Rapidly-exploring Random Trees for Testing Automated Driving Systems

Cumhur Erkan Tuncali and Georgios Fainekos

***Abstract*— This paper proposes the use of Rapidly-exploring Random Trees (RRTs) for generating maneuvers for adversarial vehicles to Automated Driving Systems (ADS). In this approach, the test generation problem is considered as a path planning problem for agent actors with the aim of creating interesting collisions with Ego vehicles (ADS). A cost function is proposed to guide the test generation toward the boundaries between safe and unsafe scenarios, *i.e.*, the neighborhood of the cases where there is a collision that could have been avoided or a near miss which could have resulted in a collision with minor changes in the inputs/trajectories.**

I. Introduction

Rapidly-exploring Random Trees, first introduced by LaValle in [1], provide an efficient method for exploring and covering high-dimensional spaces. Although RRTs are mostly used for path planning, they find applications in various domains including test case generation [2], [3], [4].

In this work, we focus on the application of RRT to the test case generation for Automated Driving Systems (ADS). The main challenge in testing ADS is that in open world environments, it is very easy to create scenarios for which a collision is inevitable for the Ego vehicle[1]. One way to address this problem is to define appropriate cost metrics for identifying critical scenarios. That is, scenarios for which the Ego vehicle came close to a collision, or it collided with a small energy (implying that with a better motion planning algorithm the collision could have been avoided).

We propose a new such cost metric and we merge Transition-based RRT (T-RRT) [5] and RRT* [8] for test case generation for ADS. T-RRT [5] extend the classical RRT by incorporating additional cost criteria to the explored paths rather than only aiming to reach a target configuration. They can be viewed as a method that is merging RRTs with stochastic optimization.

Our previous work on using falsification methods for generating agent trajectories for testing autonomous vehicles used global optimizers [6]. In that work, we proposed a cost function do discover small-speed collisions and used S-TaLiRo [7] for generating falsifying trajectories.

In this work, we improve on our previous work on falsification-based approach for testing ADS [6]:

1. In [6], we use a limited number of control points over the longitudinal position axis as the specific points where the lateral axis of the vehicle trajectories are sampled. As the number of control points decreases, possible variations in shapes of the trajectories are limited. On the other hand, increasing the number of control points increases the search space dimension.
2. In [6], the duration of the simulations is fixed. With the RRT-based approach, there is more flexibility in the duration of the simulations. So, non-promising simulations are stopped earlier while more promising ones can be executed for longer times.
3. RRT-based approach minimizes the need for the manual design of a test scenario in detail and allows more freedom in the exploration of the space compared to the falsification approach.
4. RRT-based approach is promising to avoid local minima that can be challenging for the optimizer used in [6].
5. We propose a new cost function that does not have discontinuities as the one proposed in [6] and that also considers spatial properties of accidents besides collision speed.

II. Overview of the Approach

A scenario is described by the sets of Ego vehicles, agent actors and environment. A tree grows while seeking to discover interesting behaviors. While growing the tree, instead of executing simulation traces starting from the initial configuration, only a partial simulation is executed starting from an existing tree node. For that purpose, the state of the system, the state of the controllers, and the simulation time are stored on the tree nodes. Immediate target path segments for agent are randomly sampled as sets of waypoints with constraints such as the boundaries of the roads or buildings.

Once target path segments for the agent vehicles are sampled, one node is selected from the existing tree as the initial configuration for the simulation that will be executed with the sampled target path segments. Following the selection of an optimal existing node in the tree, a partial simulation of a predefined length is executed starting from the configuration represented at that node. In this study, the notion of picking an optimal node is adopted from the RRT* method [8]. That is, in [8], when adding a new node to the tree, existing nodes in a neighborhood of the new node are all checked and the one which minimizes the cost is picked as the previous node.

After the partial simulation from the previous configuration is executed, a cost function is used to compute how close the simulation trace is to an interesting behavior. The approach described here can be utilized to discover other types of interesting/failing behaviors; however, our target in this work is to explore the behaviors that are on the boundary

This work has been partially funded by NSF 1350420 and 1446730

C.E. Tuncali, and G. Fainekos are with School of Computing, Informatics and Decision Systems, Arizona State University, Tempe, AZ 85281, USA (e-mail:{etuncali, fainekos}@asu.edu)

[1]Ego vehicle refers to the vehicle under test.

between safe and unsafe operation. Hence, an interesting behavior for our purposes would be (i) a collision between an Ego vehicle and an agent that could have been avoided with a minor change in the control applied or agent trajectories, (ii) an almost-collision (near-collision) which could have ended with a collision with a minor change in the control applied or agent trajectories.

Given a simulation trace τ, we propose the cost function:

$$\mathcal{R}(\tau) = (1 + s_{coll,\tau})(v_{coll,\tau}^2 + ttc_{min,\tau}^2) \quad (1)$$

where $s_{coll,\tau} \in [0, 1]$ is the ratio of the collision surface to the overall surface on the collision side of the vehicle, $v_{coll,\tau}$ is the relative speed of the vehicles at the moment of collision, and $ttc_{min,\tau}$ is the mimimum time-to-collision encountered during the simulation output trace τ. For simulations with a collision, $ttc_{min,\tau}$ is 0. For simulations without a collision, $s_{coll,\tau}$ and $v_{coll,\tau}$ are computed at the instance of smallest time-to-collision assuming that the vehicles continue their motion without changing their speeds and orientations.

After the cost is computed, we perform a transition check as described in T-RRT [5] and a novelty check which is modified from the "Minimal Expansion Control" step of T-RRT by computing the novelty of the new configuration with respect to old configurations. If the new configuration passes transition and novelty checks, it is added to the search tree.

A. Case Study

Scenario Setup: In our case study, we have 4 agent vehicles and 1 Ego vehicle on a multiple-lane straight road. Figure 1 gives a high-level overview of the test setup. The initial position of Agent 1 is sampled in the continuous space defined by Lanes 1 and 2, while all the other agents start somewhere in Lane 3. Initial and target velocities over the trajectories are also part of the search space.

Ego vehicle has 5 distance sensors in the perimeter of the vehicle. We remark that the Ego vehicle is autonomous and its goal is to drive in Lane 4 while avoiding collisions. Agent vehicle 1 is controlled by the *move-to-pose* controller described in [9] while Agent vehicles 2, 3, and 4 are driven with a constant speed on a straight line.

Experiment Results: For this case study, we only search for an optimal trajectory for Agent 1 with the target of minimizing the cost function in Eq. (1). The existence of Agents 2, 3, and 4 between the Ego vehicle and Agent 1 creates many local minima for the minimization problem.

In this case study, out of 45 experiments, only 3 of the minimum-cost trajectories returned by the falsification-based approach [6] were able to drive Agent 1 into the lane of the Ego vehicle, and only 1 trajectory was able to cause a collision. All the other trajectories returned by the falsification approach [6] were stuck in local-minima where Agent 1 tries to get closer to Ego vehicle and ends up colliding with one of the other agent vehicles. On the other hand, in 20 of the minimum-cost trajectories returned by the RRT-based approach, Agent 1 was able to get into the lane of Ego vehicle and it was able to cause the Ego vehicle to collide in 10 of those cases. Figure 2 shows a collision discovered. Agent 1 (red) first forces Ego (yellow) to move right to avoid a collision and then to left where it ends up colliding with Agent 3. Histories of Agent 1 and Ego vehicles are numbered to show their evolution over time.

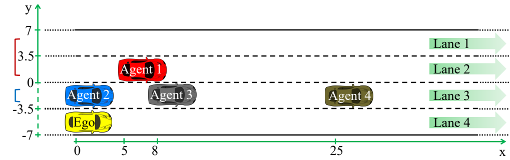

Fig. 1. Initial states of the vehicles in the simulation setup.

III. Conclusion

We proposed an approach that explores maneuvers for road occupants using Rapidly-exploring Random Trees (RRT) with the target of minimizing safety/performance cost functions defined for the ADS under test. Experimental analysis suggests that the RRT-based approach can avoid local minima that can be challenging for our previous method that is performing falsification with a global optimizer.

References

[1] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Tech. Rep., 1998.

[2] T. Dreossi, T. Dang, A. Donzé, J. Kapinski, X. Jin, and J. V. Deshmukh, "Efficient guiding strategies for testing of temporal properties of hybrid systems," in *7th International Symposium NASA Formal Methods (NFM)*, ser. LNCS, vol. 9058. Springer, 2015, pp. 127–142.

[3] J. M. Esposito, J. Kim, and V. Kumar, "Adaptive RRTs for validating hybrid robotic control systems," in *Algorithmic Foundations of Robotics VI*. Springer, 2004, pp. 107–121.

[4] E. Plaku, L. E. Kavraki, and M. Y. Vardi, "Falsification of LTL safety properties in hybrid systems," in *Proc. of the Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, ser. LNCS, vol. 5505. Springer, 2009, pp. 368 – 382.

[5] L. Jaillet, J. Cortés, and T. Siméon, "Transition-based RRT for path planning in continuous cost spaces," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008.

[6] C. E. Tuncali, T. P. Pavlic, and G. Fainekos, "Utilizing S-TaLiRo as an automatic test generation framework for autonomous vehicles," in *IEEE Intelligent Transportation Systems Conference*, 2016.

[7] G. Fainekos, S. Sankaranarayanan, K. Ueda, and H. Yazarel, "Verification of automotive control applications using S-TaLiRo," in *Proceedings of the American Control Conference*, 2012.

[8] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[9] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms In MATLAB® Second, Completely Revised*. Springer, 2017, vol. 118.

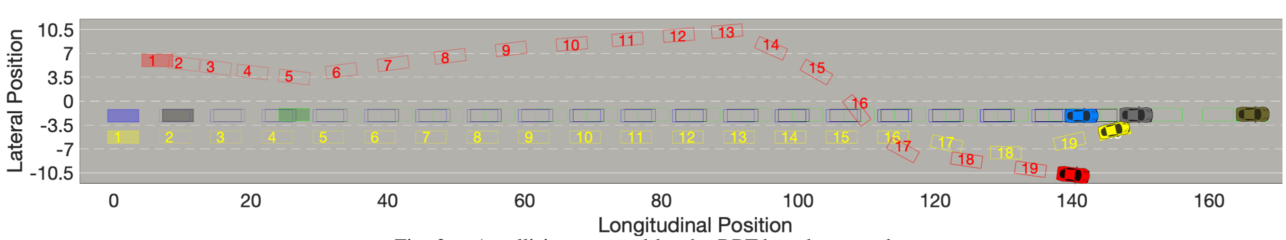

Fig. 2. A collision returned by the RRT-based approach.