AO-CAR: TRANSFER OF SPACE TECHNOLOGY TO AUTONOMOUS DRIVING WITH THE USE OF WORHP*

Laura Sommer, Matthias Rick, Andreas Folkers, Christof Büskens

WG Optimization and Optimal Control, Center for Industrial Mathematics,
University of Bremen,
Bibliothekstraße 5, 28359 Bremen, Germany

ABSTRACT

Online optimization and trajectory planning are key aspects of autonomous deep space missions. Taking into account individual target criteria, such as time or energy optimality, any spacecraft maneuver can be traced back to a general optimal control problem. This can be transcribed into a nonlinear program by time discretization and hence be handled by the official ESA NLP solver WORHP.

The effectiveness of this approach has already been demonstrated in several DLR projects, such as in the deep space missions KaNaRiA and EnEx-CAUSE. In order to make such results immediately available for terrestrial applications, a transfer to current scientific questions is appropriate. Moreover, this would provide a test platform, increase public acceptance and help planning more detailed space missions through the knowledge gained from real world testing.

In this work, the DLR project AO-Car for controlling an autonomous vehicle in road traffic is presented as a successful example for such a transfer. The concept for trajectory planning and control based on WORHP, originally developed in the context of KaNaRiA, is successfully implemented on a research vehicle, a VW Passat GTE.

***Index Terms*—** Autonomous Driving, Model Predictive Control, Optimal Control, Trajectory Planning

1. INTRODUCTION

Mobility is certainly an essential foundation of today's society. On the one hand this results in strong points of contact in everyday life and on the other hand in a high level interest in the further enhancement of existing vehicle concepts. With regard to the latter, the design of autonomous driving features was one of the most important developments of the last decade. This is driven by promises of increased road safety [1], intelligent mobility on demand concepts [2] and cost-efficient platooning approaches for trucks [3].

*The authors would like to thank *Deutsches Zentrum für Luft- und Raumfahrt (DLR) Raumfahrtmanagement, Navigation* in Bonn-Oberkassel for providing the research vehicle [grant number 50NA1615].

Fig. 1. The test vehicle used within AO-Car.

However, the development of autonomous systems is also a key aspect in the design of deep space missions such as done within the DLR projects KaNaRiA [4] and EnEx-CAUSE [5]. As precise maneuvers cannot be performed remotely due to the long travel time of communication signals, there is a need for highly intelligent and efficient systems. At the center of these innovations is the computation of control commands which, on the one hand, have to fulfill a number of optimality criteria and, on the other hand, are characterized by a high degree of safety. These requirements can be met by solving appropriate optimal control problems for the computation of high-quality trajectories. Based on the ESA solver WORHP [6] for nonlinear optimization problems and the corresponding transcription method TransWORHP [7], this has proven to be a very successful approach, for example within the simulated environments of KaNaRiA [8] and EnEx-CAUSE [9].

The same algorithms are applied to autonomous driving within the scope of the project AO-Car, handling the exemplary task of exploring a parking lot. As a result, these technologies become publicly available and a framework for their improvement through real-world testing is provided. Diverse situations such as turning, overtaking, controlled stopping or parking are all reduced to a single optimal control problem. Its successive solution results in a nonlinear model predictive control (NMPC) approach, which is presented in the following. The safety and efficiency of the proposed method is demonstrated by experiments on the VW Passat GTE shown in Figure 1.

2. RELATED WORK

Most approaches to the computation of vehicle controls are limited to a specific range of applications. Algorithms for lane keeping use the evaluation of camera images to compute control commands through deep learning [10] or deep reinforcement learning [11]. Other methods strongly rely on a predefined path which is then tracked. This can be realized with a Riccati controller [12] or by using an MPC approach which is either restricted to the lateral movement [13] or takes full control of the vehicle [14, 15, 16]. A control concept for a wide range of situations is presented in [17, 18], where offline generated look-up tables are used as an initial guess for a fast online optimization of the vehicle controls.

This work presents an online nonlinear MPC scheme for full control of an autonomous vehicle independent of a predefined path. Based on the continuous definition of target positions, this approach is capable of handling a great variety of situations through its general description of the underlying problem.

3. TRAJECTORY PLANNING AND CONTROL APPROACH

In order to calculate control commands for the vehicle, every possible scenario is traced back to one uniform task: *Transfer the vehicle from state A to state B respecting its dynamics without any collision and minimizing an individual optimization criterion*. This allows each maneuver to be formulated as an optimal control problem with *process time* T of the form

$$\begin{aligned} \min_{z,u,T} \quad & J(z,u,T) \\ \text{s.t. } & \dot{z}(t) = f(z(t),u(t)), \\ & z(0) = z_0, \quad z(T) = z_T, \\ & z_{\min} \le z(t) \le z_{\max}, \\ & u_{\min} \le u(t) \le u_{\max}, \\ & C(z(t),u(t),t) \le 0 \text{ for all } t \in [0,T]. \end{aligned} \tag{OCP}$$

The *objective* J, *states* z, *controls* u, *constraints* C and *dynamics* f with

$$\begin{aligned} J &\in \mathcal{C}^1(\mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}, \mathbb{R}), \\ z &\in \mathcal{C}^1([0,T], \mathbb{R}^n), \\ u &\in \mathcal{C}^0([0,T], \mathbb{R}^m), \\ C &\in \mathcal{C}^1(\mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}, \mathbb{R}^c), \\ f &\in \mathcal{C}^1(\mathbb{R}^n \times \mathbb{R}^m, \mathbb{R}^n), \end{aligned}$$

specify the behavior of the system.

For applications on a parking lot, only speeds below $20\,\mathrm{km/h}$ must be taken into account. Therefore we do not expect large forces in interaction with the vehicle and choose a kinematic single track model [19] to describe the dynamic behavior. Within this modeling approach the front and rear wheels are respectively combined to single wheels. Figure 2 displays the components of the utilized model.

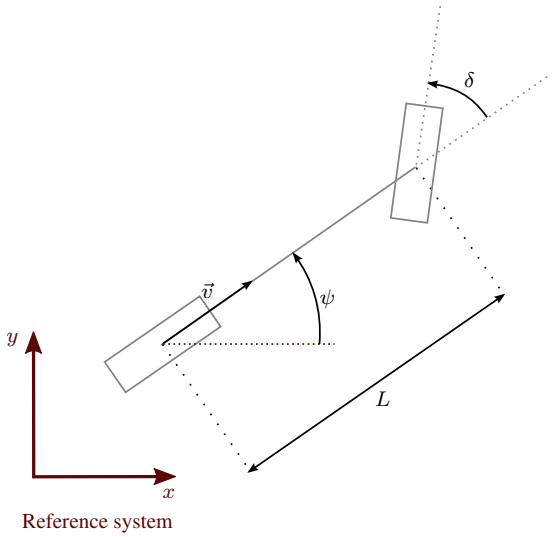

Fig. 2. Kinematic single track model.

The states x, y and $v = \|\vec{v}\|$ represent the *global position* and the *speed* of a reference point at the center of the rear axle. The length of the *wheelbase* is given by L. The angle ψ describes the vehicles *orientation* and is influenced by its *steering angle* δ. The first and second derivative of the latter are the *steering angle velocity* ω_δ and *steering angle acceleration* a_δ. The speed is affected by the *acceleration* a and its derivative, the *jerk* j. By geometric considerations one can derive a first order system of differential equations to describe the vehicle dynamics, given as

$$\begin{aligned} \dot{x} &= v \cdot \cos(\psi), & \dot{\psi} &= v \cdot \frac{\tan(\delta)}{L}, \\ \dot{y} &= v \cdot \sin(\psi), & \dot{\delta} &= \omega_\delta, \\ \dot{v} &= a, & \dot{\omega}_\delta &= a_\delta, \\ \dot{a} &= j. \end{aligned}$$

3.1. Objective Function

The aim of the trajectory planning is to ensure comfortable, efficient and above all safe driving. To provide smooth controls, we include acceleration, jerk, steering angle velocity and steering angle acceleration in the objective function. Furthermore we consider the term $v(t) - v_{\text{set}}$ to keep the vehicle on a desired speed of $v_{\text{set}} \ge 0\,\mathrm{km/h}$. Including the process time T, this leads to the objective

$$\begin{aligned} \tilde{J}(z,u,T) := & w_0 T + \int_0^T w_1 {\omega_\delta}^2 + w_2 a^2 + w_3 j^2 + w_4 {a_\delta}^2 \mathrm{d}t \\ & + \int_0^T w_5 (v - v_{\text{set}})^2 \mathrm{d}t. \end{aligned}$$

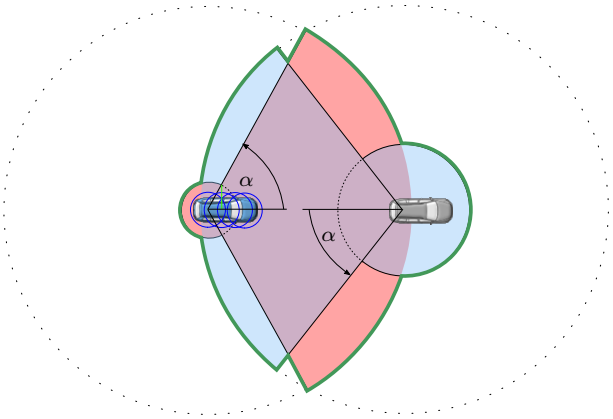

Fig. 3. Region of interest based on the positions of both the vehicle (blue) and the target (gray).

If the current target state specifies a final speed of $v_T = 0\,\mathrm{km/h}$, deviations from the desired values x_T, y_T, ψ_T and δ_T are penalized by

$$\begin{aligned} J_T(z,u,T) :=& w_6(x(T)-x_T)^2 + w_7(y(T)-y_T)^2 \\ &+ w_8(\psi(T)-\psi_T)^2 + w_9(\delta(T)-\delta_T)^2. \end{aligned}$$

In the opposite case of non-stop driving ($v_T > 0\,\mathrm{km/h}$) the weights $w_6, \ldots, w_9$ will be set to zero and we force

$$\begin{aligned} x(T) &\in [x_T - \varepsilon_x, x_T + \varepsilon_x], \\ y(T) &\in [y_T - \varepsilon_y, y_T + \varepsilon_y], \\ \psi(T) &\in [\psi_T - \varepsilon_\psi, \psi_T + \varepsilon_\psi], \\ \delta(T) &\in [\delta_T - \varepsilon_\delta, \delta_T + \varepsilon_\delta], \end{aligned}$$

with small approximation tolerances $\varepsilon_x, \varepsilon_y, \varepsilon_\psi, \varepsilon_\delta > 0$. The overall objective function

$$J(z,u,T) = \tilde{J}(z,u,T) + J_T(z,u,T)$$

comprises all these criteria which are partially conflicting with each other. Therefore the result is a compromise. Depending on the scenario, an appropriate weighting is selected. This offers the possibility of a uniform formulation and solution for a large variety of situations with different requirements.

3.2. Constraints

Limitations of states and controls for comfort reasons as well as physical limits are considered by

$$\begin{aligned} v_{\min} \leq\ & v \leq v_{\max}, \\ a_{\min} \leq\ & a \leq a_{\max}, \\ \delta_{\min} \leq\ & \delta \leq \delta_{\max}, \\ \omega_{\delta_{\min}} \leq\ & \omega_\delta \leq \omega_{\delta_{\max}}. \end{aligned}$$

(a) © GEO-BASIS/BKG 2009. (b) Predefined restrictions.

Fig. 4. Parking lot for conducting the exploration experiments (left) and the resulting prior knowledge about its geometry (right).

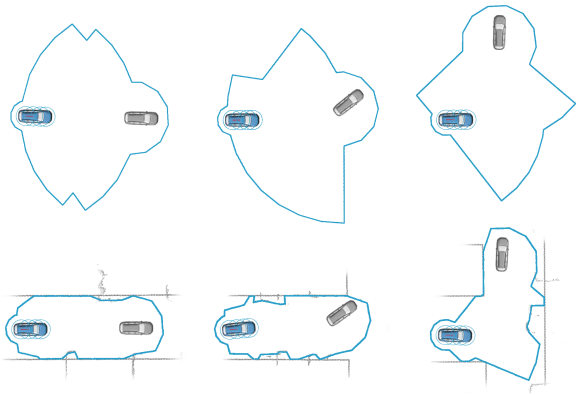

Fig. 5. Comparison of different feasibility polygons. Upper row: without consideration of obstacles; lower row: including predefined restrictions of the parking lot as well as detected obstacles.

The function C in (OCP) takes further restrictions into account, such as collision avoidance. For this, the feasible region with respect to the states x and y is provided by a polygon. Furthermore, the expansion of the vehicle in the XY-plane is approximated by four circles covering the entire body including the exterior mirrors and a safety distance, as demonstrated in Figure 3. With this, the collision avoidance within C is implemented by forcing the circles to be inside of the feasibility polygon.

For the construction of the latter, a region of interest with respect to the vehicle and target position is defined as the union of circle segments with different radii, as shown in Figure 3. The two main segments are based on the distance between vehicle and target position defining the radius of the circles. Furthermore, they are considered to have an opening angle of 2α in the direction of the respective other position. In addition to that, around both locations a second circle is constructed with a smaller size. Finally, the region of interest results from the union of the four segments. This is then adapted to the perceived map of obstacles by a ray casting

approach, resulting in the feasibility polygon. The map consists of predefined restrictions of the parking lot, as shown in Figure 4, as well as obstacles detected at runtime. In case of entering a parking space, the boundaries of the lane are not considered. Examples of feasibility polygons during the exploration of the parking lot are provided in Figure 5.

3.3. Solution of the OCP

Most optimal control problems of the form (OCP) cannot be solved analytically. However, there are two ways to address them numerically: *indirect* and *direct* methods. With the former approach, (OCP) is transformed into a boundary value problem according to Pontryagin's maximum principle. This is usually difficult to solve as a very good initial guess is required which is hard to find for the corresponding adjoints. Within the direct approach, the optimal control problem is discretized in time and then transformed into a nonlinear optimization problem of the form

$$\begin{aligned} \min_{x\in\mathbb{R}^N} \quad & F(x) \\ \text{s.t.} \quad & G(x) \leq 0, \end{aligned} \tag{NLP}$$

where $F : \mathbb{R}^N \to \mathbb{R}$ and $G : \mathbb{R}^N \to \mathbb{R}^M$. These problems are well studied and can be addressed with the ESA NLP solver WORHP [6]. This software is designed to efficiently solve huge problems (up to millions of variables) in a short amount of time and it is successfully applied to a wide range of (space) applications (including [8, 20]). The underlying SQP algorithm takes advantage of the sparsity of the gradient, Jacobian and Hessian to accelerate the calculation.

This makes optimal control problems ideally suited to be solved with WORHP, since the matrices occurring due to the discretization of their dynamics are usually highly sparse. The transcription of (OCP) to (NLP) is performed by the software library TransWORHP [7] which uses direct approaches like the trapezoidal method, multiple shooting techniques or pseudospectral methods.

3.4. Initial Guess

To find an optimal solution, the OCP solver requires a suitable initial guess for all states, controls and the process time. For this we use Reeds-Shepp paths [21] for most of the states. These result in the shortest path between two poses with respect to a combination of different segment types, consisting of circular arcs of a fixed radius and tangents connecting them. Figure 6 shows two examples of Reeds-Shepp paths. Based on the resulting shortest path, we determine matching values for the acceleration a and the corresponding process time T in an analytic way. Depending on the respective direction of travel, the maximum (minimum) possible acceleration is applied until the maximum (minimum) permissible speed is reached. Finally this results in profiles for a and the steering angle δ, from which the speed v, the coordinates x and y as

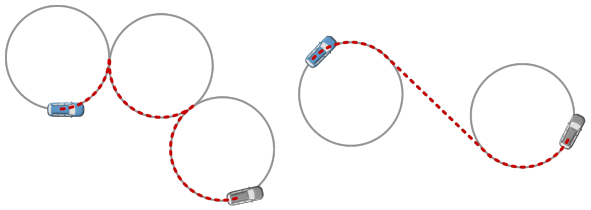

Fig. 6. Examples of Reeds-Shepp paths.

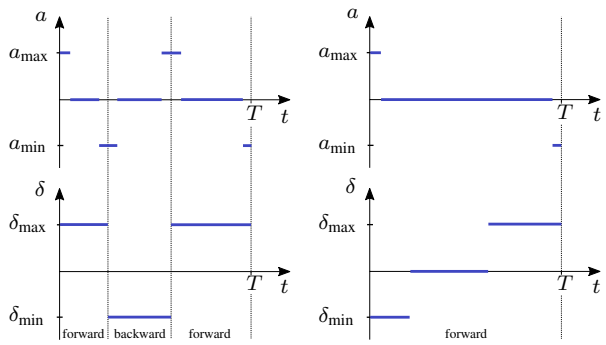

Fig. 7. Acceleration a and steering angle δ resulting from the Reeds-Shepp paths in Figure 6.

well as the orientation ψ can be deduced. The corresponding profiles of a and δ for the paths from Figure 6 are shown in Figure 7. As one can see, a and δ are discontinuous; since there are no better assumptions we use zero as initial guess for their derivatives j as well as ω_δ and α_δ.

3.5. Nonlinear Model Predictive Control Approach

In order to ensure permanent optimality and to compensate for disturbances, we repeatedly solve (OCP) using the methods of the preceding sections and only execute the first few control values. Within this nonlinear model predictive controller, the optimal solution of the previous step is used as an initial guess for the next computation. Regarding the very first optimization and in situations where no optimal solution could be determined in the previous step, the Reeds-Shepp approach for calculating an initial guess is used as described in Section 3.4.

4. AUTONOMOUS PARKING LOT EXPLORATION

4.1. System Overview

We have tested our approach in various real-life scenarios that occur during the exploration of a parking lot with the modified VW Passat GTE displayed in Figure 1. For that, the electric motor was used in all tests. The control values (longitudinal acceleration and steering angle) determined through our method are transmitted to the actuators of the vehicle at a frequency of $50\,\mathrm{Hz}$. All computations were carried out on

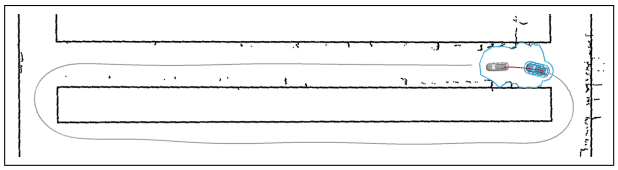

Fig. 8. Part of an exploration of the parking lot. Black dots and lines: perceived obstacles and prior knowledge about the parking lot. Gray line: actual path taken during the experiment; red line: trajectory calculated by the model predictive control algorithm at the current time step.

a *Spectra Powerbox 1295* (Intel Core i7-5850EQ 2.7 GHz, 8 GB DDR3-RAM).

Our algorithms are provided with information from the surroundings of the vehicle by forward and backward oriented laser scanners, each consisting of four layers, and a set of ultrasonic sensors. Camera systems have not been used in any way for our experiments. We utilize an INS/GNSS system with an Extended Kalman Filter as described in [22] for localization and a real-time kinematic (RTK) system to correct deviations in GNSS measurements.

During the practical realization, a superordinate algorithm frequently determines a target position up to 15 m ahead of the vehicle. With model predictive control and in consideration of obstacles we head for that target until an update is given.

4.2. Experiments and Evaluation

We show the performance of the proposed control algorithm for three situations: exploring a part of the parking lot, navigating through a narrowing of the lane and parking. All these scenarios are characterized by close proximity to obstacles and thus by high demands on the accuracy of the controller. The parking maneuver further requires a strong curvature of the resulting trajectories.

The driven path while exploring the parking lot with nonlinear model predictive control is shown in Figure 8. In addition, the obstacles detected from the view point of the current vehicle's position as well as the visibility polygon are displayed. At any time, the vehicle is able to maintain the minimum safety distance represented by the circles shown in blue. Furthermore the controller keeps the vehicle driving in the center of the lane or, depending on the position of other parked vehicles, slightly offset. Especially the turning maneuvers meet high requirements in terms of precision.

Fig. 9. Autonomous passing of a narrowing of the lane.

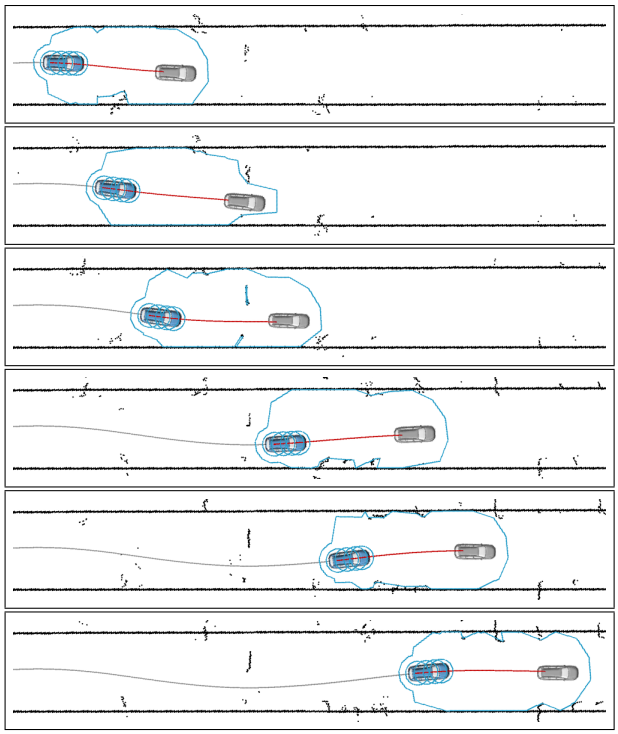

Fig. 10. Test vehicle driving through a narrowing of the lane. Black dots and lines: perceived obstacles and prior knowledge about the parking lot. Gray line: actual path taken during the maneuver; red line: trajectory calculated by the model predictive control algorithm at the current time step.

More advanced situations like obstacle avoidance can be evaluated by narrowing the lane as shown in Figure 9. To handle these conditions in a safe and comfortable way, the target position is shifted from the center of the lane to the opening of the narrowing as early as possible. This results in very smooth paths for both the planned and the executed trajectory as displayed in Figure 10. Conversely, only after passing the obstacle the target position is shifted back to the middle of the lane.

As soon as a free parking space has been detected, the predefined restrictions of the lane are disregarded in the computation of the feasibility polygon, see also Section 3.2. Instead, additional boundaries based on the geometry of the detected parking space are considered. The upper left part of Figure 11 shows the obstacle map used while exploring; the resulting perception during the subsequent parking maneuver

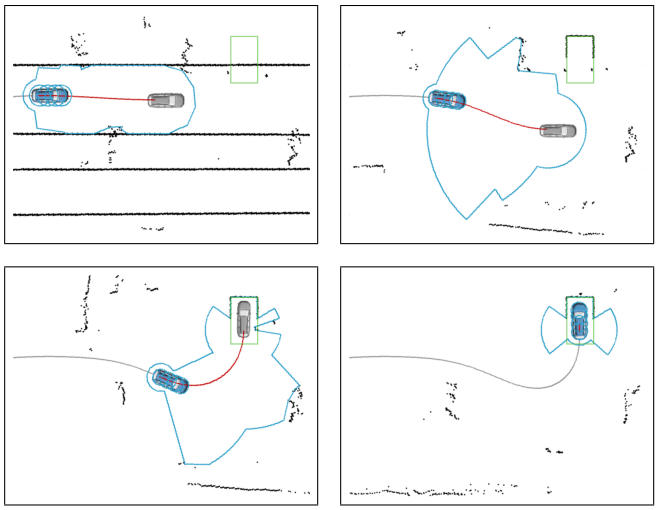

Fig. 11. Test vehicle entering a parking space. Black dots and lines: perceived obstacles and prior knowledge about the parking lot. Gray line: actual path taken during the entry; red line: trajectory calculated by the model predictive control algorithm at the current time step; green box: detected parking space.

is displayed in the remaining panels. At the beginning of the maneuver, the positioning of the target includes a small shift to the opposite side of the lane, as shown in the upper right panel. This supports the vehicle to enter via a smooth arc. Furthermore, the problem description is amended so that strict constraints are replaced by penalty terms as described in Section 3.1. Hence, while approaching the parking space, the solutions of (OCP) do not fulfill strict requirements in the deviation of x, y, ψ and δ with respect to the requested target state. However, at the final entry, a high precision of the final state is enforced by the newly introduced boundary. As a result, the orientation and position of the vehicle perfectly matches the detected parking space after completion of the maneuver. Figure 12 shows a snapshot of the autonomous parking process.

All trajectories during the parking lot exploration, including obstacle avoidance and parking, are computed on the basis of 21 discrete points in time. This results in a total of 190 variables subject to 308 constraints to be optimized in each control step. In case no optimal or feasible solution was found after 100 ms we relax the problem by disregarding the optimality criterion. Hence, only feasible solutions can be obtained in this case. A computation is finally aborted as unacceptable if a maximum duration of 500 ms is exceeded or the SQP method terminates with an error. Table 1 summarizes the computation times and further performance information.

The solution of a single (OCP) during the exploration phase takes 33 ms in average, resulting in a transmission of about two control signals before a new solution is available. Only a few computations did not lead to an optimal result,

Fig. 12. Autonomous entering of a parking space.

however most of them provide at least a feasible solution. In case of the more challenging maneuver of driving through a narrowing of the lane, the mean computation time of all optimal solutions slightly increases to 35 ms. In addition, the rate of unacceptable solutions is higher.

The problem formulation based on the penalty approach results in all solutions determined during parking being at least feasible. However, in average the computations take twice as long as in the case of ordinary exploration. Hence, for a more reliable control when entering a parking space, the time to compute a new trajectory should be significantly reduced by further development of the algorithm. This also addresses the noticeably larger standard deviation in computation times when executing more challenging maneuvers in general. However, even in the case of an increased computing time, reasonable control signals are available since an entire trajectory is provided from the previous optimal solution.

Altogether, the method described reliably leads to a safe control of the autonomous vehicle in all test cases. A video containing the corresponding maneuvers is available at:
`www.math.uni-bremen.de/zetem/aocar`

	Exploration	**Narrowing**	**Parking**
Mean computing time			
optimal	33 ms	35 ms	70 ms
non-optimal	104 ms	251 ms	-
total	33 ms	41 ms	70 ms
Standard deviation			
optimal	11 ms	19 ms	49 ms
non-optimal	121 ms	169 ms	-
total	17 ms	49 ms	49 ms
# Computations	2 486	750	480
# Optimal	2 462	725	468
# Not optimal (feasible)	18	9	12
# Not optimal (unacc.)	6	16	0
Rate of unacc. solutions	0.24 %	2.13 %	0 %
# Relax	2	4	12

Table 1. Performance of the model predictive controller in different scenarios.

5. CONCLUSION

We presented the software system for autonomously driving vehicles developed as part of the project AO-Car and derived from space applications. The need for optimal trajectories for various types of maneuvers was attributed to a uniform description through an optimal control problem. Based on its frequent solution, a model predictive control algorithm was introduced. This allows high-quality control commands to be provided in a short computation time, taking into account both arbitrary surroundings of the vehicle and a nonlinear model of its dynamics. For the exemplary tasks of exploring a parking lot, handling a narrowing of the lane and entering a parking space, we have shown the versatility of the approach in the field of autonomous driving.

A. APPENDIX

Physical parameters		
wheelbase	L	$2.786\,\mathrm{m}$
min / max velocity	$v_{\min}$ / $v_{\max}$	$0\,\mathrm{km/h}$ / $8\,\mathrm{km/h}$
min / max acceleration	$a_{\min}$ / $a_{\max}$	$-2.5\,\mathrm{m/s^2}$ / $2.5\,\mathrm{m/s^2}$
min / max steer. angle	$\delta_{\min}$ / $\delta_{\max}$	$-0.55\,\mathrm{rad}$ / $0.55\,\mathrm{rad}$
min / max steer. angle vel.	$\omega_{\delta_{\min}}$ / $\omega_{\delta_{\max}}$	$-0.3\,\mathrm{rad/s}$ / $0.3\,\mathrm{rad/s}$
Weights in objective		
process time	w_0	0.033
steering angle velocity	w_1	0.483
acceleration	w_2	0.483
jerk	w_3	0.5
steering angle acc.	w_4	0.1
keep velocity	w_5	0.1
x deviation	w_6	10
y deviation	w_7	10
ψ deviation	w_8	1
δ deviation	w_9	1
Tolerances in target state		
x deviation	ε_x	0.1
y deviation	ε_y	0.1
ψ deviation	ε_ψ	0.2
δ deviation	ε_δ	0.2
Overlapping circles		
radius		$1.30\,\mathrm{m}$
Polygon		
opening angle	α	$0.975\,\mathrm{rad}$
Optimization tolerances		
optimality		10^{-6}
feasibility		10^{-6}

Table 2. Setting of the hyperparameters in the experiments.

B. REFERENCES

[1] T. Winkle, "Safety benefits of automated vehicles: Extended findings from accident research for development, validation and testing," in *Autonomous Driving: Technical, Legal and Social Aspects*, M. Maurer, J. C. Gerdes, B. Lenz, and H. Winner, Eds., pp. 335–364. Springer Berlin Heidelberg, 2016.

[2] M. Pavone, "Autonomous mobility-on-demand systems for future urban mobility," in *Autonomous Driving: Technical, Legal and Social Aspects*, M. Maurer, J. C. Gerdes, B. Lenz, and H. Winner, Eds., pp. 387–402. Springer Berlin Heidelberg, 2016.

[3] H. Flämig, "Autonomous vehicles and autonomous driving in freight transport," in *Autonomous Driving: Technical, Legal and Social Aspects*, M. Maurer, J. C. Gerdes, B. Lenz, and H. Winner, Eds., pp. 365–385. Springer Berlin Heidelberg, 2016.

[4] A. Probst, G. González Peytaví, D. Nakath, A. Schattel, C. Rachuy, P. Lange, J. Clemens, M. Echim, V. Schwarting, A. Srinivas, K. Gadzicki, R. Förstner, B. Eissfeller, K. Schill, C. Büskens, and G. Zachmann, "Kanaria: Identifying the challenges for cognitive autonomous navigation and guidance for missions to small planetary bodies," in *66th International Astronautical Congress (IAC)*, Jerusalem, Israel, 2015.

[5] "EnEx-CAUSE - Autonome Exploration," `https://www.dlr.de/rd/en/desktopdefault.aspx/tabid-10786/18809_read-43917/`, visited on 2018-10-26.

[6] C. Büskens and D. Wassel, "The ESA NLP solver WORHP," *Modeling and Optimization in Space Engineering*, vol. 73, no. 11-12, pp. 85–110, 2013.

[7] M. Knauer and C. Büskens, "From WORHP to TransWORHP," in *Proceedings of the 5th International Conference on Astrodynamics Tools and Techniques*, 2012.

[8] A. Schattel, A. Cobus, M. Echim, and C. Büskens, "Optimization and sensitivity analysis of trajectories for autonomous small celestial body operations," in *Progress in Industrial Mathematics at ECMI 2016*. 2017, pp. 705–711, Springer International Publishing.

[9] C. Meerpohl, K. Flaßkamp, and C. Büskens, "Optimization strategies for real-time control of an autonomous melting probe," in *American Control Conference (ACC)*, Milwaukee, WI, USA, 2018.

[10] M. Bojarski, P. Yeres, A. Choromanska, K. Choromanski, B. Firner, L. Jackel, and U. Muller, "Explaining how a deep neural network trained with end-to-end learning steers a car," 2017.

[11] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley, and A. Shah, "Learning to drive in a day," 2018.

[12] D. Kim, J. Kang, and K. Yi, "Control strategy for high-speed autonomous driving in structured road," *International IEEE Conference on Intelligent Transportation Systems*, pp. 186–191, 2011.

[13] F. Borrelli, P. Falcone, T. Keviczky, J. Asgari, and D. Hrovat, "MPC–based approach to active steering for autonomous vehicle systems," *International Journal of Vehicle Autonomous Systems*, vol. 3, pp. 265–291, 2005.

[14] P. Falcone, F. Borrelli, J. Asgari, and D. Hrovat, "Low complexity MPC schemes for integrated vehicle dynamics control problems," *International Symposium on Advanced Vehicle Control*, 2008.

[15] T. Faulwasser, B. Kern, and R. Findeisen, "Model predictive path-following for constrained nonlinear systems," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, Dec 2009, pp. 8642–8647.

[16] S. Yu, X. Li, H. Chen, and F. Allgöwer, "Nonlinear model predictive control for path following problems," *IFAC Proceedings Volumes*, vol. 45, no. 17, pp. 145 – 150, 2012.

[17] D. Ferguson, T. Howard, and M. Likhachev, "Motion planning in urban environments," *Journal of Field Robotics*, vol. 25, no. 11-12, pp. 939–960, 2008.

[18] T. Howard and A. Kelly, "Optimal rough terrain trajectory generation for wheeled mobile robots," *International Journal of Robotics Research*, vol. 26, no. 2, pp. 141–166, February 2007.

[19] A. De Luca, G. Oriolo, and C. Samson, "Feedback control of a nonholonomic car-like robot," in *Robot Motion Planning and Control*, chapter 4. Springer, 1998.

[20] A. Cobus, M. Rick, L. Sommer, N. Backfisch, A. Probst, M. Echim, and C. Büskens, "Optimal control in autonomous driving," in *Proceedings in Applied Mathematics and Mechanics*, 2017, vol. 17.

[21] J. A. Reeds and L. A. Shepp, "Optimal paths for a car that goes both forwards and backwards.," *Pacific J. Math.*, vol. 145, no. 2, pp. 367–393, 1990.

[22] J. Clemens and K. Schill, "Extended Kalman filter with manifold state representation for navigating a maneuverable melting probe," in *19th International Conference on Information Fusion (FUSION)*. jul 2016, pp. 1789–1796, IEEE.