Optimal Control Synthesis from Natural Language: Opportunities and Challenges

Christopher E. Mower,[1] Hongzhan Yu[1], Antoine Grosnit[1,3], Jan Peters[3], Jun Wang[2], Haitham Bou-Ammar[1]

[1]Huawei Noah's Ark Lab, London, UK.

[2]University College London, London, UK.

[3]Technical University of Darmstadt, Darmstadt, Germany.

Correspondence: christopher.mower@huawei.com, jun.wang@cs.ucl.ac.uk, and haitham.ammar@huawei.com.

Embarking on the frontier of innovation, large language models pave the way for the seamless automatic generation of optimal controllers.

Many advancements are being made in the fields of robotics and machine learning. An essential consideration for the real-world deployment of autonomous systems within society is captured by the question: *Can we enable non-robotics engineers to effectively program robots*? While others attempted to address this problem via physical interaction [10], shared control [13], low-cost teleoperation [22], imitation learning from video [17, 2], and myoelectic and inertial interfaces [7], there is still however a steep learning curve and need for domain experts - still requiring expensive and often time consuming training.

Language, whether spoken or written text, is one of the most natural forms of communication between humans without requiring training. If we are to enable non-robotics engineers to program robots, we need to incorporate into the interface a means to synthesize controllers from natural language. Therefore the problem above is two fold: the first is to understand natural language and the second is to map the understanding of natural language to robotic affordances and behavior.

The first problem has been largely solved by large language models. Language models have been shown to be capable of handling a diverse array of tasks such as generating code [3], advanced mathematics [5, 18], generating correct chain-of-thoughts [20, 19], search [6], and many others. Although these language models generate embeddings that appear to grasp language, we seek a method to link them to control [4]. More specifically in the robotics literature,

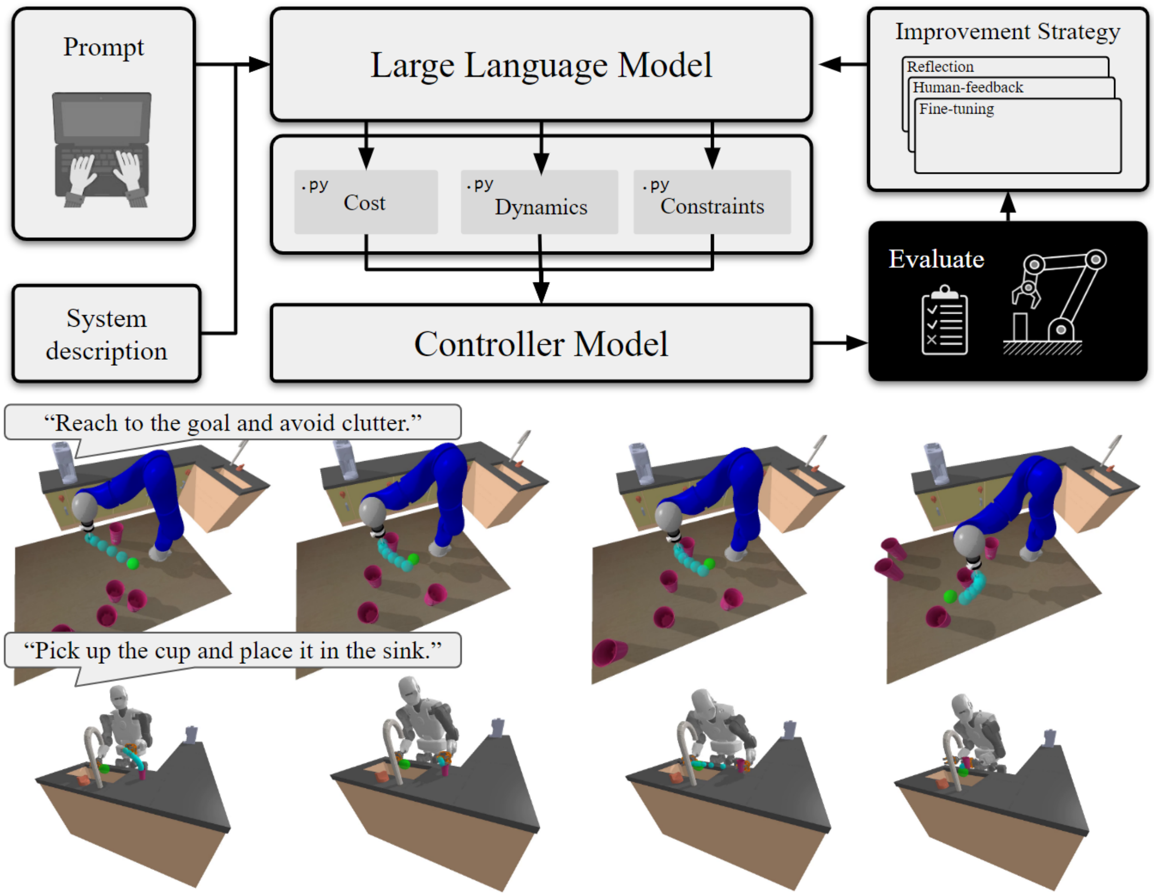

Figure 1: Overview of approach for robot control synthesis from natural language. Using the pipeline shown at the top of the image, we were able to synthesize two model-predictive controllers for each tasks: 1) reaching in clutter, and 2) pick-and-place.

mapping natural language to low-level control commands has been explored in several contexts: structured language [8, 12], code generation [9], reward design [11, 21], and human feedback [15]. Our goal is to create a framework that enables the language model to refine its latent embeddings so as to maximize controller performance on the robot and model constraints.

Several key elements are required for such a framework. 1) A method to map natural language into a controller representation. A generic optimal controller is comprised of three key parts: i) cost function dictating the goal or optimal behavior, ii) dynamics model representing the equations of motion, and iii) safety and physical limitation constraints. When these three parts are available, we can pass to an optimization or sampling based solver to generate executable control actions. 2) Given a process by which to map natural language to a controller we require a model that gives us relevant inputs that can be incorporated into those algorithms - i.e. code for the cost, dynamics, and constraint functions. 3) These controllers must then be able to interface with the robot and execute controls on the system. Given several sampled controllers, we need an approach to identify more ideal controllers. Following the reward design literature [16], we define a performance measure, typically referred to as a *fitness* function (e.g. average distance to the goal after several trials). 4) Given the designed code and fitness value, we need the language model to have the capability to improve and modify its latent space to improve further and maximize fitness. A wealth of methods could be applied in this step, such as: prompt engineering, fine-tuning, reflection, human feedback. Prompt engineering tends to be tedious to implement, fine-tuning requires a wealth of data, however, given recent literature in the machine learning community [11], reflection seems plausible.

With these fundamental components, as described above, we are inspired to establish the pipeline illustrated in the upper section of Figure 1. The lower section showcases the effective application of our framework to various robotic tasks.

How to generate optimal controllers from language?

A generic framework (shown in Figure 1) for the automatic generation of optimal control involves several key steps: a) An initial prompt from a human user and system description. b) A controller model is generated via a large language model and passed to a solver (e.g. numerical optimization). c) Evaluations are performed on the real system or in simulation and a fitness value is computed. d) The evaluation results are converted into a reflection prompt which is used to guide future improvements in the controller model. Steps b-d are repeated for several iterations. Our findings reveal, for the first time, the capability of language models in generating system dynamics and constraints.

The system receives an initial textual prompt that includes a description of the task from a non-expert user. Our lab experiments, supported by literature [11], reveal that the language model also necessitates a comprehensive system description. Where we deviate from the aforementioned literature, is that we show that the environment code is unnecessary. In our case, a description of the state, action, and parameter space is required.

The language model is directed to generate code as output (Python in our case), representing the controller model as a cost function (also known as the objective or reward function), a dynamics model capturing equations of motion, and constraints embodying the physical limitations of the system. These functions are then fed into a numerical optimization solver (e.g., IPOPT, SNOPT, KNITRO). Often, these solvers require function gradients - a practical challenge. Handling this issue has two potential options: 1) Seeking analytic gradients from the language model, leveraging its mathematical abilities [5], although this approach may be error-prone. 2) Alternatively, one could instruct the language model to utilize a library such as CasADi [1] that provides derivatives via automatic differentiation. Note, ensuring the model's familiarity with such a library can be achieved through fine-tuning or perhaps a few-shot prompting. Given the possibility of code bugs, generating multiple samples concurrently seems to be advisable.

Upon coupling the controller model with an optimization solver, multiple evaluations can be conducted in simulation or on an actual system. From these evaluations, a fitness value is computed. For instance, in tasks like reaching in obstacle-filled environments, the fitness score considers factors such as the number of goal positions acquired within a fixed time and the occurrences of collisions with obstacles.

Upon collecting evaluation metrics and computing a fitness value, a reflection prompt is generated, encompassing this information and any code errors. Work by Ma et al. [11] include in the reflection prompt additional details from the reinforcement learning algorithm, such as the learning curve. The language model's response, along with the reflection prompt, is then appended to the initial prompt, guiding the regeneration of a new controller.

Key problems for the future

The process described above yields a controller model derived from natural language. We have demonstrated the viability of the approach with two robot tasks, however there are several open problems for the future that we summarize as follows.

1) **Lack of reflection in language models**. A critical challenge arises in the face of insufficient reflective capabilities within language models. Reflection, in this context, refers to the model's ability to introspect and analyze its own decision-making processes in order to improve on the generated controller model. Specific examples include fixing code bugs within the context of the assigned task and exploring alternative behavioral models. The deficiency in reflection hampers the model's capacity to learn from its own experiences and adapt over time. Furthermore, the fitness function may not be something easily defined for generic tasks. Either we could utilize the language model to also generate the fitness or incorporate human-feedback in the reflection prompt. In addition, text-only based reflection is potentially limiting. Take, for example, a robot operating in the real world where it is not possible to track the pose of a variety of objects in a given scene (e.g. a house). Translating the success of the controller may be difficult given only text, whereas an image from a camera could be vastly more informative. Utilizing vision-language models, e.g. [14], could be useful for improving the likelihood of high performance in the real world.

2) **Handling dynamic tasks with complex physics models**. Language models exhibit competence in generating controllers which yield action sequences that successfully accomplish multi-goal tasks, such as pick-and-place operations; see Fig. 1. However, a notable difficulty arises when these models confront dynamic tasks involving more intricate physics models, especially those associated with contact dynamics (e.g. non-prehensile manipulation). Complex interactions, such as pushing an object on a table and other underactuated systems introduce challenges in accurately predicting the consequences of actions. This limitation poses a significant hurdle when attempting to apply language models to scenarios with unpredictable elements, emphasizing the need for advancements that can handle the nuances of dynamic environments and intricate physical phenomena.

3) **Ensuring safety of the system**. A critical consideration for these systems, utilizing language models for optimal control, is the necessity for meticulous care throughout the development and deployment process - these models, although powerful, are not immune to human error. During the development of our work, a check on the maximum velocity of the robot was forgotten (a somewhat typical human-error). These typical human-errors could potentially result in unsafe actuation of robotic systems. Therefore, great caution must be exercised during the developmental phases, emphasizing robust testing, validation, and the incorporation of fail-safe mechanisms. This aligns with the broader challenge of maintaining safety and reliability in autonomous systems, where the consequences of errors can have tangible impacts on the physical world.

In summary, synthesizing controller models from natural language holds both great promises and challenges. Managing language model reflections, navigating dynamic physical complexities, and ensuring safety are key considerations. As we progress, we must navigate the delicate interplay of linguistic expression and robotic actuation, shaping a future of innovative and responsible autonomous control.

References

[1] Joel A E Andersson et al. "CasADi – A software framework for nonlinear optimization and optimal control". In: *Mathematical Programming Computation* 11.1 (2019), pp. 1–36. DOI: `10.1007/s12532-018-0139-4`.

[2] Shikhar Bahl et al. "Affordances from Human Videos as a Versatile Representation for Robotics". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 13778–13790.

[3] Mark Chen et al. "Evaluating Large Language Models Trained on Code". In: (2021). arXiv: `2107.03374 [cs.LG]`.

[4] Filippos Christianos et al. *Pangu-Agent: A Fine-Tunable Generalist Agent with Structured Reasoning*. 2023. arXiv: `2312.14878 [cs.AI]`.

[5] Simon Frieder et al. "Mathematical Capabilities of ChatGPT". In: (2023). arXiv: `2301.13867 [cs.LG]`.

[6] Ehsan Kamalloo et al. "HAGRID: A Human-LLM Collaborative Dataset for Generative Information-Seeking with Attribution". In: *arXiv:2307.16883* (2023).

[7] Agamemnon Krasoulis et al. "Improved prosthetic hand control with concurrent use of myoelectric and inertial measurements". In: *Journal of neuroengineering and rehabilitation* 14 (2017), pp. 1–14.

[8] Hadas Kress-Gazit, Georgios E Fainekos, and George J Pappas. "Translating structured english to robot controllers". In: *Advanced robotics* 22.12 (2008), pp. 1343–1359.

[9] Jacky Liang et al. "Code as policies: Language model programs for embodied control". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)* (2023), pp. 9493–9500.

[10] Dylan P. Losey et al. "Physical interaction as communication: Learning robot objectives online from human corrections". In: *The International Journal of Robotics Research* 41.1 (2022), pp. 20–44. DOI: 10.1177/02783649211050958.

[11] Yecheng Jason Ma et al. *Eureka: Human-Level Reward Design via Coding Large Language Models*. 2023. arXiv: 2310.12931 [cs.RO].

[12] Cynthia Matuszek et al. "Learning to Parse Natural Language Commands to a Robot Control System". In: *Experimental Robotics: The 13th International Symposium on Experimental Robotics* (2013). Ed. by Jaydev P. Desai et al., pp. 403–415. DOI: 10.1007/978-3-319-00065-7_28.

[13] Christopher E Mower, Joao Moura, and Sethu Vijayakumar. "Skill-based Shared Control". In: *Proceedings of Robotics: Science and Systems* (July 2021). DOI: 10.15607/RSS.2021.XVII.028.

[14] Alec Radford et al. "Learning Transferable Visual Models From Natural Language Supervision". In: *Proceedings of the 38th International Conference on Machine Learning*. Proceedings of Machine Learning Research 139 (18–24 Jul 2021). Ed. by Marina Meila and Tong Zhang, pp. 8748–8763.

[15] Pratyusha Sharma et al. "Correcting Robot Plans with Natural Language Feedback". In: *Proceedings of Robotics: Science and Systems* (June 2022). DOI: 10.15607/RSS.2022.XVIII.065.

[16] Satinder Singh, Richard L Lewis, and Andrew G Barto. "Where do rewards come from". In: *Proceedings of the annual conference of the cognitive science society* (2009), pp. 2601–2606.

[17] Aravind Sivakumar, Kenneth Shaw, and Deepak Pathak. "Robotic Telekinesis: Learning a Robotic Hand Imitator by Watching Humans on YouTube". In: *Proceedings of Robotics: Science and Systems* (June 2022). DOI: 10.15607/RSS.2022.XVIII.023.

[18] Trieu H. Trinh et al. "Solving olympiad geometry without human demonstrations". In: *Nature* 625.7995 (Jan. 2024), pp. 476–482. DOI: 10.1038/s41586-023-06747-5.

[19] Rasul Tutunov et al. "Why Can Large Language Models Generate Correct Chain-of-Thoughts?" In: (2023). arXiv: 2310.13571 [cs.CL].

[20] Jason Wei et al. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models". In: ed. by Sanmi Koyejo et al. 2022.

[21] Wenhao Yu et al. "Language to Rewards for Robotic Skill Synthesis". In: *Arxiv preprint arXiv:2306.08647* (2023).

[22] Tony Z. Zhao et al. “ALOHA: Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware”. In: *Proceedings of Robotics: Science and Systems* (July 2023). DOI: 10.15607/RSS.2023.XIX.016.