Proceedings of the 2024 IEEE
International Conference on Robotics and Biomimetics
December 10-14, 2024, Bangkok, Thailand

LAC: Using LLM-based Agents as the Controller to Realize Embodied Robot

Jiahong Xu, *Member, IEEE,* Zhiwei Zheng and Zaijun Wang

***Abstract*— The rapid development of Large Language Models (LLMs) facilitates the application of robotics, especially for robotic control. LLM-based robots are a way to realize embodied intelligence, but they still lack a general control paradigm to achieve embodied robot. We propose LAC (LLM-based Agents as the Controller), a new formulation to enable embodied robots to autonomously react to high-level tasks like humans. Specifically, the controller in LAC is LLM-based agents that can plan, decompose tasks and make decisions based on linguistic input. Two agents operate in parallel within the controller to replicate both instinctive reaction and deep consideration. The output of the controller consists of linguistic parameters that activate task-specific tools, enabling autonomous execution of low-level actions. LAC translates multi-modal feedback information into linguistic messages, which are then transmitted back to the controller to establish a closed-loop control flow. Once equipped with appropriate tools, the proposed LAC can be applied across various applications. A straightforward simulated office task, TASK0, illustrates the successful completion of the embodied task by LAC. Furthermore, testing five comparative LLMs on TASK0 also demonstrates that LAC serves as a potential test platform for validating the capability of LLMs to function as the cognitive center of embodied robots.**

I. Introduction

The rapid advancement of Large Language Models (LLMs) [1] is enabling the integration of robotics, and the fusion of LLM and robots holds great promise for achieving embodied intelligence [2]. Specifically, agents capable of translating visual inputs into actions can be regarded as embodied systems [3], prompting NLP researchers to focus on constructing multi-modal LLM. However, a crucial aspect for an embodied robot lies in its capacity to operate within the physical world amidst high degrees of uncertainty and complexity [4], an area where expertise in control theory plays a pivotal role.

Hence, it is imperative to integrate the advantages of both LLM and control theory in order to develop a novel control paradigm that enables embodied robots to autonomously respond to high-level tasks akin to human behavior. Traditionally, controllers can be categorized into two main types: offline control laws such as PID control [5], and online control laws such as MPC [6]. LLM inherently functions as a statistical model, positioning it as a potential candidate for the latter type of controller. The proliferation of LLM technologies, coupled with their ability to comprehend user inputs and engage in planning and reasoning, has sparked increasing interest in leveraging LLMs for robot applications [7]. However, the current utilization of LLM in embodied robots is proficient at handling short-term tasks, yet LLM alone cannot guarantee the success of long-term tasks necessary for fully autonomous embodied robots [8].

In order to achieve autonomous embodied robot, a comprehensive control paradigm is essential for several reasons. Firstly, the embodied robot is designed to carry out tasks through interaction with its environment and humans, necessitating a closed-loop framework to facilitate intelligent operation. Secondly, the robot must be capable of responding to diverse user inputs, requiring it to treat these inputs as setpoints in accordance with control theory principles. Thirdly, the ability to generalize across different environments is crucial for an embodied robot, demanding a flexible extension of the control paradigm. Lastly, rapid response capability is also necessary; for simpler tasks, extensive deliberation is unnecessary and the embodied robot should be able to adapt its response logic in real time.

This work introduces LAC, a novel control paradigm designed to achieve autonomous embodied robots as depicted in Fig. 1, and the proposed LAC has the potential to achieve embodied intelligence. The controller of LAC is the brain, which comprises multiple LLM-based agents. The multi-agent paradigm empowers LAC with the capability to react swiftly to simple tasks and thoroughly to complex multi-step tasks. The tools module within LAC is adaptable and can be expanded or replaced based on specific user requirements and interacting environments. These flexible tools define the capacity boundary of the embodied robot, enabling it to be easily adapted for different environments by substituting necessary tools. The feedback mechanism within LAC ensures that the plans optimized by the LLM are feasible, and that actions taken by the tools contribute to task completion. Furthermore, the information flow in LAC is conveyed through linguistic language, with feedback messages and initial user inputs being converted into a deviation signal for decision-making by the agents-based controller. The innovative techniques of LLM such as RAG [9] and advanced control theory methods like MPC can be seamlessly integrated into the LAC paradigm to enhance robotics embodiment.

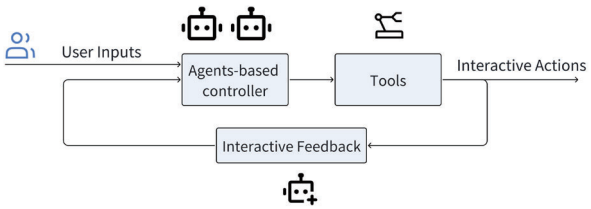

Fig. 1. The simplified control framework of LAC

Jiahong Xu is with the Robotics Institute, Ningbo University of Technology, Ningbo, Zhejiang, 315211, China; also with Joyson Robot, Ningbo, China. (corresponding author, e-mail: jiahongxu@nbut.edu.cn).

Zhiwei Zheng is with the Robotics Institute, Ningbo University of Technology, Ningbo, Zhejiang, China (e-mail: 13790391822@163.com)

Zaijun Wang is with Key Laboratory of Flight Techniques and Flight Safety, CAAC (e-mail: zaijunwang@cafuc.edu.cn).

2024 IEEE International Conference on Robotics and Biomimetics (ROBIO) | 979-8-3315-0964-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ROBIO64047.2024.10907494

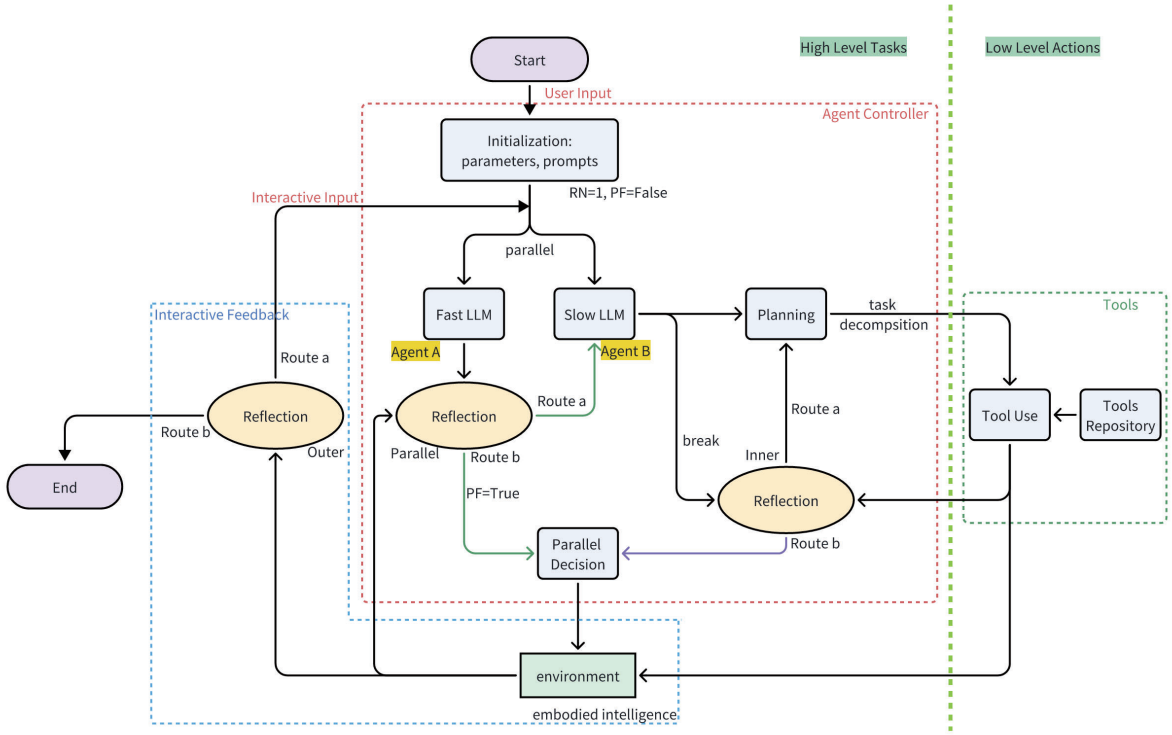

Fig. 2. A detailed control flow example of LAC.

Our main contributions are as follows:

1) We have developed a unified control framework, LAC, which integrates the reasoning capabilities of LLM with the uncertain handling abilities of feedback control. This advancement in LAC significantly contributes to the realization of embodied robots.

2) We propose a multi-agent based controller to enable rapid responses by the embodied robot in unknown situations.

3) The utilization of a flexible tool module facilitates the connection between high-level subtasks provided by LLM and low-level actions executed in the physical world.

4) An interactive feedback module is employed to monitor task progress, serving as a bridge between local and global optimality.

5) To facilitate adoption by researchers, we provide a detailed example illustrating the control flow of LAC.

II. Related Works

The widespread adoption of LLM technology renders it a suitable brain for robots, and the integration of LLM and robotics results in embodied robots capable of collaborating with humans, engaging with them in an autonomous, secure, and purposeful manner [10]. The primary concern lies in the grounding of robotics issues [11], such as visual question answering [12], image captioning [13], visual language foundation model [14]. In addition to the visual language model (VLM) [15], function calling plays a crucial role in enabling LLM-based robots to autonomously interact with the physical world while mitigating potential hallucinations within pre-trained models [16]. This function calling, also referred to as tools or functional APIs, includes notable examples like SayCan [17], Code as Policies [18], ProgPrompt [19], RT-2 [20]. Unlike previous studies, the grounding of LAC is facilitated by the flexible tools module within LAC, enhancing its generalization capabilities.

A framework has been proposed to autonomously determine when and how to request assistance from APIs in order to enhance the effectiveness of function calling [21]. In real-world scenarios, there may be reactive objects such as AGV (Automatic Guided Vehicle) coexisting with the embodied robot. To explicitly consider the reactive behaviors of interacting targets like AGV, a predictor module can be introduced [22]. High-performance tools are crucial for successful interactions, and these tools can take the form of robot-specific primitives [23] or online open sources available through platforms like hugging face [24]. Researchers have developed exceptional tools for embodied robots to utilize, such as Segment Anything [25], GOAT [26], AnyGrasp [27]. The agent-based controller in LAC intelligently determines when to invoke specific tools, while the parallel mechanism in LAC accelerates function callings.

The open-loop response to a given input is not desirable for an embodied robot; instead, feedback from interactions should be considered directly. Verbal reflection serves as one method to achieve this feedback [28], and experiences of failure can enhance future performance [29]. The chain of thoughts acts as pseudo-feedback during planning, enhancing the outcomes of LLM [30]. Conversely, React interacts with the environment in real time, using immediate feedback to improve performance [31]. Other frameworks that play a similar role to feedback include Self-refine [32], Critic [33], LEMA [34], and InteRecAgent [35]. The purpose of feedback within the LAC framework is to ensure that lower-level interactive actions contribute effectively to the given input and allow the embodied robot to adapt its higher-level tasks online promptly.

III. METHOD

A unified control framework, LAC, is designed in this section, and a detailed control flow example, as shown in Fig. 2, is explained as follows.

A. Agents-based controller

The controller is the brain of the LAC, which determines the higher-level subtasks in the embodied robot. There are two external inputs: one is the user input, and the other is the interactive input. During the control flow, LAC can respond to updated user input, making it a reactive control paradigm. There is also one internal input sent by tools, which provides feedback on the tool invocations. There are two outputs: one is an internal output to the Tools, and the other is an external output to the environment. The inputs and outputs of the controller are all linguistic messages.

A key feature of the controller is that LAC uses two parallel agents—Agent A and Agent B. Typically, Agent A is a fast agent that can respond to user input quickly, with little or no reliance on tools. In most cases, Agent A depends solely on LLM to react, and this LLM should be small. While the fast Agent A responds to the query, the slower Agent B is also active, reacting to the same query with the use of multiple tools. Fast Agent A always obtains the result first and reflects on it to decide whether this result is sufficient for the given request. If the answer is positive, the result from Agent A becomes the output of the controller, and Agent B terminates; otherwise, the controller waits for the answer from the slower Agent B. Agent B usually handles tasks that require multi-step reasoning and planning, and the initial task is often decomposed into several subtasks. Once the function calling is completed successfully, Agent B reflects on the interactive results and decides whether to conduct an inner loop or an outer loop.

B. Tools

Embodied robots should execute various tasks in different environments, and this generalization ability is guaranteed by the Tools module in LAC.

The input to the Tools module is the controller's inner output, and the outputs of the Tools module are inner feedback to the controller and outer feedback to the environment.

Given subtasks designed by the controller, tools useful for realizing the specific subtask are invoked. This is the embodiment of grounding in LAC: every subtask has corresponding tools to achieve it, and once the tools are unsatisfactory, they provide negative feedback to prompt the controller to replan the subtasks.

When deployed in different environments, LAC reconstructs its tools to adapt to various situations. For example, the Tools module can retrieve necessary tools from the Tools Repository, which can include local programs or online APIs (e.g., Hugging Face). Moreover, these tools can be other embodied robots or agents (e.g., AGVs).

C. Interactive feedback

The automation of embodied robots is guaranteed by the feedback mechanism, and LAC introduces an interactive feedback module to ensure quick reactive responses.

There are two inputs to the interactive feedback: one is the output of the controller, and the other is the output of the Tools. The first input can be either the result of fast Agent A or the result of slow Agent B. Once the controller decides to react quickly, Agent A sends messages; otherwise, the controller waits for Agent B to send messages. The second input is the intermediate results produced by the tools during the task, and this feedback monitors the task's progress. In this way, Agent B makes optimal decisions independently, and the tools strive to take actions, leading to a local optimum. The intermediate feedback reflects on this local optimum and enforces the controller to find a better solution if the local optimum is indeed not the global optimum. Thus, LAC avoids falling into the local optimum trap.

There are two outputs of interactive feedback, the first one is the inner feedback to fast Agent A. Although the controller may consider the result of Agent A to be the optimal choice, it could be a poor answer for the physical world due to LLM's hallucination. Therefore, if the result of Agent A is good enough as expected, this feedback (e.g., vision information in practice) terminates the procedure of Agent B; otherwise, the controller discards the result of Agent A and waits for the answer from Agent B. In this way, the local optimum found by Agent A is aligned with the global optimum. The other output is the outer feedback to the controller, reflecting whether the task is completed or if there is room for improvement. If the task is considered to be finished successfully and there is no further user input update, LAC stops and waits for new queries to come in.

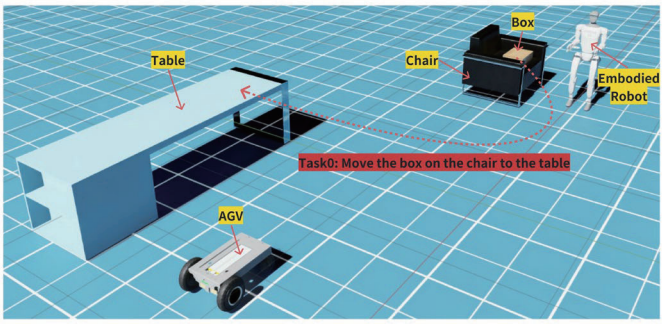

Fig. 3. The simple simulation sandbox of Task0

IV. RESULTS

A. Task Description

We define a common task in an office to test the effectiveness of the LAC as follows:

TASK0: *Move the box from the chair to the table.*

The embodied robot receives Task0 as user input and first picks up the box from the chair, then moves to the table to put down the box. Here, 'pick up' and 'put down' are tools used to interact with the physical world to complete the task. We assume all the tools are ideal and can be invoked successfully as expected.

The difficulty of the given task is that the robot does not know the weight of the box, and this information needs to be obtained through interactions. Luckily, this is what the embodied robot and the proposed control paradigm LAC are good at.

The logic of the embodied robot is as simple as that of a human: try to pick up the box first, and if it is within the robot's capability, move the box to the table; otherwise, ask for help.

In the current work, we only use a simple simulation, which is adequate to express the ideas of LAC. More complex simulations in Isaac Sim and deployment on a real robot will be conducted in our future research. The simulation sandbox is illustrated in Fig. 3, where there are: 1 embodied robot; 1 chair near the robot; 1 box weighing 5 kg placed on the chair; 1 table 5 meters away from the chair; and 1 AGV near the table. The constraints are as follows: the robot can carry objects weighing less than 4 kg over long distances, but it can only carry objects weighing between 4 kg and 6 kg over short distances (less than 1 meter); the AGV can carry objects weighing less than 10 kg and can be seen as a tool of the robot (meaning the robot can ask the AGV for help).

Now let's test this common office task to see whether the embodied robot with LAC can place the box on the table.

B. Results

Given Task0, the embodied robot uses LAC to react to this user input and complete the task autonomously. Some parts of the control flow in LAC are illustrated in Fig. 4, where the information flow is expressed in linguistic language. Brief snapshots of the Task0 progress are illustrated in Fig. 5.

1) Control flow in LAC

As shown in Fig. 4, once the LAC receives the user input "Move the box from the chair to the table", the controller activates two agents in parallel. The fast Agent A, equipped with no tools, cannot interact with the box in the physical world. Thus, Agent A replies with something like: "Sorry, I cannot interact with the real world." On the other hand, the slow Agent B is equipped with adequate tools that can complete TASK0 if invoked correctly. Thus, Agent B starts its reasoning procedure as follows: "1. Pick up the box on the chair; 2. Carry the box towards the table; 3. Put down the box on the table."

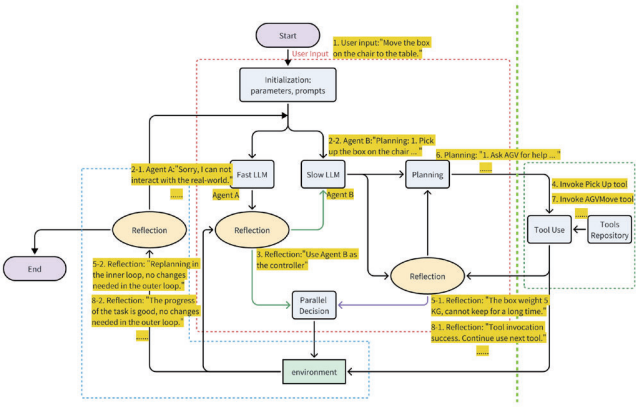

Fig. 4. Some parts of the control flow in LAC

Once we obtain the result from Agent A, the controller reflects on this result together with the given TASK0 and finds that Agent A failed to complete the task. In this way, the controller waits for the reply from Agent B, and Agent B takes over the controller.

Then the inner loop of Agent B is activated. It first invokes the tool 'Pick-Up' to complete the subtask "Pick up the box on the chair." In fact, this tool is itself a complex task, as the robot must use its visual capabilities to locate the box and determine how to use its arms to successfully grasp the box, along with other related missions. Here, we assume the 'Pick-Up' tool encapsulates these missions to simplify the illustration.

Like humans, the embodied robot should react to unexpected feedback and retry completing the initial task. In this case, we assume the Pick-Up tool feedback indicates the weight of the box, which is 5 kg. The robot finds out that while it can carry this box since it is less than 6 kg, doing so would harm its legs. The inner reflection forces Agent B to replan the subtasks, and the robot asks the AGV for help. During the inner reflection, the interactive feedback also triggers the outer

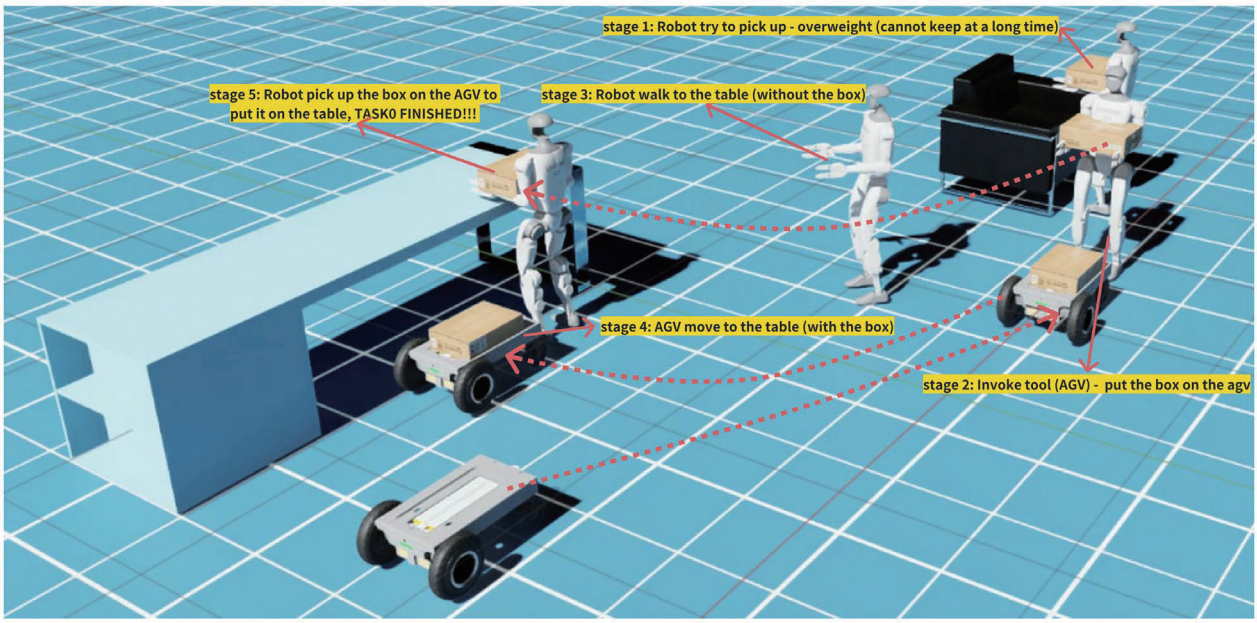

Fig. 5. Brief snapshots of the progress of TASK0 controlled by LAC.

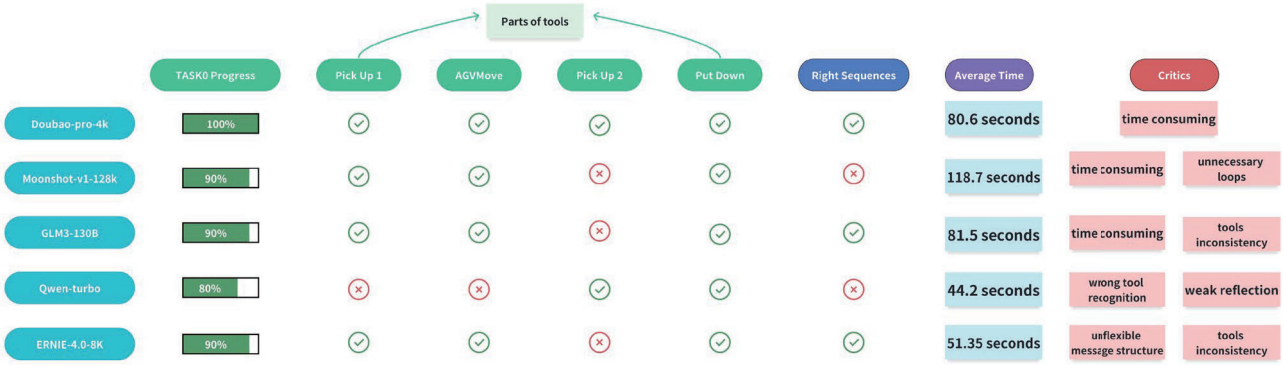

Fig. 6. The simple comparison results of different LLMs as the brain of LAC

reflection to check whether the controller should change the higher-level task. Since the inner loop works well, the outer loop will not interrupt the current control flow.

After replanning, the inner loop restarts with the AGVMove tool, and the outer loop monitors the progress of the task until TASK0 is completed successfully. The rest of the control flow is omitted for brevity, and the overall control progress is illustrated in Fig. 5 in the following subsection.

2) Snapshots of LAC control

The key feature of embodied intelligence is its ability to learn from interactions, and the LAC-based embodied robot is expected to possess this characteristic.

As shown in Fig. 5, since the robot does not know the weight of the box beforehand, it first tries to pick up the box in stage 1. After interactions, the robot obtains the knowledge that the box weighs 5 kg.

Then, in stage 2, the robot asks the AGV for help: since the robot can carry the box for a short time, it can place the box on the AGV and make the AGV carry the box toward the table.

In stage 3, the robot walks toward the table empty-handed, and in stage 4, the AGV carries the box toward the table. When they both arrive at the destination, the robot picks up the box from the AGV and puts it down on the table in stage 5.

In this way, TASK0 has been completed successfully under the control of LAC!

TABLE I. COMPARATIVE LLMS

No.	*Name*	*Reference*
1	Doubao-pro-4k	https://www.volcengine.com/
2	Moonshot-v1-128k	https://www.moonshot.cn/
3	GLM3-130B	https://www.zhipuai.cn/
4	Qwen-turbo	https://dashscope.console.aliyun.com/
5	ERNIE-4.0-8K	https://console.bce.baidu.com/qianfan/

3) LAC as a test platform for different LLMs

The reported results use Volcengine with Doubao-pro-4k as the LLM in LAC. Since LAC does not depend on specific LLMs, LAC can also serve as a candidate test platform to compare different LLMs for their capabilities to serve as the brain of an embodied robot. The comparative LLMs are reported in Table I.

The comparison results of these LLMs as the brain of LAC are illustrated in Fig. 6. We focus on the successful invocations of four important tools here: (1) Pick Up 1: Try to pick up the box according to the user input TASK0, and find out that the weight of the box is beyond the robot's capability to carry it for a long distance; (2) AGVMove: Ask the AGV for help to carry the box toward the table; (3) Pick Up 2: After the AGV has carried the box near the table, the robot also walks toward the table and picks up the box for the second time (the robot can carry a 6 kg box for a short time); (4) Put Down: The robot puts the box on the table to complete the initial TASK0.

These four tools indicate the progress of TASK0, and we assume the LLM that succeeds in invoking and finishing all four tool calls has completed TASK0 100%. Since Doubao-pro-4k succeeds in all the tools, Moonshot-v1-128k, GLM3-130B, and ERNIE-4.0-8K succeed in three tools, and Qwen-turbo succeeds in two tools, we denote the TASK0 Progress for these comparative LLMs as 100%, 90%, 90%, and 80% in Fig. 6, respectively. This TASK0 Progress tests the function calling ability of the LLM as the brain in LAC, and Doubao is the best LLM in this respect.

In order to complete the user input successfully, the right planning and decomposition are important for LAC, and every decomposed subtask has a corresponding tool to achieve it. Thus, the correct order of tools to be invoked is important, and we express this characteristic through Right Sequences in Fig. 6. Doubao-pro-4k, GLM3-130B, and ERNIE-4.0-8K can obtain the Right Sequences, while the other two have issues when connecting subsequent tools.

The inference time is important for LAC, as the embodied robot interacts with the environment in real-time. The average times for the comparative LLMs in controlling LAC to execute TASK0 are 80.6 seconds, 118.7 seconds, 81.5 seconds, 44.2 seconds, and 51.35 seconds, respectively, as illustrated in Fig. 6. Moonshot-v1-128k takes the longest time because it falls into unnecessary inner loops, and Qwen-turbo takes the shortest time because it skipped several necessary inner loops and failed to complete TASK0.

Finally, we summarize a brief critique of the comparative LLMs. Doubao-pro-4k is time-consuming and should be optimized to make it suitable for deployment in real robots. Moonshot-v1-128k is even more time-consuming and falls into unnecessary loops, so it should improve its reasoning ability. GLM3-130B is time-consuming and has issues with tool inconsistency, so it should enhance its function calling

ability. Qwen-turbo is the fastest LLM but fails to reflect on the information feedback from the environment and should improve its tool invocation capability. ERNIE-4.0-8K is fast for inference but has strict constraints on message structure, which limits the flexibility of LAC.

V. Conclusion

The proposed LAC is a general control paradigm that combines the intelligence of LLMs with the real-time response of feedback control mechanisms to realize embodied intelligence. The simulation results demonstrate that a robot equipped with LAC becomes an embodied robot, capable of interacting with environments and responding to user input autonomously, similar to humans. However, this work assumes that the tools are ideal, and the construction of high-quality tools themselves is a complex problem. In addition, the LAC is time-consuming based on the given LLMs, which makes the robot react slowly. To save time, a customized LLM should be fine-tuned to fit the LAC.

This work represents the initial idea of LAC. For future studies, complex simulations in Isaac Sim can be conducted. Additionally, true tools should be constructed to realize the embodiment of LAC. Moreover, a realistic humanoid robot should be deployed based on LAC to ultimately verify the effectiveness of the LAC control paradigm.

Acknowledgment

This research was funded by Scientific Research Foundation of NBUT under Grant No. 2170011540012. The help of Joyson Robot is gratefully acknowledged. The support of Major Special Project of Ningbo High-Tech Zone under Grant No. 2024CX050007 is gratefully acknowledged. The support of Open Fund of Key Laboratory of Flight Techniques and Flight Safety, CAAC under Grant No. FZ2022KF17 is gratefully acknowledged.

References

[1] Dubey, Abhimanyu, et al. "The llama 3 herd of models." *2024, arXiv:2407.21783*.

[2] Duan, Jiafei, et al. "A survey of embodied ai: From simulators to research tasks." *IEEE Transactions on Emerging Topics in Computational Intelligence* 6.2 (2022): 230-244.

[3] Majumdar, Arjun, et al. "Where are we in the search for an artificial visual cortex for embodied intelligence?" *Advances in Neural Information Processing Systems* 36 (2023): 655-677.

[4] Iida, Fumiya, and Fabio Giardina. "On the timescales of embodied intelligence for autonomous adaptive systems." *Annual Review of Control, Robotics, and Autonomous Systems* 6.1 (2023): 95-122.

[5] Ghith, Ehab Saif, and Farid Abdel Aziz Tolba. "Tuning PID controllers based on hybrid arithmetic optimization algorithm and artificial gorilla troop optimization for micro-robotics systems." *IEEE access* 11 (2023): 27138-27154.

[6] Katayama, Sotaro, Masaki Murooka, and Yuichi Tazaki. "Model predictive control of legged and humanoid robots: models and algorithms." *Advanced Robotics* 37.5 (2023): 298-315.

[7] Shentu, Yide, et al. "From LLMs to Actions: Latent Codes as Bridges in Hierarchical Robot Control." *2024, arXiv:2405.04798*.

[8] Long, Yonghao, et al. "Human-in-the-loop embodied intelligence with interactive simulation environment for surgical robot learning." *IEEE Robotics and Automation Letters* 8.8 (2023): 4441-4448.

[9] Fan, Wenqi, et al. "A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models." *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2024.

[10] Sandini, Giulio, Alessandra Sciutti, and Pietro Morasso. "Artificial cognition vs. artificial intelligence for next-generation autonomous robotic agents." *Frontiers in Computational Neuroscience* 18 (2024): 1349408.

[11] Driess, Danny, et al. "Palm-e: An embodied multimodal language model." *2023, arXiv:2303.03378*.

[12] Li, Lei, et al. "M^{3}IT: A Large-Scale Dataset towards Multi-Modal Multilingual Instruction Tuning." *2023, arXiv:2306.04387*.

[13] Dzabraev, Maksim, Alexander Kunitsyn, and Andrei Ivaniuta. "VLRM: Vision-Language Models act as Reward Models for Image Captioning." *2024, arXiv:2404.01911*.

[14] Li, Xinghang, et al. "Vision-language foundation models as effective robot imitators." *2023, arXiv:2311.01378*.

[15] Lin, Ji, et al. "Vila: On pre-training for visual language models." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024.

[16] Li, Boyi, et al. "Interactive task planning with language models." 2023, arXiv:2310.10645.

[17] Brohan, Anthony, et al. "Do as I can, not as I say: Grounding language in robotic affordances." *Conference on robot learning*. PMLR, 2023.

[18] Liang, Jacky, et al. "Code as policies: Language model programs for embodied control." *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023.

[19] Singh, Ishika, et al. "Progprompt: Generating situated robot task plans using large language models." *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023.

[20] Brohan, Anthony, et al. "Rt-2: Vision-language-action models transfer web knowledge to robotic control." *2023, arXiv:2307.15818*.

[21] Zhang, Jenny, et al. "Good time to ask: A learning framework for asking for help in embodied visual navigation." *2023 20th International Conference on Ubiquitous Robots (UR)*. IEEE, 2023.

[22] Lu, Kai, et al. "Learning to Catch Reactive Objects with a Behavior Predictor." *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024.

[23] Liu, Peiqi, et al. "Ok-robot: What really matters in integrating open-knowledge models for robotics." *2024, arXiv:2401.12202*.

[24] Shen, Yongliang, et al. "Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face." *Advances in Neural Information Processing Systems* 36 (2024).

[25] Kirillov, Alexander, et al. "Segment anything." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023.

[26] Chang, Matthew, et al. "Goat: Go to any thing." *2024, arXiv:2311.06430*.

[27] Fang, Hao-Shu, et al. "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains." *IEEE Transactions on Robotics* (2023).

[28] Shinn, Noah, et al. "Reflexion: Language agents with verbal reinforcement learning." *Advances in Neural Information Processing Systems* 36 (2024).

[29] Liu, Zeyi, Arpit Bahety, and Shuran Song. "Reflect: Summarizing robot experiences for failure explanation and correction." *2023, arXiv:2306.15724*.

[30] Turpin, Miles, et al. "Language models don't always say what they think: unfaithful explanations in chain-of-thought prompting." *Advances in Neural Information Processing Systems* 36 (2024).

[31] Yao, Shunyu, et al. "ReAct: Synergizing Reasoning and Acting in Language Models." *International Conference on Learning Representations (ICLR)*. 2023.

[32] Madaan, Aman, et al. "Self-refine: Iterative refinement with self-feedback." *Advances in Neural Information Processing Systems* 36 (2024).

[33] Gou, Zhibin, et al. "Critic: Large language models can self-correct with tool-interactive critiquing." 2023, *arXiv:2305.11738*.

[34] An, Shengnan, et al. "Learning from mistakes makes llm better reasoner." *2023, arXiv:2310.20689*.

[35] Huang, Xu, et al. "Recommender ai agent: Integrating large language models for interactive recommendations." *2023, arXiv:2308.16505*.