Postprint

Safe Stop Trajectory Planning for Highly Automated Vehicles: An Optimal Control Problem Formulation

Lars Svensson*, Lola Masson†, Naveen Mohan*, Erik Ward‡,
Anna Pernestål Brenden§, Lei Feng* and Martin Törngren*

Abstract—Highly automated road vehicles need the capability of stopping safely in a situation that disrupts continued normal operation, e.g. due to internal system faults. Motion planning for safe stop differs from nominal motion planning, since there is not a specific goal location. Rather, the desired behavior is that the vehicle should reach a stopped state, preferably outside of active lanes. Also, the functionality to stop safely needs to be of high integrity. The first contribution of this paper is to formulate the safe stop problem as a benchmark optimal control problem, which can be solved by dynamic programming. However, this solution method cannot be used in real-time. The second contribution is to develop a real-time safe stop trajectory planning algorithm, based on selection from a precomputed set of trajectories. By exploiting the particular properties of the safe stop problem, the cardinality of the set is decreased, making the algorithm computationally efficient. Furthermore, a monitoring based architecture concept is proposed, that ensures dependability of the safe stop function. Finally, a proof of concept simulation using the proposed architecture and the safe stop trajectory planner is presented.

Keywords—*Motion planning, Trajectory Planning, Contingency Planning, Safe Stop, Automated Road Vehicles*

I. INTRODUCTION

Vast advances have been made in development of automated vehicles in recent years, but several research challenges remain to be solved before automated transportation solutions can be launched at scale in real traffic systems. One major remaining challenge is to develop dependable capability to detect and handle rare, hazardous events in a safe manner. The hazardous events can be internal to the vehicle system, e.g., a sensor malfunction, or external, e.g., a dangerous situation caused by another traffic participant. Such events cause violations of the nominal operational conditions of the automated vehicle and need to be detected and countered by appropriate actions in order to maintain safety of passengers and other road users. Design guidelines for level 4 (L4) automated vehicles [1] state that a vehicle needs to maintain the capability of reaching a so called minimal risk condition, e.g., to stop safely by the side of the road, if nominal operational conditions are violated. In this paper, the term *safe stop* is used to denote the low dynamic maneuver for achieving a minimal risk condition, at intermediate level hazards, preferably outside of active lanes. This is distinctly different from the *emergency stop*, referring to a high dynamic maneuver that stops the vehicle as fast as possible, at severe hazards. While industry is currently developing L4 automated vehicles, [2] the problem of trajectory planning for safe stop has received limited attention in academia.

Trajectory planning is referred to in literature [3], [4] as the subfield of motion planning concerned with generating local reference trajectories for the vehicle based on the current mission and driving behavior. State of the art methods of trajectory planning [5], [6], are designed as tightly integrated parts of a nominal motion planning functionality and thus do not consider the unique properties of the safe stop problem, i.e that the final velocity is 0 and the goal location unspecified. Recent efforts have been made by Althoff et al. [7], to create trajectory planning benchmarks using optimal control terminology for facilitating reproduction and comparison of results. Their proposed framework does consider stopping maneuvers in specified target locations, but there is no concept of a safe stop area. Thus, the problem of determining where to stop in a safe stop scenario is not considered. Some previous work, like Salvado et al. [8], present algorithms specifically designed for stopping maneuvers, but do not differentiate between different stopping locations. In this work, the following contributions are presented:

- An optimal control problem (OCP) formulation for trajectory planning of a safe stop, based on design guidelines for L4 autonomy [1].

- A safe stop trajectory planning function, including an algorithm based on precomputed stopping trajectories.

The purpose of the safe stop OCP is to use it as a benchmark for quantitative evaluation of different solution strategies in trajectory planning for safe stop. As an example, a novel, real-time capable algorithm based on precomputed trajectory sets is presented and evaluated with respect to the global optimal solution of the safe stop OCP. The algorithm is also integrated and evaluated in a physics based simulation environment.

The paper is organized as follows: Section II provides background and motivation for the safe stop OCP based on automotive design regulation as well as a brief summary of related works in the field of trajectory planning for automated vehicles. Section III describes a monitor based system architecture concept and the interfaces of the safe stop trajectory planner function. Sections IV and V introduce the safe stop OCP and the trajectory library based algorithm respectively. Solution trajectories from the reference method, and from the real time algorithm are compared and discussed in Sections VI and VII.

II. LITERATURE

A. Automotive Design Guidelines

The SAE J3016 [1] provides a taxonomy and design guideline for automated vehicles. It states that an L4 vehicle must be capable of automated fallback maneuvers in case of critical faults implying that a given trip cannot, or should not, be completed. A fallback maneuver shall take the system to a "minimal risk condition", where the risk of a collision is

* Mechatronics and Embedded Control Systems, KTH Royal Institute of Technology, SE-10044 Stockholm, Sweden. `larsvens@kth.se`
†LAAS-CNRS, University of Toulouse, Toulouse, France.
‡ Robotics Perception and Learning, KTH Royal Institute of Technology.
§ Integrated Transport Research Lab, KTH Royal Institute of Technology.

reduced. The minimal risk condition may vary depending on the detected fault and current operating condition. Examples of minimal risk conditions mentioned in [1] are: *vehicle is stopped in previous travel path* and *vehicle is stopped outside active lanes.*

The ISO 26262 [9] is the standard for functional safety assurance in the automotive industry. Among other things it advocates the systematic identification and mitigation of hazards using safety mechanisms. A safety mechanism is defined as a "technical solution implemented by functions or elements, or by other technologies, to detect faults or control failures in order to achieve or maintain a safe state". For automated driving, a safe state generally corresponds to the definition of minimal risk condition in [1]. Hence, complying to [1] and [9] requires a set of functions that realize the minimal risk condition. The Waymo safety report [2] is an example of how the guidelines are implemented in practice. It states that "a minimal risk condition will vary according to the type and extent of a given failure, but may include automatically bringing the vehicle to a safe stop, preferably outside of an active lane of traffic".

B. Trajectory Planning

Trajectory planning for autonomous vehicles in nominal operational conditions has been demonstrated with a wide variety of methods and algorithms. Two notable categories of solution methods are sampling based and optimization based approaches. Ziegler et al. [5] demonstrated the capabilities of optimization based trajectory planning, where a trip of 103 km was completed autonomously without intervention. A local continuous optimization problem was formulated in a Cartesian coordinate frame, with a cost function designed to reflect dynamic feasibility and comfort. A custom solver is used to solve the problem in real-time. Optimization based methods take advantage of the rigorous theoretical framework of model predictive control (MPC). An example of this approach is [10], where an MPC problem is formulated in a curvilinear coordinate frame, thus exploiting the structure of the planning hierarchy by solving the problem locally around a path. Optimization based approaches require obstacles to be captured in convex sets prior to the optimization, as is demonstrated in [5].

A widely used example of a sampling based approach is presented by Werling et al. in [6]. The method uses quintic polynomials, that constitute minimum jerk trajectories for a point mass model. Using a closed form expression for computing the polynomial, a trajectory set is generated online by combining splines for tangential and normal motion with respect to the path in a curvilinear coordinate frame. The set is transformed to the Cartesian frame and a cost function is evaluated for each non-colliding trajectory. The method is robust with respect to disturbances and requires less preprocessing of sensor data as compared to optimization based methods. However, considering that the trajectory set is computed on-line and that there is a trade-off between model detail and computational complexity, the representation of vehicle dynamics in the quintic polynomial approach is less detailed than in comparable methods.

Methods based on precomputed trajectory libraries have been widely used in the robotics domain, since they can accommodate large state space dynamic models and complex dynamic constraints. The concept has been successfully evaluated for several domains including mobile ground robots [11], balancing humanoids [12] and autonomous helicopters, but, to our knowledge, never to safe stop trajectory planning for automated road vehicles. Arora et al. [13] introduce an Emergency Maneuver Library, applied to an autonomous helicopter. A set of pre-computed emergency trajectories are stored at design time, and when the emergency maneuver is triggered, a trajectory is selected and executed based on a cost function. A general problem with the using a precomputed trajectory set is that the size of the set tends to become infeasibly large when the number of discretized states and/or the granularity of the discretization increases. To handle this problem, [13] introduces a post-processing step, where trajecories that are unlikely to yield a useful evasive maneuver are pruned from the set ahead of runtime.

Trajectory planning for safe stop, or contingency planning, has received limited interest in academia, and is sometimes viewed as a special case of nominal planning. However, the authors of this paper argue that the unique properties of the problem and the fact that the safe stop capability is safety critical, require algorithms and functions dedicated to this particular problem. A notable example of academic work in the area is the contingency planning algorithm presented by Salvado et al. in [8]. Salvado's algorithm is based on anytime graph search among a precomputed set of motion primitives and is capable of planning a stopping trajectory, avoiding collision in high speed scenarios. However, the algorithm does not differentiate the final states based on location.

III. Monitor Based Architecture Concept

In this section, the planning and control architecture that was developed to accommodate the safe stop trajectory planner is described. The design is based on a desired behavior conforming to [1]. A schematic of the planning and control architecture is presented in Figure 1.

A. Desired Behavior

The design considers a two step maneuver to reach the minimal risk condition. In the first step the vehicle slows down in the lane to a velocity from where it is possible to plan a complete stopping trajectory within the sensor range and trajectory planning time horizon, i.e., approximately 30 km/h. In the second step, the stopping trajectory to the safest reachable area is planned and executed. If no area outside of active lanes is reachable, the vehicle may stop in the lane, or continue at reduced speed until such an area becomes available, depending on the severity of the detected fault.

B. Architecture Design

The motion planning and control functionality for automated vehicles is generally divided into several subfunctions corresponding to hierarchical levels [3], [4]. The higher levels typically have a longer planning horizon and coarser representation of the ego vehicle and environment than those of the lower levels, in order to facilitate computation. In the architecture presented in Figure 1, the nominal planning and control problem is subdivided into four hierarchical functions, i.e., supervisory controller, path planner, trajectory planner and controller. The supervisory controller keeps track of the current mission and sends commands to the path planner in the form of goal locations. The path planning function solves
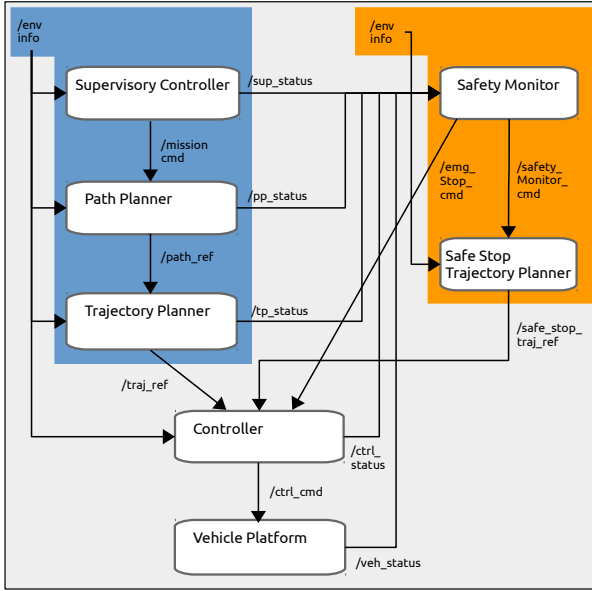
Fig. 1: A safety monitoring architecture concept for L4 automated driving. The nominal channel (blue) for motion planning consists of a supervisory controller, a path planner and a trajectory planner. The safety channel (orange) consists of a safety monitor and a safe stop trajectory planner

the problem of finding a traversable spatial path between an initial pose $\mathbf{p}_{init} = [X_0 \ Y_0 \ \psi_0]$ and a desired end pose $\mathbf{p}_{goal} = [X_{goal} \ Y_{goal} \ \psi_{goal}]$. X, Y and ψ denote position and heading in a Cartesian coordinate system. In the case of structured road environment, the solution is extracted from the map as for example the center of a lane. The trajectory planning function solves the problem of finding a reference state trajectory along the path, e.g., $\mathbf{x}(t) = [X(t) \ Y(t) \ \psi(t) \ v(t)]^T$, where $v(t)$ is the reference velocity, $t \in [0 \ t_h]$ and t_h is a finite planning horizon. The trajectory reference avoids obstacles and has a bounded deviation from the path. The controller computes steering and torque input, such that the reference trajectory is followed with a bounded tracking error.

In addition to the nominal functionality channel for planning and control, the architecture has a safety channel, including a safety monitor function with the purpose of detecting errors and issuing appropriate commands to mitigate hazardous situations. Monitoring based system architecture is a fault tolerance technique that is commonly used for ensuring safety of autonomous systems [14]. It is also referred in literature as *checker* [15] or *safety monitor* [16]. The idea is to dedicate a separated high integrity component to safety only. The monitor is responsible for keeping the system in a safe state if a hazardous event or behavior is detected, by performing actions according to safety rules. The rules are defined at design time based on safety analysis. A benefit of using a monitoring architecture is that it allows the nominal channel to use methods that are difficult/impossible to verify (e.g machine learning) while the system as a whole remains safe, with the prerequisite that erroneous behavior can be detected and handled. Introducing a safety monitor affects the design specification of several functions of the architecture. The design should be such that certain properties may be reliably monitored and the safety interventions/actions are dependable and independent of the nominal functionality. To ensure high availability of the safe

stop capability, the safety channel should be independent of the nominal one. The assumption is made that environment information for the safety channel is obtained independent of the nominal perception functionality.

In this work, three intervention actions are considered, i.e. *A: slow down in lane*, *B: safe stop* and *C: emergency stop*. Reference trajectories for realizing actions *A* and *B* are computed by the safe stop trajectory planner. If emergency stop is selected, the command is sent directly to the controller, which stops the vehicle as fast as possible. The remaining sections of this paper formulate and solve the trajectory planning problem associated with safe stop.

IV. FORMULATION OF SAFE STOP OPTIMAL CONTROL PROBLEM

This section presents an OCP formulation for trajectory planning that captures the desired behavior for safe stop according to [1]. Solutions to the OCP may serve as reference trajectories from an arbitrary initial state to a minimal risk condition.

A. Definition of Optimality

A vehicle is at a minimal risk condition if it is fully stopped and it is preferable to stop outside of active traffic lanes [1], [2]. Thus, a trajectory that terminates outside of an active lane should be ranked higher, i.e., have lower cost than one that terminates inside an active lane. It is also noted that different areas in which the vehicle can stop outside of active lanes may have attributes that make some of them more desirable than the others. For example to stop in a designated parking spot at the side of the road is preferable to stopping on the hard shoulder of the road. To capture this property, a safe stop cost associated with predefined areas in a map, is introduced.

The *optimal trajectory* is defined as the one that takes the vehicle to a stopped state in the lowest cost safe stop area, at the lowest average velocity, while avoiding collision with static and dynamic objects. Safe stop areas to be considered are those that are reachable with respect to vehicle dynamics and a fixed time horizon.

B. Assumptions

Assume that safe stop areas can be found and encoded in a map ahead of runtime and that a classification can be done within these areas with respect to how safe they are. The ranking process should be done based on analysis of the traffic system. Furthermore, it is assumed that a safety monitor that can detect internal and external hazards exists and a function that can reduce ego vehicle's speed to a value less than $v_{max} = 30km/h$ in the current lane is executed before the safe stop process if needed. Assume that static and dynamic obstacles can be detected. Passenger comfort limits for acceleration and jerk at $a_{max} = 2.0 \ m/s^2$ and $j_{max} = 1.0 \ m/s^3$ are considered.

C. Safe Stop Optimal Control Problem

The safe stop optimal control problem is formulated as

$$\min_{\mathbf{u}(t)} \quad \frac{1}{t_f} \int_0^{t_f} \frac{|v(t)|}{v_{max}} \, dt + S(O(\mathbf{x}(t_f)))$$

$$\text{s.t.} \quad \begin{aligned} &\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)), \\ &\mathbf{x}(0) = \mathbf{x}_{init}, \\ &v(t_f) = 0, \\ &g(\mathbf{x}(t), \mathbf{u}(t)) \leq 0, \\ &O(\mathbf{x}(t)) \in \mathcal{W}_{S,free}(t), \end{aligned} \qquad (1)$$

where $\mathbf{x} \in \mathbb{R}^n$, is the state vector, $\mathbf{u} \in \mathbb{R}^m$ is the vector of control inputs, v is the vehicle velocity, v_{max} is the maximum initial velocity. n and m are the numbers of states and control inputs respectively. $S : \mathbb{R}^2 \to \mathbb{Z}$ is a function that outputs the safe stop cost based on the occupancy of the final state, i.e., the area occupied by the vehicle when it has come to a complete stop. The vehicle dynamics is encoded in the function f, while g denotes the dynamic constraints including limits in curvature, acceleration, jerk, etc. As per the naming convention of [7], $O : \mathbb{R}^n \to \mathbb{R}^2$ denotes the occupancy of the vehicle at a given state $\mathbf{x}(t)$ and $\mathcal{W}_{S,free}(t) \subset \mathbb{R}^2$ the drivable area, i.e, the area not occupied by static or dynamic obstacles. S is introduced as a soft constraint in the cost function rather than as a hard constraint in order to reflect [1] in that stopping outside of a vehicle lane is preferred, but not explicitly required. S takes the value 0 for the highest ranked safe stop area, 1 for the second highest, etc. The integral term of the cost function always has a value between 0 and 1. Thus, assuming that $v_{init} \neq 0$, a trajectory that terminates in a safe stop area will always have priority over one that stops quickly in an active lane.

Solution trajectories to the safe stop OCP, were computed using the DPM dynamic programming OCP solver [17]. The dynamic programming approach was chosen since it can accommodate a wide range of constraints and cost function elements.

V. SAFE STOP TRAJECTORY PLANNER

The safe stop OCP, in Equation (1), is nonlinear and not always convex. Thus, it is problematic to solve it in real time with a detailed vehicle model. The trajectory library approach is suitable for the safe stop maneuver since it can accommodate a detailed vehicle model without affecting run-rime performance, and since the goal velocity is always zero. This property eliminates the need to discretize final velocity, greatly reducing the cardinality of the set. This mitigates to some extent the problem of infeasibly large sets mentioned in Section II-B. In principle, an arbitrarily detailed vehicle model can be used with the method, provided it can be represented as a state space model. A highly detailed model will impact set generation computational requirements, while real-time performance of the algorithm will remain unaffected. For our purposes, the following continuous time vehicle model was used,

$$\begin{aligned} \dot{X}(t) &= v(t)\cos(\psi(t)), \\ \dot{Y}(t) &= v(t)\sin(\psi(t)), \\ \dot{\psi}(t) &= v(t)\kappa(t), \\ \dot{v}(t) &= a(t), \\ \dot{a}(t) &= j(t), \\ \dot{\kappa}(t) &= \lambda(t). \end{aligned} \qquad (2)$$

where $\mathbf{x}(t) = [X(t) \; Y(t) \; \Psi(t) \; v(t) \; a(t) \; \kappa(t)]^T$ is the state vector, including X-position, Y-position, heading, velocity,

acceleration and curvature and $\mathbf{u}(t) = [j(t) \; \lambda(t)]$ is the control vector, including longitudinal jerk and curvature rate.

To compute the trajectory set, a grid of candidate final positions $\mathbf{P}_{goal} = [X_{goal} \; Y_{goal}]^T$ and initial velocities v_{init} is defined. For each pair of initial states $\mathbf{x}_{init} = [0 \; 0 \; 0 \; v_{init} \; 0 \; 0]^T$ and final states $\mathbf{x}_{goal} = [X_{goal} \; Y_{goal} \; 0 \; 0 \; 0 \; 0]^T$ in the grid, an optimal control problem formulated as

$$\min_{\mathbf{u}(t)} \quad \int_0^{t_f} ||\mathbf{P}_{goal} - \mathbf{P}(t)||_2^2 dt$$

$$\text{s.t.} \quad \begin{aligned} &\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)), \\ &\mathbf{x}(0) = \mathbf{x}_{init}, \\ &\mathbf{x}(t_f) = \mathbf{x}_{goal}, \\ &g(\mathbf{x}(t), \mathbf{u}(t)) \leq 0 \end{aligned} \qquad (3)$$

is solved. Notation is the same as in Equation (1). The cost function is chosen such that Euclidean distance between the current position $\mathbf{P}(t)$ and the goal position $\mathbf{P}_{goal}$ is minimized. The ACADO toolkit [18] is used for solving the set of optimal control problems. If a feasible solution exists for a given pair of initial an goal states, the resulting state trajectory is added to the set. After the set has been generated, a post-processing step is conducted where the set is indexed and runtime relevant parameters such as maximum jerk of the trajectory, j_{max} and the time until the vehicle reaches a stopped state, t_s.

At runtime, the subset of trajectories associated to the current velocity is selected. All trajectories in the subset are checked for collisions in an occupancy grid [19]. The cost function from Equation (1) is evaluated for all collision free trajectories, and the lowest cost trajectory is selected for execution.

VI. RESULTS AND DISCUSSION

To verify that the safe stop trajectory planning algorithm produces the intended behavior, comparisons in three scenarios are made with solutions to the safe stop OCP. The resulting trajectories, corresponding velocity profiles and associated costs are shown in Figures 2 and Table I, respectively. The three scenarios have different configurations in terms of safe stop areas, while the initial velocity of the ego vehicle is 30km/h for all scenarios. Both methods satisfy all constraints and produce the desired behavior. However, the DPM solution consistently reaches a full stop in a shorter time and demonstrates more aggressive lateral motion. This is attributed to the fact that the DPM OCP solver is limited to five state dynamic models. Therefore, no dynamic constraints were applied to $\dot{a}$ and $\dot{\kappa}$, resulting in sharper turning and braking in the DPM solution compared to the solutions from the safe stop trajectory planner. Since the desired behavior is to stop as soon as possible the DPM solutions typically have lower cost, as is evident from Table I.

To evaluate the usefulness of the safe stop trajectory planner in a realistic context, it was integrated in the software stack of the Research Concept Vehicle (RCV), [20], [21], hosted by the Integrated Transport Research Lab (ITRL) at KTH. A modified instance of the Gazebo based `car_demo` simulation environment [22] from the Open Source Robotics Foundation is used to represent a vehicle and a traffic environment. No dynamic obstacles were considered in these initial experiments. Figure 3 shows a safe stop executed in simulation. The safe
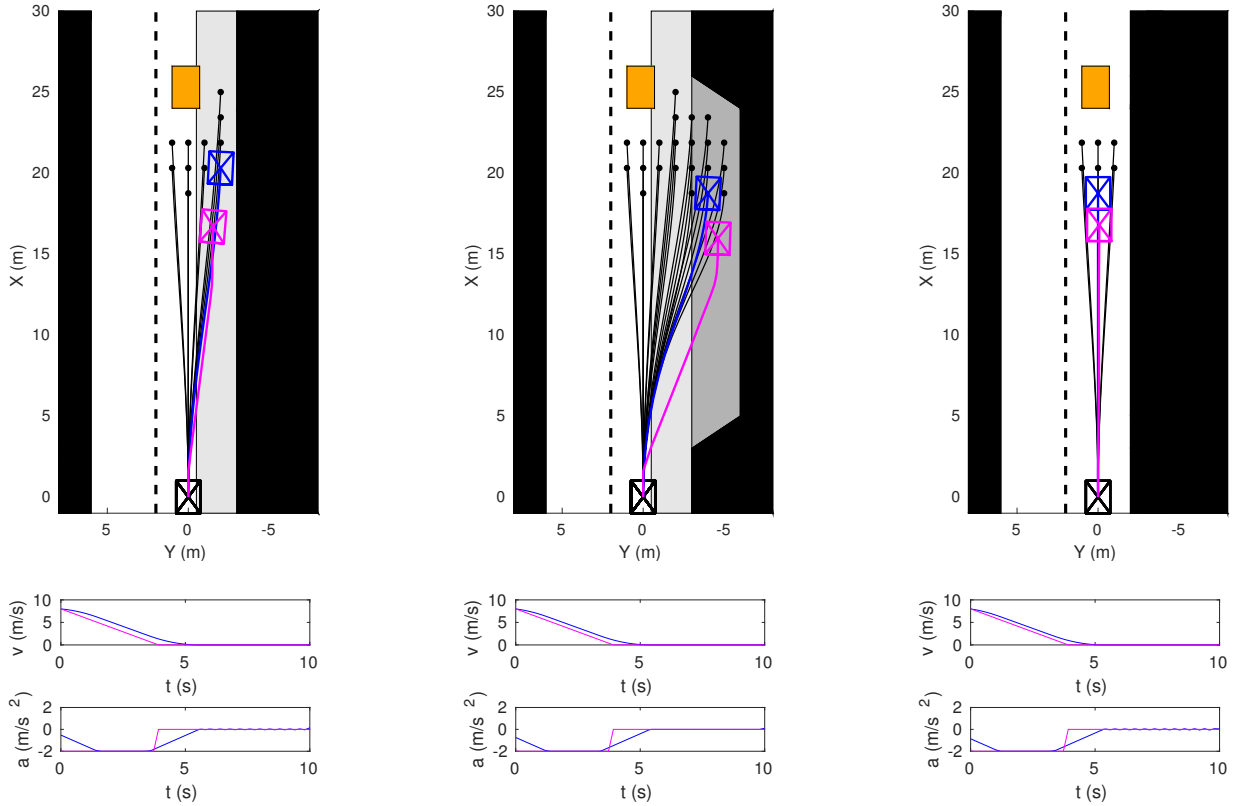
(a) Scenario 1: Shoulder of the road is available for safe stop

(b) Scenario 2: Shoulder of the road and roadside parking is available for safe stop

(c) Scenario 3: No safe stop area is available

Fig. 2: Comparison of resulting trajectories of the safe stop trajectory planner (blue) and the optimal trajectory computed using the DPM solver (magenta). Dark gray areas denote lower safe stop cost, light gray area denotes higher safe stop cost. Orange boxes represent vehicles. Black lines show collision free trajectory candidates for the safe stop trajectory planner. The safe stop trajectory planning algorithm demonstrates the same general behavior as solutions to the safe stop OCP.

TABLE I: Costs of trajectories

	Scenario 1	Scenario 2	Scenario 3
DPM	1.201	0.201	2.201
SSTP	1.257	0.243	2.237

stop trajectory planning algorithm is executed according to the description in Section V, in parallel with the nominal trajectory planner. Figure 3a shows a snapshot of the currently reachable safe stop poses during nominal operation. Green areas denote the lowest safe stop cost and red the highest. Figures 3b, 3c and 3d show snapshots of the simulated vehicle, nominal and safe stop trajectories through the stages of a safe stop maneuver. The velocity, curvature, longitudinal and lateral accelerations of the vehicle throughout the maneuver are shown in Figure 3e. The algorithm exploits the properties of the stopping maneuver in that all the precomputed trajectories have zero velocity at the final state. This significantly reduces the size of the set, and thus to some extent the problem of the trajectory library approach mentioned in Section II-B. However, there is still a trade-off between execution time and number of trajectories in the set. For this prototype implementation, the distance between candidate end poses is 1.5m in the longitudinal and 1.0m lateral dimension. The longest runtime for the example scenario is 60.0 ms, corresponding to a maximum re-planning frequency of 16.7 Hz, which is comparable to state of the art methods in nominal trajectory planning. Improving the efficiency of the implementation and/or utilizing parallel com-

puting, would enable a finer grid of candidate end poses, while maintaining adequate re-planning rate.

VII. CONCLUSION AND FUTURE WORK

To quantitatively evaluate reference trajectories for safe stop maneuvers, a benchmark optimal control problem is proposed. Our initial results show that solution trajectories from the safe stop OCP, as well as from our proposed safe stop trajectory planning algorithm, correspond to the desired safe stop behavior for L4 autonomy as described in [1]. Limitations of the DPM solver to five states result in limited accuracy with respect to vehicle dynamics. However, the solution is useful as an initial benchmark, since it captures the general behavior of solutions to the safe stop OCP. Investigating alternative solution methods to improve the benchmark is planned for future work.

The safe stop trajectory planner replicates the behavior of the base OCP solution in that it can distinguish different areas for safe stop, differentiating our algorithm from prevous works in the field. Utilizing the properties of the safe stop problem yields adequate runtime performance of the algorithm, even though the implementation is still at a prototype stage. Next steps include investigation of alternative solution methods such as model predictive control and online set generation using quintic polynomials, evaluation in more scenarios and experiments on the RCV.

(a) $t1 : 4.48$

(b) $t2 : 11.74$

(c) $t3 : 13.73$

(d) $t4 : 16.45$
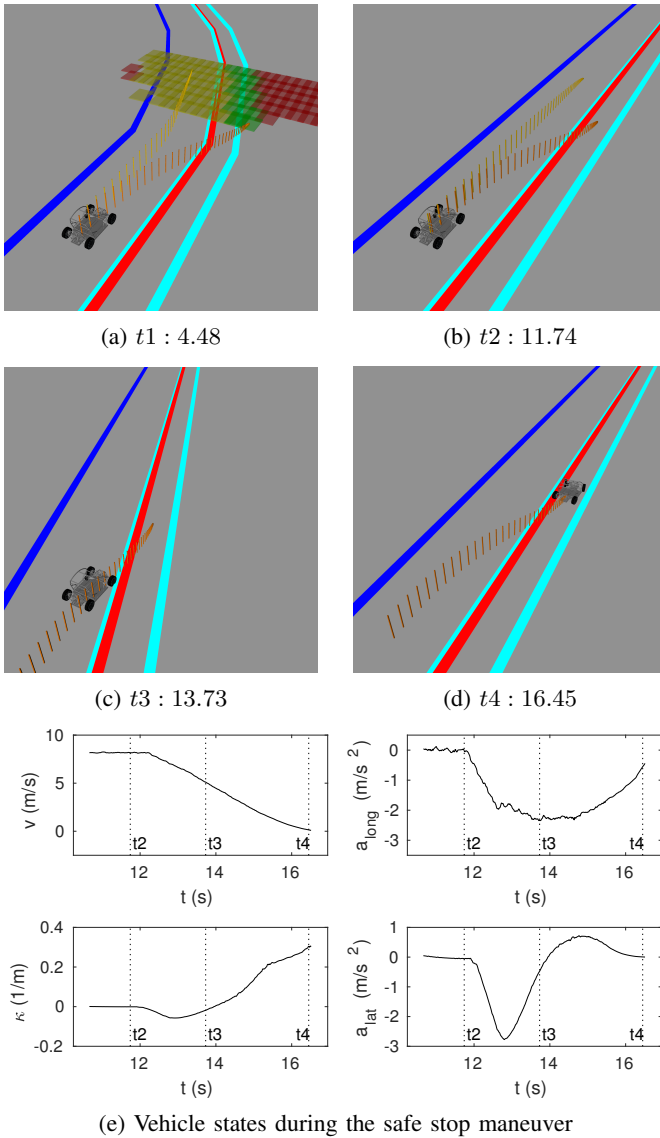
(e) Vehicle states during the safe stop maneuver

Fig. 3: A simulated safe stop of the RCV. In 3a, the currently reachable safe stop poses are shown in green, yellow and red, corresponding to decreasing safe stop scores. 3b through 3d show snapshots from the safe stop maneuver, the times of which are indicated in 3e. The nominal (yellow) and safe stop (orange) reference trajectories are visualized with a sequence of bars, where the height represent the reference velocity

REFERENCES

[1] SAE, *J3016 Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, sep 2016.

[2] (2017) Waymo safety report, on the road to fully self driving. [Online]. Available: https://waymo.com/safetyreport/

[3] B. Paden, M. p, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, March 2016.

[4] D. Gonzalez, J. Prez, V. Milans, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, April 2016.

[5] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller, E. Kaus, R. G. Herrtwich, C. Rabe, D. Pfeiffer, F. Lindner, F. Stein, F. Erbs, M. Enzweiler, C. Knoppel, J. Hipp, M. Haueis, M. Trepte, C. Brenk, A. Tamke, M. Ghanaat, M. Braun, A. Joos, H. Fritz, H. Mock, M. Hein, and E. Zeeb, "Making bertha drive; an autonomous journey on a historic route," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 2, pp. 8–20, Summer 2014.

[6] M. Werling, S. Kammel, J. Ziegler, and L. Grll, "Optimal trajectories for time-critical street scenarios using discretized terminal manifolds," *The International Journal of Robotics Research*, vol. 31, no. 3, pp. 346–359, 2012. [Online]. Available: https://doi.org/10.1177/0278364911423042

[7] M. Althoff, M. Koschi, and S. Manzinger, "Commonroad: Composable benchmarks for motion planning on roads," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, June 2017, pp. 719–726.

[8] J. Salvado, L. M. M. Custdio, and D. Hess, "Contingency planning for automated vehicles," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 2853–2858.

[9] ISO, "ISO 26262 – Road vehicles – Functional Safety," 2011.

[10] U. Rosolia, S. D. Bruyne, and A. G. Alleyne, "Autonomous vehicle control: A nonconvex approach for obstacle avoidance," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 2, pp. 469–484, March 2017.

[11] P. Sermanet, M. Scoffier, C. Crudele, U. Muller, and Y. LeCun, "Learning maneuver dictionaries for ground robot planning," in *Proc. 39th International Symposium on Robotics (ISR08)*, 2008.

[12] C. Liu and C. G. Atkeson, "Standing balance control using a trajectory library," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2009, pp. 3031–3036.

[13] S. Arora, S. Choudhury, D. Althoff, and S. Scherer, "Emergency maneuver library - ensuring safe navigation in partially known environments," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 6431–6438.

[14] A. Avizienis, J. C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11–33, Jan. 2004.

[15] F. Py and F. Ingrand, "Dependable execution control for autonomous robots," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings*, vol. 2, Sep. 2004, pp. 1136–1141 vol.2.

[16] M. Machin, J. Guiochet, H. Waeselynck, J. P. Blanquart, M. Roy, and L. Masson, "SMOF: A Safety Monitoring Framework for Autonomous Systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. PP, no. 99, pp. 1–14, 2016.

[17] O. Sundstrom and L. Guzzella, "A generic dynamic programming matlab function," in *2009 IEEE Control Applications, (CCA) Intelligent Control, (ISIC)*, July 2009, pp. 1625–1630.

[18] B. Houska, H. Ferreau, and M. Diehl, "ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization," *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011.

[19] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278–288, Jun 1991.

[20] O. Wallmark, M. Nybacka, D. Malmquist, M. Burman, P. Wennhage, and P. Georen, "Design and implementation of an experimental research and concept demonstration vehicle," in *2014 IEEE Vehicle Power and Propulsion Conference (VPPC)*, Oct 2014, pp. 1–6.

[21] G. C. Pereira, L. Svensson, P. F. Lima, and J. Mårtensson, "Lateral model predictive control for over-actuated autonomous vehicle," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, June 2017, pp. 310–316.

[22] (2017) Open Source Robotics Foundation cardemo. [Online]. Available: https://github.com/osrf/cardemo