WASP Software Engineering Assignment

Gustav Zetterqvist

August 25, 2025

1 Introduction

My research is focused on localizing sound sources, with some applications in wildlife monitoring. Specifically, I am developing new methods for direction of arrival (DOA) estimation using microphone arrays, in order to decrease the size of the array. Traditional DOA estimation methods often rely on phase differences between microphones, which can be sensitive to noise and require larger arrays for accurate estimation. My approach uses received power measurements instead, which allows for smaller arrays while still achieving reliable DOA estimation.

I'm focusing on traditional signal processing methods, rather than machine learning (ML) or artificial intelligence (AI) techniques, as I believe that these methods are more explainable and easier to deploy in real-world applications. My work is primarily implemented in MATLAB.

2 Lecture principles

2.1 Verification and Validation

One core principle from Robert's lectures is the verification and validation of systems. This principle is crucial for my research, as it ensures that the DOA estimation methods I develop are reliable and accurate. I've collected data from a microphone array in a controlled environment, which allows me to test and validate my algorithms. Verification involves checking that the algorithms are implemented correctly - "Checking whether we are building the **product right**", while validation ensures that they perform well under various conditions — "Checking whether we are building the **right product**".

As of today, I mostly do dynamic testing, where I run my algorithms on the collected data and compare the results to known ground truth values. However, the concept of formal verification, such as model checking or theorem proving, is something I would like to explore further.

2.2 Behavioral Software Engineering

Another important principle from Robert's lectures is the concept of behavioral software engineering. This approach emphasizes the need to understand and model the behavior of software systems, particularly in the context of user interactions and real-world usage scenarios. For my research, this principle is relevant because it encourages me to consider how the DOA estimation methods I develop will be used in practice, and how they can be designed to adapt to different environmental conditions and user needs.

By incorporating behavioral modeling into my research, I can better anticipate potential challenges and limitations of my algorithms, leading to more robust and user-friendly solutions. This might involve simulating various field conditions, user interactions, and system responses to ensure that the final implementation is both effective and practical for wildlife monitoring applications.

3 Guest-Lecture Principles

3.1 Requirements Engineering

I found Julian's guest lecture on requirements engineering particularly insightful. I really liked the emphasis on the importance of understanding user needs and expectations, i.e., requirements, before starting the development of a system.

> "Understand the **problem** before you build the **solution**."
>
> - Julian

This was an important reminder for me, as I often focus on the technical aspects of my research without fully considering the end-users and their needs. However, I think that this principle is more relevant for higher TRL (Technology Readiness Level) projects, where user requirements are better understood and can be more easily integrated into the development process.

3.2 Human Aspects of Software Engineering

In the guest lecture by Per, he emphasized the importance of understanding the human aspects of software engineering, such as team dynamics, communication, and collaboration. Humans don't always behave rationally, and this can lead to challenges in software development. The application he worked on, air traffic control, is a great example of a system where human factors play a crucial role in the design and implementation of software, and the consequences of errors can be severe. Also here, I think that this principle is more relevant for higher TRL projects or products in industry.

4 Data Scientists versus Software Engineers

The chapters from the CMU book on "Machine Learning in Production" [1] were interesting to read, and it had some good points about how to go from models to systems. I **mostly agree** with the presented differences between data scientists and software engineers in the book. In short, my view is that data scientists focus on developing and validating models, while software engineers are responsible for integrating these models into production systems. This distinction is important because it highlights the different skill sets and mindsets required for each role. Data scientists often work with statistical methods and data analysis, while software engineers focus on software architecture, design patterns, and system performance. However, **I believe that there is a growing need for collaboration** between these two roles, as the boundaries between them become increasingly blurred.

I think that both data scientists and software engineers will need to learn some skills from each other, as the field of AI and machine learning evolves. For example, data scientists should have a better understanding of software engineering principles, such as version control, testing, and deployment, to ensure that their models can be effectively integrated into production systems. On the other hand, software engineers should become more familiar with machine learning concepts and techniques, so they can better understand the limitations and capabilities of the models they are working with – the **T-shaped team member** concept. I believe that there is also a need for more interdisciplinary roles that combine the skills of both data scientists and software engineers, such as machine learning engineers or AI developers. These roles can help bridge the gap between the two fields and ensure that AI systems are developed in a way that is both technically sound and aligned with user needs.

5 Paper analysis

5.1 Paper 1: "From Hazard Identification to Control Design: Proactive and AI-Supported Safety Engineering for ML-powered Systems"

For the first paper [2], the core idea is to do hazard analysis to proactively identify and mitigate potential risks in ML-powered systems before they manifest as failures. They also propose the use of LLMs to support the hazard analysis process by providing insights and recommendations based on historical data and patterns. Their approach builds on the System-Theoretic Process Analysis (STPA) framework, due to its ability to handle complex system and since it can be used in the early stages of system design.

In the paper they emphasize the social risks associated with ML systems, such as bias and insufficient oversight, and advocate for a more comprehensive approach to safety engineering that includes these considerations.

The findings include that LLMs can be helpful to suggest potential hazards and recommend mitigation strategies based on historical data and patterns, also it's effective in merging similar hazards into broader categories for more efficient analysis. However, the conclusion is that while LLMs can assist in hazard analysis, they are not a replacement for human expertise and oversight in safety engineering.

In relation to my own research, this paper highlights the importance of considering safety and risk management in the development of AI/ML systems, particularly in the context of control design. While my research focuses on DOA estimation using microphone arrays, the principles of proactive hazard analysis and risk mitigation can be applied to ensure that the algorithms I develop are robust and reliable in real-world applications. Also, the idea of using LLMs to support hazard analysis is interesting, as it could potentially help with ideas and recommendations for improving the reliability of my algorithms.

For a larger AI-intensive project, such as the development of an autonomous vehicle system, the ideas presented in this paper could be applied to ensure that safety is prioritized throughout the system's lifecycle. This could involve using AI techniques to continuously monitor the system's performance and adaptively adjust control strategies in response to identified hazards.

The key take away that I can apply to my own research is the idea of using LLMs to support the development of my research on DOA estimation. By getting insights and recommendations from LLMs, I could potentially

improve the reliability and robustness of my algorithms.

5.2 Paper 2: "(Why) Is My Prompt Getting Worse? Rethinking Regression Testing for Evolving LLM APIs"

The core idea in the second paper [3] is to address the challenges of regression testing in the context of updates of large language model (LLM) APIs. As LLMs are updated and "improved", it's shown that a previously carefully engineered prompt may perform worse than before, leading to unexpected behavior and degraded performance. The authors provide a case study, with 4 different prompting strategies, and 5 different LLM models.

The findings indicate that prompt degradation is a significant issue, and often LLM APIs are updated without the user's knowledge, leading to unexpected changes in behavior. When this happens, the effectiveness of existing prompts can be compromised, and another round of prompt engineering may be necessary to restore desired performance.

As I don't use AI or ML in my research, I don't have a direct relation to this paper. However, the principles of regression testing and the challenges of maintaining performance in evolving systems are relevant to any software engineering project, including my own work on DOA estimation.

This issue is also relevant in larger AI-intensive projects, such as the development of AI-powered chatbots or virtual assistants. In these projects, regression testing is crucial to ensure that updates to the underlying LLMs do not negatively impact the user experience or the system's functionality. The paper's proposed strategies for regression testing could be applied to these projects to maintain performance and reliability.

Once again, the field of AI and ML is not directly related to my research, but the challenges of maintaining user experience and system performance in the face of evolving technologies are universal concerns in software engineering.

While my methods for DOA estimation are based on traditional signal processing techniques, I could explore incorporating machine learning approaches to enhance performance. This could involve using LLMs to analyze and optimize the algorithms used for DOA estimation.

6 Research Ethics & Synthesis Reflection

6.1 Search and Screening Process

I started my search for relevant papers by using the CAIN conference website, which lists all the papers presented at the conference. I browsed through the titles and abstracts of the papers to identify those that were most relevant to my research area, searching for keywords related to control systems, signal processing and sensor fusion.

While browsing the papers, I took note of any that seemed particularly relevant or interesting, and I saved their citations for later reference. I also looked for papers that cited or were cited by the ones I found, as this can often lead to discovering additional relevant work.

I found one paper [2] that was more related to my research area, focusing on safety engineering for ML-powered systems, specifically for control design. For the second paper I chose [3], since the title sounded interesting, even though it was less directly related to my work.

6.2 Pitfalls and Mitigations

During the search process, I encountered some challenges. The primary challenge was the initial lack of effective results when searching for papers. In response, I refined my search terms and expanded my criteria to include related topics, which helped me discover additional relevant work.

I did not encounter any misleading titles or abstracts that significantly impacted my search.

6.3 Ethical Considerations

In the papers I selected, I did not see any sign of unethical research practices, such as data fabrication or plagiarism. Both papers used LLMs in a responsible way, and they acknowledged the use of these models in their research.

References

[1] C. Kästner, *Machine Learning in Production*. Carnegie Mellon University, 2024.

[2] Y. Hong, C. Timperley, and C. Kästner, "From hazard identification to control design: Proactive and AI-supported safety engineering for ML-powered systems," in *Proceedings of the IEEE/ACM 4th International Conference on AI Engineering - Software Engineering for AI*, CAIN 2025, ACM, Apr. 2025.

[3] W. Ma, C. Yang, and C. Kästner, "(Why) is my prompt getting worse? Rethinking regression testing for evolving LLM APIs," in *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI*, CAIN 2024, pp. 166–171, ACM, Apr. 2024.