

Implementación de Tecnologías basadas en Inteligencia Artificial para el análisis de la conducta de robots móviles autónomos en espacios de configuración en colisión

Alejandro Hossian¹, Maximiliano Alveal¹

⁽¹⁾Universidad Tecnológica Nacional - Facultad Regional Neuquén

Grupo de Investigación en Robótica Industrial y Cognitiva

Bº Campamento Uno – Plaza Huincul (C.P. 8316) – Neuquén – Argentina

Teléfono (0299) 4960510

alejandrohossian@yahoo.com.ar, maximilianoalveal@hotmail.com

RESUMEN

La tecnología de la Inteligencia Artificial (IA) concatena una diversa gama de tecnologías transversales a diferentes campos disciplinares. Este es el caso de la navegación de robots móviles autónomos cuando deben realizar tareas de diferente complejidad en un ambiente estructurado.

Esta producción centra su trabajo en la conducta que registran los robots móviles en ambientes de navegación en espacios de configuración en colisión, en los cuales las estructuras permanecen fijas mientras el robot realiza las operaciones requeridas por el usuario. Los primeros experimentos tuvieron como soporte la aplicación de las Tecnologías Inteligentes de las Redes Neuronales Artificiales (RNA), los cuales se entraron en la implementación del algoritmo de aprendizaje supervisado de retropropagación del error (backpropagation). Actualmente, y con el propósito de mejorar la performance del robót movil con la tecnología aplicada, se realizan experimentos mediante la aplicación de algoritmos inteligentes de paradigma deliberativo, los cuales se orientan hacia la planificación de las tareas que el robot debe realizar dentro de su entorno de operación.

Palabras claves: Artificial Intelligence (AI), Robotic Vehicle, Artificial Neural Networks (ANR), intelligent Algorithms.

1. INTRODUCCIÓN

Las investigaciones en el campo de la robótica se realizaban en base a ambientes de desarrollo con células de trabajo fijas en sus posiciones para que el robot desarrolle sus tareas, tal es el caso del robot ensamblador o soldador en una planta de montaje.

Tan significativos fueron los avances tecnológicos en los diferentes ambientes industriales, que constituyeron una de las razones principales para dotar a los sistemas de robots de una adecuada capacidad de desplazamiento, más allá de las prestaciones que realizaban en sus células de trabajo. Es entonces cuando esta disciplina dio en llamarse “robótica móvil” y constituye uno de los mayores

desafíos que aborda la comunidad científica que trabaja dentro del rico y extenso campo de la robótica.

Cabe señalar, que las sucesivas investigaciones desarrolladas intentaron proporcionar a estos sistemas móviles de un nivel de autonomía suficiente, que le permitan navegar en su ambiente de operación y reaccionar ante situaciones que no han sido consideradas en su programa de control [1].

La navegación permite guiar el curso de un robot móvil mediante un entorno en el que se encuentran obstáculos. Se conocen diferentes esquemas para llevar a cabo esta tarea, pero todos ellos tienen el objetivo común de dirigir el vehículo hacia su destino de la manera más segura y eficiente posible. La capacidad de reacción que pueda poseer el robot cuando se encuentra ante situaciones inesperadas, debe constituir su cualidad más distintiva para desenvolverse eficazmente en el entorno donde éste deba operar, lo cual indica el grado de autonomía que este posee.

Por todo lo expuesto, en la sección 1.1 se presentan los principios fundamentales de las RNA; como así también los experimentos realizados en el campo de la Robótica Móvil, mediante la aplicación de aprendizaje supervisado del tipo Backpropagation. En la sección 1.2 se hace referencia a las características generales de los Algoritmos de Búsqueda, en particular las correspondientes a las Técnicas no informadas o a ciegas.

1.1 Aplicación de Redes Neuronales

En la primera etapa de este trabajo se presentan los resultados obtenidos mediante la aplicación de técnicas de navegación robótica basadas en redes neuronales con aprendizaje supervisado de tipo Backpropagation, representativo de las características más potentes del paradigma reactivo concerniente a la navegación autónoma de robots [2], [3]. Este trabajo se realiza en un marco un poco más amplio considerado una evolución del paradigma reactivo, que se denomina "Aproximación Basada en Comportamientos" [4]. Esto es, los comportamientos describen la forma en que se reacciona (en este caso un sistema robótico) ante un determinado estado de sensores y donde las actuaciones se determinan de manera más elaborada que una simple consulta a una tabla de correspondencias, realizando cálculos a los efectos de decidir qué acción se debe llevar a cabo.

1.2 Principios y fundamentos de los Algoritmos de búsqueda

En la segunda etapa de este trabajo se hace uso de los Algoritmos de Búsqueda, los que constituyen una de las tecnologías más importantes de la IA, cuyo objetivo principal consiste en hallar una solución válida dentro del espacio de estados. Entre otras técnicas que se pueden implementar con estos algoritmos se conocen las de búsqueda no informada y búsqueda con información. La primera debe su denominación a que el problema que se pretende resolver no proporciona ninguna información adicional que sirva de soporte para encontrar una solución de forma más rápida. En tanto que la segunda técnica sí proporciona información que ayuda a guiar la búsqueda.

En este trabajo se utilizan técnicas de búsqueda no informada, o también conocida como búsqueda a ciegas. En primer lugar se muestra un experimento relacionado con la Técnica de búsqueda en profundidad, cuyo objetivo central es encontrar una ruta entre el nodo raíz y el nodo objetivo; en segundo término se hace uso de la técnica de búsqueda en amplitud, la cual además de hallar una ruta entre el nodo raíz y el nodo objetivo, encuentra la más corta. Cabe destacar a tal efecto, que tanto el entorno estructurado como el nodo raíz y destino son los mismos a la hora de implementar ambas técnicas, a fin de poder medir la performance de cada uno de ellos.

2. EL MODELO PROPUESTO

El modelo propuesto se desarrolla en tres secciones. En la sección 2.1 se menciona brevemente el abordaje de la navegación robótica con aprendizaje supervisado mediante la aplicación de RNA de tipo Backpropagation. En la sección 2.2, se desarrolla el modelo de navegación robótica con algoritmos de búsqueda, en la subsección 2.2.1 se aborda el algoritmo de búsqueda en profundidad y en la subsección 2.2.2 se analiza el mismo problema propuesto en la subsección anterior esta vez mediante la aplicación de algoritmos de búsqueda en amplitud.

2.1 Aplicación de una Red de tipo Backpropagation

A efectos de estudiar la conducta de un robot, se propone como arquitectura una red neuronal que consta de una capa de entrada, una capa oculta y una capa de salida [5]. De esta manera, se intenta que el robot pueda realizar trayectorias de mayor complejidad que para las que fue entrenado. Durante el proceso de entrenamiento, el vehículo robótico recibe información del ambiente en el cual navega; la cual es volcada en un mapa sensor – motor, que permite esquematizar tal ambiente con obstáculos definidos, y a través de un software diseñado a tal efecto, se simula su interacción luego de su entrenamiento para evaluar su performance bajo las premisas del paradigma reactivo. Este mapa sensor-motor se traduce en una matriz con la descripción sensorial en cada posición de la grilla, que representa el entorno y los posibles movimientos en respuesta a los estímulos sentidos, conforme a las políticas de proximidad y de movimiento [6].

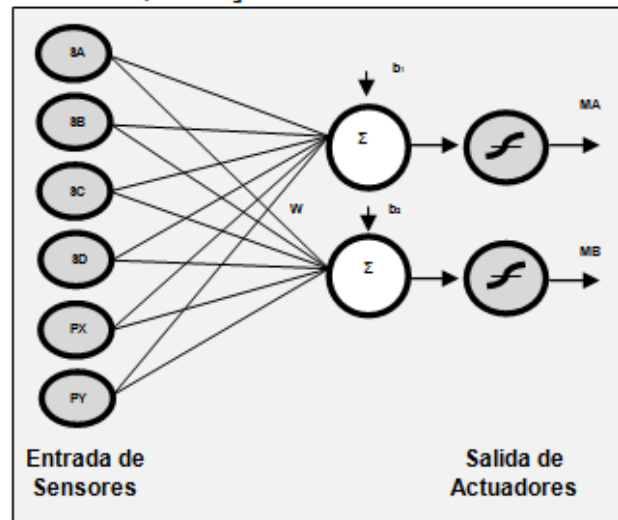


Fig. 1. Arquitectura de red de tipo Backpropagation

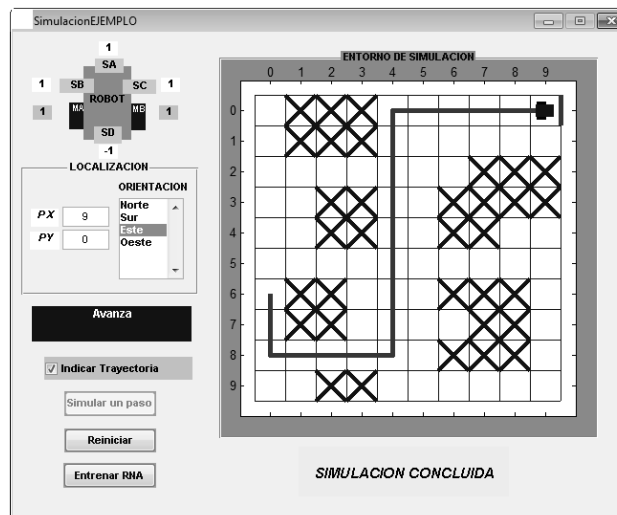


Fig. 2. Trayectoria exitosa de un robot entrenado con RNA.

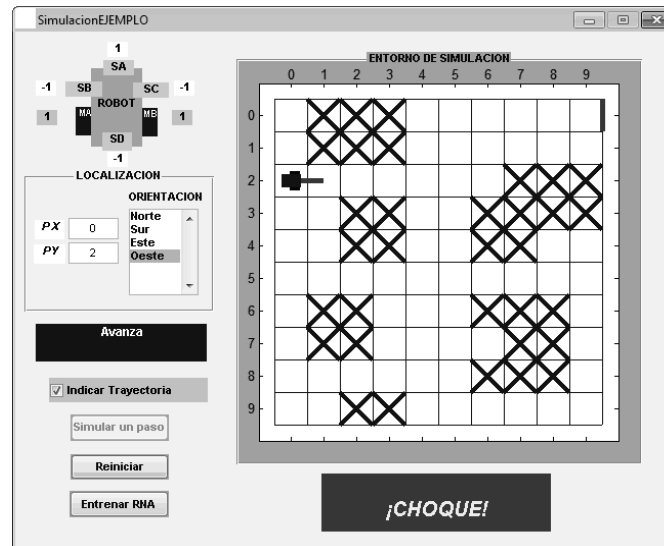


Fig. 3. Trayectoria no exitosa partiendo desde un lugar distinto al aprendido.

Considerando el mapa sensor-motor y tras el entrenamiento del robot con la tecnología inteligente de la red neuronal escogida se evalúa el robot para diferentes ambientes de operación. La combinación de rutas definidas para distintas configuraciones del ambiente de navegación, constituyen las diversas conductas de navegación que exhibe el robot [7].

El paradigma de red utilizado se describe en la figura 1, mientras que las figura 2 y 3 muestran el esquema de la organización del ambiente donde navega el robot, consistente en una grilla cuyas coordenadas (x, y) se encuadran dentro de un eje cartesiano con dos trayectorias diferentes. Este entorno visual brinda la información referida a la localización del robot (proporcionada por las coordenadas X e Y, que en este caso corresponden a 0 según X y a 6 según Y), su orientación (dada por Norte, Sur, Este y Oeste), el ambiente propiamente dicho, con la ubicación de sus obstáculos (casilleros con cruces) y las opciones para entrenar la red, reiniciar el proceso cuando sea necesario y simular los pasos de la trayectoria seguida por el robot. Asimismo, estas últimas dos figuras muestran tanto una trayectoria exitosa y una colisión para el mismo entorno partiendo desde otro origen [8].

En fase de operación, se procede a presentarle al robot otras trayectorias diferentes a las aprendidas para evaluar su performance.

2.2 Modelo de aplicación de Algoritmo de Búsqueda

El uso de grafos de navegación permite trabajar con una abstracción del entorno real y son estos usados en la navegación robótica cuando se conocen previamente las características del ambiente.

El entorno se expresa en una grilla donde los casilleros oscuros representan los obstáculos y los casilleros libres indican el espacio transitable, tal como se observa en la figura 4. En estos espacios libres se colocan nodos o vértices nombrados con letras mayúsculas, como se muestra en la figura 5, para luego trazar las aristas posibles que los conectan, figura 6.

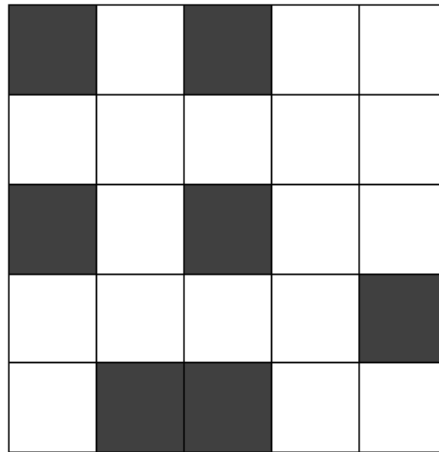


Fig. 4. Entorno propuesto.

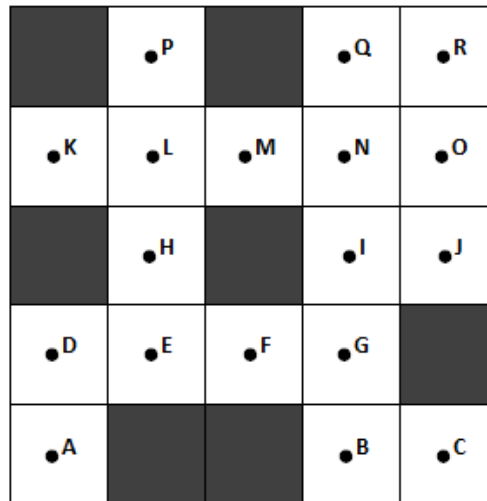


Fig. 5. Distribución de los nodos.

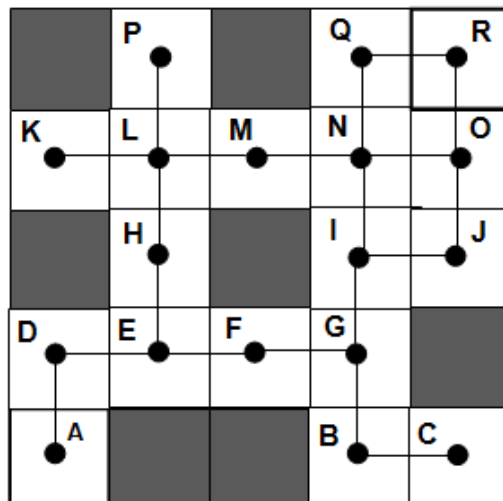


Fig. 6. Diseño del grafo en el entorno.

En este caso de estudio se utiliza un grafo que cuenta con dieciocho vértices o nodos y veinte aristas. Se conoce como grafo de navegación o navgraph y permite trabajar con una abstracción de robots cuando se conoce previamente las características del entorno [9].

Con la definición de estos elementos, se puede entonces comenzar con la presentación de los modelos de algoritmos propuestos.

Es importante señalar que estos algoritmos trabajan con cierta información. Es decir, el algoritmo aplicado conocerá cuál es el nodo objetivo al que se desea llegar y también puede guardar información sobre cuáles nodos y aristas se han visitado previamente.

2.2.1 Modelo de aplicación de Algoritmo de Búsqueda en profundidad

El algoritmo Depth First Search (DFS) o algoritmo de búsqueda en profundidad, es el primer algoritmo que se analiza. La búsqueda que realiza este algoritmo consiste en el descenso hasta el máximo nivel de profundidad de la rama llegando hasta al nodo más profundo y luego continuar con la siguiente rama. Este algoritmo se inicia en un nodo del grafo, considerado nodo raíz. A partir de ese punto, observa cuáles son los nodos hijo y visita al primero. De este nuevo nodo, ve cuáles son los hijos y visita al primero que encuentra. El proceso sigue hasta hallar a una hoja, es decir, un nodo que no tenga hijos o a uno que esté conectado a nodos que ya se hayan visitado previamente. En ese momento, regresa al nodo padre inmediato y visita al siguiente hijo del nodo. El proceso continúa hasta que se encuentra el nodo objetivo o se han visitado todos los nodos del grafo. Esto puede observarse detenidamente en el diagrama de flujo que se despliega en la figura 7.

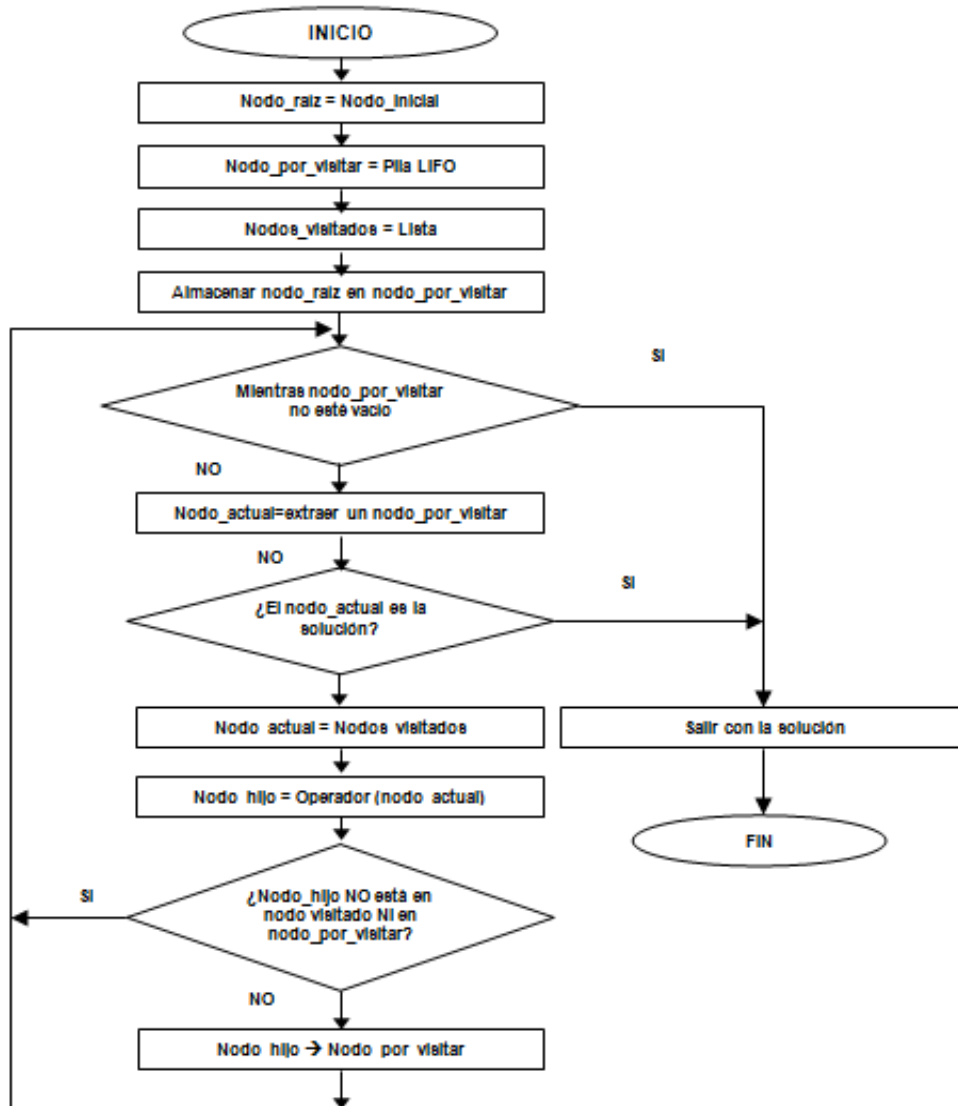


Fig. 7. Diseño del grafo en el entorno.

Este algoritmo expresado en el pseudocódigo descrito a continuación garantiza encontrar una ruta entre el nodo raíz y el objetivo, siempre y cuando el nodo objetivo exista y esté conectado al grafo.

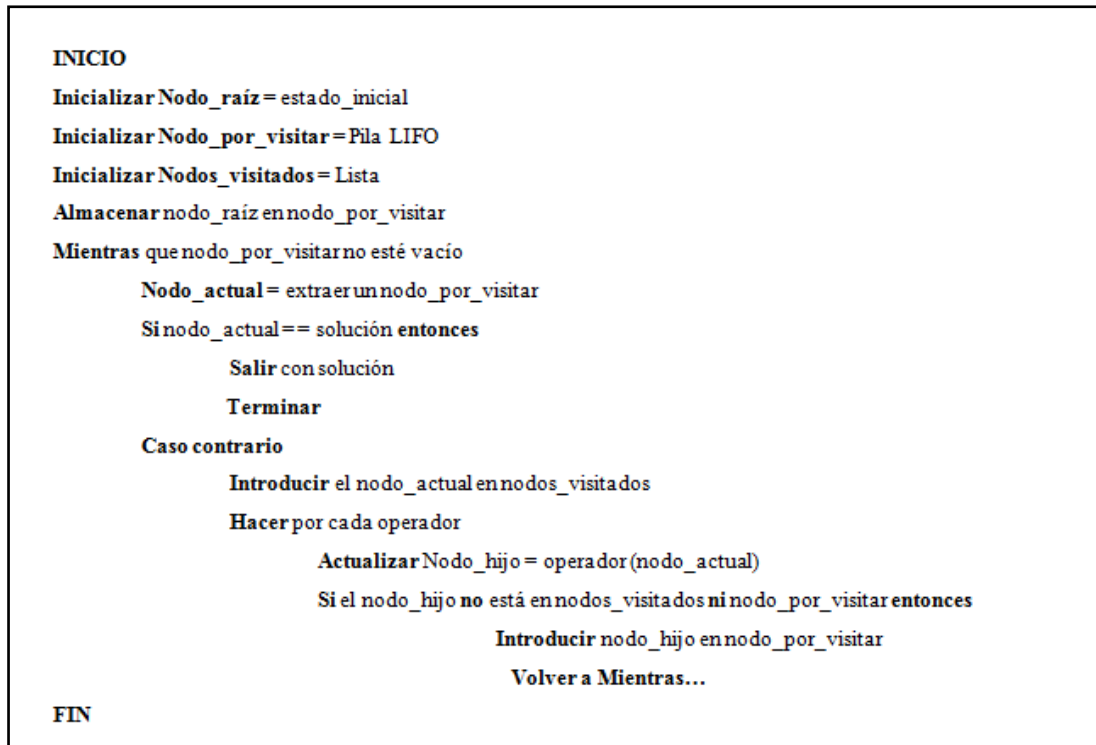


Fig. 8. Pseudocódigo del Algoritmo de Búsqueda en Profundidad.

Fig. 9.

Desde el punto de vista de la implementación, la búsqueda en profundidad utiliza una pila LIFO (Last In First Out). Una pila LIFO tiene dos operaciones: una para introducir un dato en la pila y otra para extraer un dato de la pila. El dato que se extrae de la pila es el último que se introdujo.

Por todo lo expuesto precedentemente, el ejemplo propuesto tiene como objetivo que el robot parta del nodo A y llegue al nodo R, es decir que encuentre una ruta que pueda seguir. Para ello es necesario contar con una "pila o stack" que sea capaz de guardar las aristas que faltan por visitar y además, un arreglo en el que se registren los nodos y su nodo padre. Es importante al iniciar este algoritmo, hacer una arista del nodo raíz a sí mismo para poder iniciar y colocarlo en la pila. Esto se ilustra en la figura 9:

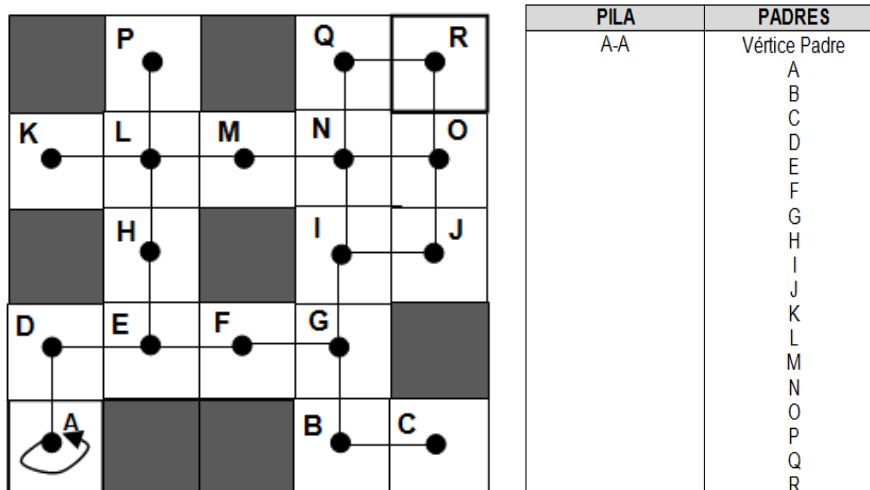
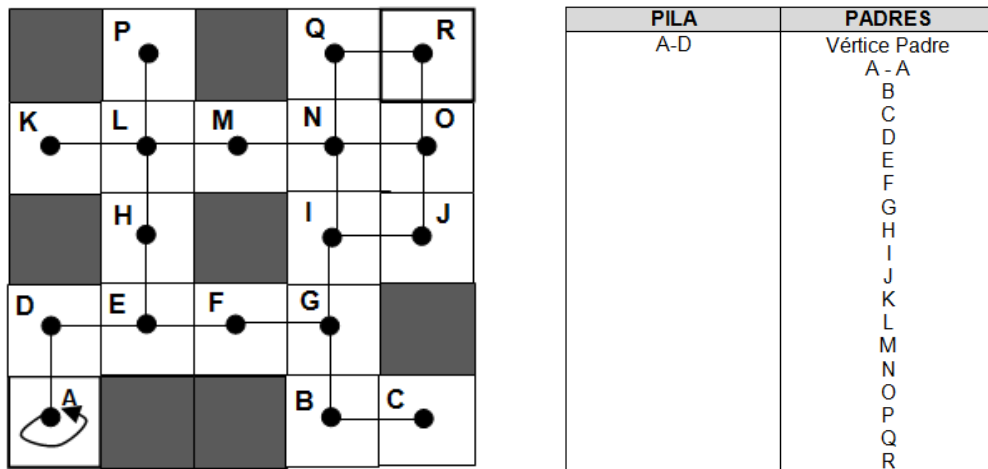


Fig. 10. Grafo propuesto con un autolazo en el nodo A a fin de dar inicio al algoritmo.

Se toma la primera arista que se encuentra en pila. Este lleva al nodo con el que se trabajará. Apenas se llega a este nuevo nodo se registra quiénes son sus nodos padres, marcando el nodo actual como ya visitado.

Luego se verifica si el nodo actual es el objetivo. Si lo es, finaliza el algoritmo, de lo contrario se continúa y se coloca en el registro de pila las aristas que se inician a partir del nodo actual. Si alguna arista ya estuviera previamente visitada, no se coloca. Esto se observa en la figura anterior y se registra nuevamente como sigue en la figura 10:



Grafo propuesto con el registro de las aristas adyacentes al nodo A.

En el siguiente paso se toma la primera arista de la pila y de esta manera el robot se posiciona en el nodo D. Se verifica que este nodo no sea el nodo destino. Su padre es el nodo A y se continúa con los pasos del algoritmo de la manera que se ilustra en la figura 11:

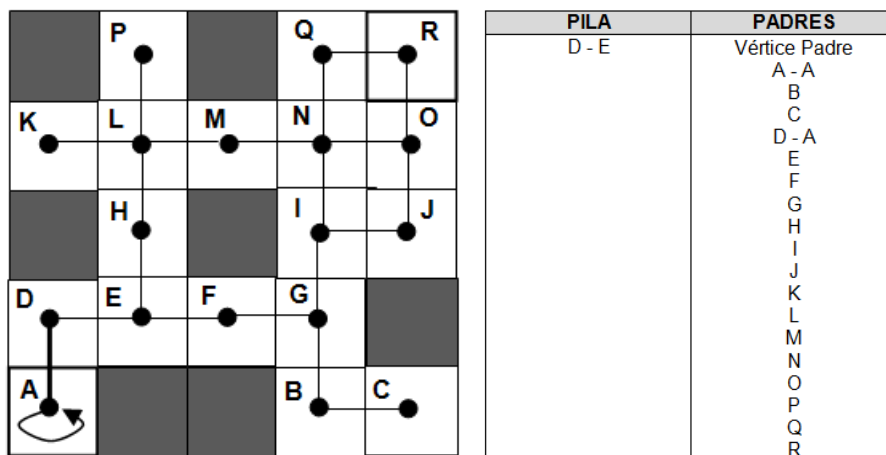
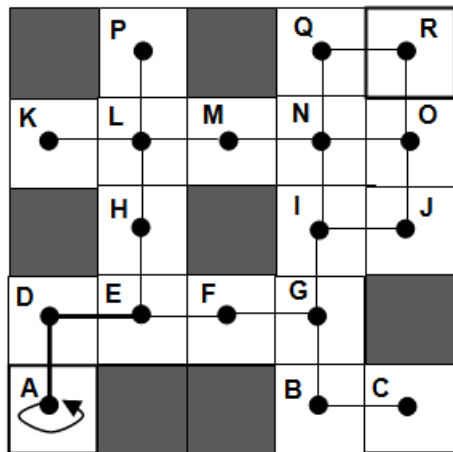


Fig. 11. El robot se posiciona en el nodo D. Se registra en la tabla a su nodo padre que es A.

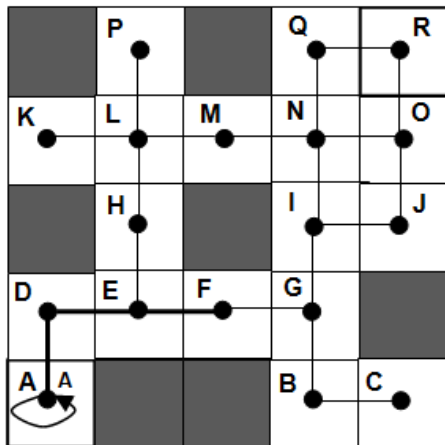
Entonces, se toma la primera arista de la pila y de esta manera el robot se posiciona en el nodo E. Su padre es el nodo D y se continúa con los pasos del algoritmo de la manera que se ilustra en la figura 12. Se observa que desde E se desprenden dos aristas: E-H y E-F.



PILA	PADRES
E - F	Vértice Padre
E - H	A - A
	B
	C
	D - A
	E - D
	F
	G
	H
	I
	J
	K
	L
	M
	N
	O
	P
	Q

Fig. 12. El robot se posiciona en el nodo E. Se registra en la tabla a su nodo padre que es D.

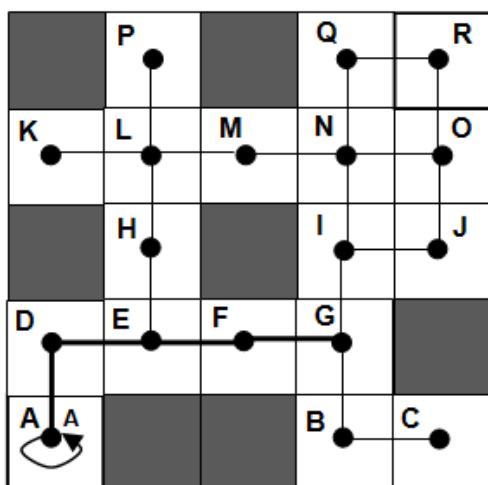
Se escoge recorrer la arista que se encuentra primero en la pila, es decir E-F y de esta forma el robot puede posicionarse en el nodo F. Ahora bien, el vértice E es el padre de F y la arista posible de ser recorrida es la arista F-G. Observe la figura 13.



PILA	PADRES
F - G	Vértice Padre
E - H	A - A
	B
	C
	D - A
	E - D
	F - E
	G
	H
	I
	J
	K
	L
	M
	N
	O
	P
	Q
	R

Fig. 13. El robot se posiciona en el nodo F. Se registra en la tabla a su nodo padre que es E.

Siguiendo esta línea, el vehículo robótico llega al nodo G. Se actualiza nuevamente la pila y el registro de los nodos padres. Tal como se observa en la figura 14:



PILA	PADRES
G - I	Vértice Padre
G - B	A - A
E - H	B
	C
	D - A
	E - D
	F - E
	G - F
	H
	I
	J
	K
	L
	M
	N
	O
	P
	Q
	R

Fig. 14. El robot se posiciona en el nodo G. Se registra en la tabla a su nodo padre que es F.

Se escoge la arista G-I, que es la primera en la pila y se guarda en ella las aristas que aún quedan pendientes por recorrer. Como en el paso anterior se registran los nodos padres. De esta manera el robot continúa hacia el nodo I.

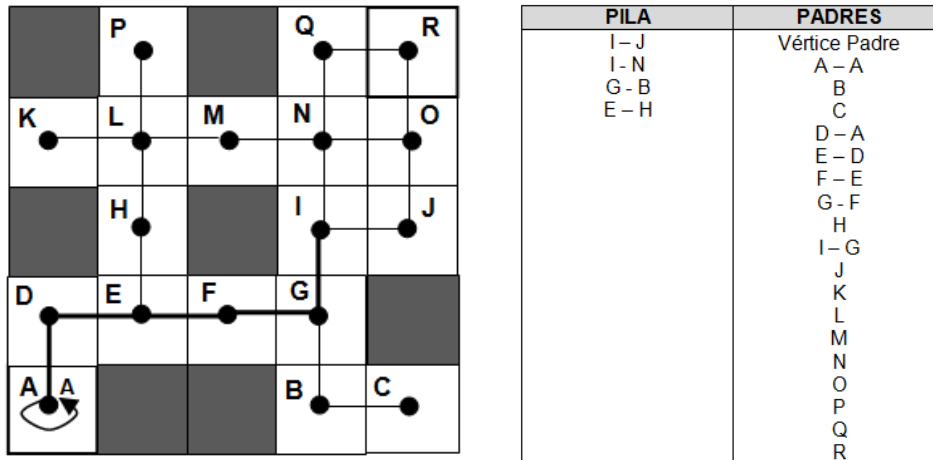


Fig. 15. El robot se posiciona en el nodo I. Se registra en la tabla a su nodo padre que es J.

Desde el vértice I, y con el objeto de ilustrar esta aplicación, se recorre hasta el vértice J. Se actualizan entonces los registros tal como se observa a continuación:

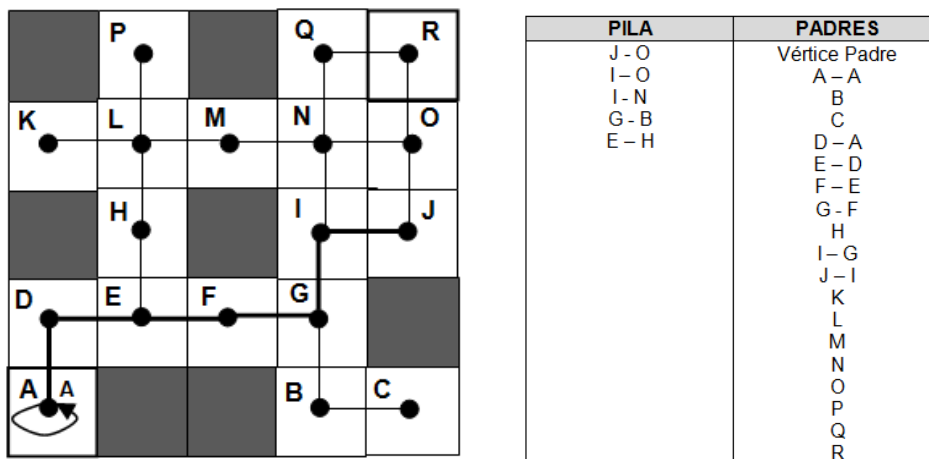


Fig. 16. El robot se posiciona en el nodo J. Se registra en la tabla a su nodo padre que es I.

Conforme se ha actualizado la pila, el robot se dirige hasta el vértice O, siendo J su nodo padre. Observe la figura 17.

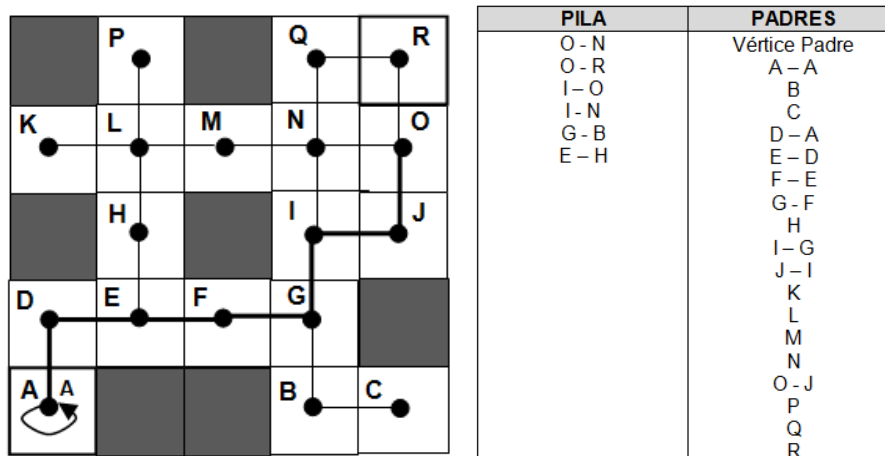


Fig. 17. El robot se posiciona en el nodo O. Se registra en la tabla a su nodo padre que es J.

Una vez actualizada la pila, el robot se dirige hasta el vértice N, siendo O su nodo padre, como se ilustra en la figura 18.

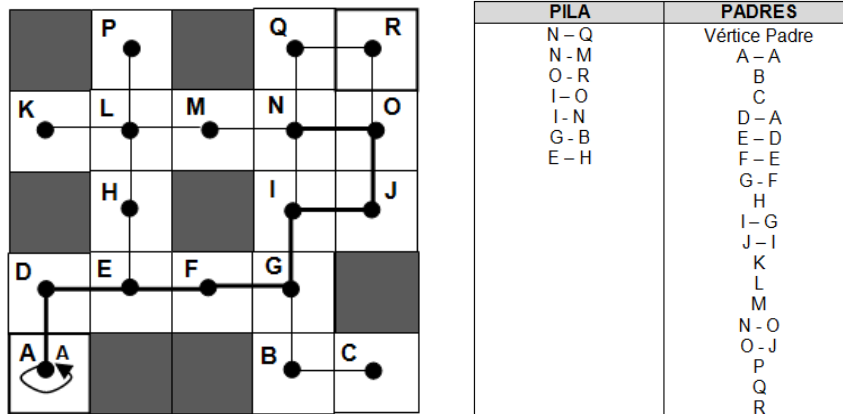


Fig. 18. El robot se posiciona en el nodo N. Se registra en la tabla a su nodo padre que es O.

Nuevamente se actualiza la pila y el robot se dirige hasta el vértice Q, siendo N su nodo padre, como se observa en la figura 19.

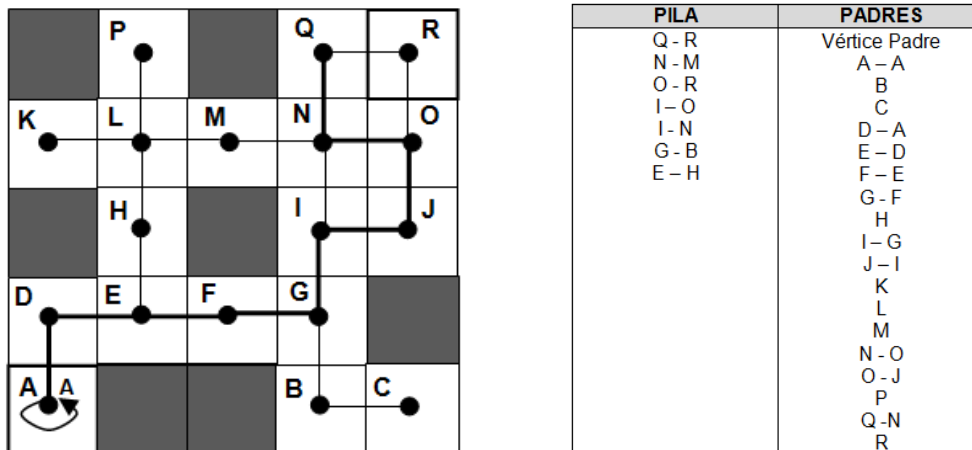


Fig. 19. El robot se posiciona en el nodo Q. Se registra en la tabla a su nodo padre que es N.

Conforme se ha actualizado la pila, el robot se dirige hasta el vértice R, siendo Q su nodo padre. Aquí se verifica que R es el nodo destino. En este caso concluye el algoritmo pues R es el punto hasta donde el robot debía llegar. De esta manera, se ha encontrado una ruta posible: A – D – E – F – G – I – J – O – N – Q – R, tal como se muestra en la figura 20.

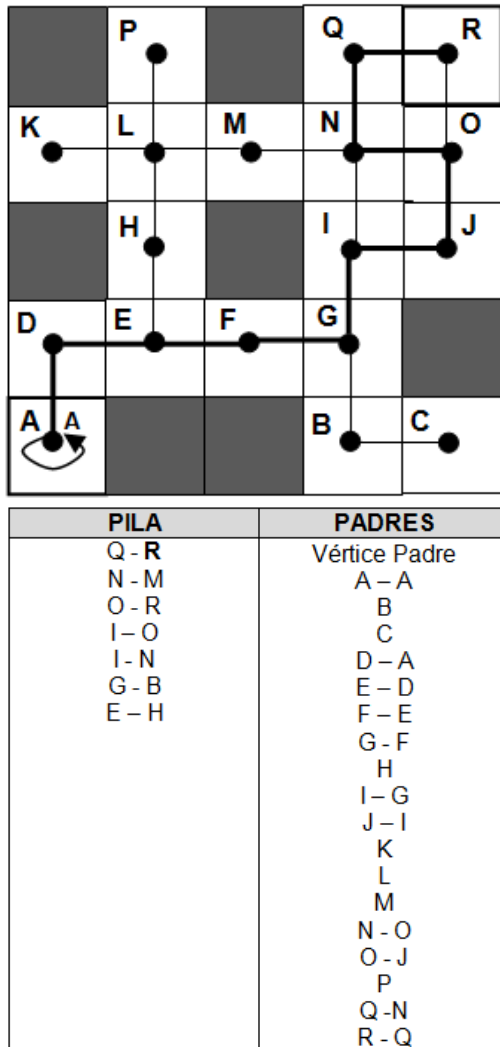


Fig. 20. El robot se posiciona en el nodo destino R.

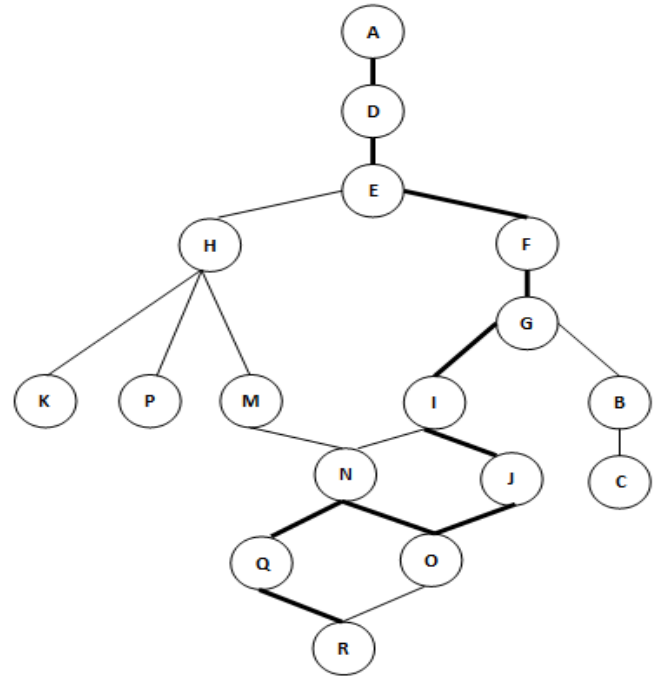


Fig. 21. Diseño del grafo solución.

Este algoritmo ha permitido encontrar una ruta, lo que no significa que sea la más corta. Esto queda representado, además, en el grafo de la figura 21.

2.2.2 Modelo de aplicación de Algoritmo de Búsqueda en Amplitud

Continuando con esta línea de investigación, se aborda el Breadth First Search o también llamado Algoritmo de Búsqueda en Amplitud. Uno de los objetivos fundamentales de este algoritmo de búsqueda consiste en la optimización de las vías que comunican los puntos de paso obligado que deben ser visitados por el vehículo robótico autónomo, redundando ello en un importante ahorro de recursos, tales como tiempo, combustible, desgaste de sus partes constitutivas, entre otros.

En especial, este algoritmo recorre el grafo por niveles. Es decir, primero se visita el nodo raíz. Seguidamente se visitan todos sus hijos y para cada hijo en el paso anterior se visitan todos sus hijos y así sucesivamente. Este algoritmo verifica los nodos del siguiente nivel antes de ir a él y hasta que no finaliza un nivel de profundidad no pasa la siguiente. Esto puede observarse detenidamente en el diagrama de flujo que se despliega a continuación.

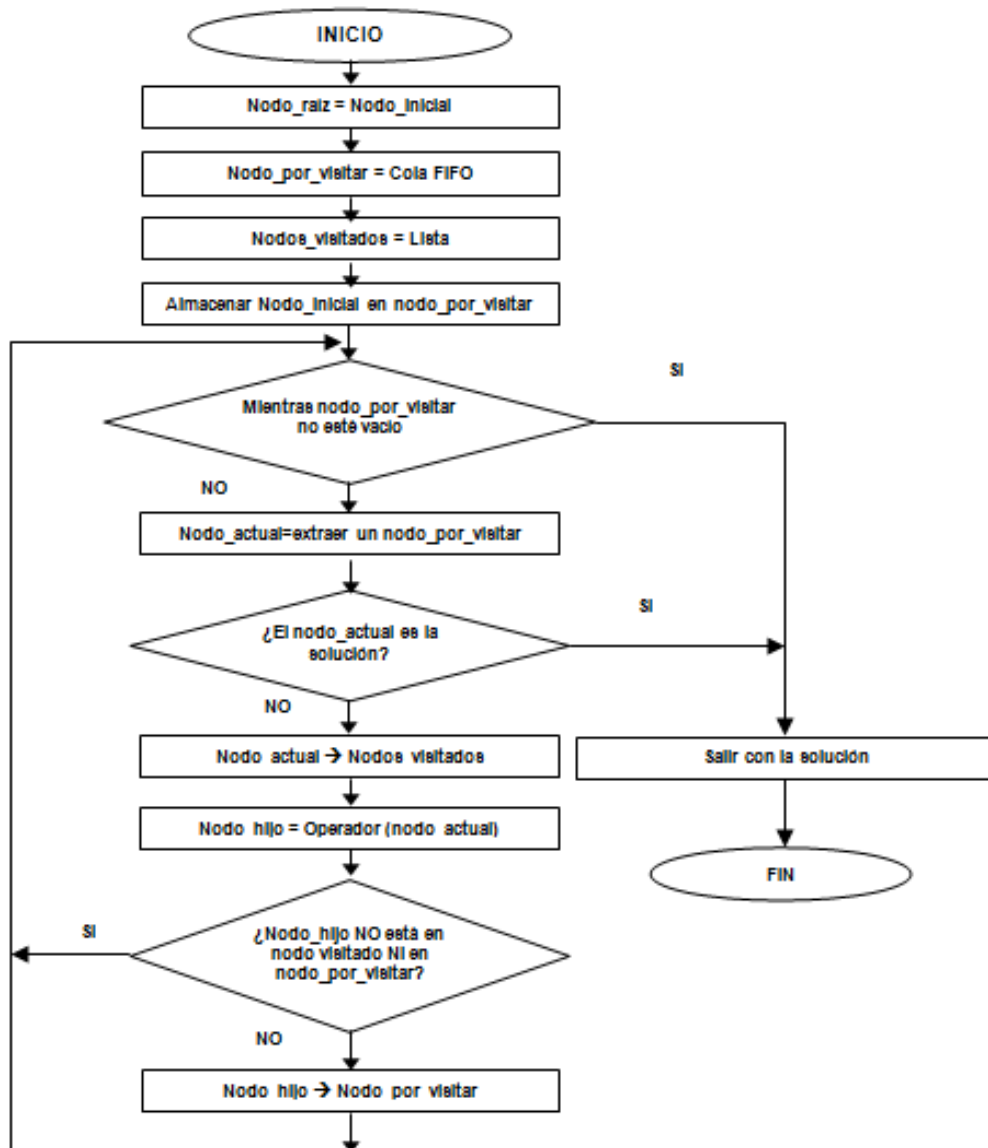


Fig. 22. Diagrama de flujos.

El pseudocódigo descrito a continuación permite encontrar la ruta más corta entre el nodo raíz y el objetivo, siempre y cuando el nodo objetivo exista y esté conectado al grafo.

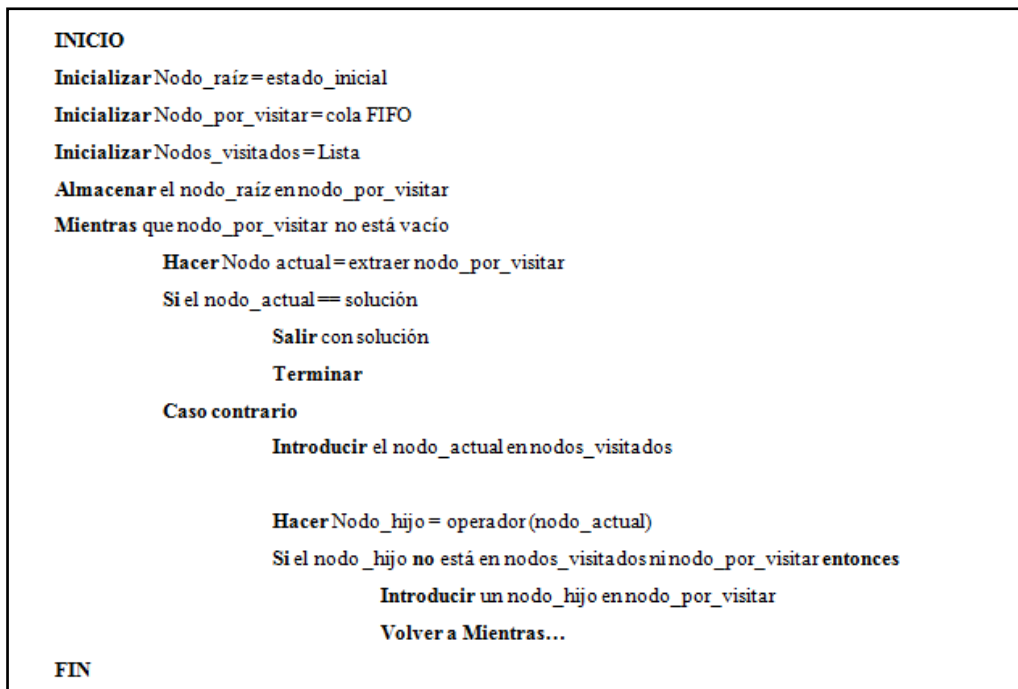


Fig. 23. Pseudocódigo del Algoritmo de Búsqueda en Amplitud

Este recorrido se implementa en la práctica usando una cola FIFO (First In First Out) para la lista de nodos adyacentes. Una cola FIFO es una estructura de datos en la que podemos hacer dos operaciones principales: almacenar un dato y sacar un dato. La particularidad de las colas FIFO es que cuando sacamos un dato, este se extrae en el mismo orden en el que se almacenó.

La aplicación de este algoritmo garantiza la existencia de una ruta óptima, es decir, la ruta más corta en términos de aristas y vértices recorridos. Si hubiera más de una ruta con la misma cantidad de nodos y aristas recorridos, la solución sería la primera de ellas que se encuentre.

Tal como se desarrolló el algoritmo de búsqueda en profundidad y considerando los elementos definidos en la sección precedentemente, se proponemos la aplicación de este algoritmo tal como se ilustra en la figura 24. Allí se muestran las condiciones de inicio. Se crea un autolazo en el nodo A para dar comienzo al algoritmo y se registra el nodo A como visitado. Se recuerda que el objetivo es llegar al nodo R transitando la ruta más corta.

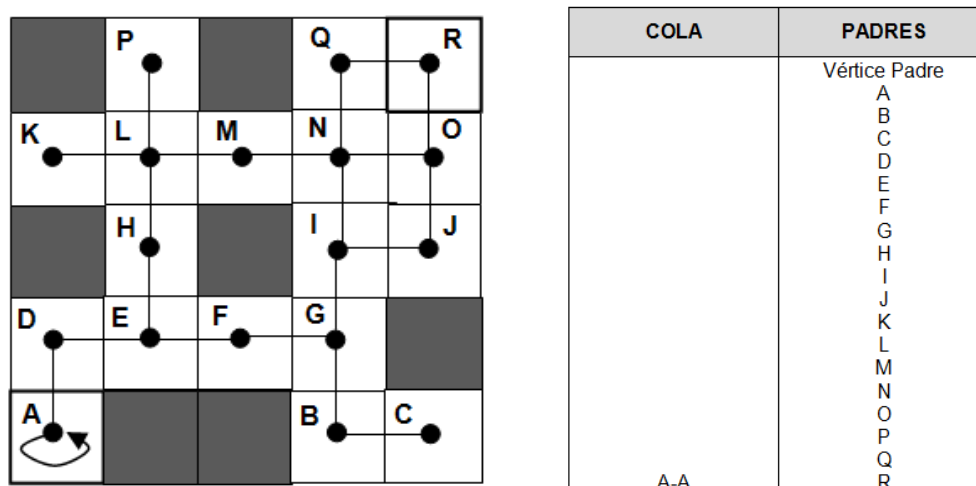


Fig. 24. Condiciones iniciales. Registro de los nodos padres.

Se toma de la cola la arista, se coloca a A como padre del nodo A, se adiciona la arista que sale del nodo A y se marca el nodo que recibe a la arista como conocida. De esta manera, se verifica que el nuevo nodo no es el objetivo. Observe la figura 25.

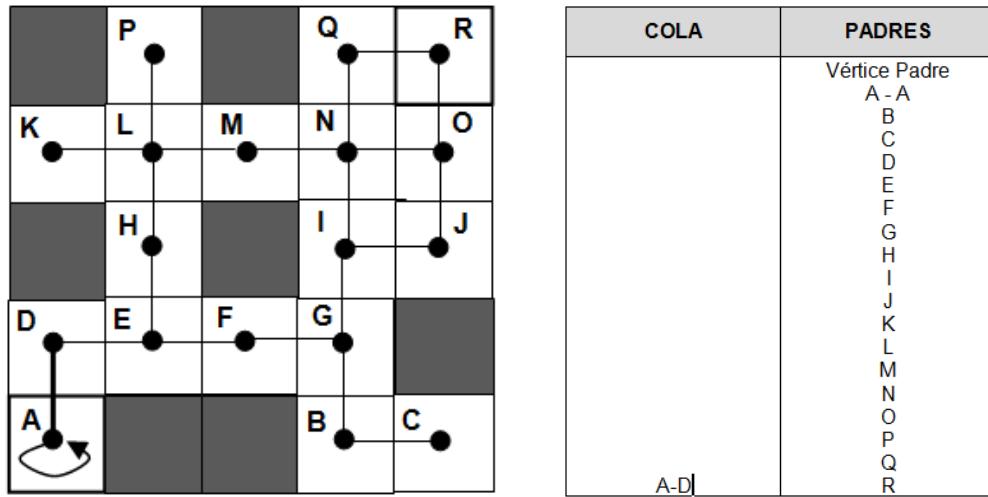


Fig. 25. El robot llega al nodo D. Se registran los datos en la cola.

Se toma la siguiente arista, A-D, se coloca a A como padre de D y se registran las aristas que salen de D, como se muestra en la figura 26.

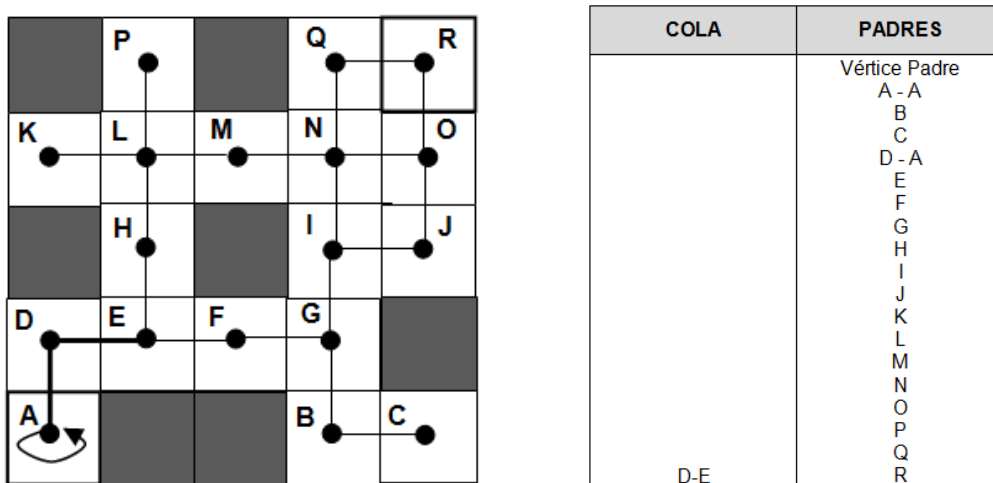


Fig. 26. El robot llega al nodo E. Se registran los nodos padres.

Luego, se toma la siguiente arista, D - E, se coloca a D como padre de E y se registran las aristas que salen de E, E - H y E - F, como se muestra en la figura 27. En este momento se marcan los nodos H y F como conocidos.

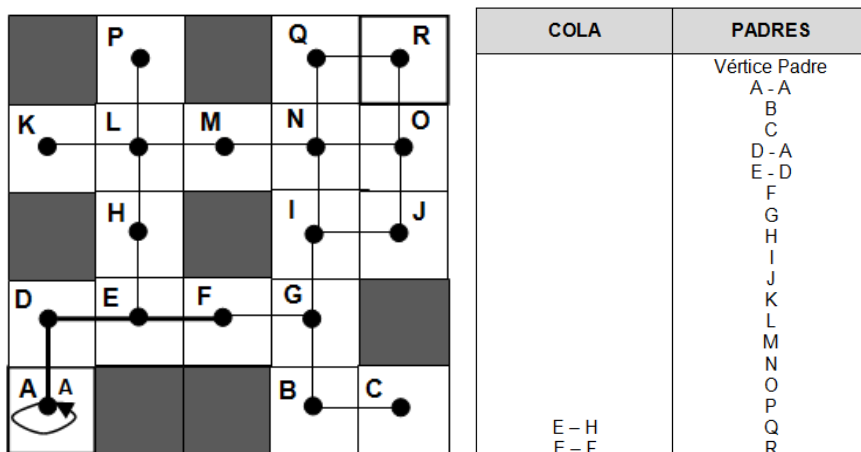


Fig. 27. El robot llega al nodo F. Se registran los nodos padres.

Mientras no se encuentre el nodo objetivo se continúa con la aplicación del algoritmo tal como se demostró para el caso del algoritmo de búsqueda en profundidad. A modo de ilustración, en las figuras 28 y 29, se observa la ruta más corta encontrada tras la conclusión del algoritmo, como así también el grafo resultante. La ruta más corta para este caso está dada por el recorrido de los nodos: A-D-E-F-G-I-J-O-R.

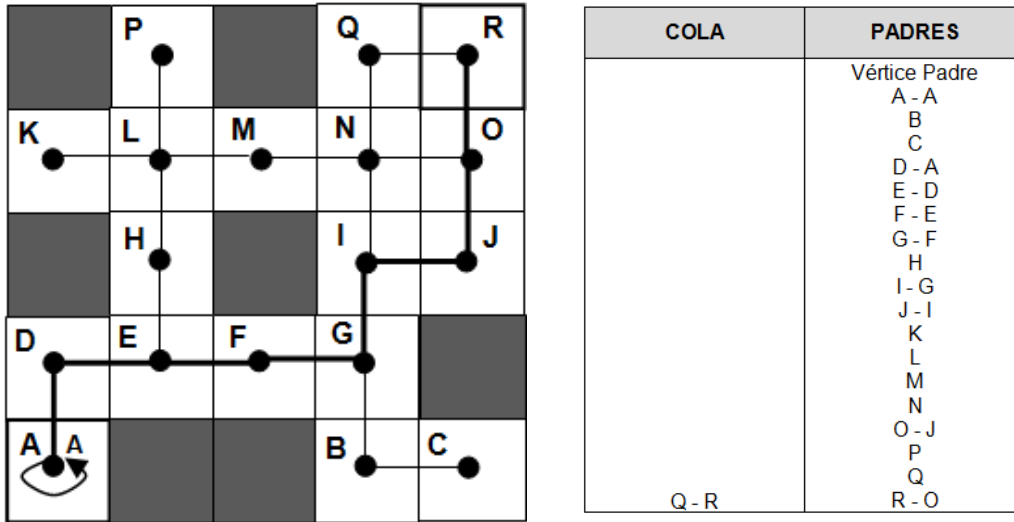


Fig. 28. El robot llega al nodo destino R.

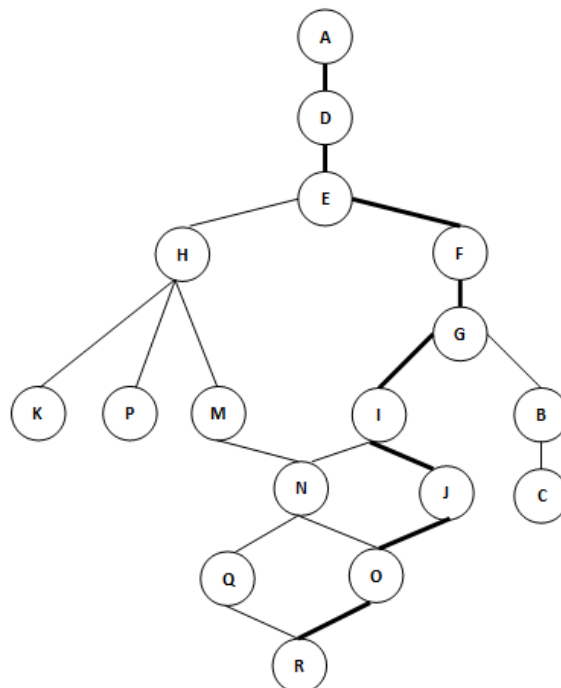


Fig. 29. Diagrama del grafo resultante

3. RESULTADOS OBTENIDOS

En la sección 3.1 se presentan los resultados que oportunamente se obtuvieron mediante la aplicación del Modelo de Red de tipo Backpropagation; y en la sección 3.2, los obtenidos en base a la implementación de los Algoritmos de búsqueda en profundidad y en amplitud respectivamente.

3.1 Resultados obtenidos de la aplicación de la Red de tipo Backpropagation

Considerando que el paradigma es netamente reactivo, basado en la filosofía de estímulo-reacción, los resultados obtenidos por la aplicación de la tecnología de las RNA de tipo Backpropagation en función del entorno, del tipo de robot y su interacción, ha evidenciado las diferentes conductas del robot. Las observaciones realizadas para distintas orientaciones, distintos puntos de partidas, entre otras, deja en evidencia que el modelo resulta suficiente para algunos casos y limitada para otros. Asimismo, cuando el mundo adquiere mayor complejidad, o bien, cuando se hace partir al robot desde otro origen muy diferente al punto de origen con el cual se ha entrenado, éste adquiere conductas que le permiten resolver el problema a veces acertadamente y otras terminan en una colisión o en el ingreso a un bucle del que no logra salir.

Si bien el tiempo de ejecución es más rápido, puesto que actúa en función de la información que le proveen los sensores, no siempre resulta esto eficiente, puesto que se ha observado que eventualmente no logra el objetivo, aunque actúe con mayor velocidad.

3.2 Resultados obtenidos de la aplicación de Algoritmos de Búsqueda

Los experimentos realizados con algoritmos de búsqueda garantizan la existencia de una ruta siempre y cuando el nodo objetivo exista y esté conectado al grafo.

El algoritmo de búsqueda en profundidad ha permitido encontrar una ruta que puede o no ser la más corta mientras que el algoritmo de búsqueda en amplitud ha permitido encontrar una ruta de navegación más corta en términos de nodos visitados. La optimalidad de este algoritmo se basa en función del objetivo que se ha definido para el problema en cuestión; es decir que el robot parta de un nodo raíz y llegue a un nodo destino [10].

Desde el punto de vista computacional, el algoritmo de búsqueda en profundidad requiere mucho menos memoria pues sólo debe guardar el camino actual.

El algoritmo de búsqueda en amplitud no se “pierde” explorando caminos infructuosos que consumen mucho tiempo sin llegar a una solución o de los que no se vuelve nunca; pero computacionalmente, el requerimiento de memoria es mayor, y en consecuencia si aumentara la complejidad del entorno en términos de cantidad de nodos, la situación sería comprometida.

4. CONCLUSIONES

4.1 Sobre la aplicación de la Red Backpropagation

De acuerdo a las investigaciones y experimentos realizados se puede concluir que el robot se desempeña bien dependiendo del lugar de salida e intentando siempre hallar la trayectoria de entrenamiento.

Cabe señalar que el patrón de entrenamiento ha sido suficiente para algunas trayectorias, aunque en ciertos casos dicho patrón no resulta ser representativo del entorno en el cual opera el robot, ya que la mayoría de las salidas de la red correspondientes al movimiento de los motores del robot, es avanzar y por ese motivo es que choca partiendo desde otras posiciones. O bien, ingresa en un bucle sin poder resolver esa situación, quedando inmóvil.

A efectos de disminuir este problema, se aumenta el número de patrones de entrenamiento utilizando varias trayectorias o más largas, vigilando siempre la convergencia de la red para obtener mayor convergencia para los patrones de entrenamiento y máxima generalización, para las nuevas posibles situaciones.

En síntesis, se asume que el uso de las RNA como técnica de navegación de características reactivas proporciona resultados satisfactorios para ciertas trayectorias en la fase de operación, tanto más en la medida que estas trayectorias presenten mayor similitud con las que desarrolló en la fase de entrenamiento; así es que se tendrá por caso, que el robot buscará girar más para el lado que lo hace en la trayectoria de entrenamiento que para el otro. A medida que las trayectorias que se le proponen al robot son tanto más complejas que la que este entrenó, este paradigma exhibe sus limitaciones haciendo que la red no converja y se produzcan situaciones de colisión o bucles en el ambiente de navegación. En otros términos, se produce una incorrecta generalización de la red neuronal para las nuevas situaciones que debe afrontar el robot, las cuales no se encontraban presentes en las trayectorias de entrenamiento.

La idea que subyace detrás de este objetivo central, es que dichas mejoras puedan verse reflejadas en términos de evitación de obstáculos, velocidad de respuesta, optimización de las trayectorias y logro de los objetivos [4].

4.2 Sobre la aplicación de Algoritmo de Búsqueda

La experimentación realizada con el algoritmo en profundidad encontró un camino entre el nodo A y el nodo R cumpliendo con el objetivo propuesto, no asegurando que ésta sea la ruta con el cual el robot visita menos puntos de paso. El experimento realizado con el algoritmo en amplitud encontró un camino entre el nodo A y R cumpliendo no sólo con el objetivo de llegar a destino, sino que además ha visita menos nodos a su paso. Por esa razón, ambos algoritmos se consideran completos, dado que aseguran que aseguran la existencia de una ruta entre el nodo raíz y el nodo destino.

La medida de la performance de ambos algoritmos, en términos de los puntos de paso visitados por el robot en su recorrido para desarrollar su trayectoria, deja ver que la búsqueda en profundidad puede considerarse menos eficiente (subóptima) que la búsqueda en amplitud (óptima). Dado que para cubrir la misma trayectoria entre el mismo origen y el mismo destino y la misma complejidad de mundo, la búsqueda en amplitud necesitó visitar menos puntos en la trayectoria realizada.

El algoritmo de búsqueda en profundidad ha permitido recorrer todos los nodos del grafo de manera ordenada, pero no uniforme mientras que el algoritmo de búsqueda en amplitud recorre todos los nodos de un árbol de manera uniforme y expande cada uno de los nodos de un nivel antes de continuar con el siguiente por esta razón sólo asegura la existencia de una ruta.

El algoritmo de búsqueda en profundidad es muy similar al de búsqueda en amplitud, aunque poseen una diferencia fundamental en cuanto a la estructura de datos usada en cada caso para guardar el registro de las aristas.

A diferencia de las RNA, los algoritmos de búsqueda conocen cuál es el nodo objetivo al que se desea llegar y pueden guardar información sobre cuáles nodos y aristas se han visitado previamente; mientras que las RNA actúan en función de los estímulos que recibe y de los ejemplos con los cuales ha sido entrenada, es decir que si es mayor el número de ejemplos con el cual se entrena puede inferirse que aprenderá mejor. En ambos casos deben tener un conocimiento previo del entorno con el cual interactúan.

En este sentido, las redes neuronales no siempre encuentran una ruta acertada de acuerdo a los requerimientos, y los algoritmos de búsqueda, de acuerdo al tipo que se utilice, no sólo entrega una ruta, sino que puede incluso entregar la ruta más corta en términos de puntos de paso visitados, los que pueden traducirse en espacios de trabajo que el robot deba visitar previamente antes de llegar a destino.

Por otro lado, dado que la arquitectura de la red Backpropagation actúa en base a un sistema sensor-actuar de características reactivas, esto le proporciona mayor velocidad al vehículo robótico; como contrapartida, la implementación de los algoritmos de búsqueda, de carácter deliberativo y secuencial (desde el punto de vista de la programación), por lo tanto es más lento, lo cual puede traducirse, en un mundo complejo, en una desventaja en términos de tiempo, dinero, consumo energético, entre otros.

5. FUTURAS LÍNEAS DE INVESTIGACIÓN

Por todo lo expuesto precedentemente cabe considerar la consiguiente investigación sobre el objetivo de mejorar el desempeño del robot en lo que se refiere a sus conductas de navegación.

Complementando el concepto expresado en el punto anterior, considerar la aplicación de técnicas de razonamiento de alto nivel de tipo deliberativas (aprendizaje automático y planificación autónoma de tareas entre otras) como complemento de las técnicas reactivas; las cuales, si bien son de menor

velocidad de reacción que estas, también le permiten al robot afrontar requerimientos de desempeño más complejos.

En este sentido, se propone como futura línea de investigación el análisis de la performance del robot en ambientes estructurados de mayor complejidad como así también su desempeño en ambientes de operación de carácter dinámico, en los cuales los obstáculos pueden cambiar de posición mientras el robot realiza su tarea.

Asimismo, y dado el objetivo planteado por los autores, se considera la aplicación de algoritmos de búsqueda más eficientes como nuevas técnicas para la resolución de este caso de estudio. Esto es posible, pues el robot se desempeña en un ambiente estructurado, es decir invariante en el tiempo. Con lo cual, y bajo estas circunstancias se plantean alternativas como algoritmos de búsqueda con costos, como posibles tópicos de análisis, como así también el abordaje de algoritmos de búsqueda informada, tales como el Algoritmo de Dijkstra y el Algoritmo A*.

6. REFERENCIAS

- [1] R. García Martínez, M. Servente, D. Pasquín. "Sistemas Inteligentes", Ed. Nueva Librería, Buenos Aires, Argentina, 2003.
- [2] J. Santos, R. Duro, "Evolución Artificial y Robótica Autónoma", Ed. Alfaomega – Ra-Ma, México 2005.
- [3] I. Harvey, "Artificial Evolution and Real Robots, Proceedings of International Symposium on Artificial Life and Robotics (AROB)", Masanori Sugisaka (Ed), Beppu, Japan, pp. 138-141, 1996.
- [4] A. Ollero Baturone, "Robótica Manipuladores y robots móviles". Ed. Alfaomega – Marcombo, Barcelona España, 2007.
- [5] J. R. Hílera, V. J. Martínez, "Redes Neuronales Artificiales. Fundamentos, modelos y aplicaciones", Ed. Alfaomega Ra-Ma. Madrid, España, 2000.
- [6] A. A. Hossian, L. A. Cejas, L. V. Olivera, "Estudio del comportamiento de vehículos robóticos en ambientes computarizados. Hacia un enfoque basado en conductas del tipo estímulo-respuesta", IVº Jornadas de Enseñanza de la Ingeniería JEIN - UTN – FRA, Buenos Aires, Argentina 2014.
- [7] A. Scillato, D. Colón, J. Balbuena, "Técnicas de Navegación Híbrida para Navegación de Robots Móviles", Ed. Rama de Estudiantes del IEEE. Tesis de grado para obtener el grado de Ingeniero Electrónico. Universidad Nacional del Comahue 2005.
- [8] A. A. Hossian, R. García Martínez, L. V. Olivera, "Robótica de Navegadores. Un enfoque desde las Tecnologías Inteligentes", Ed. Nueva Librería, Buenos Aires 2014.
- [9] N. Arriola Landa Cosio, "Inteligencia Artificial", Gradi, Buenos Aires 2007.
- [10] A. García Serrano, "Inteligencia Artificial. Fundamentos, práctica y aplicaciones", Alfaomega, México 2012.