Fall 2025: Human-Robot Collaboration and Companionship Lab Report

Jonathan Distler, Cornell MAE '27

***Abstract*—This report is the final deliverable for an ongoing research project for the HRC^2 lab as part of a research-for-credit initiative. The goal of the project was to mediate tasks between a human and robot agent using a LLM. The report outlines an introduction to the research including a motivation for the use of LLMs; the research including setups; outcomes of my contributions to the project; a reflection on the process over the Fall 2025 semester.**

I. Introduction

Human-Robot Interactions involve understanding the complex interactions between humans and robots in shared environments, given unique constraints placed on both the human and robot agents (i.e. humans limited load bearing capabilities and robots limited manipulation capabilities). Having a mutual understanding of goals, roles, and task decompositions is crucial in successful teams. As is the case with human-human teams, communication plays a vital role in shaping each agent's mental models with respect to plans, goals, and constraints. In human-human teams, successful communication can be accomplished through verbal and non-verbal communication (i.e. eye contact) and the ongoing dialogue is a prerequisite for restrategizing throughout a task. However, robots typically haven't been built, trained, or programmed to invoke these dialogues.

Given the prevalence of artificial intelligence (AI) across many domains of society, previous researchers have attempted to leverage Large Language Models (LLMs) to solve long-horizon tasks, often involving multiple agents [1]–[3]. Many studies have used LLMs as a stand-in for dialogue on behalf of robots in order to facilitate mutual understanding among agents [4], [5].

This project aims to use LLMs to mediate a long horizon task between a human and robot; delegating tasks among agents based on constraints (including willingness to complete a task on each agent's end) and goals for the task. The LLM will produce a plan for each agent, which will be converted in to a PDDL. The PDDL will then be solved for each agent, showing the step by step tasks for each agent, at which point each agent will attempt to complete their respective tasks.

II. Research

A key component of this study was understanding the collaboration between robots and humans, and the development of a shared mental model among agents. This is accomplished via the delegation of tasks between each agent based on their respective physical constraints and environmental constraints, and in the case of the human agent, their respective wants. The interplay between each agent with their communication and updated and shared body of knowledge was an important aspect, as was translating those tasks into successfully completing a task.

The goal of the project was to take a natural language description and pass the description to an LLM. The LLM would then break the problem down into subgoals; for each subgoal, the agent (one of either a human or robot) is given the provided subgoal and turns it into a relevant PDDL with the help of a LLM. Although not implemented in our current project, with the newly devised PDDL setup (domain and problem file), a PDDL algorithm (i.e. fast-forward heuristic) would solve the PDDL problem and pass the output to the main agent to execute.

There was an LLM chatbot that took user inputs such as task goals (i.e. making a sandwich), human and robot constraints (i.e. human has a broken hand and the robot has a height limit), and environmental constraints (i.e. a chair blocking the only pathway towards the ingredients for the robot which it cannot move). The LLM chatbot provides a visual output to the user, showing the step-by-step goal for each agent, and the shared knowledge base, analogous to the aforementioned mental models ubiquitous in human-human teams. The LLM produces a plan between the agents, which can be rejected or accepted by the human, along with user input. For example, if the human has too few tasks. They can reject the proposed plan and provide input indicating that they would prefer to have more tasks or specific tasks. Then, the LLM would reformulate a plan until a plan is agreed upon by the human. After the LLM produces a plan agreed on by the human, the output plan is converted into PDDL and solved using any one of a sequence of PDDL solvers.

A. PDDL

As for my work, my first goal was to design and a domain and planner for a PDDL (Planning Domain Definition Language) as a part of a small scale system, to mimic the larger system of future tasks. Then, to take the hypothetical solved-PDDL output and convert that to primitive tasks which the robot could accomplish, based on the required actions and locations of the objects.

Figure 1 shows an example PDDL output for a single task, composed into three distinct actions.

- "drive-robot stretch a b"

The script would recognize the primitive as "drive" and the locations as "a" and "b" which could represent any

This report was produced by Jonathan Distler
Written December 4, 2025

```
* Starting ASTAR search with FAST_FORWARD heuristic
* ASTAR search succeeded

found plan as follows:

0: (                    drive-robot stretch a b) [0]
1: (load-robot bottle stretcharm stretch b heightlo) [0]
2: (  drive-robot-package stretch b tableloc bottle) [0]

time spent:       0.04 seconds parsing
                  0.07 seconds encoding
                  0.01 seconds searching
                  0.11 seconds total time

memory used:      0.23 MBytes for problem representation
                  0.00 MBytes for searching
                  0.23 MBytes total
```

Fig. 1. Simplified PDDL Output for a Pick-and-Place Task

number of metaphorical locations (i.e. the pantry, the stove-top, specific object locations, etc.). Then, the robot would map the predefined verb to an action; for example, ”drive” represents a forward movement of the base as defined in the Stretch Robot class.

B. Global Coordinates

My next task, my goal was to string together primtive tasks in order to accomplish a long horizon task. The robot I used, *Hello Robot Stretch 2*, had some predefined functionalities such as moving the base, rotating the base, moving the arm up and down, and telescoping the arm in and out. However, the Hello Robot class only specified relative differences between the current and goal positions, and it was my job to define those differences based on a coordinate system. This first involved defining a global coordinate system projected onto the room, with each object being given a position relative to the semi-arbitrarily defined origin; the position of the robot was defined, too. All are in the following script: Global Position Coordinates Matlab Script.

C. Linear Interpolation

For my next step, I intended to write a script for the robot that uses the predefined Hello Robot class functions to generate movement patterns. A key issue was that when the robot rotated, its center-of-mass would translate horizontally, likely as a result of rolling-and-slipping, which prevented the robot from staying on the same path in the event of a rotation. I determined the average offset for a 90-degree rotation and developed an interpolation method [6] that calibrated the translation of robots base after rotation to make up for the unintentional offset.

After these basic functionalities were implemented I could string together multiple functions for longer-horizon tasks. For example, I passed the arguments for the robot’s current state vector, the position of an object, and the drop-off location for the object. Then, the function would calculate the distances and pairs of rotations needed to travel between the robot and the object, the movements needed to pick up the object, and then the movements needed to move in the environment from the object’s location to the drop-off location, before maneuvering the arm and dropping off the object. The code is linked here: Robot Functionality.

III. Outcomes

I was largely unsuccessful in developing an algorithm to take the PDDL output and navigate around the room in order to pick-and-place objects. I hadn’t factored in the constraints of the table-legs on picking-and-placing objects. During testing, as the robot approached the table to pick up an object, its base would collide with the table, or if I tried to extend the arm out more and keep the base further from the table, the arm would spin and hit unintended obstacles, or the base would still make contact, even with differentiated tables. The following represents a sample output of the situation in which it ran into the obstacles, but did have the correct sequence of actions (Terminal Output) I was able to run a pick-and-place task by specifying a start and end location, however, anything more than a pure translation with a single rotation resulted in crashes.

- Test Idealized Example: Successful Action
- Test Failure Example: Running into Table During Test
- Test Failure Example: Running into Suspended Plank

IV. Reflection

As noted in my research goals for the Fall Semester, I was intending to accomplish the following, as well as trying to have a working demo of the robot completing a pick-and-place task:

- Get more familiar with using robots - a better understanding of using ROS and using robot planning algorithms for human-robot collaboration tasks.
- Work on my understanding of VLMs and how they can help robots plan in human-robot collaboration tasks.
- Learn to design user study tasks and run human-subject experiments.

A. Insights

My biggest insights into the research project and research in general came from the process. I’d been very quick to jump into a project and start working on a solution immediately without a clear goal in mind. I tried that on this particular research project to no avail. To really understand the problem and all of its underlying complexities one has to read a lot of targeted literature. My quickness ended up costing me a lot of time on the project because I didn’t completely understand the problem and I had no context on how other people had worked on similar problems.

Additionally, I gained a lot of insight into how successful teams operate. In previous experiences, meetings were intended to start and finish very quickly to save everyone time in the short-term. I think that having set time limits on the weekly meetings, having targeted questions, and being put on the spot at the meetings really helped improve my personal understanding of the project. Even though the meetings took longer than I was used to, it saved me a lot of time and frustration because I was able to hash out any ideas I had, and I was able to gauge my understanding of the topics I was working on based on how well I could answer questions from my colleagues.

B. Learning Outcomes

My first goal for the semester was to get more familiar with using robots. I had a semester to work alongside the Stretch 2 robot and learn about the technicalities of hardware, such as the necessity of frequent calibration, the awkwardness of end-effectors, and how advertised metrics on a system aren't truly representative of the robots capabilities throughout differentiated tasks (i.e. the Stretch 2 robot was rated to reach its arm up roughly 1m off of its base, however, with a payload it wasn't able to get more than a fraction of that idealized height). I also became familiar with the fact that there are implicit constraints to the robot, for example the coupling between its arm and base, which limits the true functionality of the robot, especially in pick-and-place tasks.

My second goal was to get more familiar using the Robot Operating System (ROS) framework. I had some previous knowledge on the system and a general understanding of topics. However, after being asked to explain the system in a lab meeting I gauged that I really didn't understand as much as I thought I knew. I tasked myself with settings up a publisher and subscriber independent of the research project, as well as reading more literature on Pub-Sub architectures. This really helped me understand the operating system, and I was able to successfully debug RVIZ issues and I could competently use ROS for a variety of tasks related to the Stretch Robot.

My next goal was to learn how to design user study tasks and run human-subject experiments. Throughout my time on the project I iterated over how to conduct literature reviews successfully with the advise of Dr. Hoffman and Dr. Tabrez, prioritizing a few very pertinent research studies (and their respective websites and Github pages) over a plethora of semi-relevant research studies. Through these targeted reviews, I was able to understand how many other researchers conducted user study tasks and human-subject experiments. Then, throughout our work on the project, I was able to improve my understanding on the topic by working on a very small-scale setup with our team.

My final goal was a functional pick-and-place task. I was able to code a pure translational solution to the pick-and-place task, but it wasn't robust, it was only along one dimension, so it didn't have very much complexity. The main failure in my pick-and-place task came from not accounting for the constraints of the table legs, as well as the tight-environment the robot was navigating through. Had I been able to implement the FunMap, I could easily avoid the obstacles and would more than likely have been able to complete the task successfully. Also, the task may have been too difficult for the robot to accomplish with a higher degree of difficulty because of the coupling between the arm and the base (the arm was always aligned with the front of the robot), and the initial offset of the arm.

C. Roadblocks

My biggest roadblocks during the semester were a result of trying to implement a Hello Robot specific package known as FunMap [7] (a derivation of the frequently used SLAM architecture). The goal of FunMap was to map the room such that the robot could navigate to specific positions without explicitly having to code each action based on positional differences. For example, instead of specifying the Stretch robot's base to move forward 1.5m, turn 90-degrees, move forward .25m, one could just specify the endpoint and the Stretch robot would navigate all obstacles to reach the position. This would robustify the pick-and-place tasks as a part of the project and allow for more complex movements with quicker times.

The main issue in implementing the package was a Numpy version issue. I felt tentative about reaching out to other people on the team because I thought it should be a trivial solution that I should have been able to figure out on my own, but asking for help would have helped me solve the problem given their expertise on the subject. Throughout the time trying to debug the issue, I also realized that I lacked a true debugging methodology. Going forward I plan on trying to solve these issues with more understanding of the error messages and a better understanding of the library's versions, rather than blindly copying and pasting code from related forums as I had been doing this semester.

D. Future Improvements

In the future, I hope to implement the FunMap to robustify the pick-and-place tasks, making sure to avoid base contacts, allowing for more interesting human-robot interactions. I also plan on reaching out to my teammates more, so they can better help me understand the software aspect of the work. Similarly, I also plan on trying to stay in the loop with the other projects more to see if I can draw parallels across the research projects, and also improve my knowledge in the field of human-robot collaboration.

References

[1] H. Wang, K. Kedia, J. Ren, R. Abdullah, A. Bhardwaj, A. Chao, K. Y. Chen, N. Chin, P. Dan, X. Fan, G. Gonzalez-Pumariega, A. Kompella, M. A. Pace, Y. Sharma, X. Sun, N. Sunkara, and S. Choudhury, "MOSAIC: Modular Foundation Models for Assistive and Interactive Cooking," arXiv:2402.18796, 2024.

[2] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "ProgPrompt: Program Generation for Situated Robot Task Planning using Large Language Models," *Autonomous Robots*, vol. 47, pp. 999–1012, 2023.

[3] D. Bai, I. Singh, D. Traum, and J. Thomason, "TwoStep: Multi-agent Task Planning using Classical Planners and Large Language Models," arXiv:2403.17246, 2024.

[4] J. Ivey, S. Kumar, J. Liu, H. Shen, S. Rakshit, R. Raju, H. Zhang, A. Ananthasubramaniam, J. Kim, B. Yi, D. Wright, A. Israeli, A. G. Møller, L. Zhang, and D. Jurgens, "Real or Robotic? Assessing Whether LLMs Accurately Simulate Qualities of Human Responses in Dialogue," Proc. ACM Conference, 2024. DOI: 10.1145/3610978.3641112. [Online]. Available: https://doi.org/10.1145/3610978.3641112

[5] A. Addlesee, N. Cherakara, N. Nelson, D. H. García, N. Gunson, W. Sieińska, M. Romeo, C. Dondrup, and O. Lemon, "A Multi-party Conversational Social Robot Using LLMs," inCompanion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction (HRI '24), 2024, pp. 1273–1275. doi:10.1145/3610978.3641112.

[6] "Linear Interpolation Formula," Cuemath. Available: https://www.cuemath.com/linear-interpolation-formula/ (accessed Dec. 6, 2025).

[7] stretch_funmap, Hello-Robot Documentation. Available: https://docs.hello-robot.com/0.2/stretch-ros/stretch_funmap/ (accessed Dec. 6, 2025).