2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Kinodynamic Motion Planning Using Multi-Objective Optimization

Patrick Hart[1] and Alois Knoll[2]

***Abstract*— As autonomous driving gains importance, universally applicable motion planning approaches that offer safe and comfortable rides have to be developed. Most planning methods up-to-date still struggle when dealing with dynamic environments. They require extensive parameter-fine tuning in order to generate comfortable and safe solutions and it is not known prior to optimization which set of parameters would produce the "best" solution. Therefore, we introduce a multi-objective optimization that plans a set of trajectories using several weights and targets (e.g. desired velocity or lanes). Thus, reducing the need of extensive parameter fine-tuning and increasing the planner's capabilities to handle dynamic environments. Furthermore, in order to plan multiple trajectories in real-time, a *smart-initialization* of the optimization problem is introduced that speeds up the multi-objective optimization further. Due to the proposed architecture that consists of a Planning-, Evaluation- and Selection-module, the planner is capable of providing a high level of comfort and safety – even in the case of non-convergence of the optimization. The novel motion planning approach is evaluated in terms of its applicability and performance.**

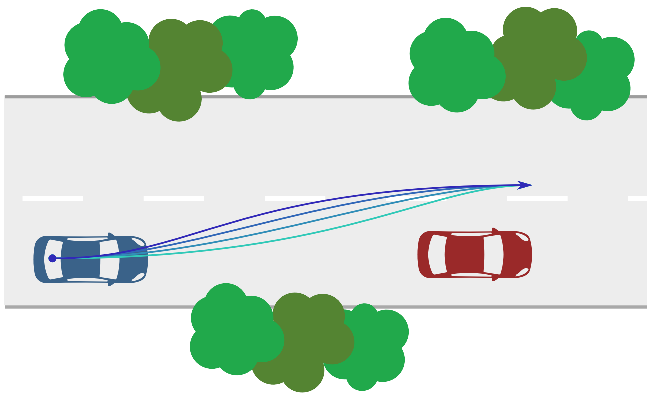


Fig. 1: Planned trajectories in order to overtake the red vehicle. The colors indicate the costs of the safety-metric ranging from bright colors being unsafe towards darker ones being safer.

I. Introduction

With autonomous driving gaining importance and the first Level-3 systems driving on public roads, methods used in these have to be refined when striving for Level-5 autonomy – fully autonomously acting agents. Not only will autonomous driving provide higher comfort and safety whilst driving, but it will also enable passengers to use their time more productively, e.g. whilst commuting to work.

Recent advances in deep learning have increased the capabilities of autonomous vehicles to detect, track and classify objects. In contrast, motion planning in highly-dynamic environments and interacting with other traffic participants mostly remains yet to be solved. This is partially due to the vast combinatorial space motion planning problems span. In order for autonomous vehicles to be accepted by the broader public, the motion planner should be able to plan trajectories that provide maximum comfort whilst being collision-free. Optimization-based methods have proven to work well in planning motions for autonomous vehicles. However, they often require parameter fine-tuning or require the planning problem to be decomposed and solved locally.

In this work, we introduce a multi-objective optimization that uses several weights and targets (e.g. reference lanes or the desired speed) in order to generate a greater flexibility and variety of solutions and to reduce the need of extensive parameter fine-tuning. Additionally, it mitigates the problem of not knowing which set of weights and targets would produce the "best" trajectory for a situation prior to optimization. This is illustrated in Figure 1 where the vehicle has to perform a lane-change in order to avoid the other vehicle. In order to plan multiple trajectories in real-time, a *smart-initialization* is used that further speeds up the optimization. This is achieved by using results of prior optimizations that are similar as initial estimates.

Additionally, a novel motion planning architecture that separates the Planning- (optimization), Evaluation- and the Selection-module is introduced. Since these modules are independent of each other, the Evaluation- and Selection-module are capable of rating and selecting trajectories independent of the objective-function used in the optimization. The Selection-module can incorporate external weights in order to execute higher-level strategies, such as staying on a lane versus performing a lane-change.

To sum up, the novel approach offers a multitude of benefits in motion planning for autonomous driving:

- Is able to handle a wide variety of situations due to the use of multiple weights and targets in the optimization.
- Introduces a novel planning architecture that ensures a high level of comfort and safety and that is able to include a higher-level strategy.
- Produces kinodynamic and jerk-optimal trajectories by using a model-based optimization approach.

This work is further organized as follows: Section II gives a brief literature overview of optimization-based and multi-objective optimization used in motion planning for au-

[1]Patrick Hart is with the fortiss GmbH, An-Institut Technische Universität München, Munich, Germany. Email: patrick.hart@tum.de

[2]Alois Knoll is with Robotics and Embedded Systems, Technische Universität München, Munich, Germany

tonomous driving. The proposed motion planning approach is presented in Section III. Section IV will give brief details about the implementation of the novel planning approach and in Section V, the planner is evaluated in terms of its feasibility and performance. Finally, in Section VI, a conclusion and outlook is provided.

II. Related Work

Several methods have been applied in motion planning for autonomous driving. This section provides a brief overview of optimization-based methods for motion planning in autonomous driving as well as a brief introduction to multi-objective optimization.

A. Optimization-Based Motion Planning

Optimization-based methods for motion planning have proven to work well in autonomous driving. With only these being able to generate truly optimal trajectories in respect to a pre-defined objective, these are prime candidates for motion planning in autonomous driving. Analytical solutions in optimization are often not possible since the problems are of too complex nature [1]. Therefore, direct methods are utilized in order to solve motion planning problems more efficiently [2]. These can generally be categorized into geometric- and model-based optimization approaches.

Ziegler *et al.* introduced a geometric optimization approach which has been developed within the Bertha-Benz memorial drive project [3]. This approach has proven to work well in mixed-traffic and successfully drove the complete historic Bertha-Benz route without any human intervention. However, geometric approaches neglect the kinematics and dynamics of vehicles and these have to be integrated indirectly using boundary conditions. Contrary to this, the novel planning approach will use a model-based optimization in order to intrinsically incorporate the vehicle kinematics.

Werling *et al.* use a model-based optimization approach utilizing the Frenet Coordinate System (FCOS). They transform the planning problem into an arclength-based coordinate system which makes the planning problem easier to solve [4]. However, the vehicle model has to be modelled in the FCOS and each point of the trajectory has to be transformed back to world coordinates in oder to check the points for feasibility and collisions which requires high computational effort [5].

Further motion planning methods combine optimization-based techniques with machine learning approaches, e.g. as demonstrated by Borrelli *et al.* [6]. In their work they introduced an optimization based approach which is able to "learn" in order to perform an inter-lap optimization and to achieve faster lap-times on a race track over time. Their approach could potentially be sped up by using a multi-objective optimization as the planner might be capable of learning faster.

B. Multi-Objective Optimization

Multi-objective optimization is used in a variety of engineering applications. Especially, in fields that have multiple correlated or opposing parameters and targets, such as optimizing financial portfolios or mechanic structures.

Pavone *et al.* implemented a real-time stochastic kinodynamic motion planning method which utilizes a multi-objective search called PUMP (Parallel Uncertainty-aware Multiobjective Planning) algorithm [7]. This algorithm is search-based and explores the state space in a parallel and discrete fashion. In their work, they plan trajectories for quadrocopters in indoor environments.

Zhang *et al.* use a multi-objective optimization based on swarm optimization resulting in a set of trajectories which are then evaluated by a metric [8].

In our work, a discrete and model-based approach will be introduced that uses a muli-objective optimization in order to deal with multiple weights and targets. By obtaining multiple resulting trajectories, the novel planner has a greater flexibility than conventional approaches do.

III. Kinodynamic Multi-Objective Motion Planning

This section introduces the novel planning approach. First, the overall architecture of the planner that enables a safe and comfortable behaviour of the vehicle – even in the case of non-convergence – is outlined. Next, the prerequisites (kinematic model and the objective function) for understanding the modules outlined in the architecture are introduced. Finally, the Multi-Objective Optimization-, the Evaluation- and the Selection-module of the architecture are described in greater detail, respectively.

A. Architecture of the Planner

The motion planning architecture consists of three modules in order to provide a high level of safety and comfort. The first module, the Multi-Objective Optimization, generates a set of feasible trajectories using pre-defined sets of weights and targets (e.g. the reference lane or desired speed). The second module, the Evaluation, then rates these obtained trajectories in respect to several in this work defined metrics. This enables an independent of the objective-function evaluation of the trajectories. The third module, the Selection, then uses the costs determined by the Evaluation-module and selects the "best" trajectory for a given situation. Therefore, the second and third module can be seen as an external "judge" that rates the trajectories and selects the best trajectory. Furthermore, external weights can be passed into the Selection-module in order to execute higher-level strategies, such as favoring lane-changes to staying on the current lane. In Figure 2, the proposed architecture is outlined in a block-diagram.

The architecture is not restricted to applications using optimization-based approaches, but can also be used for search- or interpolation-based motion planning approaches.

B. Kinematic Vehicle Model

In order for the vehicle to be able to execute the planned motion and to include its non-holonomic characteristics, a model-based optimization is used. In this work, an extended

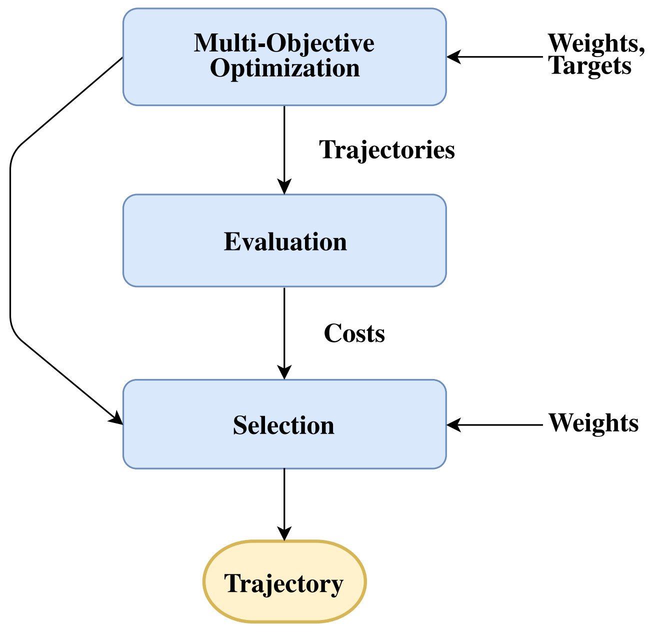


Fig. 2: Block-diagram of the motion planning architecture. The planning architecture is split into three modules that plan, evaluate and select trajectories.

single-track vehicle model is used as introduced in [9]. It can be mathematically denoted in the state-space as

$$F(\underline{x}(t), \underline{u}(t)) = \begin{bmatrix} v\cos(\theta) \\ v\sin(\theta) \\ v\frac{\tan(\delta)}{l} \\ a \end{bmatrix} \quad (1)$$

with the state $\underline{x} = [x, y, \theta, v]$ and the input vector $\underline{u} = [\delta, a]$ consisting of the steering-angle and the acceleration. In order to obtain a trajectory that can be optimized, this model is forward-simulated using the explicit Euler's method over a time-horizon T. The optimizer then gradually changes the system inputs in order to obtain an optimal trajectory in an iterative fashion. The choice of the vehicle model embodies an important decision as it defines the planner's limitations – whether it is capable of handling highly dynamic situations or not.

C. Objective Function

The objective function is an important choice, that mainly determines the outcome of the optimization and whether it converges. It should provide a high level of safety whilst also enabling a comfortable ride. The objective used in the optimization is three-fold and is defined as

$$\begin{aligned} \mathrm{J}(\mathcal{T}, \mathbf{W}_i, \mathbf{T}_i) = \mathrm{J}_{\mathrm{Safe}}(\mathcal{T}, \mathbf{W}_\mathrm{i}) + \mathrm{J}_{\mathrm{Comf}}(\mathcal{T}, \mathbf{W}_\mathrm{i}) + \quad (2) \\ \mathrm{J}_{\mathrm{Targets}}(\mathcal{T}, \mathbf{T}_\mathrm{i}). \quad (3) \end{aligned}$$

with $\mathcal{T} = [\underline{x}_0, \ldots, \underline{x}_\mathrm{N}]$ representing the state-trajectory. Moreover, by injecting the weight matrices $\mathbf{W}_i$ and $\mathbf{T}_i$ into Equation (3), the differential values of the trajectory as well as the safety terms are weighted. The first term $\mathrm{J}_{\mathrm{Safe}}$ penalizes proximity to other objects and incorporates boundary conditions imposed onto the optimization-problem. Furthermore, to ensure a continuous gradient, we use a pseudo-distance function as introduced in [10]. The second term $\mathrm{J}_{\mathrm{Comf}}$ rates the differential values of the trajectory to provide a high level of comfort. In order to obtain the derivatives of a trajectory, a finite forward differentiation method as shown in Appendix A is used. By using higher-order derivatives it can be ensured that the resulting trajectories are $C2$-continuous and, thus, jerk-optimal. The last term $\mathrm{J}_{\mathrm{Targets}}$ penalizes any deviation from the targets, such as the offset to the reference lane or desired velocity. The last two terms in the objective function generate a comfortable motion whereas the first term $\mathrm{J}_{\mathrm{Safe}}$ ensures the safety of the planned trajectory. By injecting several sets of weights and targets into Equation (3), multiple objective function are generated. This is described in greater detail in the next section.

D. Multi-Objective Optimization-Module

In order to plan a trajectory, a non-linear model- and optimization-based motion planner is used in this work. A multi-objective optimization motion planning problem can mathematically be denoted as

$$\begin{aligned} \text{minimize} \quad & (\mathrm{J}_1(\cdot), \mathrm{J}_2(\cdot), \ldots, \mathrm{J}_k(\cdot)) && (4) \\ \text{subject to} \quad & \dot{\underline{x}} = F(\underline{x}(t), \underline{u}(t)) && (5) \\ & \underline{x}_\mathrm{k} \in \mathcal{X} && (6) \\ & \mathrm{k} = 1, \ldots, N && (7) \\ & \underline{u}_\mathrm{k} \in \mathcal{U} && (8) \\ & \mathrm{k} = 1, \ldots, N && (9) \end{aligned}$$

with $\mathrm{J}_\mathrm{i}(\cdot)$ representing a objective function, $\dot{\underline{x}} = F(\underline{x}(t), \underline{u}(t))$ the kinematic system and $\mathcal{U}$ and $\mathcal{X}$ are polytopes that define feasible regions of the optimization problem. Due to the vehicle model used in the optimization, the resulting trajectories are kinodynamic and can directly be executed by the vehicle. In order to generate several objectives, sets of weights $[\mathbf{W_1}, \ldots, \mathbf{W_R}]$ and targets $[\mathbf{T_1}, \ldots, \mathbf{T_M}]$ are injected into Equation (3) and, thus, a resulting set of objective functions $[\mathrm{J}_1(\cdot), \ldots, \mathrm{J}_{RxM}(\cdot)]$ is obtained.

By changing the weights and targets gradually in one direction, e.g. penalizing the jerk more and more, and taking previous results as initial estimates, the multi-objective optimization is sped up and a larger set of trajectories can be planned in real-time. Therefore, the time-to-convergence is reduced significantly – this is then referred to as *smart-initialization*. In order to solve the multi-objective optimization problem, a non-linear solver is used. The implementation and simulation environment is explained in greater detail in Section IV.

E. Evaluation-Module

The Evaluation-module defines a set of metrics and rates trajectories obtained from the Multi-Objective Optimization-module. By defining these metrics independent of the objective function, an additional layer of safety is introduced.

The two foremost important criteria in autonomous driving are safety and comfort. Thus, we will introduce metrics rating these and, additionally, have a metric that penalizes any deviation of the desired targets. In this work, the safety metric is distance-based and penalizes the proximity to any polygonal shaped object and is defined as

$$\mathrm{J}_{\mathrm{Safety}}(\mathcal{T}) = \min \sum_{\mathrm{i}=0}^{\mathrm{O}} \sum_{\mathrm{j}=0}^{\mathrm{N}} \delta_{\mathrm{eucl}}(\underline{o}_{\mathrm{ij}}, \underline{x}_{\mathrm{j}}) \tag{10}$$

with $\mathcal{T} = [\underline{x}_0, \dots, \underline{x}_\mathrm{N}]$ being the state-trajectory of the ego-vehicle and O denoting time dependent objects. The Euclidean distance $\delta_{\mathrm{eucl}}(\cdot)$ is used in order to calculate the distance between the polygonal shapes.

In order to provide a comfortable ride for passengers, the trajectories should be jerk-optimal. In order to obtain jerk-optimal trajectories, the integral of the derivatives up to the third degree is used. The comfort metric is defined as

$$\mathrm{J}_{\mathrm{Comfort}}(\mathcal{T}) = \int_{t_0}^{t_0+T} L(\dot{\mathrm{x}}, \ddot{\mathrm{x}}, \dddot{\mathrm{x}})dt \tag{11}$$

with L penalizing the derivatives of the trajectory $\mathcal{T}$. The metric $\mathrm{J}_{\mathrm{Targets}}$ penalizes any deviation to a defined target, such as the offset to a reference-curve $\Gamma(t)$ or to a desired velocity v_{des}. This metric is defined as the squared sum of deviations. Further metrics could be introduced, e.g. one rating the energy consumption denoted by $\mathrm{J}_{\mathrm{Energy}}$.

F. Selection-Module

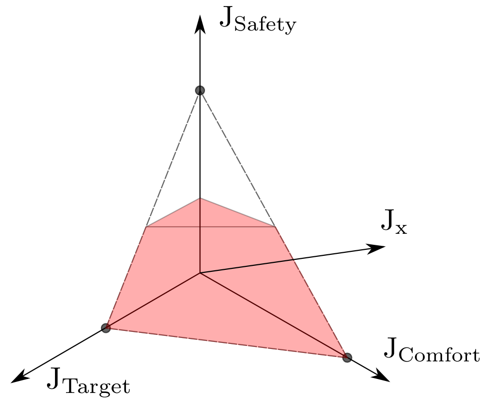
The Selection-module chooses a trajectory based on the set of optimized trajectories and costs provided by the Evaluation-module and additional weights that are passed in by an external strategy module. Therefore, for example, a lane-change can be favored versus staying on the current lane and a higher-level strategy can be incorporated. Furthermore, due to the independent evaluation of the optimized trajectories, the Selection-module can be seen as an external "judge" that is capable of selecting a trajectory independent of the objective function. By defining viable regions on the metric axes, a safe and comfortable behavior of the vehicle can be guaranteed. For example, in order to provide a safe behavior of the vehicle, a minimum safety threshold can be defined on the safety-metric axis. This is visualized in Figure 3, where the red-shaded area illustrates the infeasible region. The selection cost result of a linear combination and can mathematically be expressed as

$$\mathrm{J}_{\mathrm{Total}} = w_0 \mathrm{J}_{\mathrm{Comfort}} + w_1 \mathrm{J}_{\mathrm{Safety}} + \cdots + w_{\mathrm{R}} \mathrm{J}_{\mathrm{R}} \tag{12}$$

with $[w_1, \dots, w_R]$ being the set of weights that is passed into the Evaluation-module from an external strategy module and J_{Total} are the resulting total costs used for selecting the best trajectory.

IV. Implementation

The optimization-based motion planning method has been implemented using C++ and state-of-the-art libraries, such as the ceres-solver [11]. For better accessibility, the C++ code



$\mathrm{J}_{\mathrm{Safety}}$
J_{x}
$\mathrm{J}_{\mathrm{Comfort}}$
$\mathrm{J}_{\mathrm{Target}}$

Fig. 3: Evaluation metrics spanning a high-dimensional space. The red-shaded area indicates the infeasible region in respect to the safety metric.

has been wrapped using the Pybind11 library [12]. Based on this, a framework and simulation environment has been developed in Python that can read and simulate scenarios from XML-files. The performance of the optimization could further be improved by implementing more efficient metrics, so that the iterative process in the optimization is sped-up. Moreover, by using a C++ only implementation, the performance could also be improved. All experiments and results presented in this work have been generated using Ubuntu with an i7-6700HQ-processor of the 6th-generation.

V. Experiments and Evaluation

In order to evaluate the performance and applicability of the novel planning approach, a scenario to avoid an obstacle is chosen that results in a lane-change. At first, the weights and targets are varied, obtaining a set of trajectories, each possessing a unique point in the evaluation metric space. It will be shown, that the choice of weights and targets is of high importance for the level of safety and comfort. Next, the benefits of using a *smart-initialization* and the resulting increase in performance when optimizing several weights and targets are shown.

A. Varying Weights and Targets

In this section, the benefits of using several weights and targets are highlighted. At first, a single target and multiple weights are planned. The results in Figure 4 on the left-hand side show the importance of the choice of weights. Slight modifications in the set of the weights already lead to large differences in the resulting set of trajectories. As shown by the gray trajectories in Figure 4, not all trajectories are feasible and might be too critical when evaluated in respect to a safety metric. Thus, conventional approaches optimizing only a single objective function might struggle to find a feasible solution. With the novel motion planning approach, the chances of finding a suitable trajectory are therefore increased.

On the right-hand side in Figure 4 the results of varying weights and targets are shown. In this experiment, the

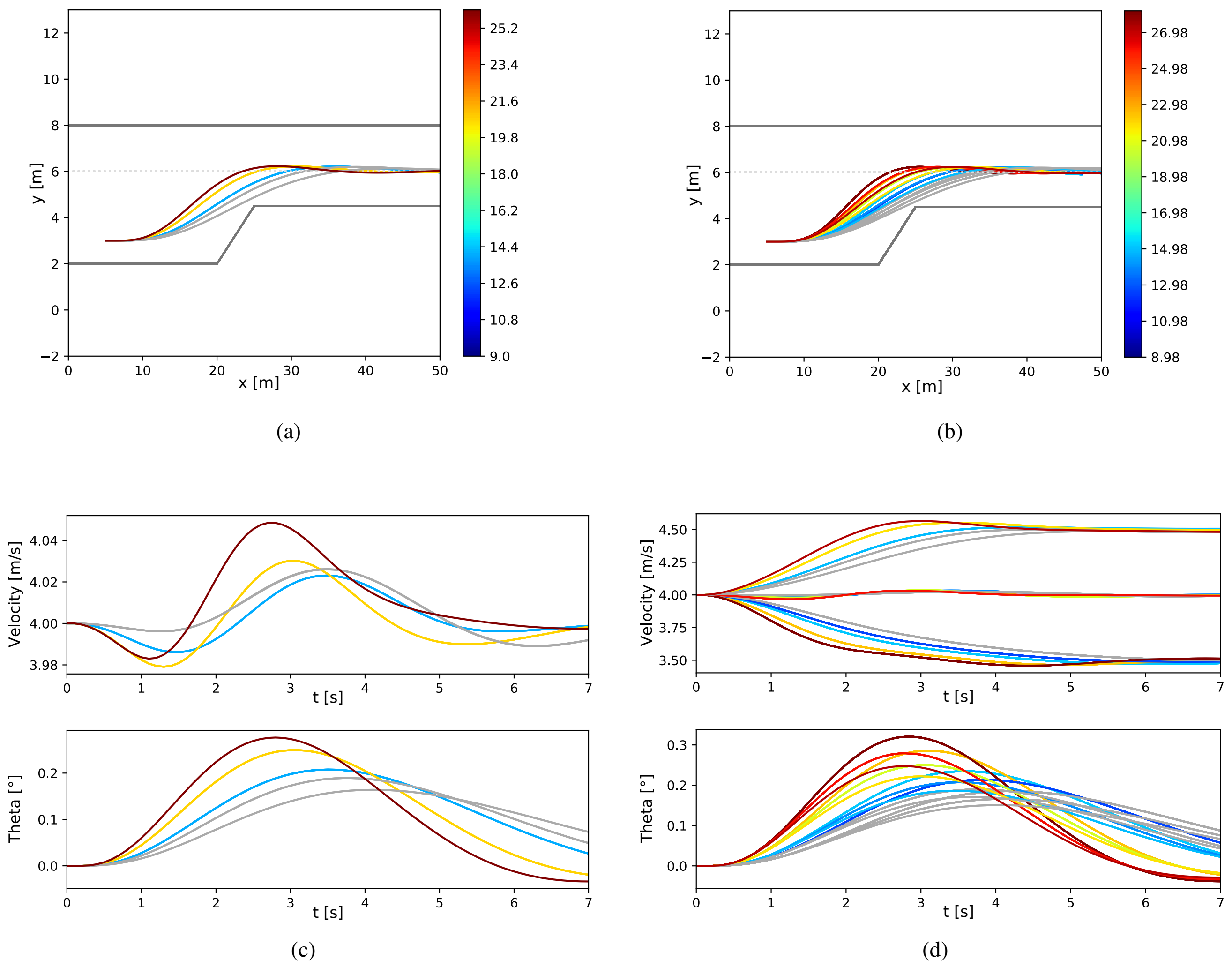


Fig. 4: (a), (b) Cartesian plot of the planned trajectories. The boundaries of the optimization problem are displayed as the solid gray lines whereas the dotted one indicates the desired reference lane. The color-coding in this plot shows the cost of $\mathrm{J}_{\mathrm{Total}}$ of the Selection-module. The gray trajectories are infeasible due to their proximity to the obstacle. (c), (d) show the velocity and vehicle angle θ plotted over the time-horizon. The color-coding is similar to figures (a) and (b).

desired velocity is changed additionally to the weights. By varying the weights and targets, the amount of trajectories rises quadratically. However, by using a *smart-initialization*, the multi-objective optimization can be solved significantly faster and the computational-time does not rise quadratically as will be shown in the next sub-section. By changing both, the variety of solutions is further increased. Due to this, the novel planning approach trends towards universally applicability and is able to handle a larger variety of situations without the need of extensive parameter fine-tuning whilst still being real-time capable. The results shown in Figure 4 are generated using a time-horizon of $T = 10s$ and a step-time of $\Delta t = 0.15s$. The colors indicate the costs $\mathrm{J}_{\mathrm{Total}}$ of the Selection-module and gray indicates that a trajectory is within the infeasible region.

B. *Performance*

This section evaluates the performance and speed-gain due to the use of *smart-initialization* in the novel planning approach. Table I shows the performance for optimizing a single trajectory over several time-horizons T and step-sizes Δt. In time-critical situations, only a limited amount of trajectories can be planned in real-time. Therefore, the novel planning approach uses *smart-initialization* in order to plan multiple trajectories in such situations and, thus, make its behavior more flexible. The resulting speed-gain due to the *smart-initialization*, when multiple weights and targets are varied, is shown in Table II. The used time-horizon in this case is 6 seconds with a step-time of $\Delta t = 0.2s$. Due to an average speed gain of 27% a larger set of weights and targets can be optimized in every time-step and the variety and flexibility of the planner is increased.

TABLE I: Performance of the Optimization

N	Δt	Time-horizon [s]	Run-time [s]	Iterations [N]
10	0.1	1	0.0012	7
20	0.1	2	0.0443	92
30	0.1	3	0.1453	172
40	0.1	4	0.206995	142
10	0.2	2	0.0012	16
20	0.2	4	0.0443	135
30	0.2	6	0.1453	76
40	0.2	8	0.206995	168

The data in this table has been generated using a single set of weights and targets. The step-size Δt and the time-horizon are varied.

TABLE II: *Smart-Initialization*

Targets	Weights	Decrease [s]	Decrease [%]	Total-Time [s]
1	5	-0.467822	25%	0.78
5	1	-0.28829	40%	0.66
5	5	-0.85	16%	4.00

Performance-gain using prior results of the optimization as initial estimates. The last column shows the total computational time using *smart-initialization*.

VI. Conclusion and Outlook

In this work, the feasibility and benefits of a multi-objective optimization used in motion planning for autonomous driving have been shown. Due to the increased solution space, the planner is able to cope well with a large variety of situations and, therefore, trends towards universal applicability. Furthermore, by using a *smart-initialization*, the time to convergence is decreased significantly when optimizing multiple trajectories. Due to the three-fold modular architecture, the planning approach is capable of evaluating the planned trajectories independent of the objective function and, thus, capable of providing a higher level of safety. Furthermore, the Selection-module can not only select the best trajectory based on the Evaluation-module, but can also include higher-order strategies, such as favoring lane-changes to staying on the current lane.

Future research should be focused on situation dependent targets and weights in order to further improve the performance and capabilities of the planner. This could be achieved by learning a mapping of the current environment (relative to the ego vehicle) to specific weights and targets used in the optimization – creating “intuition” in motion planning for autonomous vehicles.

References

[1] Jeong Hwan Jeon, Sertac Karaman, and Emilio Frazzoli. “Anytime computation of time-optimal off-road vehicle maneuvers using the RRT”. In: *Proceedings of the IEEE Conference on Decision and Control* (2011), pp. 3276–3282.

[2] Vincenzo D’Onofrio, Marco Sagliano, and Yunus E. Arslantas. “Exact Hybrid Jacobian Computation for Optimal Trajectories via Dual Number Theory”. In: *AIAA Guidance, Navigation, and Control Conference* January (2016), p. 23.

[3] Julius Ziegler et al. “Making Bertha drive — An autonomous journey on a historic route”. In: *IEEE Intelligent Transportation Systems Magazine* 6.2 (2014), pp. 8–20.

[4] Benjamin Gutjahr and Moritz Werling. “Optimale Fahrzeugquerführung mittels linearer, zeitvarianter MPC”. In: Workshop Fahrerassistenzsysteme (2015).

[5] Wenda Xu et al. “A real-time motion planner with trajectory optimization for autonomous vehicles”. In: *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2061–2067.

[6] Maximilian Brunner et al. “Repetitive learning model predictive control: An autonomous racing example”. In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. 1. IEEE, 2017, pp. 2545–2550.

[7] Brian Ichter et al. “Real-time stochastic kinodynamic motion planning via multiobjective search on GPUs”. In: *Proceedings - IEEE International Conference on Robotics and Automation* (2017), pp. 5019–5026.

[8] Yong Zhang, Dun-wei Gong, and Jian-hua Zhang. “Robot path planning in uncertain environment using multi-objective particle swarm optimization”. In: *Neurocomputing* 103 (2013), pp. 172–185.

[9] Moritz Werling. *Ein neues Konzept für die Trajektoriengenerierung und -stabilisierung in zeitkritischen Verkehrsszenarien*. Vol. 60. 1. 2012, pp. 53–54.

[10] Julius Ziegler et al. “Trajectory planning for Bertha — A local, continuous method”. In: *Proceedings of the IEEE Intelligent Vehicle Symposium* (2014), pp. 450–457.

[11] Sameer Agarwal and Keir Mierle. *Ceres Solver*. 2012.

[12] Wenzel Jakob Moldovan, Jason Rhinelander, and Dean. *PyBind11*. 2017.

Appendix

A. Differentiation using Finite-Differences

The derivatives of a discrete trajectory $\mathcal{T}$ can be calculated using finite-differences. The first three discrete derivatives using forward differentiation can be mathematically expressed as

$$\dot{\mathrm{x}}_\mathrm{d} = \frac{x_{\mathrm{k}+1} - x_\mathrm{k}}{\Delta t} \tag{13a}$$

$$\ddot{\mathrm{x}}_\mathrm{d} = \frac{x_{\mathrm{k}+1} - 2x_\mathrm{k} + x_{\mathrm{k}-1}}{\Delta t^2}. \tag{13b}$$

$$\dddot{\mathrm{x}}_\mathrm{d} = \frac{x_{\mathrm{k}+3} - 3x_{\mathrm{k}+2} + 3x_{\mathrm{k}+1} - x_\mathrm{k}}{\Delta t^3}. \tag{13c}$$

It has to be taken into account, that the above presented equations introduce round-off and truncation errors.