**Co-design of Computation and Control Algorithms**

**for Low–Cost and Low-Power Autonomous Vehicles**

**PI:** Rahul Mangharam

**Team:**

Matthew O’Kelly, Univ. of Pennsylvania

Yash Pant, Univ. of Pennsylvania

Houssam Abbas, Univ. of Pennsylvania

Madhur Behl, Univ. of Pennsylvania

In collaboration: Shin’ichi Shirashi, Toyota InfoTechnology Center, CA

Prof. Shinpei Kato, Nagoya University, Japan

Sicun Gao, MIT

*DISCLAIMER*

*The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation’s University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.*

**Problem:**

This project has a dual focus for the development of Autonomous Vehicles (AV):

(1) *AV Safety:* ensuring the decision-controller in autonomous vehicles always makes safe decisions (or how to get a driver’s license for AVs); and

(2) *AV Efficiency:* developing an algorithmic framework for co-design of perception/computation algorithms and control systems to lower the cost and energy consumption in future autonomous vehicles (or how to reduce the AV perception and control to operate on a smartphone platform form factor).

1. **APEX: Autonomous Vehicle Plan Verification and Execution:** Autonomous vehicles (AVs) have already driven millions of miles on public roads, but even the simplest scenarios, such as a lane change maneuver, have not been certified for safety. This is a significant problem as the insurance liability of autonomous vehicles currently is entirely on the manufacturer as there is no systematic method to bound and minimize the risk of decisions made by the vehilce’s decision controller. Current methodologies for the verification of AV's decision and control systems attempt to divorce the lower level, short-term trajectory planning and trajectory tracking functions from the behavioral rules-based framework that governs mid-term actions.

We have developed APEX, a tool for verification and execution of autonomous vehicle planning and control decisions across a variety of driving scenarios.

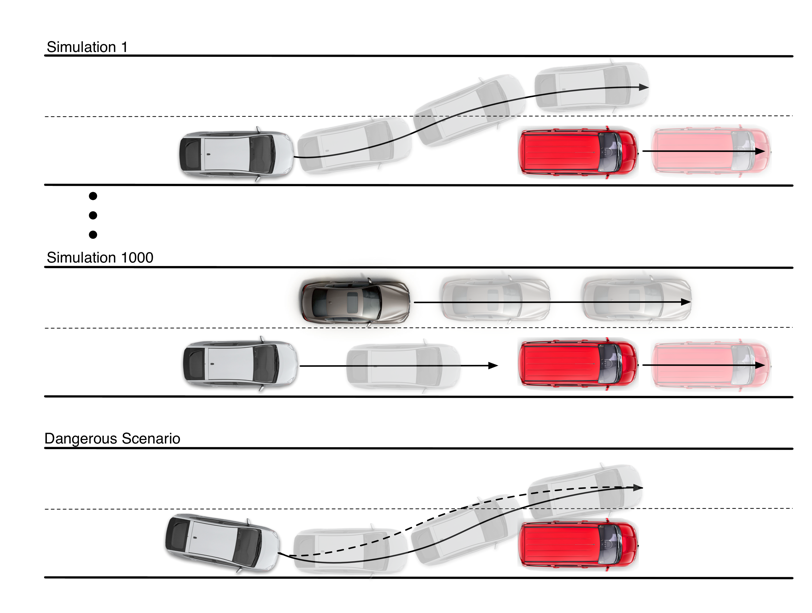


Figure 1: Simulation is not sufficient to fully verify a lane change. After a large number of simulations, the unsafe scenario at the bottom may still not be detected as simulation-based testing is not exhaustive and leaves a verification gap.

1. **Co-design of Anytime Computation and Robust Control for low-cost and low-power AV patforms:** Control software of autonomous robots has stringent real-time requirements that must be met to achieve the control objectives. One source of variability in the performance of a control system is the execution time and accuracy of the state estimator that provides the controller with state information. This estimator is typically perception-based and is computationally expensive. When the computational resources of the hardware platform become overloaded, the estimation delay can compromise control performance and even stability. In this project, we define a framework for co-designing anytime estimation and control algorithms, in a manner that accounts for implementation issues like delays and inaccuracies. Such methodologies can be used to reduce cost and power consumption in AV hardware.

**Approach:**

1) **The APEX verification tool** investigates the combined action of a behavioral planner and state lattice-based motion planner to guarantee a safe vehicle trajectory is chosen. In APEX, decisions made at the behavioral layer can be traced through to the spatio-temporal evolution of the AV and verified. Thus, there is no need to create abstractions of the AV’s controllers, and aggressive trajectories required for evasive maneuvers can be accurately investigated. In APEX, decisions made at the behavioral layer can be traced through to the spatio-temporal evolution of the AV and verified. Recent work includes:

(a) A simulation environment for autonomous vehicles (AVs), which allows the creation of scenarios like intersections and highways.

(b) Implementation of a state lattice local planner and trajectory generator. Including validation of the algorithm on a real Prius and in simulation.

(c) Creation of an agile reduced-scale AV is being developed. Equipped with a camera, an optical flow sensor and a LIDAR, this AV will serve as the experimental testbed on which verified control algorithms will be run.

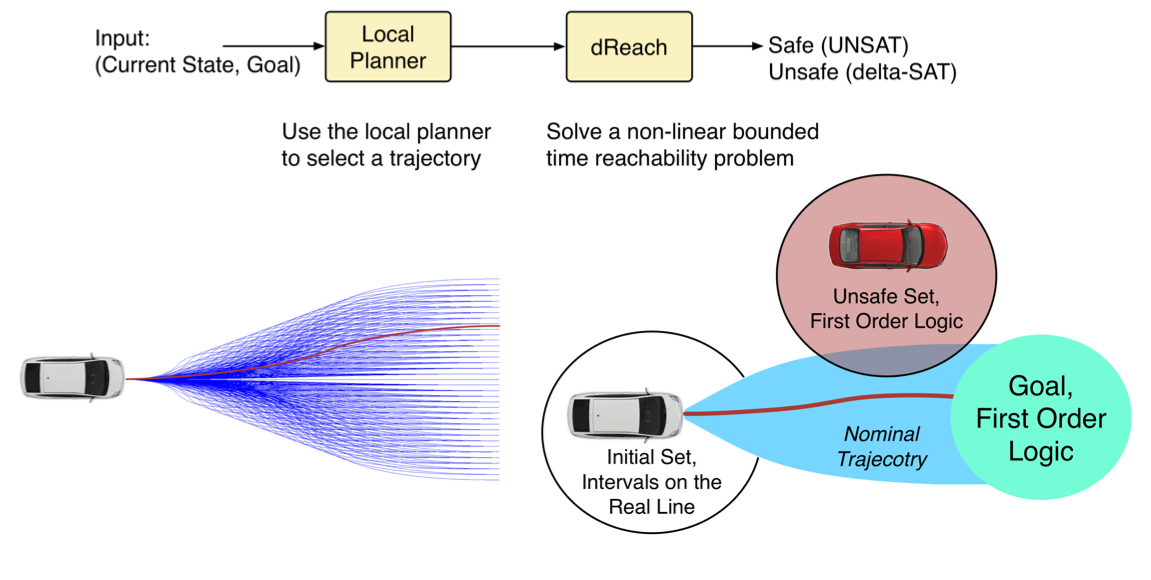


Figure 2: One step in the APEX tool: the local planner gener- ates a trajectory, which is automatically input into the mission description file and verified using dReach

2) **Co-design of Anytime Computation and Robust Control for low-cost and low-power AV patforms:** Control software of autonomous robots has stringent real-time requirements that must be met to achieve the control objectives. One source of variability in the performance of a control system is the execution time and accuracy of the state estimator that provides the controller with state information. This estimator is typically perception-based (e.g., Computer Vision-based) and is computationally expensive. When the computational resources of the hardware platform become overloaded, the estimation delay can compromise control performance and even stability. In this project, we have developed a framework for co-designing anytime estimation and control algorithms, in a manner that accounts for implementation issues like delays and inaccuracies.

We construct an anytime perception-based estimator from standard off-the-shelf Computer Vision algorithms, and show how to obtain a trade-off curve for its delay vs estimate error behavior. We use this anytime estimator in a controller that can use this trade- off curve at runtime to achieve its control objectives at a reduced energy cost. When the estimation delay is too large for correct operation, we provide an optimal manner in which the controller can use this curve to reduce estimation delay at the cost of higher inaccuracy, all the while guaranteeing basic objectives are met. We illustrate our approach on an autonomous hexrotor, autonomous ground vehicle and demonstrate its advantage over a system that does not exploit co-design.



Overloaded

Figure 3. Contract-based anytime controller and estimator

**Methodology:**

**1) APEX:** Our main contribution is an approach to *formally* verifying the trajectory planning and trajectory tracking stacks of an ADAS/AV. This approach is implemented in a software tool, APEX, and illustrated with examples of a lane change maneu- ver. The verification approach has two characteristics:

1. It is formal: we are *guaranteed* that if APEX determines a scenario to be safe, then it is safe. No amount of simulation can find an unsafe behavior in a scenario verified as correct by APEX.
2. It allows the use of an arbitrary trajectory planner, includ- ing one that only exists as code. That is, there is no need to model the trajectory planner, which is often very complex software. Moreover, the same trajectory planner can then be run on a real vehicle. APEX uses a trajectory planner that has been tested on a real vehicle.
3. In APEX, the verification engineer can

• Specify the low-level dynamics of the vehicle, including  the trajectory tracker. These can be nonlinear. The default model in APEX is a 7D bicycle model.

* Provide a motion planner that takes in a starting position and end position and returns a trajectory that links the two points. The motion planner can be *any piece of software*: there are no restrictions on it. The default planner in APEX is a state lattice planner incorporated in ROS and tested on a real vehicle.
* Specify a sequence of goal positions (or *waypoints*) that the vehicle must visit, or a behavioral planner that com- putes these waypoints in a reactive manner. The default behavioral planner in APEX is a simple 2-state automaton that decides whether to do lane following or lane changing.
* Specify the uncertainty sets for the ego vehicle and the other agents in the scenario.
* Specify the unsafe conditions to be avoided by the vehi- cle. APEX supports a rich specification language (namely, Linear Temporal Logic) for describing unsafe behaviors.

APEX will then verify, in an exhaustive fashion, that the ego vehicle can complete the scenario under the specified uncer- tainty, or return a specific case where it fails. The engineers can then use this *counter-example* in order to debug the controllers, and better understand how to avoid this failure. APEX is to be used at design-time by the verification engineers. It will allow them to quickly make modifications to the car’s controllers, and exhaustively verify the scenarios of interest.

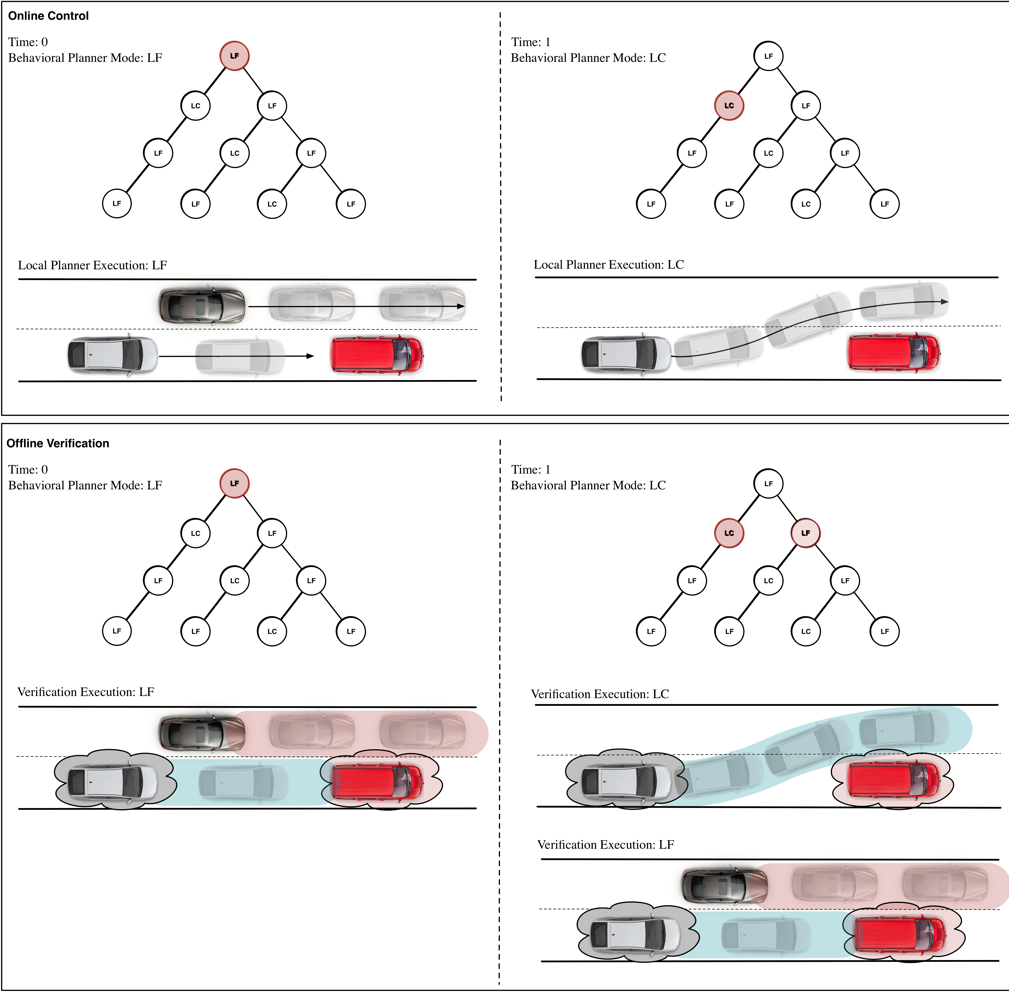


Figure 4: Stages of APEX verification and their correspondence to control execution. Top left: at t = 0, mode is Lane Follow (LF) and the vehicle follows the current lane. Top right: at t = 1, mode is Lane Change (LC) and vehicle starts a lane change maneuver. Bottom left: offline, APEX verifies that in mode LF, the vehicle can track the trajectory. Bottom right: offline, APEX verifies both possible executions, a lane change and a lane following.

2) **Co-design of Anytime Computation and Robust Control for low-cost and low-power AV patforms:** In this work, we develop a *co- design framework* for a real-time control systems, where the controller and estimator communicate via *contracts*. A contract is a guarantee requested by the controller, and fulfilled by the estimator, that the latter can provide an estimate with a certain maximum error ε, and within a certain deadline δ. Both the deadline and the error bound are part of the contract. Using these contracts, we show how the controller can throttle the execution time of the estimation task to preserve good performance and to reduce energy consumption. Our work focuses on estimators that incorporate computationally intensive Computer Vision (CV) algorithms, such as those used in autonomous robot navigation. We refer to these as *perception-based estimators*. Our experiments validate that the execution time of these algorithms is significant and far exceeds the computation time of the control software, and can have an effect on control performance.



Fig. 4 presents the proposed structure of contract-based estimation and control. It shows a traditional feedback loop incorporating estimator, controller and the physical system, augmented with the (Delay, Error) contract between con- troller and estimator. This contract forms the basis of the proposed approach.

Summary of contributions: We present a contract-based framework for the co-design of real-time controller and estimator algorithms, consisting of:

• a well-defined interface between control and estimation, in the form of operating modes or *contracts* on the accuracy and delay provided by the estimator

• a controller design that can vary the accuracy and delay of the estimation to achieve control objectives at a lower energy cost, and

• a general procedure to compose run-to-completion esti- mation algorithms into a contract-based estimator.

• We illustrate our approach on an autonomous flying robot, ground robot and demonstrate performance and energy gains using our approach over a classical controller.

Macintosh HD:Users:rahulm:Publications:AnytimeCPS2015:RTSS15:Paper:RTSS15_Camera:Paper:RTSS15_Paper:figures:CostAndModes.pdf

Fig 5. For a quadrotor in autonomous flight, the tracking cost at each time step for Model-Predictive Control (MPC) and our proposed Robust Adaptive MPC (RAMPC). Note how the RAMPC performs better (lower cost) than the MPC and there is dynamic switching of estimator modes at runtime leading to improved performance for the RAMPC.

**Findings:**

[1] M. O'Kelly, H. Abbas, S. Gao, S. Shiraishi, S. Kato, R. Mangharam. "APEX: Autonomous Vehicle Plan Verification and Execution,” in SAE International, April 2016.

[2] Y.V. Pant, H. Abbas, K. Mohta, T. X. Nghiem, J. Devietti, R. Mangharam, Co-Design of Anytime Computation and Robust Control, Proceedings of the Real Time Systems Symposium, San Antonio, U.S.A , 2015.

[3] K.N. Nischal, P. Kelkar, D. Kumar, Y. V. Pant, H. Abbas, J. Devietti, R. Mangharam, Hardware optimizations for anytime perception and control, Work-in-Progress at the Real Time Systems Symposium, San Antonio, U.S.A , 2015.

[4] Y. V. Pant, H. Abbas, K.N. Nischal, P. Kelkar, D. Kumar, J. Devietti, R. Mangharam, Power-efficient algorithms for autonomous navigation. Proceedings of the IEEE International Conference On Complex Systems and Engineering, 2015.

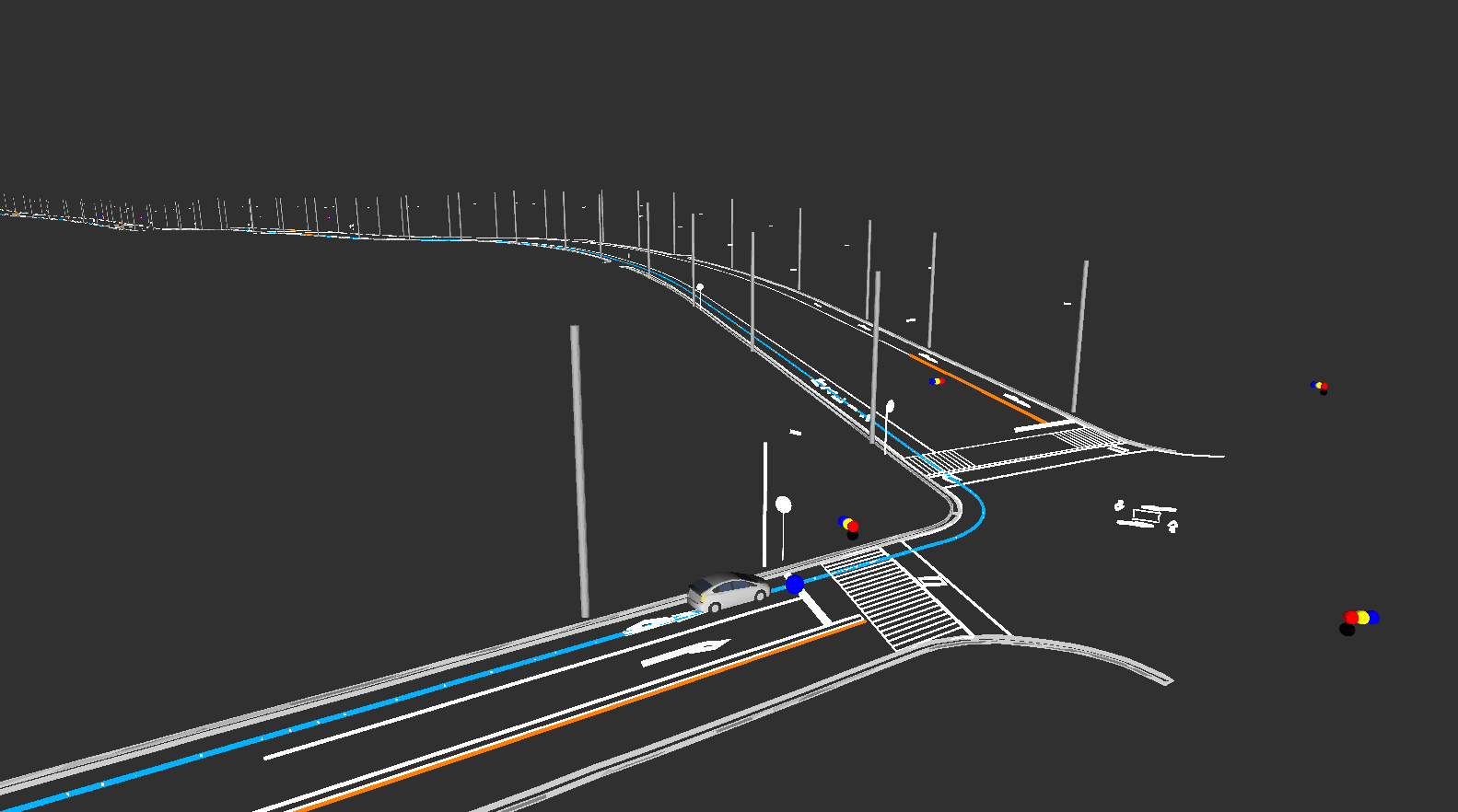
**Conclusions:**

1) APEX is a tool for formally verifying the trajectory planning and tracking stacks of ADAS/AV cars, and is available for download. It can perform formal verification on realistic au- tonomous vehicle planning stacks. Future work will incorpo- rate more complex behavioral controllers for other scenarios, including synthesized planners, and will add a GUI to the tool.

2) Anytime Computation and Robust Control: Early results show the good performance of our scheme and how it outperforms regular Model Predictive Control which does not leverage co-design. A key result showed how our closed loop solution is more energy efficient than MPC while achieving better tracking performance. A focus of ongoing research is to overcome the necessity of the contracts always being met by the estimator. Another focus is on an automated tool chain to profile perception algorithms commonly used in autonomous systems.

**Recommendations:**

We are currently pursuing potential collaborations with industry to produce a deployable prototype on a Toyota Prius. We have the support of Toyota ITC, Mountain View, CA. We also have stong collaboration on the AV software from Nagoya University. Continued funding should help realize the development of such a prototype.



APEX deployment in progress with real vehicle prototype.