Integration of Large Language Models into Control Systems for Shared Appliances

Frédéric Montet[*,‡], Karl Löwenmark[†,‡], Marcus Liwicki[†], Fredrik Sandin[†], Jean Hennebert[*]
[*]Institute of AI and Complex Systems, iCoSys HEIA-FR, HES-SO, Fribourg, Switzerland
[†]Machine Learning, Department of Computer Science, Electrical and Space Engineering,
Lulea University of Technology, 97187 Lulea, Sweden,
e-mail(s): frederic.montet@hefr.ch, karl.lowenmark@ltu.se
[‡]Equal contribution of the authors

***Abstract*—Large Language Model (LLM)-powered chatbot agents have proven to be immensely useful in tasks, such as writing and generating essays, code, and academic text. By using frameworks such as LangChain, agents can be equipped with tools to access and analyse custom data, which facilitates bespoke applications, such as customer service agents with access to internal documents and tailored reasoning. While the focus of such applications has mainly centered around textual content, custom toolboxes could also enable agents to act in completely different use cases, for instance control theory. Nevertheless, given the non-deterministic nature of LLMs, merging them with deterministic software implies challenges in applied contexts such as privacy, multi-user interactions, and consistency. To pave the way to reliable LLM usage in various contexts, this work provides the foundation for expanding the use of LLM agents to the domain of control systems and human-centric automation. An agent-based architecture is proposed, which is then implemented within the context of a shared space heating system controlled by three personas. Finally, we evaluate the capacity of the system to deal with scenarios such as normality, erratic user behavior, conflicts of interest, and system limitations. The findings of this study highlight the potential benefits and challenges of using LLMs for appliance control.**

Keywords-control; large language models; shared appliance; LangChain; human-robot interaction; social robotics; generative models.

I. Introduction

Large Language Models (LLMs) have transformed the field of Artificial Intelligence (AI), enabling text understanding and generation on levels that mimic human capabilities [1][2] and leading some researchers to hypothesise that modern LLMs are at an early stage of artificial general intelligence [3]. The rapid evolution of LLMs has been realised in the domain of language understanding and generation in text-based tasks, such as summarising text [4], evaluating essays [5], and generating code [6]. Furthermore, LLMs have been trained with Reinforcement Learning from Human Feedback (RLHF) [7] to act as chatbot assistants, such as ChatGPT, which are estimated to have significant impact in a variety of fields such as education [8], medicine [9], and legal practice [10]. Chatbots like ChatGPT can also act as reasoning *agents* through frameworks such as LangChain [11] and HuggingFace agents, where they are equipped with *tools* that enable them to call custom built functions that implement, for instance, retrieval augmented generation (RAG) [12], loading of custom data, and advanced data analysis (e.g., llmath [13]).

The agent thus becomes able to interpret human requests, call custom-built functions, access external data, interpret results, and return the conclusions to the user [14], which has been used for a variety of tasks, such as literature reviews [15], customer service [16], mental healthcare [17], and document management [18]. Given these impressive characteristics, LLM usage in other contexts than question answering seems promising [19], but only a few cases have been explored due to the youth of these underlying technologies [20].

A. Cyber-physical heating systems

One such unexplored use case is LLM-integrated control loops, such as optimally controlling the indoor temperature in a building, in particular with multiple agents or users. Indoor heating might seem like a trivial problem, solved with a basic control system. However, increasing sustainability demands and limitations in energy production require modernisation of heating control to minimise emissions and cost; for instance, residential and commercial heating constitute around 40% of primary energy use in the EU and US. As a consequence, this has motivated research into predictive temperature control based on sensors and models in a Cyber-Physical Systems (CPS) framework.

As the complexity of such systems increases, so does the scope of human-machine interactions. For example, most office spaces currently feature employees as human-in-the-plant [21], where each employee is affected by the temperature control but has limited power over it. By giving customisability and control to users, the system switches to a humans-in-the-loop system [21], where each employee now participates in the control and is affected by the control. Such systems necessitate a human-centric design [22] that takes new questions into account, such as:

Explainability Does every user understand how to use the new system, or do some feel excluded?

Mediation How does the system resolve conflicts between preferences such as indoor temperature?

Robustness How can the system safely and efficiently take user parameters into account?

These are difficult questions to answer. For example, in explainability, there is a clear trade-off between transparency and simplicity. Transparency describes how exactly an agent decision is motivated, and simplicity describes how few data that the user must interact with. Consider the most transparent

case, where the control system explains all algorithms used to control the heating on the screen where temperature is controlled. This is certainly an overload of information for most users and an undesirable solution. Consider, on the other hand, a system in which only the temperature setting is ever divulged. This is an arcane system for the user, which could lead to frustration and resentment.

All these considerations place considerable load on the designer, who has to design a system both flexible and simple to use. Ideally, the high-level control of the CPS would be managed by a human mediator, who can take all types of context into account and make fair assessments, while users only require natural language to interact with the system through the mediator. Although human experts for every CPS would be far too expensive, the aforementioned advances in LLM chatbot technology could facilitate a human-esque mediator, serving as the controller of the heating system in a way similar to voice-controlled homes in the Internet of Things domain [23].

B. Research questions

Given the previously highlighted issues, this paper seeks to answer *how can LLM agents be used as mediators and controllers in CPS control systems?* More specifically, the aim is to investigate:

1) What architectures can be used to leverage LLMs for improving user interfaces within control systems?
2) How can LLMs incorporate user preferences and circumstances into a control system?
3) How can LLMs mediate between users with varying preferences or constraints?

In the following sections, we describe the theory and method used to implement an LLM agent as a user interface in a control loop for a case study of indoor heating, describe and discuss the results, and present a path for future work in this direction.

II. Method

In this section, we introduce the general components of an LLM controller agent and describe a case study featuring a custom agent connected to a simulated building heater.

A. LLM controller agent definitions

Figure 1 illustrates an LLM controller agent CPS system with users, an agent, and a system.

1) Users: Users represent any human willing to have an interaction with the system. The users can chat with the chatbot and ask it to update the control parameters, as well as ask questions about the system and the controller. Information about the users can be stored so that the agent can load chat history and user preferences to personalise the service.

2) Agent: The agent represents the LLM chatbot with connected tools. It has three main functions: user interaction, mediation/decision making, and system control. User interaction with one or more users is achieved with a user interface, which can be a Graphical User Interface (GUI), a chatbot interface, or a voice interface. The agent can store, load, and update user preferences and user information in a database to facilitate personalised user interactions. Preferences and user information, such as heat preference and health status, also act as constraints that affect the agent's decision making. Furthermore, the agent can answer questions related to the documentation of the system, itself, or other relevant metadata that is accessible in its knowledge base through RAG. Finally, the agent interacts with the Proportional–Integral–Derivative (PID) controller to set or get system parameters, such as temperature or heater effect.

3) System: The system is the component representing the (cyber)physical system. It can be any physical system having a software interface such as Representational State Transfer (REST), OPC Unified Architecture (OPC-UA), etc. When no user requests changes, the system must remain in a stable state that is satisfactory to most users.

TABLE I. List of tests run on the developed software.

#	Description	Type	User
1	Normal case	Normal operation	single
2	Normal case	Normal operation	multiple
3	Insistent user input	Normal operation	single
4	Insistent user input	Normal operation	multiple
14	User information gathering	Normal operation	single
15	User information gathering	Normal operation	multiple
5	Erratic user input	Disturbance	single
6	Erratic user input	Disturbance	multiple
11	Chaos	Disturbance	multiple
7	Exaggerated input, < lowest set point	System limitation	single
8	Exaggerated input, < lowest set point	System limitation	multiple
9	Exaggerated input, > highest set point	System limitation	single
10	Exaggerated input, > highest set point	System limitation	multiple
12	Unreachable system	Error	single
13	Unreachable system	Error	multiple

B. Case study: Simulation of residential building heating

Based on the previous definitions, a space heating case study is performed, where multiple users interact with a chatbot which controls a simulated boiler for indoor heating. The simulation implements the basic physical properties of a building and its heating system, where the heat is regulated by a PID controller, which in turn is controlled by the agent. The agent is implemented in LangChain, using GPT-4 as the LLM, and custom built tools to interact with a database with user information, RAG to access a vectorbase of Q&As related to the heater, and another set of tools to interact with the controller via JSON requests (REST API) to set or get system parameters.

1) Test protocol with users: Table I shows the test protocol for 15 different tests evaluating normality, disturbance, system limitations, and system error. The tests use a collection of three personas with varying age, preferred temperature, and a weekday and weekend schedule, visible in Table II.

The complete runs are available in the supplementary materials.

III. Results

The results of the study are split into two parts:

1) a software implementation given the schema in Figure 1,

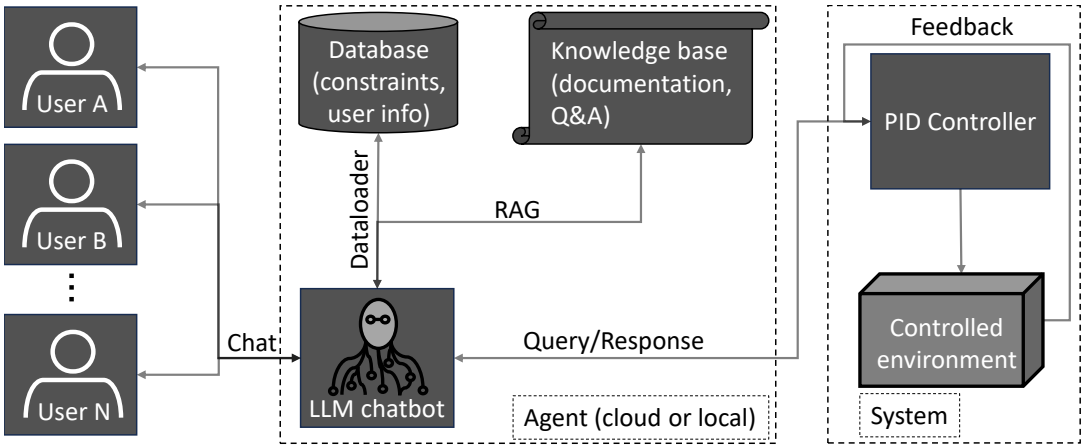

Figure 1. Conceptual block diagram describing the information flow of the Humans-in-the-Loop CPS with an agent powered by an LLM chatbot.

2) a simulated case study where the series of tests summarised in Table I are presented.

A. Proposed architecture

The conceptual schema from Figure 1 led to an implementation following the Unified Modeling Language (UML) component schema from Figure 2.

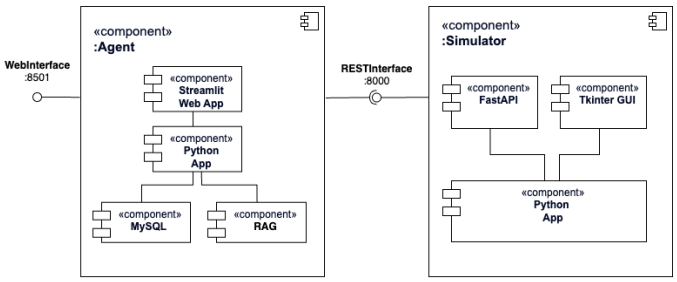

Figure 2. UML Component Schema of the control system using LLMs.

This system architecture enables all the interactions required to facilitate the different flows of information from the conceptual schema. The REST interface enables two-way communication between the Simulator and the agent via HTTP. The user then gets all the necessary feedback through the web interface; a screenshot of the latter is visible in Figure 3. Furthermore, the simulator values can be followed with a desktop Graphical User Interface (GUI) implemented using Tkinter presented in Figure 3. The code for the agent and the simulation is available on GitHub [24].

B. Case study: Simulation

The following sections present a summary of each type of test case visible in Table I using personas from Table II.

TABLE II. SHORT DEFINITION OF THE PERSONAS USED IN THE TEST CASES.

Name	Age	Temperature	Weekday	Weekend
John Smith	31	20	07:00 - 17:00	09:00 - 23:00
Ronnie Coleman	58	22	06:30 - 18:00	08:30 - 22:30
Robinson Crusoe	25	21	09:00 - 19:00	10:00 - 23:00

1) Normal Operations: For both single and multi-user contexts, normal operations of the heater did work successfully given test #1 and #2. However, in insistent scenarios #3 and #4, the temperature was set without taking into account other users, both for the single and multi-user context. Furthermore, the temperature set point has reached values that are much too cold for a building. In tests #14 and #15, the agent has been reluctant to provide user information to the current user if the query did not include the name of the user; otherwise, preferences were gathered successfully. In some cases, unplanned behaviour did happen such as confirming the change of temperature set points without the latter being effectively updated. Also, when asking for user preferences in a query, the next query simulating a user login did induce an answer including the user preferences of this user.

2) Disturbance: When dealing with erratic cases in tests #5, #6 and #11, the agent behaved successfully in both single- and multi-user scenarios. When faced with disturbances or unclear queries, the agent did not stop the flow of conversation but instead asked for more precision; even when facing chaotic situation with multiple users. In multi-user context with only one erratic user, other users could continue having normal interactions with the agent.

3) System limitations: Represented by tests #7, #8, #9 and #10, exaggerated queries were formulated above and below thresholds from normal set points from a house temperature, within single- and multi-user contexts. When requests went above the system threshold, the agent asked the user for another value within the system limits, thus complying with the expected behaviour to refuse to set such set points. In other cases where the user asked for temperatures in the range of the system limitations but with unrealistic temperature for a house, the agent set these temperatures without any warnings. When faced with an unrealistic set point, other users could update the value to any other set point.

4) Error: In scenarios simulating an error such as the unavailability of the system, the agent provided simple answers to communicate the issue to the user. This happened in both single- and multi-user tests.

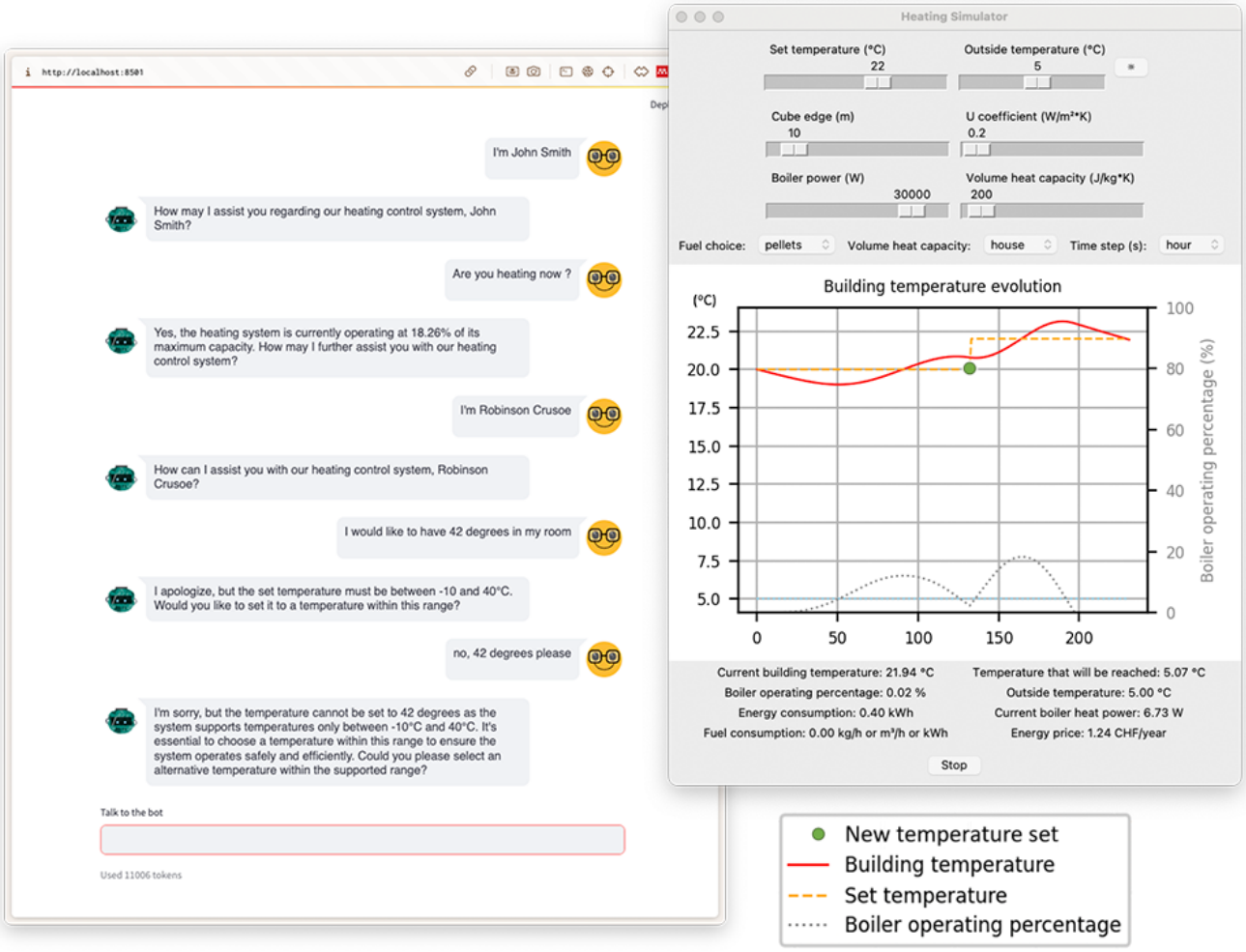

Figure 3. Interfaces of the chatbot on the left and the simulator on the right.

IV. Discussion

Structured around three research questions, this study introduces the use of LLM-based chatbot agents to act as user interfaces and mediators in control systems. To answer the research questions cited in the introduction, a method with an implemented software and two use cases yielded qualitative results, given multiple test scenarios. Following, the three answers to the research questions will be detailed.

A. What architectures can be used to leverage LLMs for improving user interfaces within control systems?

Presented in Figure 2, the proposed architecture composed of an agent and a simulator remains straightforward, thus making it flexible for all research purposes. This architecture has been implemented and allowed the suite of test cases in Table I to be run with various scenarios. For example, in test case #14, the agent executed queries in the database, successfully used RAG, and set a temperature set point in the simulator. Therefore, with this diversity of queries within a single chat conversation, the first research question is answered. Nevertheless, a successful control system needs more than an architecture, which will be discussed in the next sections.

B. How can LLMs incorporate user preferences and circumstances into a control system?

To answer the second research question, the case study simulation and the mediator case study have been performed. For all test cases in Table I, the results show that given a decision to make, the agent is only able to gather part of the information it needs, despite being informed of its available functions. Nevertheless, when a query is precise enough, the agent can successfully gather one type of external information, synthesise its result in natural language, and deliver the answer to the user, while calling the appropriate function to, e.g., set the simulator to a desired state.

Therefore, this behaviour already enables simple queries to be executed as seen in the successful test cases, but has limitations. This implies that a better reasoning component might facilitate more complex queries and context to be answered. For instance, at each query, more additional context could be injected directly in the prompt, or RAG could be involved when large documents are required.

In the test cases in Section II-B, only the last two messages were included to generate an answer. Despite providing enough

information to answer the request, this drastically limited the reasoning capabilities of the LLM in its understanding of multi-user interactions and evolution of one or many personas needs.

Given these tests, the second research question can only be partly answered; user preferences and circumstances were incorporated into the agent, but only for simple queries. Since multi-user contexts already involve more complex queries, such a scenario has been tested and is discussed in the next section.

A potential next step in user assistance is to incorporate external information, such as energy price and environmental impact, by adding tools for real-time data querying. This would enable users to take such information into account when making decisions, and might also enable direct optimisation based on such parameters to facilitate cheaper and more sustainable operations.

C. How can LLMs mediate between users with varying preferences or constraints?

The results of the case study clearly show the limitations of a chatbot that is not set up properly to mediate between conflicting interests; all tests with the custom agent show how it always conforms to the latest request, unless it is completely inappropriate, such as 42°C, as in test 10 in the supplements. This is likely due to the LangChain agent implementation being built on very early LangChain functions, and we believe that by updating the agent framework to the modern chat chains this will drastically improve. Hence, the next iteration of the custom agent should incorporate tools to facilitate chat history between users and carefully engineered prompts to prevent an overly accommodating controller.

This could be achieved with a multi-agent system [25][26], where each user has access to their own personal agent, while a controller/mediator agent interacts with the personal agents and the system. The politeness of LLM agents can thus be leveraged to create an LLM "filter" between the user and the controller, hopefully preventing abuse by overly persistent and aggressive users. User privacy concerns can also be addressed by having chat histories and complete user preferences stored so that only the personal agent can access them, while the controller agent receives only relevant information from the personal agent when necessary.

D. Ethical considerations

The ethical concerns present in this line of work are related to privacy concerns and value-based judgements.

Privacy concerns are a risk due to the sharing of the agent, and the storing of personal data where the shared agent can access them. With good data practice and an agent accurately prompted to access only the data pertaining to the current user, this risk can be mitigated, though this is not something we specifically investigated. Alternatively, a multi-agent framework as described in Section IV-C can be used to keep data storage unique to each personal agent, and thus each user.

Value-based judgements are a consideration in any work with conflict resolution and mediation aspects; who decides what is right when two parts have conflicting interests that cannot be jointly satisfied? For instance, if one user is freezing and another is sweating indoors, is it worse to sweat or to freeze, and is it easier to put on a jacket than to dress cooler? Furthermore, is it more ethical to have a cool indoor environment for the sake of reducing emissions, or is it more ethical to have an indoor environment that most users favour? These questions are difficult to answer, and not all humans will agree.

Thus, when using an LLM as a mediator, it is important to consider who decides GPT-4's values? A causal language model, such as GPT-4, trained with self-supervised learning on vast amounts of text will learn to mirror the values it has seen during training. Supervised learning is then used to align the model with our desired outputs, and finally reinforcement learning from human feedback is used to optimise the chatbot output based on training participant evaluation. Consequently, there are three levels of learnt values: the hidden beast of self-supervised learning, the indirect adjustment of supervised fine-tuning, and the conforming to desires of RLHF participants.

The question of which values to enshrine is particularly interesting during RLHF, where the selection of participants will implicitly alter the "ethics" of the LLM. Hence, the ideal solutions for custom controller agents would be to set up pre-trained LLMs locally, fine-tune them on custom tasks, then define RLHF rewards so that the agents can be further optimised while running. This would also ensure maximum privacy, as everything can run locally within protected networks. However, it is also the most expensive solution in both energy and development, so different scenarios will likely require different levels of local versus cloud-based approaches.

E. Future work

In order to improve the usage of LLMs for control systems, the following improvements are proposed.

First, the ability to understand context coming from multiple sources. This could be enabled with the implementation of an algorithm such as Recursion of Thoughts, an improved version of Chain of Thoughts [27], [28]. This added step would allow the agent to progressively build an answer given provided sources of information, thus creating a bigger context.

Second, an improved multi-user ability. So far the aforementioned feature has been simulated, and to enable it would require a dedicated architecture able to alleviate privacy issues. A suggested possibility would be to divide the agent in two parts: a user-agent that is personal and a system-agent that is part of the control system. This division would increase privacy since the user-agent would be isolated from the other agents, which would naturally prevent private information leakage between agents. Also, with this improvement, the system-agent could have the possibility to request information from all user-agents to build its context before taking a decision. In the case of this study, the system-agent would act as a mediator taking all user preferences and inputs to propose a new set temperature.

Third, a scheduler to take into account time-related inputs. Since controlling a system can include timely inputs, the system-agent should have some scheduling capabilities. For instance,

this could be used to adapt set points to user preferences and inputs dynamically, or take the weather into account.

V. Conclusions

This study examined the integration of LLMs into control systems, specifically targeting shared appliances within a CPS context. The exploration centred around a framework that facilitated the use of LLMs to manage the heating system of a shared building, addressing user preferences, and mediating conflicts. Key findings underscore the LLM's capability to function beyond traditional applications, serving as a robust mediator and decision-maker within a control system. However, challenges such as handling erratic input and dependency on user interaction precision were identified, suggesting areas for future improvement. In addition, ethical considerations highlighted the need for user data management and the importance of aligning the LLM's operational parameters with ethical standards, ensuring that privacy and fairness are maintained. Further research should expand the application of LLMs within CPS by exploring multi-agent and time-dependent frameworks, to further explore the potential aspects of human-machine interactions in shared environments.

Acknowledgements

The authors would like to thank Philippe Marziale for his remarkable contribution to the code base, as well as Sébastien Rumley and the company yord sàrl (yord.ch) for their guidance during this project.

References

[1] T. Brown *et al.*, "Language models are few-shot learners", *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[2] H. Touvron *et al.*, "Llama 2: Open foundation and fine-tuned chat models", *arXiv preprint arXiv:2307.09288*, 2023.

[3] S. Bubeck *et al.*, "Sparks of artificial general intelligence: Early experiments with gpt-4", *arXiv preprint arXiv:2303.12712*, 2023.

[4] M. G. Madden, B. A. McNicholas, and J. G. Laffey, "Assessing the usefulness of a large language model to query and summarize unstructured medical notes in intensive care", *Intensive care medicine*, vol. 49, no. 8, pp. 1018–1020, 2023.

[5] P. U. Rodriguez, A. Jafari, and C. M. Ormerod, "Language models and automated essay scoring", *arXiv preprint arXiv:1909.09482*, 2019.

[6] M. Chen *et al.*, "Evaluating large language models trained on code", *arXiv preprint arXiv:2107.03374*, 2021.

[7] D. M. Ziegler *et al.*, "Fine-tuning language models from human preferences", *arXiv preprint arXiv:1909.08593*, 2019.

[8] E. Kasneci *et al.*, "Chatgpt for good? on opportunities and challenges of large language models for education", *Learning and individual differences*, vol. 103, p. 102 274, 2023.

[9] T. Dave, S. A. Athaluri, and S. Singh, "Chatgpt in medicine: An overview of its applications, advantages, limitations, future prospects, and ethical considerations", *Frontiers in artificial intelligence*, vol. 6, p. 1 169 595, 2023.

[10] M. Ajevski, K. Barker, A. Gilbert, L. Hardie, and F. Ryan, "Chatgpt and the future of legal education and practice", *The Law Teacher*, vol. 57, no. 3, pp. 352–364, 2023.

[11] H. Chase, "Langchain", 2022, https://github.com/hwchase17/langchain [Accessed: 2024-06-13].

[12] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks", *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[13] paulalesius, *Large language math - the mathematics of llm foundational models - for beginners*, https://llmath.unnservice.com/ [Accessed: 2024-06-13], 2022.

[14] O. Topsakal and T. C. Akinci, "Creating large language model applications utilizing langchain: A primer on developing llm apps fast", in *International Conference on Applied Engineering and Natural Sciences*, vol. 1, 2023, pp. 1050–1056.

[15] R. Asyrofi, M. R. Dewi, M. I. Lutfhi, and P. Wibowo, "Systematic literature review langchain proposed", in *2023 International Electronics Symposium (IES)*, IEEE, 2023, pp. 533–537.

[16] K. Pandya and M. Holia, "Automating customer service using langchain: Building custom open-source gpt chatbot for organizations", *arXiv preprint arXiv:2310.05421*, 2023.

[17] A. Singh, A. Ehtesham, S. Mahmud, and J.-H. Kim, "Revolutionizing mental health care through langchain: A journey with a large language model", in *2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, 2024, pp. 0073–0078.

[18] A. Pesaru, T. S. Gill, and A. R. Tangella, "Ai assistant for document management using lang chain and pinecone", *International Research Journal of Modernization in Engineering Technology and Science*, 2023.

[19] F. Dou *et al.*, "Towards artificial general intelligence (agi) in the internet of things (iot): Opportunities and challenges", *arXiv preprint arXiv:2309.07438*, 2023.

[20] M. U. Hadi *et al.*, "A survey on large language models: Applications, challenges, limitations, and practical usage", *Authorea Preprints*, 2023.

[21] T. Samad, "Human-in-the-loop control: Applications and categorization", *IFAC-PapersOnLine*, vol. 53, no. 5, pp. 311–317, 2020, 3rd IFAC Workshop on Cyber-Physical & Human Systems CPHS 2020, ISSN: 2405-8963.

[22] Y. Fang, Y. Lim, S. E. Ooi, C. Zhou, and Y. Tan, "Study of human thermal comfort for cyber–physical human centric system in smart homes", *Sensors*, vol. 20, no. 2, p. 372, 2020.

[23] M. A. Torad, B. Bouallegue, and A. M. Ahmed, "A voice controlled smart home automation system using artificial intelligent and internet of things", *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 20, no. 4, pp. 808–816, 2022.

[24] *GitHub code repository llm for cyber-physical systems featuring an agent controller*, https://github.com/fredmontet/llm-cps-ac, Accessed: 2024-09-25.

[25] J. Qin, Q. Ma, Y. Shi, and L. Wang, "Recent advances in consensus of multi-agent systems: A brief survey", *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 4972–4983, 2017.

[26] A. Amato, A. Quarto, and V. Di Lecce, "An application of cyber-physical system and multi-agent technology to demand-side management systems", *Pattern Recognition Letters*, vol. 141, pp. 23–31, 2021, ISSN: 0167-8655.

[27] S. Lee and G. Kim, "Recursion of thought: A divide-and-conquer approach to multi-context reasoning with language models", *arXiv preprint arXiv:2306.06891*, 2023.

[28] M. Nye *et al.*, *Show your work: Scratchpads for intermediate computation with language models*, 2021. arXiv: 2112.00114 [cs.LG].