Robot Localization with ROS: EKF-based Sensor Fusion Implementation

Mini-Project 1 Progress 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 18, 2025

Outline

1. Team & Scope

2. Project Overview

3. Status & What's Next

Team Workflow & Scope for This Week

- All teammates use macOS; VMs made Wi-Fi connection to the robot tricky.
- We split work to move faster:
  - **Group A:** Fix connectivity to the real robot (prep for mapping in Step 2).
  - **Group B:** Use the dataset (.bag) and implement Step 1 (EKF with `robot_localization`).
- Collaboration: shared notes, common checklist, quick pair-debug sessions on TF and timing.
- **This presentation:** Focus on Step 1 progress with the dataset. Step 2 will follow next week.

Project Objectives

Main Goal

Implement and test robot self-localization using Extended Kalman Filter (EKF) based localization with ROS `robot_localization` package

- **Platform:** TurtleBot3 Waffle Pi with real sensor data
- **Sensors:** IMU, wheel odometry, laser scanner
- **Method:** EKF-based sensor fusion
- **Data:** Pre-recorded rosbags with ground truth from Motion Capture System
- **Framework:** ROS Noetic environment

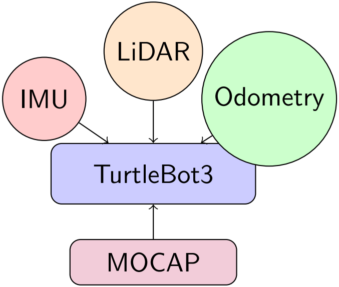
Key Learning Outcomes

Understanding Bayesian filtering, ROS navigation stack, and practical sensor fusion implementation

Dataset Information

What's inside:

- Wheel odometry (`/odom`)
- IMU measurements (`/imu`)
- Laser scans (`/scan`)
- Ground truth in TF (`mocap` but not used this week yet)
- Camera topics are present but not used this week



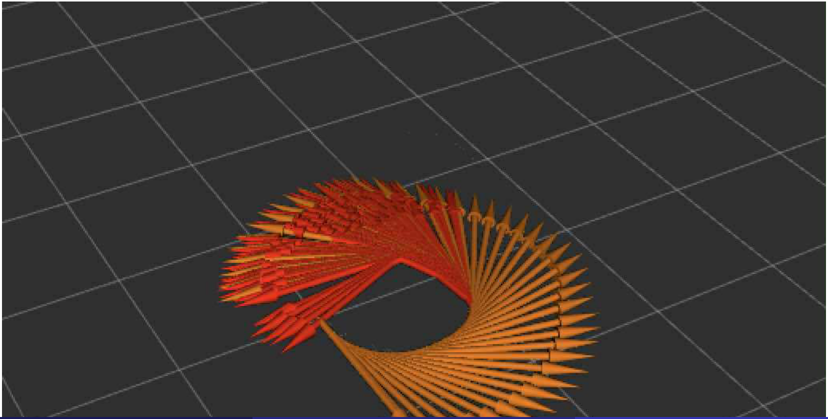
Dataset source:
`https://github.com/irob-labs-ist/turtlebot3_datasets`
This week we did not use the helper TF script and we did not compare to mocap yet.

Status and Next Steps

- **Status:** EKF running on the dataset; we compared `/odom` (orange) vs `/odometry/filtered` (red) in RViz.
- **Observation:** There is a small, slow drift between the two trajectories.
- **Next:** Extract mocap ground-truth trajectory and quantitatively compare against both.
- **Coming up:** Map-based trajectory comparison and gmapping with the real robot (Step 2).

RViz Trajectory Comparison

- **Fixed Frame:** `odom`.
- **Odometry (orange):** `/odom`
- **Filtered odometry (red):** `/odometry/filtered`
- We did not use the map-based trajectory yet, that's for next week.
- We can see a slow drift between the two. To know which is more accurate, we'll extract the mocap ground-truth trajectory next week and compare.



Final Remarks

Progress Summary

Successfully established a working EKF setup with visual comparison between odometry and filtered odometry. We kept the analysis simple and focused on Step 1.

Next Session Goals

1. Extract mocap ground-truth trajectory and align frames
2. Quantitatively compare mocap vs `/odom` and `/odometry/filtered`
3. Add map-based trajectory comparison
4. Prepare for Phase 2: SLAM with gmapping

Thank you for your attention!

Questions?