Tree-of-Thought (ToT)-enhanced LLM System for Robot Operation in Construction: A Preliminary Study

Yizhi Liu
Department of Computer Science
Columbia University
New York, United States
yl4993@columbia.edu

Abstract – **To address the ever-existing low productivity and labor shortage in the construction industry, automation, and robotic technology have been gradually introduced. To better operate these machines, researchers developed a series of robot control mechanisms. However, most of them remain in the early stages of development and require users (workers) to possess programming or mechanical engineering expertise, limiting the practical feasibility. With recent advances in AI, Large Language Models (LLM) have been used for human-machine interaction, offering the potential for more intuitive and user-friendly robot control approaches. To this end, this project first conducted a literature review of existing robot control methods. Based on the insights from the literature review, the project investigated the feasibility of integrating ToT-based LLM into the robot control interface, aiming to create a system that allows users to use natural language for robot control. The experimental results showed the potential of the ToT-enhanced LLM system for robot control, achieving over a 70% success rate in generating feasible waypoints for robotic arms to execute tasks like bricklaying and rope moving. Moreover, this project discussed the limitations and future direction of the LLM-based robot control interface. This work can potentially facilitate the adoption of robotics in traditional industries like construction and civil engineering. The project is available on GitHub: https://github.com/liuyiz1994/COMS-E6156-TREEBOT**

I. INTRODUCTION/**SYNOPSIS**

Construction robots are being introduced to address the ever-existing challenges in the construction industry, like labor shortages and low productivity issues [1,2]. Due to their mechanical strength, precision, and speed, various robots have been deployed for a series of construction tasks. To be more specific, drones have been widely used for infrastructure inspection in recent years. Legged robots, like Boston Dynamics' Spot, have been deployed for site and progress monitoring in construction. Wheeled robots, like HUSKY A200, have been applied in material delivery and mapping of unknown environments in several cases [3–5]. Among these robots, robotic arms have emerged as one of the most accepted robots. This technology has been deployed in a wide range of tasks in construction, including rebar tying, bricklaying, and timber assembly [6,7]. One example is the Semi-automated Mason (SAM), a robotic arm-based system developed and deployed to collaborate with workers on bricklaying tasks. SAM will pick up and place the brick, and workers will clean the mortar. This robot can lay over 300 bricks per day and achieve productivity 3 times higher than manual operation. Another example is Hadrian X, a large, wheeled robot equipped with a robotic arm capable of automating the "block" laying process for structural walls, with 200 blocks per hour. All these examples show the capability of the robotic arm to help construction workers perform tasks more efficiently.

However, introducing a new technology into a field will bring new challenges [8]. While robotic arms show great potential to enhance productivity and mitigate the labor shortage in construction, their control/operation remains a challenging task. The above-mentioned arm-based robots, SAM and Hadrian X, rely on pre-programmed control and operation, which will face challenges in the dynamic and unstructured construction site, as this pre-programmed control mechanism cannot adapt to unexpected situations in practice, leading to collisions and safety concerns.

To address this, recent research has explored a series of control methods that allow human workers (human supervisors) to teach or teleoperate robotic systems. According to the literature [9], the reinforcement learning (RL)-based method has gained attention for allowing robots to learn actions based on a trial-and-error process rather than the pre-programmed mechanism. However, RL in construction faces two main limitations: first, the construction environments still have lots of unexpected and unseen situations that cannot be learned by the RL-based method. Second, the sim-to-real gap in RL will lead the policy learned in simulation to fail to perform in actual construction environments. In addition to RL, researchers developed "human-robot interfaces" that enable human workers to directly control and interact with robotic arms. Several recently explored interfaces include gesture control [10,11], wearable motion-capture suits [12], and brain-computer interfaces [13]. The main objective of these interfaces is to better allow a human to transfer their domain knowledge and skills to robot control commands in a more intuitive way (gestures, arm movements, and spatial awareness) than programming. While promising, each interface has its own advantages and limitations. Gesture-based control, for example, requires workers to understand the mechanical structure/constraints of the robot to generate mechanically feasible gestures for robot control. The motion-capture suit shows potential in "indoor" lab settings but may lack practicality in outdoor environments (since I don't think it is feasible at outdoor construction sites to capture human motion using motion-capture systems). Brain-computer interface requires the translation between workers' brainwaves into robot control, suffering from limited accuracy (less than 80%) and motion artifact issues in dynamic construction sites. In other words, the current human-robot interfaces remain at a proof-of-concept stage and are not ready for field deployment. Given these challenges, there is a clear need for a more intuitive and robust control method that allows human workers to guide robots.

To this end, I am thinking if it is possible to use a large-language model (LLM) as an interface for robot control. The LLM-based methods can allow workers to use their natural language or domain knowledge as inputs. The LLM will interpret human inputs and transfer them to robot waypoints. The robotic arm will follow the waypoints to execute the assigned task. Compared to existing interfaces, this LLM-based method will not require workers to have knowledge of mechanical engineering to operate the system [14]; it will enable workers to use only voice or text input to activate the control process (which means this method can be deployed to various environments); and allow workers to use their domain knowledge without requiring programming to control the robot (more intuitive). Based on what I learnt from midterm project, ToT-based LLM can ensure a balance between the model efficiency and the computational cost. Therefore, my final project adopts the ToT as the reasoning structure for the LLM-based robot control mechanism.

Novelty: LLM-based robot control is a very new topic. Related studies were conducted after 2024. Among these methods, most of these models based on CoT reasoning structure. This study will try to integrate the ToT into the control mechanism, ensuring the control commands more reliable and accurate.

Value to User Community: the value of this project is obvious. As stated, the construction industry plans to adopt robots to solve labor shortage and productivity issues. However, due to the low technology adoption rate in the field, most workers do not have the required knowledge in robot control, including mechanical engineering and programming. This project can help bridge this gap: the LLM can interpret workers' natural language into robotic commands, thereby workers/users only need to use their daily language to communicate with robots to perform the task. If success, this project can contribute to the existing human-robot interface, greatly enhancing workers' productivity and safety in the construction industry.

In the rest of my final report, I will first conduct a literature review of existing robot control methods, focusing on their operation mechanism, technical advantages and limitations of each method. Then, I will discuss the existing development of the LLM-based robot control method, explaining the technical advantages of this type of method over the traditional method and its limitations. Next, I will modify the existing LLM-based robot control method based on my midterm project. I will conduct a case study to test the performance of my method and discuss its results. Building upon the understanding and results gained from both my midterm and final projects, I will discuss the future direction and implications of this project. Finally, I will conclude the final project. The result of this project has the potential to facilitate automation in the current construction industry.

II. RELATED WORK

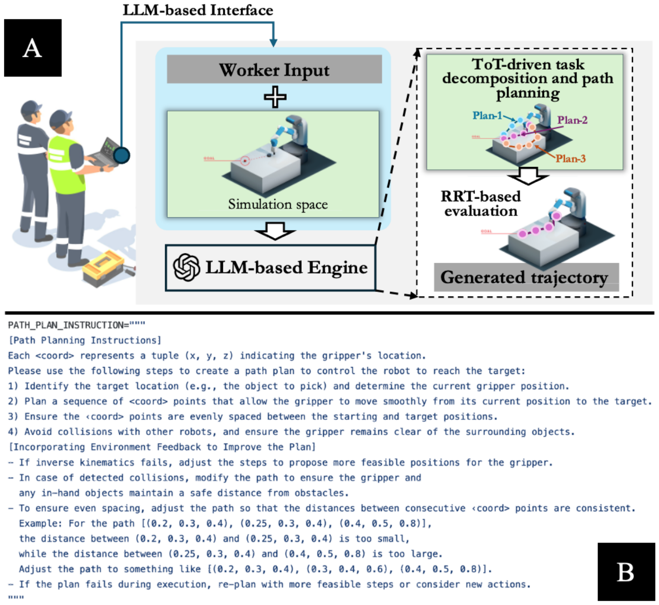
To improve how human workers control/operate robots, a series of robot control techniques have been proposed. In the early stages of robot control, methods were straightforward: workers used physical controllers to change the directions, orientations, and configurations of the robot. However, in the construction industry, this method was not feasible, as it requires workers to have domain knowledge in mechanical engineering, occupy both hands during operation, and need extensive training [13,15]. To this end, researchers have proposed other robotic control methods, aiming to enable workers to have a more intuitive and simpler way to command robots. One approach is to track human gaze movements for robotic control. Shi et al. [16] used eye-tracking devices to monitor gaze direction and fixation, guiding a robotic arm to select the controller's intended object. Likewise, Moniri et al. [16] developed a case where an object was presented in both virtual and real environments, and robots were instructed to pick up the objects upon users' gaze. Mehlmann et al. [17] stated that the gaze-based control methods establish joint attention between humans and robots, enhancing interaction efficiency. Huang and Mutlu [18] supported this idea by showing that robots capable of understanding users' gaze objects can foster improved collaboration between workers and autonomous systems. In most cases, eye-tracking technology integrated with AR/VR headsets is used to enable real-time tracking of operators' gaze direction and focus of attention (e.g., staring, blinking) to generate robotic commands [19]. However, in practice, these devices are intrusive and block workers' visibility [19], thereby introducing safety risks in the dynamic construction field.

In addition to gaze movement-based control, researchers developed human gesture-tracking methods for robotic controls. Existing studies leveraged vision- or sensor-based methods to track human body gestures and transfer the captured postures into corresponding robotic configurations [20]. Vision-based techniques use cameras to capture human gestures and then use computer vision frameworks, like 3D-ConvNets to transfer the human postures to the robot. Terreran et al. [20] developed a skeleton-based action and gesture recognition framework utilizing vision-based 3D pose estimation techniques to identify human general body actions and hand gestures for robotic controls. For the sensor-based method, human workers were asked to wear motion capture sensors/markers, and the method integrated signal decomposition modules, combined with deep networks [21,22], to enable gesture recognition. Wang et al. [21] developed a sensor-based method to track finger motions for robotic dump truck control. However, both vision-based and sensor-based methods have limitations in practice: vision-based methods are sensitive to environmental conditions (lighting, dust, and dynamic moving objects). Sensor-based human gesture recognition remains intrusive, as they require workers to wear sensor sets, which are often impractical on actual job sites [23]. Moreover, both methods require that workers understand the mechanical constraints of robotic motion to generate feasible gestures, limiting accessibility further.

Recent research has also explored biosensor-based robot control by transferring workers' brainwave signals into robotic control commands. Compared to the above-mentioned method, one advantage of these methods is to free up workers' hands and bodies, allowing workers to use brainwave signals to control robots. Liu et al. [13] developed a brain-computer interface that captured and translated EEG signals into robotic commands with up to 90% accuracy. Schaaff and Schultz [24] proposed using EEG signals to interpret human emotions for robot teleoperation. Other studies have also investigated various biosensor-based robotic controls, including robotic assembly, robot navigation, and error correction [25]. However, biosensor-based methods remain a proof of concept. In practice, they suffer from performance degradation due to signal artifacts/noises from eye blinking and body motions, making it impossible to control robots with a reliable performance.

Due to the practical limitations of existing control methods, researchers have begun exploring LLM-based control for robotics. Compared to existing methods, the LLM-based methods (1) remove the need to use hand, body, and gaze to generate robot control information, only rely on natural language commands; (2) use the LLM's built-in domain knowledge to interpret workers' inputs into mechanically feasible robot actions, even if workers have no programming or robotics knowledge. To the best of my knowledge, Zhao et al. [14] were among the first to implement an LLM-based robot control method. Their method enabled the LLM to interpret both task scenarios and the communication between humans and robots, thereby generating collision-free waypoints for robots to perform a series of tasks. In this method, Zhao et al. leveraged a predefined prompt and CoT reasoning to guide LLM to generate step-by-step waypoints to guide robots in tasks such as cube sorting, grocery packing, sandwich making, and floor-sweeping.

Despite the effectiveness of CoT in solving reasoning questions, as mentioned in the Introduction and my midterm project, CoT still has limitations. The reasoning capability of CoT is a single "linear" sequence: the LLM may get locked into the wrong direction at the beginning and bring this wrong direction to the end. In other words, if the model generates a wrong waypoint in the first intermediate step, then all its following waypoints will include this mistake and finally generate a wrong robot control (collisions). There is no mechanism to re-check the earlier steps or explore alternatives within a single CoT reasoning process. This limitation opened the door to new reasoning structures [26–29]. One of the methods was called the Tree of Thought (ToT). ToT was first developed by Yao et al. (2023; [30]) and concurrently by Long (2023; [28]). ToT was developed based on the general idea of CoT but extending the CoT process to a search tree. To be more specific, instead of generating a single chain of thought, the ToT has a backtracking mechanism and can explore different directions/approaches, similar to the process of finding alternate solutions to a question. By following this structure, ToT expands the reasoning process from a linear setting like CoT to a tree-like branching structure. In other words, the capability of the LLM to solve questions with multiple solution paths or with a multi-step reasoning process can be greatly enhanced [28,30,31]. For an LLM-based robot control scenario, ToT has the potential to allow the system to evaluate the correctness of each generated waypoint, ensuring successful and collision-free robot control. In addition to CoT and ToT, researchers have proposed a new reasoning process as a connected graph. GoT was developed to mimic how humans solve the question, but in a different direction than ToT [29]. GoT considers that, in practice, human thinking is not strictly hierarchical like ToT – human ideas can diverge or converge, information from different resources can be combined, and there may depend on other thoughts, and all these thoughts may contribute to the final answer to the question. Compared to CoT and ToT, GoT is a natural improvement of the reasoning structure, and more flexible thought structures can lead to better problem-solving [29]. However, one challenge with GoT is managing the complexity: the search needs to keep track of many pieces of information (i.e., all the nodes and edges) and decide how to integrate them. While some studies indicate that GoT is promising to achieve a "relatively" computationally cheap process, but it adds another complex layer to prompt engineering. As such, current GoT methods are only suitable for specialized tasks to showcase transformation. General-purpose GoT that can be used for any arbitrary problem remains an area of active research.



```
PATH_PLAN_INSTRUCTION="""
[Path Planning Instructions]
Each <coord> represents a tuple (x, y, z) indicating the gripper's location.
Please use the following steps to create a path plan to control the robot to reach the target:
1) Identify the target location (e.g., the object to pick) and determine the current gripper position.
2) Plan a sequence of <coord> points that allow the gripper to move smoothly from its current position to the target.
3) Ensure the ‹coord> points are evenly spaced between the starting and target positions.
4) Avoid collisions with other robots, and ensure the gripper remains clear of the surrounding objects.
[Incorporating Environment Feedback to Improve the Plan]
- If inverse kinematics fails, adjust the steps to propose more feasible positions for the gripper.
- In case of detected collisions, modify the path to ensure the gripper and
  any in-hand objects maintain a safe distance from obstacles.
- To ensure even spacing, adjust the path so that the distances between consecutive ‹coord> points are consistent.
  Example: For the path [(0.2, 0.3, 0.4), (0.25, 0.3, 0.4), (0.4, 0.5, 0.8)],
  the distance between (0.2, 0.3, 0.4) and (0.25, 0.3, 0.4) is too small,
  while the distance between (0.25, 0.3, 0.4) and (0.4, 0.5, 0.8) is too large.
  Adjust the path to something like [(0.2, 0.3, 0.4), (0.3, 0.4, 0.6), (0.4, 0.5, 0.8)].
- If the plan fails during execution, re-plan with more feasible steps or consider new actions.
"""
```

Fig.1. (A) Overview of the proposed ToT-enhanced LLM System for Robot Control; (B) Prompt for Waypoint Generation Component in LLM.

Therefore, to enhance the existing LLM-based robot control method that relies on CoT, this final project will try to integrate the ToT into the existing robotic control system. This project will evaluate if the enhanced reasoning structure can improve the robot control efficiency and accuracy.

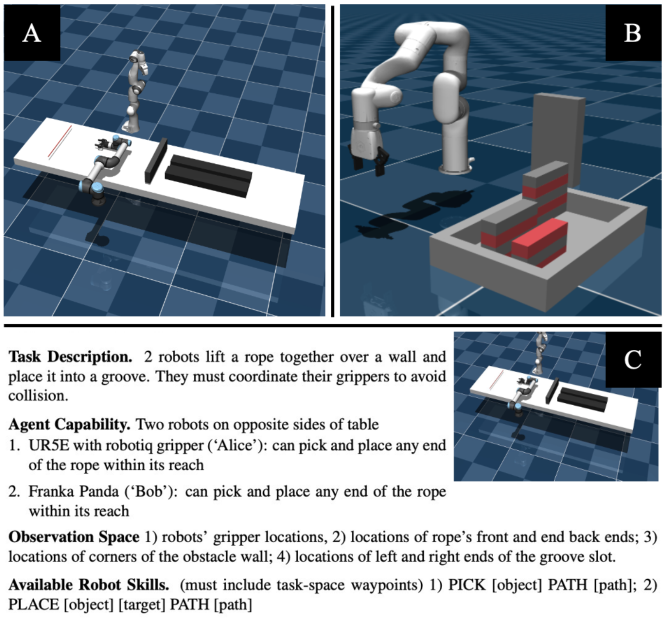
III. RESEARCH FOCUS AND QUESTION

Building upon the Literature Review and the progress made in the midterm project, the final project will address **the following research questions:** **(1)** How can we enhance the accessibility and "ease of use" of existing robot control methods by integrating the ToT framework? **(2)** How does the proposed ToT framework in comparison with CoT-based models (baseline)? **(3)** What will be the practical limitations of the LLM-based robot control method? To answer the question 1, I will modify the ToT framework developed during my midterm project and integrate it into the LLM-based robot control method. In other words, I will develop a new ToT-enhanced LLM system for robot control, allowing users to use their natural language input (spoken or typed) to achieve a more intuitive and user-friendly robot control. Then, this system will be further evaluated and compared with baseline methods to answer the questions 2 and 3. This project will try to enhance the reasoning capability of current LLM-based robot control methods.

IV. METHODOLOGY AND EXPERIMENT

Fig.1-a shows an overview of the methodology. As shown, for this ToT-enhanced LLM system for robot control, the input includes user-typed or spoken natural language. Then, the user input and the corresponding task (simulated) will be entered into the LLM engine. Here, the simulated environment is encoded in XML format, so that the LLM can read and understand the content (spatial and object-level) of the environment. Next, the ToT reasoning structure is activated to (1) interpret these inputs, (2) generate corresponding waypoints based on its tree-like structure, and (3) evaluate the feasibility of each waypoint. The waypoint generation will be driven by a predefined prompt shown in Fig.1-b. For each layer of the ToT structure, it generates multiple collision-free waypoints.

The ToT will select one feasible waypoint in each layer and move to the next layer using the DFS search until the robot reaches the target position. This system builds on the LLM-based robot control method proposed by Zhao et al. [14], with the main difference being updating the reasoning structure from CoT to ToT to better analyze and generate waypoints. In my midterm project, I explored the original Tree-of-Thought (ToT) framework developed in the study [30]. To apply the ToT reasoning structure to any task (robot control), five components are required [28,30] - adaptive component, decomposition component, generation component, evaluation component, and search component. Each of them plays an important role in the reasoning process. The adaptive component will define the task-specific parameters, such as the maximum tree depth and search strategy; the decomposition component will break the input problem into smaller and more manageable steps; the generation component develops the feasible/potential candidates or intermediate solutions that can contribute to the input question; the evaluation component then scores and ranks the generated candidates; and the search component describes the details of the search strategy and navigates the problem space. Together, these components enable the framework to reason through complex tasks (e.g., robot control) step by step. In this final project, the main structure of the ToT was kept the same, but the evaluation component was modified to check if the generated waypoint could lead to a collision and be mechanically feasible. If true, all waypoints will pass to an RRT-based motion planner to generate the final trajectories. To be more specific, each waypoint was assigned an evaluation score ($S = C_{col} \times C_{dist}$), where C_{col} measures the collision avoidance, and C_{dist} calculates the distance to the target. C_{col} was a binary value, with 0 indicating a collision and 1 indicating a collision-free point. C_{dist} indicates how close the end effector came to the goal. As such, this score assessed each waypoint's safety and accuracy. If the score is not 0, the waypoint is considered as viable. In addition to the evaluation component, the search mechanism of the ToT determines the exploration strategy of the problem-solving tree. While the Breadth-First Search (BFS) strategy explores all nodes at a given depth before moving to the next layer, it is computationally expensive for complex tasks like robot control. DFS can drive the ToT process to explore as far along each branch as possible before backtracking to explore other branches, allowing the tree to quickly reach potential waypoints to drive the robotic arm to the target goal. Therefore, in my final project, the search component was determined as DFS.



Task Description. 2 robots lift a rope together over a wall and place it into a groove. They must coordinate their grippers to avoid collision.

Agent Capability. Two robots on opposite sides of table

1. UR5E with robotiq gripper ('Alice'): can pick and place any end of the rope within its reach
2. Franka Panda ('Bob'): can pick and place any end of the rope within its reach

Observation Space 1) robots' gripper locations, 2) locations of rope's front and end back ends; 3) locations of corners of the obstacle wall; 4) locations of left and right ends of the groove slot.

Available Robot Skills. (must include task-space waypoints) 1) PICK [object] PATH [path]; 2) PLACE [object] [target] PATH [path]

Fig.2. (A) Rope-moving Environment Simulated in MuJoCo; (B) Bricklaying Environment Simulated in MoJoCo; (C) Task Annotations for Rope-moving Environments.

To evaluate the performance of the ToT-enhanced LLM system for robot control, I used the RoCoBench benchmark dataset. RoCoBench consists of 6 robot collaboration scenarios. Each scenario has three important properties ("annotations") to help evaluate the proposed system. Specifically, Fig.2-a shows a rope-picking scenario in RoCoBench, which further includes "Task Description"; "Agent Capability"; "Observation Space"; and "Available Robot Skills" information. Task description describes the overview of this task in the simulated environments, including (1) the number of robots; (2) the target object, and (3) the mission of this task. Agent capability and Available Robot Skills define the task and capability of each robot. The Observation Space further defines the scope of the task, so the LLM will not focus on other scenarios in simulation unrelated to this task. Fig.2-c shows the above information about the rope moving task. In addition to RoCoBench, I developed a custom simulation on robot-assisted bricklaying to test the performance of the system. Fig.2-b shows this new scenario.

The proposed LLM control method was integrated into these simulated scenarios. For each task, users input a high-level command to the system, like "Please work together to pick up the rope and move it to the groove." Then, the system analyzed the input commands and generated a series of waypoints using the ToT structure and motion planner, allowing the robot to perform assigned tasks (e.g., rope moving and bricklaying). The next section will show the results of this project.

Table 1. Performance of the proposed ToT-enhanced interface in robot control task (10 trails in total)

	Performance	Time			Number of calls		LLM Usage	
[a] 1	2	3	4	5	6	7	8	9
Task 1	90.0%	882	186	696	166	260	202,441	335,702
Task 2	80.0%	943	222	721	306	402	264,727	488,108

Note: [a] *1 = Task; 2 = Success Rate; 3 = Total Elapsed Time (Sec); 4 = Total Generation Time (Sec); 5 = Total Evaluation Time (Sec); 6 = Total Number of Generations; 7 = Total Number of Evaluations; 8 = Completion Tokens; 9 = Prompt Tokens.*

V. RESULTS

This section reports the evaluation results from the experiment. Table 1 shows eight metrics to measure the performance of robot control experiments. Like my midterm project, among these eight metrics, three different run times were measured – total elapsed time, total generation time, and total evaluation time. The total elapsed time is the cumulative time required to complete

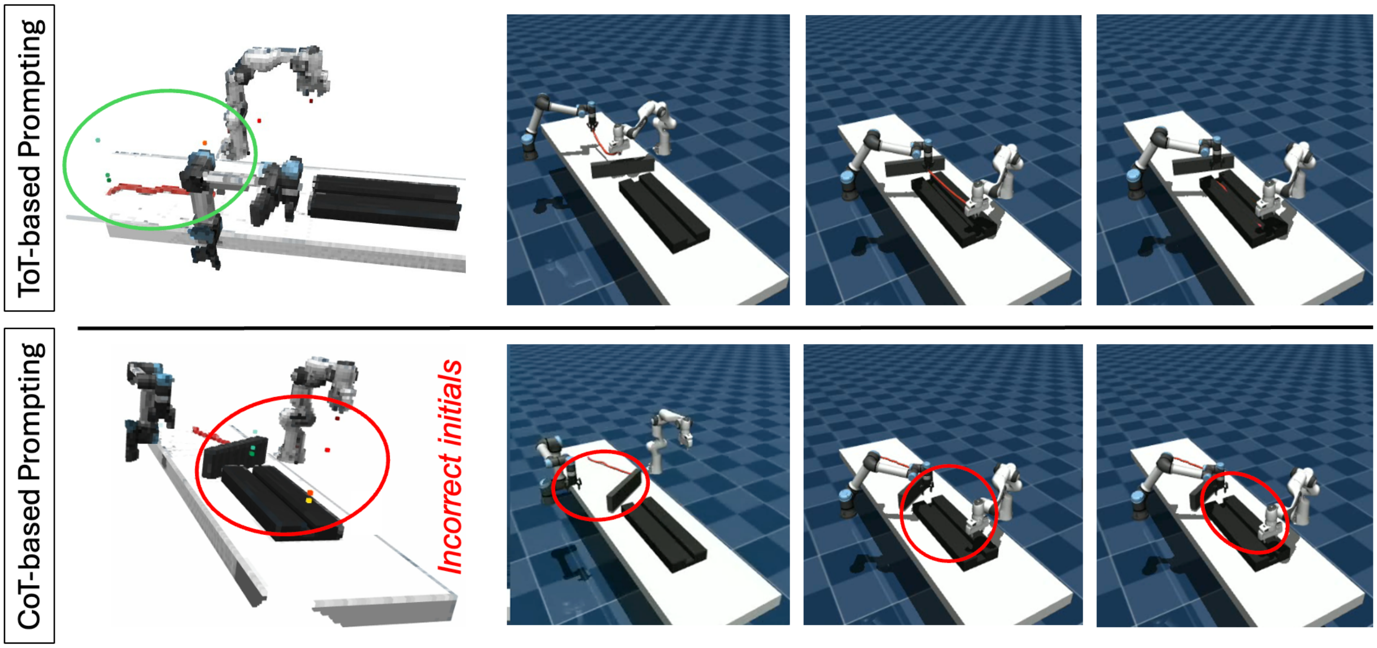


Fig.3. Comparison between the CoT-based and ToT-based Robot Control Mechanism (CoT will lead to mistakes in the initial waypoints generation).

each task. The total generation time indicates how long the interface spent producing potential solutions (waypoints), and the total evaluation time measures how long the interface spent evaluating those waypoints using the evaluation component. All these times were aggregated across the users' inputs. Additionally, Table 1 included the total number of generations and evaluations, indicating how often the LLM-based interface was called for generating and evaluating candidates, respectively. Completion tokens represented the interface's output tokens, and prompt tokens referred to the input tokens. The success rate metric was used to assess the effectiveness of the interface in controlling the robotic arm. Notably, due to the API token limits (TPM limits) and computational constraints, Table 1 only reports the performance of the ToT-enhanced system in rope moving (Task 2) and bricklaying tasks (Task 1) mentioned in the previous section. For other RoCoBench scenarios, like cabinet arranging and grocery packing, the ToT-enhanced system exceeded ChatGPT's TPM (30,000 tokens per minute) limitation and failed to generate feasible waypoints.

As shown in Table 1, for both scenarios, the LLM-based control method enabled the robotic arm to perform the task with a success rate of over 75% (10 trials per task). In the more complex rope-moving task, the token usage was around 30% higher than in the bricklaying task. However, in terms of time cost (total elapsed time), these two scenarios were comparable, with only a 6.9% difference observed. One plausible reason is that the evaluation component of ToT applied a similar process to assess the feasibility of each waypoint.

ABLATION STUDY: In addition to the results in Table 1, Fig.3 (ToT-based Prompting) shows the intermediate steps taken by the ToT-based robot control system during the rope-moving task. During the control process, the ToT-based LLM successfully interpreted the user's input and produced a sequence of collision-free waypoints, enabling the robotic arm to complete the task smoothly. As a comparison, Fig.3 (CoT-based Prompting) presents the results generated by the CoT-based LLM, with the same input. While CoT also generated the collision-free waypoints, the initial waypoint was incorrect, preventing the arm from picking up the rope and moving the rope to the groove. This comparison highlighted the limitations of the CoT: the CoT-driven LLM may get locked into the wrong direction at the beginning and bring this wrong direction to the end. In other words, if the model generates a wrong waypoint in the first intermediate step, then all its following waypoints will include this mistake and finally generate a wrong robot control (collisions). In contrast, the ToT-based method could better solve this issue by evaluating the correctness of each waypoint at each step. However, one important limitation of ToT can be observed from Table 1: high token consumption. As Table 1 indicates, for each task, the ToT method consumed around 30,000 tokens, which was computationally expensive and hard to scale for more complex tasks in practice.

VI. CONCLUSION AND FUTURE WORKS

The final project first provides a literature review covering existing robot control interfaces as well as current reasoning structures (CoT, ToT, and GoT) in

LLMs. Then, upon insights from this review, the author proposed a ToT-enhanced LLM-based robot control system designed to allow users to directly and more intuitively control robots without programming and mechanical engineering expertise. Next, the author introduced an experiment conducted to test the performance of the proposed system. The experimental results reveal the potential and feasibility of the proposed system for intuitive robot control: Users could directly input their high-level natural language instructions, which the system could interpret and translate into the control commands (waypoints) for robots. In this way, the system has the potential to reduce the barrier to robot control/operation for users (construction workers) without domain knowledge in mechanical engineering and programming. However, the experiment further revealed one interesting thing (a limitation). While the interface enabled users to directly control robots, it led to high token consumption and time cost, which will raise a concern about control delay and computational efficiency in practice. Based on the above findings, the next steps will explore a more efficient reasoning structure that integrates both the advantages of CoT (cost efficiency) and ToT (backtracking). For simpler cases that can be solved by CoT, the interface can activate CoT for faster and cost-efficient reasoning. For a more complex case, ToT will be activated to ensure the accuracy of the robot control. In other words, a dynamic search mechanism that can optimize the balance between accuracy and efficiency is needed in the future for a more efficient LLM-based robot control system.

VII. DELIVERABLES

The materials related to this final project can be found in https://github.com/liuyiz1994/COMS-E6156-TREEBOT. Please note: to run the code, you need to use your OpenAI API. Please refer to ReadMe document for more details.

VIII. SELF-EVALUATION

In this final project, I expanded my midterm project (ToT) to a more realistic application in robotic control. I developed a ToT-enhanced LLM system for robotic arm control. In addition, I created a new scenario and added it to the RoCoBench dataset. I applied two robot control scenarios to test my proposed system. I further compared my proposed method with the baseline (CoT-based method) and analyze the limitations of each method. The challenges I met were how to select the suitable search engines for the ToT reasoning structure and how to train the proposed system. As I mentioned in Result section, the LLM-based robot control methods require a great amount of token, therefore it was difficult for me to evaluate my system across all scenarios in the benchmark.

In this project, I learnt a lot related to LLM, Robot Simulation, and Robot Control. One important thing is that I learnt this step-by-step. During my midterm project, I mainly focused on ToT, which allowed me to explore the ToT-based robot control in the final project. During the process, I trained my research skills and "relatively" understand how to do research gradually.

REFERENCES

[1] W. Fang, P.E.D. Love, H. Luo, L. Ding, Computer vision for behaviour-based safety in construction: A review and future directions, Advanced Engineering Informatics. 43 (2020) pp. 100980. https://doi.org/10.1016/j.aei.2019.100980.

[2] S. Shayesteh, A. Ojha, Y. Liu, H. Jebelli, Human-robot teaming in construction: Evaluative safety training through the integration of immersive technologies and wearable physiological sensing, Safety Science. 159 (2023) pp. 106019. https://doi.org/10.1016/j.ssci.2022.106019.

[3] B. Chu, K. Jung, M.-T. Lim, D. Hong, Robot-based construction automation: An application to steel beam assembly (Part I), Automation in Construction. 32 (2013) pp. 46–61. https://doi.org/10.1016/j.autcon.2012.12.016.

[4] D. Floreano, R.J. Wood, Science, technology and the future of small autonomous drones, Nature. 521 (2015) pp. 460–466. https://doi.org/10.1038/nature14542.

[5] J. Irizarry, M. Gheisari, B.N. Walker, Usability assessment of drone technology as safety inspection tools, Journal of Information Technology in Construction (ITcon). 17 (2012) pp. 194–212.

[6] T. Linner, W. Pan, R. Hu, C. Zhao, K. Iturralde, M. Taghavi, J. Trummer, M. Schlandt, T. Bock, A technology management system for the development of single-task construction robots, Construction Innovation. 20 (2020) pp. 96–111. https://doi.org/10.1108/CI-06-2019-0053.

[7] T. Bock, T. Linner, Single-Task Construction Robots by Category, in: Construction Robots, Cambridge University Press, Cambridge, 2016: pp. 14–290. https://doi.org/10.1017/CBO9781139872041.002.

[8] Y. Liu, M. Habibnezhad, H. Jebelli, Brainwave-driven human-robot collaboration in construction, Automation in Construction. 124 (2021) pp. 103556. https://doi.org/10.1016/j.autcon.2021.103556.

[9] Ruchik Kashyapkumar Thaker, Robotics in Construction: A Critical Review of Reinforcement Learning, Imitation Learning, and Industry-specific Challenges for Adoption, International Journal For Multidisciplinary Research. 6 (2024). https://doi.org/10.36948/ijfmr.2024.v06i02.29707.

[10] E. Seemann, K. Nickel, R. Stiefelhagen, Head pose estimation using stereo vision for human-robot

interaction, Proceedings - Sixth IEEE International Conference on Automatic Face and Gesture Recognition. (2004) pp. 626–631. https://doi.org/10.1109/AFGR.2004.1301603.

[11] J.Y. Lee, G.W. Rhee, D.W. Seo, Hand gesture-based tangible interactions for manipulating virtual objects in a mixed reality environment, The International Journal of Advanced Manufacturing Technology. 51 (2010) pp. 1069–1082. https://doi.org/10.1007/s00170-010-2671-x.

[12] M. Svenstrup, S. Tranberg, H.J. Andersen, T. Bak, Pose estimation and adaptive robot behaviour for human-robot interaction, in: 2009 IEEE International Conference on Robotics and Automation, IEEE, 2009: pp. 3571–3576. https://doi.org/10.1109/ROBOT.2009.5152690.

[13] Y. Liu, M. Habibnezhad, H. Jebelli, Brain-computer interface for hands-free teleoperation of construction robots, Automation in Construction. 123 (2021) pp. 103523. https://doi.org/10.1016/j.autcon.2020.103523.

[14] Z. Mandi, S. Jain, S. Song, RoCo: Dialectic Multi-Robot Collaboration with Large Language Models, in: 2024 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2024: pp. 286–299. https://doi.org/10.1109/ICRA57147.2024.10610855.

[15] A. Aryal, A. Ghahramani, B. Becerik-Gerber, Monitoring fatigue in construction workers using physiological measurements, Automation in Construction. 82 (2017) pp. 154–165. https://doi.org/10.1016/j.autcon.2017.03.003.

[16] M.M. Moniri, F.A.E. Valcarcel, D. Merkel, D. Sonntag, Human Gaze and Focus-of-Attention in Dual Reality Human-Robot Collaboration, in: 2016 12th International Conference on Intelligent Environments (IE), IEEE, 2016: pp. 238–241. https://doi.org/10.1109/IE.2016.54.

[17] G. Mehlmann, M. Häring, K. Janowski, T. Baur, P. Gebhard, E. André, Exploring a Model of Gaze for Grounding in Multimodal HRI, in: Proceedings of the 16th International Conference on Multimodal Interaction, ACM, New York, NY, USA, 2014: pp. 247–254. https://doi.org/10.1145/2663204.2663275.

[18] C.-M. Huang, B. Mutlu, Anticipatory robot control for efficient human-robot collaboration, in: 2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI), IEEE, 2016: pp. 83–90. https://doi.org/10.1109/HRI.2016.7451737.

[19] J. Berg, A. Lottermoser, C. Richter, G. Reinhart, Human-Robot-Interaction for mobile industrial robot teams, Procedia CIRP. 79 (2019) pp. 614–619. https://doi.org/10.1016/j.procir.2019.02.080.

[20] M. Terreran, L. Barcellona, S. Ghidoni, A general skeleton-based action and gesture recognition framework for human–robot collaboration, Robotics and Autonomous Systems. 170 (2023) pp. 104523. https://doi.org/10.1016/j.robot.2023.104523.

[21] X. Wang, D. Veeramani, Z. Zhu, Wearable Sensors-Based Hand Gesture Recognition for Human–Robot Collaboration in Construction, IEEE Sensors Journal. 23 (2023) pp. 495–505. https://doi.org/10.1109/JSEN.2022.3222801.

[22] D. Jirak, S. Tietz, H. Ali, S. Wermter, Echo State Networks and Long Short-Term Memory for Continuous Gesture Recognition: a Comparative Study, Cognitive Computation. 15 (2023) pp. 1427–1439. https://doi.org/10.1007/s12559-020-09754-0.

[23] M. Oudah, A. Al-Naji, J. Chahl, Hand Gesture Recognition Based on Computer Vision: A Review of Techniques, Journal of Imaging. 6 (2020) pp. 73. https://doi.org/10.3390/jimaging6080073.

[24] K. Schaaff, T. Schultz, Towards an EEG-based emotion recognizer for humanoid robots, in: RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interactive Communication, IEEE, 2009: pp. 792–796. https://doi.org/10.1109/ROMAN.2009.5326306.

[25] A.F. Salazar-Gomez, J. Delpreto, S. Gil, F.H. Guenther, D. Rus, Correcting robot mistakes in real time using EEG signals, Proceedings - IEEE International Conference on Robotics and Automation. (2017) pp. 6570–6577. https://doi.org/10.1109/ICRA.2017.7989777.

[26] Q. Lyu, S. Havaldar, A. Stein, L. Zhang, D. Rao, E. Wong, M. Apidianaki, C. Callison-Burch, Faithful Chain-of-Thought Reasoning, ArXiv Preprint. (2023). https://doi.org/https://doi.org/10.48550/arXiv.2301.13379.

[27] C. Li, G. Dong, M. Xue, R. Peng, X. Wang, D. Liu, DotaMath: Decomposition of Thought with Code Assistance and Self-correction for Mathematical Reasoning, ArXiv Preprint. (2024). https://doi.org/https://doi.org/10.48550/arXiv.2407.04078.

[28] J. Long, Large Language Model Guided Tree-of-Thought, ArXiv Preprint. (2023). https://doi.org/https://doi.org/10.48550/arXiv.2305.08291.

[29] M. Besta, N. Blach, A. Kubicek, R. Gerstenberger, M. Podstawski, L. Gianinazzi, J. Gajda, T. Lehmann, H. Niewiadomski, P. Nyczyk, T. Hoefler, Graph of Thoughts: Solving Elaborate Problems with Large Language Models, Proceedings of the AAAI Conference on Artificial Intelligence. 38 (2024) pp. 17682–17690. https://doi.org/10.1609/aaai.v38i16.29720.

[30] S. Yao, D. Yu, J. Zhao, I. Shafran, T.L. Griffiths, Y. Cao, K. Narasimhan, Tree of Thoughts: Deliberate Problem Solving with Large Language Models, NIPS '23: Proceedings of the 37th International Conference on Neural Information Processing Systems. (2023). https://doi.org/https://doi.org/10.48550/arXiv.2305.10601.

[31] E. Sgouritsa, V. Aglietti, Y.W. Teh, A. Doucet, A. Gretton, S. Chiappa, Prompting Strategies for Enabling Large Language Models to Infer Causation from Correlation, ArXiv Preprint. (2024). https://doi.org/https://doi.org/10.48550/arXiv.2412.13952.