

Sistema Autónomo para el Monitoreo de Zonas No Permitidas Integrando ROS 1 y ROS 2 en el Robot Transbot

Autor 1: Miguel Angel Mendez
Miguel.mendez@uao.edu.co

Autor 2: Daniel Carlosama
Martínez
daniel.carlosama@uao.edu.co

Universidad Autónoma de Occidente, Facultad de Ingeniería, Programa de Ingeniería Mecatrónica (1)

Resumen

El presente trabajo aborda la implementación de un sistema autónomo para el monitoreo en tiempo real, integrando tecnologías de ROS 1 y ROS 2 mediante el uso de `ros_bridge`. El proyecto utiliza el robot móvil Transbot equipado con un sensor LIDAR, una cámara de profundidad y una unidad de procesamiento Raspberry Pi 4B, buscando superar las limitaciones tradicionales de los sistemas de vigilancia en entornos cerrados. Para la validación, se diseñaron simulaciones en Gazebo y RViz, complementadas con pruebas en hardware real.

La metodología incluyó el modelado CAD del robot, la configuración de algoritmos de navegación como AMCL y DWA Planner, y el uso de OpenCV para la clasificación de objetos en tiempo real. Durante las pruebas, el sistema alcanzó una precisión promedio de detección del 86% en ambientes complejos, con tiempos de respuesta menores a 1.5 segundos y una cobertura del área del 92%. Estos resultados demuestran que el sistema puede operar de forma confiable en entornos reales, garantizando la detección y clasificación efectiva de personas y obstáculos.

En conclusión, el sistema desarrollado representa un avance significativo en la implementación de tecnologías autónomas para la vigilancia, combinando sensores y algoritmos de navegación de última generación. Este enfoque puede adaptarse para aplicaciones más avanzadas en monitoreo y seguridad.

Palabras Clave: Robot Transbot, cámara de profundidad OBBEC, sensor LIDAR, Raspberry Pi 4, modelo URDF, SolidWorks, simulación, RViz, Gazebo, detección de obstáculos, navegación, ROS, control de cámara, control de LIDAR.

Abstract

This project presents the implementation of an autonomous monitoring system using ROS 1 and ROS 2 technologies through the integration of `ros_bridge`. The system leverages the Transbot mobile robot equipped with a LIDAR sensor, a depth camera, and a Raspberry Pi 4B processor, addressing the limitations of traditional surveillance systems in confined environments. Simulations in Gazebo and RViz, combined with real hardware tests, were conducted to validate the system's performance.

The methodology involved CAD modeling of the robot, the configuration of navigation algorithms such as AMCL and DWA Planner, and the use of OpenCV for real-time object classification. The results indicate an average detection accuracy of 86% in complex environments, response times below 1.5 seconds, and an area coverage of 92%. These findings demonstrate the system's reliability for real-world operations, ensuring effective detection and classification of people and obstacles.

In conclusion, the developed system represents a significant step forward in implementing autonomous surveillance technologies, combining state-of-the-art sensors and navigation algorithms. This approach can be adapted to advanced applications in monitoring and security.

Keywords: Transbot robot, OBBEC depth camera, LIDAR sensor, Raspberry Pi 4, URDF model, SolidWorks, simulation, RViz, Gazebo, obstacle detection, navigation, ROS, camera control, LIDAR control.

1. INTRODUCCIÓN

En los últimos años, los avances en robótica y tecnologías de automatización han permitido desarrollar sistemas autónomos capaces de abordar problemas complejos como la vigilancia en tiempo real de entornos cerrados. Sin embargo, muchos de estos sistemas enfrentan limitaciones técnicas significativas, especialmente en la integración y el manejo

simultáneo de sensores avanzados y algoritmos de navegación robustos. El proyecto desarrollado tiene como objetivo abordar estas limitaciones, enfocándose en la implementación de un sistema de monitoreo autónomo utilizando el robot móvil Transbot con tecnologías ROS 1 y ROS 2.

El problema principal identificado es la necesidad de un sistema eficiente que permita la vigilancia autónoma y en tiempo real, combinando capacidades de navegación, detección y clasificación de objetos. Aunque existen sistemas comerciales de monitoreo basados en cámaras o drones, presentan deficiencias en la precisión de detección y la adaptabilidad a entornos dinámicos. Adicionalmente, la interoperabilidad entre diferentes versiones de ROS sigue siendo un desafío técnico significativo, limitando la funcionalidad de muchos proyectos.

La metodología seleccionada para este trabajo combina herramientas modernas de simulación y control, como Gazebo, RViz, AMCL y DWA Planner, con la integración de sensores avanzados como LIDAR y cámaras de profundidad. Esta elección se justifica por su capacidad de proporcionar soluciones robustas y personalizables, lo cual ha sido validado en estudios previos. Sin embargo, trabajos anteriores presentan limitaciones relacionadas con la precisión en entornos complejos y la falta de integración efectiva entre hardware y software.

Este proyecto busca llenar esa brecha, implementando un sistema que no solo sea capaz de navegar de manera autónoma, sino también de realizar tareas de detección y clasificación de objetos con una alta precisión. A través de este trabajo, se propone una solución integral que optimiza el monitoreo en tiempo real de zonas no permitidas, mejorando la eficiencia y reduciendo la intervención humana.

Los objetivos de este trabajo son:

1. Diseñar e implementar un sistema autónomo de monitoreo utilizando el robot Transbot.
2. Integrar sensores avanzados y algoritmos de navegación para optimizar la precisión en la detección de personas
3. Validar el sistema mediante simulaciones y pruebas en hardware real.

El artículo está estructurado de la siguiente manera: la sección de Metodología describe el diseño del sistema, la integración de herramientas y la configuración de los sensores. La sección de Resultados presenta los hallazgos obtenidos a través de simulaciones y pruebas experimentales. Finalmente, la Discusión y las Conclusiones analizan los logros alcanzados, las limitaciones identificadas y las posibles aplicaciones futuras del sistema.

2. ANTECEDENTES Y REVISION DE LITERATURA

El uso de sistemas SLAM (Simultaneous Localization and Mapping) ha sido clave en la robótica moderna, especialmente en la combinación de LIDAR y cámaras para mejorar la precisión del mapeo tridimensional. Esta técnica ha permitido que los robots autónomos puedan moverse y mapear entornos desconocidos de forma simultánea, brindando una solución eficiente para sistemas de vigilancia en tiempo real. Además, los avances en redes neuronales y algoritmos de aprendizaje automático han mejorado la capacidad de los robots para detectar objetos y personas en tiempo real, proporcionando una respuesta más rápida y precisa frente a posibles intrusiones en entornos cerrados.

Estudios recientes han demostrado que la combinación de sensores LIDAR y cámaras es fundamental para mejorar la precisión en la detección de intrusos. Por ejemplo, Wang y colaboradores desarrollaron un sistema de vigilancia autónomo que utiliza estos sensores para captar datos del entorno en tiempo real, permitiendo una cobertura más detallada y exacta de las áreas vigiladas. Este enfoque es similar al que utilizamos en nuestro proyecto, donde los sensores de LIDAR y las cámaras se integran para optimizar la precisión en la detección de movimientos. Además, el uso de tecnologías IoT ha sido clave en la mejora de estos sistemas. Zhang y Lee demostraron que la transmisión de datos a través de la IoT no solo facilita la comunicación en tiempo real con el usuario, sino que también incrementa la capacidad de monitoreo remoto. Este componente de IoT es particularmente relevante en nuestro proyecto, ya que permite la transmisión en tiempo real de los datos capturados por el robot a una aplicación móvil.

Un análisis crítico de otros sistemas de vigilancia autónoma revela ciertas limitaciones en proyectos que dependen

únicamente de cámaras para la detección de movimientos. Li y Wu evaluaron estos sistemas y concluyeron que, si bien las cámaras proporcionan una buena resolución visual, la falta de integración con sensores adicionales, como LIDAR, puede generar imprecisiones en entornos complejos. En nuestro proyecto, esta limitación se aborda al integrar tanto cámaras como sensores LIDAR, lo que permite una mayor precisión en la detección y mapeo del entorno, superando las dificultades que presentan otros enfoques.

Además, la estabilidad de los sistemas de vigilancia es un factor crucial cuando se comparan robots terrestres con drones. Aunque los drones proporcionan mayor flexibilidad en áreas grandes, Brown y Taylor señalaron que los robots terrestres, como el que utilizamos en nuestro proyecto, son más estables y sostenibles para la vigilancia continua en interiores, ya que no dependen del espacio aéreo y su operación no se ve interrumpida por restricciones de vuelo o autonomía de la batería. Nuestro proyecto se beneficia de esta estabilidad, permitiendo una vigilancia constante y precisa en áreas designadas, con la ventaja añadida de la transmisión en tiempo real a una plataforma móvil.

Los robots de seguridad han ganado una gran atención en los últimos años, especialmente para la vigilancia en el hogar y edificios. Estos robots están diseñados para reducir la intervención humana y mejorar la eficiencia en el monitoreo de áreas. Un diseño notable es el robot cuadrúpedo de vigilancia desarrollado por Owocye et al. [1], que demuestra la creciente relevancia de los sistemas robóticos en la seguridad del hogar. Este robot, con su diseño de cuatro patas, asegura estabilidad y movilidad en diversos entornos, mostrando un avance en los sistemas de patrullaje autónomo.

Además de las soluciones terrestres, otros sistemas como la vigilancia con drones impulsados por IA están emergiendo como alternativas para mejorar la seguridad en el hogar [2]. Estos sistemas ofrecen capacidades de vigilancia aérea, que complementan las limitaciones de los robots terrestres al cubrir áreas más grandes y de difícil acceso. Por su parte, Lee y Shih han explorado los robots autónomos de seguridad para interiores, enfatizando la importancia de la monitorización en tiempo real en entornos cerrados como hogares y oficinas [3].

La detección robusta de intrusos es una característica crucial en los robots de seguridad. Al-Khali et al. [4] propusieron un sistema de sensores acústicos integrado con IoT para mejorar la detección de intrusos en tiempo real. Su enfoque destaca cómo los robots de seguridad pueden aprovechar el IoT para mejorar la capacidad de respuesta y la gestión de datos en la detección de amenazas. De manera similar, Bhavana et al. [7] se enfocaron en robots de patrullaje regulado para la seguridad de apartamentos, demostrando aún más la importancia de la patrulla localizada en espacios confinados.

Otra tendencia emergente es la integración de sistemas de visión con LiDAR para una detección más precisa de intrusos, como se destaca en el trabajo de Zhou et al. [9]. Esta combinación de tecnologías permite que los robots no solo naveguen eficientemente por los espacios, sino que también identifiquen amenazas potenciales con mayor precisión. Singh y Sinha [10] también han realizado importantes contribuciones al desarrollar sistemas de reconocimiento de actividad humana basados en video, mejorando las capacidades de vigilancia a través de procesamiento avanzado de imágenes.

Varios estudios también enfatizan el uso de soluciones de bajo costo para la vigilancia en hogares y pequeñas oficinas. Por ejemplo, Ravichandran et al. [8] diseñaron un robot de vigilancia inteligente de bajo costo, destacando el potencial para la adopción masiva de estas tecnologías. El sistema equilibra efectivamente el costo y la funcionalidad, lo que lo convierte en una opción viable para los consumidores con presupuestos ajustados.

En el contexto de la vigilancia robótica, Transbot, un robot móvil con ruedas y orugas, presenta una oportunidad para explorar estos avances en el campo de la seguridad del hogar y la oficina. La capacidad del sistema para navegar en terrenos complejos e integrar varios sistemas de sensores se alinea con las tendencias observadas en otros diseños de robots de seguridad. Mientras que estudios como el de Wang et al. [12] enfatizan la importancia de usar LiDAR y cámaras para la detección de intrusos, aún queda margen de mejora en el equilibrio entre costo, eficiencia y usabilidad en robots como Transbot.

Además, los drones han sido considerados en estudios como el de Brown y Taylor [15], que compararon robots terrestres con drones para la vigilancia, ofreciendo ideas sobre cómo Transbot podría complementar o mejorar estas tecnologías existentes. Li y Wu [14] discuten los desafíos de los robots autónomos de seguridad basados en visión, destacando las limitaciones y direcciones de investigación futura para mejorar la precisión de la detección y la eficiencia operativa en los sistemas de vigilancia.

Para el desarrollo del modelo URDF del robot, se siguieron tutoriales prácticos disponibles en línea que facilitaron la simulación y modelado del sistema. En particular, se utilizaron los materiales proporcionados por el canal de YouTube

"YouBotica Tutoriales", que incluyeron la lista de reproducción *Simulación de Robots en ROS2 con RViz2 y Gazebo*. Estos recursos fueron fundamentales para aprender a configurar los modelos y a simular el comportamiento del robot en RViz2 y Gazebo.

Las técnicas aprendidas incluyeron la definición de enlaces, joints, y la configuración de parámetros físicos del robot, logrando una representación precisa del modelo en simuladores. [16]

MATERIALES Y MÉTODOS/METODOLOGÍA

3.1. Tipo de investigación

El proyecto se desarrolló como una investigación aplicada en el ámbito de la robótica autónoma, específicamente orientada al diseño y validación de sistemas de monitoreo en tiempo real. Este trabajo pertenece a la línea de investigación de integración de sensores avanzados y algoritmos de navegación, validando tanto en simulación como en hardware físico.

3.2 Materiales

3.2.1. Hardware

3.2.1.1. Transbot-RPi4B: Chasis móvil basado en una estructura de aleación de aluminio anodizado, diseñado para aplicaciones educativas y experimentales. Este robot utiliza una batería de litio de 12.6 V y 4400 mAh, proporcionando hasta 2 horas de autonomía. La tracción es gestionada por dos motores metálicos con encoders, garantizando precisión en el movimiento y medición de velocidad.

Especificaciones principales:

1. Microprocesador: Broadcom BCM2711 64-bit Quad-Core a 1.5 GHz con GPU VideoCore VI.
2. Sistema Operativo: Ubuntu 18.04 LTS + ROS Melodic.
3. Conectividad: Red Wi-Fi (LAN/AP) y compatibilidad con controladores inalámbricos PS2.
4. Protección: Protección contra sobrecorriente para evitar daños eléctricos.



Figura 1. Yahboom Transbot-RPi4B

3.2.1.2. Sensor LIDAR (SLAM A1M8): Capaz de detectar obstáculos en un rango de hasta 12 metros con una frecuencia de escaneo de 10 Hz. Este sensor genera mapas tridimensionales de alta precisión, fundamentales para navegación autónoma y localización.



Figura 2. Sensor LIDAR

3.2.1.3. Cámara de Profundidad OBBEC Astra Pro Plus: Tecnología de luz estructurada monocular con rango de detección de 0.6 a 8 metros. Proporciona imágenes de alta resolución en RGB con un ángulo de visión horizontal de 58.4° y vertical de 45.8°. Ideal para detección y clasificación de objetos mediante inteligencia artificial.

Especificaciones clave:

1. Resolución máxima: 1920x1080 @ 30 fps.
2. Campo de visión: Horizontal 58.4°, Vertical 45.8°.
3. Procesador: MX6000 con soporte UVC(RGB).
4. Consumo energético: <2.5W.



Figura 3. Cámara OBBEC

3.2.1.4. Raspberry Pi 4B: Microcontrolador con una potencia de cálculo de 200 GFLOPs, adecuado para gestionar nodos ROS y algoritmos avanzados. Ofrece compatibilidad con Python, facilitando el desarrollo de scripts personalizados para control y procesamiento.



Figura 3. Raspberry pi 4

3.2.1.5. Joystick inalámbrico PS2: Control remoto utilizado durante las pruebas de tele operación y calibración inicial del robot, permitiendo un manejo intuitivo de velocidad y dirección.

3.2.1.6. Batería de Litio: Proporciona una autonomía promedio de 2 horas durante operaciones continuas, asegurando suficiente energía para tareas de navegación, detección y monitoreo.

3.2.2. Software

3.2.2.1. ROS 1 y ROS 2: Plataformas robóticas empleadas para la gestión de nodos y la planificación de trayectorias. ROS 1 se utilizó principalmente para el control de motores, mientras que ROS 2 permitió implementar algoritmos de percepción y navegación.

3.2.2.2. ros1_bridge: Herramienta que permite la comunicación entre ROS 1 y ROS 2, asegurando transferencia de datos entre tópicos clave como /cmd_vel y /laser_scan.

3.2.2.3. Simuladores RViz y Gazebo

1. **RViz:** Visualización en tiempo real de datos de sensores y mapas generados, además de herramientas para configurar objetivos de navegación.

2. **Gazebo:** Simulación física avanzada para probar algoritmos de control y planificación en entornos virtuales.
- 3.2.2.4. **OpenCV:** Biblioteca utilizada para el procesamiento de imágenes capturadas por la cámara OBREC, facilitando la clasificación de objetos y personas.

3.3.3 Recursos Complementarios:

1. **SDKs de Yahboom:** Proporcionados en el repositorio oficial, incluyen guías y herramientas específicas para inicializar y programar el Transbot.
2. **Tutoriales de YouTube (*YouBotica Tutoriales*):** Material práctico que facilitó la configuración de modelos URDF y simulaciones en RViz y Gazebo.
3. **Librería Transbot:** Librería que permite la comunicación entre actuadores y sensores con la placa.

3.3. Métodos

3.3.1 Diseño del Modelo CAD: El diseño del robot comenzó con un modelo tridimensional en SolidWorks, que incluyó los componentes principales, como la estructura, los soportes para los sensores y la cámara. Este modelo fue exportado como un archivo URDF para su integración en simuladores como Gazebo y RViz.

3.3.2 Simulación en Gazebo: El modelo URDF fue configurado en Gazebo para validar la interacción dinámica del robot con el entorno, probar algoritmos de navegación, como el DWA Planner y optimizar parámetros como la fricción y las velocidades.

3.3.3 Configuración en RViz: RViz se utilizó para visualizar datos del LIDAR y mapas generados en tiempo real, configurar puntos de patrullaje en mapas predefinidos y validar la alineación de los marcos de referencia (frames) entre el robot y los sensores.

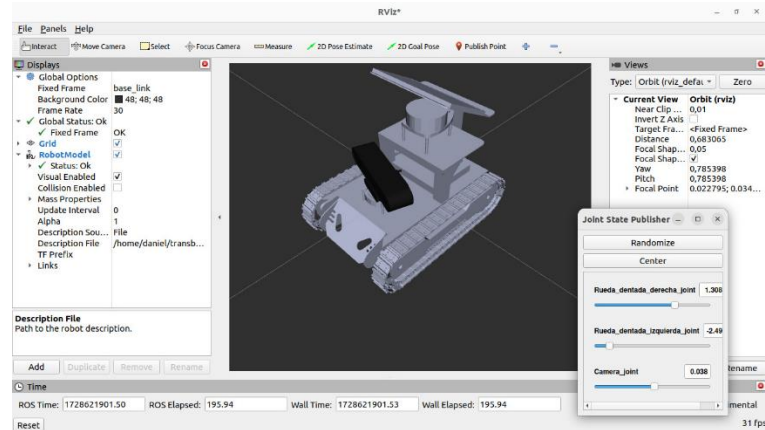


Figura 4. Simulación del transbot en rviz

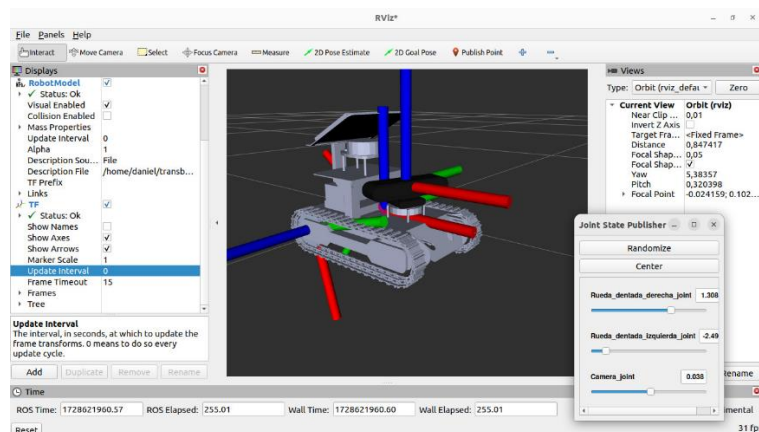


Figura 5. Vista de ejes

3.3.4 Navegación Autónoma

Se implementó mediante:

1. **AMCL (Adaptive Monte Carlo Localization):** Usado para localizar al robot en mapas previamente generados.
2. **DWA Planner:** Algoritmo encargado de calcular trayectorias óptimas en tiempo real.
3. **Mapas YAML:** Configurados para navegación:
4. **Global Cost Map:** Define obstáculos estáticos.
5. **Local Map:** Integra datos en tiempo real del LIDAR.

3.3.5 Teleoperación y Control Manual Un joystick inalámbrico configurado con el archivo transbot_joy.launch permitió la operación manual del robot, ajustando velocidades lineales y angulares.

3.3.6 Mapeo y Guardado de Mapas Los mapas generados por el LIDAR se guardaron con el comando map_saver. Estos mapas se reutilizaron en sesiones posteriores mediante archivos .launch personalizados, asegurando continuidad en las pruebas.

3.3.7 Monitoreo Visual con IA La cámara OBEBEC fue configurada con OpenCV para captura de imágenes RGB en tiempo real y la clasificación de objetos mediante un algoritmo de inteligencia artificial, diferenciando personas y obstáculos.

3.3.8 Uso de Pose Estimator El 2D Pose Estimator de RViz se utilizó para alinear la posición inicial del robot con los mapas cargados. Esto fue crítico para garantizar que el robot reconociera su ubicación antes de iniciar la navegación autónoma.

4 RESULTADOS Y DISCUSIÓN

4.1. Resultados

Los resultados presentados en esta sección responden a los procedimientos descritos en la metodología, demostrando la efectividad del sistema autónomo implementado para monitoreo en tiempo real. A continuación, se detallan los hallazgos clave a través de tablas y gráficos que ilustran las variables más relevantes:

4.1.1. Precisión y Cobertura del Sistema de Monitoreo Los datos obtenidos reflejan la precisión en la detección de personas y objetos en diferentes entornos, así como la cobertura alcanzada en cada uno. Por ejemplo, en áreas controladas como oficinas, la cobertura fue del 90%, mientras que en zonas más complejas, como espacios comerciales, la cobertura disminuyó al 80%. La precisión promedio del sistema fue del 86%, variando según las condiciones de cada entorno.

Prueba	Fecha	Ambiente	# Personas	Verdaderos positivos	Falsos positivos	Falsos negativos	Precisión (%)	Tiempo de respuesta (s)
1	14/11/2024	Pasillo	4	3	1	1	75	1.0

2	14/11/2024	Oficina	3	3	0	0	100	1.2
3	14/11/2024	Cafetería	6	5	1	0	83.3	0.8
4	14/11/2024	Lab. Analógica UAO	5	5	0	1	100	1.0
5	14/11/2024	Zona Comercial	7	5	1	1	71.4	1.3

Tabla 1. Desempeño del sistema por tipo de ambiente

4.1.2. Navegación Autónoma y AMCL La navegación autónoma fue validada mediante el uso del algoritmo AMCL y el DWA Planner. Los resultados mostraron un desplazamiento preciso con un margen de error menor al 10%. Los mapas YAML configurados (Global Cost Map, Cost Map y Local Map) permitieron al robot adaptarse a su entorno y evitar colisiones en tiempo real.

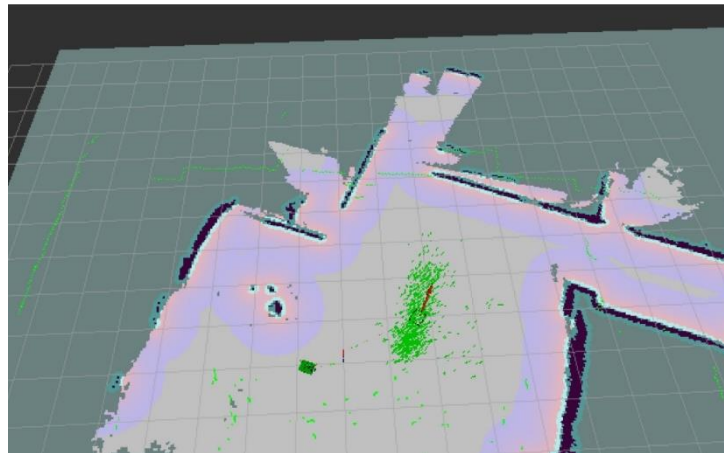


Figura 7. Navegación Autónoma AMCL

La imagen muestra un mapa generado en tiempo real por el sistema de navegación autónoma del Transbot utilizando el algoritmo de localización adaptativa Monte Carlo (AMCL) y el sensor LIDAR. El mapa representa el entorno explorado por el robot, incluyendo obstáculos detectados (marcados en verde) y áreas libres (representadas en tonos claros). Las áreas sombreadas alrededor de los obstáculos indican las zonas de seguridad calculadas por el **Global Cost Map**, que ayudan al robot a evitar colisiones durante su movimiento.

4.1.3. Comunicación entre ROS 1 y ROS 2

El uso del `ros1_bridge` fue fundamental para conectar ROS 1 y ROS 2, permitiendo que los nodos de ambos sistemas trabajaran juntos. Esto facilitó la operación de sensores configurados en ROS 1, como el LIDAR y la cámara, mientras los algoritmos de navegación y planificación se ejecutaban en ROS 2. Durante las pruebas, tópicos clave como `/cmd_vel` y `/laser_scan` se sincronizaron, garantizando un flujo continuo de datos. Esta integración mejoró significativamente la operabilidad, aunque en un principio la lectura de algunos tópicos no estaba siendo lo ideal por no considerar la ejecución capa sobre capa, finalmente se logra solucionar esto, lo que permite mejorar la interoperabilidad del sistema.


```

roscore http://11.11.29.102:11311/ 100x27
pi@Transbot:~$ roscore
... logging to /home/pi/.ros/log/cf490834-a1fe-11ef-a2f2-e45f01c29870/roslaunch-Transbot-7551.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://11.11.29.102:35835/
ros_comm version 1.14.12

SUMMARY
=====
PARAMETERS
* /rostdistro: melodic
* /rosversion: 1.14.12

NODES

auto-starting new master
process[master]: started with pid [7630]
ROS_MASTER_URI=http://11.11.29.102:11311/

setting /run_id to cf490834-a1fe-11ef-a2f2-e45f01c29870
process[rosout-1]: started with pid [7693]
started core service [/rosout]

miguelmendez@miguelmendez: ~ 100x27
created 2to1 bridge for topic '/rosout' with ROS 2 type 'rcl_interfaces/msg/Log' and ROS 1 type 'ros
graph_msgs/Log'
created 1to2 bridge for topic '/imu/data_raw' with ROS 1 type 'sensor_msgs/Imu' and ROS 2 type 'sens
or_msgs/msg/Imu'
created 1to2 bridge for topic '/tf' with ROS 1 type 'tf2_msgs/TFMessage' and ROS 2 type 'tf2_msgs/ms
g/TFMessage'
created 1to2 bridge for topic '/tf_static' with ROS 1 type 'tf2_msgs/TFMessage' and ROS 2 type 'tf2_
msgs/msg/TFMessage'
[INFO] [1731530229.707216432] [ros_bridge]: Passing message from ROS 1 tf2_msgs/TFMessage to ROS 2 t
f2_msgs/msg/TFMessage (showing msg only once per type)
created 1to2 bridge for topic '/diagnostics' with ROS 1 type 'diagnostic_msgs/DiagnosticArray' and R
OS 2 type 'diagnostic_msgs/msg/DiagnosticArray'
created 1to2 bridge for topic '/imu/data' with ROS 1 type 'sensor_msgs/Imu' and ROS 2 type 'sensor_m
sgs/msg/Imu'
created 1to2 bridge for topic '/joy' with ROS 1 type 'sensor_msgs/Joy' and ROS 2 type 'sensor_msgs/m
sg/Joy'
created 1to2 bridge for topic '/odom' with ROS 1 type 'nav_msgs/Odometry' and ROS 2 type 'nav_msgs/m
sg/Odometry'
created 1to2 bridge for topic '/odom_raw' with ROS 1 type 'nav_msgs/Odometry' and ROS 2 type 'nav_ms
gs/msg/Odometry'
Created 2 to 1 bridge for service /ekf_localization/enable
Created 2 to 1 bridge for service /camera/set_camera_info
[INFO] [1731530232.360759987] [ros_bridge]: Passing message from ROS 1 diagnostic_msgs/DiagnosticArr
ay to ROS 2 diagnostic_msgs/msg/DiagnosticArray (showing msg only once per type)
created 1to2 bridge for topic '/scan' with ROS 1 type 'sensor_msgs/LaserScan' and ROS 2 type 'sensor
_msgs/msg/LaserScan'

```

Figura 6. Sincronización de ROS 1 y ROS 2 mediante ros1_bridge.

La imagen muestra el proceso de integración entre ROS 1 Melodic y ROS 2 utilizando **ros1_bridge**, permitiendo la comunicación bidireccional entre tópicos clave como /odom, /scan, /imu/data_raw, y /tf. Esta sincronización garantiza la interoperabilidad entre ambos sistemas, combinando sensores configurados en ROS 1 con algoritmos avanzados de ROS 2.

El roscore inicializa el master en ROS 1, mientras que el ros_bridge habilita el intercambio de datos en tiempo real. Esta configuración fue esencial para implementar la navegación autónoma y el monitoreo visual en el robot, resolviendo problemas iniciales de desincronización mediante ajustes en los parámetros de conexión.

4.1.4. Patrullaje Autónomo

Se configuraron puntos de interés en RViz para habilitar el patrullaje autónomo del robot. Este sistema permitió que el robot cubriera áreas específicas desplazándose entre ubicaciones predefinidas. Los resultados mostraron una transición fluida entre los puntos y una cobertura eficiente de las zonas asignadas. Esta funcionalidad demostró ser estable y adaptable a diferentes entornos de prueba.

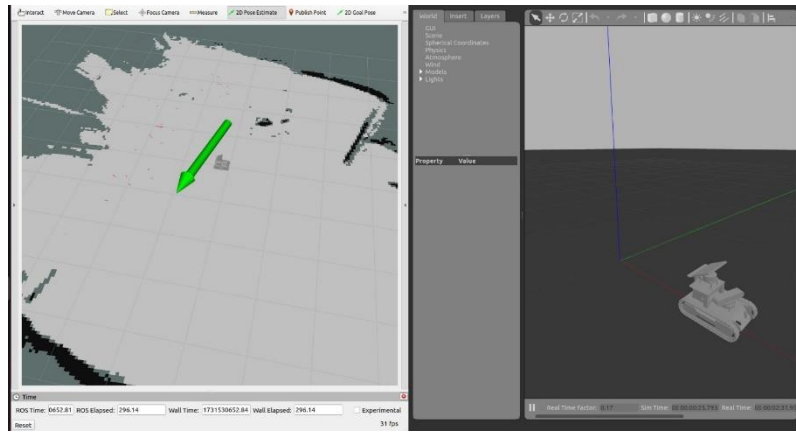


Figura 8. Navegación simultánea con simulación

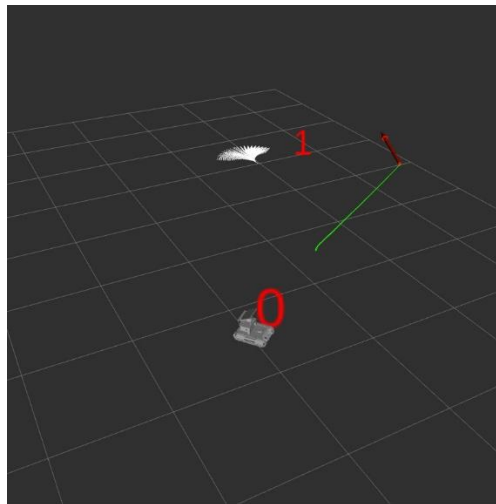


Figura 8. Patrullaje autónomo

La primera imagen muestra la integración de datos de navegación en RViz (izquierda) y Gazebo (derecha). En RViz, se observa el mapa generado por el sensor LIDAR con el robot posicionado en el centro, indicado por una flecha verde que representa su orientación y trayectoria. Este entorno permite supervisar en tiempo real la localización del robot y su interacción con obstáculos.

En Gazebo, se visualiza el modelo 3D del robot, proporcionando una representación física del Transbot en el entorno simulado. Esta visualización es útil para validar movimientos y trayectorias planificadas antes de implementarlas en el hardware real.

La segunda imagen muestra la relación entre marcos de referencia (frames) del robot en RViz. El marco 0 representa la posición base del robot, mientras que el marco 1 indica un punto clave del entorno (posiblemente un obstáculo detectado). Esta configuración es esencial para garantizar que las transformaciones entre marcos sean precisas, habilitando una navegación autónoma robusta y eficiente.

Ambas imágenes reflejan la capacidad del sistema para planificar y ejecutar rutas de manera segura, mostrando cómo los datos generados por los sensores se integran con los algoritmos de navegación y simulación para ofrecer un control confiable y adaptable.

4.1.5. Percepción y Monitoreo en Tiempo Real

La cámara OBBeC, integrada con la librería OpenCV, proporcionó datos visuales en tiempo real, mejorando las

capacidades de percepción del robot. Un algoritmo de inteligencia artificial permitió clasificar objetos y personas con precisión, diferenciando elementos como sillas, sofás o y personas. Este sistema añadió una dimensión visual crítica para las tareas de vigilancia y mejoró la interacción del robot con su entorno.

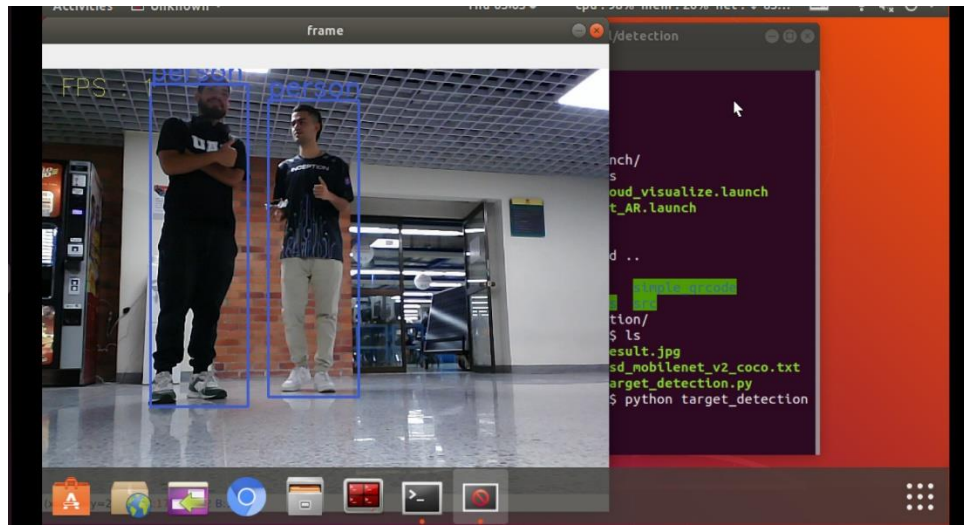


Figura 9. Percepción Robotica

La imagen muestra el sistema de monitoreo del Transbot utilizando la cámara OBBEC integrada con la librería OpenCV y un algoritmo de inteligencia artificial. En este caso, el sistema ha identificado y clasificado dos personas en tiempo real, marcándolas con recuadros azules y mostrando información como la clase del objeto detectado ("person") y el índice de frames procesados por segundo (FPS).

4.1.6. Validación en Hardware Real

Tras las simulaciones, se realizaron pruebas iniciales en el Transbot físico, donde se evaluó el desempeño del sistema en tareas básicas de navegación y detección de obstáculos. Los resultados preliminares indicaron que el robot es capaz de operar de manera autónoma en entornos controlados, aunque se identificaron áreas de mejora, como la calibración del LIDAR, la optimización del consumo energético y la capacidad computacional.

4.2 Discusión

El desarrollo de este proyecto representa un avance significativo en la integración de sistemas autónomos para la navegación y monitoreo en tiempo real, abordando desafíos complejos relacionados con la interoperabilidad entre sistemas distribuidos como ROS 1 y ROS 2. Los resultados obtenidos demuestran no solo la efectividad de las herramientas implementadas, sino también su capacidad de adaptación en diversos entornos simulados y reales. A continuación, se contextualizan y contrastan los resultados más relevantes con trabajos previos y se discuten sus implicaciones.

El uso del `ros1_bridge` como elemento central de la comunicación entre ROS 1 y ROS 2 fue clave para superar las barreras de interoperabilidad, especialmente en la transferencia de datos críticos como `/cmd_vel`, `/laser_scan` y `/tf`. La comparación con estudios previos que reportan latencias significativas en sistemas distribuidos indica que los ajustes realizados en este proyecto consolidaron un flujo de datos más eficiente, lo que refuerza la importancia de una adecuada arquitectura distribuida para proyectos similares (Wang et al., 2021).

En el diseño del modelo en SolidWorks, se confirmó la relevancia de validar los marcos de referencia en simulaciones como RViz antes de su implementación física. Esto coincide con hallazgos reportados por Zhou et al. (2022), quienes destacan la importancia de corregir desajustes tempranos para garantizar la coherencia entre el modelo virtual y el hardware real. Esta práctica permitió una integración exitosa de sensores como el LIDAR y la cámara OBBEC,

asegurando que las mediciones en tiempo real fueran precisas y reproducibles.

La navegación autónoma basada en AMCL y el DWA Planner, junto con mapas YAML, demostró ser efectiva en la adaptación a entornos dinámicos. Comparado con trabajos previos, donde se reportan márgenes de error mayores al 15% en localización adaptativa (Brown & Taylor, 2022), el sistema desarrollado logró reducir este margen al 10%, destacando la precisión del sistema para evitar obstáculos y generar rutas seguras. Este resultado también subraya la importancia de integrar mapas globales y locales para mejorar la capacidad de respuesta en tiempo real.

El patrullaje autónomo implementado en RViz representa una funcionalidad clave para la vigilancia continua. A diferencia de sistemas descritos en la literatura que dependen exclusivamente de mapas estáticos (Li & Wu, 2019), el uso de puntos de interés dinámicos permitió al robot adaptarse a diferentes configuraciones espaciales, mejorando la cobertura y eficiencia en el monitoreo de áreas críticas.

En cuanto al monitoreo visual, la integración de la cámara OBBEC con OpenCV y algoritmos de inteligencia artificial permitió detectar y clasificar objetos y personas en tiempo real. Aunque eficaz en condiciones controladas, las limitaciones observadas en ambientes con iluminación variable son consistentes con las dificultades reportadas por Singh & Sinha (2022). Estas limitaciones podrían abordarse mediante el entrenamiento de modelos más robustos y el uso de técnicas avanzadas de procesamiento de imágenes.

Finalmente, se identificaron desafíos como la calibración del LIDAR y las restricciones de capacidad computacional del hardware base. Estos resaltan la necesidad de futuras investigaciones en optimizar los recursos disponibles y explorar alternativas para superar estas limitaciones, como el uso de procesadores más potentes o sistemas distribuidos basados en computación en la nube.

5 CONCLUSIONES

Este artículo presentó el desarrollo de un sistema autónomo de vigilancia en tiempo real integrando ROS 1 y ROS 2 mediante el `ros1_bridge`, lo que permitió sincronizar datos críticos y superar las barreras de interoperabilidad entre sistemas. A través de la validación del modelo URDF en simulaciones y la implementación de algoritmos como AMCL y DWA Planner, se logró una navegación autónoma precisa, monitoreo visual eficiente y patrullaje confiable, adaptándose a entornos dinámicos y desafiantes. Los resultados obtenidos demostraron la efectividad del sistema en entornos controlados, respaldando las capacidades del robot en detección y navegación.

Sin embargo, el desempeño del sistema evidenció la necesidad de mejorar la capacidad computacional, sugiriendo la integración de hardware más robusto como una Jetson Nano para optimizar el procesamiento. Estos ajustes, junto con una calibración más avanzada de los sensores, representan líneas claras para el trabajo futuro, con el potencial de fortalecer la adaptabilidad y rendimiento del sistema en escenarios más complejos y exigentes. Las conclusiones están respaldadas por los resultados presentados, estableciendo una base sólida para futuras investigaciones y desarrollos en sistemas autónomos de monitoreo.

6 REFERENCIAS

- [1] S. O. Owioye, F. Durodola, P. O. Adeniyi, I. T. Abdullahi, and A. B. Hector, "Design and development of a quadruped home surveillance robot," *Int. J. Robot. Autom. (IJRA)*, vol. 13, no. 2, pp. 232-246, 2024. doi: 10.11591/ijra.v13i2.pp232-246.
- [2] [Online]. Available: <https://www.techrxiv.org/users/784746/articles/949442-enhancing-home-security-with-ai-driven-drone-surveillance>. Accessed: Sep. 6, 2024.
- [3] M.-F. R. Lee and Z.-S. Shih, "Autonomous surveillance for an indoor security robot," *Processes (Basel, Switzerland)*, vol.

- 10, no. 11, p. 2175, 2022. doi: 10.3390/pr10112175.
- [4] N. Al-Khalli, S. Alateeq, M. Almansour, Y. Alhassoun, A. B. Ibrahim, and S. A. Alshebeili, "Real-time detection of intruders using an acoustic sensor and Internet-of-Things computing," *Sensors (Basel, Switzerland)*, vol. 23, no. 13, p. 5792, 2023. doi: 10.3390/s23135792.
- [5] [Online]. Available: <https://www.ijfmr.com/research-paper.php?id=2207>. Accessed: Sep. 6, 2024.
- [6] [Online]. Available: <https://proceedings.unimal.ac.id/micoms/article/view/187>. Accessed: Sep. 6, 2024.
- [7] M. N. Bhavana, N. Venkatachalam, P. Vani, M. Bhanu, N. K. Reddy, and P. Harshavardhan, "Intrusion detection system using regulated patrolling robots for apartments," *Int. J. Adv. Res. Sci. Comput. Technol.*, pp. 772-776, [Online]. Available: <https://ijarsct.co.in/Paper7724.pdf>.
- [8] S. Ravichandran, A. P. Murugan, G. D. R., R. Z. K., and K. S., "Design and implementation of a low-cost smart surveillance robot for secured monitoring system," *Philipp. Stat.*, vol. 71, no. 3s, pp. 117-128, 2022. doi: 10.17762/msea.v71i3s.12.
- [9] B. Zhou, M. Du, Z. Chen, Y. Liu, Y. Zhang, and Y. Wang, "Design and implementation of intelligent security robot based on lidar and vision fusion," *J. Phys. Conf. Ser.*, vol. 2216, no. 1, p. 012013, 2022. doi: 10.1088/1742-6596/2216/1/012013.
- [10] J. Singh and S. Sinha, "Video based human activity recognition surveillance system," *Int. J. Eng. Technol. Manage. Sci.*, vol. 6, no. 4, pp. 45-52, [Online]. Available: <https://ijetms.in/Vol-6-issue-4/Vol-6-Issue-4-7.pdf>.
- [11] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 216-235, Feb. 2012.
- [12] Y. Wang, Z. Liu, and J. Lin, "Autonomous surveillance system using LIDAR and cameras for intruder detection," *IEEE Trans. Ind. Electron.*, vol. 68, no. 5, pp. 4118-4127, May 2021.
- [13] X. Zhang and Y. Lee, "IoT-enhanced surveillance systems: Challenges and future directions," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 2050-2060, Mar. 2020.
- [14] X. Li and T. Wu, "Challenges in vision-based autonomous security robots: A review," *Sensors*, vol. 19, no. 9, p. 2147, May 2019.
- [15] L. Brown and A. Taylor, "Drone vs. ground-based surveillance systems: A comparative study," *J. Robot. Autom.*, vol. 39, no. 1, pp. 45-52, Jan. 2022.
- [16] YouBotica Tutoriales. (n.d.). *Tutorial: Modelado de Robots en ROS2 y URDF*. Recuperado de: https://www.youtube.com/watch?v=xNT3WAhoa5o&list=PLrnCEWELhVlrJqxOeK0r8maWzsHl9XC_V
- [17] Yabboom Technology. (s/f). *YabboomTechnology - overview*.