

Solution for Assignment 1

Nana Wang

August 30, 2023

Introduction of my research

My research is about simultaneous topology identification and control for heterogeneous multi-agent systems. Recent advances in communication and computation have motivated a progressive transition to the handling of larger-scale problems in the identification and control fields. The combination between network identification and control has received increasing interest due to its application to multi-robot systems intelligent transport, as well as social and biological networks. Topology identification aims to find the communication graph for the large-scale systems where the connection graph are unknown or partly unknown. Take the human immune cell system as an example, a large number of different cell populations coexist within a largely unknown interaction network. one goal of my research is the identification of the network structure through recurring input and output data of the cell concentrations and then deciding how and where to stimulate the network to derive an appropriate response. As for multi-robot and transport systems, network identification could find faulty links and direct the appropriate modification of the control strategy. My research focuses on developing a novel distributed method for simultaneous network identification and control of multi-agent systems. I want to identify the topology of multi-agent systems based on system identification and graph theory. Then optimize the network based on the previously identified topology and design distributed control algorithms of multi-agent systems. Potentially, my research can be applied to identify the communication topology between cells of immune systems.

Discuss about Robert's lectures

I want to discuss what I have learned from Robert's lectures and discuss how they relate to my research and topic area. The first idea is the main difference between computer science and software engineering. Software engineering is more about how to specify requirements, implementation and tests, while computer science focus on theory part and tend to solve some abstract theoretical problems. Software engineering involves planning, requirement clarification and test models to aims to design a software to address the desired practical objectives. The systematical design from software engineering help me understand my research topic deeper. First I need to clarify my research problems, then list the requirements for algorithms of topology identification and control, design the algorithms, and consider how to test them. And the whole process will iterate during the research and it is quite normal for a better product. Another inspiration for me is to think about the practical issues of application in simultaneous topology identification and control for multi-agent systems. Hence, some important problems

in my topic are how to combine these two topics appropriately and how to scale the algorithms for topology identification and control. If the network is quite large, the computation for identifying the topology is rather high and it is also a challenge to combine the control with topology identification. As the achievement of control tasks rely on the network topology which may needs more time to calculate. So distributed algorithm for topology identification can be new direction. Except these, when specifying the practical issues, the next step is to design algorithms to address these issues and test these algorithms' performance for verification. Applying these principles in my research allows me to address my research problems systematically. Besides, human factors also have more effect than I expected. My research topic involves the application of biological systems. However, my background is in the control part and my research focuses on algorithms of topology identification and control, but I also want to contribute to how to identify the communication topology of cells in immune systems. I collaborate with the researchers from KI and they can do experiments to generate the data for the identification of communication topology. However, how to communicate with people from different backgrounds, especially from different subjects, is a challenge for both of us. Therefore, how to ensure there is no mistakes or misunderstanding between us is very important for our collaboration. First, realize the importance of communication during collaboration and remove the potential bias between us, may by communicating with them frequently and asking detailed questions.

Discuss about Guests' lectures

From the guest lectures, I learned lots of stuff and I want to discuss more about how to test the software for quality assurance and what the productivity is for a software developer. The boundary value test is one way to test the software's performance and reveals discrepancies between desired and actual behaviour. For my research, I can also consider the boundary value test to test the topology identification algorithms in practical scenarios to see if these algorithms work well. When using ML methods to identify the models, I need to consider how to test the performance of the models in practical scenarios and explore the boundary values to test the performance of the algorithms. Moreover, for the control issues in multi-agent systems, how to design a robust controller in the worst case is a similar idea to the boundary value test. For the second concept, productivity is not just one dimension that measures how many lines the code is or how much time is spent to address a fixed problem but also includes the other four dimensions in the space model. Specifically, SPACE models to describe productivity involves five parts: satisfaction and well-being, activity, performance, activity, communication and collaboration, and efficiency and flow. When I want to evaluate my research work systematically and fairly, I can borrow this model for further evaluation and use this model to guide my research plan later. The communication is very important for my work to improve my research effect. It's better to build personal website and spread our research work via researchgate, linkedin. Besides, the presentation is as important as solving research problems. Hence, I should emphasize the presentation and train my presentation skills for better research.

Discuss about quality assurance

I want to discuss more about quality assurance. Quality assurance is an important part of software engineering and it means monitoring the software process and can assure the quality of the designed software. It can provide us with much more reliability of the software with quality assurance. Normally, quality management involves many topics, such as specifying quality requirements, assessing quality and quality assurance. There are a couple of challenges to quality assurance in software engineering, including solving its drawbacks and potential applications in AI-based software[1]. One is that the quality requirements are changing and unclear, making quality assessment quite difficult. A clearer quality requirement has more potential to assess the quality properly. More and more software systems become increasingly complex, like AI-based systems, which compose of lots of components. To ensure the quality of such complex systems, we need to ensure the quality of individual components and the integration of these components, which is challenging and time-consuming. Different quality attributes of complex systems are defined and there are no standard quality attributes without contexts. Quality assurance needs to be defined and tested specifically. The AI-based system, as special complex software, needs different quality attributes from that of traditional software and new quality assurance methods are needed[1]. AI-based software focuses on safety, functional correctness, robustness, reliability, security, accuracy, explainability and testability [2]. As for its connection to simultaneous topology identification and control for multi-agent systems, the requirement should be defined clearly and specifically, like the accuracy, the computation time and complexity, scalability and so on. Based on that, the assessment of the quality of the algorithms will become easy to implement. Other than that, it is also important to assess the integration of algorithms for complex systems. In different scenarios, with access to different information, we should propose different algorithms that can address the topology identification problem and the control problem for multi-agent systems.

Discuss about Software requirement

Software requirement, from my understanding is to collect, analyze, specify and validate software requirements to guarantee the correct understanding of what needs to be developed. It is the first and key step in designing software. Since the software is designed to meet the needs that need to be correctly understood. Otherwise, it can produce undesired software. Some challenges in requirement engineering need to be addressed. One problem is how to eliminate the ambiguity of software requirements [3], resulting in a waste of time and money. Hence, in practice, the elimination needs to consider how to detect, mark and handle ambiguity in requirement engineering. Besides this challenge, there are other barriers faced by the requirement, like lack of knowledge management at distributed sites, lack of standard and procedure of requirement engineering, language barrier among geographically distributed teams, and cultural difference [4]. From the analysis of the barriers in requirement engineering, we can see some commercial opportunities. Standards and procedures of how to specify, analyze and test requirements should be set up to unify different standards and procedures. As for language barriers and cultural differences, more communication is needed to avoid potential misunderstandings. AI/ML has developed so fast

and attracted lots of attention. As a complex software system, requirement engineering can be used in AI/ML-based software. However, there are lots of different focuses on requirement engineering for AI/ML-based software, so it cannot apply requirement-engineering methods to AI/ML-based software. There are many challenges in how to analyze, specify and test requirements in AI/ML-based software. How to improve the specification of requirements for explainable or safe AI-based systems. The AI-based system has high adaptability and learning ability, defining requirements that can evolve with time and data is quite challenging. It should have some mechanisms that can handle concept drift or changing user preferences. After specifying the requirement, new techniques for validation and verification are needed for AI-based systems because AI-based systems are complex and have non-deterministic behaviour. Besides this, AI-based systems are increasingly collaborating with humans, so the requirement should address the interfaces, feedback mechanisms and user experience considerations to ensure the smooth collaboration between humans and AI. As for the relation to my topic, requirements should be specified when solving topology identification problems, like how many agents the multi-agent system involves and how to test the feasibility and performance of algorithms. As for the interaction with humans, how to build interfaces and feedback loops between algorithms and multi-agent systems are also needed to consider for further application.

Future trends of SE for ML

As for the future trends of modelling and control for multi-agents with software engineering, the monitoring and analysis for the multi-agent system should be automated. When the task specification is clear, the modelling and control algorithms can accomplish these tasks fast and well. At the same time, concerning the larger networks, the algorithms for controlling can be distributed to implement the coordination tasks. For the modelling, the data collection and management should be distributed in large networks. Except this, in complex environment how to describe the control tasks and design the controller to ensure the task could be achieved automate is also one challenge to solve based on AI. We can specify the tasks based on AI via formal methods and design a controller to achieve the tasks, which can combine task-planning and control together.

ML/AI Engineering will change lots of fields and have more applications in the coming 10 years. To make AI/ML applicable in reality, there are lots of challenges, like protecting the privacy of data, how much we can rely on the results generated by AI/ML tools, how to interpret the model, and how much we can rely on the results of the model. AI-based systems have security, usability and privacy concerns and the analysis methods are needed to design and check for these attributes. Based on SE, explainability, data centricity, verifiability and change propagation are also critical issues in designing the structure and behavior of AI-enabled systems [5]. We also need to solve how to reproduce the model and test the performance of the models.

In terms of ML/AI Engineering, data play an central role in AI systems, including data collection, cleansing, management and continuous updates. Therefore, resource and execution planning need to be considered for the pre-processing of data, the training and testing. It is not just the responsibility

of data scientist, but also needs the coordination between data scientist and software engineers. Data privacy is also a challenge for the application for many applications, considering ethical and legal concerns. Addressing these challenges requires more research and work in different domains.

About the future opportunities in academic of SE for multi-agent systems, decentralization and autonomy will be the research trend, allowing agents to make more independent decisions and collaborate effectively without centralized control. As the application of multi-agent systems will become more complex and involves a high number of agents, some other future trends will be how to increase the scalability of algorithms, such as efficient communication protocols, adaptive resource allocation. How to integrating ML/AI into multi-agent system will improve the self-learning and adaptivity of each agent. Because multi-agent systems have various real-world applications, like smart cities, autonomous vehicles, biological systems, software engineers can work on innovative solutions for these real-world challenges. For the career opportunities, there are some engineers working on designing, developing algorithms for multi-agent systems in applications, like interaction protocols and system architecture, control algorithms, planning algorithms.

References

- [1] B. Gezici and A. K. Tarhan, “Systematic literature review on software quality for ai-based software,” *Empirical Software Engineering*, vol. 27, no. 3, p. 66, 2022.
- [2] S. Martinez-Fernández, J. Bogner, X. Franch, *et al.*, “Software engineering for ai-based systems: A survey,” *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 31, no. 2, pp. 1–59, 2022.
- [3] V. Gervasi, A. Ferrari, D. Zowghi, and P. Spoletini, “Ambiguity in requirements engineering: Towards a unifying framework,” *From Software Engineering to Formal Methods and Tools, and Back: Essays Dedicated to Stefania Gnesi on the Occasion of Her 65th Birthday*, pp. 191–210, 2019.
- [4] M. A. Akbar, A. Alsanad, S. Mahmood, A. A. Alsanad, and A. Gumaei, “A systematic study to improve the requirements engineering process in the domain of global software development,” *IEEE Access*, vol. 8, pp. 53 374–53 393, 2020.
- [5] I. Ozkaya, “What is really different in engineering ai-enabled systems?” *IEEE software*, vol. 37, no. 4, pp. 3–6, 2020.