INTRO TO SLAM & NAV

Presented by Victoria Rotich

Presented by Victoria Rotich

OVERVIEW

01 What is Navigation

02 Why Navigation matters

03 Why SLAM is needed

04 Applications of SLAM

05 SLAM Approaches

06 ROS2 Tools for SLAM & Navigation

07 Types of SLAM Systems

08 Installation Guide

09 Simulation Demo

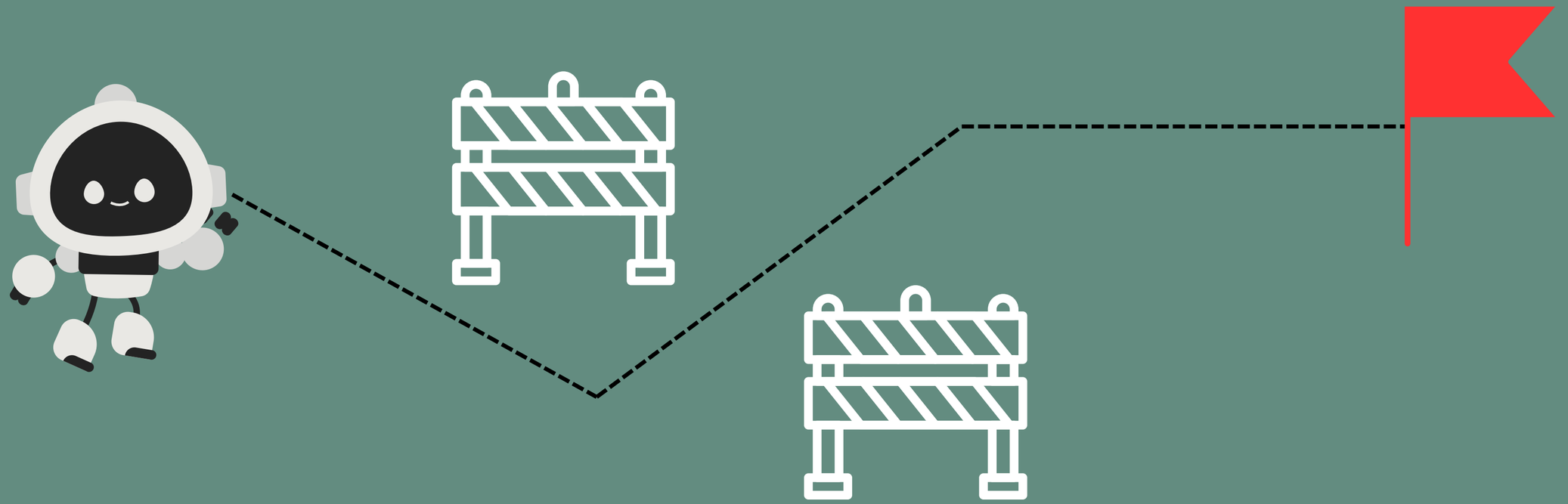
10 Path Planning in Detail

11 RViz Plugin Setup

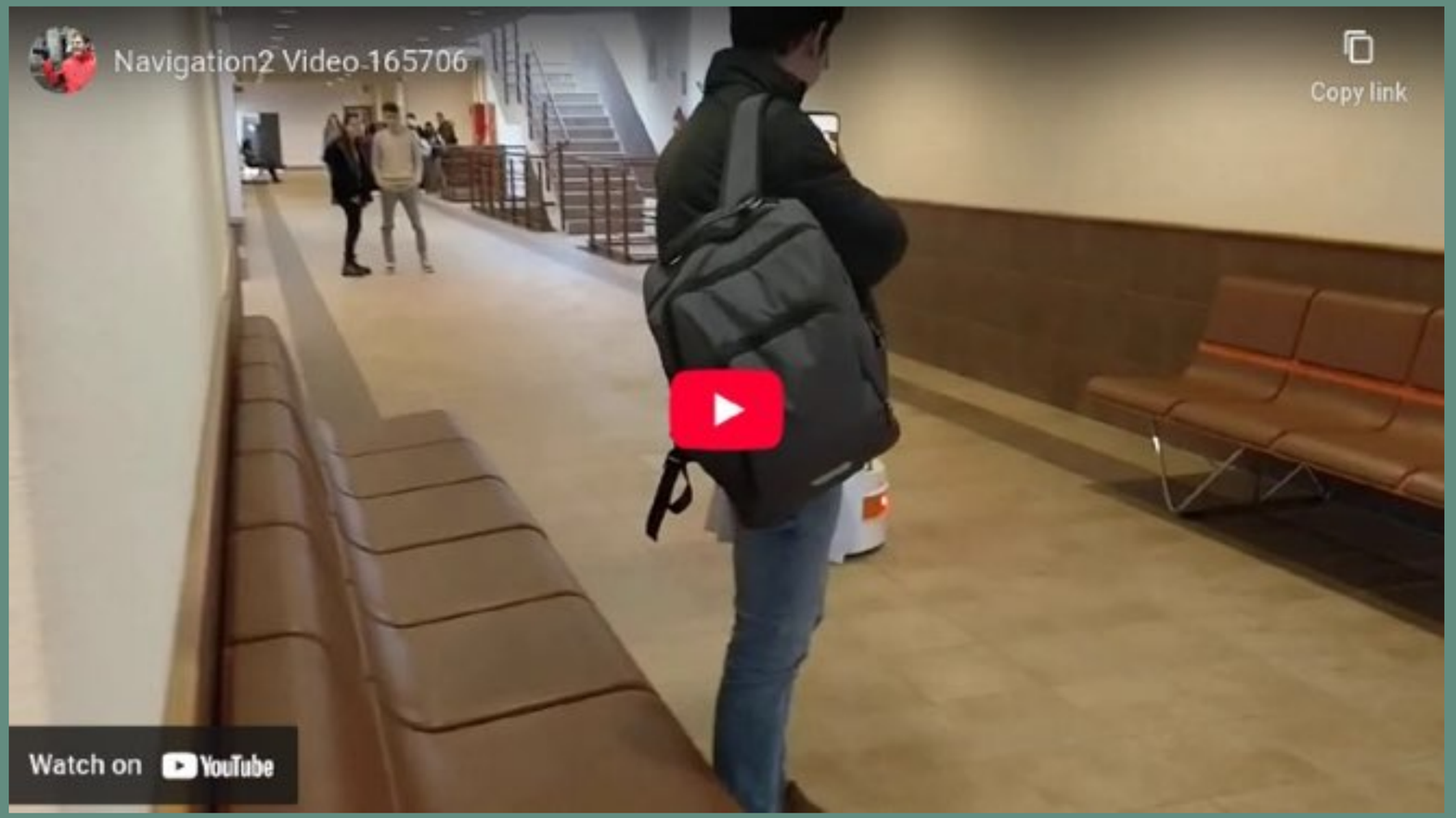
12 SLAM_Toolbox vs AMCL

WHAT IS NAVIGATION ?

- Robot navigation = moving from a start → goal, safely & efficiently



WHAT IS NAVIGATION ?

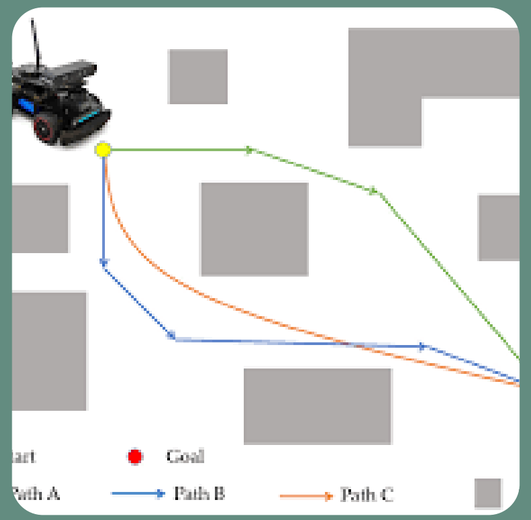


WHAT IS NAVIGATION ?

It requires the following:

Perception
(sense world)

Localization
(know where you are)



Path Planning
(decide where to go)

Control
(execute movement)

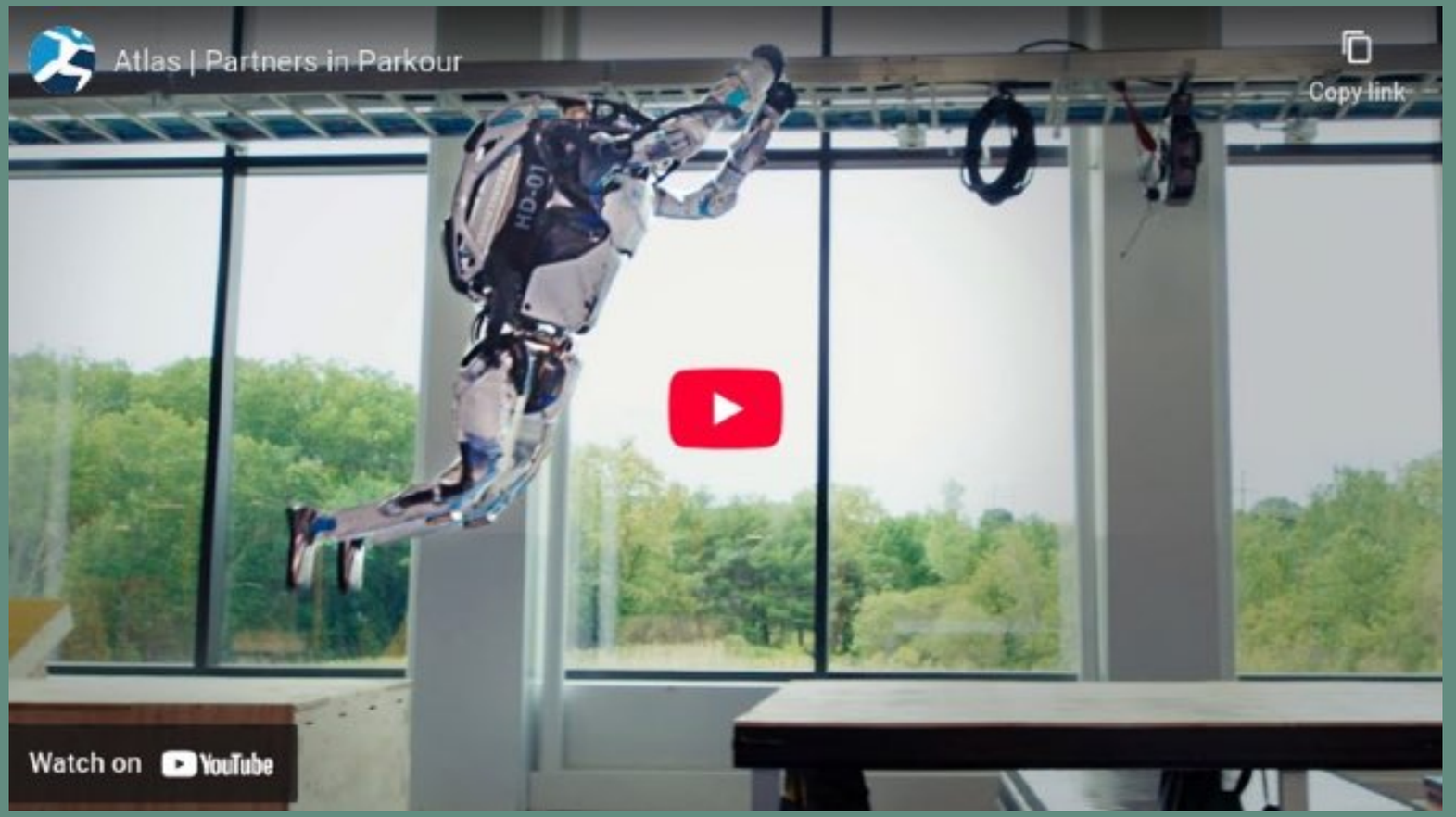
WHAT IS NAVIGATION ?

Q: Why do you think navigation matters?

WHY NAVIGATION MATTERS

- Autonomous Operation
- Obstacle Avoidance and Safety
- Efficient and Optimized Paths
- Adaptabiity to changing Environments
- Enabling Complex Tasks
- Foundation for Advanced Capabilities

WHY NAVIGATION MATTERS



WHY SLAM IS NEEDED

- SLAM - Simultaneous Localization and Mapping
- Refers to creating a map of an unknown environment while simultaneously determining location
- Robots need maps for navigation in unknown spaces
- GPS is unreliable indoors

APPLICATIONS OF SLAM

Automated vehicles & robotics

AR & VR

Agriculture

Medicine

SLAM IS DIFFICULT

OR

SLAM IS DIFFICULT

In order to accurately localize, we need good maps, but to have good maps, we need to accurately localize

SLAM APPROACHES

We'll break them down into 4 types:

-Filter Based SLAM

-Graph Based SLAM

-Deep Learning Based SLAM

- Modern Methods

1. FILTER BASED SLAM

Q: What types of filters have you worked with before?

1. FILTER BASED SLAM

Overview

A class of algorithms that use a probabilistic filter, to recursively estimate a robot's pose and simultaneously build a map of an unknown environment from sensor data

1. FILTER BASED SLAM

Overview

A class of algorithms that use a probabilistic filter, to recursively estimate a robot's pose and simultaneously build a map of an unknown environment from sensor data

How it works

1. State Estimation
2. Prediction Step
3. Measurement Update (Correction Step)
4. Recursion

1. FILTER BASED SLAM

Kalman Filter-based SLAM

- Extended Kalman Filter (EKF) SLAM
  →For non-linear systems
- Unscented Kalman Filter (UKF) SLAM
  →An improvement over EKF

Particle Filter (PF) SLAM

- Also known as FastSLAM
- Uses a collection of particles to represent the robot's state distribution

2. GRAPH BASED SLAM

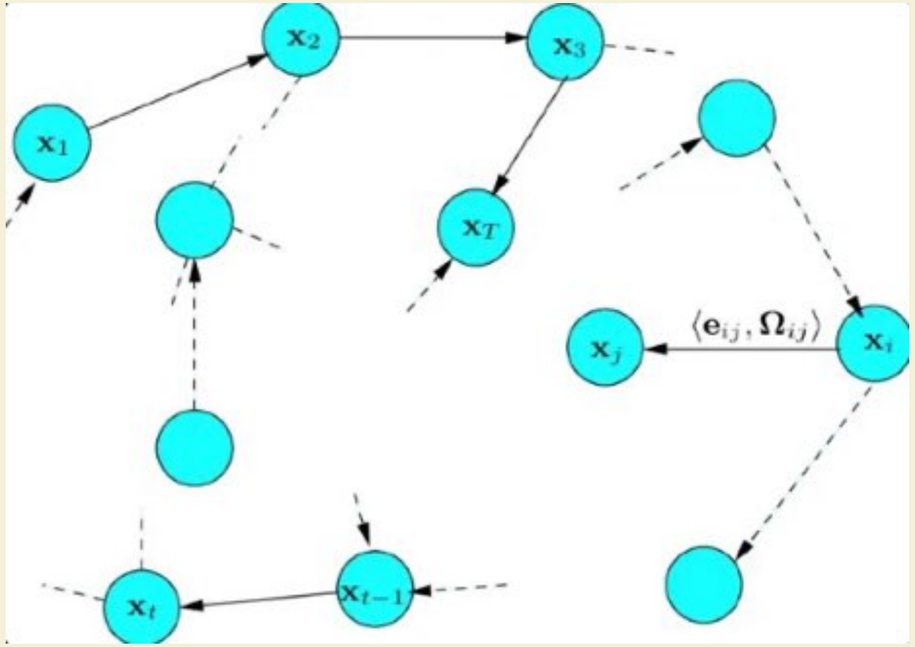
Overview

A technique that represents the problem as a graph where nodes are the robot's poses over time and edges are constraints from sensor measurements and odometry.

How it works

1. Graph Construction
2. Error Minimization

2. GRAPH BASED SLAM



3. DEEP LEARNING BASED SLAM

Overview

Attempts to solve the SLAM problem using neural networks and deep learning.

Existing Algorithms:

They include:

1. RatSLAM
2. LIFT-SLAM
3. EnvSLAM

4. MODERN METHODS

A. Visual SLAM

Uses camera-based visual input to simultaneously build a map of an unknown environment and track its own position and orientation within that map.

4. MODERN METHODS

A. Visual SLAM

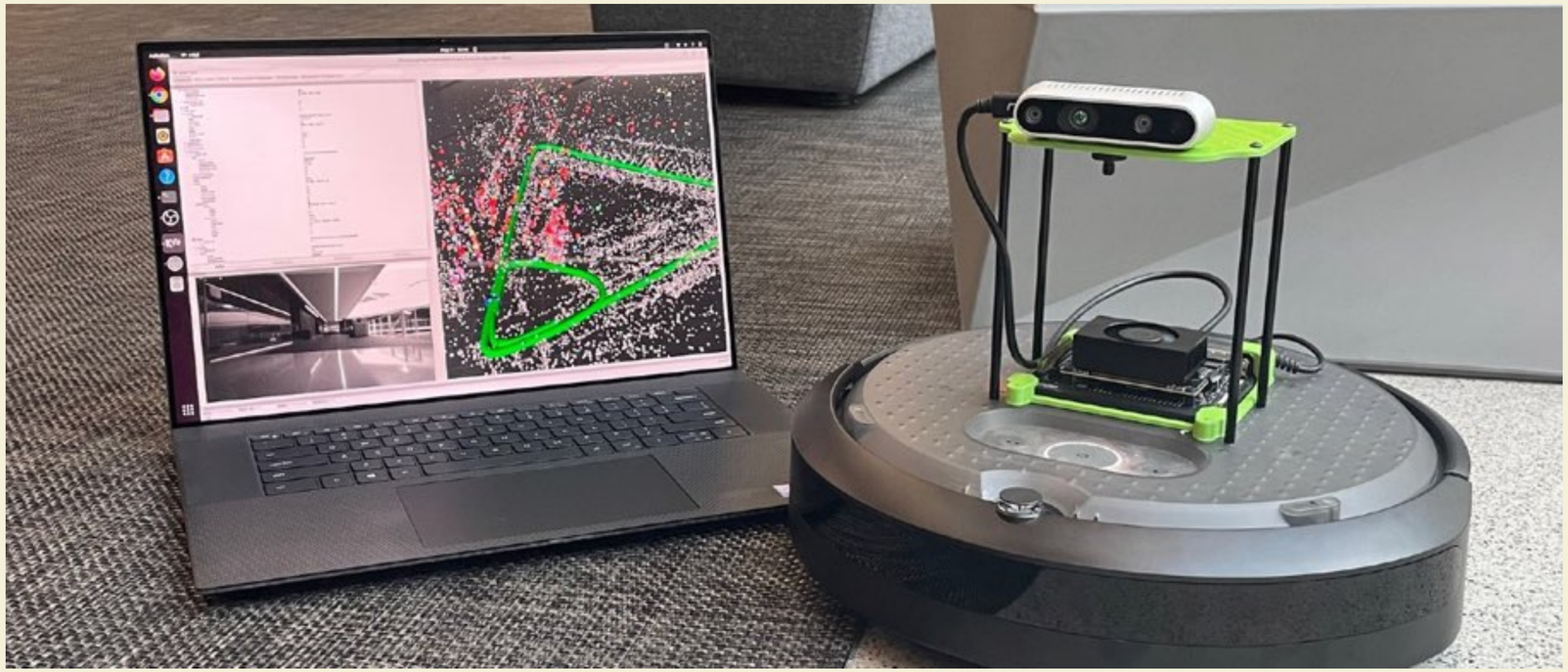
Uses camera-based visual input to simultaneously build a map of an unknown environment and track its own position and orientation within that map.

B. LiDAR-based SLAM

Uses a laser sensor to generate a 3D map of an environment while simultaneously determining the sensor's precise location within that map.

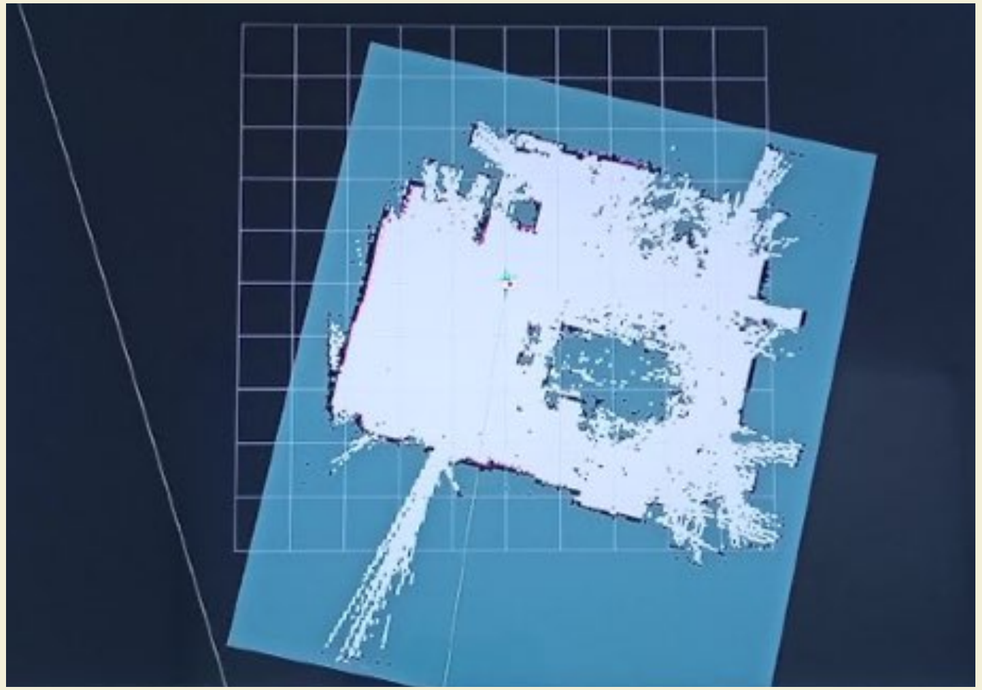
4. MODERN METHODS

A. Visual SLAM



4. MODERN METHODS

B. LiDAR-based SLAM



ROS2 TOOLS FOR SLAM & NAVIGATION

1.slam_toolbox → 2D LiDAR-based mapping

ROS2 TOOLS FOR SLAM & NAVIGATION

1.slam_toolbox

2.Nav2 (Navigation2) → stack for path planning, control, recovery behaviors

ROS2 TOOLS FOR SLAM & NAVIGATION

1.slam_toolbox

2.Nav2 (Navigation2)

3. TurtleBot4 → mobile robot intended for education and research

TYPES OF SLAM SYSTEMS

Synchronous and Asychronous

Online & offline

TYPES OF SLAM SYSTEMS

Synchronous SLAM

Requires all operations to complete in strict order before the next one can begin, acting as a blocking architecture

TYPES OF SLAM SYSTEMS

Synchronous SLAM

Requires all operations to complete in strict order before the next one can begin, acting as a blocking architecture

Asynchronous SLAM

Allows operations to run in parallel or be deferred, not blocking the main process and providing more flexibility for tasks like map updates

TYPES OF SLAM SYSTEMS

Online SLAM

A real-time process, used by robots during operation to build maps while simultaneously navigating by prioritizing speed and immediate control, even if it means a less precise or incomplete map.

TYPES OF SLAM SYSTEMS

Online SLAM

A real-time process, used by robots during operation to build maps while simultaneously navigating by prioritizing speed and immediate control, even if it means a less precise or incomplete map.

Offline SLAM

A post-processing technique that takes place after data collection, allows for computationally intensive algorithms, and focuses on creating a high-quality, comprehensive map and closing loops to ensure accuracy

INSTALLATION GUIDE

Prerequisites:

RViZ setup

Ignition Gazebo Fortress already installed

INSTALLATION GUIDE

Installing slam_toolbox and Nav2:

```
sudo apt install ros-humble-slam-toolbox ros-humble-navigation2 ros-humble-nav2-bringup
```

INSTALLATION GUIDE

For WSL2 Users

Setting Environment Variables:

```
export LIBGL_ALWAYS_SOFTWARE=1
export MESA_GL_VERSION_OVERRIDE=3.3
export MESA_GLSL_VERSION_OVERRIDE=330
```

INSTALLATION GUIDE

For WSL2 Users

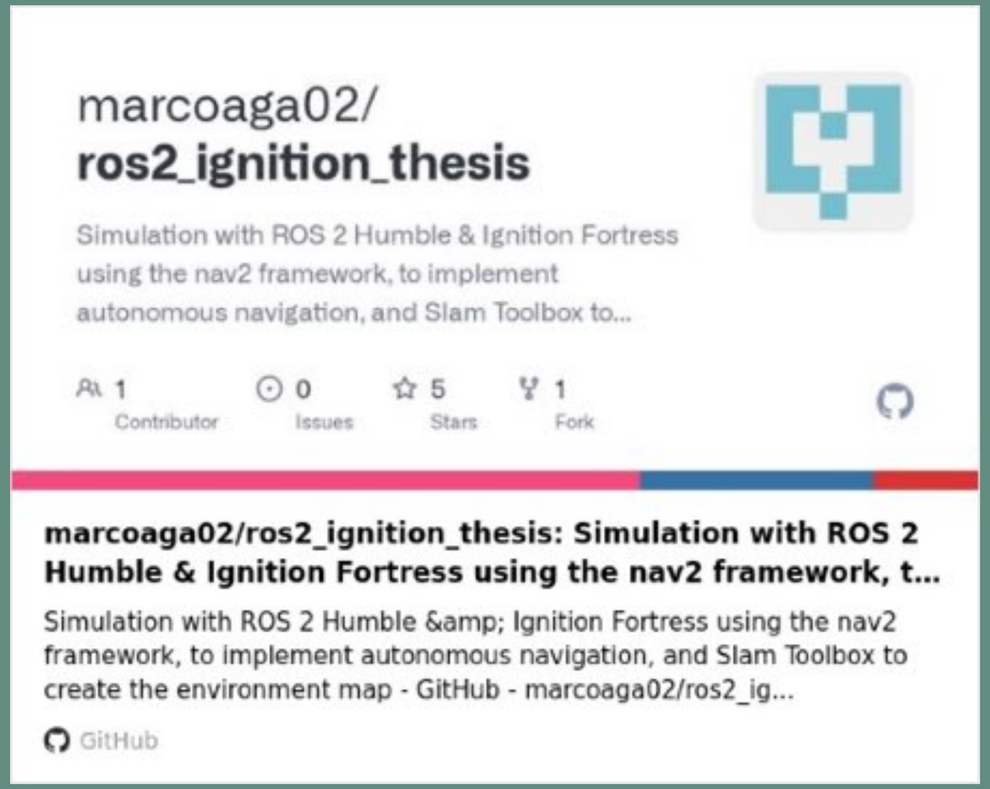
Setting Environment Variables:

```
echo "export LIBGL_ALWAYS_SOFTWARE=1" >> ~/.bashrc
echo "export MESA_GL_VERSION_OVERRIDE=3.3" >> ~/.bashrc
echo "export MESA_GLSL_VERSION_OVERRIDE=330" >> ~/.bashrc
```

SIMULATION DEMO

Watch a simulation

SIMULATION DEMO



PATH PLANNING IN DETAIL

A. Global vs Local planners

PATH PLANNING IN DETAIL

A. Global vs Local planners

1. Global planner → generates a high-level, long-term path on a known map

PATH PLANNING IN DETAIL

A. Global vs Local planners

1. Global planner

2. Local planner → uses real-time sensor data to navigate the robot along this path, dynamically avoiding immediate, previously unknown obstacles and making adjustments to avoid collisions

PATH PLANNING IN DETAIL

The global planner provides the "*big picture*" route, and the local planner handles the "*in-the-moment*" path execution and obstacle avoidance.

PATH PLANNING IN DETAIL

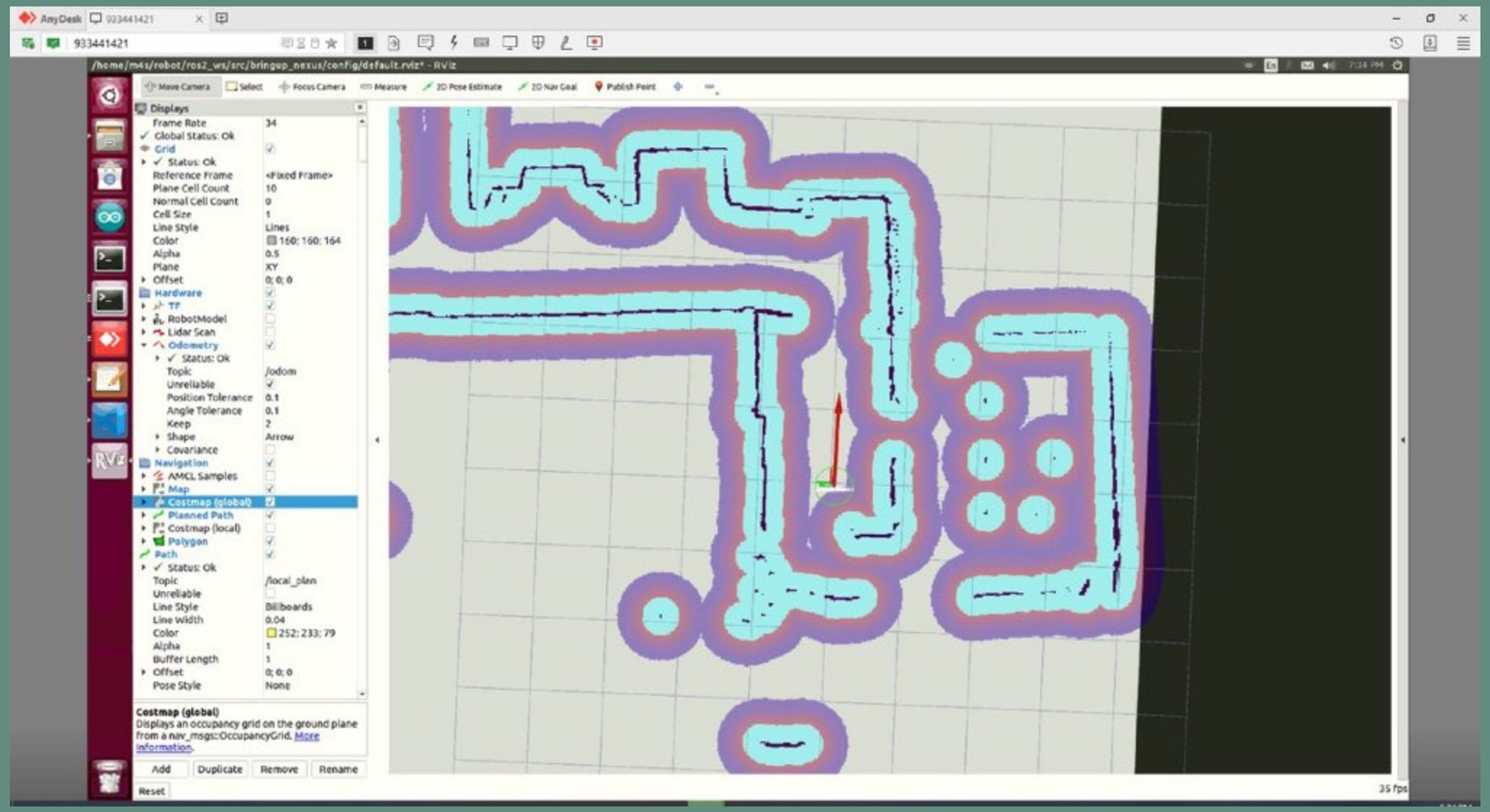
Helps us generate a costmap

PATH PLANNING IN DETAIL

The costmap holds information about the environment, such as obstacles or areas that the vehicle cannot traverse.

To check for collisions, the costmap inflates obstacles using the inflation radius specified in the CollisionChecker property.

PATH PLANNING IN DETAIL



PATH PLANNING IN DETAIL

B. Algortithms

PATH PLANNING IN DETAIL

B. Algorithms

1.Dijkstra's Algorithm:

- Finds the shortest path between nodes in a graph
- Can be computationally expensive in large environments.

PATH PLANNING IN DETAIL

B. Algorithms

2. Sampling-Based Algorithms:

- Rapidly- exploring Random Tree (RRT) and its variants
- Probabilistic Roadmaps (PRM)

PATH PLANNING IN DETAIL

B. Algorithms

3. Intelligent Algorithms (Inpired by Bionics):

- Ant Colony Optimization
- Dolphin SLAM

PATH PLANNING IN DETAIL

B. Algorithms

4. Optimization-Based Algorithms:

- Trajectory Optimization
- Time Elastic Band
- Greedy Optimization

RVIZ PLUGIN SETUP

Launch RViZ by typing "rviz2"

AMCL

- AMCL- Adaptive Monte Carlo Localization
- A probabilistic algorithm that uses a particle filter to estimate a robot's position and orientation (pose) on a known 2D map by matching laser scan data from sensors like LIDAR with the map.

SLAM_TOOLBOX VS AMCL

- AMCL → A localization-only system that uses a pre-existing map to determine a robot's position

SLAM_TOOLBOX VS AMCL

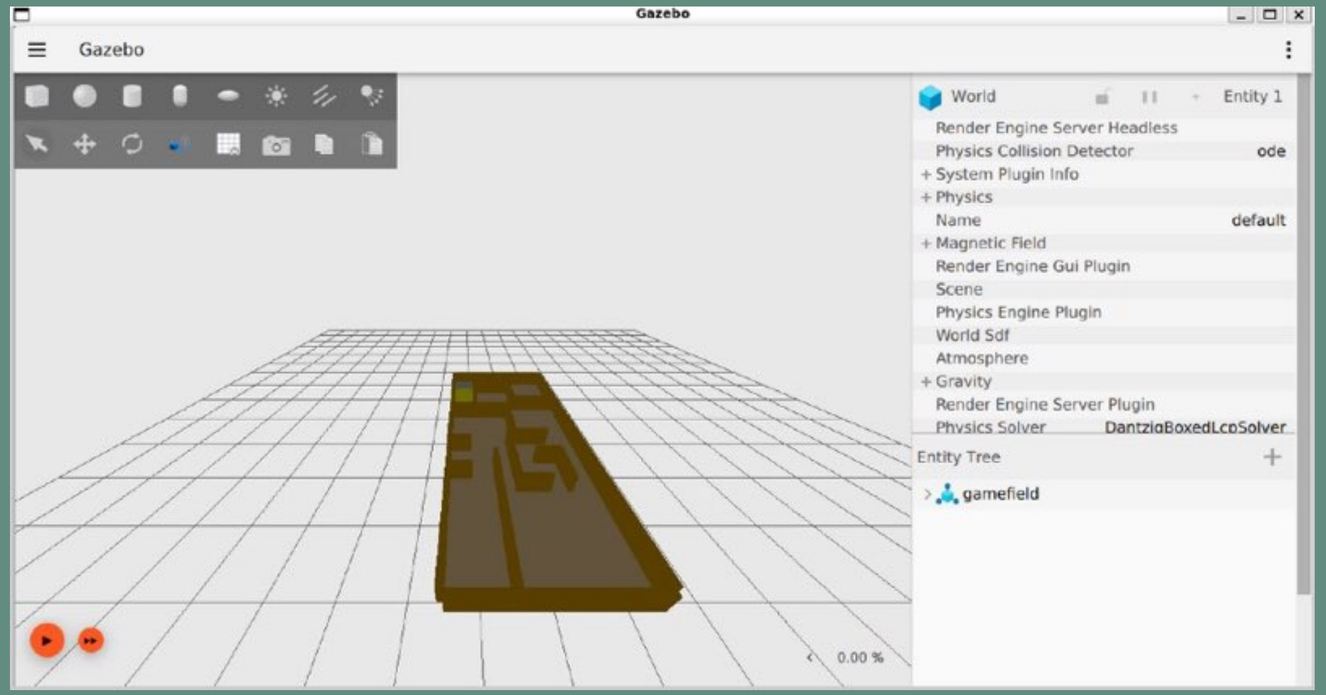
- AMCL → A localization-only system that uses a pre-existing map to determine a robot's position
- Slam_toolbox → system that creates a map and localizes the robot within it simultaneously, and can also be used for long-term mapping and updating existing maps

SLAM_TOOLBOX VS AMCL

In essence, AMCL needs a map to work, whereas SLAM_Toolbox builds the map itself.

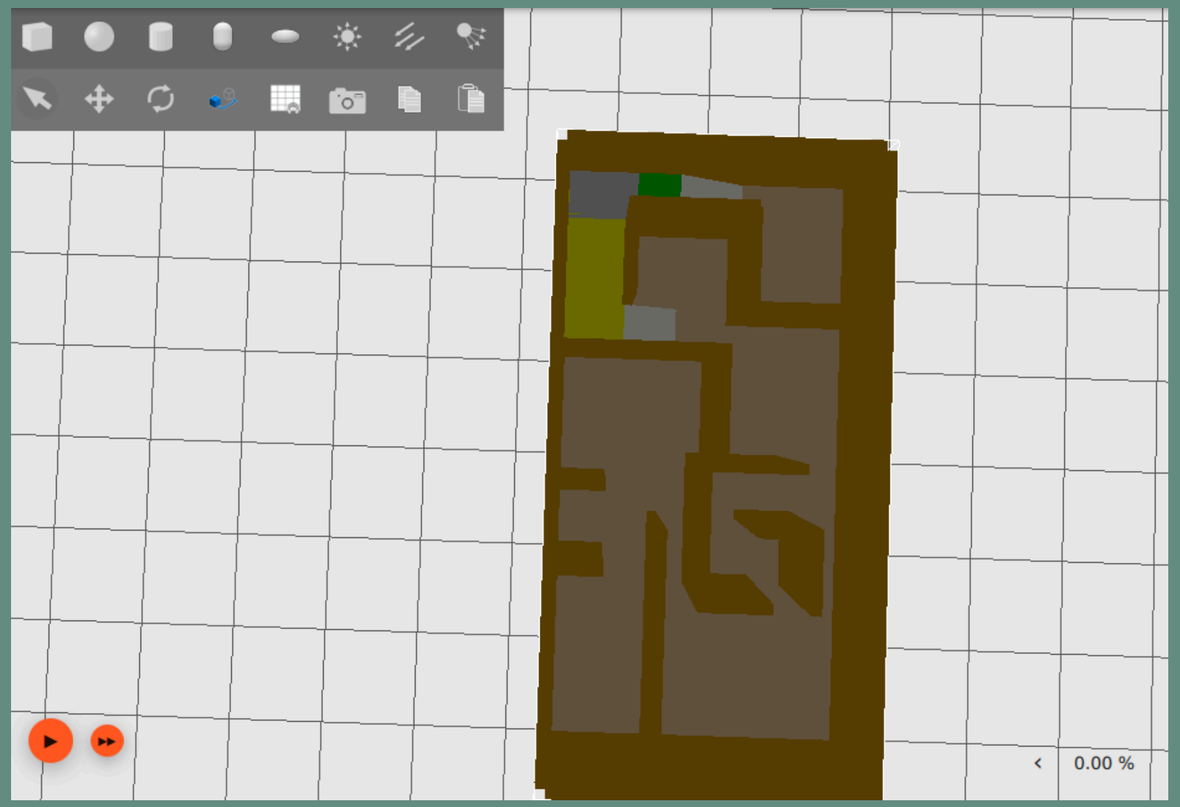
NEXT STEPS

Sharing the gamefield world next week



NEXT STEPS

Sharing the gamefield world next week



RESOURCES

- Rviz Setup
- Intro to Gazebo Fortress
- Intro to SLAM
- Intro to Navigation
- Worlds in simulation
- Why simulate

Thank You

Presented by Victoria Rotich