Towards Agentic AI on Particle Accelerators

Antonin Sulc
Helmholtz Zentrum Berlin
Berlin, Germany
antonin.sulc@helmholtz-berlin.de

Thorsten Hellert
LBNL,
Berkeley, USA
thellert@lbl.gov

Raimund Kammering
DESY,
Hamburg, Germany
raimund.kammering@desy.de

Hayden Houscher
FNAL,
Batavia, IL, USA
haydenh@fnal.gov

Jason St. John
FNAL,
Batavia, IL, USA
stjohn@fnal.gov

Abstract

As particle accelerators grow in complexity, traditional control methods face increasing challenges in achieving optimal performance. This paper envisions a paradigm shift: a decentralized multi-agent framework for accelerator control, powered by Large Language Models (LLMs) and distributed among autonomous agents. We present a proposition of a self-improving decentralized system where intelligent agents handle high-level tasks and communication and each agent is specialized to control individual accelerator components.

This approach raises some questions: What are the future applications of AI in particle accelerators? How can we implement an autonomous complex system such as a particle accelerator where agents gradually improve through experience and human feedback? What are the implications of integrating a human-in-the-loop component for labeling operational data and providing expert guidance? We show three examples, where we demonstrate the viability of such architecture.

1 Introduction

Particle accelerator operation involves a complex interplay of interrelated systems, each requiring precise control and optimization. Traditionally, these systems have been managed through a combination of human expertise and highly specialized algorithms with recent developments including machine learning techniques [8, 9] where some are successfully deployed in operation like *e.g.* [10, 14, 34].

Most of these algorithms are designed to operate in the narrow domain of accelerator functionality. The systems have noise, tight operational limits, and deterministic interplay through the control system to ensure human and equipment safety, preventing scenarios that could lead to hazardous conditions or equipment damage. While effective, this approach can sometimes struggle to achieve optimal overall performance due to the challenges of integrating these disparate systems. Even when a sufficiently comprehensive model of the facility is constructed, it can be susceptible and prone to problems as the accelerator transitions through different states or experiences drifts and thus requires human intervention for setup and ongoing maintenance.

This paper proposes a paradigm shift in accelerator control systems, envisioning a future where self-improving agents control separated sub-components using algorithms that inform operators of significant events. In this framework, we show how autonomous agents controlled by LLMs serve not only in the creation and retrieval of accelerator operation documentation but also as high-level coordinators, handling complex tasks such as interpreting the machine state from analyzing control system channels or inter-agent communication with a degree of autonomy in decision-making and

Machine Learning and the Physical Sciences Workshop, NeurIPS 2024.

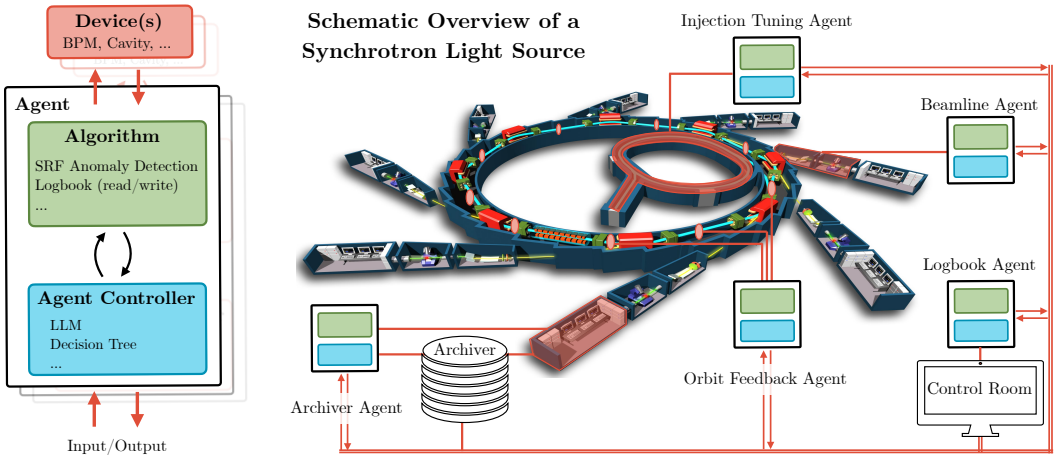

Figure 1: Schematic overview of a decentralized agent-based control architecture. The left diagram illustrates the modular structure where each agent, consisting of devices, algorithms, and agent controllers, manages specific subsystems. The right figure shows these agents interacting with both the physical components of the accelerator and the control room, enabling prompting current state, monitoring, decision-making, and adaptive responses to complex operational scenarios.

communication. Meanwhile, dedicated agents controlled by LLMs execute specific, well-established calculations or operations within their domains of expertise. This decentralized approach, illustrated in our conceptual diagram Fig. 1, showcases a potential future for the operation of particle accelerators with the flexibility to deploy more advanced agents.

By exploring this forward-looking perspective, we aim to stimulate discussion on the future of particle accelerator operations and the broader implications of decentralized AI architectures in scientific instrumentation. The proposed framework offers potential solutions to the challenges of integrating disparate systems and managing complex state transitions, while also providing a flexible architecture that can adapt to evolving requirements and technological advancements.

2 Related Work

Machine learning techniques have increasingly been applied to various aspects of accelerator physics in recent years. Edelen *et al*. [8, 9] provide a comprehensive review of machine learning applications in particle accelerators, covering areas such as beam diagnostics, control systems, and performance optimization.

2.1 Agentic AI and LLMs in Complex Reasoning Tasks

There is a rise of agentic AI that has demonstrated remarkable capabilities in complex reasoning tasks [3]. According to [31], an effective agent in an agentic system should possess autonomy in independent operation, reactivity in environmental response, and proactiveness in pursuing its objectives, enabling it to function intelligently and adaptively in complex environments. We will highlight the most relevant works that align with our problem.

Yao *et al*. [32] introduced the ReAct framework, which showed how LLMs can effectively combine reasoning and acting in multi-step tasks and is widely used to operate in complex multi-step reasoning.

The use of AI agents to effectively engage with their environment and complete a wide array of tasks has been gaining increasing popularity. Park *et al*. [22] introduce *generative agents* that use LLMs to simulate human behavior in interactive environments, showcasing how LLMs can be utilized to model intricate multi-agent interactions and environmental complexities. Wang *et al*. [28] uses an LLM-powered agent to autonomously explore, acquire skills, and make discoveries in unfamiliar environments using components like an automatic curriculum and skill library. AgentVerse [4] shows a framework for collaborative groups of LLM-powered expert agents, showing improved performance over single agents in various tasks. This multi-agent approach leverages LLMs for

high-level tasks while enabling specialized components to work together autonomously on complex problems. AutoGen [30] framework allows building LLM applications using multiple conversable agents. They show that autonomous, communicating agents can control subcomponents of a system. Particle accelerator control often utilizes operation sequencing [2, 11], there LLM+P [20] can help by translating natural language into formal planning languages, potentially bridging the gap between planning, natural language, and accelerator controls. To deal with environment changes, Shinn *et al*. [26] proposes Reflexion, a framework that reinforces language agents through linguistic feedback rather than traditional weight updates. Reflexion agents verbally reflect on task feedback and maintain reflective text in an episodic memory buffer to improve decision-making in subsequent trials. To impose the factuality, [7] proposes *multiagent debate* where multiple LLMs debate to improve reasoning and factual accuracy of results which is essential for reliable operations. LLMs have gained prominence as coding assistants [17] demonstrating the capability to function as coders [24], problem solvers [25] or even data scientists [19]. Automated Design of Agentic Systems [16] aims to automatically create AI agent designs by having a meta-agent iteratively program new agents in code.

2.2 LLMs as Assistants in Operations

Carrasco *et al*. [3] explores fine-tuned LLMs for autonomous spacecraft control in simulations, showing their efficacy in handling language-based inputs and outputs. It shows the potential of using LLMs for complex tasks like accelerator control.

In accelerators, Mayet's [21] GAIA system uses LLMs with the ReAct framework [32] to assist in operations by integrating multiple expert tools *e.g*. knowledge retrieval, machine control, and Python script generation for autonomous and semi-autonomous management of complex accelerator environments.

In [27] they show joint efforts across particle accelerator facilities to utilize Retrieval Augmented Generation (RAG) models and other AI techniques for enhancing eLogs usability and automation, aiming to unlock operational insights and improve data accessibility.

3 Features of the System

Four key aspects of agents can significantly enhance operational effectiveness. First, operating particle accelerators is complex and requires extensive human expertise. We suggest that agents can gradually improve through experience [26] where the system incorporates continuous learning from operational data and **learns** based on outcomes or from "*human-in-the-loop*". It can be an important step forward since most of these particle accelerators operate in already high reliabilities above 90%.

Second, the potential to uncover **causal relationships** in accelerator operations by adding Reasoning Agents [13]. LLM-powered agents can provide a human-readable interface to complex machine operations, enabling faster exploration of causal connections via reasoning about the information available (or gradually reasoning about it via ReAct [32]). By integrating agents designed to reveal casual relationships, we aim to enhance system interpretability and streamline diagnostics. This approach could reduce the learning curve for new operators and offer valuable insights to experienced physicists, leading to more efficient accelerator management.

The third key component is agent **autonomy** implementable via rules, decision trees, or language models. LLMs are currently preferred due to their natural language interface and intelligent decision-making. LLM's high computational needs may cause delays, limiting use in real-time control. The choice of autonomy implementation thus requires balancing LLMs' flexibility against performance needs in time-sensitive applications.

Lastly, other interesting examples of agents can be: planning agent [20] to execute and run complex plans with standard tools, such as to execute tasks at *e.g*. European XFEL defined in Taskomat [11] or [2] at Fermi, coding agents [24], and data scientist agents [19] to summarize.

4 Examples

In this section, we present three examples, where the aforementioned features can be directly applicable by integrating them into the control systems like [5, 15, 23] and enhancing operational efficiency and stability: the Advanced Light Source (ALS) orbit feedback system, the European XFEL longitudinal feedback manager and Fermi coding assistant.

4.1 ALS Example: Orbit Feedback

Maintaining precise control over the beam orbit through orbit feedback systems is crucial for the stable operation of a storage ring. However, this task can be complicated in practice by machine drifts caused by environmental factors or by maintenance activities that alter the operational characteristics of specific sections of the ring. Under the current operational paradigm, when the orbit feedback system fails to converge during user operation, a physicist must intervene. This process typically involves a detailed analysis of the situation, comparing current conditions with previous runs, and relying heavily on experience to make a judgment call on how to adjust the system for optimal performance.

As illustrated in Fig. 2, in our envisioned framework, this diagnostic role would be managed by a specialized feedback agent. Upon detecting abnormal behavior, the agent would identify the underperforming area and consult the logbook agent for recent events that could explain the issue. For instance, if maintenance work had occurred in the affected sector, the logbook agent would provide this information. The feedback agent would then draft a report outlining the problem, its likely root cause, and a suggested course of action, which would be sent to the control room for review.

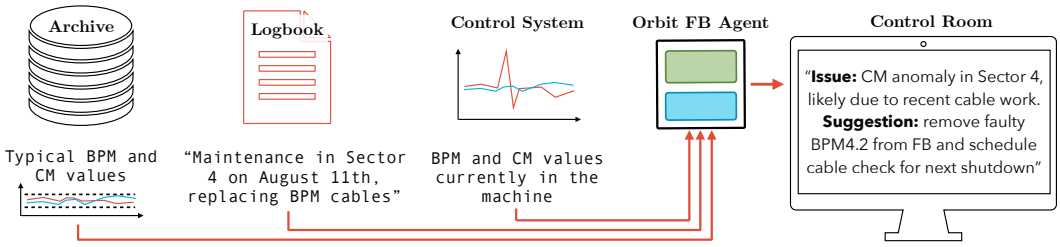

Figure 2: Diagram of the orbit feedback agent. The Orbit FB Agent detects abnormal behavior, consults the Logbook Agent for relevant events, and suggests an action for the control room to review.

4.2 European XFEL Example: Longitudinal Feedback Manager

The particle beam in the linear accelerator, like the European XFEL, is stabilized within well-defined, narrow parameter spaces through the use of various feedback loops. To effectively orchestrate the complex interplay of these feedback loops, expert systems are often employed [18, 6], which encapsulate the logic of this interaction in software. A Feedback Agent can be trained to learn the possible and desired states, illustrated in Fig. 3. In the next step, such an agent would initially be able to assist the operator as a kind of recommender system and, potentially, could take over the entire operation of all longitudinal feedback loops in a later step.

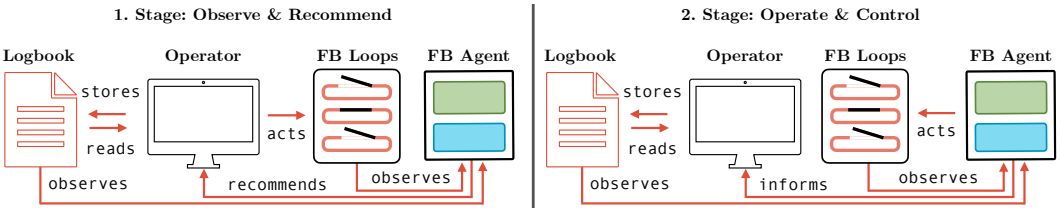

Figure 3: Diagram of the Longitudinal feedback manager. Left: The operator sets feedback loops and the FB Manager Agent and Logbook Agent observe and recommend actions to the operator. Right: FB Manager Agent takes over the operation and informs the operator.

Examples of additional agents can be: Orbit Feedback Agent for controlling transversal beam parameters; Undulator Agents for optimizing photon beam production; Beamline Agents overseeing experimental stations; and Feedback Coordination Agent acting as a central coordinator to orchestrate overall feedback operation. Each agent functions as a recommender system, assisting human operators in decision-making. As confidence in their performance grows, they could potentially take over the entire operation of individual feedback, *e.g.* through [26]. Crucially, these agents maintain constant communication with each other. Before implementing any changes, each agent informs the others of

its intended actions. This allows for a collective evaluation of the proposed changes, ensuring that they don't lead to suboptimal or dangerous configurations.

4.3 Fermilab Example: ALC Coding Assistant

Accelerator Command Language (ACL), developed at Fermilab, is a scripting language designed for non-programmers, such as engineers and accelerator operators, to control complex accelerator systems. Despite its specialized nature, AI agents can assist with ACL, even without specific training on the language itself.

A Code Retrieval Agent utilizes a preexisting semantic search API to query ACL's extensive documentation, retrieving relevant code snippets based on detailed descriptions. After gathering these results, the Retrieval Agent reviews the information and selects the most relevant snippets to fulfill the user's request. These snippets are then passed to a Code Generation Agent, which uses the retrieved information to infer the correct ACL code structure and generate context-aware script suggestions using *e.g.* [12].

This two-step process simplifies working with ACL and provides intelligent coding assistance, helping users efficiently generate scripts for a proprietary language specific to Fermilab.

However, one potential challenge is that the initially generated code may not always be correct. To address this, combining Reflexion [26] for verbal reinforcement learning and embeddings for long-term memory such as [33] could help guide the model toward generating fully functional code. This approach would enable the model to iteratively refine its outputs based on feedback and past experiences, ensuring more accurate results over time.

5 Safety Issues and Hallucinations

The tendency of LLMs to hallucinate creates substantial risks for critical system applications. To mitigate potential issues arising from hallucinations and uncertainty, several approaches can be implemented. These include constraining output using defined grammars or regular expressions [29], employing multiple agents to cross-check decisions [1], and integrating real-time sensor data feedback loops.

It's important to emphasize that this proposal is for an AI-based, high-level control system that would operate alongside existing control systems. These existing systems have built-in safety measures for protecting equipment and personnel, which would remain independent of the proposed AI control. Our approach is to begin with passive monitoring and operator suggestions before progressing to limited control of non-critical systems.

Furthermore, fallback mechanisms should automatically revert to traditional control systems if anomalies are detected. Continuous validation against established algorithms and expert decisions will be an integral part of the proof of concept. The primary role of the system in this context remains sensing and informing human operators rather than autonomous control, thereby maintaining critical human oversight in accelerator operations.

6 Conclusion

This paper shows a paradigm shift in particle accelerator control through a decentralized multi-agent framework powered by LLMs. By integrating AI agents for high-level tasks and specialized agents for component management, we address the increasing complexity of modern accelerator systems.

Three examples demonstrate the potential of AI agents in assisting complex accelerator tasks, opening the way for higher autonomy and posing a question of the use case of intelligent agents for assistance with operating particle accelerators. This approach offers exciting possibilities for improved performance and our practical examples showcase a vision for more intelligent and adaptive accelerator operations.

References

[1] Maciej Besta, Lorenzo Paleari, Ales Kubicek, Piotr Nyczyk, Robert Gerstenberger, Patrick Iff, Tomasz Lehmann, Hubert Niewiadomski, and Torsten Hoefler. CheckEmbed: Effective Verification of LLM Solutions to Open-Ended Tasks, June 2024.

[2] Timofei Bolshakov, AD Petrov, and Sharon Lackey. Synoptic display—a client-server system for graphical data representation. *proceedings of the 2003 ICALEPCS, Gyeongju, Korea*, 2003.

[3] Alejandro Carrasco, Victor Rodriguez-Fernandez, and Richard Linares. Fine-tuning llms for autonomous spacecraft control: A case study using kerbal space program, 2024.

[4] Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *The Twelfth International Conference on Learning Representations*, 2023.

[5] L. Dalesio, J. Hill, M. Kraimer, S. Lewis, D. Murray, S. Hunt, W. Watson, M. Clausen, and J. Dalesio. The experimental physics and industrial control system architecture: past, present, and future. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 352:179–184, 1994.

[6] Hannes Dinter. *Longitudinal diagnostics for beam-based intra bunch-train feedback at FLASH and the European XFEL*. PhD thesis, Staats-und Universitätsbibliothek Hamburg Carl von Ossietzky, 2018.

[7] Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

[8] Auralee Edelen and Xiaobiao Huang. Machine learning for design and control of particle accelerators: A look backward and forward. *Annual Review of Nuclear and Particle Science*, 74(1):557–581, 2024.

[9] Auralee Edelen, Christopher Mayes, Daniel Bowring, Daniel Ratner, Andreas Adelmann, Rasmus Ischebeck, Jochem Snuverink, Ilya Agapov, Raimund Kammering, Jonathan Edelen, et al. Opportunities in machine learning for particle accelerators. *arXiv preprint arXiv:1811.03172*, 2018.

[10] Annika Eichler, Florian Burkart, Jan Kaiser, Willi Kuropka, Oliver Stein, Erik Bründermann, Andrea Santamaria Garcia, and Chenran Xu. First steps toward an autonomous accelerator, a common project between desy and kit. *Proc. IPAC'21*, pages 2182–2185, 2021.

[11] L Fröhlich, O Hensler, U Jastrow, M Walla, and J Wilgen. Taskomat & taskolib: A versatile, programmable sequencer for process automation. *PCaPAC 2022 hosted by ELI Beamlines*, page 94, 2022.

[12] Jing Guo, Nan Li, Jianchuan Qi, Hang Yang, Ruiqiao Li, Yuzhen Feng, Si Zhang, and Ming Xu. Empowering working memory for large language model agents. *arXiv preprint arXiv:2312.17259*, 2023.

[13] Kairong Han, Kun Kuang, Ziyu Zhao, Junjian Ye, and Fei Wu. Causal agent based on large language model. *arXiv preprint arXiv:2408.06849*, 2024.

[14] Thorsten Hellert, Tynan Ford, Simon C. Leemann, Hiroshi Nishimura, Marco Venturini, and Andrea Pollastro. Application of deep learning methods for beam size control during user operation at the advanced light source. *Phys. Rev. Accel. Beams*, 27:074602, Jul 2024.

[15] O Hensler and K Rehlich. Doocs: A distributed object oriented control system. In *Proceedings of XV Workshop on Charged Particle Accelerators, Protvino*, 1996.

[16] Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. *arXiv preprint arXiv:2408.08435*, 2024.

[17] Juyong Jiang, Fan Wang, Jiasi Shen, Sungju Kim, and Sunghun Kim. A survey on large language models for code generation. *arXiv preprint arXiv:2406.00515*, 2024.

[18] Raimund Kammering and Christian Schmidt. Feedbacks and automation at the free electron laser in hamburg (flash). *Proc. ICALEPCS'13*, pages 1345–1347, 2013.

[19] Yuhang Lai, Chengxi Li, Yiming Wang, Tianyi Zhang, Ruiqi Zhong, Luke Zettlemoyer, Wen-tau Yih, Daniel Fried, Sida Wang, and Tao Yu. Ds-1000: A natural and reliable benchmark for data science code generation. In *International Conference on Machine Learning*, pages 18319–18345. PMLR, 2023.

[20] Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. Llm+ p: Empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477*, 2023.

[21] Frank Mayet. Gaia: A general ai assistant for intelligent accelerator operations. *arXiv preprint arXiv:2405.01359*, 2024.

[22] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22, 2023.

[23] James Patrick. The fermilab accelerator control system. *Proc. ICAP'06*, pages 246–249, 2006.

[24] Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, et al. Chatdev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15174–15186, 2024.

[25] Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 2023.

[26] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[27] Antonin Sulc, Alex Bien, Annika Eichler, Daniel Ratner, Florian Rehm, Frank Mayet, Gregor Hartmann, Hayden Hoschouer, Henrik Tuennermann, Jan Kaiser, et al. Towards unlocking insights from logbooks using ai. *arXiv preprint arXiv:2406.12881*, 2024.

[28] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

[29] Brandon T Willard and Rémi Louf. Efficient guided generation for llms. *arXiv preprint arXiv:2307.09702*, 2023.

[30] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*, 2023.

[31] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*, 2023.

[32] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

[33] Kechi Zhang, Jia Li, Ge Li, Xianjie Shi, and Zhi Jin. Codeagent: Enhancing code generation with tool-integrated agent systems for real-world repo-level coding challenges. *arXiv preprint arXiv:2401.07339*, 2024.

[34] Z. Zhang, M. Böse, A.L. Edelen, J.R. Garrahan, Y. Hidaka, C.E. Mayes, S.A. Miskovich, D.F. Ratner, R.J. Roussel, J. Shtalenkova, S. Tomin, and G.M. Wang. Badger: The Missing Optimizer in ACR. In *Proc. IPAC'22*, number 13 in International Particle Accelerator Conference, pages 999–1002. JACoW Publishing, Geneva, Switzerland, 07 2022.