ROBOT MÓVIL PARA INVESTIGACIÓN EN ALGORITMOS DE PLANEAMIENTO DE RUTAS: SISTEMA DE PLANEAMIENTO DE RUTAS

Acosta, Iván.
idacostat@utn.edu.ec
Carrera de Ingeniería en Mecatrónica, Universidad Técnica del Norte, Ibarra, Ecuador

Resumen—El presente artículo consiste en el estudio y simulación de algoritmos de planificación de trayectorias de un robot móvil. Entre los problemas más importantes de la robótica móvil se encuentra la planificación de trayectorias, que desempeña una tarea fundamental de la navegación autónoma del robot. Se realiza una interfaz gráfica en Matlab para simular los algoritmos de planificación. La interfaz de la ventana de simulación del algoritmo permite crear un mapa de rejilla en el que se pueden colocar las posiciones de inicio y llegada del robot, así como las posiciones de los obstáculos a través de los cuales el robot debe encontrar y graficar la ruta óptima. Los resultados de este proyecto reflejan la correcta aplicación de los algoritmos, ya que cada uno de ellos cumple con la función de encontrar la ruta con costo óptimo.

Índice de Términos—Planificación de trayectorias, A*, Dijkstra.

I. INTRODUCCIÓN

Para lograr la autonomía de un vehículo, este debe tener la capacidad de conocer su posición dentro de su ambiente. Un vehículo que no pueda localizarse a sí mismo corre el riesgo de chocar contra obstáculos, elegir las rutas inadecuadas o no poder evitar las áreas peligrosas.

En la actualidad existen varias plataformas de desarrollo en el área de la robótica móvil que son usadas para varias aplicaciones entre las que se encuentra la planificación de trayectorias. Sin embargo, estas plataformas tienen un costo demasiado elevado y son cerradas ya que no permiten actualización en su hardware. Es necesario tomar en consideración que la robótica móvil ha aportado con grandes soluciones para problemas cotidianos referentes a la navegación autónoma.

Hay que tener presente que dentro de la UTN, no existen plataformas de robots móviles, ya sean orientados para el uso académico o como base para el desarrollo de investigación. La causa para que no exista investigación dentro del área de la robótica móvil es atribuible a la inexistencia de plataformas que fomenten el desarrollo de estudio e investigación.

Por otro lado, en los laboratorios de mecatrónica existen hardware y software poco aprovechados, como son ciertas tarjetas con sistemas embebidos de gran capacidad computacional y software propietario con módulos de robótica. Usando éstos se obtendría un robot con costo bajo y de fácil construcción, cuya arquitectura podría ser escalable tanto en software como en hardware, diferenciándose notablemente de las soluciones presentes en el mercado local.

El objetivo simular el sistema de planificación de trayectoria de un robot móvil, realizando el recorrido del robot móvil a través de un mapa, evitando los obstáculos y calculando la ruta óptima; se emplea dos de los algoritmos más usados en este tipo de aplicaciones, como son el A* y de Dijkstra.

II. METODOLOGÍA

El desarrollo del proyecto de investigación se detalla a continuación, partiendo desde la investigación, el análisis de los algoritmos, la realización de la interfaz gráfica de simulación que está dividida desde una ventana principal y dos ventanas secundarias de cada algoritmo; se determina el funcionamiento de los algoritmos del simulador de planificación de trayectorias.

Investigación

Se realizó la búsqueda de información respecto al tema, que permitan tener un mayor conocimiento acerca de la robótica móvil, planeación de trayectorias y los algoritmos que pueden ser usados. Se investigó sobre el software que se usó para la programación de los algoritmos y creación de la interfaz gráfica necesaria para el simulador.

Planificación de trayectorias

El término planificación de trayectorias se ha desarrollado en muchos campos, como la robótica, la inteligencia artificial o la teoría de control; por tal motivo cada científico utiliza la definición propia de este término [8].

En la robótica, la planificación de rutas se refiere al problema de cómo mover un robot de un punto a otro. En la inteligencia artificial, la planificación de trayectoria significa

una búsqueda de una secuencia de acciones lógicas que transforman una posición inicial del robot en una posición objetivo deseado. En la teoría de control, la planificación de trayectos se ocupa de cuestiones de estabilidad, retroalimentación y optimización. Es por eso que la planificación de trayectorias de un robot móvil es un problema amplio y existen muchos métodos y enfoques [8].

Selección de los algoritmos

Los algoritmos a usar son el A* y el de Dijkstra por ser de los más usados en investigaciones de robótica móvil y son de los primeros algoritmos en aprender para problemas de planeación de trayectorias; estos dos algoritmos son para gráficos y sirven para el tipo de simulación en mapas de cuadricula [18].

*Algoritmo A**

El algoritmo de búsqueda A* fue presentado por Hart, Nilsson y Raphael en 1968, por el problema de determinar la ruta de costo mínimo a través de un gráfico [14].

En robótica, el algoritmo A* se emplea principalmente en una red ambiental que se incluye el conocimiento de la función heurística h(n). La función heurística seleccionada como función de distancia nos brinda la distancia entre la celda en expansión actual y la meta [15].

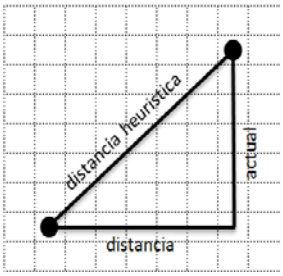

Figura 1. Distancia heurística.

Esta función heurística entre dos nodos es la distancia euclidiana, que es menor que la longitud real de la trayectoria en la cuadrícula; la función hace que el algoritmo sea eficiente para obtener conocimiento adicional sobre la cuadrícula [12] [15].

Se evalúa la función de evaluación de costos de cada celda mediante

$$f(n) = g(n) + h(n) \quad (1)$$

donde $n = (x_n, y_n)$ es el nodo o celda en expansión actual; $f(n)$ es el costo mínimo estimado de la celda entre todos los caminos desde el nodo de inicio S al nodo de meta G forzado a pasar por el nodo n; $g(n)$ es la función de costo real desde el nodo de inicio $S(x_S, y_S)$ hasta el nodo de expansión actual n; $h(n)$ es la estimación heurística del costo mínimo de una trayectoria desde el nodo de expansión actual n al nodo objetivo $G(x_G, y_G)$ [15].

La distancia d entre dos puntos viene dada por

$$d = \sqrt{(x_n - x_G)^2 + (y_n - y_G)^2} \quad (2)$$

y $h(n)$ en (1) puede estimarse como

$$h(n) = \sqrt{(x_n - x_G)^2 + (y_n - y_G)^2} \quad (3)$$

esta $h(n)$ es la distancia euclidiana entre el nodo objetivo y el nodo en expansión actual. Ahora la solución de menor costo se puede averiguar con la ayuda de las ecuaciones anteriores y retrocedidas desde la posición de meta G a la posición de inicio S [15].

El algoritmo A* tiene las siguientes características:

- Es uno de los algoritmos de planificación más usados, ya que de existir una solución, busca la ruta más corta entre dos nodos.
- Usa una función heurística entre dos nodos que sirve como referencia para evaluar el camino con el menor costo.
- La función de evaluación de la ruta viene dado por la sumatoria del costo del camino recorrido y el valor de la heurística en (1), para hallar la ruta óptima.
- Es óptimamente eficiente ya que expande sólo los nodos que tienen el menor valor en la función de evaluación, haciendo que su búsqueda sea más rápida.
- Evita caminos que no necesitan ser evaluados.
- Es considerado un algoritmo completo.

Algoritmo de Dijkstra

El algoritmo de Dijkstra fue presentado por el holandés Edsger Wybe Dijkstra en 1959, quien lo describió por el problema de construir un árbol de longitud total mínima entre "n" nodos y de encontrar la ruta de longitud total mínima entre dos nodos [16].

Es un algoritmo clásico para resolver el problema del camino más corto, es simple y fácil, un método excelente para la trayectoria del camino del robot y ha sido ampliamente utilizado en optimización de red, transporte, logística, electrónica y otros campos [17].

Correll [9] explica el algoritmo de Dijkstra de la siguiente manera: A partir del vértice inicial donde debe iniciarse la ruta, el algoritmo marca todos los vecinos directos del vértice inicial con el costo para llegar allí. Luego procede desde el vértice con el menor costo a todos sus vértices adyacentes y los marca con el costo de llegar a ellos por sí mismo; si este costo es menor. Después que se han comprobado todos los vecinos de un vértice, el algoritmo procede al vértice siguiente con el costo más bajo. Una vez que el algoritmo alcanza el vértice de la meta, termina y el robot puede seguir los bordes que apuntan hacia el costo del borde más bajo.

El algoritmo de Dijkstra tiene las siguientes características:

- Este algoritmo es considerado como uno de los más relevantes en la teoría de grafos.
- Es un algoritmo conocido para la búsqueda de ruta óptima entre dos nodos.
- Sirve para determinar el camino más corto dado un nodo origen al resto de nodos.

- Es eficiente pero puede llegar a desperdiciar tiempo analizando nodos que no son necesarios o que se encuentran lejos del nodo de llegada.
- No trabaja con costos de aristas negativos.
- Es considerado un algoritmo completo.

Características de la interfaz de simulación

La interfaz de simulación consiste en mostrar un mapa, que está representado por una cuadrícula de ocupación, la misma que aparece en la ventana de la interfaz después de escoger el tipo de algoritmo; la ventana presenta las opciones para ingresar la coordenada (X,Y) del punto de llegada del robot móvil, así como las coordenadas de los puntos que son ocupados por los obstáculos que debe evadir el robot.

Un botón es el que da inicio al funcionamiento del algoritmo así como también un botón se encarga de limpiar el mapa y otro para salir de la ventana de simulación.

El robot es representado por un punto sin dimensiones y se coloca en la mitad de los cuadros blancos del mapa, los mismos en los que el robot tiene la capacidad de moverse; la posición de inicio del robot móvil es por defecto en la parte inferior izquierda de la cuadrícula o mapa en la coordenada (1,1); los obstáculos son cuadrados negros que no pueden ser ocupados por el robot.

Desarrollo de la interfaz gráfica

La interfaz se realizó en Matlab, que es una herramienta muy utilizada para aplicaciones de ingeniería; nos da facilidad en el trabajo con matrices, en la representación tanto de datos como de funciones, permite la implementación de algoritmos y la creación de una interfaz gráfica de usuario.

A través de la herramienta GUIDE que es el entorno de desarrollo de GUI y los componentes que tiene, nos permitió diseñar gráficamente la interfaz de usuario, la cual generó un código de los elementos que incluye y el mismo que se modificó para programar los algoritmos de planeación de trayectorias del robot móvil.

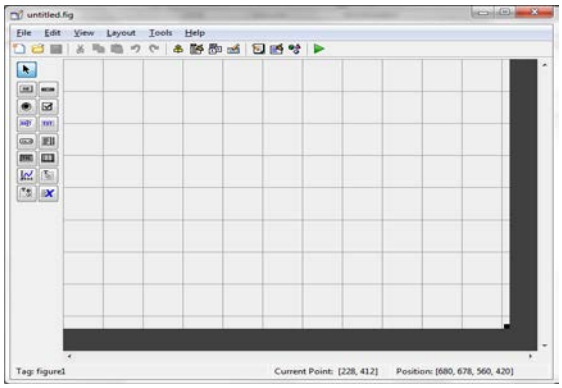

Figura 2. Entorno GUIDE

Ventana del simulador

La ventana del simulador está formada por cuadros de texto editable que permiten el ingreso de valores numéricos, botones para ejecutar una acción y un cuadro gráfico para la visualización del mapa. Presenta sus respectivas partes que son para ejecutar las funciones del simulador:

- Ingresar el tamaño del mapa y la creación del mismo,
- Ingresar la posición de llegada e inicio del robot,
- Ingresar las posiciones de los obstáculos,
- Ejecutar el algoritmo para visualizar la simulación de planeación de trayectoria,
- Limpiar el cuadro grafico para poder realizar una nueva simulación y
- Salir de la ventana del simulador del algoritmo.

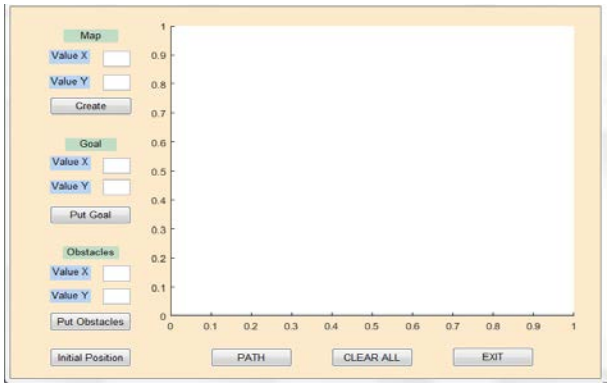

Figura 3. Ventana del simulador

*Diagrama de funcionamiento del simulador del algoritmo A**

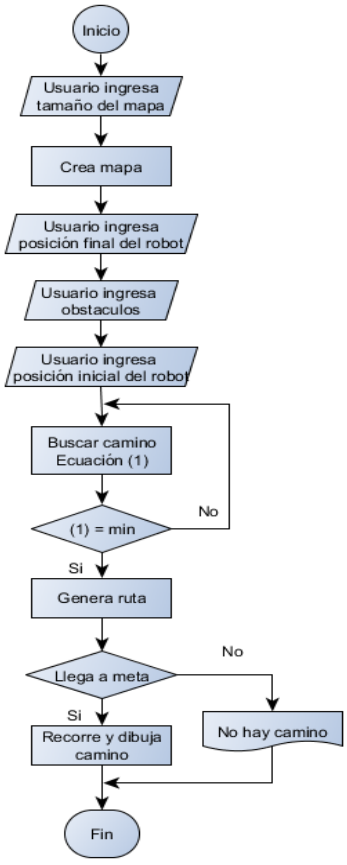

Figura 4. Diagrama de flujo del simulador del algoritmo A*

Diagrama de funcionamiento del simulador del algoritmo de Dijkstra

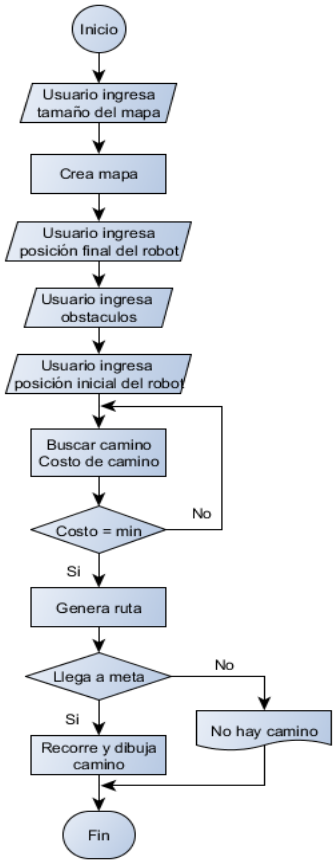

Figura 5. Diagrama de flujo del simulador del algoritmo de Dijkstra

III. RESULTADOS

*Simulación del Algoritmo A**

Se comprobó que en realidad encuentra el camino óptimo. En la figura 3, se puede observas que la ruta óptima tiene una distancia recorrida de 8.23.

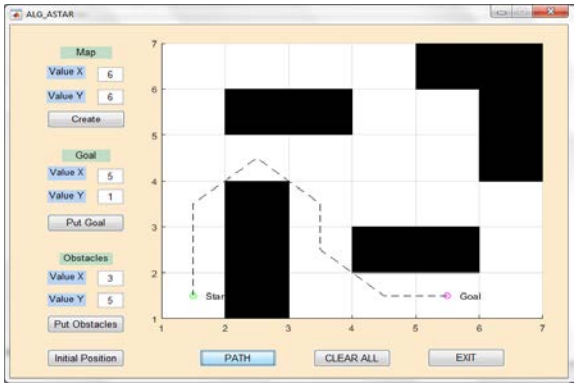

Figura 6. Simulación 1

En la figura 4, se puede observas que la ruta óptima tiene una distancia recorrida de 12.23.

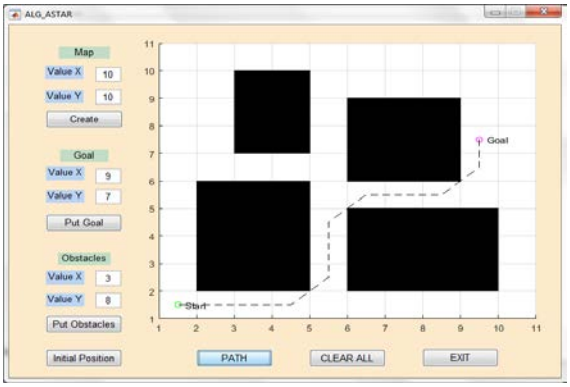

Figura 7. Simulación 2

Simulación del Algoritmo de Dijkstra

En la figura 5, se puede observas que la ruta óptima tiene una distancia recorrida de 5.82.

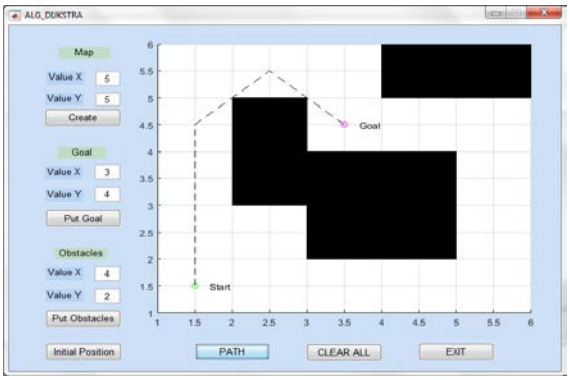

Figura 8. Simulación 3

En la figura 6, se puede observas que la ruta óptima tiene una distancia recorrida de 7.82.

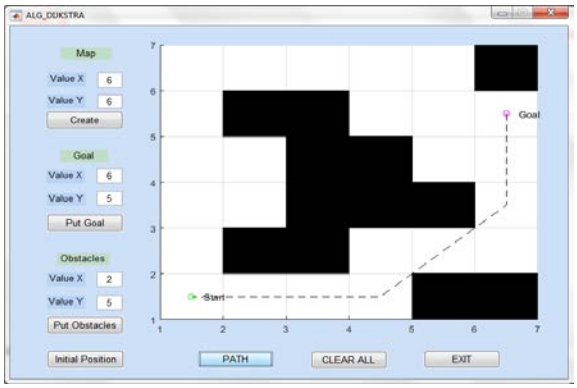

Figura 9. Simulación 4

Repetitividad

Al realizar la simulación tanto con el algoritmo A* y de Dijkstra en un mismo tipo de mapa por más de una ocasión, se puede observar que el simulador presenta repetitividad en las rutas calculadas y graficadas.

Tiempo de procesamiento de simulación

El tiempo de procesamiento se obtiene como resultado del programa Matlab. Se realiza simulaciones para cada uno de

los algoritmos con el propósito de saber cuál presenta mayor eficiencia en velocidad de análisis y ejecución de la simulación.

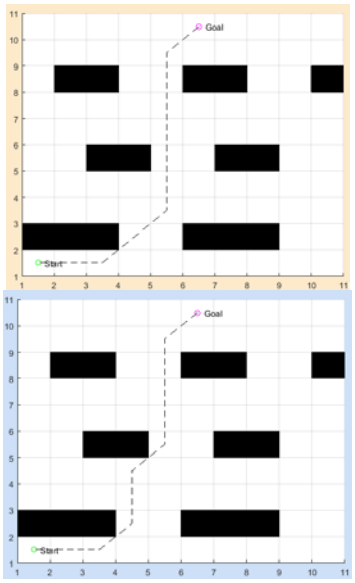

Figura 10. Mapa1

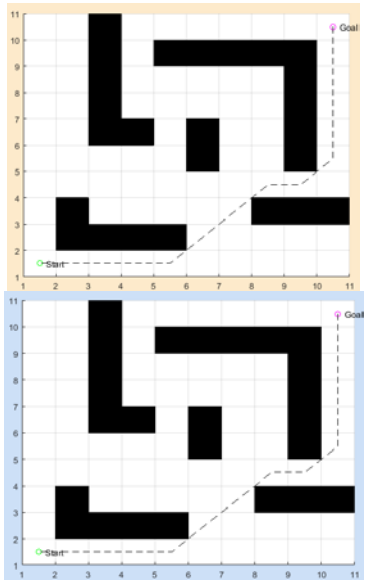

Figura 11. Mapa 2

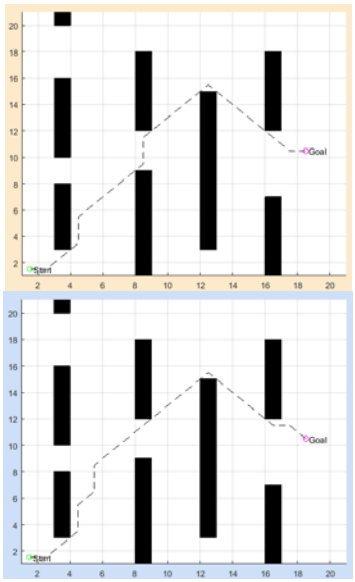

Figura 12. Mapa 3

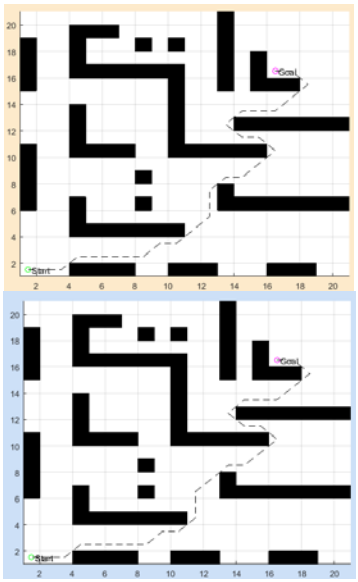

Figura 13. Mapa 4

Las simulaciones se realizaron en una computadora Intel Core i5-2450M CPU 2.50GHz, 8,0GB RAM; en la tabla 1 se observa las coordenadas de posición de inicio y llegada del robot, así como el tiempo que se demoró en ejecutar el análisis y simulación cada algoritmo.

Tabla1. Posición del robot y tiempo de procesamiento

Mapa	Posición de inicio	Posición de meta	A* (s)	Dijsktra (s)
1	Inicio=(1,1)	Meta=(6,10)	3.3934	3.6447
2	Inicio=(1,1)	Meta=(10,10)	4.0627	4.3390
3	Inicio=(1,1)	Meta=(18,10)	6.5727	6.6351
4	Inicio=(1,1)	Meta=(16,16)	8.6146	8.7523

Como se puede observar el tiempo de procesamiento de la simulación varía según el tamaño del mapa y de la cantidad de obstáculos que este presenta. Hay que tomar en cuenta que los tiempos de cálculo de respuesta obtenidos pueden variar,

dependiendo de la velocidad en la que se esté ejecutando Matlab en la computadora.

Tabla 2. Diferencia de velocidad en segundos y porcentual

Mapa	Diferencia de velocidad (s)	Diferencia de velocidad (%)
1	0.2513	6.89%
2	0.2763	6.36%
3	0.0624	0.94%
4	0.1377	1.57%

Al analizar los valores de tiempo obtenidos para el algoritmo A* y de Dijkstra, se puede ver que es más rápido el algoritmo A* en un promedio estimado entre los cuatro mapas del 3.94%, ya que se guía con el valor de la heurística y analiza los nodos que presentan una función de costo menor.

Hay que tomar en cuenta el resultado de la simulación del mapa 2 en las figura 11, que presentan la misma trayectoria para los dos algoritmos.

IV. Conclusiones

- La planeación de trayectorias representa un estudio muy importante para la navegación autónoma de robots móviles por la necesidad de obtener una ruta que se encuentre libre de obstáculos y que sea óptima para el robot.
- Con el análisis de la información obtenida en nuestra bibliografía se determinó que para el desarrollo del proyecto se necesita algoritmos de búsqueda de gráficos y árboles.
- Uno de los algoritmos de búsqueda más utilizados en investigaciones de aplicación de planificación de caminos es el algoritmo A*.
- Tanto el algoritmo A* como el de Dijkstra son considerados completos por su eficiencia en encontrar el camino de menor costo; lo que diferencia al algoritmo A* del de Dijkstra es la función heurística que usa como referencia de distancia para buscar la ruta al puto de llegada del robot móvil.
- Mediante el uso de Matlab se pudo desarrollar la interfaz de simulación por la herramienta GUIDE que presenta y por su facilidad de trabajar con matrices, que fueron necesarias para la programación de los algoritmos de planificación.
- Después de investigar cómo se desarrolla el funcionamiento de cada algoritmo aplicado a un mapa de rejilla y de su comprobación con el software de simulación, se puede observar que el simulador de los algoritmos funcionan de manera correcta dando como resultado una trayectoria óptima; presentando el 100% de fiabilidad en las pruebas de repetitividad.
- Dependiendo del entorno o mapa, el algoritmo A* realiza el análisis de búsqueda mucho más rápido que el algoritmo de Dijkstra, superándolo en un promedio estimado del 3.94%; estos algoritmos incluso pueden llegar a determinar la misma ruta si fuera el caso.

V. Recomendaciones

- Desarrollar mejoras en la interfaz de simulación y en la programación de los algoritmos para aumentar la eficiencia de los mismos y facilitar su uso.
- Realizar un análisis como lo propone Goyal y Nagla [15], en el que consideran una dimensión para el robot.
- Realizar investigación en otros tipos de algoritmos para implementarlos en la interfaz del simulador.
- Continuar desarrollando el proyecto, para que los programas de los algoritmos de planificación puedan ser aplicados en la plataforma robótica móvil que se deja construida y así comprobar el funcionamiento en la misma.

REFERENCIAS

[1] L. A. Arellano Zea, *Diseño e implementación de un robot móvil con control de trayectoria mediante principios odométricos*, Lima, 2015.

[2] F. Benavides, *Planificación de movimientos aplicada en robótica autónoma móvil*, Montevideo, 2012.

[3] V. Zambrano, *Implementacíon de Algoritmos de Determinacion de Rutas para el Robotino® de Festo*, Quito, 2015.

[4] A. Yandún, *Planeación y Seguimiento de Trayectorias para un Robot Móvil*, Quito, 2011.

[5] M. Fernández, D. Fernández y C. Valmaseda, *Planificacion de trayectorias para un Robot Móvil*, Madrid, 2009.

[6] P. Corke, Robotics, Vision and Control, Springer, 2011.

[7] L. Álvarez y J. Figueroa, *Implementación de algoritmos de navegación utilizando la plataforma iRobot Create y módulos de comunicación inalambrica Xbee*, 2011.

[8] F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico y L. Jurišica, «Path Planning with Modified A Star Algorithm for a Mobile Robot,» *ScienceDirect*, vol. 96, pp. 59-69, 2014

[9] N. Correll, Introduction to Autonomous Robots, 1 ed., 2016.

[10] H. E. Espitia Cuchango y J. I. Sofrony Esmeral, «Algoritmo para Planear Trayectorias de Robots Móviles, Empleando Campos Potenciales y Enjambres de Partículas Activas Brownianas,» *Ciencia e Ingeniería Neogranadina*, vol. 22, nº 2, pp. 75-96, Diciembre 2012.

[11] R. Martínez Ángel, J. Barrero Pérez y D. A. Tibaduiza Burgos, «Algoritmos de Planificación de Trayectorias para un Robot Móvil,» *ResearchGate*, Agosto 2006.

[12] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki y S. Thrun, Principles of Robot Motion: Theory, Algorithms, and Implementation, Cambridge: MIT Press, 2007.

[13] S. M. LaValle, Planning Algorithms, Cambridge, 2006

[14] P. E. Hart, N. J. Nilsson y B. Raphael, «A Formal Basis for the Heuristic Determination of Minimum Cost Paths.,» *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, pp. 100-107, 1968

[15] J. K. Goyal y K. Nagla, «A New Approach of Path Planning for Mobile Robots,» *International Conference on Advances in Computing, Communications and Informatics (ICACCI),* 2014.
[16] E. W. Dijkstra, «A Note on Two Problems in Connexion with Graphs,» *Numerische Mathemathik,* vol. 1, pp. 269-271, 1959.
[17] Z. Zhang y Z. Zhao, «A Multiple Mobile Robots Path planning Algorithm Based on A-star and Dijkstra Algorithm,» *International Journal of Smart Home,* vol. 8, nº 3, pp. 75-86, 2014.
[18] N. O. Eraghi, F. López-Colino, A. de Castro y J. Garrido, «Path Length Comparison in Grid Maps of Planning Algorithms: HCTNav, A* and Dijkstra,» *Design of Circuits and Integrated Systems,* pp. 1-6, 2014.
[19] A. Ollero Baturone, Robótica Manipuladores y robots móviles, España: Marcombo, 2001.
[20] R. Siegwart y I. R. Nourbakhsh, Introduction to Autonomous Mobile Robots, London, 2004
[21] T. Bräunl, Embedded Robotics Mobile Robot Design and Aplications with Embedded Systems, Perth: Springer, 2006.
[22] D. O. Barragán Guerrero, «Repositorio de ESPOL,» 25 Mayo 2008. [En línea]. Available: https://www.dspace.espol.edu.ec/bitstream/123456789/10740/11/MATLAB_GUIDE.pdf.
[23] MathWorks, Inc, «Matlab Primer,» Marzo 2017. [En línea]. Available: http://www.mathworks.com/help/index.html.
[24] A. M. Cueva, *Generación global de trayectorias para robots móviles, basada en curvas betaspline,* Sevilla, 2014.
[25] W. Shu-xi y Z. Xing-qui, «The Improved Dijkstra's Shortest Path Algorithm,» *International Conference on Natural Computation,* pp. 2313-2316, 2011.