Automation of Robot Navigation with ROS and MPC
==

Overview

This project implements an automated navigation pipeline for a differential-drive
robot using ROS (Robot Operating System) and Model Predictive Control (MPC).
The system allows a user to select a target site on a map. The automation then
computes and executes a safe and efficient path to the target.

Key Components
----
1. **User Interface**
   - The user selects a target location on a digital map (e.g., via a GUI or web-based
interface).
   - The target coordinates are sent to the automation system.

2. **Path Planning Service**
   - Computes a feasible path from the robot's current location to the target site.
   - May use algorithms like graph search or geometric planners.
   - Outputs a reference trajectory that respects obstacles and environment layout.

3. **Model Predictive Control (MPC)**
   - Uses a dynamic model of the robot to optimize control inputs (wheel velocities).
   - Considers prediction horizon, system constraints, and path-following objectives.
   - Continuously updates control commands based on real-time feedback.

4. **ROS Client and Server Nodes**
   - **Server (`lhd_server.py`)**: Provides services to compute paths and control
commands.
   - **Client (`lhd_client.py`)**: Interfaces with user input and sends goals to the
server.
   - Communication is handled using ROS topics and services.

5. **Execution Layer**
   - The computed control signals are sent to the robot actuators (or a simulator).
   - The robot moves along the path while the system monitors progress.

Automation Flow
----
1. User selects a target site on the map.
2. Target coordinates are sent to the path planning service.
3. Path planner computes a trajectory and passes it to the MPC controller.
4. MPC computes optimal control commands considering robot dynamics and constraints.
5. ROS nodes publish commands to the robot.
6. Robot navigates autonomously to the target while the system continuously monitors and
corrects.

Benefits of This Automation

- Human operator only needs to specify the goal, not the detailed motion.
- Optimized trajectories increase safety and efficiency.
- ROS modularity enables integration with sensors, simulators, or real robots.

Conclusion

This automation framework demonstrates how modern robotics integrates user interfaces, planning algorithms, and optimal control. By allowing goal selection on a map and delegating the navigation to an MPC-driven ROS system, the robot can navigate efficiently with minimal human intervention.