Synergy: Verified Control of Cooperative Autonomous Vehicles

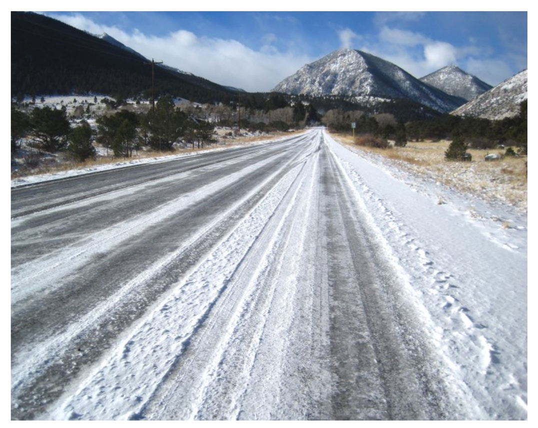
Christoffer Heckman, Lijun Chen, Dirk Grunwald, John Hauser, Sriram Sankaranarayanan

University of Colorado, Boulder

Award #1646556

Problem

Autonomous (self-driving) vehicles hold the promise of revolutionizing areas such as transportation, logistics, manufacturing, remote sensing and defense. However, the potential widespread use of autonomous vehicles has raised numerous **safety concerns involving the reliability of the control algorithms** that drive these vehicles. In particular, these algorithms control vehicle maneuvers, wherein faulty control can potentially endanger human life and property. For autonomous cars, response to unforeseen situations involving imminent collisions and suboptimal driving conditions such as **icy, wet or damaged roads** may involve "aggressive" maneuvers that exploit the complex nonlinear vehicle dynamics. However, verifying such controllers is challenging.



Project Goals

Our work seeks to develop **verified maneuver regulation algorithms** to characterize the types of maneuvers that can be controlled in a safe and stable manner [5, 13].

Goal #1:

Construct guaranteed maneuver regulation control algorithms and characterize the space of maneuvers that are controllable given "driving conditions."

Goal #2:

Transition from model level to augmenting the overall autonomous vehicle design.

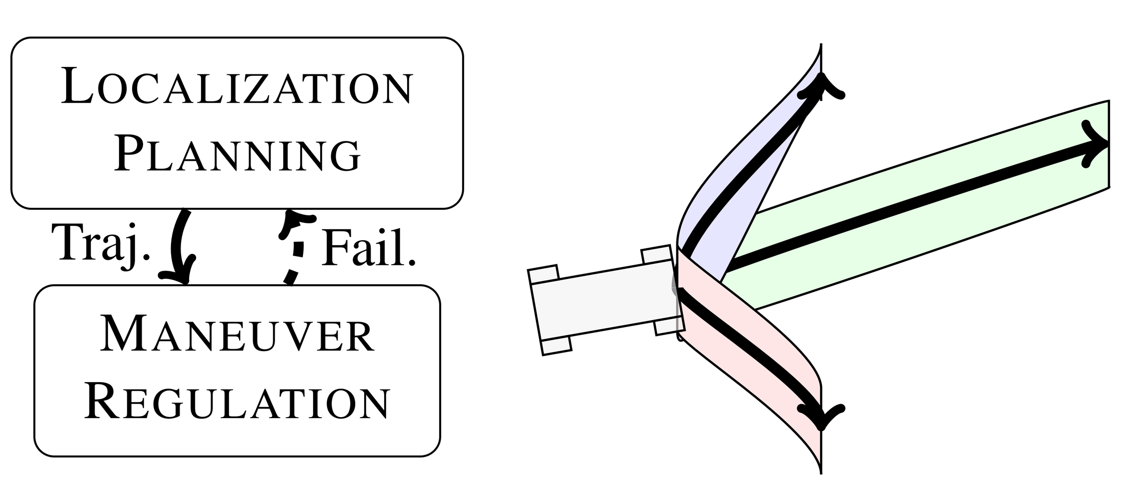
Goal #3:

Lift ideas from a single vehicle to multiple co-operating vehicles.

Evaluation Testbed:

Evaluate each step of our work using the Ninja Car platform.

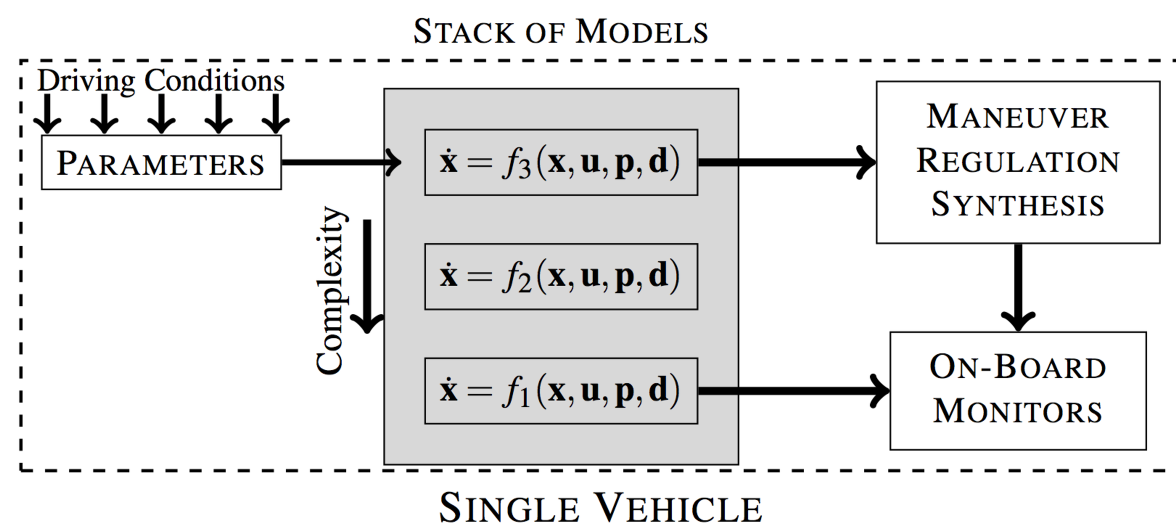
Control Model



Guaranteed safe and stable *maneuver regulation* is key to building reliable and robust autonomous vehicles. Our approach treats the control-planning problem as a hierarchical structure in which **trajectory optimization and maneuver regulation are interwoven**, with the possibility for interruption through failure events which would then adjust the controller in response.

Our approach lifts from characterizing the space of maneuvers possible for a single vehicle to a cooperative setting [11]. We are investigating how vehicles can **cooperatively share driving information to map road conditions** in real time. Furthermore, we are creating protocols for avoiding collisions cooperatively by negotiating maneuver selection to avoid collisions in real-time. Here we focus on determining what complexity of state space is needed to facilitate cooperation for time-constrained incidents in challenging networking environments.

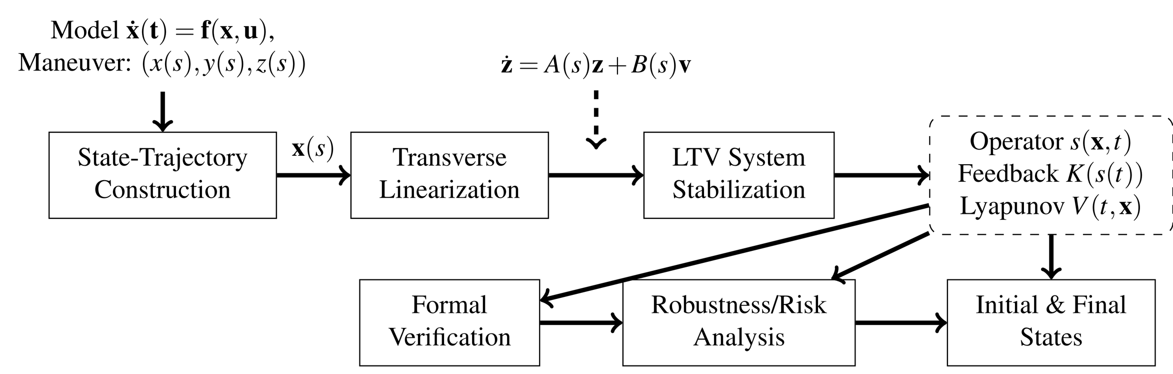
Dynamical Models and Parameter Identification



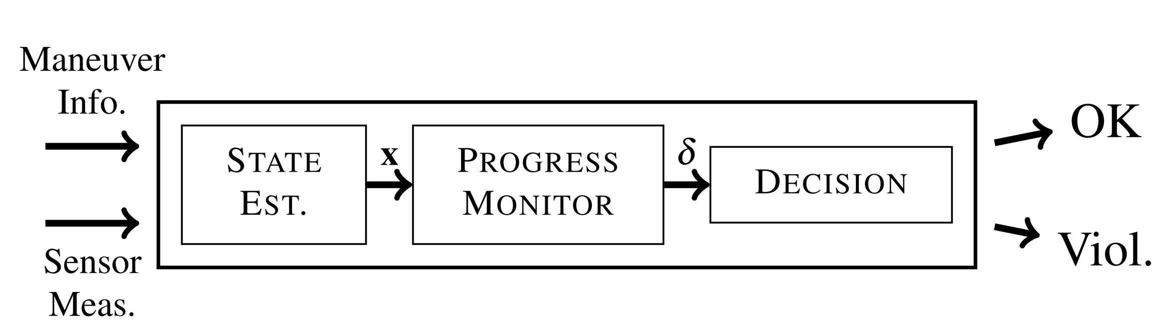
Our maneuver regulation algorithms are based on dynamical models of the vehicle with (time-varying) model parameters that capture characteristics of the vehicles and road conditions. Our work then investigates the **construction of a *stack of models*** and characterization of the driving conditions in terms of representative sets of model parameters. An existing experimental testbed, called the *Ninja Car*, at the University of Colorado, Boulder will be used to construct these models [6]. Our framework will simulate various driving conditions in the laboratory and identify parameters for the proposed models.

Currently, the approach used to simulate the Ninja Car platform's dynamics has 28 parameters and over 30 state variables, some of which evolve according to constraints that may be used to decompose the model to 12 state variables. The state variables refer to $\mathbb{SE}(3)$ pose and time derivatives, and the model parameters describe physical quantities e.g. the wheel base, torque-speed curve slope, and spring stiffness.

Guaranteed Maneuver Regulation

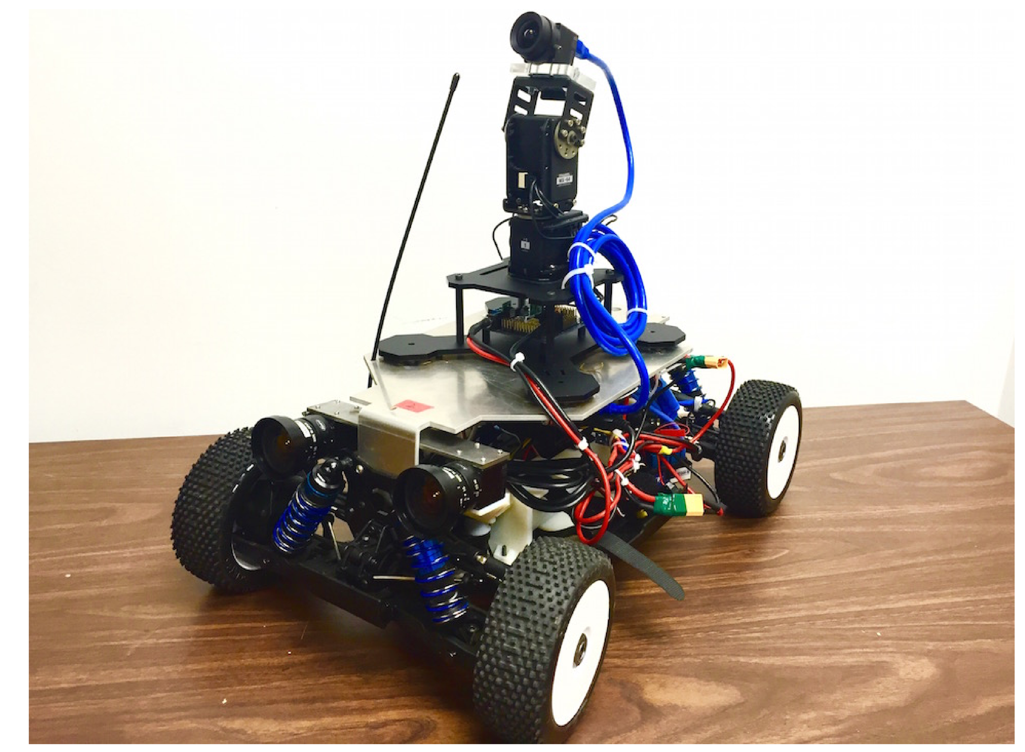


With the dynamical models are constructed, the next stage of our effort involves characterizing the set of maneuvers for the given model and parameter set [5, 13]. The result is a **succinctly represented set of trajectories** that can be carried out for given sets of initial states of the model and parameter values (much along the lines of a flowpipe construction [1, 4]), wherein for each trajectory, we construct associated Lyapunov and barrier functions. Sampling the trajectory space densely yields a *database of maneuvers* that incorporate trajectory information along with associated feedback law, Lyapunov and barrier functions.



Recent work at the intersection of formal methods and control design has given us promising directions for building control algorithms that incorporate **formal high level guarantees such as safety and stability** [2, 7, 8, 10, 12]. Building on these works, we will synthesize guaranteed control algorithms to regulate maneuvers of autonomous vehicles by bridging the gap between the capabilities of formal control design schemes and the practical details involved in translating them to realistic driving scenarios. We will employ planning algorithms such using sample-based stochastic searches in order to search for possible trajectories while also guaranteeing their behavior through these methods. To address modeling uncertainties, **on-board controllers** under development by our team will repeatedly estimate the model state and verify that the Lyapunov/barrier certificates are respected during runtime [9, 14].

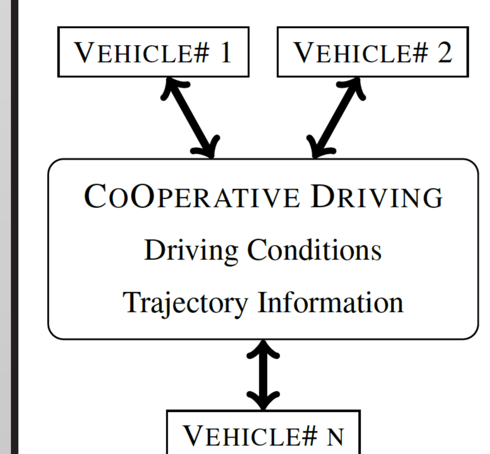
Experimental Testbed



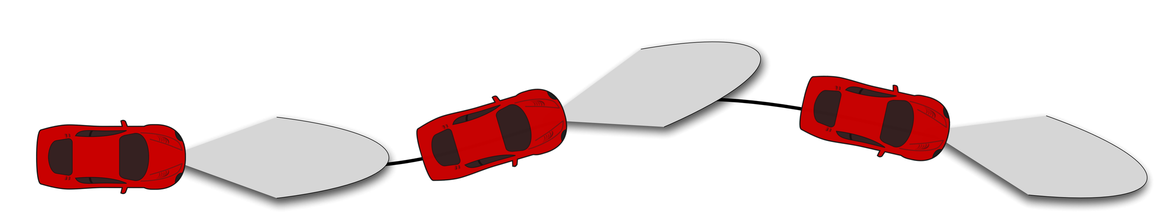
The Ninja Car is a $\frac{1}{8}$-scale model of a car that has been modified for autonomous operation using **on-board sensing and computation**. It is capable of driving in various environments through structural augmentations and has on-board computation abilities to support experimental autonomy algorithms for driving. The API for the vehicle's control and sensing algorithms provide the inputs available for all levels of controller synthesis. Currently, the test environment consists of only a carpeted surface, but other driving conditions such as icy, sticky and granular surfaces will also be introduced to test the elasticity of our algorithms to changes in the environment. Furthermore, different levels of abstraction for controller synthesis (e.g. velocity control versus torque control) are implemented to allow for exploration over control input spaces. This enable our rapid testing of control algorithms on experimental platforms in order to guide ongoing development.

The Ninja Car platform is outfitted with advanced **state feedback sensors**, including a global-shutter stereo rig, high-end MEMS IMU, swing arm encoders, high-precision optical wheel encoders, augmented structural components, and a quad-core i7 miniature computer with a discrete graphics card. The mast with attached pan-tilt unit as pictured is removable.

Multi-Vehicle Design



It is impractical to communicate the detailed information on road conditions, possible maneuvers, and predicted trajectories to other vehicles. Therefore we require a **coarse characterization of the admissible maneuver and trajectory space** as well as the road condition. We are investigating the levels of granularity for characterizing the relevant information spaces to strike an optimal tradeoff between the communication complexity and the utility of the receiving information for prediction and coordination. This effort requires codesign with guaranteed methods in radio transmission [3].



The tradeoff involved in cooperative driving in challenging situations is different from that for the single vehicle case. For example, for a sharp turn on a loose ground, a single vehicle may allow large lateral displacement for maximum stability while we may have to bound the displacement in the multi-vehicle situation. This requires ranking (and possibly pruning) maneuvers based on certain computed bounds *e.g.*, the expected maximum longitudinal and lateral displacements, as indicated above. In this figure, the arc represents the maximum possible maneuvering range with a probability estimate for a given region being the likely maneuver to be attempted. Based on this probability, we may **prune the maneuver space**, *e.g.*, not consider those maneuvers that have a maximum lateral displacement more than a certain value.

Outreach and Education

We are integrating our research into a series of education and outreach activities that will ensure the broader impacts of this project. The Ninja Car testbed will be **disseminated as a "do-it-yourself" project** for interested students and enthusiasts under \$300. The testbed is already being used in project-oriented classes offered to undergraduate students in engineering; these classes are offered to select students, to motivate their choice of an engineering discipline, and develop an unified understanding of engineering and design. The testbed is also being used in focused graduate classes on building autonomous vehicles from the ground up. Furthermore, we will support local teams of students participating in competitions such as those organized by `sparkfun.com`.

Finally, we are incorporating our testbed into our ongoing online education initiatives. We are designing a **series of project oriented courses** that will leverage an existing MOOC on linear and integer programming on the Coursera platform, wherein the focus will be on using optimization techniques for control of autonomous vehicles.

References

[1] X. Chen, E. Ábrahám, and S. Sankaranarayanan. Flow*: An analyzer for non-linear hybrid systems. In *Proc. of CAV'13*, volume 8044 of *LNCS*, pages 258–263. Springer, 2013.

[2] R. Dimitrova and R. Majumdar. Deductive control synthesis for alternating-time logics. In *Embedded Software (EMSOFT), 2014 International Conference on*, pages 1–10. IEEE, 2014.

[3] A. Dutta, D. Saha, D. Grunwald, and D. Sicker. SMACK: a SMart ACKnowledgment scheme for broadcast messages in wireless networks. In *ACM SIGCOMM Computer Communication Review*, volume 39, pages 15–26. ACM, 2009.

[4] G. Frehse. PHAVer: Algorithmic Verification of Hybrid Systems Past HyTech. *International Journal on Software Tools for Technology Transfer (STTT)*, 10(3), June 2008.

[5] J. Hauser and R. Hindman. Maneuver regulation from trajectory tracking: feedback linearizable systems. In *Proc. IFAC Symp. on Nonlinear Control Systems Design (NOLCOS)*, pages 595–600, 1995.

[6] C. Heckman, N. Keivan, and G. Sibley. Simulation-in-the-loop for planning and model-predictive control. In *Robotics Science and Systems (RSS) Workshop on Realistic, Repeatable and Robust Simulation*, 2015.

[7] M. Kloetzer and C. Belta. A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Trans. on Aut. Control*, 53:287–297, 2008.

[8] H. Kress-Gazit, G. Fainekos, and G. G. Pappas. Temporal logic based reactive mission and motion planning. *IEEE Trans. on Robotics*, 25:1370–1381, 2009.

[9] S. Mitsch and A. Platzer. ModelPlex: Verified runtime validation of verified cyber-physical system models. *Form. Methods Syst. Des.*, 2016.

[10] S. Mouelhi, A. Girard, and G. Gössler. Cosyma: a tool for controller synthesis using multi-scale abstractions. In *Proceedings of the 16th international conference on Hybrid systems: computation and control*, pages 83–88. ACM, 2013.

[11] R. M. Murray. Recent research in cooperative control of multivehicle systems. *Journal of Dynamic Systems, Measurement, and Control*, 129(5):571–583, 2007.

[12] H. Ravanbakhsh and S. Sankaranarayanan. Counter-example guided synthesis of control lyapunov functions for switched systems. In *Decision and Control (CDC), 2015 IEEE 54rd Annual Conference on*. IEEE, 2015.

[13] A. Saccon, J. Hauser, and A. Beghi. A virtual rider for motorcycles: Maneuver regulation of a multi-body vehicle model. *IEEE Trans. on Control Systems Technology*, 21:332–346, 2013.

[14] S. D. Stoller, E. Bartocci, J. Seyster, R. Grosu, K. Havelund, S. A. Smolka, and E. Zadok. Runtime verification with state estimation. In *Proceedings of the Second International Conference on Runtime Verification*, RV'11, pages 193–207, Berlin, Heidelberg, 2012. Springer-Verlag.