Proceedings of the Joint Austrian Computer Vision and Robotics Workshop 2020 DOI: 10.3217/978-3-85125-752-6-12

Reactive Motion Planning Framework Inspired by Hybrid Automata

Csaba Hajdu, Áron Ballagi
Széchenyi István University
{hajdu.csaba,ballagi}@sze.hu

Abstract. *This paper presents a motion planning framework controlled by reactive events and producing feedback data suitable to be processed by various learning and verification methods (e.g. reinforcement learning, runtime monitoring). Our architecture decomposes subtasks of motion planning into separate perception and trajectory planner parts. In our architecture, we interact between these distributed parts through discrete-timed events controlled by timed state machines, besides classical continuous state flow. Our research primarily focuses on autonomous vehicle research, so this framework is supposed to satisfy the requirements of this field. The motion planner framework interfaces a widely-used robotic middleware.*

1. Introduction

Motion planning (or trajectory planning) is a mandatory task both in mobile robotics and in autonomous vehicle navigation [4]. The field has been actively researched and used, providing efficient algorithms suitable for different domains and robot setups. The role of motion planning in robotics is to create a feasible, collision-free path between the location of the agent (mobile robot or vehicle) and an arbitrarily defined goal point, based on the agent's sensory input and actuation. On the other hand, the emergence of autonomous vehicles and other special UGVs requires high-reliability, computational efficiency and optimization of velocity profile even in rough environmental conditions.

The typical problems of motion planner frameworks are their relatively hard extension and limited verification capabilities. In this paper, we propose a prototype of a motion planning architecture with the focus on providing comprehensive verification output and extension capabilities.

Figure 1. Electronically modified autonomous test vehicle

2. Motivation and related work

The development of a new motion planning framework was motivated by ongoing research at our university. We are developing an autonomous vehicle (an electronically modified Nissan Leaf equipped with numerous sensors, Figure 1) and a differential drive robot in various projects. Both rely on motion planning, thus our aim is to create a motion planner framework usable in both application - with minimal configuration effort.

Many commercially available unmanned ground vehicles (UGV) use ROS and its integrated navigation component, *move_base*. This framework is a monolithic implementation with plugin-oriented extension and occupancy grids as a basis of environment representation. Some of these issues had been addressed in *move_base_flex* [5]. In autonomous vehicle frameworks, Autoware [3] provides a loose architecture enabling the replacement of its built-in motion planning component with different solutions. In both approaches, reactive events (e.g. synchronization of all incoming topics, the transition to replanning, etc.) in both systems are relatively hard to trace and debug.

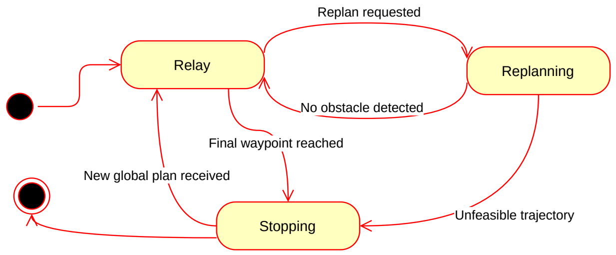

Figure 2. Overview of the local planner state machine described as a state machine

3. Architecture proposal

In this section, we propose the architecture of our motion planner framework[1]. By investigating other planner frameworks we found the following typical characteristics:

1. The underlying planner is controlled by events (e.g. transition to re-planning, recovery initiation) in a fashion of a state-machine based approach (a brief overview of the local planner state machine is shown on Figure 2).

2. Execution should not start before all input information has been received recently.

3. Transitions between states are not necessarily instantaneous, requiring smoother switching between behaviors.

The development of a (hybrid) timed state machine library was motivated by these properties, especially to efficiently resolve property 1 and 2. Hybrid automata [1] is a well-studied way to model and verify systems with both discrete and continuous timed properties and also to describe robot behaviors [2]. A behavior similar to what is presented on Figure 2 can be easily mapped to hybrid automata formalism. Transitions are either governed by discrete events and continuous activities. For example, a continuous variable in this case could be the distance to the closest obstacle detected and a typical discrete event is the request to replan a segment of the trajectory or to execute fallback scenario. We ensure, that each transitions are published to a middleware framework, enabling versatile runtime verification.

Our goal was to follow a highly-distributed architecture, where sub-tasks are decomposed from the planner component. In our approach, the perception related tasks like obstacle detection and classification are decomposed from other specific planner tasks.

[1] Available at https://github.com/kyberszittya/hotaru_planner.git

This enables the reuse of components and isolated verification. Perception components are interacting with planner components by inducing discrete events and modifying continuous signals. For instance, an obstacle detection component may trigger the local planner to replan by raising a discrete-timed event. After the obstacle is avoided, the planner restores the remainder of the original trajectory in relay mode.

4. Conclusion

In conclusion, we provided an overview of a new motion planning framework under development which can be easily extended with new algorithms and tuned to specific domain requirements. A new initial motion planner framework version is created. The extension of our framework with various local planner methods is a primary focus. Global trajectory planner methods will be integrated in the future. Our automata framework and the related code-generator tool will be also enhanced.

5. Acknowledgments

This work was supported by the Hungarian Government and the European Union within the frames of the Széchenyi 2020 Programme through grant GINOP-2.3.4-15-2016-00003

References

[1] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138(1):3 – 34, 1995.

[2] M. Egerstedt, K. Johansson, J. Lygeros, and S. Sastry. Behavior based robotics using regularized hybrid automata. In *Proceedings of the IEEE Conference on Decision and Control*, volume 4, pages 3400 – 3405 vol.4, 1999.

[3] S. Kato, S. Tokunaga, Y. Maruyama, S. Maeda, M. Hirabayashi, Y. Kitsukawa, A. Monrroy, T. Ando, Y. Fujii, and T. Azumi. Autoware on Board: Enabling Autonomous Vehicles with Embedded Systems. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, pages 287–296, Apr. 2018.

[4] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, New York, NY, USA, 2006.

[5] S. Pütz, J. S. Simón, and J. Hertzberg. Move Base Flex: A Highly Flexible Navigation Framework for Mobile Robots. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018.