

Language-Enabled Robotic Control Platform

Problem Statements

- **Robots lack autonomous decision-making:** Today's robots can execute only the commands we give them step-by-step. They cannot "think" or plan on their own. For example, a robot vacuum can clean but must be told exactly where and how to move. Our project adds an AI "brain" (a GPT-like language model) so the robot can interpret a high-level task ("clean the kitchen") and figure out the necessary actions. This addresses the limitation that traditional robots require rigid instructions, whereas an LLM can translate a single natural-language command into a sequence of robot movements and operations ¹ ².
- **Complex robot tools are hard for non-experts:** Even a skilled software developer may struggle to use robot simulators like Gazebo or ROS without specialized knowledge. Our platform provides a simple interface so anyone can command the robot in everyday language, without learning Gazebo's technical details. In other words, a person who doesn't know robot programming can still tell the robot what to do in plain words. This is similar to recent frameworks that let non-experts program robots via chatbots ³. By hiding the complexity of the simulation tools behind a natural-language interface, we make robot control accessible to users who aren't robotics specialists.

Background and Literature Review

Large Language Models (LLMs) like GPT-4 or Claude can understand human instructions and break them into actionable steps. In robotics, an LLM acts as a high-level planner: it takes a natural-language command (e.g. "go to the kitchen and fetch a cup") and generates a sequence of robot actions. The embedded infographic shows this concept: an LLM receives a user's command, decomposes it into sub-commands (like "navigate forward" or "pick up object"), and issues those to the robot. Crucially, the LLM communicates with the robot through a set of **tools** or functions (for example, "navigate to location", "start mapping", etc.) that the robot exposes. Surveys have noted that LLMs can serve as intelligent agents in robot systems by "*interpreting user intent, generate plans, [and] invoke robot APIs*" without replacing the underlying robot software ² ⁴. In practice, the LLM's output triggers existing robotics modules (navigation, mapping, manipulation, etc.) inside the simulation.

Researchers have found that integrating LLMs into robotics can greatly improve human-robot interaction and autonomy ⁵ ². For example, robots can learn to understand varied natural-language instructions and personalize their actions, instead of relying on fixed scripts ⁶. Early projects have demonstrated LLMs planning complex tasks: one university demo used an LLM (Anthropic's Claude) connected to a Gazebo-simulated house robot, allowing a user to tell the robot to "navigate to the kitchen" or "explore the map" in plain language ⁷ ⁴. In that demo, the LLM successfully translated "navigate to the kitchen" into a navigation goal, and the robot planned and followed a path to the kitchen (a red path shown in RViz) ⁴. Similarly, the LLM could start autonomous exploration and use SLAM to map the house ⁸. These examples show how exposing robot capabilities (navigation, SLAM, sensor reading, etc.) to the LLM lets it control the robot without the user writing code.

At the same time, surveys warn of challenges. LLMs require large compute resources and usually run in the cloud, so a robot must stay networked and cannot run the model locally ⁹ ¹⁰. They can also “hallucinate” or suggest incorrect actions if not carefully managed. In one survey, authors note LLM-based systems must address issues like grounding instructions in the real world, memory of context, and safety of generated commands ¹⁰ ¹¹. In short, current literature shows that LLMs *greatly* expand how we can command robots – turning language into motion – but also that ensuring reliability and efficiency remains an open challenge.

Proposed Project Name

- **RoboLingua** – emphasizes the fusion of robotics and language understanding.
- **LLM Navigator** – highlights the LLM-driven navigation of robots.
- **SmartSim Robot** – suggests an intelligent simulator.

A strong choice is “**RoboLingua**”, which conveys that the robot follows human language instructions. This name reflects our goal of using a language model to guide robot actions.

Related Work and Research Gaps

Existing Implementations: Recent projects have explored similar ideas. The demo above (Mike Hart *et al.*) shows an LLM guiding a simulated robot in Gazebo via ROS ⁷. The LLM had access to numerous “tools”: it could command movement, start SLAM mapping, read sensors, and so on ¹² ⁴. In academic research, **ROS-LLM** is a framework that connects chat interfaces with ROS, so non-experts can program robots through dialogue ³. For manipulation tasks, the **MALMM** system splits one big LLM into three specialized agents (Planner, Coder, Supervisor) to coordinate a robot arm ¹³ ¹⁴. For multi-robot coordination, the **DELIVER** system combines an LLM (LLaMA3) with Voronoi-based planning: given a human instruction, it uses the LLM to identify pickup/dropoff locations and then coordinates multiple robots via ROS/Gazebo to carry out delivery tasks ¹⁵. Survey papers also categorize these efforts as part of an emerging “agentic AI” approach, where LLMs interface with robot middleware like ROS to extend flexibility ² ¹⁶.

Gaps and Challenges: Despite these advances, several gaps remain for our project. First, many systems have only been tested on simple or pre-defined tasks. For example, MALMM handles block-stacking with a robot arm ¹³, but multi-agent LLM coordination for navigating a simulated home (with mapping and vision) is still new territory. Second, LLMs can “hallucinate” – that is, propose impossible or unsafe actions. The MALMM paper notes this problem and addresses it by adding supervisor agents and step-by-step feedback to recover from errors ¹³. We must similarly design safeguards in our system. Third, performance and reliability: because LLM inference is remote, any network issue can disrupt control ⁹. In one study it was noted that if the robot’s internet link fails, it cannot get instructions from the LLM ⁹. We need robust handling for such issues. Fourth, grounding and context: surveys emphasize the need for grounding language in the actual environment and maintaining memory of tasks ¹¹. Our platform should keep track of what the robot has seen (its map, objects, etc.) so that the LLM’s high-level plans stay consistent with reality.

Finally, tooling and integration remain an open area. We plan to expose functions like navigation (`navigate_to(location)`), mapping (`start_exploration()`), etc. to the LLM, as others have done ¹² ¹⁷. But deciding which tools and how to present them to the LLM is still an under-explored design choice. In summary, existing work shows the promise of LLM-guided robot control (from single-robot

demos ⁷ to multi-agent planners ¹⁵), but our project must address the remaining gaps: multi-agent coordination in complex environments, error recovery, network/deployment constraints, and aligning the LLM's reasoning with the robot's sensors and world model ¹³ ¹¹.

Sources: Research on LLMs in robotics ⁵ ² and recent demonstrations ⁷ ⁴ underpin this review. We also draw on specific works like MALMM ¹³ and DELIVER ¹⁵ for examples of multi-agent LLM control, as well as frameworks for user-friendly robot programming ³ ⁴. These sources highlight both progress and open challenges in our project's domain.

¹ ⁴ ⁷ ⁸ ⁹ ¹² Controlling Robots using a Large Language Model | Mike Likes Robots

<https://mikelikesrobots.github.io/blog/llm-robot-control/>

² ¹¹ Towards Embodied Agentic AI: Review and Classification of LLM- and VLM-Driven Robot Autonomy and Interaction

<https://arxiv.org/html/2508.05294v1>

³ ¹⁷ ROS-LLM: A ROS framework for embodied AI with task feedback and structured reasoning

<https://arxiv.org/html/2406.19741v3>

⁵ ⁶ ¹⁰ [2311.07226] Large Language Models for Robotics: A Survey

<https://arxiv.org/html/2311.07226v1>

¹³ ¹⁴ MALMM: Multi-Agent Large Language Models for Zero-Shot Robotic Manipulation

<https://arxiv.org/html/2411.17636v2>

¹⁵ DELIVER: A System for LLM-Guided Coordinated Multi-Robot Pickup and Delivery using Voronoi-Based Relay Planning

<https://arxiv.org/html/2508.19114v1>

¹⁶ Multi-Agent Systems for Robotic Autonomy with LLMs

<https://arxiv.org/html/2505.05762v1>