2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Shared Autonomy for Teleoperated Driving: A Real-Time Interactive Path Planning Approach

Dmitrij Schitz[1], Shuai Bao[2], Dominik Rieth[3] and Harald Aschemann[1]

Abstract— **Teleoperation deals with extraordinary situations where an external operator takes over the control of an autonomous vehicle. Especially in complex urban scenarios, this may cause a too high workload for the human operator, resulting in suboptimal solutions. This contribution presents a teleoperation paradigm to raise the autonomy level of teleoperated driving, while the operator still remains the main decision-maker in all driving tasks. The introduced approach generates collision-free paths using LiDAR sensor information and suggests them to the operator. Therefore, a new hybrid path planning method has been developed, which searches and clusters in the first phase all feasible paths in the environment using a modified Rapidly-Exploring-Random Tree (RRT). In the second phase, the path selected by the operator is optimized online by a modified CHOMP algorithm. Real driving experiments confirm the effectiveness of the approach and highlight both the achieved driving safety and real time capability.**

I. Introduction

In recent years, the development of automated driving has undergone a rapid process. Current autonomous vehicles are able to drive autonomously in low challenging environments. However, complex urban scenarios cannot be completed without any faults, see [1].

Meanwhile, teleoperated driving has proven a high potential to overcome the functional limitations of autonomous driving with the current level of machine perception [1]. In the conventional direct control teleoperation method, the human is kept in the vehicle control loop. In this way, possible weaknesses with regard to machine perception can be compensated. While this teleoperation paradigm works well at low speed without any automated driving functions, complex urban scenarios cause a high workload for the operator, which may lead to a stop-and-go driving behavior. This problem can be traced back to the communication time delay on the one hand and to the lack of three-dimensional perception on the other. This forces the operator to drive more slowly and more carefully than in normal driving.

In order to relieve the human operator in complex teleoperation tasks, this work introduces a real-time capable supervisory control for interactive path planning between operator and autonomous vehicle. Since the operators input is not used to close the control loop, the proposed interactive planning scheme is insensitive w.r.t. any time delays that may occur in a conventional mobile network connection. Using LiDAR sensor information the proposed approach searches the vehicle's environment for reasonable paths and suggests them to the operator in the first phase by a modified RRT, see Fig 1. Since the environment changes while the vehicle is moving, the path selected by the operator that previously appeared to be beneficial may turn out as collision-afflicted. In the second phase, the selected path is thus optimized online in a computationally efficient manner. For this purpose, CHOMP, a trajectory optimization, is employed for a local path adaptation of kinematically constrained vehicles. While the automated vehicle progresses along the selected path, new path suggestions for the operator are generated.

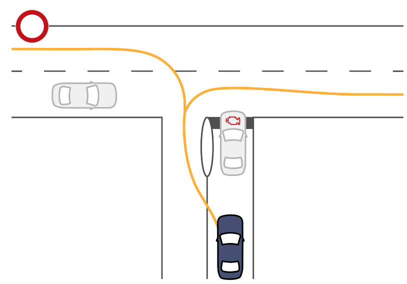


Fig. 1: Generated paths among which an operator can choose based on his perception.

The remainder of this paper is structured as follows: In Sec. II, related work on supervisory controls for teleoperated driving and path planning techniques are presented. After a brief problem statement, the proposed interactive path planning approach and the developed algorithms are introduced in Sec. III. The benefits of the proposed approach are highlighted in different urban scenarios by real driving experiments in Sec. IV. The paper ends with a conclusion in Sec. V.

II. Literature Overview

A. Supervisory Controls for Teleoperated Driving

In supervisory control, the human operator is only involved in planning tasks: The operator communicates high-level goals over the delayed communication channel that the robot executes on its own afterwards [2].

In this context, a Human Machine Interface for collaborative control has been developed in [3]. The work in [4] introduces an approach for shared control in which trajectories are generated manually by the operator using the conventional steering wheel and foot pedals. Although

[1]Chair of Mechatronics, University of Rostock, 18051 Rostock, Germany {dmitrij.schitz, harald.aschemann}@uni-rostock.de
[2] Department of Mechanical Engineering, Technical University Munich, 85748 Garching, Germany shuai.bao@tum.de
[3]BMW Group, Knorrstraße 147, 80788 Munich, Germany dominik.rieth@bmw.de

978-1-7281-9077-8/21/$31.00 ©2021 IEEE

this approach helps the operator to drive along straight-line paths, the manual generation of appropriate trajectories in real-time with curved paths or in turning scenarios resulted in a stop-and-go driving behaviour.

Higher automated concepts have been proposed in [5] and [6]. Similar to the idea of this work, the operator is given path suggestions. The vehicle then follows the path chosen by the operator. However, these concepts are only suitable for well-structured environments in which all relevant obstacles are known. That is not the case in almost every urban scenario.

B. Path Planning

The development of path planning algorithms has a long history in the field of robotics. Basically, they can be divided into three categories: search based methods, sample based methods and numerical optimization [7].

Search-based methods, like A^* and D^* along with their descendents, generate paths by connecting points along the discrete environment representation, the occupancy grid. Both search speed and path quality of these algorithms highly depend on the resolution of the discrete representation. Furthermore, the majority of the search-based algorithms are not able to ensure the feasibility of the paths due to the discretization [8].

The Rapidly-Exploring-Random Tree (RRT) is the most known sample-based method. It builds up a tree-structured graph that grows towards stochastic samples from the search space. In general, this algorithm uses the nearest distance to the current sample to determine the node to be expanded [9]. In addition, kinematic constraints can be taken into account, which renders it attractive for application to autonomous vehicles.

Numerical optimization methods aim to minimize an objective function subject to different constrained variables [7]. Representative approaches are Potential Field Methods [10] and Model Predictive Control [11]. They have the capability to take dynamical models into account and thus generate smooth and feasible paths. The evaluation of the dynamic model, however, leads to a high calculation time, which may not be feasible for many real-time scenarios. Furthermore, these algorithms need a collision-free guess to initialize the optimization. CHOMP, on the other hand – a relatively novel numerical optimization technique – has already shown its successful application as a trajectory planner in high-dimensional spaces thanks to its computational efficiency, see [12] and [13]. Unlike other optimization techniques, the path for initializing the optimization can be collision-afflicted. However, kinematic constraints are not taken into account in this algorithm.

A hybrid path planning was presented in [14], for instance. This approach benefits from both search strategies and optimization algorithms. For autonomous navigation in parking lots an A^* variant generates a path in the first phase. Subsequently the path is smoothed by a MILP-based approach. However, this method requires a predefined target pose that is available in its use case. A hybrid path planning that does not require a predefined goal is presented in [5]. Unlike the previously mentioned method, the RRT algorithm is used instead of A^*. Both approaches use a numeric optimization, which requires collision-free initial paths. Therefore, they can only be used in simple scenarios where all relevant objects are captured by LiDAR sensors. In almost every urban situation, obstacles exist that are either not completely covered by the LiDAR measurement or partly hidden by other obstacles. The motion of the vehicle on a planned path may lead to the discovery of new obstacles unseen before. This may change the situation completely – a path that previously appeared to be collision-free and beneficial may turn out as infeasible. On the other hand, both approaches have an optimization time of more than 200 ms, which is not sufficient for automated driving.

III. Overview on the Approach

Teleoperated driving in complex urban scenarios leads to a high workload for the human operator. Stabilizing the vehicle and avoiding possible collisions during remote control require both full attention and a high level of cognitive performance. Main issues are, especially, a delayed video transmission of the vehicle's environment as well as the lack of a three-dimensional perception.

A useful approach to relieve the operator in complex scenarios is to continuously generate further feasible paths and, then, forwards them to the operator, see Fig. 1. Since it is desirable that the operator's cognitive abilities are taken into account, it is important that the found paths are not selected automatically by the machine. For this reason, the generated paths are suggested to the operator in each command step. Only after their confirmation by the operator, the corresponding path will be followed by the vehicle.

The proposed interactive approach for teleoperated driving does not use predefined target points. Instead, it is the task of a global path search algorithm to generate reasonable paths and suggest them to the operator. The suggestion of too many paths, however, must be avoided because it may overstrain the operator. Instead of stopping the vehicle at the end of a selected path, new path suggestions are generated beforehand. However, while the vehicle navigates as selected by the operator, the vehicle environment changes continuously. Therefore, it is up to a very fast reactive local path adopter to ensure that no collision occurs. Therefore we seek to adapt the CHOMP algorithm – originally a trajectory optimization technique for motion planning in high-dimensional spaces [15] – as the local obstacle avoidance algorithm in this paper. A major advantage of this modified optimization technique is that the initial path does not have to be collision-free. In almost every urban situation, obstacles exist that are either not completely covered by the LiDAR measurement or partly hidden by other obstacles. The motion of the vehicle on a planned path leads to the discovery of new obstacles unseen before. This becomes particularly effective, when the last optimal path that previously appeared to be collision-free and beneficial turns out as infeasible.

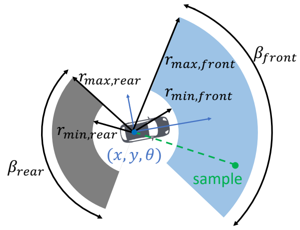


Fig. 2: Sampling range of the modified RRT.

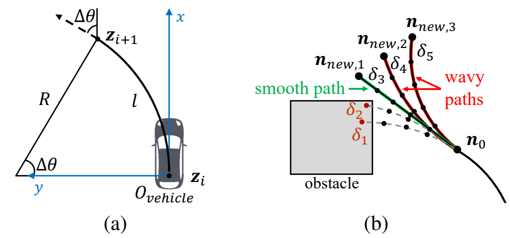


Fig. 3: Tree expansion: (a) a single step with one control input, (b) one expansion step with added nodes.

A. RRT-based Path Search

The RRT algorithm is commonly used to find a path between a starting point and an end point [16]. In the context of the proposed interactive path planning, the RRT is modified to explore the entire occupancy grid map and to build up a tree with all feasible paths. For this purpose, the stochastic samples are generated in the area of interest around the planning pose (x, y, θ) in a polar coordinate system that is centered on the vehicle rear axis:

$$\begin{bmatrix} x_{sample} \\ y_{sample} \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + r \begin{bmatrix} cos(\alpha) \\ sin(\alpha) \end{bmatrix}, \tag{1}$$

where the sampling distance r is a stochastic value in the range of $[r_{min}, r_{max}]$, whereas the stochastic angle lies in the range of $[\theta - \frac{\beta}{2}, \theta + \frac{\beta}{2}]$. The clearance distance r_{min} neglects the sampling in the vicinity of the planning pose, since they are less helpful for the tree expansion. They would lead to unreasonable U-turns, which are rare maneuvers in urban scenarios. The maximum radial component r_{max} specifies the planning horizon, whereas the angular component takes the the kinematic constraints into account.

In order to speed up the process of finding further paths while driving, we subdivide between forward sampling and backward sampling, see Fig. 2. At the moment when no possible paths in the forward direction could be found, additional computational effort is accepted to enable backwards driving. Such situations usually arise from a standstill position, when the operator has just been contacted to resolve a blocking situation. At this point, however, the additional computational effort can be neglected, as it takes the operator some time to understand the scenario.

A typical approach to reduce the computational effort of the original RRT is the use of heuristics [5]. The cost-to-go estimation represents a well-known improvement [17]. However, this heuristic requires a specific goal, which is not present in our approach. Instead, we decide which node to expand next among the existing ones in the tree by introducing a new measurement criterion γ. The next node is the one with the smallest criterion value. This criterion consists of three components which are added up for each node:

$$\gamma = \gamma_{smooth} + \gamma_{length} + \gamma_{distance}\,. \tag{2}$$

The term γ_{smooth} rates the control commands, i.e. the steering angles that were needed from the starting pose to the node. This way, smooth paths are preferred in the expansion process. The second component γ_{length} favors the short branches of the tree to be expanded, which leads to a more thorough search of the occupancy grid map. The last component $\gamma_{distance}$ is motivated by the original exploring heuristic [9], which measures the euclidian distance between sample point and node.

In contrast to other approaches, where only one node is expanded into the direction of the sample point, up to five nodes are added to the expansion tree in this work. The nodes are generated by means of a kinematic single track model. The control inputs are a series of steering angles δ. By limiting the value of δ, the vehicle kinematic constraints can be guaranteed. With a fixed step length l and the previous vehicle state $z_i = (x_i, y_i, \theta_i)$, the subsequent state can be calculated as

$$\begin{cases} \Delta\theta = \frac{l \cdot tan\delta}{L}. \\ R = \frac{l}{\Delta\theta} \\ x_{i+1} = x_i + R \cdot (sin(\theta_i + \Delta\theta) + sin\theta_i) \\ y_{i+1} = y_i + R \cdot (cos\theta_i - cos(\theta_i + \Delta\theta)) \\ \theta_{i+1} = \theta_i + \Delta\theta \end{cases}, \tag{3}$$

where R stands for the turning radius, L is the wheelbase and $\Delta\theta$ represents the change of vehicle orientation. see Fig. 3 (a). To avoid collision in the segment between two nodes the intermediate states are checked for collision, see Fig. 3 (b). Only if all intermediate states are collision-free, the node is added to the expansion tree.

In the original RRT algorithm, the search process is terminated after a feasible path from starting point to end point has been found [9]. Since this is not available in our approach, the nodes are expanded until a predefined path length is reached. After a specific number of paths are found or maximal sample attempts has been reached, the path search algorithm is terminated.

The result of the modified RRT, is hence, a multitude of feasible paths. To avoid overstraining the operator with the path selection task, the number of the paths has to be reduced without discarding relevant ones. For this purpose the found paths are clustered according to their end positions by means of the DBSCAN algorithm [18].

B. Path Optimization using modified CHOMP

Sample-based planning algorithms – especially RRTs – generally do not provide optimal solutions. For this reason, the results of the RRT are improved online by means of a modified CHOMP algorithm. The aim is to achieve optimality in terms of obstacle avoidance and smoothness while maintaining the drivability of the path. The CHOMP algorithm in its original form iteratively improves the quality of an initial path ξ_0 by optimizing an objective functional

$$U(\xi) = f_{obs}(\xi) + \lambda_{\text{smooth}} \cdot f_{smooth}(\xi) \tag{4}$$

that represents a trade-off between obstacle avoidance and path smoothness. Here, λ_{smooth} denotes a weighting factor, cf. [12]. The path ξ is represented by a uniform discretization: $\xi = (\ \mathbf{q}_1^\top,\ \mathbf{q}_2^\top,\ \ldots,\ \mathbf{q}_n^\top)$, with $\mathbf{q}_0 = (x_0,\ y_0)$ as the fixed starting point. The objective functional $U(\xi)$ is minimized by iteratively solving the sequential quadratic problem

$$\xi_{i+1} = \arg\min_{\xi} \left(U(\xi_i) + (\xi - \xi_i)^\top \nabla U(\xi_i) + \tfrac{\eta_i}{2} ||\xi - \xi_i||_{\mathbf{A}} \right), \tag{5}$$

where i denotes the iteration number. The matrix $\mathbf{A}$ corresponds to Riemann's metric, and η is a regularization coefficient for the balance between step size and minimization of the cost function $U(\xi)$. The given optimization problem in (5) is solved by means of a dual projected Newton method. For this purpose, the gradient of the objective functional $\nabla U(\xi_i)$ needs to be calculated in advance numerically.

1) Extension by Vehicle Dimensions: In order to take the vehicle dimensions into account in the optimization, the vehicle's body is over-approximated by a set of three circles. Using this approximation, the distance of the vehicle to any obstacle can be calculated efficiently. In the case of a circle, the distance to any point in the plane results from the distance to the centre of the corresponding circle subtracted by its radius. By using the Jacobian, which projects every body point onto the configuration ξ, the origin obstacle objective and its gradient are reformulated as the sum of the set of body points, cf. [19].

2) Curvature Constraints: As mentioned before, the objective functional of the original CHOMP algorithm represents a trade-off between an obstacle avoidance and path smoothness. In complex urban scenarios, for reasons of the passenger's and the operator's feeling of security, it is desirable to keep a greater distance to obstacles instead of generating smoother paths with narrow distances to obstacles. For this reason, the obstacle objective is weighted higher than the smoothness objective. In rather rare edge cases, however, this may lead to an infeasible path. The fact that the curvature limits of the vehicle may be exceeded in such situations cannot be neglected. For this purpose, we introduce a curvature objective as follows

$$\begin{aligned} f_{curv}(\xi) &= \sum_{k=1}^{n} f_{curv,k}(\xi), \text{ where} \\ f_{curv,k}(\xi) &= \begin{cases} 0 & \text{if } \|\kappa_k\| \leq \kappa_{max} \\ \frac{1}{2}(\kappa_k - \kappa_{max})^2 & \text{else} \end{cases} \end{aligned} \tag{6}$$

with the corresponding gradient components

$$\begin{aligned} \nabla f_{curv,k}(\xi) &= \begin{cases} 0 & \text{if } \|\kappa_k\| \leq \kappa_{max} \\ (\kappa_k - \kappa_{max}) \nabla \kappa_k & \text{else} \end{cases} \\ &\text{for } k = 1 \ldots n\ ,\ \nabla f_{curv,k}(\xi) \in \nabla f_{curv}(\xi)\ . \end{aligned} \tag{7}$$

The term κ denotes the curvature of the path ξ, defined by

$$\kappa = \frac{1}{\|\dot{\xi}\|^2} \left(\mathbf{I} - \hat{\dot{\xi}}\, \hat{\dot{\xi}}^\top \right) \cdot \ddot{\xi}\ , \tag{8}$$

where $\hat{\dot{\xi}} = \xi / \|\xi\|$ is the normalized vector and $\dot{\xi}$ the first time derivative of ξ using a uniform time discretization, cf. [20]. The gradient $\nabla \kappa$ is derived as the partial derivative for all path points. In an optimization step, first the path is pushed away from the obstacles. If the limit κ_{max} is exceeded, the curvature objective pulls the path back within the constraints, see Fig. 4. The term κ_{max} results from the vehicle's drivable curvature limit subtracted by a threshold.

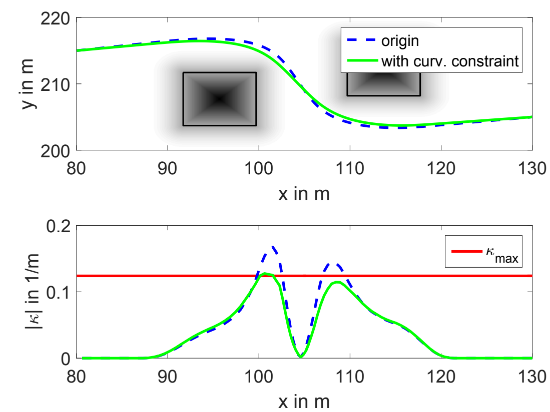


Fig. 4: Simulation results regarding the curvature constraints.

3) Reference Path Objective: The solutions generated by the RRT include the evaluation of a kinematic model. Consequently, the results already offer good conditions for a global optimum according to the respective cluster. Therefore, we extend the objective functional $U(\xi)$ by a further objective

$$f_{ref}(\xi) = \frac{1}{2} \sum_{k=1}^{n} d_k(\mathbf{q}_k, \xi_0)^2\ , \tag{9}$$

which penalizes the distance to the reference path ξ_0 – the result of RRT – in the form of the Frenet-transformed coordinate. The corresponding gradient components of the reference path objective become

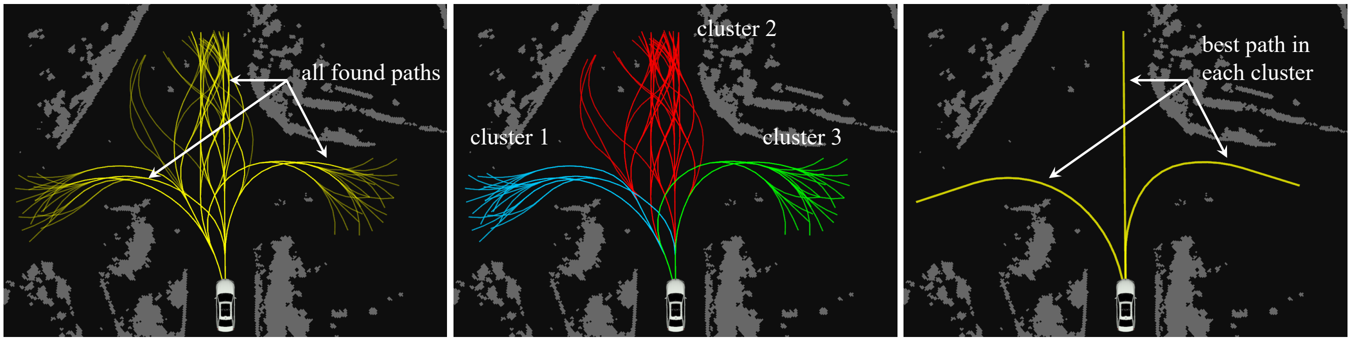
$$\begin{aligned} \nabla f_{ref,k}(\xi) &= d_k(\mathbf{q}_k, \xi_0) \left[sin(\theta_k) \quad -cos(\theta_k) \right] \\ &\text{for } k = 1 \ldots n\ ,\ \nabla f_{ref,k}(\xi) \in \nabla f_{ref}(\xi)\ . \end{aligned} \tag{10}$$

Together with the curvature objective, the objective functional $U(\xi)$ is reformulated to

$$\begin{aligned} U(\xi) = &\lambda_{obs} \cdot f_{obs}(\xi) + \lambda_{smooth} \cdot f_{smooth}(\xi) \\ &+ \lambda_{curv} \cdot f_{curv}(\xi) + \lambda_{ref} \cdot f_{ref}(\xi)\ . \end{aligned} \tag{11}$$

IV. Experimental Results

The effectiveness of the proposed interactive path planning approach is assessed by means of real driving experiments in different scenarios. The occupancy grid map forms the basis of the planning process. The needed data is provided by a LiDAR sensor which is mounted on top of the experimental vehicle. In addition to the occupancy grid map, a camera



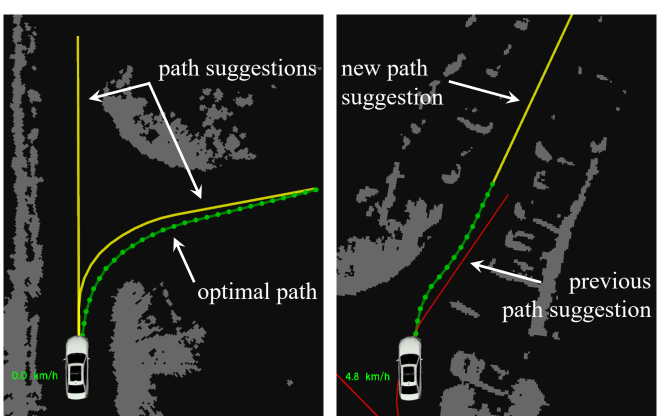
(a) (b) (c)

Fig. 5: Results of clustered RRT path search: (a) all found paths, (b) clustered results, (c) path suggestions for operator.

transmission is used for operator feedback. All calculations were carried out on an Intel i9-9980XE 3 GHz processor with an average computation time per calculation step of 11 ms for the path search with the modified RRT and 8 ms for the path optimization with the modified CHOMP algorithm.

An example of the clustered path search is shown in Fig. 5 using an intersection scenario. The intention is to analyze whether all reasonable paths are found and only meaningful ones are suggested. The paths found by the modified RRT in this scenario are depicted in Fig. 5 (a). As can be seen, the exploration process generates a lot of feasible paths. Proposing all of them to the operator is irresponsible, since it can be expected that the operator will be overwhelmed during the selection task. This is particularly important to avoid while the vehicle is moving. For this purpose, the feasible paths are clustered using the DBSCAN algorithm in order to reduce the suggested solutions to the most relevant ones. The subdivided clusters found in this scenario are illustrated in Fig. 5 (b). For each cluster found, only one path is subsequently suggested to the operator. The selection of the path to be suggested is based on the cost criterion value of the last node, see Sec. III-A. The final result of this experiment is shown in Fig. 5 (c). The operator is given three reasonable paths to choose from.

In another urban scenario, the clustered path search resulted in two solutions between which the operator could select, see Fig. 6 (a). The operator's decision led to the task for the automated vehicle to make a right turn into a side street, which is only partially visible from the vehicle's starting position, see Fig. 7 (a). The modified RRT algorithm itself does not deliver optimality [16]. As can be seen in Fig. 6 (a), the modified CHOMP algorithm is able to remove extraneous motion of the selected option and generates an optimal path. Optimality at this point in time is essentially achieved as a tradeoff between smoothness and the distance to the reference path – the selected path option. As the vehicle progresses along the calculated path, the modified CHOMP algorithm optimizes the path further in real time based on newly received sensor information. In the real driving experiment, for instance, previously hidden obstacles may appear as the vehicle turns into the side street, see Fig. 6



(a) (b)

Fig. 6: Real driving results at two consecutive time stamps.

(a) Before the turning operation

(b) After the turning operation facing an unexpected obstacle

Fig. 7: Image sections of the camera view belonging to Fig. 6.

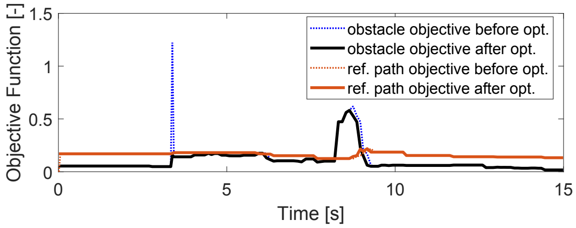


Fig. 8: Time evolution of the objective functions corresponding to the scenario in Fig. 6.

(b) and Fig. 7 (b). Based on the newest sensor information, the optimization algorithm is able to adapt the path online and, hence, to avoid collisions. In this way the end point of the optimal path was moved away from the reference path in order to reduce the value of the obstacle objective. The value of the reference path objective, on the other hand, increased, see Fig. 8 at approx. 9 sec. Since the curvature constraints were not violated, the curvature objective had no influence in this experiment. For this reason, it has been suppressed in Fig. 8. The optimal path forms the basis for a further clustered path search. As shown in Fig. 6 (b), the path search algorithm generates a straight path option for the operator. The link to the video associated with the described scenario can be found in the footnote[1] below.

V. Conclusions

This paper presents an interactive path planning approach for teleoperated driving in order to relieve the human operator in complex urban scenarios. In this context, the RRT algorithm is modified in such a way that further paths are generated without predefining target positions, among which the operator can select. In order to generate smooth optimal paths and to react quickly to obstacles, we employ a modified CHOMP algorithm for path planning in automated driving. The necessary modifications are discussed in this work. The originally introduced obstacle objective, for instance, is extended to address the vehicle geometry in the optimization. In addition, we introduce two new objective functions that take into account deviations from a reference path on the one hand and the vehicle's kinematic constraints on the other. The algorithms have been evaluated in real driving experiments, where the system successfully generates suitable path suggestions and where the finally selected option is optimized in real time to avoid collisions. The studies revealed that the use of the CHOMP algorithm for a path optimization in automated driving tasks offers clear advantages in terms of calculation time in comparison to traditional methods. Although a direct comparison of the computing effort is only possible qualitatively due to the different hardware used, the path optimization time of 50 - 300 ms in [7] or [14] in relation to only 8 ms in this work shows a significant increase in efficiency on a comparable processor.

In future work, the system will be expanded by a trajectory planner, to consider dynamic objects along the path optimized by CHOMP. Subsequently, as a visual feedback for the operator, the optimized path will be projected into the camera streams.

[1] Video of the experiment: https://youtu.be/POhICIi4FJI

References

[1] S. Neumeier, N. Gay, C. Dannheim, and C. Facchi, "On the way to autonomous vehicles teleoperated driving," *Automotive meets Electronics; 9th GMM-Symposium*, pp. 1–6, 2018.

[2] T. B. Sheridan, "Space teleoperation through time delay: Review and prognosis," *IEEE Transactions on robotics and Automation*, vol. 9, no. 5, pp. 592–606, 1993.

[3] T. Fong, C. Thorpe, and C. Baur, *Collaborative control: A robot-centric model for vehicle teleoperation*. Carnegie Mellon University, The Robotics Institute Pittsburgh, 2001, vol. 1.

[4] S. Gnatzig, F. Schuller, and M. Lienkamp, "Human-machine interaction as key technology for driverless driving – a trajectory-based shared autonomy control approach," *IEEE International Symposium on Robot and Human Interactive Communication*, pp. 913–918, 2012.

[5] A. Hosseini, T. Wiedemann, and M. Lienkamp, "Interactive path planning for teleoperated road vehicles in urban environments," *IEEE Conference on Intelligent Transportation Systems*, pp. 400–405, 2014.

[6] G. Tanzmeister, M. Friedl, D. Wollherr, and M. Buss, "Path planning on grid maps with unknown goal poses," *IEEE Conference on Intelligent Transportation Systems*, pp. 430–435, 2013.

[7] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1135–1145, 2015.

[8] D. Ferguson, M. Likhachev, and A. Stentz, "A guide to heuristic-based path planning," *Proceedings of the International Workshop on Planning under Uncertainty for Autonomous Systems, International Conference on Automated Planning and Scheduling*, pp. 9–18, 2005.

[9] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The international journal of robotics research*, vol. 20, no. 5, pp. 378–400, 2001.

[10] J. Ren, K. A. McIsaac, and R. V. Patel, "Modified newton's method applied to potential field-based navigation for mobile robots," *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 384–391, 2006.

[11] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer Science & Business Media, 2013.

[12] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "CHOMP: Gradient optimization techniques for efficient motion planning," *IEEE International Conference on Robotic and Automation*, pp. 489–494, 2009.

[13] M. Zucker, Y. Jun, B. Killen, T.-G. Kim, and P. Oh, "Continuous trajectory optimization for autonomous humanoid door opening," *IEEE Conference on Technologies for Practical Robot Applications*, pp. 1–5, 2013.

[14] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 485–501, 2010.

[15] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, "Chomp: Covariant hamiltonian optimization for motion planning," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1164–1193, 2013.

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

[17] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, "Real-time motion planning with applications to autonomous urban driving," *IEEE Transactions on control systems technology*, vol. 17, no. 5, pp. 1105–1118, 2009.

[18] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." *International Conference on Knowledge Discovery and Data Mining*, vol. 96, no. 34, pp. 226–231, 1996.

[19] D. Schitz, G. Graf, D. Rieth, and H. Aschemann, "Corridor-based shared autonomy for teleoperated driving," *IFAC World Congress*, 2020.

[20] S. Choudhury and S. Scherer, "Constrained CHOMP using dual projected newton method," Carnegie Mellon University, Pittsburgh, PA, Tech. Rep., 2016.