

Investigación: ROS, MoveIt, Gazebo, LIDAR, SLAM y Docker en IoT

Edward Sosa , Sebastian Zapata

15 de octubre de 2025

1. Introducción

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

2. Herramientas Fundamentales

Definición: ROS (Robot Operating System)

ROS es un framework de código abierto para el desarrollo de software de robots. Proporcione servicios similares a un sistema operativo, incluyendo abstracción de hardware, control de dispositivos de bajo nivel, implementación de funcionalidades de uso común, paso de mensajes entre procesos y gestión de paquetes [1].

Alcance en IoT:

- Integración de sensores y actuadores
- Comunicación distribuida entre dispositivos
- Gestión de datos en tiempo real

Aplicaciones:

- Robots de servicio
- Vehículos autónomos
- Sistemas de manufactura inteligente

Definición: MoveIt

MoveIt es el framework de código abierto más popular para la manipulación de robots móviles. Proporciona algoritmos avanzados para cinemática, planificación de movimientos, manipulación, percepción 3D y control [2].

Alcance en IoT:

- Control preciso de actuadores robóticos
- Integración con sistemas de percepción
- Planificación de tareas autónomas

Aplicaciones:

- Brazos robóticos industriales
- Sistemas de pick and place
- Robots de asistencia médica

Definición: Gazebo

Gazebo es un simulador de robots 3D de código abierto que ofrece la capacidad de simular de manera precisa y eficiente poblaciones de robots en entornos complejos tanto interiores como exteriores [3].

Alcance en IoT:

- Pruebas de algoritmos sin hardware físico
- Simulación de entornos IoT complejos
- Validación de sistemas multi-robot

Aplicaciones:

- Desarrollo de algoritmos de navegación
- Pruebas de integración de sensores
- Simulación de fábricas inteligentes

3. Robótica Actual y Tecnologías Asociadas

Definición: Robótica Actual

La robótica moderna integra inteligencia artificial, aprendizaje automático y tecnologías IoT para crear sistemas autónomos capaces de interactuar con su entorno y tomar decisiones complejas [4].

Tipos orientados a IoT:

- Robots de servicio conectados
- Sistemas robóticos en la nube
- Enjambres de robots cooperativos

Definición: Sistemas LIDAR

LIDAR (Light Detection and Ranging) es una tecnología de detección remota que utiliza pulsos de láser para medir distancias y crear mapas 3D del entorno [5].

Tipos orientados a IoT:

- LIDAR mecánico de 360°
- LIDAR sólido-state
- LIDAR de bajo costo para robótica

Definición: Sistemas SLAM

SLAM (Simultaneous Localization and Mapping) es una técnica computacional que construye un mapa de un entorno desconocido mientras simultáneamente rastrea la posición del robot dentro de ese mapa [6].

Tipos orientados a IoT:

- Visual SLAM
- LIDAR SLAM
- SLAM basado en filtros de partículas

4. Docker en IoT Robótica

Definición: Docker en IoT

Docker permite empaquetar aplicaciones y sus dependencias en contenedores portables, facilitando el despliegue y escalado de aplicaciones robóticas en entornos IoT [7].

Potencial:

- Aislamiento de dependencias
- Portabilidad entre diferentes sistemas
- Escalabilidad en edge computing

5. Ejemplos Prácticos

Ejemplo 1: Robot Sencillo con ROS

Objetivo: Crear un robot básico con capacidad de movimiento usando ROS, MoveIt y Gazebo.

Pasos:

1. Instalar ROS y dependencias
2. Crear el paquete del robot
3. Definir el modelo URDF del robot
4. Configurar MoveIt para planificación
5. Implementar el entorno en Gazebo
6. Probar el movimiento del robot

Código básico:

```
# Ejemplo de comando ROS
roslaunch my_robot gazebo.launch
roslaunch moveit_commander move_group.py
```

Ejemplo 2: Robot con LIDAR y SLAM

Objetivo: Implementar un robot con capacidad de mapeo y localización usando LIDAR y SLAM.

Pasos:

1. Integrar sensor LIDAR en el modelo
2. Configurar drivers del LIDAR en ROS
3. Implementar algoritmo SLAM (gmapping)
4. Visualizar mapas con RVIZ
5. Probar navegación autónoma

Código básico:

```
# Ejemplo de SLAM con LIDAR
roslaunch turtlebot3_slam turtlebot3_slam.launch
roslaunch turtlebot3_navigation navigation.launch
```

6. Conclusión

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

7. Referencias

Referencias

- [1] ROS Documentation. (2024). *ROS.org*. Recuperado de: <https://www.ros.org/>
- [2] MoveIt Documentation. (2024). *MoveIt.ros.org*. Recuperado de: <https://moveit.ros.org/>
- [3] Gazebo Documentation. (2024). *GazeboSim.org*. Recuperado de: <https://gazebo.org/>
- [4] Lynch, K. M., & Park, F. C. (2024). *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press.
- [5] Vosselman, G., & Maas, H. G. (2024). *Airborne and Terrestrial Laser Scanning*. Whittles Publishing.

- [6] Durrant-Whyte, H., & Bailey, T. (2024). *Simultaneous Localization and Mapping: Part I*. IEEE Robotics & Automation Magazine.
- [7] Docker Documentation. (2024). *Docker.com*. Recuperado de: <https://docs.docker.com/>