Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Corridor-Based Shared Autonomy for Teleoperated Driving

Dmitrij Schitz *, **Gaetano Graf** **, **Dominik Rieth** ***, **Harald Aschemann** *

* *Chair of Mechatronics, University of Rostock, (e-mail: {Dmitrij.Schitz, Harald.Aschemann}@uni-rostock.de)*
** *Institute of Informatics, University of Munich (LMU), (e-mail: Gaetano.Graf@ifi.lmu.de)*
*** *BMW Group, Munich, (e-mail: Dominik.Rieth@bmw.de)*

Abstract: Given the on-going development of powerful hardware, software and algorithms for automated driving, the number of tasks that vehicles can solve autonomously steadily increases. However, fully autonomous driving in all situations is highly demanding and currently not feasible yet. It may happen that the vehicle faces situations in which the decision system is overstrained and, hence, falls back into a predefined safe state. In the future, moreover, neither control interfaces like a steering wheel and foot pedals nor a qualified driver may be available to assist the car in a blocking situation. This contribution, hence, presents a concept for teleoperated driving, i.e., a remote operation with distinct human machine interactions that explicitly addresses such highly complex driving tasks. It quickly copes with such undesired blocking situations in order to minimize the need for a both time- and cost-intensive road assistance. The concept focusses on teleoperated driving of road vehicles in urban environments. Based on the methodology of a shared autonomy, a corridor-based planning scheme is derived. The remote operation task takes advantage of a fusion of automated driving functions and human-predefined corridors. Within this specified corridor, a path planning algorithm using dual projected Newton method determines a collision-free path that the vehicle is capable to follow. Simulation results show the effectiveness of the proposed method and highlight the achieved driving safety.

Keywords: Teleoperated driving, autonomous vehicles, shared control, shared autonomy, human-machine interaction (HMI), path planning

1. INTRODUCTION

World-wide research activities on autonomous vehicles (AV) aim at a completely driver-less transport. According to estimates of Neumeier et al. (2018), however, the first fully autonomous vehicles will not be introduced until 2028 to 2030. Graf (2019) points out that humans still outperform machines regarding perception and processing tasks. Consequently, situations may occur where the decision system reaches its limits and unforeseen traffic scenarios may lead to a blocking situation. Even with fully autonomous cars, there will be scenarios where human intervention is mandatory, e.g. in the case of severe sensor failures. Cars without any passengers or with passengers that are unqualified to drive are conceivable in the future. Accordingly, no one is in place to take control of the vehicle, and a blocking situation may cause disturbing traffic jams. In order to offer immediate action without the need for road assistance, teleoperation – supporting critical urban spots – represents a promising option.

In space robotics, teleoperation has always been a common solution, especially if technical issues prevent an autonomous operation, see Sheridan (1989). As illustrated in Fig. 1, the advantage of teleoperation is that the human operator remains locally independent while monitoring a vehicle and having impact on the motion using certain control inputs. Thereby, the human is still part of the mission control, can react to unforeseen situations and lead the mission progress (Winfield (2000), Gnatzig et al. (2012)). Tzafestas (2007) points out that a meaningful compromise between automation level and manual adaptation is the key element for the design of a teleoperation system. Only with an optimal task distribution, the overall system can benefit at its best from human-machine interaction. As mentioned in Ferrell and Sheridan (1967), an important issue arises from latency effects in automation tasks: The larger the impact of time delays, the larger the potential towards a reduction of the execution time. Whereas humans often solve delay-free tasks with cognitive challenges

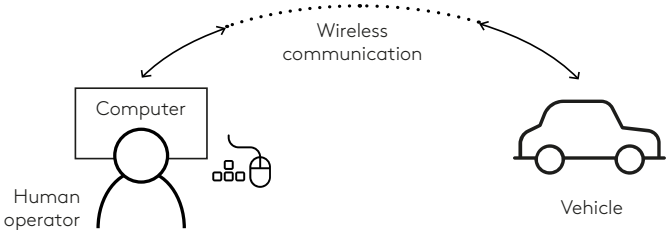


Fig. 1. Teleoperation in robotic applications.

faster than automatized systems, this relationship reverses with the presence of time delays. Depending on the delay, an optimal percentage of the machine task taken over by humans will result in a minimum completion time, cf. Gnatzig et al. (2013).
This paper presents a teleoperation concept based on shared autonomy using human-machine interaction on the level of navigation. The aim of this approach is to find an efficient and successful interaction of human decision-making and automated machine driving.

2. RELATED WORK

According to Sheridan (1993), the unavoidable communication link always introduces time delays in teleoperation. Pongrac (2008) points out that round-trip times of more than 200 ms already decrease human task performance. In particular, it is shown that a variable time delay has a very negative impact on human performance: Due to the limited bandwidth and variable delays in mobile networks, direct control of teleoperated vehicles – with a human in the closed loop – may be impossible in some situations, see Kay (1997). For this reason, an indirect control approach is proposed, where the human operator is only involved in planning tasks: The operator defines high level goals which the automated robots afterwards execute on their own.
As proposed in Kay (1997), such goals might be waypoints. Since the operator input is not used to close the control loop, this approach is insensitive w.r.t. any time delays. The proposed STRIPE system (Supervised TeleRobotics using Incremental Polyhedral Earth reprojection) is based on the transmission of individual images from a front camera mounted at the vehicle. The operator defines a sequence of waypoints in the camera image and transmits them to the vehicle, where the feedback control system guarantees path following. Since the mission scenario of the experimental vehicle NavLab 2, developed at the Carnegie Mellon University, are mainly off-road tasks, the system estimates the terrain topology on the basis of the previous route to relate the camera pixels to the real 3D environment. The individually defined waypoints are used as interpolation points of a cubic spline, so that a continuous path is forwarded to the vehicle control for tracking purposes. While the vehicle is automatically driving to the end of the defined route, a new image is sent to the operator from which the operator can again select waypoints and transmit them to the vehicle. For this purpose, the camera can be swivelled by the operator. The average round-trip time from transmitting the waypoints to obtaining the next camera image is around 12.9 sec. This is caused by the bandwidth limitation of the communication channel, and the limited computing power for capturing and compressing the image data, cf. Kay (1997). Moreover, NASA's Mars rovers are also guided by a waypoint-based system. Due to the very large time delays resulting from the distance to Mars as well as the limited communication that depends on favourable Earth-Mars constellations, the NASA operators had to send the entire sequence of movements for a Mars day in advance to the rover. The rover then has to accomplish the route using on-board autonomy alone (Bajracharya et al. (2008)). For path planning, a 3D environment was used that stems from stereo cameras mounted on the rover. In a simulation environment, the control commands are tested beforehand. Here, easy motion segments, spin commands and high-level waypoint commands can be selected (Cooper (1998)). The control sequence is verified first within the simulation environment and, after a successful simulation, sent to the rover.
Most recently, Gnatzig (2015) proposed a trajectory-based approach to remotely control road vehicles. Unlike waypoint-based control, the operator can directly define a sequence of trajectories in a video stream while the car is driving. By using the conventional steering wheel and pedal combinations, the operator can edit both the length and the curvature of the individual trajectory segments. Moreover, the robustness and the safety of this approach are proved for latencies of up to 600 ms.

3. CONCEPT OVERVIEW

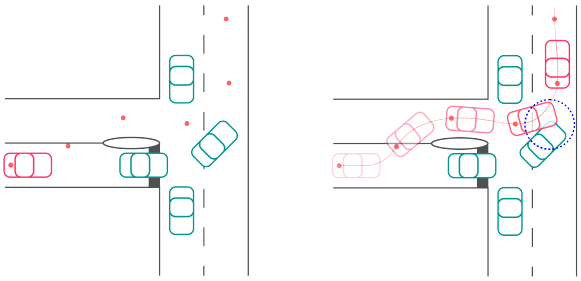
The teleoperation of vehicles causes an extra workload for the human operator. As explained in Hosseini et al. (2014), the delayed perception during teleoperated driving significantly increases the effort of the human, and the operator has to take care of the environment more intensively. In particular, this holds true in cases, where possible collisions have to be avoided.

3.1 Problem Statement

Trajectory-based control, where the vehicle follows straight-line paths based on a time-delayed camera stream input, has already proven to be easier for the operator in comparison to direct control. The manual generation of appropriate trajectories in real-time with curved paths or in turning scenarios, however, may be too challenging for the operator using this method. Manual corrections of the given paths may either result in a significant delay in the vehicle navigation or may even lead to a stop-and-go behaviour.
A similar drawback becomes obvious with waypoint-based control. Due to a possible misjudgement of the vehicle geometry, collision-afflicted paths may be generated in complex driving scenarios. Fig. 2 shows an example of a failed attempt to create a collision-free path using waypoint-based control. When setting waypoints in a complex scenario, no reliable statement can be made regarding possible collisions along the resulting path, see Fig. 2 (a). Only a subsequent simulation addressing both the vehicle geometry and the kinematics clarifies that a collision would happen in the considered scenario, see Fig. 2 (b).

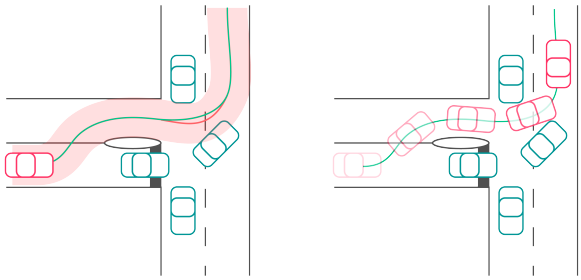
3.2 Corridor-Based Control Approach

As a remedy, the corridor-based control proposed in this paper should support the operator while generating paths in challenging situations. The idea of this approach is to specify areas in which the vehicle is allowed to move safely and without danger of collisions. In a corridor specified by the operator, the autonomous vehicle calculates, on its own, a collision-free path, which is then forwarded to the automated driving system and followed accurately by the vehicle. In this innovative teleoperation approach, the operator could either generate a complete corridor to the destination in advance or specify only a subarea first and further append corridor segments to it while the vehicle is



(a) (b)

Fig. 2. Example for a failed waypoint-based control attempt: (a) placement of the waypoints, (b) resulting collision-afflicted path.



(a) (b)

Fig. 3. Illustration of the corridor-based control concept: (a) specified corridor with a collision-afflicted – red – initial path as well as the collision-free – green – optimized path, (b) visualization of the autonomous execution of the collision-free path that resolves the blocking situation.

driving autonomously.
Please note that a manual definition of appropriate collision-free paths with waypoint-based and trajectory-based control represents a challenging task in complex urban scenarios. This may urge the operator to perform this task repetitively until he finds a feasible, collision-free solution. This problem can be eliminated if the operator specifies a corridor in which the vehicle, based on forthcoming sensor information, autonomously calculates a collision-free path using adequate optimization techniques. The planning algorithm may be supported by camera and LIDAR measurements as well as map information.
The concept for a corridor-based approach is shown schematically in Fig. 3. The approach allows the human operator to specify an area, i.e., the corridor in which the software on-board the vehicle is permitted to search autonomously for feasible paths. This becomes especially effective if, e.g., pedestrian islands or other flat objects are not detected as obstacles from sensor data and the associated evaluation software. The human operator can, hence, take advantage of the camera information, specify the boundaries of the corridor in such a way that either undetected obstacles can be avoided or flat obstacles may be included in the search space. Another important advantage is that the human operator does not need to find the collision-free path on his own. As mentioned in de Visser et al. (2018), human strengths are located in the area of cognitive skills, like situation analysis and behavioural decision-making. Gnatzig et al. (2012) highlights that – after years of experience and learning – a human operator may easily cope with difficult road topologies and confusing traffic scenarios. Automatized machines, on the other hand, have strong skills in vehicle stabilization and collision detection/avoidance due to their precise localization methodologies. The corridor-based approach reflects these findings appropriately. The human operator contributes his skill in the analysis of the situation and decision making. After an appropriate situation assessment, the human operator provides the vehicle with the navigation decision by means of a specified corridor. The vehicle then takes over, and autonomously calculates a collision-free path that is accurately followed afterwards. Thereby, blocking situations in critical urban environments can be resolved and handled efficiently as well as safely.

4. CORRIDOR SPECIFICATION

As described in the previous section, the implementation of the corridor-based approach consists of two main steps: The first step is the specification of a corridor, which can be interpreted as search space of a subsequent optimization step. In this work, this is done by:

- a spline-based definition of the centre line using desired poses of the operator $(x, y, \theta, \kappa, \dot{\kappa})$,
- a corresponding calculation of the corridor boundaries resulting from a predefined width.

The spline function represents a smooth polynomial curve that interpolates between given points $\mathbf{p}_A = [x_A\,,\ y_A]$, $\mathbf{p}_B = [x_B\,,\ y_B]$ with associated orientations defined by angles θ_A, θ_B, scalar curvatures κ_A, κ_B and the curvature derivatives, respectively. According to Reuter (1998), a smooth path generation should take into account the curvature derivatives as well. Therefore, a C^3-spline is necessary. Piazzi et al. (2007) proposed an alternative solution for this problem and employed a polynomial curve with degree seven according to $\mathbf{p}(w) = [x(w)\,,\ y(w)]\,,\ w \in [0, 1]$ for the interpolation:

$$\mathbf{p}(w) = [x(w)\,,\ y(w)] = \left[\sum_{i=0}^{7} \alpha_i \cdot w^i\ ,\quad \sum_{i=0}^{7} \beta_i \cdot w^i\right] \tag{1}$$

The variables α_i and β_i depend on the poses defined by the operator and the freely selectable curve-form factors η_k, $k = 1, \ldots, 6$ (Piazzi et al. (2007)).

5. PATH MANAGEMENT

For the planning of a collision-free path within the corridor specified by the human operator, the constrained CHOMP algorithm – a path planning method developed by Choudhury and Scherer (2016) – is employed in this work. CHOMP stands for Covariant Hamiltonian Optimization for Motion Planning and was introduced by Ratliff et al. (2009). Originally, the algorithm was developed for robotic applications as a trajectory optimization technique for motion planning in high-dimensional spaces. Since the constrained CHOMP is based on projected Newton method, an initial trajectory is required. According to

Dragan et al. (2011), the initial trajectory does not have to be collision-free. For this purpose, the centre line of the corridor as specified by the human operator is used for the initialization of the optimization. In order to implement the optimization, an efficient representation of the vehicle is required (Zucker et al. (2013a)). For this purpose, simplifying assumptions about the vehicle's geometry are made. The vehicle's body is over-approximated by a set B of three circles (see Fig. 4 and 5), described by their workspace positions $\mathbf{x}(\boldsymbol{\xi}(t), u) = [x_u(t), y_u(t)]^\top$, indexed by the configuration $\boldsymbol{\xi}(t) = [x(t), y(t), \theta(t)]^\top$ at time t and the robot's body point $u \in B$. For simplicity, we approximate B by a set of geometric primitives, i.e., circles. Consequently, the nearest distance of the approximated vehicle model can be easily computed. In the case of a circle, the distance to any point in the plane is given by the distance to the centre of the circle subtracted by its radius.

5.1 Objective Function

The goal of CHOMP is to find a smooth and collision-free trajectory in the vehicle configuration space $\mathbb{R}^3$ by iteratively improving the quality of the initial trajectory $\boldsymbol{\xi}_0$. This is accomplished by minimizing an objective function $U(\boldsymbol{\xi})$ that represents a trade-off between obstacle avoidance and path smoothness

$$U(\boldsymbol{\xi}) = f_{obs}(\boldsymbol{\xi}) + \lambda \cdot f_{smooth}(\boldsymbol{\xi}) \ , \tag{2}$$

where λ denotes a weighting factor. The term $f_{obs}(\boldsymbol{\xi})$ addresses the costs of being close to obstacles, whereas $f_{smooth}(\boldsymbol{\xi})$ penalizes the trajectory $\boldsymbol{\xi}$ w.r.t. smoothness and acceleration, cf. David et al. (2017).

Smoothness Objective
Using a uniform discretization which samples the trajectory over equal time steps of length Δt: $\boldsymbol{\xi} \approx (\mathbf{q}_1^\top, \mathbf{q}_2^\top, \ldots, \mathbf{q}_n^\top) \in \mathbb{R}^{n\times 3}$, with $\mathbf{q}_0$ and $\mathbf{q}_{n+1}$ as the fixed starting and ending points, the term describing the smoothness objective as the sum of squared first-order difference quotients w.r.t. time is

$$f_{smooth}(\boldsymbol{\xi}) = \frac{1}{2}\sum_{t=0}^{n} \left|\left| \frac{\mathbf{q}_{t+1} - \mathbf{q}_t}{\Delta t} \right|\right|^2 \ . \tag{3}$$

With the finite difference matrix $\mathbf{K}$ and the vector $\mathbf{e}$, which addresses the boundary conditions $\mathbf{q}_0$ and $\mathbf{q}_{n+1}$, the term in (3), according to Ratliff et al. (2009), can be rewritten as

$$f_{smooth}(\boldsymbol{\xi}) = \frac{1}{2}||\mathbf{K}\,\boldsymbol{\xi} + \mathbf{e}||^2 = \frac{1}{2}\boldsymbol{\xi}^\top \mathbf{A}\,\boldsymbol{\xi} + \boldsymbol{\xi}^\top \mathbf{b} + g \ , \tag{4}$$

with $\mathbf{A} = \mathbf{K}^\top\mathbf{K}$, $\mathbf{b} = \mathbf{K}^\top\mathbf{e}$ and $g = \mathbf{e}^\top\mathbf{e}/2$. Since the constrained CHOMP algorithm represents a projected Newton method, the gradients of the corresponding objective functions are required as well. Given the quadratic equation in (4), the computation of the smoothness gradient becomes straightforward

$$\nabla f_{smooth}(\boldsymbol{\xi}) = \mathbf{A}\,\boldsymbol{\xi} + \mathbf{b} \ . \tag{5}$$

Obstacle Objective
To obtain collision-free trajectories, the term $f_{obs}(\boldsymbol{\xi})$ penalizes the proximity of the vehicle to any object in the environment. Taking into account each body element $u \in B$, Ratliff et al. (2009) defines the obstacle objective as an integration of the workspace potential $c\big(\mathbf{x}(\boldsymbol{\xi}(t), u)\big)$ with respect to an arc-length parametrization. Using the same discretization as for $f_{smooth}(\boldsymbol{\xi})$, the obstacle objective is reformulated as follows

$$f_{obs}(\boldsymbol{\xi}) = \sum_{u\in B}\sum_{k=1}^{n} c\big(\mathbf{x}(\boldsymbol{\xi}(k\cdot\Delta t), u)\big)\ ||\dot{\mathbf{x}}(\boldsymbol{\xi}(k\cdot\Delta t), u)|| \ , \tag{6}$$

where $\dot{\mathbf{x}}(\boldsymbol{\xi}(k\cdot\Delta t), u)$ is the first time derivative of $\mathbf{x}(\boldsymbol{\xi}(k\cdot\Delta t), u)$. The workspace potential $c\big(\mathbf{x}(\boldsymbol{\xi}(k\cdot\Delta t), u)\big)$ quantifies the cost of a body element of residing at a particular point $\mathbf{x}$ in the workspace. If the vehicle is close to the obstacle, $c\big(\mathbf{x}(\boldsymbol{\xi}(k\cdot\Delta t), u)\big)$ attains large values. Additionally, high velocities are penalized. Therefore, high velocities close to obstacles lead to higher costs than small velocities in a larger distance. Depending on the definition of the workspace cost function, an arbitrary robot motion can be attained. In this work, the definition of Zucker et al. (2013b) is employed. According to Bosshard (2015), the corresponding gradient of the obstacle cost term results in

$$\nabla f_{obs} = \sum_{u\in B}\sum_{k=1}^{n} \mathbf{J}^\top ||\dot{\mathbf{x}}|| \left[(\mathbf{I} - \hat{\dot{\mathbf{x}}}\,\hat{\dot{\mathbf{x}}}^\top)\nabla c - c\boldsymbol{\kappa}\right] \ . \tag{7}$$

To simplify the notation we have suppressed the dependance of $\mathbf{J}$, $\mathbf{x}$ and c on summation variables k and u. In equation (7), the term $\hat{\mathbf{x}}$ denotes the normalized vector $\mathbf{x}/||\mathbf{x}||$. The Jacobian $\mathbf{J}$ describes the projection of the point u between configuration and geometrical space. The term $\boldsymbol{\kappa}$ denotes the curvature of the trajectory, defined by

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

Since a constant vehicle velocity along a geometrical path is assumed, the trajectory planning problem is identical to a path planning problem. Therefore, $\boldsymbol{\xi}$ is referred to as path in the following.

5.2 Dual Projected Newton Method

Given a finite-dimensional parametrization of the initial path $\boldsymbol{\xi}_0$ and the corresponding gradient of the objective function $U(\boldsymbol{\xi})$, the iterative optimization of the initial path is performed by using the dual projected Newton method. Choudhury and Scherer (2016) proposed this iterative algorithm for the solution of a sequential quadratic problem with linear inequality constraints

$$\begin{gathered}\boldsymbol{\xi}_{i+1} = \arg\min_{\boldsymbol{\xi}} \left[U(\boldsymbol{\xi}_i) + (\boldsymbol{\xi} - \boldsymbol{\xi}_i)^\top \nabla U(\boldsymbol{\xi}_i) + \frac{\eta_i}{2}||\boldsymbol{\xi} - \boldsymbol{\xi}_i||_{\mathbf{A}}\right] \\ \text{s.t } \mathbf{C}\,\boldsymbol{\xi} \le \mathbf{d} \ .\end{gathered} \tag{9}$$

Here, i denotes the iteration, whereas the expression $\mathbf{C}\,\boldsymbol{\xi} \le \mathbf{d}$ describes linear inequality constraints w.r.t. the path $\boldsymbol{\xi}$. In the given case, the corridor boundaries are addressed by this term.
With the initial primal path $\boldsymbol{\xi}_0$ and Lagrange multipliers $\mathbf{u}_0$, the iterative optimization can be stated as follows:

1. Perform an iteration of the projected Newton method

$$\mathbf{u}_{i+1} = \mathbf{P}_{\ge 0}\left(\mathbf{u}_i - \alpha_i \begin{bmatrix} \mathbf{S}_i & 0 \\ 0 & \mathbf{I}\end{bmatrix}\begin{bmatrix}\big[\nabla\mathbf{G}(u_i))\big]_{\mathbf{F}_i} \\ \big[\nabla\mathbf{G}(u_i))\big]_{\mathbf{B}_i}\end{bmatrix}\right) \ ,$$

$$\begin{aligned}\text{where } \ \mathbf{G}(\mathbf{u}) = \tfrac{1}{2\eta_i}\mathbf{u}^\top\mathbf{C}\mathbf{A}^{-1}\mathbf{C}^\top\mathbf{u} - \ldots \\ \mathbf{u}^\top\left(\mathbf{C}\boldsymbol{\xi}_i - \mathbf{d} - \tfrac{1}{\eta_i}\mathbf{C}\mathbf{A}^{-1}\nabla U(\boldsymbol{\xi}_i)\right) \ .\end{aligned} \tag{10}$$

2. Perform a primal path update

$$\boldsymbol{\xi}_{i+1} = \boldsymbol{\xi} - \frac{1}{\eta_i}\mathbf{A}^{-1}\nabla U(\boldsymbol{\xi}_i) - \frac{1}{\eta_i}\mathbf{A}^{-1}\mathbf{C}^\top \mathbf{u}_{i+1} \,. \quad (11)$$

Here, the update rule (11) is derived from the primal solution (9) using a Lagrangian dual method. $\mathbf{P}_{\geq 0}$ projects each variable to the positive half-plane, η is a regularization coefficient that specifies the trade-off between the step size α and the minimization of the cost function $U(\boldsymbol{\xi})$. Furthermore, the term $\mathbf{B}_i$ defines the binding set, $\mathbf{F}_i$ the free set, and $\mathbf{S}_i = [\nabla^2\mathbf{G}(\mathbf{u}_i)]^{-1}_{\mathbf{F}_i}$ represents a submatrix of the inverse Hessian along the direction of free variables $\mathbf{F}_i$, cf. Choudhury and Scherer (2016). For the optimization, two stop criteria are used in parallel: One stop criterion is the number of iteration steps, whereas the other one applies if the differential change of the cost function $U(\boldsymbol{\xi})$ becomes smaller than a specified threshold.

6. RESULTS AND DISCUSSION

The effectiveness of the concept is assessed by means of Matlab simulations. The scenario used for this purpose is shown in Fig. 4. A map of the real environment with self-defined obstacles is employed. As can be seen from the scenario, the road lane on which the autonomous vehicle should drive is blocked by several objects. In addition, a pedestrian island is not perceived by the object detection. The method proposed in Sect. 4 enables the operator to plan a corridor according to his perception. By placing a pose to avoid the undetected pedestrian island and specifying the point where the vehicle should return to the original lane, the operator manages to generate a plausible shape of the corridor in compliance with his situation analysis through two actions. Accordingly, the operator plans a corridor in the region of the opposing traffic that is located behind the pedestrian island and, between two obstacles, finally goes back to the own lane. It can be seen that the corridor comprises three obstacles. Consequently, the initial path, i.e., the centre line of the corridor, is collision-afflicted.

The constrained CHOMP algorithm can now compute an optimal path using the corridor bounds as inequality constraints. As a result, the iterative path optimization

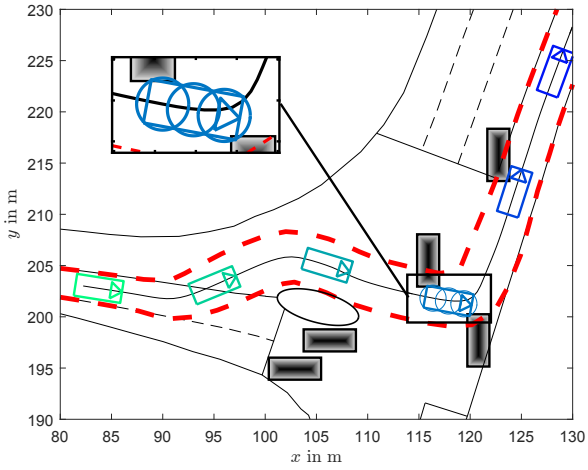


Fig. 4. Operator-specified corridor with a collision-afflicted initial path.

generates a smooth and collision-free path in the corridor specified by the operator. The optimal path with respect to the corridor boundaries and consideration of the obstacles is shown in Fig. 5. The corresponding evolution of the objective function $U(\boldsymbol{\xi})$ is illustrated in Fig. 6. Here, the optimization terminated after 91 iteration steps due to a fulfilled stop criterion w.r.t. a minimal change of the objective function $U(\boldsymbol{\xi})$. This resulted in an average computation time of 0.14 sec using an Intel i5-8350U 1.7 GHz processor.

The simulation results show that the corridor-based approach enables the human operator to easily generate optimal solutions. Due to the nature of the corridor specification, the proposed approach corresponds to a waypoint-based control. Unlike the waypoint-based control, however, the human operator himself is not responsible for creating collision-free paths. In this concept, the operator merely specifies the search space, i.e., the admissible corridor, in which the optimal solution has to be determined. Thanks to the subsequent generation of collision-free paths by the algorithm, the human operator is relieved during the teleoperation process.

Optimization in system processes takes time. Nevertheless, as shown by the result of the chosen simulation example, it is in the range of 0.1 s in this scenario. This is almost negligible and eliminates the necessity for the human operator to find collision-free paths by himself. This would turn out to be time consuming due to the lack of a three-dimensional perception, as pointed out in Sect. 3.1.

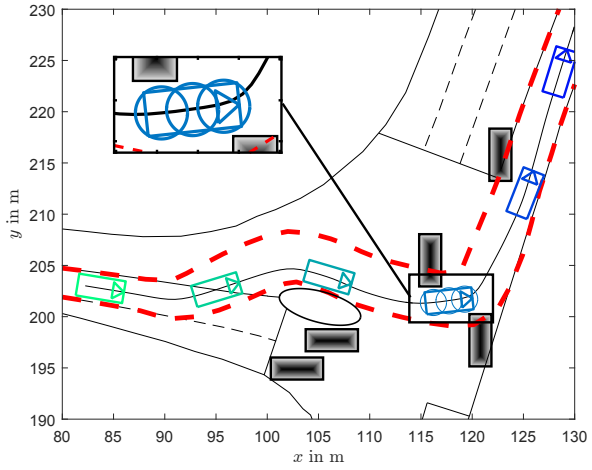


Fig. 5. Optimal path located fully in the admissible corridor specified by the operator.

7. CONCLUSIONS AND FUTURE WORK

This paper presents a concept for a teleoperated driving to disburden a human operator. Derived from the methodology of shared autonomy, this approach advantageously combines both human-predefined corridors and automated driving functions. The proposed system uses a spline-based corridor specification and the constrained CHOMP algorithm, a dual projected Newton method, to determine optimal paths that the autonomous vehicle is able to follow without collisions. The algorithm has been investigated properly within a simulation environment, corresponding

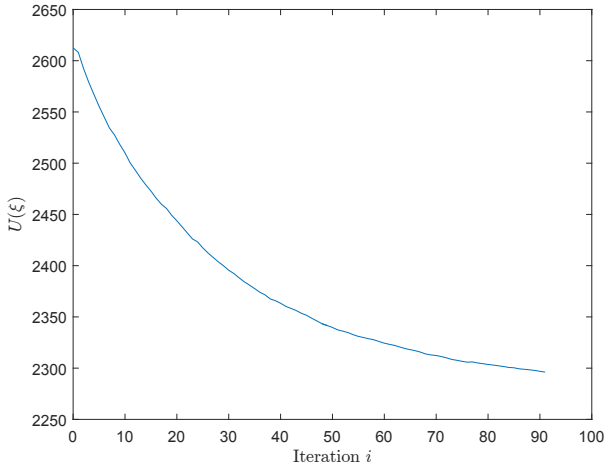


Fig. 6. Evolution of the objective function $U(\boldsymbol{\xi})$ at each iteration i.

to a realistic scenario where the vehicle successfully avoids all obstacles and complies with all path boundaries. The results confirm that the proposed approach provides the envisaged outcome, despite its simplified assumptions.
Future work will focus on improving the approach by addressing the maximum curvature that is drivable by the vehicle in the path optimization. Furthermore, investigations will be carried out on how to recover from falling into local regions by the dual projected Newton method and how to enable backward driving as a collision-free maneuver. Additionally, the algorithm will be extended w.r.t. dynamic objects in the trajectory calculation. Finally, the improved algorithm will be integrated and evaluated in a real test vehicle.

REFERENCES

Bajracharya, M., Maimone, M.W., and Helmick, D. (2008). Autonomy for mars rovers: Past, present, and future. *Computer*, 41(12), 44–50.

Bosshard, P.F. (2015). Investigation of trajectory optimization for multiple car-like vehicles. Technical report, School of Information Science, Computer and Electrical Engineering, Halmstad University.

Choudhury, S. and Scherer, S. (2016). Constrained chomp using dual projected newton method. Technical report, Carnegie Mellon University, Pittsburgh, PA.

Cooper, B. (1998). Driving on the surface of mars using the rover control workstation. Technical report, Jet Propulsion Laboratory, National Aeronautics and Space.

David, J., Valencia, R., Philippsen, R., Bosshard, P., and Iagnemma, K. (2017). Gradient based path optimization method for autonomous driving. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4501–4508.

de Visser, E.J., Pak, R., and Shaw, T.H. (2018). From 'automation' to 'autonomy': the importance of trust repair in human–machine interaction. *Ergonomics*, 61(10), 1409–1427.

Dragan, A.D., Ratliff, N.D., and Srinivasa, S.S. (2011). Manipulation planning with goal sets using constrained trajectory optimization. *IEEE International Conference on Robotics and Automation*, 4582–4588.

Ferrell, W.R. and Sheridan, T.B. (1967). Supervisory control of remote manipulation. *IEEE Spectrum*, 4(10), 81–88.

Gnatzig, S. (2015). *Trajektorienbasierte Teleoperation von Straßenfahrzeugen auf basis eines Shared-Control-Ansatzes (in German)*. Ph.D. thesis, Technische Universität München.

Gnatzig, S., Chucholowski, F., Tang, T., and Lienkamp, M. (2013). A system design for teleoperated road vehicles. *ICINCO (2)*, 231–238.

Gnatzig, S., Schuller, F., and Lienkamp, M. (2012). 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*, 913–918.

Graf, G. (2019). Combining direct and indirect control for teleoperated autonomous vehicle. *Conference on Human Factors in Computing Systems*.

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

Kay, J. (1997). Stripe: Remote driving using limited image data. Technical report, Carnegie Mellon University - Computer Science Department.

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

Piazzi, A., Bianco, C.G.L., and Romano, M. (2007). Smooth path generation for wheeled mobile robots using η^3-splines. *IEEE Transactions on Robotics*, 1089–1095.

Pongrac, H. (2008). *Gestaltung und Evaluation von virtuellen und Telepräsenzsystemen an Hand von Aufgabenleistung und Präsenzempfinden (in German)*. Ph.D. thesis, Universität der Bundeswehr.

Ratliff, N., Zucker, M., Bagnell, J.A., and Srinivasa, S. (2009). Chomp: Gradient optimization techniques for efficient motion planning. *IEEE International Conference on Robotic and Automation*, 489–494.

Reuter, J. (1998). Mobile robots trajectories with continuously differentiable curvature: an optimal control approach. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1, 38–43.

Sheridan, T.B. (1989). Telerobotics. *Automatica*, 25(4), 487–507.

Sheridan, T.B. (1993). Space teleoperation through time delay: Review and prognosis. *IEEE Transactions on robotics and Automation*, 9(5), 592–606.

Tzafestas, C.S. (2007). Virtual and mixed reality in telerobotics: A survey. *Industrial Robotics: Programming, Simulation and Applications*, 437–470.

Winfield, A.F. (2000). Future directions in tele-operated robotics. *Telerobotic applications*, 147–163.

Zucker, M., Jun, Y., Killen, B., Kim, T.G., and Oh, P. (2013a). Continuous trajectory optimization for autonomous humanoid door opening. *IEEE Conference on Technologies for Practical Robot Applications*, 1–5.

Zucker, M., Ratliff, N., Dragan, A.D., Pivtoraiko, M., Klingensmith, M., Dellin, C.M., Bagnell, J.A., and Srinivasa, S.S. (2013b). Chomp: Covariant hamiltonian optimization for motion planning. *The International Journal of Robotics Research*, 32(9-10), 1164–1193.