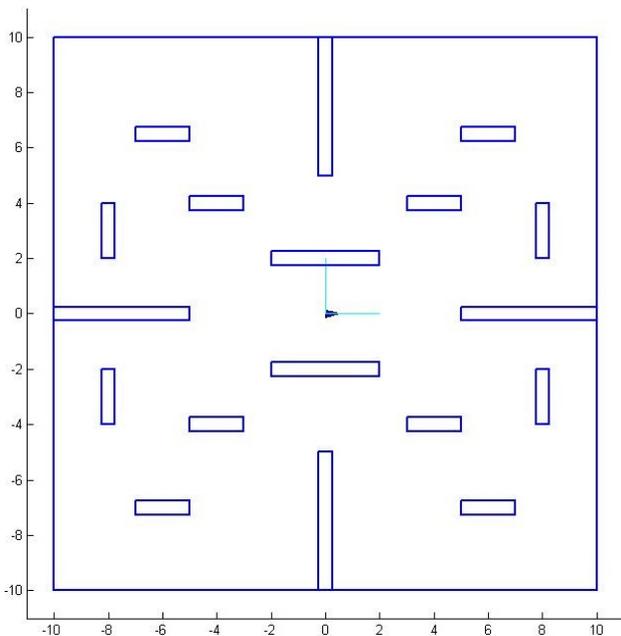


Figura 6. Ambiente SRT jardín utilizado para las pruebas

La Figura 7 muestra la estructura generada por el método DEG una vez finalizada la exploración; en ella, es fácil

observar el modo de operación del método y la cobertura del ambiente de trabajo de trabajo.

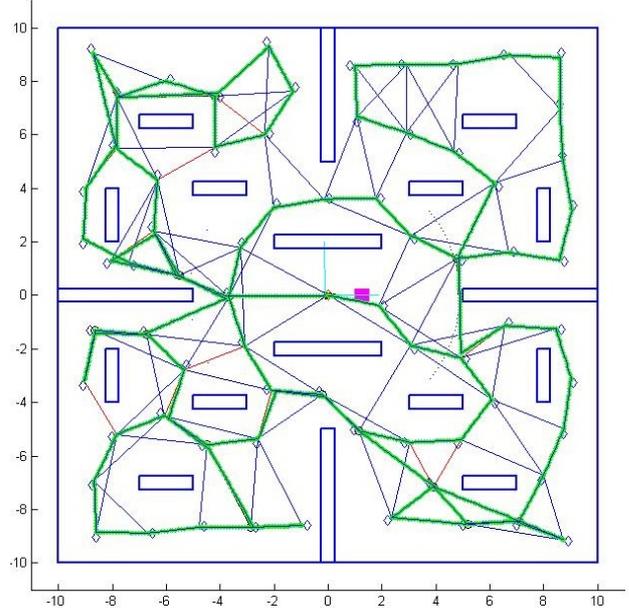


Figura 7. Estructura y trayectoria generadas por el método DEG

Con respecto a los datos comparativos obtenidos, tanto el método REG como el método DEG lograron una cobertura del 100% del área en todas las pruebas realizadas; esto, gracias al conocimiento acumulativo generado por ambos métodos sobre qué áreas no se fueron exploradas completamente. Para medir este parámetro, el ambiente de trabajo se dividió en celdas de exploración como lo sugiere el método presentado en (Gil et al., 2015); sin embargo, para nuestro caso, la división del ambiente sólo se utilizó para determinar qué celdas fueron exploradas, lo que nos permitió conocer la superficie total examinada por los métodos (Figura 8).

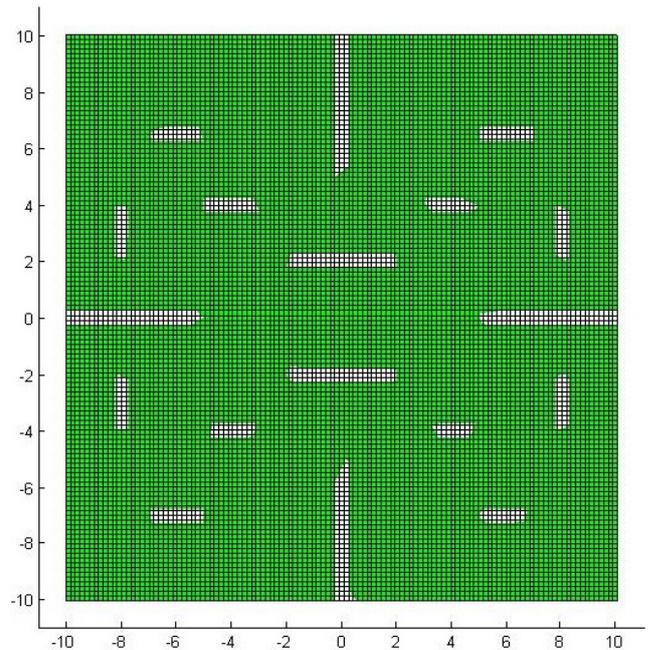


Figura 8. Mapa obtenido dividido en rejillas para determinar el porcentaje de cobertura de los métodos de exploración

Sobre el tiempo requerido para completar la exploración, los resultados se muestran en la Tabla 1; en ella, puede observarse que el método DEG requirió menos tiempo para completar la tarea, y que la desviación estándar es mucho menor para el método DEG, lo cual puede explicarse en base a la forma determinista en que este método toma la decisión de dirigir la exploración.

Tabla 1. Tiempos de exploración obtenidos durante las pruebas por los métodos REG y DEG

| Número de prueba | Tiempo total requerido para completar la exploración por el método DEG | Tiempo total requerido para completar la exploración por el método REG |
|---------------------------------------|--|--|
| 1 | 825.640 | 1030.937 |
| 2 | 829.502 | 961.623 |
| 3 | 801.898 | 1135.340 |
| 4 | 845.028 | 1026.707 |
| 5 | 845.607 | 1137.710 |
| 6 | 838.786 | 1093.894 |
| 7 | 824.184 | 1112.798 |
| 8 | 819.447 | 1074.635 |
| 9 | 833.836 | 950.195 |
| 10 | 849.485 | 1148.659 |
| 11 | 819.953 | 960.116 |
| 12 | 822.007 | 961.598 |
| 13 | 844.421 | 1050.055 |
| 14 | 801.516 | 1088.129 |
| 15 | 812.154 | 1020.022 |
| 16 | 815.359 | 1144.013 |
| 17 | 833.817 | 1075.370 |
| 18 | 830.063 | 1145.751 |
| 19 | 816.935 | 984.593 |
| 20 | 822.587 | 1067.288 |
| 21 | 811.771 | 1078.404 |
| 22 | 815.604 | 960.414 |
| 23 | 847.476 | 1071.893 |
| 24 | 830.152 | 1091.413 |
| 25 | 832.206 | 1097.742 |
| 26 | 838.545 | 1126.433 |
| 27 | 828.100 | 1042.094 |
| 28 | 849.260 | 1070.876 |
| 29 | 834.604 | 1061.961 |
| 30 | 840.345 | 1130.799 |
| Tiempo Promedio de exploración | 828.676 | 1077.008 |
| Desviación Estándar | 13.423 | 62.224 |

Las Figuras 9, 10 y 11 muestran los errores odométricos acumulados en las coordenadas X e Y de la posición del robot, así como el error acumulado en su orientación. Como se puede ver, los errores reportados por el método DEG son considerablemente menores que los reportados por su contraparte. Esto se debe a que los errores de orientación acumulativa causan serios problemas de ubicación en el sistema odométrico; por lo tanto, al evitar excesivos cambios de orientación, el error también se minimiza, lo que es un principio fundamental del método DEG y que lo diferencia de métodos aleatorios como lo es el método REG. También, es necesario recordar que el error odométrico no se puede reducir a cero, ya que su precisión depende en gran medida del hardware utilizado.

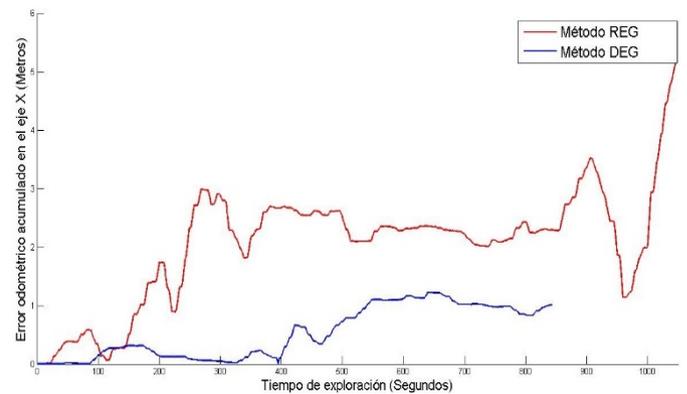


Figura 9. Comparativo de errores odométricos acumulados en la coordenada X por los métodos DEG y REG, obtenidos durante las pruebas

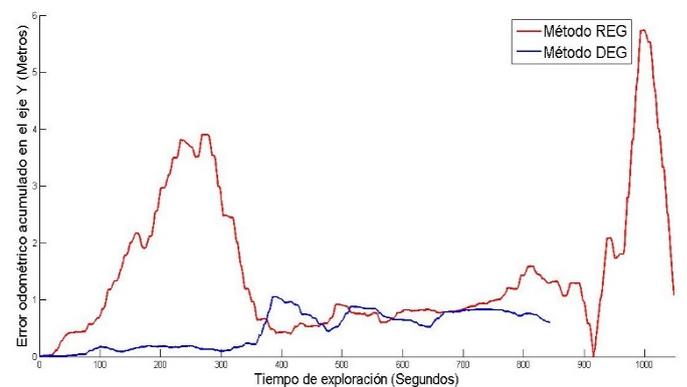


Figura 10. Comparativo de errores odométricos acumulados en la coordenada Y por los métodos DEG y REG, obtenidos durante las pruebas

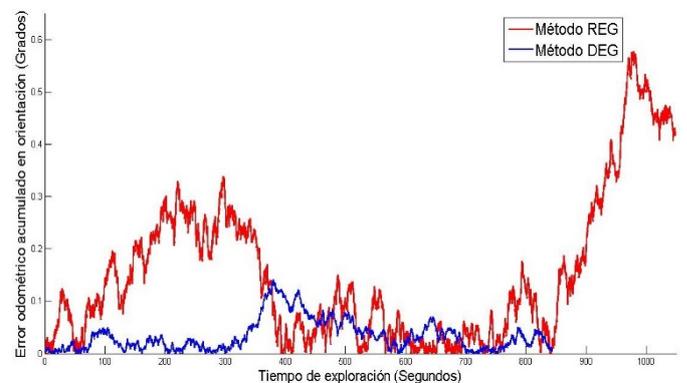


Figura 11. Comparativo de errores odométricos acumulados en orientación por los métodos DEG y REG, obtenidos durante las pruebas

Las Tablas 2,3 y 4 muestran los resultados completos de los errores odométricos obtenidos en las 30 pruebas.

Tabla 2. Errores odometricos máximos en el eje X, obtenidos durante las pruebas por los métodos DEG y REG

| Número de prueba | Error odométrico acumulado máximo en el eje X obtenido por el método DEG (Metros) | Error odométrico acumulado máximo en el eje X obtenido por el método REG (Metros) |
|--|---|---|
| 1 | 1.2201 | 5.2554 |
| 2 | 1.6086 | 6.1287 |
| 3 | 1.6448 | 5.0656 |
| 4 | 1.4327 | 5.9799 |
| 5 | 1.7355 | 6.9570 |
| 6 | 1.5315 | 4.9756 |
| 7 | 1.4104 | 4.2827 |
| 8 | 1.2711 | 5.5217 |
| 9 | 1.7132 | 6.6202 |
| 10 | 1.2227 | 6.3613 |
| 11 | 1.3700 | 5.5423 |
| 12 | 1.8160 | 6.3332 |
| 13 | 1.5522 | 6.4990 |
| 14 | 1.5541 | 4.2396 |
| 15 | 1.8782 | 4.3500 |
| 16 | 1.2816 | 4.1910 |
| 17 | 1.3345 | 5.5357 |
| 18 | 1.0988 | 4.7093 |
| 19 | 1.1208 | 4.3560 |
| 20 | 1.9776 | 6.5626 |
| 21 | 1.3049 | 4.2058 |
| 22 | 1.4792 | 5.8699 |
| 23 | 1.0362 | 6.5425 |
| 24 | 1.5100 | 6.0549 |
| 25 | 1.4607 | 5.9436 |
| 26 | 1.5946 | 5.4270 |
| 27 | 1.6065 | 5.8807 |
| 28 | 1.3975 | 4.9343 |
| 29 | 1.0700 | 5.7410 |
| 30 | 1.7105 | 5.9786 |
| Error odométrico promedio en el eje X | 1.4648 | 5.5348 |
| Desviación Estándar | 0.2426 | 0.8358 |

Tabla 3. Errores odometricos máximos en el eje Y, obtenidos durante las pruebas por los métodos DEG y REG

| Número de prueba | Error odométrico acumulado máximo en el eje Y obtenido por el método DEG (Metros) | Error odométrico acumulado máximo en el eje Y obtenido por el método REG (Metros) |
|--|---|---|
| 1 | 1.0510 | 5.7380 |
| 2 | 1.3437 | 7.8838 |
| 3 | 1.2551 | 5.1917 |
| 4 | 1.7452 | 5.5304 |
| 5 | 1.7376 | 6.6835 |
| 6 | 1.1825 | 6.3161 |
| 7 | 1.4428 | 5.7131 |
| 8 | 1.6824 | 7.7810 |
| 9 | 1.5293 | 7.0823 |
| 10 | 1.7640 | 6.3921 |
| 11 | 1.8081 | 4.5518 |
| 12 | 1.4443 | 4.6615 |
| 13 | 1.2459 | 7.0443 |
| 14 | 1.5359 | 6.1156 |
| 15 | 1.4020 | 7.5653 |
| 16 | 1.5122 | 7.3976 |
| 17 | 1.2206 | 5.3003 |
| 18 | 1.2743 | 7.9258 |
| 19 | 1.1076 | 5.8897 |
| 20 | 1.5252 | 5.8625 |
| 21 | 1.1928 | 5.4909 |
| 22 | 1.6136 | 5.1548 |
| 23 | 1.7497 | 5.4774 |
| 24 | 1.3241 | 5.5842 |
| 25 | 1.6182 | 6.2326 |
| 26 | 1.1511 | 6.1464 |
| 27 | 1.7193 | 6.1193 |
| 28 | 1.8422 | 6.0504 |
| 29 | 1.9771 | 7.8889 |
| 30 | 1.1254 | 5.5536 |
| Error odométrico promedio en el eje Y | 1.4708 | 6.2108 |
| Desviación Estándar | 0.2567 | 0.9686 |

Tabla 4. Errores odométricos máximos en orientación, obtenidos durante las pruebas por los métodos DEG y REG

| Número de prueba | Error odométrico acumulado máximo en orientación obtenido por el método DEG (Grados) | Error odométrico acumulado máximo en orientación obtenido por el método REG (Grados) |
|---|--|--|
| 1 | 0.1407 | 0.5770 |
| 2 | 0.1569 | 0.5946 |
| 3 | 0.1130 | 0.5073 |
| 4 | 0.1273 | 0.5208 |
| 5 | 0.1138 | 0.4625 |
| 6 | 0.1366 | 0.5658 |
| 7 | 0.1443 | 0.4310 |
| 8 | 0.1390 | 0.5828 |
| 9 | 0.1160 | 0.6782 |
| 10 | 0.1391 | 0.6537 |
| 11 | 0.1283 | 0.5187 |
| 12 | 0.1205 | 0.5321 |
| 13 | 0.1138 | 0.4638 |
| 14 | 0.1324 | 0.4611 |
| 15 | 0.1547 | 0.5069 |
| 16 | 0.1237 | 0.6308 |
| 17 | 0.1552 | 0.4919 |
| 18 | 0.1215 | 0.5019 |
| 19 | 0.1257 | 0.4526 |
| 20 | 0.1445 | 0.6351 |
| 21 | 0.1433 | 0.4632 |
| 22 | 0.1421 | 0.5561 |
| 23 | 0.1177 | 0.6928 |
| 24 | 0.1255 | 0.6138 |
| 25 | 0.1269 | 0.5122 |
| 26 | 0.1292 | 0.6190 |
| 27 | 0.1349 | 0.4223 |
| 28 | 0.1473 | 0.5548 |
| 29 | 0.1579 | 0.4992 |
| 30 | 0.1296 | 0.6292 |
| Error odométrico promedio en orientación | 0.1334 | 0.5444 |
| Desviación Estándar | 0.0134 | 0.0756 |

Finalmente, los mapas obtenidos después de realizar la exploración del ambiente con los métodos DEG y REG son mostrados en las Figuras 12 y 13; en ambos casos, el mapeo solo considera información odométrica para posicionar espacialmente los obstáculos encontrados.

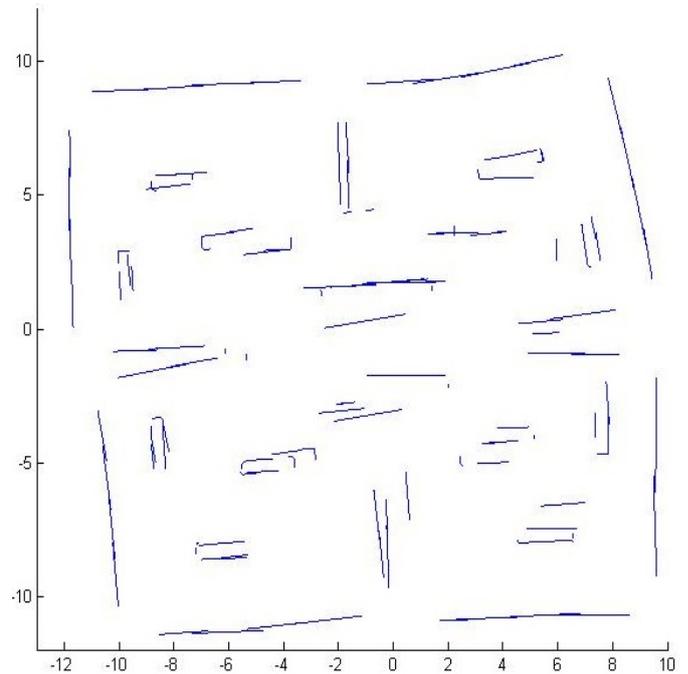


Figura 12. Mapa del ambiente SRT obtenido con el método de exploración DEG, considerando solo información odométrica

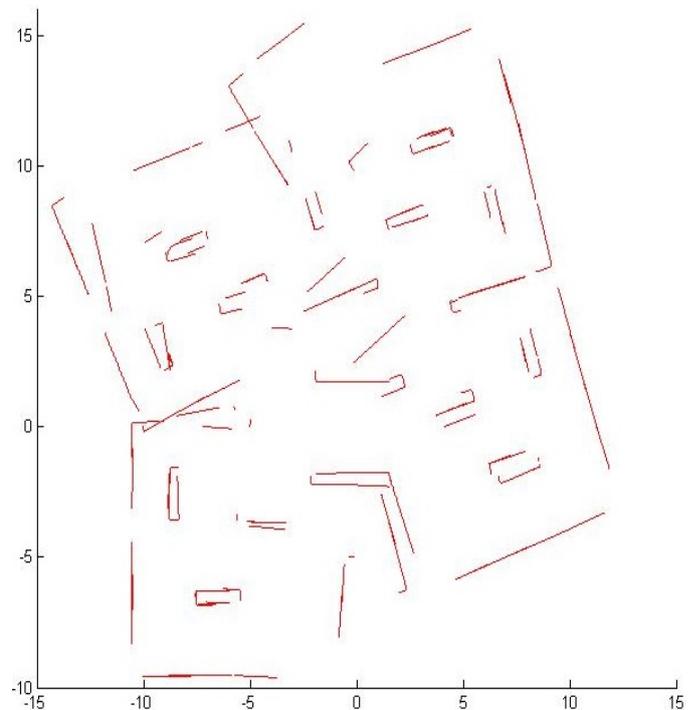


Figura 13. Mapa del ambiente SRT obtenido con el método de exploración REG, considerando solo información odométrica

En estas figuras puede notarse que, aunque ambos mapas contienen errores, el obtenido por el método DEG presenta una

aproximación más cercana al ambiente utilizado en las pruebas.

4. Conclusiones

Como hemos mencionado, la exploración de ambientes para SLAM es un elemento fundamental que permite dotar de verdadera autonomía a los robots móviles en las tareas de adquisición de mapas de ambientes desconocidos. Este problema sin embargo, necesita considerarse como el control de movimientos afecta a la ubicación odométrica del robot, y en consecuencia, a los recursos que serán necesarios para las tareas de localización y el mapeo.

De lo anterior, en este trabajo se presentó una estrategia para el problema de exploración; la cual, está basada en el uso de un método de exploración determinista que permite minimizar los movimientos que el robot tiene que realizar para completar la tarea y maximizar la ganancia de información del ambiente, lo que reduce significativamente la acumulación del error odométrico. A diferencia de otros métodos de exploración presentados en la literatura, la principal contribución realizada en este artículo es un algoritmo enfocado no solo a explorar ambientes desconocidos en la menor cantidad de tiempo posible, sino a realizar un control de movimientos efectivo durante la tarea de exploración que impacte directamente en la odometría del robot.

El método presentado, fue sometido a numerosos experimentos en un ambiente simulado, lo que permitió validar la robustez de la propuesta en variables tales como, el tiempo de exploración, la cobertura del ambiente, y el error odométrico obtenido.

Además, se realizaron pruebas comparativas con un método de reciente aparición en la literatura conocido como el método REG; el cual, en sí mismo, ha demostrado ser mejor que otras propuestas de los llamados métodos “basados en fronteras”. En estas pruebas, el método DEG arrojó datos significativamente mejores, lo que permite afirmar que nuestra aproximación tiene un mejor rendimiento.

Referencias

- Abbas, T., Arif, M., Ahmed, W., 2006. Measurement and correction of systematic odometry errors caused by kinematics imperfections in mobile robots. *SICE-ICASE International Joint Conference, 2073-2078*. DOI: 10.1109/SICE.2006.315554
- Borenstein, J., 1998. Experimental results from internal odometry error correction with the OmniMate mobile robot. *IEEE Transactions on Robotics and Automation, 14(6), 963-969*. DOI: 10.1109/70.736779
- Brossard, M., Bonnabel, S., 2018. Learning Wheel Odometry and IMU Errors for Localization. *IEEE International Conference on Robotics and Automation (ICRA)*,
- Burgard, W., Moors, M., Stachniss, C., Schneider, F. E., 2005. Coordinated multi-robot exploration. *IEEE Transactions on robotics, 21(3), 376-386*. DOI: 10.1109/TRO.2004.839232
- Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Leonard, J. J., 2016. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics, 32(6), 1309-1332*. DOI: 10.1109/TRO.2016.2624754
- Campos, F. M., Marques, M., Carreira, F., Calado, J. M. F., 2017. A complete frontier-based exploration method for Pose-SLAM. *IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), 79-84*. DOI: 10.1109/ICARSC.2017.7964056
- Chen, N. Y., Shaw, J., Lin, H. I., 2017. Exploration method improvements of autonomous robot for a 2-D environment navigation. *Journal of Marine Science and Technology, 25(1), 34-42*. DOI: 10.6119/JMST-016-0719-1
- Franchi, A., Freda, L., Oriolo, G., Vendittelli, M., 2007. A randomized strategy for cooperative robot exploration. *IEEE International Conference on Robotics and Automation, 768-774*. DOI: 10.1109/ROBOT.2007.363079
- Franchi, A., Freda, L., Oriolo, G., Vendittelli, M., 2009. The sensor-based random graph method for cooperative robot exploration. *IEEE/ASME Transactions on Mechatronics, 14(2), 163-175*. DOI: 10.1109/TMECH.2009.2013617
- Freda, L., Loiudice, F., Oriolo, G., 2006. A randomized method for integrated exploration. *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2457-2464*. DOI: 10.1109/IROS.2006.281689
- Gil, A., Juliá, M., Reinoso, Ó., 2015. Occupancy grid based graph-SLAM using the distance transform, SURF features and SGD. *Engineering Applications of Artificial Intelligence, 40, 1-10*. DOI: 10.1016/j.engappai.2014.12.010
- Hähnel, D., Thrun, S., Wegbreit, B., Burgard, W., 2005. Towards lazy data association in SLAM. *Eleventh International Symposium Robotics Research, 421-431*. DOI: 10.1007/11008941_45
- Hanif, M. S., Bilal, M., Munawar, K., Balamash, A. S., 2018. Implementation of an Embedded Testbed for Indoor SLAM. In *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, 1-8. DOI: 10.1109/AICCSA.2018.8612782
- Hidalgo-Carrió, J., Hennes, D., Schwendner, J., Kirchner, F., 2017. Gaussian process estimation of odometry errors for localization and mapping. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 5696-5701. DOI: 10.1109/ICRA.2017.7989670
- Holz, D., Basilico, N., Amigoni, F., Behnke, S., 2010. Evaluating the efficiency of frontier-based exploration strategies. In *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*, 1-8.
- Jin, J., Chung, W., 2019. Obstacle Avoidance of Two-Wheel Differential Robots Considering the Uncertainty of Robot Motion on the Basis of Encoder Odometry Information. *Sensors, 19(2), 289-299*. DOI: 10.3390/s19020289
- Juliá, M., Gil, A., Payá, L., Reinoso, O., 2008. Local minima detection in potential field based cooperative multirobot exploration. *International Journal of Factory Automation, Robotics and Soft Computing, 3*.
- Lamon, P., Siegart, R., 2007. 3D position tracking in challenging terrain. *The International Journal of Robotics Research, 26(2), 167-186*. DOI: 10.1007/978-3-540-33453-8_44
- LaValle, S. M., 1998. Rapidly-exploring random trees: A new tool for path planning.
- Lou, Q., González, F., Kövecses, J., 2019. Kinematic Modeling and State Estimation of Exploration Rovers. *IEEE Robotics and Automation Letters, 4(2), 1311-1318*. DOI: 10.1109/LRA.2019.2895393
- Maddahi, Y., Sepahi, N., Maddahi, A., Abdolmohammadi, M., 2012. Calibration of wheeled mobile robots with differential drive mechanisms: An experimental approach. *Robotica, 30(6)*. DOI: 10.1017/S0263574711001329.
- Maddahi, Y., 2018. Off-Line Calibration of Autonomous Wheeled Mobile Robots. In *Handbook of Research on Biomimetics and Biomedical Robotics, 375-389*. DOI: 10.4018/978-1-5225-2993-4.ch016.
- Ojeda, L., Borenstein, J., 2004. Methods for the reduction of odometry errors in over-constrained mobile robots. *Autonomous Robots, 16(3), 273-286*. DOI: 10.1023/B:AURO.0000025791.45313.01
- Prieto, R. A., Cuadra-Troncoso, J. M., Álvarez-Sánchez, J. R., Santosjuanes, I. N., 2013. Reactive Navigation and Online SLAM in Autonomous Frontier-Based Exploration. In *International Work-Conference on the Interplay Between Natural and Artificial Computation, 45-55*. DOI: 10.1007/978-3-642-38622-0_5
- Romero, L., Morales, E. F., Sucar, L. E., 2002. An exploration approach for indoor mobile robots reducing odometric errors. In *Mexican International Conference on Artificial Intelligence, 51-60*. DOI: 10.1007/3-540-46016-0_6
- Toriz P. A., Sánchez L. A., Bedolla Cordero, J. M. E., 2017. The random exploration graph for optimal exploration of unknown environments. *International Journal of Advanced Robotic Systems, 14(1)*. DOI: 10.1177/1729881416687110
- Torres-González, A., Martínez-de Dios, J., Ollero, A., 2014. An adaptive scheme for robot localization and mapping with dynamically configurable inter-beacon range measurements. *Sensors, 14(5), 7684-7710*. DOI: 10.3390/s140507684
- Yu, N., Wang, S., 2019. Enhanced Autonomous Exploration and Mapping of an Unknown Environment with the Fusion of Dual RGB-D Sensors. *Engineering, 5(1), 164-172*. DOI: 10.1016/j.eng.2018.11.01