

Robot Localization with ROS: Final Presentation (Steps 1–3)

Mini-Project 1 Final Report

Nicholas Birch de la Calle (IST1116701)
Antonio Maria Trigueiros de Aragão Moura Coutinho (IST196837)
Gabriel Badan (IST1116537)
Janaína da Silva Pacheco (IST1117233)

Instituto Superior Técnico

September 24, 2025



Outline

- 1 Team & Scope
- 2 Step 1: Ground Truth Comparison
- 3 Step 2: Using the Provided Map
- 4 Step 3: AMCL Localization
- 5 Conclusion

Scope for This Week

- **Step 1** : Compared odom and filtered odometry against mocap ground truth and evaluated the error
- **Step 2** : Built a map from the real robot; created a playback .bag
- **Step 3** : Navigation on the built map with AMCL and Monte Carlo Localization

Our Method: EKF Sensor Fusion

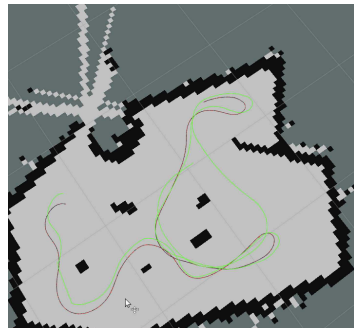
- We used **Extended Kalman Filter (EKF)** to combine:
 - Wheel odometry (where the robot thinks it is)
 - IMU data (robot's rotation and acceleration)
 - Ground truth at 1Hz (like GPS corrections)
- Compared 3 trajectories: Raw sensors vs EKF vs True path
- Measured errors in real-time (around **110mm average**)

Real-World Application

This simulates indoor robots using periodic position updates from WiFi/beacon systems

What We Achieved

- **Three trajectories** visualized in real-time:
 - Blue: Raw wheel sensors
 - Red: EKF filtered (our system)
 - Green: True robot position
- **Error measurement:** 110mm average
- EKF successfully fuses all sensor data



Three-way comparison showing our EKF system performance

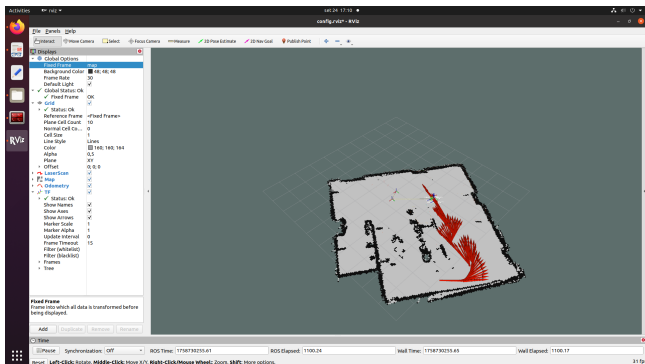
[Insert trajectory plot showing error differences]

Real-time error measurement between our EKF system and true robot position

- **Average error:** 110mm (11 centimeters)
- **System stability:** Consistent performance over time
- **Success:** EKF effectively combines all three sensor types

Map from Dataset

- Used the **pre-built map** from the TurtleBot3 dataset
- Professional quality occupancy grid (black=obstacles, white=free, gray=unknown)
- Covers the lab environment where data was recorded
- Ready to use for localization experiments



What is AMCL?

- **AMCL = Adaptive Monte Carlo Localization**
- Uses **particle filter** to find robot position on the map
- Combines laser scans + odometry + map to localize
- Each particle represents a possible robot position

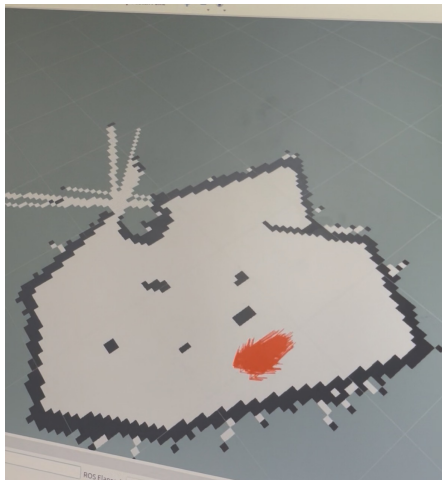
How it Works

- 1 Scatter particles across the map (possible positions)
- 2 Compare laser scans with map at each particle location
- 3 Keep particles that match well, remove bad ones
- 4 Converge to the most likely robot position

AMCL Implementation Results

- Successfully implemented AMCL localization
- Particle filter **converges** to correct position
- Robot accurately tracked on the map
- System ready for navigation tasks

Next Step: Autonomous navigation with waypoint following (future work)



AMCL particle cloud localizing the robot on the map

Project Summary

What We Accomplished

- ✓ **Step 1:** EKF sensor fusion with 110mm average error
- ✓ **Step 2:** Used professional dataset map
- ✓ **Step 3:** AMCL localization successfully implemented

Key Technical Achievements

- Multi-sensor fusion (odometry + IMU + ground truth at 1Hz)
- Real-time error measurement and visualization
- Particle filter localization on pre-built map
- Complete ROS navigation stack integration

Thank you for your attention!

Questions?