

RECONOCIMIENTO DE ESPACIOS CON OPTIMIZACIÓN DE TRAYECTORIAS UTILIZANDO EL ROBOT "TURTLEBOT3 BURGER"

SPACE RECOGNITION WITH
PATH OPTIMIZATION USING THE
"TURTLEBOT3 BURGER" ROBOT.

RECONHECIMENTO DO
ESPAÇO COM
OPTIMIZAÇÃO DO
CAMINHO UTILIZANDO O
ROBOT "TURTLEBOT3
BURGER"

David Leonardo Martínez

Conecta Comunicaciones S.A.S,
david-leonardo.martinez-
herrera@veolia.com,

Oscar Hernan Rojas Barreto

Conecta Comunicaciones S.A.S
orojas.conecta.com@gmail.com,

William Fernando Bernal

Universidad Juan de Castellanos,
wbernal@jdc.edu.co

Fecha de Recepción: 10 de marzo de 2020

Fecha de aprobación: 12 de mayo de 2020

Resumen

Esta investigación es un aporte a la robótica móvil para la planeación de trayectorias de un robot autónomo. Para el caso de estudio se utilizará el robot TURTLEBOT3 BURGER. Con este dispositivo se realizaron pruebas de rendimiento dentro de una pista hexagonal. Cada prueba tiene un nivel de complejidad creciente, lo que permitió determinar una ejecución algorítmica con resultados que conducen hacia el desarrollo incremental en la programación del robot. Los resultados obtenidos están condicionados por los modos de conexión y programación hardware y software, mediante el uso de Robot Operating System (ROS), y la ejecución de una serie de pruebas que se centran en aplicar algoritmos "Dijkstra y A Star" que permiten optimizar las rutas con un robot y simularlas en "Rviz" como herramienta de visualización en 3D para aplicaciones ROS. Finalmente, la parte teórica relacionada con el trabajo de algoritmos y su programación se comprobó mediante esta plataforma robótica, lo que logra evidenciar que se cumple con los objetivos inicialmente propuestos, definidos, como el reconocimiento y planeación de una trayectoria sobre un terreno plano con obstáculos, desde un punto inicial hasta un punto final como punto objetivo o de llegada.

Palabras clave - Algoritmo, Navegación, Reconocimiento de obstáculos, Robótica, ROS.

Abstract

This research is a contribution to mobile robotics for the planning of trajectories of an autonomous robot. For the case study, the

TURTLEBOT3 BURGER robot will be used. With this device performance tests were carried out inside a hexagonal track. Each test has an increasing level of complexity, which allowed determining an algorithmic execution with results that lead to incremental development in robot programming. The results obtained are conditioned by the hardware and software connection and programming modes, through the use of the Robot Operating System (ROS), and the execution of a series of tests that focus on applying "Dijkstra and A Star" algorithms that allow optimizing routes with a robot and simulate them in "Rviz" as a 3D visualization tool for ROS applications. Finally, the theoretical part related to the work of algorithms and their programming was verified using this robotic platform, which manages to demonstrate that the initially proposed, defined objectives are met, such as the recognition and planning of a trajectory on flat terrain with obstacles, from a starting point to an end point as a target or arrival point.

Keywords- Algorithm, Navigation, Obstacle recognition, Robotics, ROS

Resumo

Esta investigação é uma contribuição para a robótica móvel para o planeamento da trajetória de um robô autónomo. Para o estudo do caso, será utilizado o robô TURTLEBOT3 BURGER. Com este dispositivo, foram efectuados testes de desempenho dentro de uma pista hexagonal. Cada teste tem um nível de complexidade crescente, o que permitiu determinar uma execução algorítmica com resultados que levaram ao desenvolvimento incremental na

programação do robô. Os resultados obtidos são condicionados pela programação de hardware e software e modos de ligação, através da utilização de Robot Operating System (ROS), e a execução de uma série de testes que se concentram na aplicação de algoritmos "Dijkstra e A Star" que permitem otimizar os caminhos com um robot e simulá-los em "Rviz" como uma ferramenta de visualização 3D para aplicações ROS. Finalmente, a parte teórica relacionada com o trabalho dos algoritmos e a sua programação foi verificada através desta plataforma robótica, que consegue provar que os objetivos inicialmente propostos, definidos como o reconhecimento e planeamento de uma trajetória sobre um terreno plano com obstáculos, desde um ponto inicial até um ponto final como alvo ou ponto de chegada, são cumpridos.

Palavras-chave - Algoritmo, Navegação, Reconhecimento de Obstáculos, Robótica, ROS.

Introducción

El presente trabajo tiene como principal objetivo aplicar algoritmos que permitan optimizar rutas para un robot de navegación libre en un terreno plano con obstáculos, desde un punto inicial y final predefinidos. La optimización de trayectorias es definida como el análisis del recorrido de unas rutas según la longitud de acceso y los obstáculos implícitos en esta, para finalmente seleccionar el camino más corto y más fácil de recorrer.

La característica principal de buscar optimizar una trayectoria es tomar una decisión entre

varias rutas que pueden ser probablemente tomadas, pero finalmente optar por seleccionar la ruta que contenga mayor linealidad y mayor espacio para transitar.

Con el estudio de optimización en las trayectorias se abre la puerta para hacer múltiples avances en las investigaciones sobre las decisiones autónomas que puede tomar un robot, desde su punto de partida el cual inicia la ejecución de una trayectoria hasta su punto de llegada (métrica). Durante este proceso el ambiente puede contener elementos de ruido, como objetos que deben estar considerados dentro del proceso de análisis de la navegación autónoma del robot, siempre que exista una ruta posible el robot estará realizando cálculos para planear la trayectoria hasta el punto final programado como de llegada.

Uno de los aspectos que permitirá la aplicación para optimizar una trayectoria es analizar los recorridos que podría tomar y los obstáculos presentes para así determinar la mejor ruta y que a su vez asegure un menor tiempo y mayor agilidad en los procesos.

En la actualidad se está aplicando esta técnica en automóviles eléctricos autónomos para que tomen decisiones dentro de una ciudad en la cual se encuentra una serie de obstáculos fijos y en movimiento, la cual transitaría analizando múltiples circunstancias que analizara para así asegurar no generar ningún incidente o accidente que ponga en riesgo a los transeúntes, esto implica considerar paralelamente los trayectos o las posibles rutas que el automóvil seleccionaría dentro de su recorrido en una ciudad.

En la actualidad existen compañías como Tesla

y Google que están implementando nuevas tecnologías aplicada a coches autónomos mediante el uso de sensores, radares y cámaras que les permiten desplazarse de manera segura hasta por 160 Km dentro de una ciudad con todas las complejidades que esto implica.

Antecedentes

En la aplicación del algoritmo Dijkstra para la optimización de trayectoria junto con el algoritmo local Dynamic Window Approach (DWA) para la generación de rutas, lo cual permite tener en cuenta las limitaciones dinámicas y cinemáticas del robot.

Una segunda etapa se realiza con el análisis de la detección y predicción de varios automóviles que estén presentes en el camino, pero debe tenerse en cuenta la selección de éste. En última instancia es conveniente mencionar que se permite tener la visión y localización de los puntos de llegada y de los automóviles sobre la pista que está dentro del área que se le ha definido. Durante la explicación que da este artículo se fundamenta el análisis del Algoritmo Dijkstra para la generación de trayectorias, se hace la detección de obstáculos y automóviles para que durante el tiempo de corrido no exista la posibilidad de colisiones [1].

Otro tipo de investigaciones a nivel internacional presentan trabajos en los cuales se estudia la planeación y seguimiento de trayectoria de robots por medio de la repetición de trayectorias que finalmente se conjugan como un solo algoritmo funcional. Con el presente trabajo se puede argumentar que la eficiencia en el seguimiento de la ruta es mayor y generada a partir de un ambiente

selectivo. La fundamentación para que se de esta efectividad, obedece a la utilización del algoritmo Dijkstra, el cual determina la ruta más corta utilizando como entrada un grafo y luego su posterior análisis computacional, generando cálculos internos que permiten encontrar el camino apropiado hasta alcanzar su destino. El software embebido en el robot fue el V-REP de código abierto y de fácil uso, allí se encuentra implícito el algoritmo Dijkstra, el cual es el centro de cálculo para la selección de la trayectoria óptima. [2]

En el contexto internacional la búsqueda de proyectos similares da como resultado que se encuentren trabajos en los cuales se definen la generación de trayectorias para un robot aéreo, éste transita por espacios seguros y detecta obstáculos, de igual forma su programación lo libra de colisión con cualquier tipo de objeto. Para esto se analizan matrices de tres dimensiones mediante MATLAB, aunque la complejidad es mayor por ser un vehículo aéreo que se mueve en tres coordenadas, el sistema no permite la pérdida del control y por lo tanto el vehículo aéreo no colisiona con objetos. Y de la misma forma que en otros proyectos encontrados la base fundamental es el algoritmo de control, por tanto, aquí se hace un análisis en la construcción vectorial que genera el desplazamiento y rotación, el modelo representado corresponde a un control no lineal, para este modelo se establece un valor mínimo entre el obstáculo y el robot para un radio definido, el modelamiento del Sistema de control no lineal se construyó por medio de un modelo espectral llamado Radau-pseudo. [3].

Un artículo local presenta un método para el establecimiento de trayectorias, mediante la generación del control de frecuencia que

relaciona varios elementos circuitales de un robot dentro una competición de bajo nivel. Un elemento novedoso alrededor de este prototipo tiene que ver con la predicción de movimientos mediante la aplicación de "B-spline" y el filtro "Kalman". "Ksdealma" es un filtro que puede predecir el próximo movimiento sobre el mismo tiempo de procesamiento, además con el algoritmo "B-spline" se controla la optimización de los puntos

de la trayectoria y se suavizan las trayectorias curvas esquivando los obstáculos. Este artículo permite determinar la curva de trayectoria de un robot desde un punto de partida hasta el punto final. [4]

Planeación de recorrido

Para la planeación del recorrido se realizan una serie de pasos para su funcionamiento.

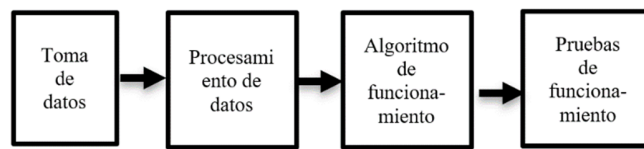


Fig.1. Secuencia de toma y procesamiento de datos.

Fuente: Autores.

El funcionamiento del robot" TURTLEBOT3 BURGER" se basa en un conjunto de bibliotecas de software y herramientas "ROS", y en la programación de la tarjeta "openCR" y la tarjeta "Raspberry Pi" que controla los motores y funcionamiento general del robot.

realizado utilizando el sensor Sensor de Distancia Laser (LDS) ubicado en la parte superior. Este sensor toma datos a su alrededor (360°), con una variación de 1° a una distancia máxima de 3.5m y mínima de 0.12m.

El robot se desplaza por una plataforma hexagonal, en la cual el robot empieza hacer un mapeo y adquisición de datos alrededor del terreno, el reconocimiento de espacios es

El escaneo de este sensor para procesos de navegación genera colores blancos para el espacio libre y negros para representar los obstáculos, ver Figura 2.

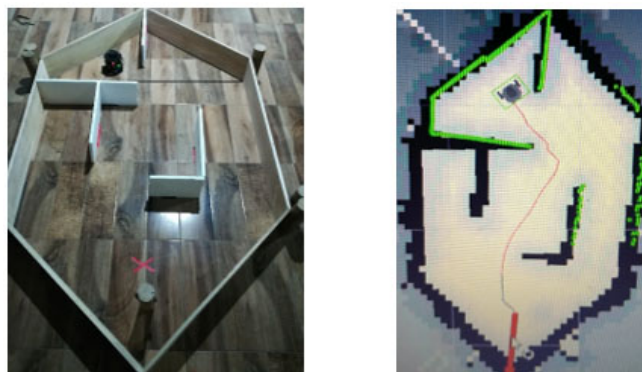


Fig2. Reconocimiento de espacios

Fuente: Autores.

En la figura 2. El robot se posiciona en una parte de la pista hexagonal, adquiriendo una ubicación dentro del espacio de tal manera que pueda navegar a través del terreno y formar el mapa total del sitio. Para la realización del mapa se efectúa un monitoreo

simultaneo de localización (SLAM).

Además, se plantearon diferentes obstáculos dentro de la pista hexagonal, donde se evidencio que el robot los podía reconocer para evadirlos y continuar su ruta.



Fig.3. Pruebas de funcionamiento.
Fuente: Autores

En la figura 3. Se ejemplifica como el robot puede llegar a tomar varias rutas y decidir cual tomar normalmente de forma curva, buscando llegar al destino que el usuario ha definido su

punto de llegada de forma previa. Finalmente, la ruta definida es la que permite mayor facilidad de acceso y menor tiempo de desplazamiento.



Fig. 4. Tres diferentes rutas para llegar a un destino.
Fuente: Autores

En la figura 4 se plantea un punto de llegada a la izquierda del terreno, en este caso el robot puede tomar la ruta de color negra, verde o roja. Al ejecutar el proceso, se evidencia que el

robot selecciona la ruta más corta para acceder a su destino representada por la línea roja

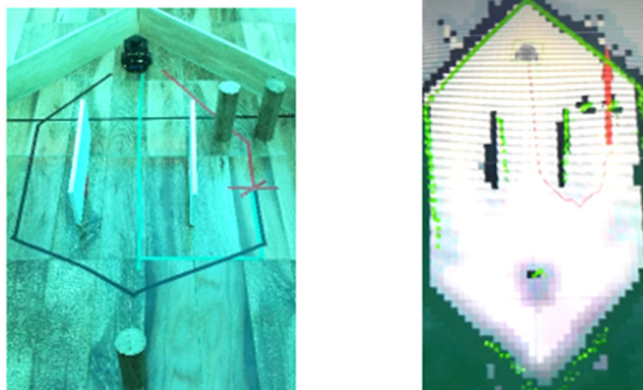


Fig. 5. Generación de ruta con obstáculo sobre su principal recorrido.
Fuente: Autores

En la figura 5 el robot toma la segunda ruta más corta para llegar a su punto final, ya que un obstáculo se interpone en su recorrido en la ruta establecida en el ejercicio anterior como la más corta. El recorrido que toma tiene una trayectoria curva para llegar a su punto final de destino. El robot empieza su desplazamiento en el momento en el cual el usuario define el punto final de la trayectoria o punto de llegada, este punto es definido previamente, lo cual permite que el robot aplique los algoritmos correspondientes de reconocimiento y trasado de ruta.

Para realizar la visualización del mapa se ejecuta un archivo llamado "Rviz", el cual permite que el robot pueda ser monitoreado y visualizado desde un computador.

A continuación, se presente el desarrollo de una prueba en la que son ubicados varios obstáculos dentro del hexágono. Aquí se pone a prueba el funcionamiento del algoritmo Dijkstra para seguir la ruta óptima dentro de un pequeño laberinto

118



Fig. 6. Resultado de la prueba 5 aplicando el algoritmo A-star(A*) junto con Dijkstra con obstáculos.
Fuente. Autores.

El robot puede llegar a tener dificultad de movimiento si los obstáculos están muy cerca, o si el espacio entre obstáculos es menor al diámetro del robot. En esta prueba el robot es colocado en un lugar donde pueda avanzar sin dificultad, cuando el espacio es reducido el robot empieza a avanzar y retroceder buscando girar, pero si gira y otro obstáculo impide su avance el algoritmo físico de funcionamiento determinara el detenimiento del avance del robot colocando los motores una velocidad lineal y angular igual a cero.

En la prueba el robot buscara seguir la ruta más

corta que permita llegar a la cruz de color rojo. La forma de la trayectoria es en zig-zag (Figura 7). Cuando la trayectoria generada es adecuada el robot buscara avanzar de forma lineal a medida que una velocidad angular es aplicada para permitir tomar una curva dentro del camino trazado. La mayoría de las veces seguir la ruta trazada dependerá de la posición, orientación y sentido del robot, si el robot se encuentra en una posición u orientación que haga difícil el avance tanto lineal como angular, el robot terminará deteniéndose.

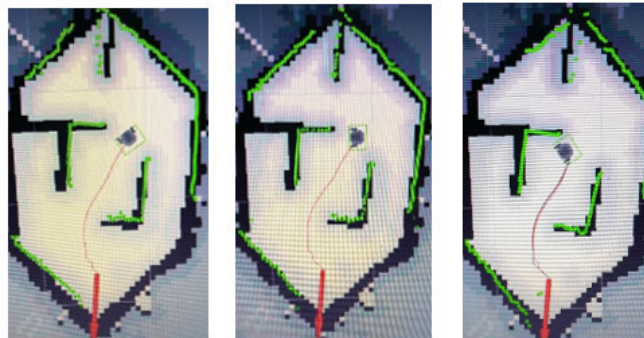


Fig. 7. El robot realiza una rotación para cambiar su sentido.

Fuente. Autores.

Para dar inicio a la prueba el robot es ubicado en una posición con velocidad lineal y angular de cero y una vez establecida la ruta más corta, el robot avanza con una velocidad lineal constante, si el robot tiene la opción de girar, la velocidad lineal disminuye hasta llegar a cero y la velocidad angular aumenta para darle la orientación correspondiente y que el robot encuentre la trayectoria trazada. Cuando la

trayectoria trazada es de tipo lineal, el robot avanza con una velocidad lineal constante y fija su ruta hasta llegar al objetivo o punto final. Con el cambio de giro de los motores, el robot puede ejecutar acciones de retroceso y esto permite minimizar el tiempo que se tarda para llegar a su punto de referencia. Ver Figura 8.

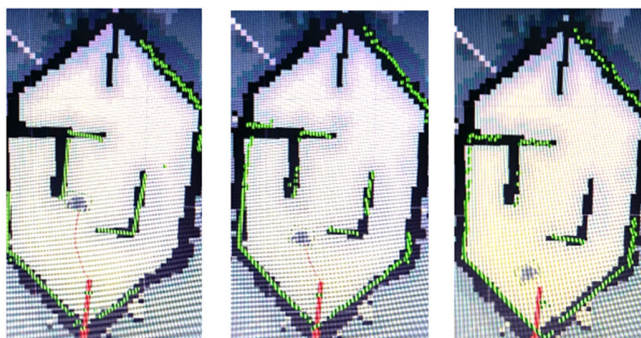


Fig. 8. El robot turtlebot3 llega a la cruz roja siguiendo la trayectoria más corta en zig-zag.
Fuente. Autor.

El robot busca seguir la trayectoria y a medida que avanza evidencia la selección de la línea roja como la ruta más corta desde la posición actual, punto donde se encuentra robot turtlebot3 Burger, hacia el objetivo o punto final. Ver Figura 8, este tipo de ejecuciones también han sido probadas en otras investigaciones, y siendo otros los algoritmos de trabajo, también buscan la mejor trayectoria con sistemas robóticos u otro tipo de dispositivos. [5],[6].

Comunicación

El sistema operativo para el computador y el robot es: "Ubuntu mate 16.4", el cual es basado en Linux y desde el terminal se ejecuta comandos para la ejecución de archivos.

Dentro del sistema operativo se programa en "ROS", el cual permite al sistema operativo leer mediante comandos los archivos que están asociados al funcionamiento general del robot "Turtlebot3 Burger".

Para la generación del movimiento y visualización el robot estará conectado a un computador tomando a este último como controlador y al robot como seguidor.

El robot y el computador estarán conectados a internet, por lo cual se asignan direcciones IP a cada uno de estos dos elementos con el fin de asegurar una correcta comunicación, ver Figura 9.

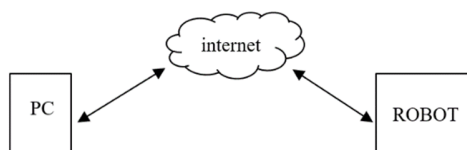


Fig. 9. Enlace de comunicación permanente.
Fuente: Autores.

El protocolo de comunicación es XML-RPC el cual se basa en una conexión a internet y se caracteriza por ser simple, y de trama corta, lo cual se adapta a las necesidades de esta investigación, y permite desde los repositorios

generar el mapa del terreno en tiempo real con la ayuda del programa "Rviz".

Conclusiones

Existen algoritmos genéricos o métodos de campos de potencial que calculan campos imaginarios de repulsión que emanan de los obstáculos y permiten definir una trayectoria hacia un objeto. Un ejemplo de este tipo de algoritmos es conocido como A*, por su fácil programación y buen desarrollo para la ejecución en tiempo real y puede ser comparado con otro tipo de algoritmos por su eficiencia en el cálculo del menos trayecto.

Uno de los problemas de optimización ampliamente estudiados consiste en la búsqueda de caminos mínimos desde un origen a un destino, es por esto que el uso de un algoritmo Dijkstra que toma como origen un grafo contribuye a reducir el tiempo de respuesta en la búsqueda de trayectorias.

Al aplicarse los algoritmos A-STAR y Dijkstra permitió planear la trayectoria, en el cual las aproximaciones dinámicas de Windows (DWA) permitió que el robot buscaba definir una ruta que le pudiera seguir la trayectoria planeada por el algoritmo.

Durante las pruebas de funcionamiento con el robot "Turtlebot3 Burger", para hacer efectivo el mapeo fue necesaria una permanente conexión a internet de tal forma que existiese una continua transmisión de datos al computador y poder tener actualizaciones cada 2 ms de la ubicación y mapeo del entorno.

Para que el robot tuviera mejor traslación y movilidad dentro del terreno, el robot ejecuta una operación de reconocimiento del entorno, mediante el uso del Código "Rviz", de tal forma

que puede finalmente realizar cálculos de trayectoria considerando los posibles obstáculos y determinando la ruta que le permite llegar a su destino.

Referencias

- [1] Mahdavi A. y M. Carvalho, «Optimal Trajectory and Schedule Planning for Autonomous Guided Vehicles in Flexible Manufacturing System,» IEEE, 2018.
- [2] Julius S. Fusic, P. Ramkumar y K. Hariharan, «Path planning of robot using modified Dijkstra Algorithm,» IEEE, 2018.
- [3] Kosari A. y M. M. Teshnizi , «Optimal Trajectory Design for Conflict Resolution,» Proceedings of the 6th RSI, International Conference on Robotics and Mechatronics (IcRoM 2018) IEEE, pp. 23-25, 2018.
- [4] Restrepo J. C., J. Villegas, A. Arias, S. Serna y C. madrigal, Trajectory Generation for a Robotic in a Robocup Test Scenery using Kalman filter and B-Spline curves, Antioquia: IEEE, 2012.
- [5] Alfonso-Mora, Margareth Lorena; Ávila-Barón, Adolfo, Cambios cinemáticos de la marcha en pacientes con artrosis de rodilla con diferentes descargas de peso, Revista Ciencias de la Salud, vol. 12, núm. 3, -, 2014, pp. 319-329, Universidad del Rosario Bogotá, Colombia.

- [6] Mobile Robot Using an Adaptable Fuzzy Control on a Digital Signal Processor, BA Ávila - 2006 - Tesis de Maestría Universidad Nacional de Colombia.
- [7] R. Martínez Ángel, J. Barrero Pérez y D. A. Tibaduiza Burgos, «Algoritmos de Planificación de Trayectorias para un Robot Móvil,» ResearchGate, Agosto 2006.
- [8] F. Benavides, Planificación de movimientos aplicada en robótica autónoma móvil, Montevideo, 2012.
- [9] V. Zambrano, Implementación de Algoritmos de Determinación de Rutas para el Robotino® de Festo, Quito, 2015.
- [10] MR. Reina, SLC Chía, DM Ávila, Automatización residencial un desafío profesional para el monitoreo de personas en condición de discapacidad visual, Ingenio Magno 10-2, pg 50-64.
- [11] 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, 1998