Towards Computational Resources-Aware Control Solutions for Automotive CPS

Jonathan Sprinkle and Ricardo G. Sanfelice

University of Arizona

Abstract

Automotive cyber-physical systems will need to address self-parking, advanced steering control, hazardous situation recovery, limited autonomous driving, and even more complex tasks in the coming decades. Verification of the safe behavior of these tasks for multiple vehicle configurations (weight, wheelbase, front/rear/all-wheel drive, etc.) will require significant advancements in the computational theory, as well as new approaches to compose behaviors and computational constraints with hybrid control theory and system modeling. This position paper addresses the hybrid nature of controllers for such systems, from the perspective of tradeoffs in computational burden versus desired robustness.

1 Introduction

Computational resources are a universal limitation of cyber-physical systems of today. This limitation imposes a hard constraint that is difficult to check at the design stage due to its complex dependence on implementation, hardware, and software. It compromises the satisfaction of design specifications that are crucial for self-driving vehicles to operate safely and efficiently in urban environments. Unfortunately, the unbounded expansion of computational resources is not typically an option since more computational power is directly proportional to cost and weight. Thus, new approaches are needed to understand how to treat bounded error and (potentially) unbounded time to compute a solution.

This position paper examines challenges and possible solutions in the areas of open experimental platforms, component based design, dynamic adaptation, and modeling languages and conventions.

2 Current Limitations

• *Limitations of Software Reuse:* Customized software solutions for every vehicle platform or model is not only expensive for development, but also claims resources for testing and (likely, someday) certification of software implementations. For example, simple tasks may be performed by a traditional P-controller, which has a computation time that can be analytically bounded. Thus, it is possible to re-use a software version of this controller in any suitable application by simply changing the gains.

As more advanced behaviors become a differentiator among manufacturers, however, it may not be possible to rely on such fundamental algorithms; thus, the complexity of behavior raises along with the complexity of the computational task. One approach may be to use online optimization to take advantage of various optimal control strategies for making decisions.

•*Limitations in Vehicle-Independent Algorithms:* Providing more vehicle autonomy for complex tasks means that it may no longer be valid to assume that a closed-form optimization solution can be obtained. Thus it becomes more difficult to predict with certainty the worst-case execution time (if existing at all!) of a software component that utilizes online optimization. In order to effectively field vehicle features across a broad set of vehicle platforms, however, there is no opportunity to provide a custom implementation for each vehicle. Rather, it makes more sense to understand whether a generic kinematic or dynamical model may be used with bounded error, in order to take advantage of bounded-time optimization techniques.

For example, automatic parking exists in many vehicle platforms for specific manufacturers, but for arbitrary vehicles—with generic software—the state of the art is limited. In future years, tasks will become more complicated than "just" parking, but foundations laid now will continue to mature. Active, certified control for hazardous maneuver avoidance is a highly desired and expected to be the next automotive automation milestone. Fully-autonomous (with human-in-the-loop) vehicles can build upon such crash-avoiding algorithms.

• *Limitations in Computing Resources:* In 20 years, it is not disparate to expect that the on-board computers in our neighboring vehicles — not the "cloud", but the "SMOG" (Shared Multi-vehicle Optimization Grid) — can help "old" vehicles make better decisions. This would permit a legacy car to utilize the advanced on-board capabilities in newer, neighboring vehicles. Building on the foundation of a component-based design, an abstracted communication framework permits distributed optimization, various strategies, etc., but must be robust to communication failures and delays, unexpected events, and premature stops of computation of control algorithms.

New theory is needed to bridge the gap between computation and control to quantify (somehow) the robustness, and more importantly the computational burden for various maneuvers, at various speeds, constrained by the acceptable error of the maneuver. The design space will be too big to permit a lookup-style table for every problem at every level, which leaves rapid online decision-making execution of optimization as a potential solution.

• *Limitations in Experimental Platforms:* Although it is certainly a cliche, truly the difference between theory and practice is smaller in theory than it is in practice. CPS research in automotive systems requires not only high-level design and testing, but also validation on high-resolution simulators, or physical hardware. Physical hardware is prohibitively expensive for widespread research, and hi-resolution simulators are unlikely to run fast enough in real-time to integrate with hardware sensors or with experimental data. To validate research results, open-experimental platforms that are friendly for theory validation are critical.

3 Potential Approaches

A potential solution is the generation of *software* for automotive tasks that accounts for the *computational* resources required to perform the task with the necessary accuracy. The constraints imposed by the environment, such as roads, traffic systems, obstacles, pedestrians, and maximum allowed speeds translate into spatial and temporal design constraints that the algorithms implemented in software have to take into account.

Only in appropriate conditions can such problems be solved relatively fast using traditional dynamic state-feedback control algorithms obtained via numerical optimization methods. A general issue of such methods is the lack of a guaranteed termination time. In fact, unexpected system conditions may result in unbounded time during iteration of the algorithms to, for example, update the control inputs to a self-driving vehicle. To address issues such as these, we pose several potential approaches.

⋆ *Approximate, analytical models of computational requirements:* Decisions on what maneuvers are available at any given time must be made with respect to the resources available to execute those maneuvers. For this, models capturing the computational power at the current time are required. If computational resources are available in a componentized form, this permits the utilization of resources in neighboring vehicles, or from smart infrastructure.

⋆ *Design Tools:* Algorithms that account for computational resource constraints may take a hybrid control approach, switching between predictive plant models, or automatically changing the lookahead time in order to bound error. New tools are needed to enable the designer to make informed decisions about what online options are feasible, what is the approximate worst-case computation time, and how to design robust switching laws selecting among many plant models and controllers. These tools will need to handle not only single vehicles but rather networks of vehicles, for which interconnection and composition theories for cyber-physical systems will become a cornerstone.

⋆ *Open-Experimental Platforms:* An OEP that permits component-based integration for autonomy and other tasks, will permit collaboration among researchers, as well as comparison alternative approaches to the same task in identical conditions. A model-based approach will permit software synthesis from a high-level, as well as integration with well-established modeling tools for control and computational tasks for prototyping. With this foundation, collaboration could also permit suggested experiments by researchers who do not have access to the hardware.

⋆ *Model Transformations for Safe Experiments:* A challenge with OEP access to a wide audience is to avoid damage to the platform, either through bad design or invalid safety parameters. A potential solution this is the transformation of a component-based design to add fault-tolerance and safety observers to the system. Model-transformations can "surround" controllers with these new blocks, and be used to log additional data, or to obey various testing/validation criteria (low speeds, low turn rate, etc.) to build confidence in the initial design, and relax these constraints (if necessary) to test more aggressive behaviors.

4 Conclusions

As the decisions made by smart automobiles increase in complexity, their computational requirements will also increase. In this position paper we describe current limitations in improving the intelligence of vehicles, as well as potential approaches to improve the state of the art.

5 Biographical and Contact Information

Dr. Jonathan Sprinkle (sprinkle@ECE.Arizona.Edu) is an Assistant Professor of Electrical and Computer Engineering at the University of Arizona. In 2009, he received the UA's Ed and Joan Biggers Faculty Support Grant for work in autonomous systems. Until June 2007, he was the Executive Director of the Center for Hybrid and Embedded Software Systems at the University of California, Berkeley. His research interests and experience are in systems control and engineering, through modeling and metamodeling, and he teaches in controls and systems modeling.

Ricardo G. Sanfelice (sricardo@u.arizona.edu) is an Assistant Professor of Aerospace and Mechanical Engineering at the University of Arizona. He received M.S. and Ph.D. degrees from the University of California, Santa Barbara in 2004 and 2007, respectively. Before joining the University of Arizona in 2009, he held a postdoctoral position at MIT. His research interests are in modeling, stability, robust control, and simulation of nonlinear, hybrid, and embedded systems with applications intersecting the areas of robotics, aerospace, and biology.