Software Engineering Assignment

Gustav Vallinder

31/08/2023

1 AllDrive

The current generation of software stacks for autonomous driving are typically based on the assumption that road conditions are sufficiently good and that the level of road friction does not need to be taken into account. In order to allow autonomous driving in climates where this assumption doesn't hold, algorithms need to be developed that sense the road friction and adapt the decision making, planning and control of the autonomous vehicle. AllDrive is a joint project between Luleå University of Technology and KTH on the academic side and Klimator and Scania on the industry side, with the goal of enabling autonomous driving of semi-trailer trucks on low-friction roads. The Vinnova FFI-funded project is split into three separate PhD-projects.

- Friction estimation and sensor fusion
- Adaptive decision-making and motion planning
- Friction-aware motion control for semi-trailer trucks

The third project, "Friction-aware motion control for semi-trailer trucks", is my research topic. The research explores vehicle- and tire models with a focus on friction, and incorporates the models in methods for motion control. An example of such methods is Model Predictive Control, which is commonly used in motion control applications.

2 Concepts related to my area

I think the most obvious concept that directly relates to my area of research is requirements engineering. The driving motivation for the project is to improve "Safety and Performance" of autonomous semi-trailer trucks, which by itself doesn't mean anything unless "Safety" and "Performance" are broken down into measurable performance indicator. Important is also to be aware of the environment assumptions, which in the autonomous driving world is typically called its Operational Design Domain, which specifies these environmental assumptions for which the vehicle can properly operate. The design of a friction-aware driving system should the in every step be designed with these performance indicators and environmental assumptions in mind.

Another concept to discuss could be the disconnect between software prototyping for academic purposes versus the more process-oriented software development that otherwise is required for automotive software. As an industrial PhD it is of course part of the job to write software to test my developed algorithms. As an industrial PhD, the desired target to test this is on real Scania vehicles. There is a trade-off between fast enough development methods to be able to conduct experiments, and good enough software engineering processes to ensure safety and robustness of the algorithm implementations. Finding this balance is not a trivial task. Perhaps the pragmatic software engineering rules can be of use?

3 Concepts from the guest lecturers

On the SPACE model for developer productivity as mentioned in Linda Erlenhovs lecture: It seems like a great model for use in companies where the productivity of teams and team members are to be evaluated, but for a single PhD developer might not be what we are optimizing for. Perhaps it would rather be something like the rate at which we can implement and test software in combination with the ease of doing so. Things like satisfaction, well-being, collaboration and communication are still important but less so than in a business setting where there are long careers and customer needs in the equation.

The good news is that AI4SE and DevBots are especially well suited for the type of productivity I describe for "PhD developers". Tools like copilot or ChatGPT can absolutely be used to improve the ease of speed of development of research code.

4 Topics in SE

4.1 Software Architecture for Autonomous Driving

Software architecture in my understanding is the structure required to put different building blocks of software together to form a complete software system, in analogy to how building blocks can be put together to form a house.

Autonomous driving (AD) is a task of enormous complexity. The autonomous car must at all times know where it is and at the same time keep track of its surroundings and other road users (Perception). Once those tasks are solved, the autonomous car will also need to plan a global route and local path to follow, while selecting an appropriate behaviour (Planning). Finally, the car will need to be able to find control inputs that track the path and stabilizes the vehicle on the road (Control). All of these tasks need to be solved robustly, in real-time and on limited hardware. A requirement for making this possible is to make the best possible use of the existing tools withing software engineering. Some focus areas include but are not limited to:

- Testing
- Modularity
- Security

A lot of research effort goes into the specific driving tasks, and the focus areas mentioned are well understood within software engineering. However, not much literature goes into the specific structures used for autonomous driving. An example of an early L5 AD stacks is the Stanford University car Stanley, which participated in the 2005 DARPA Grand Challenge[1]. The authors provide some insight to the architecture and the individual modules, but today it can be viewed as slightly outdated since the progress in everything from hardware to software to algorithms has been immense since 2005.

More recent examples can be taken from the autonomous racing community, for example the 2018 ETH Zürich AMZ Driverless car, or the 2019 TUM RoboRace car [2][3]. These are more recent contributions, but are limited in the sense that they are designed for a narrow operational design domain, namely race tracks. A positive contribution is the Autoware project, which is open-source software stack for self-driving vehicles, built on ROS. The thoroughly documented project is the best attempt yet for a project that allows researchers, students and developers to get an understanding of the entire AD stack[4].

5 Future Outlook

I have some thoughts about the trajectory of everything

- ML/AI will continue to be important
- My impression is that ML/AI engineering methods and tools have improved and standardized these past few years and I believe it will continue
- Some of the AI hype will die and NVIDIA stock P/E-ratio will no longer be 200 in 5-10 years time
- Energy consumption will be a focus for both ML/AI engineering and cloud providers since energy will be more expensive

How it will affect my career in 5-10 years is more difficult to predict, but I would bet that I will have more insight into the possible impact of incorporating learning into the control software for semi-trailer trucks by then.

References

[1] Sebastian Thrun et al. “Stanley: The robot that won the DARPA Grand Challenge”. In: *Journal of field Robotics* 23.9 (2006), pp. 661–692.

[2] Juraj Kabzan et al. “AMZ driverless: The full autonomous racing system”. In: *Journal of Field Robotics* 37.7 (2020), pp. 1267–1294.

[3] Johannes Betz et al. “A software architecture for an autonomous racecar”. In: *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. IEEE. 2019, pp. 1–6.

[4] Shinpei Kato et al. “An open approach to autonomous vehicles”. In: *IEEE Micro* 35.6 (2015), pp. 60–68.