



Ra Ximhai

ISSN: 1665-0441

raximhai@uaim.edu.mx

Universidad Autónoma Indígena de México
México

Mariscal-García, Efraín

Planeación y seguimiento de trayectorias de robots móviles en una simulación de un ambiente real

Ra Ximhai, vol. 1, núm. 1, enero-abril, 2005, pp. 177-200

Universidad Autónoma Indígena de México

El Fuerte, México

Disponible en: <http://www.redalyc.org/articulo.oa?id=46110112>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

PLANEACIÓN Y SEGUIMIENTO DE TRAYECTORIAS DE ROBOTS MÓVILES EN UNA SIMULACIÓN DE UN AMBIENTE REAL

PLANNING AND FOLLOWING OF TRAJECTORIES FOR MOBILE ROBOTS IN A SIMULATION OF A REAL WORK ENVIRONMENT

Efraín Mariscal-García

Titular Académico. Universidad Autónoma Indígena de México. Los Mochis Sinaloa, México. Correo Electrónico: emariscal@uaim.edu.mx

RESUMEN

Este trabajo de investigación, presenta el desarrollo de un sistema de planeación y seguimiento de trayectorias por medio de un robot móvil para transportar objetos en cooperación con otros robots en un ambiente real de trabajo y que pueda ser implementado a bajo costo, ya que no requiere de ambiente controlado, sin sacrificio de confiabilidad o precisión, para aplicación inmediata en problemas reales. Los principales objetivos de esta investigación son: construir un ambiente simulado en una computadora para llevar a cabo el desarrollo de las actividades de un robot móvil, utilizando algunas herramientas programación de Basic 7 con rutinas de lectura de pantalla en lenguaje de ensamblador. Desarrollar la planeación de la trayectoria del mismo en la cual se desplazará evadiendo obstáculos y llevándola de manera suave. Se aplicaron algoritmos para generación de espacios de trabajo, exploración del ambiente, creación del mapa y trazar trayectorias. Se encontró que los robots móviles pueden seguir una trayectoria segura sin colisiones utilizando la técnica de exploración a través de un sistema de señales en los que las marcas indicadas en el suelo hacen que el trabajo sea más preciso.

Palabras Claves: Robots móviles, ambiente simulado.

SUMMARY

This research work illustrates the development of a system for planning and following the trajectories of a group of mobile robots used for transportation of objects in an actual work environment in cooperation with other robots, being able to be implemented at low cost, without a controlled environment without sacrificing reliability nor precision with immediate application in actual situation. The main objectives of this research are: building a simulated environment in a computer to carry out the development of the activities of a mobile robot, using some programming tools of the BASIC 7 programming language with routines for screen reading in assembler language. To develop the planning of trajectory of the same robot in which it will move evading obstacles in a smooth way. Some algorithms were applied for generation of work spaces, scouting the environment, creating an area map and drawing trajectories. It was found that mobile robots are capable of following a safe trajectory without collisions by using scouting technique through a signaling system in which the marks are indicated in the floor making the movements more precise.

Key Words: Mobile robots, simulated environment.

INTRODUCCIÓN

Por siglos el ser humano ha construido máquinas que imitan las partes del cuerpo humano. El desarrollo en la tecnología, donde se incluyen las poderosas computadoras electrónicas, los actuadores de control retroalimentados, y transmisión de potencia a través de engranes, la tecnología en sensores han contribuido a flexibilizar los mecanismos autómatas para desempeñar tareas dentro de la industria.

El término "robótica" le debe gran parte de su popularidad al escritor Isaac Asimov, quien la comenzó a utilizar en una de sus obras en 1942. Éste propuso "Leyes de la Robótica", las cuales en un principio fueron sólo tres, pero luego añadió una cuarta llamada ley cero. Estas son (Pasos, 1984):

- **Ley cero:** Un robot no puede dañar a la humanidad, o a través de su inacción, permitir que se dañe a la humanidad.
- **Primera ley:** Un robot no puede dañar a un ser humano, o a través de su inacción, permitir que se dañe a un ser humano.
- **Segunda ley:** Un robot debe obedecer las órdenes dadas por los seres humanos, excepto cuando tales órdenes estén en contra de la primera ley.
- **Tercera ley:** Un robot debe proteger su propia existencia, siempre y cuando esta protección no entre en conflicto con la primera y la segunda ley.

Robots móviles

Son máquinas móviles, mediante rodamientos o extremidades flexibles, controladas por una inteligencia artificial, pueden pensar, decidir y aprender de su experiencia. Pueden implementar capacidades de aprendizaje y de tipo evolutivo, lo que le puede llevar a reconocer y aprender ante las situaciones que se les presenten en su interacción con el mundo que les rodea (Ollero, 2001; Hirzinger, 2001).

Limitaciones de los robots

La mayoría de los robots se utilizan en lugares cerrados y bajo ciertas normas de seguridad, aunque sigue investigando para lograr que los robots se enfrenten en lugares más difíciles, como lo son las calles. Cuando se encuentran en lugares públicos, deben contar con sensores suficientes para poder detectar a una persona ya sea niño o adulto, debe de reducir velocidad para evitar algún choque.

Otra dificultad es que necesitan recargar sus baterías de manera periódica, ya que consumen una gran cantidad de energía de manera rápida debido a los trabajos forzosos que realizan. Tienen limitaciones para desplazarse en algunos ambientes tales como: escaleras, tierra, lugares robustos (entre palos y piedras), lo cual dificulta al robot a seguir con su trayectoria (Sugar y Kumar, 2001).

Planeación de trayectorias

Se reconoce a la planeación de trayectorias como la búsqueda de una sucesión de posiciones de un robot que permiten llevarlo de un estado inicial a uno final, entendiéndose como estado a la descripción de la ubicación del robot referenciada a un marco absoluto generalmente expresada por la combinación de las coordenadas cartesianas del centro del robot y la posición angular del eje principal de éste. La configuración que adquiere la trayectoria se define por la distribución de los obstáculos a lo largo de todo el ambiente de trabajo y por supuesto de la geometría del robot, así como de sus capacidades del movimiento (Yamashita *et al.*, 2003).

De esta manera, la topología del ambiente del trabajo, restringirá el espacio libre de obstáculos por el cual se pueden expresar las posibles trayectorias para alcanzar el estado final deseado. Generalmente se recurre a discretizar la presentación del espacio en el ambiente de trabajo con lo que se extrae una representación segura, es decir, se tendrá garantía que el espacio libre podrá ser ocupado por el robot sin riesgo de colisión, por lo que es necesario que tal discretización se haga en relación a las características geométricas tanto del robot como de los obstáculos (Sugar y Kumar, 2001).

La planeación de trayectorias puede realizarse de una forma dinámica, considerando la posición actual del vehículo y los puntos intermedios de paso definidos en la planeación de la ruta. La definición de la trayectoria debe tener en cuenta las características cinemáticas del vehículo, por ejemplo: en vehículos con rueda y tracción convencional, interesa definir trayectorias de curvatura continua que pueden ejecutarse con el menor tiempo posible (Sugar y Kumar, 2001).

Una vez realizada la planificación de la trayectoria, es necesario planificar movimientos concretos y controlarlos para mantener al vehículo en la trayectoria planificada. Así se plantea el problema para el seguimiento de caminos, que para vehículos con ruedas se concreta de determinar el ángulo de dirección teniendo en cuenta la posición y orientación actual del vehículo con respecto a la trayectoria que debe seguir.

Robots como agentes cooperativos

Los robots cooperativos son un grupo de robots capaces de planear cooperativamente movimientos de objetos, cada robot sabe cooperar con los demás cuando éstos requieran apoyo de él, así como pedir la cooperación de los demás cuando sea requerida. Un robot puede estar atendiendo diferentes procesos o lógicamente en uno mismo. Un robot distribuido puede planear o actuar, pero también puede responder a otros robots que soliciten ayuda (Smith, 1981).

El proceso de distribución se enfoca principalmente al problema de coordinar a un grupo de robots que realizan un conjunto de tareas afines y solucionan problemas en forma conjunta y cooperativamente; identificando aquellos subproblemas que cada uno puede resolver, solucionándolo concurrentemente e integrando sus resultados parciales para obtener una solución global del problema (Smith, 1981).

Al distribuir a los robots se distribuye también el control y las responsabilidades de cada uno en un problema. Cada robot debe conocer cuáles son sus propias capacidades y razonar acerca de las interacciones, creencias y planes de otros robots. Debido a ello, para crear sistemas que cuenten con inteligencia se deben tomar en cuenta aspectos importantes como: las acciones de los robots deben ser coherentes a las metas globales

del grupo, así como el control de la capacidad de transmisión y recepción de mensajes que los robots deben poseer para una eficiente cooperación.

Los principales objetivos de esta investigación fueron construir un ambiente simulado en una computadora para llevar a cabo el desarrollo de las actividades de un robot móvil, utilizando algunas herramientas programación de Basic 7 con rutinas de lectura de pantalla en lenguaje de ensamblador. Desarrollar la planeación de la trayectoria y su desplazamiento evadiendo obstáculos y llevándola de manera suave.

MATERIALES Y MÉTODOS

Etapas de trabajo de acción de los robots cooperantes

Para el mejor manejo de la investigación fue necesario dividir las tareas en dos puntos importantes:

- Formar equipos y asignar funciones.
- Trazar trayectorias.

Formar equipos y asignar funciones

Para el desarrollo de las etapas de trabajo antes mencionadas, se recopilaron datos tales como: cantidad de robots, objetos posibles y las distancias entre ambos elementos. El objetivo de esta etapa es detectar el objeto más cercano, por lo menos a dos robots.

Trazar trayectorias

Nuestro espacio de trabajo se dividió en secciones por celdas, en donde los robots scout son los que se encargan de explorar todo el espacio de trabajo y poder encontrar las posibles trayectorias. Teniendo ya calculados todos los caminos posibles tras el proceso de inspección de los robots scout, un algoritmo encontró un camino óptimo y más seguro, en este caso; el más corto de entre dos puntos. La distancia entre los robots y los objetos es de gran importancia ya que es la que determina la jerarquización de los robots. Por lo tanto, el robot que se encuentre más cerca de un objeto toma el mando de

liderazgo, el que sigue se considera el de carga y el último de la jerarquía el robot explorador (scout).

Cuando los scout trazan las posibles trayectorias, le envían el mapa del ambiente con las trayectorias trazadas al líder, cuando ya se tiene conocimiento de la mejor trayectoria se opta por seguir la línea de la trayectoria, mientras que el de carga va siguiendo al líder. Si la trayectoria trazada se bloquea ya sea que se mueven o ponen objetos dentro de la ruta que había sido trazada por el robot explorador; inmediatamente el líder le envía información al scout de que la trayectoria trazada ha sido bloqueada, inmediatamente el explorador vuelve hacer un escaneo del área para encontrar otra trayectoria posible y que el robot líder pueda llegar a su meta junto con el robot cargador.

Algoritmos para la planeación de trayectorias

Planear la trayectoria ha sido uno de los elementos más importantes para los robots móviles, ya que estos robots tienden a moverse por lugares desconocidos y a su vez evadir obstáculos de manera precisa. La utilización de robots-vehículos ha tenido lugar para realizar ciertos trabajos peligrosos para la vida humana como por ejemplo: desactivación de bombas, exploración en túneles, en carreras aeroespaciales y transporte de material peligroso.

La generación de trayectorias en este tipo de robots son ejecutados por poderosos programas que manejan y manipulan datos necesarios en los cuales son representados por un mapa virtual de su medio (posiciones del robot y lugares con obstáculos).

Descripción de la planeación de trayectorias

Dado un robot con su posición y orientación inicial, así como la descripción del espacio de trabajo, debemos encontrar una trayectoria para así poder obtener una posición de orientación final, pero sin embargo cuando ponemos obstáculos en el espacio de trabajo del robot, obliga al robot a encontrar dicha trayectoria desplazándose y evitando cualquier colisión con obstáculos.

Por lo cual, podemos resolver el problema de la planeación de trayectorias libres de colisiones (choque) explorando el ambiente de trabajo y mapeando los obstáculos que

se encuentren alrededor del robot, dentro de un espacio cartesiano, para después crear los modelos de los objetos. Uno de los problemas iniciales sería hacer que un robot móvil recuerde un camino desde la entrada de un cierto lugar hasta la salida y que ese camino sea el más adecuado para seguir la trayectoria planeada. Este proceso para planear la generación de la trayectoria se subdivide en tres algoritmos que serían creados para la solución de problemas específicos y puntuales. De este modo los algoritmos a realizar serían los siguientes:

- Un algoritmo que controlará la inspección de la trayectoria hecha por el robot y ya teniendo el conocimiento del entorno, podrá generar puntos libres de obstáculos que servirán para los otros algoritmos.
- Otro algoritmo para que el robot móvil siga una trayectoria calculada.
- Un tercer algoritmo para generar la trayectoria.

Sabemos que para el funcionamiento de los algoritmos es necesario detallar parámetros de entrada ya que necesitaremos conocer el ambiente para que los algoritmos funcionen correctamente para lo cual tenemos los siguientes puntos:

- Coordenadas X y Y del robot representado en un mapa cartesiano.
- Detección de obstáculos (los obstáculos que se van a evadir).
- Medidas físicas del espacio de trabajo en donde se va a mover el robot.

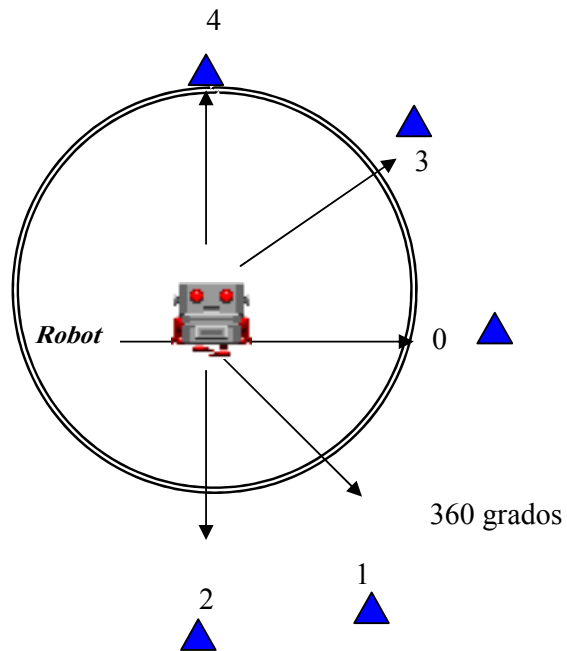
Teniendo claros los datos que necesitaremos del entorno del espacio de trabajo, se explica cada uno de los siguientes algoritmos:

Algoritmo de inspección

Como se ha anticipado, este algoritmo proporciona los puntos libres de obstáculos que necesitarán los otros algoritmos para realizar su tarea. Este algoritmo controla al robot en la tarea de inspección del espacio de trabajo, en donde el robot construye un mapa virtual de las paredes del lugar teniéndolo memorizado para saber qué alternativas posibles tiene para ir desde un punto inicial a un punto final. El robot detecta la

presencia o la ausencia de obstáculos en las direcciones norte, noreste, noroeste, este y oeste, referenciada al propio robot (Figura 1).

Figura 1. Dirección del robot a 360 grados de detección de obstáculos.



Con estas posiciones para la detección de obstáculos, el robot trabajará para detectar y memorizar alternativas para luego ejecutarlas si es necesario. El sistema de detección de obstáculos debe de tener una distancia de detección tal que sea capaz de visualizar las paredes del lugar y en esos casos no existen alternativas posibles, pero ha de poder ver huecos en las paredes que serían alternativas para llegar a la salida del espacio de trabajo, para poder ver los huecos. La distancia de detección será de un valor que dependa de la dimensión del lugar, concretamente el ancho del pasillo del laberinto, para que la ausencia de un obstáculo signifique invariablemente la presencia de un hueco en la pared, es decir una alternativa, y la detección de obstáculos signifique una pared sólida donde no existe salida por ahí. Veamos las Figuras 2 y 3 en donde se muestran los valores límites de la distancia de detección de obstáculos.

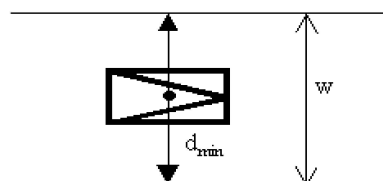


Figura 2. Valor mínimo de la distancia de detección de obstáculos.

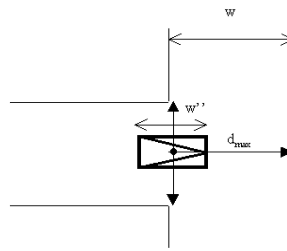
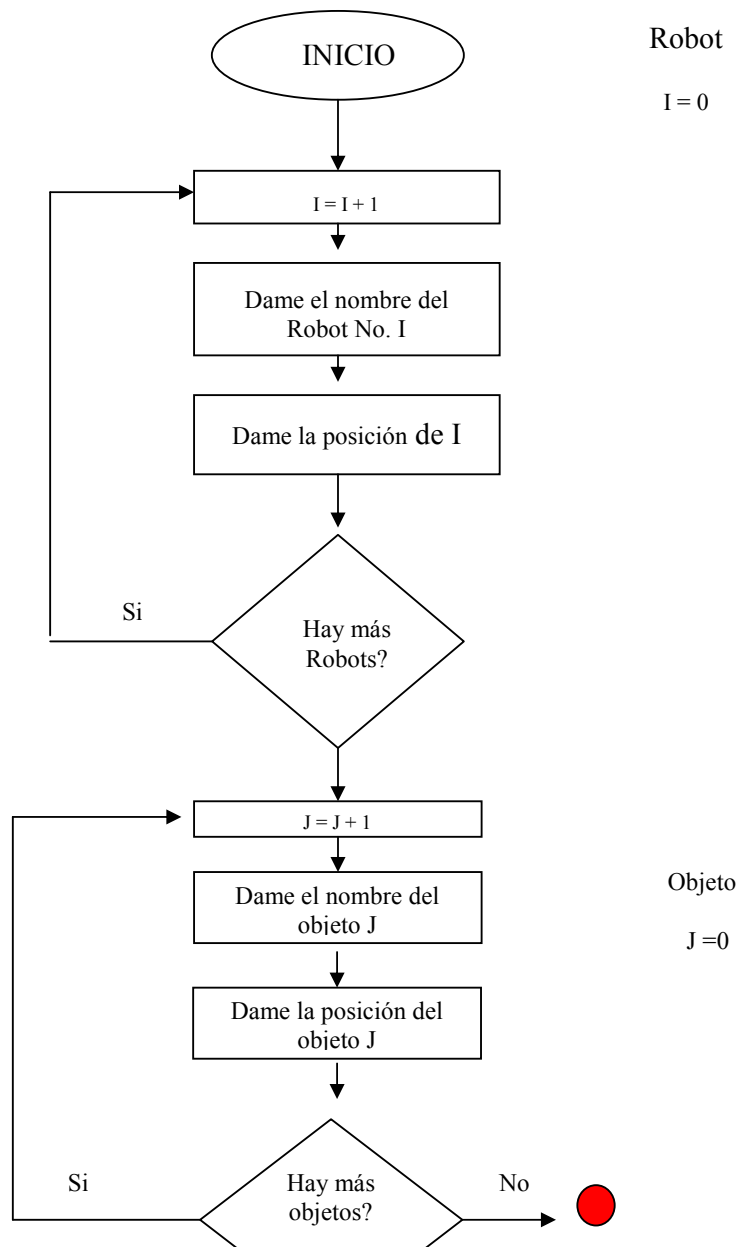


Figura 3. Valor máximo de la distancia de detección de obstáculos.

La Figura 4, muestra el diagrama de flujo del algoritmo, para encontrar en el espacio de trabajo cuántos robots y objetos activos se encuentran disponibles. La función de este algoritmo es encontrar todos los robots y los tipos de objetos con sus posiciones dadas en las coordenadas en X y Y del mapa cartesiano.

Figura 4. Diagrama de selección de objetos y robots en el mapa cartesiano.



La Figura 5 muestra el diagrama de flujo del algoritmo para calcular la distancia de los robots entre los objetos, para saber cual es el X robot más cercano de Y objeto y así empezar a calcular la jerarquización de cada uno de los robots.

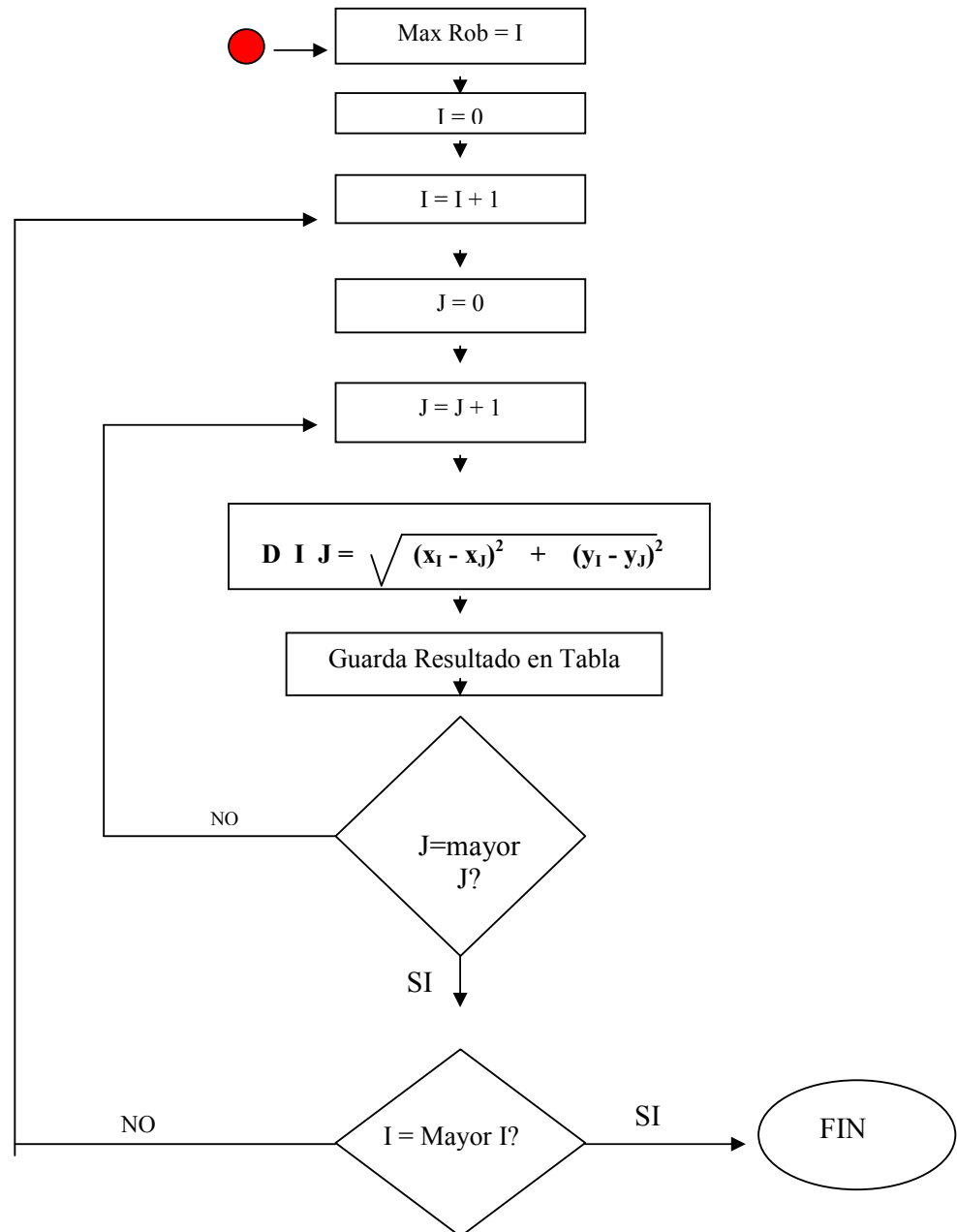


Figura 5. Diagrama de flujo para calcular distancias.

Algoritmo de asignación de jerarquización de miembros de equipo

Una constante indica el número de renglones en la tabla OD (orden por distancia) en donde se utilizaron cuatro variables temporales (Cuadro 1) la cual describimos a continuación:

- La primera variable lleva el valor del renglón de la tabla OD actual.
- La segunda variable, lleva el rango de la asignación siguiente.
- La tercera variable lleva el número o el nombre del robot al que se le asignará la función.
- La cuarta variable contienen el número o el nombre del robot que se encuentre en el renglón de la tabla OD que se está analizando.

Cuadro 1. Variables y constantes del equipo de trabajo (robots)

Renglones de tabla	=	constante
Renglón actual	=	variable 1
Asignación	=	variable 2
Robot anterior	=	variable 3
Robot actual	=	variable 4

Para el mejor manejo de conceptos técnicos de las palabras a utilizar, fue necesario crear abreviaturas para utilizarlas en los diagramas de flujo de los algoritmos.

- RT = Renglones de tabla
- RA = Renglón actual
- ASG = Asignación
- RobAnt = Robot anterior
- RobActual = Robot actual
- OD = Orden por distancia

Para el funcionamiento de los robots fue necesario crear un diagrama de flujo del algoritmo de asignación de funciones el que se va a encargar de decidir la jerarquización (liderazgo, carga y exploración) (Figura 6).

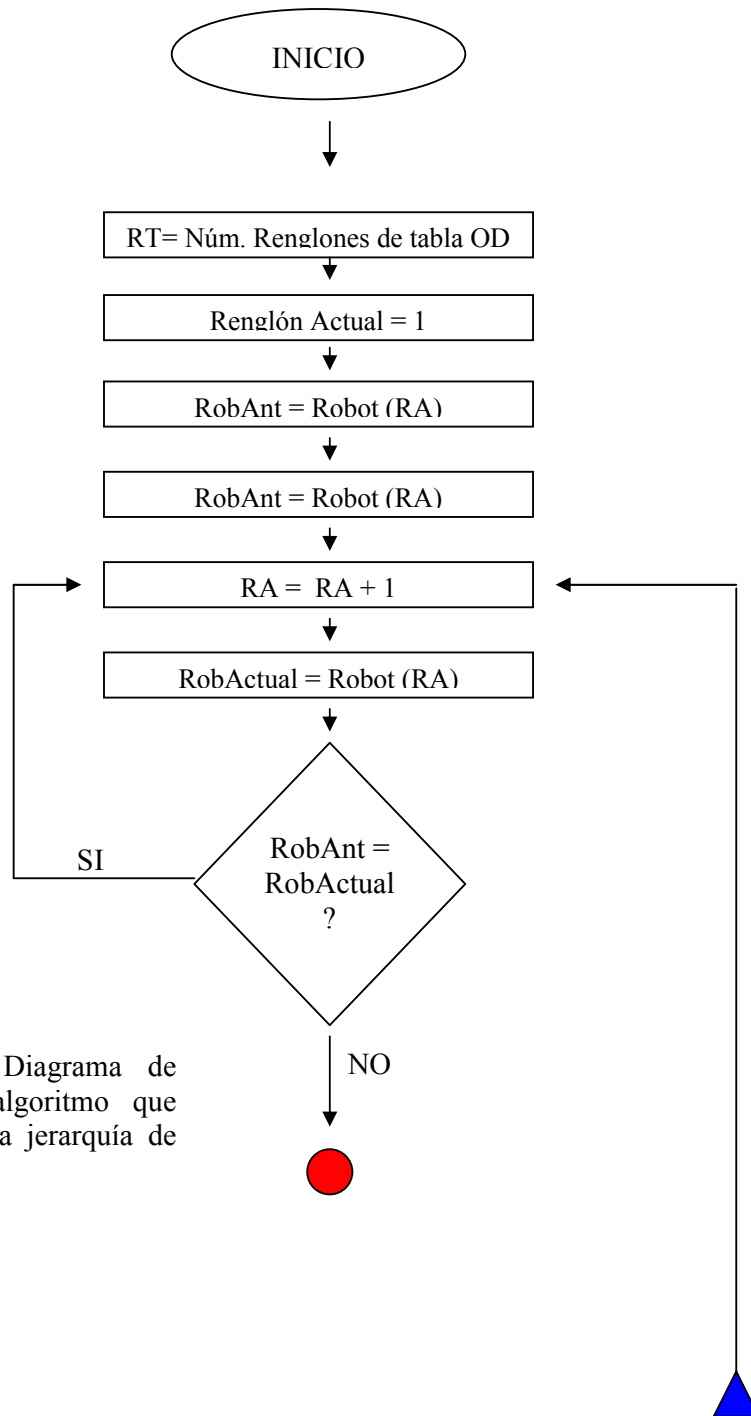
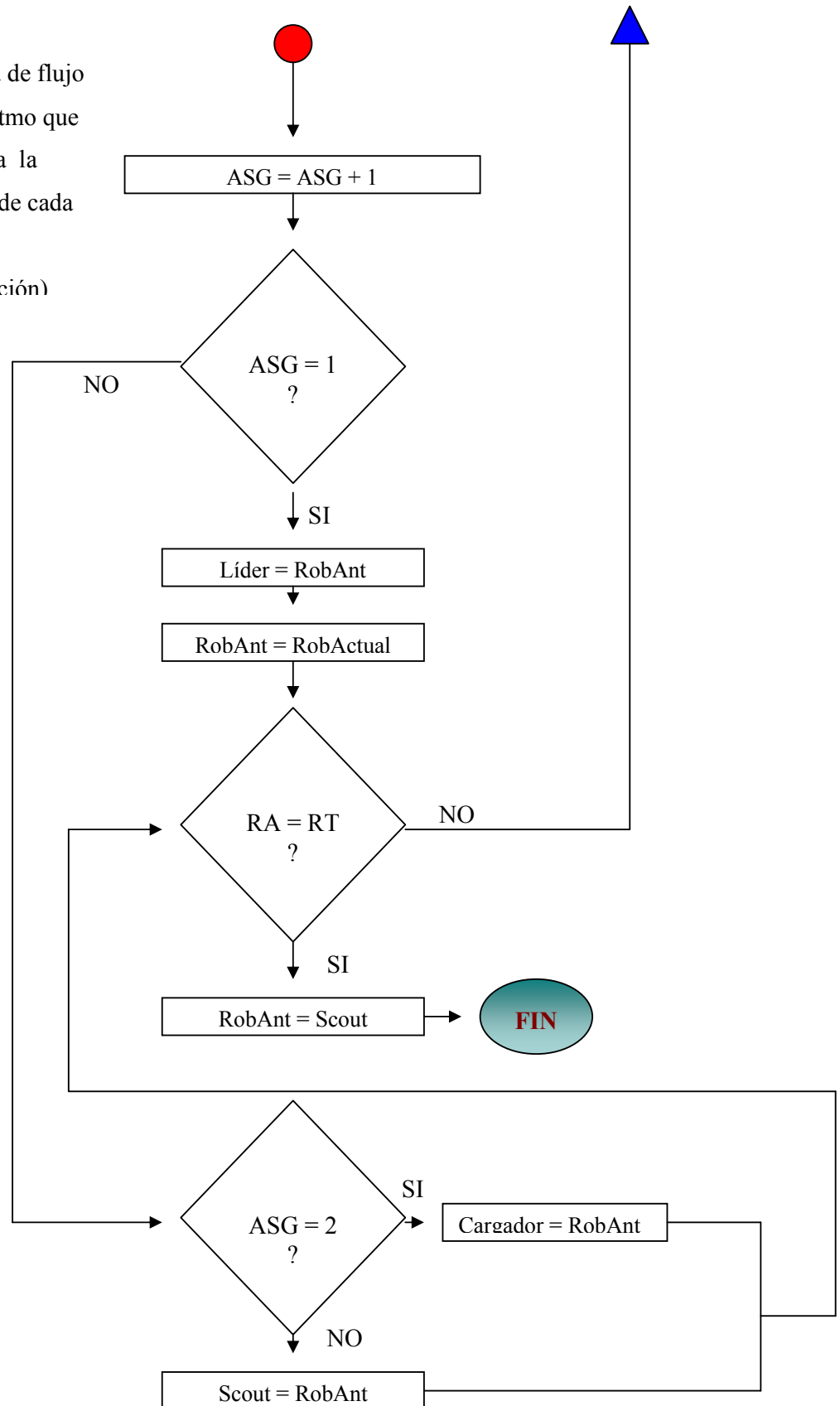


Figura 6. Diagrama de flujo del algoritmo que selecciona la jerarquía de cada robot.

Figura 6.
Diagrama de flujo
del algoritmo que
selecciona la
jerarquía de cada
robot
(continuación)



Analizando el algoritmo propuesto anterior, se realizó un análisis del diagrama de flujo para la asignación de jerarquías de cada robot (Cuadro 2).

Cuadro 2. Análisis del algoritmo de jerarquización

1. RT = 9, RA = 1, RobAnt = 2
2. RA = 2, RobActual = 3, ASG = 1, Robot 2 = Líder, RobAnt = 3
3. RA = 3, RobActual = 3
4. RA = 4, RobActual = 3
5. RA = 5, RobActual = 2, ASG = 2, Robot 3 = Cargador, RobAnt = 2
6. RA = 6, RobActual = 2
7. RA = 7, RobActual = 1, ASG = 3, Robot 2 = Scout, RobAnt = 1
8. RA = 8, RobActual = 1
9. RA = 9

Para lograr que los robots interactúen en un espacio de trabajo, fue necesario utilizar el mapa cartesiano para representar las coordenadas de X y Y de los robots y los objetos.

Robots	X	Y
A	3	4
B	6	8
C	4	5
D	7	2
E	6	1

La siguiente fórmula sirve para sacar la distancia de los objetos J hacia los robots I para después vaciarlos en una tabla y conocer cuál es el nivel de mando de cada robot (Cuadro 3).

$$D I J = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

En el Cuadro 3 se muestra claramente, que al aplicar la fórmula de la distancia para encontrar la jerarquización de los robots, el robot B está más cerca del objeto B; eso quiere decir que el robot B pasa a ser el líder del grupo, el C pasa a ser el cargador y por último el A es el scout.

Cuadro 3. Tabla de la distancia de los robots hacia los objetos.

Robots	Objetos	Distancia
A		
	B	6
	r	8
B		
	B	2
	r	9
C		
	B	4

Algoritmo de seguimiento de una trayectoria

Para este algoritmo, la única entrada necesaria es la trayectoria a seguir, cosa que ya nos ha de haber sido proporcionada con la ayuda de los anteriores algoritmos, su función básica es guiar al robot desde la entrada hasta la salida del camino a través de la trayectoria dada. En la Figura 7 se muestran las posiciones actuales para el seguimiento de la trayectoria.

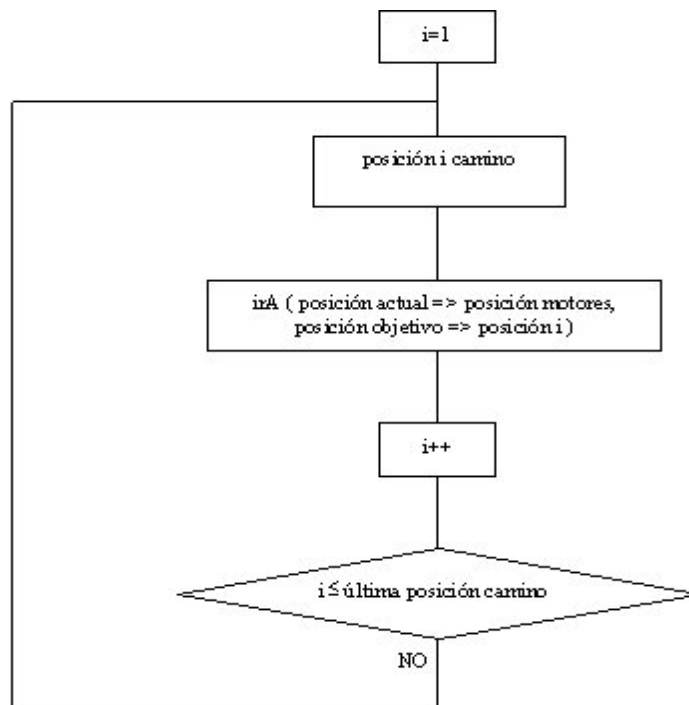


Figura 7. Diagrama de flujo del algoritmo que realiza las posiciones posibles para el seguimiento de la trayectoria del robot

En lo siguiente, el robot líder se encargará de generar las trayectorias a partir de un mapa escalado que es proporcionado por los robots scout. Este mapa debe contener todos los obstáculos con dimensiones y ubicaciones lo más apegada a la realidad, ya que de esto dependerá la exactitud del movimiento emprendido a posteriori por el robot. El sistema está compuesto por una serie de rutinas programadas en lenguaje Basic, las cuales tienen entre sus varios objetivos: la generación del mapa del ambiente y la ejecución de la trayectoria.

La generación del mapa consta de una serie de rutinas que permiten el diseño del propio mapa en un ambiente gráfico con dimensiones reales, en donde el usuario introduce los obstáculos por posicionamiento del cursor en el área deseada. Se tiene la facilidad de salvar y guardar los mapas; lo que permite emplear al robot en diversos ambientes. Se cuenta además, con una opción de conversión del mapa a espacio de trabajo, la cual se encargará de expandir todos los obstáculos en las dimensiones del robot, con la finalidad de que el mapa final sea adecuado y confiable para poder buscar una trayectoria. De esta manera, el mapa es discretizado a las dimensiones del robot y se representan por una matriz de unos y ceros, donde los ceros significan espacio libre y los unos obstáculos.

Una vez que los robots scout hayan explorado el ambiente o espacio de trabajo, es necesario crear un punto de inicio y el punto en donde se quiere que llegue el robot. Con estos datos, el robot líder tendrá que buscar, sin instrucciones externas adicionales, generar una trayectoria que sea óptima en distancia.

Después de que los robots scout hayan explorado el ambiente de trabajo, los robots llegan a conocer la zona sobre la cual se van a mover, es decir, los algoritmos para la planeación de trayectorias toman como entrada un mapa sobre el cual debe navegar el robot y entregan como salida una secuencia de movimientos, que se encadenan como una trayectoria para el robot buscando generalmente llevarlo del punto actual a un punto diferente definido como meta.

El sistema de control del robot consiste en 3 elementos, integrados en una representación homogénea de conductas:

- Navegación básica: consiste en evitar obstáculos y seguimiento de bordes.
- Detección de marcas: categorización del mundo identificado.
- Construcción y actualización del mapa y la planeación de caminos.

Descripción del espacio de trabajo

Para la creación de un ambiente de trabajo simulado como un universo para la realización de tareas, éste debe contar con la siguiente descripción:

- Que los robots y los objetos cuenten con alguna geometría, ya que son los elementos esenciales y más importantes para la creación del ambiente o modelo.
- La forma física de los objetos tienen una tarea importante de representación y así poder manipularlos a nuestra conveniencia.
- En las características de los robots es necesario considerar los alcances y limitaciones tales como la manipulación de la articulación, así como la aceleración y capacidad sensorial.

En las diferentes áreas de la robótica, la geometría juega un papel muy importante, por ejemplo: para encontrar una trayectoria segura sin choque (colisión) para el robot, es claro que necesitamos crear un modelo que cuente con las formas y dimensiones totales de los objetos con los que se van a trabajar y sobre todo las restricciones mecánicas del robot.

Entonces, para alcanzar el objetivo de la robótica, es necesario crear una forma representativa de todo el conocimiento sobre el espacio o universo real, su geometría, leyes físicas y sus errores. La geometría es importante, pero la existencia de las fuerzas e incertidumbres hacen que los planteamientos de las acciones no sean claramente exactos. El espacio de trabajo de los robots es representado por una forma geométrica rectangular, si bien es cierto que raramente el espacio de trabajo tiene forma rectangular, también es verdad que raramente se ocupan todas las configuraciones posibles de las que pudiera alcanzar un robot.

Los obstáculos que se encuentren dentro del área de trabajo de los robots son modelados por formas geométricas, ya que el objetivo es el cálculo de la planeación de la trayectoria evitando obstáculo.

El área de trabajo puede definirse de dos maneras:

- Área de trabajo fijo
- Área de trabajo dinámico

En el área de trabajo fijo o estático, las posiciones de los objetos en el área son predeterminados.

En el caso del área de trabajo dinámico las posiciones de los objetos son aleatorias. El área de trabajo es representado, por rectángulos y los objetos por imágenes reales.

Algoritmo de navegación básica

Este algoritmo se diseñó para facilitar el reconocimiento de marcas y la construcción de los mapas. Su tarea principal es mantener al robot en movimiento constante de manera segura en un ambiente gráfico estático (sin alteraciones), evitando colisiones con obstáculos y con personas. La estrategia de navegación consiste en: evitar las colisiones entre el robot y los obstáculos, y mantener al robot a una distancia mínima de los objetos. Esta conducta es útil ya que la precisión de los sonares es maximizada en la proximidad de objetos detectables, donde el robot puede obtener información sobre el ambiente.

Programación Basic

Basic es la manera más rápida y fácil de crear aplicaciones. Proporciona un conjunto completo de herramientas que simplifican el desarrollo rápido de aplicaciones. Cuenta con reglas de manejo de gráfico propios en ensamblador que permite tomar control absoluto del proceso

Creación de la interfase del usuario

La interfase de usuario es quizás la parte más importante de una aplicación; ciertamente, es la más visible. Para los usuarios, la interfase es la aplicación; seguramente a ellos no les interesa el código que se ejecuta detrás. Independientemente del tiempo y el esfuerzo que haya empleado en la escritura y optimización del código, la facilidad de uso de su aplicación depende de la interfase.

Cuando se diseña una aplicación hay que tomar muchas decisiones relacionadas con la interfase. ¿Debe usar el estilo de documento único o el de documentos múltiples? ¿cuántos formularios diferentes necesitará? ¿qué comandos incluirá en los menús? ¿Usará barras de herramientas para duplicar funciones de los menús? ¿cómo va a hacer los cuadros de diálogo que interactúan con el usuario? ¿qué nivel de asistencia necesita proporcionar?

Antes de empezar a diseñar la interfase del usuario, se tuvo que pensar en el propósito de la aplicación. El diseño de una aplicación principal de uso constante debe ser diferente del de una que sólo se utiliza ocasionalmente, durante breves periodos de tiempo. Una aplicación cuyo propósito fundamental sea de presentar información tiene unos requisitos distintos que otra que se utilice para obtener información.

Programación de objetos

Los objetos son el fundamento de la programación en Basic. Los formularios y los controles son objetos. Las bases de datos son objetos. Hay objetos por todas partes.

Descripción del simulador

Como se dijo anteriormente, este simulador cuenta con un ambiente gráfico muy amigable para que el usuario final pueda manejarlo sin ningún problema. Las características con las que debe contar este simulador son: especificar la ubicación y la

distancia que tienen los robots hacia los objetos, también cuenta con un espacio en donde son presentadas las simulaciones llevadas a cabo al aplicar algoritmos de trabajo.

El simulador está formado por dos etapas: la primera etapa es que al iniciar el programa se debe especificar el tipo de ambiente en donde queramos trabajar, ya sea un almacén, oficina o el campo de una escuela.

En la segunda etapa se asignan las tres jerarquías de los robots que se autojerarquizan dependiendo de que objeto se encuentre cerca de cada robot, por ejemplo: se inicia el sistema, busca las distancias de los objetos entre los robots, el robot que se encuentre más cerca de x objeto toma el mando de liderazgo, el otro robot seguido del líder toma el mando de cargador, y el último se caracteriza como scout.

Cuando ya se tienen las jerarquías, el sistema automáticamente indica a los scout hacer una exploración de zona, ya que el espacio del simulador está diseñado por celdas para la mejor búsqueda de una trayectoria óptima, en donde los scout son esparcidos por cada una de las celdas.

Este programa cuenta con una caja de objetos como: sillas, bancos, sofás, cestos de basura, cajas, etc. Estos objetos son representados en el espacio de trabajo del simulador en donde los robots interactúan.

RESULTADOS Y DISCUSIÓN

Como lo hemos ido mencionando anteriormente, al aplicar los algoritmos de exploración, trazo de la trayectoria y seguimiento, los resultados de los algoritmos desarrollados son los siguientes:

En la Figura 8, los robots scout escanean el área de trabajo, después de que terminan de explorar el ambiente, cargan la información del mapa y lo convierten en una imagen digital para después cargarlo a la memoria del robot líder.

Primeramente presentamos la imagen original que fue construida dentro del mismo programa, el color de la gráfica de la pantalla nos muestra la originalidad de la imagen antes de que sea explorada por los robots scout.

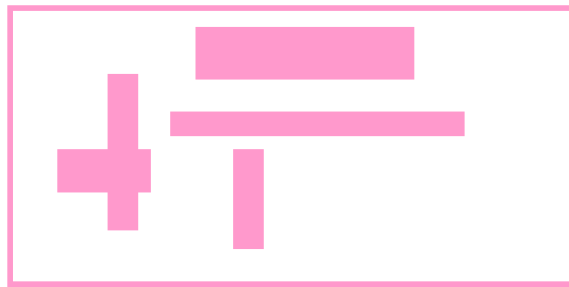


Figura 8. Representación de la imagen original

En la Figura 9, vemos como los scout empiezan a escasear el área de trabajo aplicando el algoritmo de exploración.

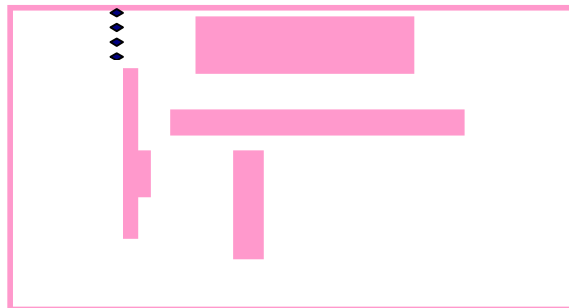


Figura 9. Scout explorando el ambiente de trabajo

Terminando el escaneo del espacio de trabajo por los robots scout, el sistema crea el mapa digital y muestra las posibles trayectorias que los robots van a seguir, después de tener los datos exactos del espacio de trabajo, se cargan a la memoria del robot. Siguiendo el ordenamiento de los algoritmos, el sistema inmediatamente crea el punto inicial y el final, ya teniendo estos dos puntos para trazar la trayectoria, el robot primeramente traza una línea recta de los puntos A y B, que es la más razonable a trazar, pero si se tienen objetos obstruyendo la trayectoria, el robot empieza a trazar otra línea recta de los puntos A y B evadiendo los obstáculos, tal es así que hace varios intentos por buscar la trayectoria más precisa hasta llegar a su punto final. En las

Figuras 10 y 11 vemos claramente los puntos trazados de la trayectoria del robot realizando 7 intentos en la cual evade los obstáculos hasta llegar a la meta.

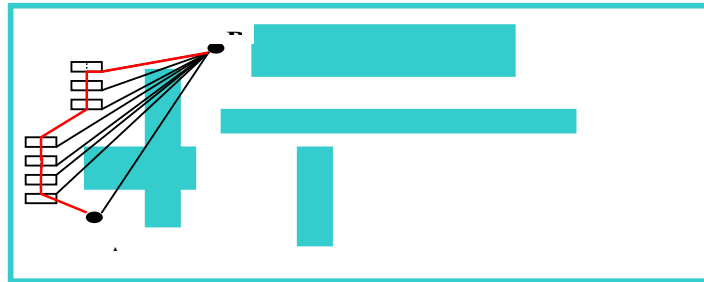


Figura 10. Trayectoria trazada por el robot

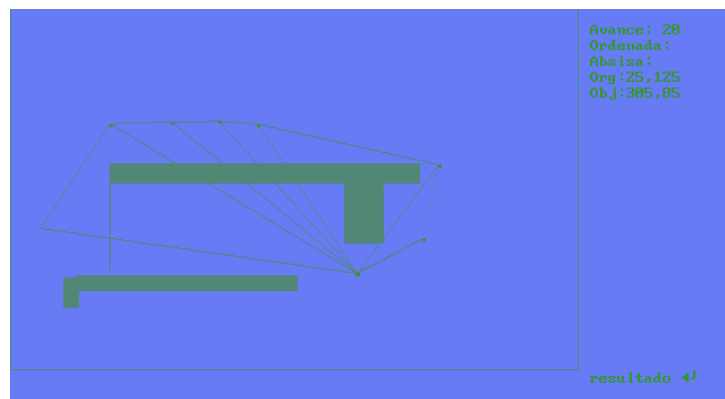


Figura 11. Posibles trayectorias trazadas por el robot

En el trazo de trayectoria de los robots, se pueden encontrar con un error de trazo y seguimiento de la línea creada, en lo cual se muestran algunos resultados en donde el robot realizó un primer intento del trazo de la trayectoria, ya que no pudo continuar su seguimiento debido a que no encontró su camino por la obstrucción de un gran obstáculo. Inmediatamente busca otro camino para llegar a su meta.

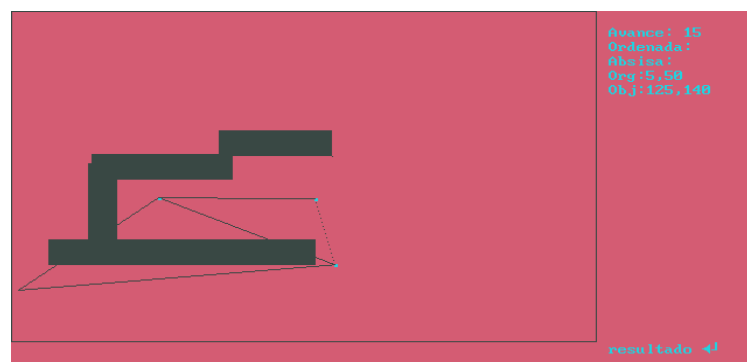


Figura 12. Seguimiento de la trayectoria encontrada después de dos intentos

Continuando con los análisis de los resultados del seguimiento de trayectorias, mostramos el Cuadro 5 donde se muestran claramente los resultados finales del robot al seguir diferentes trayectorias trazadas.

Cuadro 5. Resultados finales de las trayectorias

Origen		Destino		Pendiente	Sentido	Intentos	Croquis
X	Y	X	Y				
25	125	305	85	-0.14	sent 1	8	UAIM
185	225	185	25	1	sent 1	3	UAIM
25	45	265	125	0.33	sent 1	4	UAIM
5	61	159	201	0.9	sent 1	3	UAIM2
33	131	257	145	0.06	sent 1	6	UAIM2
5	50	125	140	0.75	sent 1	3	UAIM2
35	80	400	155	0.2	sent 1	11	PRU1
110	75	376	250	0.65	sent 1	4	COMPUTO

CONCLUSIONES

La planeación de trayectorias, en espacios representados en diferentes dimensiones, es una área ampliamente estudiada ya que demostramos que con poca programación y en lenguajes de bajo nivel podemos implementar una solución confiable al problema de trazo y seguimiento de trayectorias para uso en desplazamiento de robots, lo que significa una reducción de los costos inherentes ya que esto es uno de los principales problemas en la realización de trabajos robotizados. Cuando el líder tiene creado el mapa del lugar, va trazando una línea en el piso en donde el cargador va siguiendo la dirección que el líder le indica hasta llegar a su punto final.

Uno de los propósitos de este trabajo era comprobar que los algoritmos propuestos cumplieran con la función específica en la planeación de las trayectorias. Como se esperaba, al incrementar el número de obstáculos también aumentaba el número de robots, lo cual podría considerarse como una desventaja, ya que las alternativas a buscar una trayectoria aumentarían de la misma manera.

Por lo anterior, es importante decir, que aunque en esta investigación no se realizó un estudio más detallado para la creación de un mapa, se planteó una forma para crear el mapa, de esta manera el robot podría navegar y construir el mapa de su ambiente de trabajo utilizando el algoritmo aquí propuesto. Una vez que se conoce el mapa del

ambiente es posible marcar la ruta completa y en casos no previstos como: (objetos en movimientos y objetos que han sido cambiados de lugar) utilizar recursivamente estas reglas de navegación. La combinación de estos dos métodos de creación de mapas y marcado de la meta; nos permite reducir la posibilidad de que caiga en lugares engañosos como los son callejones, pasillos, etc. Estamos lejos de crear una verdadera inteligencia, pero tomando en cuenta que se han hecho avances importantes que permiten perfeccionar el funcionamiento de pocas máquinas como lo son los robots móviles, y que son de amplia y evidente aplicación.

LITERATURA CITADA

Ollero Baturone Aníbal

2001 **“Robótica, manipuladores y robots móviles”**. Editorial Alfaomega, España, 447 pp.

Hirzinger Gerd

2001 **“Institute of Robotics and Mechatronics”**. Wessling Germany. 210 pp.

Pasos Sierra Juan

1984 **“Inteligencia Artificial”**. Facultad de Informática, Madrid España, pp. 132.

Smith Reid G.

1981 **“Frameworks for Cooperation in Distributed Problem Solving”**. IEEE Transaction on System, Man and Cybernetics. Vol. SMC-11, No.1, January.

Sugar T. G. y V. Kumar

2001 **“Control of cooperating mobile manipulators”**. IEEE Trans. Robot. Automat. Vol. 18, p. 3373-3378.

Yamashita A., Arai, T. Jun Ota y Asama H.

2003 **“Motion planning of multiple mobile robots for Cooperative manipulation and transportation”**. Robotics and Automation, IEEE Transactionson, Volume: 19 Issue: 2: 223 -237.