Revista de Ciencias Tecnológicas (RECIT). Universidad Autónoma de Baja California ISSN 2594-1925
Volumen 4 (4): 353-364. Octubre-Diciembre. https://doi.org/10.37636/recit.v44353364.

Interfaz gráfica para el control de seguimiento de trayectorias de un robot sanitizador en espacios controlados

Graphical interface for the control of the trajectory following of a sanitizing robot in controlled spaces

Carlos Guillermo Miguélez Machado, Ángel Israel Soto Marrufo, Israel Ulises Ponce Monarrez, Francesco García Luna

Instituto de Ingeniería y Tecnología IIT, Universidad Autónoma de Ciudad Juárez, Ciudad Juárez, Chihuahua, México.

Autor de correspondencia: Ángel Israel Soto Marrufo, Instituto de Ingeniería y Tecnología IIT, Universidad Autónoma de Ciudad Juárez, Ciudad Juárez, Chihuahua, México. E-mail: angel.soto@uacj.mx. ORCID: 0000-0001-6471-9127.

Recibido: 15 de Septiembre del 2021 **Aceptado:** 12 de Noviembre del 2021 **Publicado:** 24 de Noviembre del 2021

Resumen. *–En el presente trabajo se propone un Desarrollo de Interfaz Gráfica de Usuario para la interacción con un modelo de simulación tridimensional de un robot higienizador en una sala simulada para las pruebas de algoritmos de navegación implementados en el mismo. Al mismo tiempo, se implementan los comportamientos de exploración y planificación de rutas de desinfección. El modelo simulado del robot se basa en la Rueda Omnidireccional Nexus 4WOmni de la que también se propone el modelo cinemático. Además, se propone una interfaz gráfica de usuario para dar comandos básicos al robot simulado. Los resultados de la implementación se comprueban mediante la implementación de algoritmos de navegación al robot, también los comportamientos de exploración, planificación de rutas y seguimiento de trayectoria de desinfección e interacción de la interfaz gráfica con la simulación.*

Palabras clave: Interfaz gráfica de usuario; Robot desinfectante; Modelo simulado del robot; Planificación de trayectorias y seguimiento de trayectorias.

Abstract. *– In the present work, a Development Graphical User Interface for interaction with a tridimensional simulation model of a sanitizing robot in a simulated room is proposed for navigation algorithms tests implemented on it. At the same time, the behaviors of exploration and disinfection path planning are implemented. The simulated model of the robot is based on the Omnidirectional Nexus 4WOmni Wheel of which the kinematic model is also proposed. Additionally, a graphic user interface to give basic commands to the simulated robot is proposed. The results of the implementation are proved through the implementation of navigation algorithms to the robot, also the behaviors of exploration, path planning, and disinfection trajectory tracking, and interaction of the graphic interface with the simulation.*

Keywords: Graphic user interface; Sanitizing robot; Simulated model of the robot; Path planning and trajectory tracking.

1. Introducción

La pandemia generada por la enfermedad COVID-19 ha planteado la necesidad de hacer un mayor uso de la tecnología para disminuir el contacto de los humanos con la misma [1]. En este contexto, los robots se unen al personal médico en la primera línea, especialmente los robots móviles que utilizan lámparas de luz ultravioleta tipo C (UV-C) para desinfectar locales [2]–[4]. Específicamente, en el sistema de navegación del robot de desinfección con lámparas ultravioletas se necesitan algoritmos de planeación de rutas que garanticen el acercamiento del robot a los objetos a desinfectar [5], [6]. A su vez, el sistema debe ser capaz de gestionar las metas que administra dicho algoritmo al mismo tiempo que deja la opción al usuario de tele operar el robot. Finalmente, se han realizado estudios de la radiación recibida por unidad de área comprobando la eficacia de los sistemas autónomos móviles de desinfección. [7], [8].

Es necesario resaltar, que los proyectos de robótica, por lo general, requieren de alto presupuesto para implementar los algoritmos sobre un robot real. Por lo tanto, es bastante como un plantear resultados iniciales de un proyecto en simulación. Específicamente, en el campo de la navegación de robots móviles, Phunopas e Inoue[9] plantean los resultados de un sistema para corregir la medición de odometría de un robot móvil mediante el uso de la simulación. Por su parte, Nguyen et al. [10] proponen un sistema de navegación autónoma usando aprendizaje por reforzamiento y es probado en un modelo simulado de un robot omnidireccional de 4 ruedas. Los resultados del estudio anterior son presentados en un entorno de simulación en Gazebo y la arquitectura del sistema fue desplegada en Robotic Operating System (ROS). Cabe destacar, las dificultades que enfrentan los autores y los desarrolladores en el uso de los simuladores. Por ejemplo, consumo computacional del simulador, la fidelidad a la realidad del mismo y la construcción del modelo simulado del robot, constituyen algunas de estas dificultades. En este contexto, los autores Afsal et al. [11] realizaron un estudio de los principales problemas que presentan los motores de simulación actuales mediante encuestas online a desarrolladores y autores en el ámbito de la robótica.

En el presente trabajo se propone una interfaz gráfica de usuario para interactuar con el modelo de simulación de un robot de desinfección con luces ultravioletas. Adicionalmente, se explica la arquitectura de software que incluye el sistema de navegación. Por su parte, el sistema de generación de metas de desinfección es basado en el mapa construido por el robot mediante la exploración autónoma del local. Los resultados de la arquitectura son mostrados en un entorno simulado. Adicionalmente, se plantea el modelo cinemático y la dinámica de los actuadores de la plataforma móvil sobre la cual se intenta implementar el sistema. Es importante definir como limitante, que esta investigación no propone un control de bajo nivel para la generación de las velocidades del robot real, este aspecto se planea tratar en futura investigación. Adicionalmente, aunque se plantea el modelo del robot real a utilizar, no es contenido de esta investigación realizar el levantamiento instrumental de todo el hardware necesario para la puesta en marcha del robot real. El alcance de este trabajo llega hasta el diseño de un entorno simulado para el robot móvil con la tarea de desinfección, al igual que el diseño de una interfaz gráfica de usuario y las pruebas del algoritmo de desinfección sobre distintos mapas de entornos reales y virtuales.

El trabajo se estructura de la siguiente manera, en la sección 2 dos, marco teórico, se cubren los principales conceptos de interés como lo son

ISSN: 2594-1925

ROS, generación de trayectorias y el modelo cinemático y la dinámica de los actuadores. En la sección 3, Materiales y Métodos, se describe la interfaz diseñada para la interacción con el robot a nivel de usuario. En la sección 4, Resultados y Discusión, se plantean los resultados principales del trabajo. Por último, se realizan las conclusiones del mismo.

1. Marco Teórico

1.1 Plataforma experimental

Los resultados de la implementación del sistema se prueban en un entorno simulado con un modelo de simulación basado en el robot Nexus 4Womni ver figura 1 que posee 4 ruedas omnidireccionales.

Figura 1. Robot Nexus 4WOmni utilizado como base para el modelo de simulación del proyecto

1.2 Modelo cinemático

Primeramente, se define R como el radio del centro del robot hasta el centro de las ruedas, r como el radio de las ruedas $w_1;w_2;w_3;w_4$ como las velocidades angulares de cada una de las ruedas y las velocidades V_x; V_y; Θ como las velocidades en los ejes (x, y) y la velocidad angular del robot [9]. Luego, El modelo cinemático directo viene dado por la ecuación 1

$$\begin{bmatrix} V_x \\ V_y \\ \Theta \end{bmatrix} = \frac{r}{2} J \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} \quad (1)$$

Donde

$$J = \begin{bmatrix} -\sin\left(\pi+\frac{\pi}{4}\right) & -\sin\left(\pi+\frac{3\pi}{4}\right) & -\sin\left(\pi+\frac{5\pi}{4}\right) & -\sin\left(\pi+\frac{5\pi}{4}\right) \\ \cos\left(\pi+\frac{\pi}{4}\right) & \cos\left(\pi+\frac{3\pi}{4}\right) & \cos\left(\pi+\frac{5\pi}{4}\right) & \cos\left(\pi+\frac{5\pi}{4}\right) \\ R & R & R & R \end{bmatrix} \quad (2)$$

A su vez, el modelo cinemático inverso viene dado por la expresión 3

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \frac{r}{2} J^T \begin{bmatrix} V_x \\ V_y \\ \Theta \end{bmatrix} \quad (3)$$

1.3 Modelo dinámico de los Actuadores

Para analizar la dinámica de un motor se toman los siguientes parámetros

Tabla 1. Parámetros del robot y de los motores

Término	Descripción	Valor	Unidad
v	Voltaje de armadura		**V**
Ia	corriente del motor		**A**
Ra	resistencia de armadura	**1.9**	**Ω**
Ka	constante de torque	**13,4x10^{-3}**	**N-m($\frac{Nm}{A}$)**
Kb	constante de fuerza contra-electromotriz	**13,4x10^{-3}**	$\boldsymbol{\frac{V-s}{rad}}$
Jm	inercia del eje del motor	**5,7x10^{-3}**	**Kgm^2**
Fm(qm)	fricción en el eje del motor	**0.001**	**Nm**
r_e	relación de reducción de engranes	**64:1**	
φ	ángulo en la rueda i		**Rad**
$T_{\varphi i}$	torque en la rueda i		**Nm**

Las ecuaciones que describen el comportamiento del motor son

ISSN: 2594-1925

$$v = i_a R_a + \dot{q_{mi}} k_b \quad (4)$$

$$\dot{q}_{mi} = r_e \dot{chi}_i \quad (5)$$

$$\ddot{q_{mi}} = r_e \ddot{chi}_i \quad (6)$$

$$T_m = i_a k_a \quad (7)$$

$$J_m \ddot{q_{mi}} = T_m - f_m(\dot{q}_{mi}) - \frac{T_{chi}}{r_e} \quad (8)$$

1.4 Generación de trayectorias

La navegación autónoma requiere de sistemas que permitan generar y seguir la ruta hasta los puntos deseados. Por lo cual, se necesita un planeador global que sea capaz de tomar como entradas el mapa (2D o 3D) del mundo, la postura del robot y la meta deseada, para generar una trayectoria que guiara al robot de su posición y orientación actual a la deseada. Luego, se necesita de un planeador local que tome como entradas el plan global y la percepción que tiene el robot del ambiente, para generar las velocidades lineares y angulares que el mismo deberá alcanzar para seguir el plan global y evitar colisionar con obstáculos [12].

El estado del arte actual de los sistemas de navegación muestra familias extensas de algoritmos tanto de planeadores globales como locales. Las familias más denotadas de planeadores globales incluyen técnicas como grafos de visibilidad, exploración de árboles aleatorios [13], hojas de ruta probabilísticas y funciones de navegación [12]. Mientras que, entre los planeadores locales más destacados se encuentran en enfoque de ventanas dinámicas (DWA) [14], el enfoque de banda elástica (EBand) y el enfoque temporal de la (TEB) [12], [15].

Robotic Operating System (ROS) provee de capacidades de navegación en su colección de paquetes *move base* el cual utiliza planeadores globales y locales. Por consiguiente, corresponde al desarrollador de la aplicación elegir entre los distintos planeadores que ofrece ROS o implementar uno propio, posibilidad que se tiene por la flexibilidad que ofrece ROS como middleware. Entre los planeadores globales implementados en ROS están navfn y *global planner* [16]. A su vez, los planeadores locales de mayor relevancia implementados en ROS son los ya mencionados DWA, EBand y TEB [14], [17], [18].

En aras tomar la elección entre los navegadores más adecuados para la aplicación, Los autores Cybulski1 y Wegierska [15] compararon las implementaciones en ROS de los algoritmos DWA, EBand y Time Elastic Band (TEB) en simulación y en un robot real. Del estudio anterior, los autores concluyeron que DWA posee mayor repetibilidad. Mientras que, EBand posee mayor precisión al arribar a la meta. Por último, en términos de velocidad de acción TEB resultó ser el más rápido. Por otro lado, Pittner et al. [18] realizaron un estudio similar que compara los mismos algoritmos en términos del camino generado comparando su comportamiento en un entorno simulado. La anterior investigación arroja a TEB como el mejor, puesto que DWA tuvo problemas con los obstáculos dinámicos y EBand errores con los obstáculos estáticos. A su vez, Filotheou et al. [12] realizaron una comparación exhaustiva de planeadores teniendo en cuenta la calidad de la documentación existente, el soporte actual (en el año 2020), las facilidades de instalación, la modularidad y la consistencia en el rendimiento. Del estudio anterior, los autores reportaron que los algoritmos navfn como planeador global y teb local planner como planeador local eran los que mejor cumplan los parámetros antes mencionados.

Como muestran los estudios citados en los párrafos anteriores la combinación de navfn como planeador global y teb local planner como planeador local resulta ser la más adecuada para el sistema de navegación que se necesita en este trabajo.

ISSN: 2594-1925

2. Materiales y Métodos

La distribución de ROS empleada en el trabajo es ROS Melodic Morenia cuya fecha de lanzamiento fue el 23 de mayo del 2018 con fecha de caducidad en mayo del 2023. El mismo, se ejecuta sobre la distribución Ubuntu 18.04 Bionic. Adicionalmente, se utilizó Gazebo 9.0.0 para la simulación. La programación se realiza con la versión de Python 3.5 y el procesamiento de imágenes con OpenCV 3.0.

Para la prueba del sistema se construye un modelo simulado de la plataforma móvil y del sensor LiDAR en lenguaje URDF. Este lenguaje descriptivo se basa en xml y básicamente representa el robot mediante diferentes links que se unen entre sí mediante joints. En el esquema de la figura 2 se visualiza las conexiones entre los links del robot.

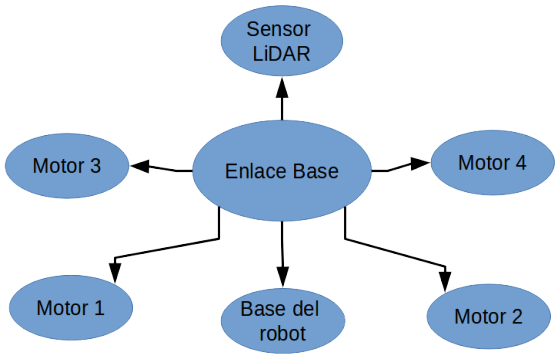

Figura 2. Relación entre los diferentes links del modelo simulado en Gazebo del Robot Nexus.

Así mismo, el sistema motor-rueda con rueda es descrito de manera independiente como se muestra en la figura 3. Lo anterior, debido a la complejidad y la cantidad de links que tiene una rueda omnidireccional.

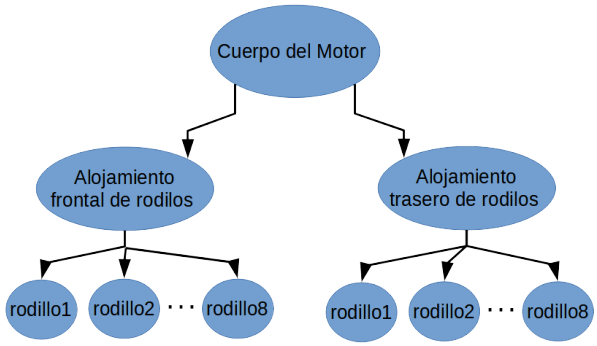

Figura 3. Relación entre los diferentes links del sistema motor rueda del robot en Gazebo.

El resultado de la construcción del modelo simulado del robot se muestra en la figura 4. El modelo del robot se construye con las dimensiones y el peso del robot Nexus mostrado en la figura (figura del robot Nexus). Además, el modelo del sensor LiDAR es construido con las dimensiones, el peso y las características del sensor RPLidar A2. Es decir, con una resolución de 1 grado y 12 metros de alcance.

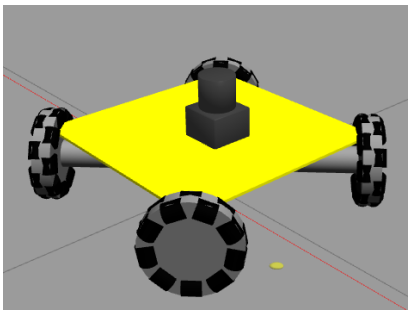

Figura 4. Modelo simulado en Gazebo del Robot Nexus.

2.1 Interfaz Gráfica de usuario

Se desarrolló una interfaz gráfica de usuario (GUI) básica con el objetivo de facilitar la interacción con el robot (figura 5). La GUI fue desplegada en un servidor web en la computadora local de trabajo.

ISSN: 2594-1925

Una de las funcionalidades básicas de la GUI es la comunicación con el robot, para lo cual, se utiliza un cuadro de entrada de texto en donde se introduce el IP y el puerto del robot (figura 5). El botón de conectar realiza la conexión con el robot, que en este caso es el modelo simulado que se encuentra en la misma máquina de trabajo, o sea, localhost:9090. Adicionalmente, se cuenta con un botón para solicitar al robot una exploración del entorno de trabajo y un botón para solicitar al robot que trace y siga la ruta de desinfección.

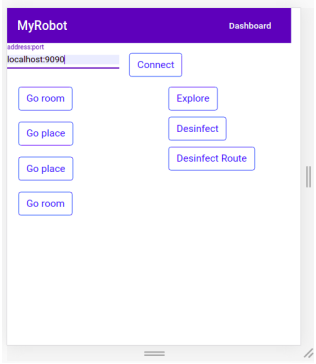

Figura 5. Interfaz gráfica de usuario pantalla frontal.

Además, se brinda al usuario la posibilidad de ingresar su propia meta introduciendo el grupo de datos coordenados (x,y,z,roll,pitch,yaw) como se muestra en la figura 6. En este mismo control, se cuenta con otra pestaña donde se pueden visualizar todas las metas creadas por el usuario, así como las opciones de eliminar o editar dichas metas.

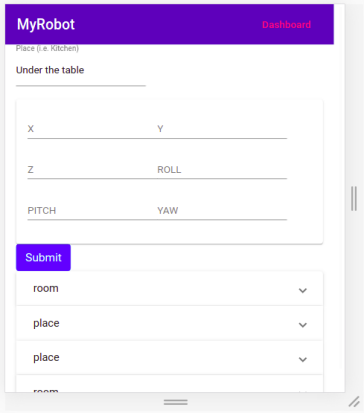

Figura 6. Interfaz gráfica de usuario pantalla de añadir metas manualmente.

3. Resultados y Discusión

3.1 Análisis de Interfaz Gráfica de usuario

La interfaz gráfica desarrollada con herramientas web de software abierto brinda la ventaja de ser escalable. Similarmente a lo propuesto, Sooryavanshi P et al. [19] desarrollan interfaz web mediante la implementación de Node.js en la computadora de su robot. La arquitectura propuesta por Sooryavanshi P contiene tanto la visual como la lógica de servidor lo que la hace más compacta pero más dependiente de la conexión con el dispositivo. Sin embargo, una arquitectura como la propuesta en el presente trabajo con *API Rest* en el robot desacoplada de una *Single Page Application* garantiza que el usuario no dependa completamente de la conexión con el robot para interactuar con la interfaz, lo anterior constituye un aporte. En términos de modularidad e integración, Cruz R et al. [20] proponen una arquitectura humano-robot desacoplada con posibilidad de integrar nuevos dispositivos. Sin embargo, utilizan un HMI el cual no es un elemento que le incorpora robustez, pero encarece el producto. En la propuesta realizada en este trabajo se pretende alcanzar la compatibilidad con cualquier dispositivo móvil sin necesidad de integrar un elemento industrial.

Por otra parte, en términos de usabilidad Ily M. y Magid E. [21] desarrollan una interfaz gráfica con un alto número de funcionalidades en comparación con las propuestas en este trabajo. Así mismo muestran mucha más información interna del robot. Sin embargo, el objetivo de los autores de este trabajo es el desarrollo de una interfaz para usuarios básicos con poco o ningún conocimiento de robótica, por lo cual, la sencillez de la interfaz propuesta.

ISSN: 2594-1925

Con respecto al uso de tecnologías avanzadas, la propuesta realizada se basa en tecnologías web modernas y con sostenibilidad. Aun así, autores como Miatliuk K. [22] proponen un diseño conceptual de control mental del dispositivo mediante la lectura de ondas cerebrales. Así mismo, Gego D. [23] propone algo similar pero mediante el seguimiento de la pupila del ojo del usuario. En términos de tecnología los trabajos de Miatliuk y Gego resultan más avanzados, aunque los mismos se encuentren en etapa conceptual. También, es válido cuestionar si el encarecimiento del producto y la aceptación por parte de un usuario final hacen factible el uso de dicha tecnología. De igual manera, otros trabajos enriquecen la interacción humano-robot mediante comandos de voz [24]–[26]. Dicha tecnología, ha tenido aceptación en los usuarios y resulta cómoda. Por lo anterior, se propone el enriquecimiento de la propuesta actual con módulos de comandos de voz.

3.2 Análisis del algoritmo de Desinfect_Router

Primeramente, se construyó una habitación en simulación (figura 7) para poner a prueba el funcionamiento del sistema de navegación del modelo simulado del robot. Mediante la interfaz de usuario se comanda la exploración de dicha habitación y se ordena al robot que valla a una meta específica entrada manualmente.

Seguidamente, Se realizó un algoritmo de autogeneración de metas mediante la identificación de contornos sobre el mapa 2D para generar una serie de metas que cubren la mayor parte de los objetos a ser desinfectados en el espacio de trabajo. La subrutina del algoritmo se activa una vez presionado el botón *Desinfect Route*. Finalmente, el seguimiento de trayectorias se realiza mediante la utilización del planeador local Time Elastic Band (TEB). A continuación, se presentan los resultados de dicho algoritmo, primero el mapa generado mediante una habitación simulada, luego en diversos mapas de habitaciones reales.

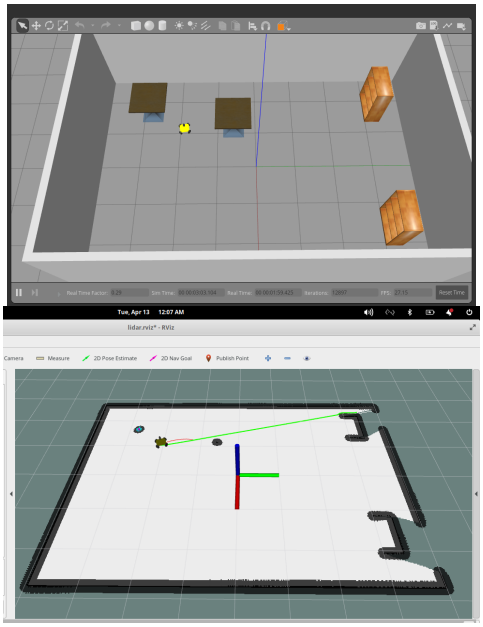

Figura 7. Habitación simulada construida (arriba visa en el simulador Gazebo, abajo vista en Rviz).

La figura 8 representa el algoritmo de autogeneración de metas sobre el mapa de la habitación simulada de la figura 7. La elección del algoritmo de planeamiento local TEB se realizó siguiendo la comparación realizada en [18]. El mismo se aplica de tal manera que se logra la autonomía en el seguimiento de ruta. Otras soluciones han sido el uso de inteligencia artificial en la navegación [27]. Los resultados en términos de eficiencia en la navegación resultaron similares. Pero, la complejidad de la implementación es mayor. Otras aproximaciones con respecto al control de bajo nivel ha sido la aplicación de técnicas de control avanzado [28][29][30]. Dichas soluciones son muy acertadas para un robot real y se propone la aplicación de las mismas en este trabajo en un futuro cercano.

Otra aproximación sería la de usar control tele-operado para mover el robot desde un punto remoto [31]. Sin embargo, esto obliga al

ISSN: 2594-1925

usuario a estar pendiente del proceso, y no resulta conveniente para la aplicación propuesta. Una aproximación más intensiva a la propuesta en este trabajo es la mezcla del mundo virtual simulado con la realidad para lograr ciertos patrones en el control de movimiento del robot [32]. Igualmente, la aproximación anterior requiere la atención de un operador. Por último, una aproximación muy simular a la propuesta de utilizar la simulación y la visualización en el desarrollo, es la desarrollada por Zea A. y Hanebeck D [33]. En la misma proponen un sistema similar a RViz pero en dispositivos móviles.

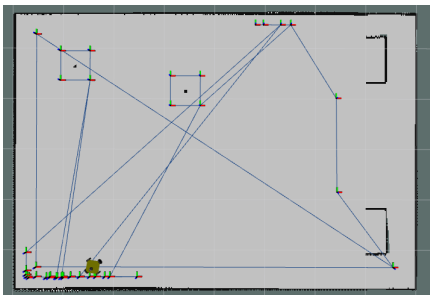

Figura 8. Metas generadas en la habitación simulada mediante el algoritmo de auto generación de metas.

3.3 Resultados sobre mapas construidos a partir de habitaciones reales

La figura 9 representa una habitación cuadrada (arriba sin obstáculos en el centro, abajo con obstáculos). Como se observa, el algoritmo dispone metas en el centro de la habitación cuando esta tienen posibles objetos a desinfectar.

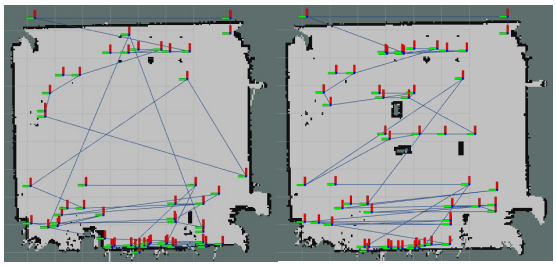

Figura 9. Relación entre los diferentes links del sistema motor rueda del robot en Gazebo.

La figura 10 representa habitaciones con una geometría más complicada. Se aprecia como el algoritmo coloca metas incluso en el exterior de la habitación. Aun así, dichas metas serán rechazadas por el planeador al no encontrar una ruta que conlleve a ellas.

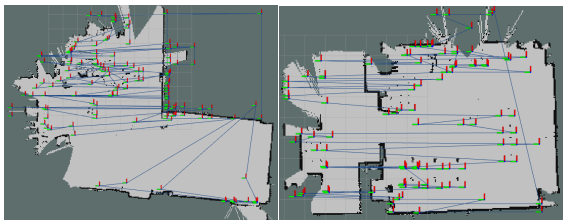

Figura 10. Metas generadas por el generador automático de metas en habitaciones de geometría compleja

La figura 11 arriba representa a el comportamiento del algoritmo sobre un mapa incompleto y abajo sobre un mapa con una geometría complicada con obstáculos en dispuestos en todo el mapa.

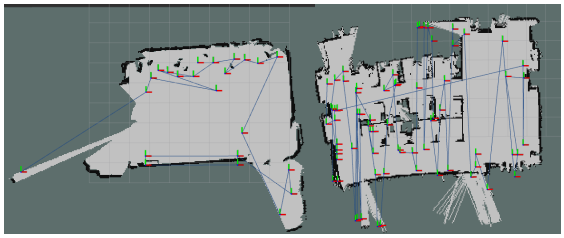

Figura 11. Metas generadas en una habitación no explorada completamente (arriba) y en una habitación de geometría complicada con obstáculos grandes en el centro

Similarmente al trabajo desarrollado con el planificador de rutas autónomo Desinfect Route, los autores (Ruan, K y, Zehao, Q, 2021) [34] plantean un robot para la desinfección autónoma que puede ser tele operado por el usuario y afirman implementar un método para la planificación de rutas autónoma. Sin embargo, no se especifica cómo se escogen las metas autónomamente. Por otra parte, los autores (Hu D. et al. 2020) [35] utilizan una cámara 3D y un algoritmo de segmentación de la nube de puntos para elegir autónomamente

ISSN: 2594-1925

las metas en el entorno para desinfectar, la desventaja de la solución propuesta es el requerimiento de sensores mucho más costosos que los necesarios en la solución propuesta en este trabajo.

4. Conclusiones

En el presente trabajo se desarrolló una interfaz de usuario gráfica para la interacción con un robot de desinfección ultravioleta. Además, se desarrolló un algoritmo de trazado de ruta de desinfección basado en los contornos del mapa previamente explorado. Finalmente, lo se puso a prueba el sistema en un entorno de simulación y visualización construido.

La interfaz de usuario es novedosa en esta área del conocimiento por la arquitectura de API REST que desacopla el robot de la interface y por la simpleza para el usuario. La misma se probó mediante el envío de comandos a un modelo del robot en una habitación simulada. Adicionalmente, se desarrolló el algoritmo de identificación de contornos para generar automáticamente las metas resulta una alternativa a los sistemas propuestos. Dicho algoritmo se probó en el mapa de la habitación simulada y en diversos mapas de habitaciones reales. Resultando el comportamiento del algoritmo mejor en habitaciones que tienden a ser cuadradas que en habitaciones con geometría más compleja. El algoritmo requiere de la utilización del sensor LiDAR que no resulta tan costoso como una cámara 3D. Finalmente, dicho algoritmo también se prueba que se puede aplicar a habitaciones parcialmente exploradas en cuyo caso el algoritmo solo coloca metas en terreno conocido.

El presente trabajo es implementarle en un robot real como el propuesto en la sección 2, requiriendo, además, de un sensor LiDAR para su correcta implementación. Sin embargo, para un mejor reconocimiento y desinfección del terreno se recomienda la mejora del sistema con una cámara 3D. La implementación del trabajo es aplicable en entornos hospitalarios y en entornos hogareños para la desinfección en general mediante radiación ultravioleta.

Como trabajo futuro a corto plazo se implementará el sistema en el robot real, para la interacción con la interfaz de usuario y la puesta en marcha en entornos reales.

5. Agradecimientos

Esta investigación ha sido financiada por la beca CONACyT para estudiantes y realizada como parte de la Maestría en Tecnología del Instituto de Ingeniería y Tecnología de la UACJ, Ciudad Juárez, Chihuahua.

6. Reconocimiento de autoría

Carlos Guillermo Miguélez Machado: Escritura, Borrador Original, Investigación, Metodología. *Ángel Israel Soto Marrufo:* Revisión, Supervisión, Análisis matemático, Metodología. *Israel Ulises Ponce Monarrez:* Revisión, Supervisión, Administración, Edición, *Francesco García Luna:* Revisión Supervisión, Administración, Edición.

Referencias

[1] D. Conte, S. Leamy, and T. Furukawa, "Design and Map-based Teleoperation of a Robot for Disinfection of COVID-19 in Complex Indoor Environments," 2020 IEEE Int. Symp. Safety, Secur. Rescue Robot. SSRR 2020, pp. 276-282, 2020. https://doi.org/10.1109/SSRR50563.2020.9292625.

[2] W. Chanprakon, P., Sae-Oung, T., Treebupachatsakul, T., Hannanta-Anan, P., Piyawattanametha, "An Ultra-violet sterilization robot for desinfection," 5th Int. Conf. Eng. Appl. Sci. Technol., vol. 5, pp. 44-47, 2019. https://doi.org/10.1109/ICEAST.2019.8802528

ISSN: 2594-1925

[3] A. Ray and H. Ray, "PSLB: Portable Sanitization Locomotive Bot," 2020 4th Int. Conf. Electron. Mater. Eng. Nano-Technology, IEMENTech 2020, 2020. https://doi.org/10.1109/IEMENTech51367.2020.9270096

[4] A. Vyshnavi, A. Manasa, C. Hamsika, and P. Shalini, "UV Disinfection Robot with Automatic Switching on Human Detection," EAI Endorsed Trans. Internet Things, vol. 6, no. 23, p. 166364, 2020. https://doi.org/10.4108/eai.25-9-2020.166364

[5] D. Hu, H. Zhong, S. Li, J. Tan, and Q. He, "Segmenting areas of potential contamination for adaptive robotic disinfection in built environments," Build. Environ., vol. 184, no. June, 2020. https://doi.org/10.1016/j.buildenv.2020.107226

[6] M. A. V. J. Muthugala, S. M. B. P. Samarakoon, M. M. Rayguru, B. Ramalingam, and M. R. Elara, "Wall-following behavior for a disinfection robot using type 1 and type 2 fuzzy logic systems," Sensors (Switzerland), vol. 20, no. 16, pp. 1-22, 2020. https://doi.org/10.3390/s20164445

[7] A. E. M. de Alba, M. B. Rubio, M. E. Morán-Diez, C. Bernabéu, R. Hermosa, and E. Monte, "Microbiological evaluation of the disinfecting potential of UV-C and UV-C plus ozone generating robots," Microorganisms, vol. 9, no. 1, pp. 1-12, 2021. https://doi.org/10.3390/microorganisms9010172

[8] L. Tiseni, D. Chiaradia, M. Gabardi, M. Solazzi, D. Leonardis, and A. Frisoli, "UV-C Mobile Robots with Optimized Path Planning: Algorithm Design and On-Field Measurements to Improve Surface Disinfection against SARS-CoV-2," IEEE Robot. Autom. Mag., vol. 28, no. 1, 2021. https://doi.org/10.1109/MRA.2020.3045069

[9] A. Phunopas and S. Inoue, "Motion Improvement of Four-Wheeled Omnidirectional Mobile Robots for Indoor Terrain," Proc. Int. Conf. Artif. Life Robot., vol. 22, no. 4, pp. 607-612, 2017. https://doi.org/10.5954/ICAROB.2017.GS3-1

[10] V. N. T. Thanh et al., "Autonomous navigation for omnidirectional robot based on deep reinforcement learning," Int. J. Mech. Eng. Robot. Res., vol. 9, no. 8, 2020. https://doi.org/10.18178/ijmerr.9.8.1134-1139

[11] A. Afzal, D. S. Katz, C. Le Goues, and C. S. Timperley, "A Study on the Challenges of Using Robotics Simulators for Testing," 2020. Accessed: Apr. 11, 2021. [Online]. Available: https://discourse.ros.org.

[12] A. Filotheou, E. Tsardoulias, A. Dimitriou, A. Symeonidis, and L. Petrou, "Quantitative and Qualitative Evaluation of ROS-Enabled Local and Global Planners in 2D Static Environments," J. Intell. Robot. Syst. Theory Appl., vol. 98, no. 3-4, pp. 567-601, 2020. https://doi.org/10.1007/s10846-019-01086-y

[13] I. Noreen, A. Khan, K. Asghar, and Z. Habib, "A path-planning performance comparison of RRT*-AB with MEA* in a 2-Dimensional Environment," Symmetry (Basel)., vol. 11, no. 7, 2019. https://doi.org/10.3390/sym11070945

[14] B. Tang, K. Hirota, J. Wang, Y. Dai, and Z. Jia, "An Improved Dynamic Window Approach for Intelligent Pedestrian Avoidance of Mobile Robot," 2020. Accessed: Apr. 06, 2021. [Online]. Available: https://isciia2020.bit.edu.cn/docs/20201114080545836985.pdf.

[15] B. Cybulski, A. Wegierska, and G. Granosik, "Accuracy comparison of navigation local planners on ROS-based mobile robot," in 12th International Workshop on Robot Motion and Control, RoMoCo 2019 - Workshop Proceedings, 2019, pp. 104-111. https://doi.org/10.1109/RoMoCo.2019.8787346

[16] A. A. Gelan, "AUTONOMOUS SEARCH AND RESCUE ROBOT USING ROS PLATFORM," NEAR EAST UNIVERSITY, 2019. http://docs.neu.edu.tr/library/6813958828.pdf

ISSN: 2594-1925

[17] H. Q. T. Ngo, V. N. Le, V. D. N. Thien, T. P. Nguyen, and H. Nguyen, "Develop the socially human-aware navigation system using dynamic window approach and optimize cost function for autonomous medical robot," Adv. Mech. Eng., vol. 12, no. 12, 2020. https://doi.org/10.1177/1687814020979430

[18] M. Pittner, M. Hiller, F. Particke, L. Patiño-Studencki, and J. Thielecke, "Systematic analysis of global and local planners for optimal trajectory planning," 2018. https://ieeexplore.ieee.org/abstract/document/8470582

[19] P. Sooryavanshi, S. Upganlawar, and A. Bhosle, "Implementation of node.js server on Raspberry pi to control a remote vehicle for defense use," in Proceedings of the International Conference on Intelligent Sustainable Systems, ICISS 2017, Jun. 2018, pp. 816-819. https://doi.org/10.1109/ISS1.2017.8389290

[20] R. Cruz, L. Garrote, A. Lopes, and U. J. Nunes, "Modular software architecture for human-robot interaction applied to the InterBot mobile robot," in 18th IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2018, Jun. 2018, pp. 17-23. https://doi.org/10.1109/ICARSC.2018.8374154

[21] M. Ily, L. Roman, and E. Magid, "Development of a graphical user interface for a crawler mobile robot servosila engineer," in Proceedings - International Conference on Developments in eSystems Engineering, DeSE, Feb. 2019, vol. 2018-Septe, pp. 192-197. https://doi.org/10.1109/DeSE.2018.00044

[22] K. Miatliuk, A. Nawrocka, K. Holewa, and V. Moulianitis, "Conceptual Design of BCI for Mobile Robot Control," https://doi.org/10.3390/app10072557

[23] D. Gego, C. Carreto, and L. Figueiredo, "Teleoperation of a mobile robot based on eye-gaze tracking," Jul. 2017. https://doi.org/10.23919/CISTI.2017.7975673

[24] Y. H. Chen and K. T. Song, "Voice control design of a mobile robot using shared-control approach," in 2017 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2017, Nov. 2017, vol. 2017-Janua, pp. 105-110. https://doi.org/10.1109/SMC.2017.8122586

[25] S. Sharan, T. Q. Nguyen, P. Nauth, and R. Araujo, "Implementation and testing of voice control in a mobile robot for navigation," in IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM, Jul. 2019, vol. 2019-July, pp. 145-150. https://doi.org/10.1109/AIM.2019.8868892

[26] S. Pleshkova, Z. Zahariev, and A. Bekiarski, "Development of Speech Recognition Algorithm and LabView Model for Voice Command Control of Mobille Robot Motio," Dec. 2018. https://doi.org/10.1109/HiTech.2018.8566257

[27] G. Bai, L. Liu, Y. Meng, W. Luo, Q. Gu, and J. Wang, "Path Tracking of Wheeled Mobile Robots Based on Dynamic Prediction Model," IEEE Access, vol. 7, pp. 39690-39701, 2019. https://doi.org/10.1109/ACCESS.2019.2903934

[28] S. Morales, J. Magallanes, C. Delgado, and R. Canahuire, "LQR Trajectory Tracking Control of an Omnidirectional Wheeled Mobile Robot," Dec. 2018. https://doi.org/10.1109/CCRA.2018.8588146

[29] T. Tongloy, S. Chuwongin, K. Jaksukam, C. Chousangsuntorn, and S. Boonsang, "Asynchronous deep reinforcement learning for the mobile robot navigation with supervised auxiliary tasks," in 2017 2nd International Conference on Robotics and Automation Engineering, ICRAE 2017, Feb. 2018, vol. 2017-Decem, pp. 68-72. https://doi.org/10.1109/ICRAE.2017.8291355

[30] A. L. Saleh, M. A. Hussain, and S. M. Klim, "Optimal Trajectory Tracking Control for a Wheeled Mobile Robot Using Fractional Order PID

ISSN: 2594-1925

Controller," J. Univ. Babylon Eng. Sci., vol. 26, no. 4, pp. 292-306, Feb. 2018. https://doi.org/10.29196/jubes.v26i4.1087

[31] D. Kiryanov and R. Lavrenov, "Remote Control Application for 'Servosila Engineer' on Android Mobile Devices," Proc. Int. Conf. Artif. Life Robot., vol. 25, no. February, pp. 440-443, 2020. https://doi.org/10.5954/ICAROB.2020.OS18-4

[32] M. Wu, S. L. Dai, and C. Yang, "Mixed reality enhanced user interactive path planning for omnidirectional mobile robot," Appl. Sci., vol. 10, no. 3, p. 1135, Feb. 2020. https://doi.org/10.3390/app10031135

[33] A. Zea and U. D. Hanebeck, "IVIZ: a ROS visualization app for mobile devices," arXiv, 2020. https://doi.org/10.1016/j.simpa.2021.100057

[34] K. Ruan, Z. Wu, and Q. Xu, "Smart cleaner: A new autonomous indoor disinfection robot for combating the covid-19 pandemic," Robotics, vol. 10, no. 3, 2021. https://doi.org/10.3390/robotics10030087

[35] D. Hu, H. Zhong, S. Li, J. Tan, and Q. He, "Segmenting areas of potential contamination for adaptive robotic disinfection in built environments," Build. Environ., vol. 184, 2020. https://doi.org/10.1016/j.buildenv.2020.107226

ISSN: 2594-1925