Autonomous Navigation with ROS 2

Summer School - Day 2

Soham and Tarun

June 11 2025

Table of Contents

o

Table of Contents

- Recap & Our Plan for Today
- The End Goal: Autonomous Navigation Demo
- Deconstructing the Autonomous Stack
- Tying It All Together
- Assignment: Let's Get Autonomous
- Conclusion & Next Steps

Recap of Day 1

- We built a digital twin of our robot using **URDF**.
- We established a standard hardware interface with **ros2_control**.
- We visualized our robot in **RViz** and simulated its physics in **Gazebo**.
- We successfully drove the robot manually using a **teleop** node.
- **Result:** We have a robot that reliably listens to velocity commands, both in simulation and on hardware.

Our Plan for Day 2: From Listening to Thinking

- Today, we give our robot 'eyes' and a 'brain'.
- Our goal is to move from manual control to **full autonomy**.
- **The Core Problems We Will Solve:**
  1. How does a robot perceive its environment? $\rightarrow$ **Sensor Integration**
  2. How does it build a map and know its location? $\rightarrow$ **SLAM**
  3. How does it plan and execute a path? $\rightarrow$ **Navigation (Nav2)**
- We will continue our theme of learning concepts *implicitly* by building a complete system.

Table of Contents

- Recap & Our Plan for Today
- The End Goal: Autonomous Navigation Demo
- Deconstructing the Autonomous Stack
- Tying It All Together
- Assignment: Let's Get Autonomous
- Conclusion & Next Steps

Demo Overview: The Autonomous Robot

- **Objective**: Show a complete autonomous pipeline, building on our Day 1 robot.
- We will see the robot:
    1. Use a LIDAR sensor to see the world.
    2. Build a 2D map of its environment using `slam_toolbox`.
    3. Use the `nav2` stack to navigate to a goal on that map.
- This provides the big-picture context before we dive into each new component.

Table of Contents

- Recap & Our Plan for Today
- The End Goal: Autonomous Navigation Demo
- Deconstructing the Autonomous Stack
- Tying It All Together
- Assignment: Let’s Get Autonomous
- Conclusion & Next Steps

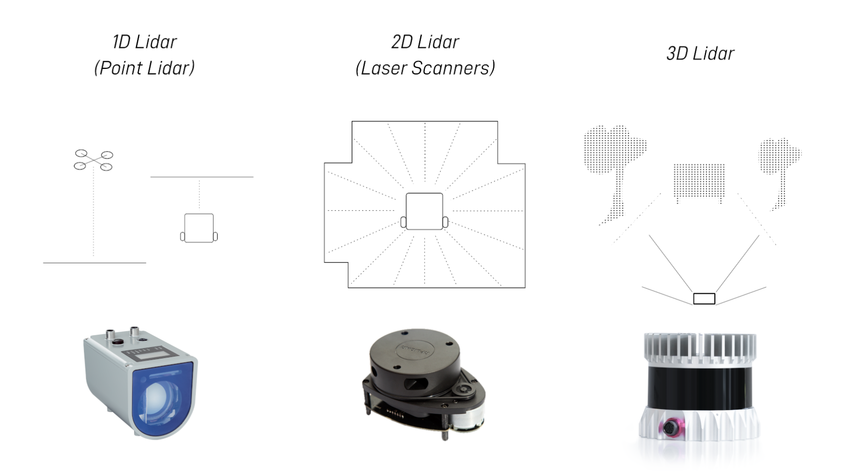
Step 1: Giving the Robot ”Eyes”

Why Sensors?

- For a robot to act autonomously, it must first **perceive** its environment.
- Sensors are the bridge between the physical world and the robot’s software.
- Today’s focus: **LIDAR** (Light Detection and Ranging), which provides distance measurements.
- Real sensors have **drivers** (nodes) that publish data to ROS topics.
- Simulated sensors in Gazebo use **plugins** to publish data to the *exact same topics*.
- For LIDAR, the standard message type is `sensor_msgs/msg/LaserScan`.
- This abstraction is key for seamless sim-to-real transfer!

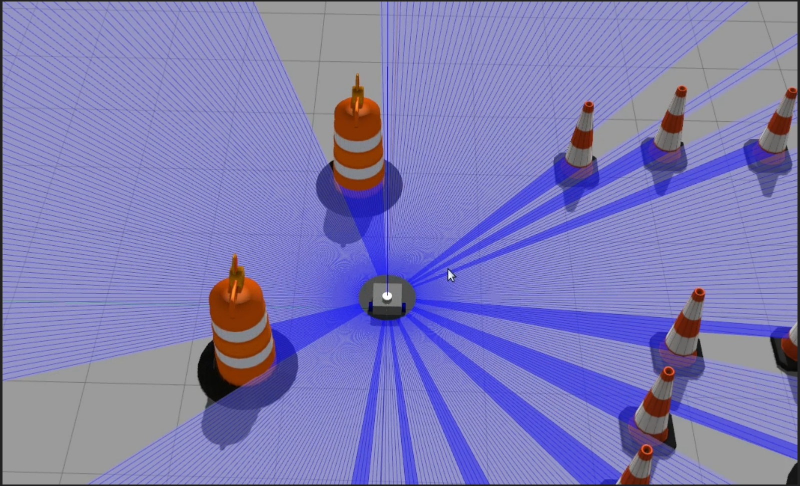
LiDARs

3 Deconstructing the Autonomous Stack



Gazebo Sensors

3 Deconstructing the Autonomous Stack



Step 2: Building a Map & Knowing Your Place

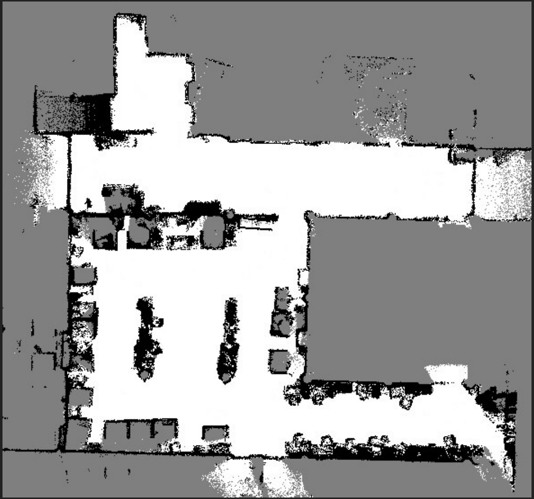
The SLAM Problem

- **S**imultaneous **L**ocalization **a**nd **M**apping.
- The core "chicken-and-egg" problem of mobile robotics:
  - To build an accurate map, you need to know your precise location.
  - To know your precise location, you need an accurate map.
- SLAM algorithms solve both problems at the same time.

SLAM in ROS: `slam_toolbox`

- We don't have to reinvent the wheel! ROS provides powerful, ready-to-use packages.
- `slam_toolbox` is a popular and robust SLAM implementation for ROS 2.
- **Inputs**:
  - Sensor data (e.g., `/scan` from our LIDAR).
  - Transform data (`/tf`, especially the `odom` frame from Day 1).
- **Outputs**:
  - The map (`/map` topic).
  - A corrected transform (`map` $\rightarrow$ `odom`) to fix odometry drift.

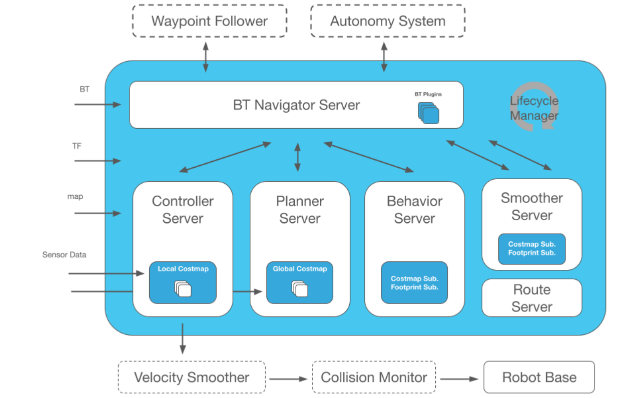
Grid Map



Step 3: Intelligent Movement with Nav2

3 Deconstructing the Autonomous Stack

- Now that we have a map and know our location, we can navigate intelligently.
- **Nav2** is the modern, highly-configurable navigation stack in ROS 2.



How Nav2 Connects to Our Robot

- **The "Aha!" Moment:** How does Nav2 actually move the robot?
- It uses a layered map called a **Costmap** to plan paths that avoid obstacles.
- Its local planner continuously calculates and publishes velocity commands to the `/cmd_vel` topic!
- This is the **exact same interface** our `teleop` node used and our `ros2_control` hardware interface listens to.

Goal Pose (RViz) $\rightarrow$ **Nav2** $\rightarrow$ `/cmd_vel` $\rightarrow$ **ros2_control** $\rightarrow$ **Motors**

Table of Contents

- Recap & Our Plan for Today
- The End Goal: Autonomous Navigation Demo
- Deconstructing the Autonomous Stack
- Tying It All Together
- Assignment: Let’s Get Autonomous
- Conclusion & Next Steps

The Full System Graph

4 Tying It All Together

- Let's visualize how all the nodes from Day 1 and Day 2 work together.
- This graph shows the flow of information, from sensor perception to motor actuation.

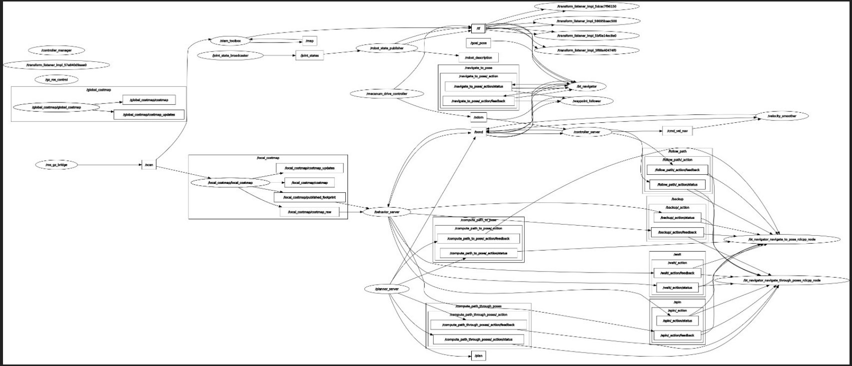


Table of Contents

▶ Recap & Our Plan for Today

▶ The End Goal: Autonomous Navigation Demo

▶ Deconstructing the Autonomous Stack

▶ Tying It All Together

▶ Assignment: Let’s Get Autonomous

▶ Conclusion & Next Steps

Problem Setup: From Teleop to Autonomy

- We will start with your completed robot from the Day 1 assignment.
- The necessary sensor URDF additions and Gazebo plugins are provided in the repository.
- Your goal is to launch and configure the autonomy stack.
- **Tasks**:
  1. **Integrate Sensors**: Launch the robot with its new LIDAR sensor. Verify in RViz that you are seeing `/scan` data.
  2. **Map the World**: Launch `slam_toolbox`. Drive the robot around the simulation environment using `teleop` to build a complete map.
  3. **Save the Map**: Use the provided tools to save your generated map to a file.
  4. **Navigate!**: Launch the `nav2` stack, loading your saved map. Use the "2D Goal Pose" button in RViz to give your robot a destination.
- **Submission Criteria**:
  - A screen recording showing your robot successfully navigating to a goal in the simulation without collisions.
  - The saved map files you generated.

Table of Contents

- ▶ Recap & Our Plan for Today
- ▶ The End Goal: Autonomous Navigation Demo
- ▶ Deconstructing the Autonomous Stack
- ▶ Tying It All Together
- ▶ Assignment: Let’s Get Autonomous
- ▶ Conclusion & Next Steps

Conclusion & Next Steps

- Today we gave our robot the ability to see, understand, and move through its world autonomously.
- We saw how high-level autonomy stacks like Nav2 build upon the foundational components from Day 1.
- **Where to go from here?**:
  - **Advanced Perception**: Integrating cameras, 3D LIDARs, and object detection.
  - **Manipulation**: Using the same principles with `MoveIt!` for robotic arms.
  - **Custom Behaviors**: Writing your own plugins for the Nav2 behavior tree.
- Q&A and feedback

Questions?