

Internet de las Cosas IoT

Taller 2

Jhon Vargas Cod 2273974, Felipe Garzon Cod 2269970, and Mateo Suarez Cod 2262893

Abstract—El siguiente informe contiene el proceso de investigación de las herramientas ROS, MoveIt y Gazebo. Una breve introducción a sistemas LIDAR y SLAM y la importancia del uso de Docker y algunos ejemplos.

Index Terms—Streamlit, IoT, Datos, Información, Análisis.

I. INTRODUCTION

La robótica moderna ha experimentado un avance significativo gracias al uso de tecnologías como ROS (Robot Operating System), MoveIt, Gazebo, y Docker. Estas herramientas, junto con sensores avanzados como LIDAR y algoritmos de localización y mapeo como SLAM, permiten el desarrollo de sistemas autónomos eficientes. Este documento presenta una investigación detallada sobre estas tecnologías, su aplicación en el contexto del Internet de las Cosas (IoT), y ejemplos prácticos de implementación.

II. HERRAMIENTAS FUNDAMENTALES

A. 1. ROS (Robot Operating System)

ROS es un conjunto de bibliotecas y herramientas que ayudan a los desarrolladores a crear aplicaciones robóticas. Su arquitectura distribuida permite la comunicación entre múltiples nodos que representan sensores, actuadores y lógica de control. ROS puede integrarse con redes IoT para recopilar, procesar y actuar sobre datos del entorno, habilitando robótica conectada y colaborativa [1].

B. 2. MoveIt

MoveIt es una plataforma de planificación de movimiento para robots manipuladores. Facilita la planificación de trayectorias, control de brazo robótico, y la interacción con sensores. Ideal para sistemas industriales donde se requiere manipulación remota precisa y colaborativa mediante sensores distribuidos [2].

C. 3. Gazebo

Gazebo es un simulador 3D que permite modelar robots, sensores y ambientes complejos. Se integra perfectamente con ROS. Permite probar interacciones entre dispositivos IoT y robots en un entorno seguro antes del despliegue físico [3].

III. TECNOLOGÍAS DE ROBÓTICA ACTUAL

IV. SISTEMAS LIDAR

LIDAR (Light Detection and Ranging) es una tecnología de detección que usa láseres para medir distancias con alta precisión. Funciona emitiendo pulsos de luz láser hacia el entorno y midiendo el tiempo que tarda cada pulso en regresar tras reflejarse en los objetos. Esta técnica permite construir modelos tridimensionales detallados del entorno en tiempo real.

Tipos

- LIDAR rotativo: Utiliza un conjunto giratorio de emisores láser para escanear el entorno en 360°, común en vehículos autónomos.
- LIDAR sólido: No tiene partes móviles, lo que lo hace más resistente y compacto; ideal para aplicaciones en drones y dispositivos móviles.
- LIDAR flash: Emite pulsos de luz sobre un área amplia simultáneamente, permitiendo capturas rápidas del entorno, útil en entornos interiores.

V. SISTEMAS SLAM

SLAM (Simultaneous Localization and Mapping) permite que un robot construya un mapa del entorno mientras localiza su posición dentro de él. Esta tecnología es crucial para la navegación autónoma, ya que elimina la necesidad de mapas preexistentes, permitiendo que los robots operen en entornos desconocidos o dinámicos. El SLAM combina datos de sensores como LIDAR, cámaras RGB-D, IMU y odometría para generar mapas mientras calcula simultáneamente la posición del robot. Esto se logra mediante algoritmos como Extended Kalman Filter (EKF), FastSLAM, y Graph-Based SLAM. [4].

Tipos

- SLAM visual: Utiliza cámaras monocromáticas o estéreo para generar mapas visuales.
- SLAM con LIDAR: Basado en escaneos láser para mayor precisión en entornos estructurados.
- SLAM inercial: Integra datos de acelerómetros y giroscopos para mejorar la estabilidad en entornos con poca visibilidad.

VI. POTENCIAL DE DOCKER EN ROBÓTICA E IoT

Docker permite encapsular entornos de desarrollo y ejecución en contenedores ligeros y portables.

Ventajas

- Reproducibilidad del entorno.
- Escalabilidad y despliegue en la nube.
- Compatibilidad multiplataforma.

VII. EJEMPLOS PRÁCTICOS

A. Robot con ROS, MoveIt y Gazebo

- 1) Crear un contenedor Docker con ROS y Gazebo.
- 2) Lanzar un modelo de robot móvil.
- 3) Simular movimiento mediante teleoperación y planificación con MoveIt [5].

B. Robot con LIDAR y SLAM

- 1) Agregar sensores LIDAR al modelo anterior.
- 2) Lanzar SLAM con gmapping.
- 3) Visualizar el mapeo en Rviz.

VIII. PUNTO 2 TALLER

Objetivo: Controlar un brazo UR5 utilizando PyBullet.

- Se compartirán dos repositorios de código para que los estudiantes puedan ejecutarlos y probar la simulación del brazo robótico.
- Posteriormente, los estudiantes deberán crear un contenedor utilizando Docker, incluyendo un Dockerfile donde se definan las dependencias necesarias.
- El objetivo final es que monten y corran la simulación dentro del contenedor de Docker.

Se busca controlar un brazo robótico UR5 usando PyBullet. Para ello, se proporcionan dos repositorios de código que los estudiantes deben ejecutar y luego empaquetar en un contenedor Docker utilizando un Dockerfile. Dentro del repositorio se encuentran los códigos necesarios para realizar la simulación.

IX. PUNTO 3 TALLER

El objetivo es implementar un robot móvil (TurtleBot3 modelo "tortuga") con un sensor LIDAR, utilizando ROS Noetic y el algoritmo de SLAM (gmapping), todo ejecutándose dentro de un contenedor Docker, para crear un mapa del entorno en tiempo real.

SIMULACIÓN DE TURTLEBOT3 CON ROS, SLAM Y DOCKER

- 1) Se crea un archivo Dockerfile para construir la imagen que tendrá ROS, TurtleBot3 y SLAM instalados.
 - Se parte de una imagen base de ROS Noetic con escritorio completo.
 - Se instalan los paquetes de TurtleBot3, simulaciones y gmapping.
 - Se configura el modelo de TurtleBot3 que se usará.
 - Al iniciar el contenedor, se lanza automáticamente el entorno Gazebo con el mundo de simulación.
- 2) Una vez creada la imagen de Docker, se deben seguir varios pasos para lanzar la simulación, el SLAM y la teleoperación.

- Se ejecuta el contenedor con acceso al servidor gráfico (DISPLAY) para visualizar Gazebo.
- Se usa la red del host (--network=host) para evitar problemas de comunicación ROS.
- El contenedor se nombra slam-bot.

3) Visualizar el mapa en RViz.

- Se lanza RViz con una configuración de visualización predefinida para gmapping.
- Desde RViz se puede ver cómo el TurtleBot3 construye el mapa en tiempo real conforme se mueve.

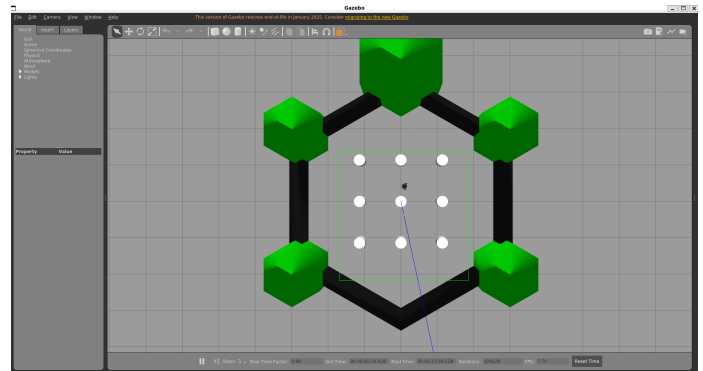


Fig. 1. Imagen Gazebo.

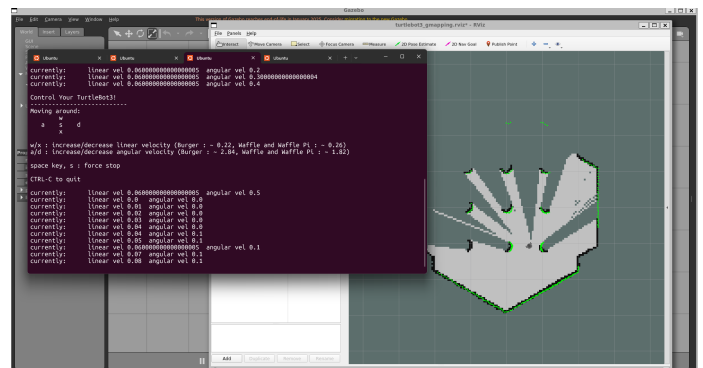


Fig. 2. Imagen RViz.

REFERENCES

- [1] M. Quigley, B. Gerkey, and W. D. Smart, *Programming Robots with ROS: A Practical Introduction to the Robot Operating System*. O'Reilly Media, 2014.
- [2] Open Source Robotics Foundation, "ROS.org Documentation," <https://docs.ros.org/>, accessed: 2025-04-21.
- [3] Open Robotics, "Gazebo Simulator Documentation," <https://gazebo.org/docs>, accessed: 2025-04-21.
- [4] Y. Zheng, B. Chen, and T. Wang, "Slam algorithms for mobile robotics in iot applications: A survey," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4735–4750, 2020.
- [5] Docker Inc., "Docker Documentation," <https://docs.docker.com/>, accessed: 2025-04-21.