Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 52-8 (2019) 182–187

Autonomous Driving Based on Nonlinear Model Predictive Control and Multi-Sensor Fusion ⋆

M. Rick * **J. Clemens** ** **L. Sommer** * **A. Folkers** * **K. Schill** ** **C. Büskens** *

* *Center for Industrial Mathematics,*
** *Cognitive Neuroinformatics,*
University of Bremen, 28359 Bremen, Germany,
{*mrick, jclemens, lsommer, afolkers, kschill, bueskens*}*@uni-bremen.de.*

Abstract: Algorithms for controlling fully autonomous systems must meet especially high requirements with respect to safety and robustness. A particularly challenging example are autonomous deep space missions, which we investigated in several projects. In this context, we showed that a safe and robust autonomous system can be realized through nonlinear model predictive control approaches using optimization techniques in combination with multi-sensor fusion based on an extended representation of uncertainty. The focus of this paper is on demonstrating the versatility of that concept by transferring the corresponding algorithms to the also very challenging application of autonomous driving. In particular, we propose a system concept for a self-driving car based on our methodology. Furthermore, we present results of a real world research vehicle that autonomously explores a parking lot, dynamically takes obstacles into account, and finally performs a parking maneuver.

Keywords: Autonomous vehicles, Automotive control, Nonlinear model predictive control, Vehicle models, Multi-sensor fusion, Evidential mapping

1. INTRODUCTION

The development of autonomous vehicles is an ongoing process in which remarkable results were already achieved in the 1990s (Dickmanns, 1997). Since then, considerable effort has been put by the industry and science into continuously improving this technology, which will ultimately enable new mobility concepts, increase road safety, and achieve a higher degree of automation of logistical processes (Maurer et al., 2016).

To realize a fully autonomous system, the collaboration of experts from several fields of science is crucial. For example, the safe maneuvering of a vehicle should, among others, be based on an accurate estimate of its own state and of its environment as well as a subsequent computation of robust control commands. This results in highly sophisticated algorithms as for example developed within the DARPA challenge for self-driving cars (Ferguson et al., 2008; Wille and Form, 2008). However, similar requirements also apply to entirely different areas of research, such as autonomous deep space missions like those considered in KaNaRiA (Probst et al., 2015) and EnEx-CAUSE (Clemens et al., 2018). Here, state estimation is realized using a Bayesian filter (Clemens and Schill, 2016; Nakath et al., 2018), while the control approach is based on numerical optimization for solving optimal control problems (Schattel, 2018; Meerpohl et al., 2018). The result is not only highly efficient but most noticeably also applicable to other scenarios such as the safe and robust control of an autonomous vehicle. This was demonstrated in the project AO-Car. Here, the algorithms originating from space applications were adapted in the time frame of one year to the exemplary scenario of autonomous parking lot exploration using a research vehicle. In the further course of this work, we present the main components of the resulting system and demonstrate its capabilities.

Fig. 1. The research vehicle used in AO-Car during the exploration of the parking lot of the University of Bremen.

2. RESEARCH VEHICLE AND SYSTEM OVERVIEW

The proposed system was implemented on the modified Volkswagen Passat GTE shown in Fig. 1. In the following, we give a short overview of the essential hardware and software components.

2.1 Hardware Components of the Research Vehicle

One aim during planning and realizing the modifications of the vehicle was that they are as close as possible to production in terms of hardware selection and visual integration. The original

⋆ The authors would like to thank the Federal Ministry for Economic Affairs and Energy of Germany (BMWi) and the German Aerospace Center (DLR) Space Administration for supporting this work (grant nos. 50 NA 1615 and 50 NA 1909).

10.1016/j.ifacol.2019.08.068

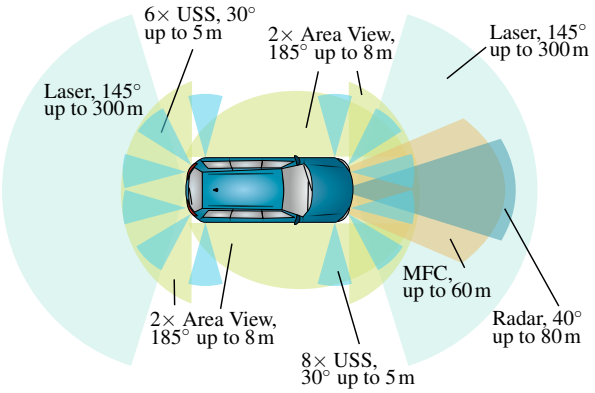


Fig. 2. Field of view of mounted sensors.

sensor system, consisting of 360° cameras (Area View), ultrasonic sensors (USS), a multifunction camera (MFC) and radar, was extended by two Ibeo ScaLa laser scanners. The coverage of all sensors is shown in Fig. 2. However, due to the short project duration, camera and radar data is not considered yet. In addition, a consumer global navigation satellite system (GNSS) receiver was installed, which computes RTK-supported position estimates, and a low-cost inertial measurement unit (IMU) provides the accelerations and the turning rates of the vehicle.

Two computers (Spectra PowerBox) were installed for conducting all computations, while acceleration and steering wheel angle values are used to control the vehicle. The control signals are transmitted via the interface originally used by the Volkswagen driver assistance systems, which provides additional safety features. Furthermore, a safety driver has the possibility to manually oversteer the system (longitudinal and lateral) at any time.

2.2 Overview of the Software System

The proposed system can be divided into three main modules: *Multi-Sensor Fusion* (FUS), *Decision Making* (DM) and *Control* (CTRL). An overview is shown in Fig. 3. The FUS module fuses the sensor data in order to estimate the current state of the vehicle and to aggregate information about its environment. The results are used by DM to decide on the vehicle's behavior and to define an appropriate target state at every time step. Finally, CTRL computes corresponding control commands based on a nonlinear model predictive control approach. The single components are discussed in the following sections.

3. MULTI-SENSOR FUSION

The FUS module fuses all sensor data in order to localize the vehicle, to create a map of the environment, and to detect obstacles in the vicinity of the vehicle. All three tasks are described in the following.

3.1 Localization

The goal of localization is to estimate the state x_t of the vehicle based on absolute measurements z_t and relative state transition measurements u_t. The latter contain IMU data (acceleration and yaw rate) and the steering angle, which are used to predict the state x_t based on the previous estimate x_{t-1}. Afterwards, x_t is corrected with respect to absolute references using z_t, which contains GNSS position estimates as well as the speed reported by the vehicle. The state of the

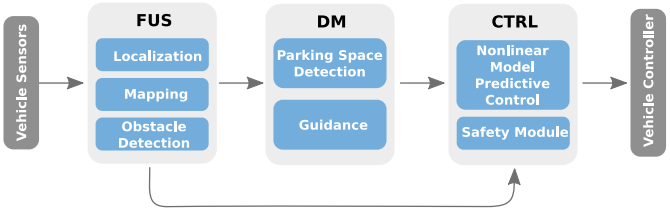


Fig. 3. Overview of the different modules.

vehicle is composed of its pose (position and orientation) and its speed. Furthermore, in order to provide a robust estimate, the bias of the accelerometer and of the gyroscope has to be estimated as well. Accordingly, the state space is defined as $\mathscr{X} := \mathbb{R}^2 \times SO(2) \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} = SE(2) \times \mathbb{R}^3$, where $SO(2)$ denotes the 2D rotation group and $SE(2)$ is the 2D special Euclidean group. The state vector $x_t \in \mathscr{X}$ is given by $x_t := (r^\top, \psi, v, \delta_a, \delta_\omega)^\top$, where $r := (r_x, r_y)^\top$ is the position, ψ is the orientation, v is the speed, and δ_a, δ_ω are the accelerometer bias and the gyroscope bias, respectively.

Solving the state estimation problem corresponds to calculating the distribution $p(x_t | z_{0:t}, u_{1:t})$ of the state x_t given all absolute measurements $z_{0:t}$ and all state transition measurements $u_{1:t}$. We assume the state posterior is a normal distribution with mean $\mu_t \in \mathscr{X}$ and covariance $\Sigma_t \in \mathbb{R}^{6\times6}$, i.e., the state is distributed corresponding to

$$x_t | z_{0:t}, u_{1:t} \sim \mathscr{N}(\mu_t, \Sigma_t). \tag{1}$$

Accordingly, we can use an *extended Kalman filter* (EKF) (Thrun et al., 2005, Sect. 3.3) to estimate the parameters μ_t, Σ_t. More precisely, we are using the implementation of Clemens and Schill (2016), which is based on the so-called $\boxplus$-method [1] (Hertzberg et al., 2013). This formalism allows a seamless integration of manifold state spaces like $\mathscr{X}$ into sensor fusion algorithms.

3.2 Mapping

The mapping submodule of FUS aims to provide a representation of the vehicle's environment. In particular, we are estimating an *evidential grid map* (Pagac et al., 1998), which is based on the belief function theory (Shafer, 1976; Smets and Kennes, 1994) and can be seen as a generalization of a classical probabilistic grid map (Thrun et al., 2005, Chap. 9). Here, each grid cell $Y_i \subseteq \Theta_Y$, $1 \leq i \leq M$ is an evidential variable, which indicates whether the corresponding area is occupied (o) or free (f). Accordingly, the frame of discernment, i.e., the set of all mutually exclusive possibilities in the domain, is defined as $\Theta_Y := \{o, f\}$. Furthermore, M denotes the total number of cells and the entire map is given by $Y \in \Theta_Y^M$. In contrast to probabilistic grid maps, one cannot only assign belief mass to the singletons of the hypotheses space but to all subsets as well, including the superset Θ_Y and the empty set $\emptyset$. By that means, evidential grid maps allow the representation and the distinction between different dimensions of uncertainty (Reineking and Clemens, 2014). For example, a lack of knowledge is represented by mass on Θ_Y, while contradictory measurements result in mass on $\emptyset$. In the case of probabilistic grid maps, both is represented by a uniform distribution $P(o) = P(f) = 0.5$, and thus, one cannot discriminate the different causes from each other. The additional information provided by evidential grid maps can for example be utilized for path planning and exploration tasks (Clemens et al., 2016).

[1] pronounced "boxplus-method"

For creating the map, we follow the methodology of *mapping with known pose* (Thrun et al., 2005, Chap. 9), i.e., the trajectory $x_{0:t}$ is estimated using the algorithm discussed in Sect. 3.1, but its uncertainty is neglected during the mapping process. Accordingly, we aim for estimating the conditional mass function $m(Y|x_{0:t},\bar{z}_{0:t})$ of the map Y given the trajectory $x_{0:t}$ and all laser scanner measurements $\bar{z}_{0:t}$. In order to make the computation feasible, we assume conditional independence among the grid cells, which allows us to factorize the joint mass function of all cells into marginal cell mass functions according to

$$m(Y|x_{0:t},z_{0:t}) = \prod_{i=1}^{M} m(Y_i|x_{0:t},\bar{z}_{0:t})\,. \tag{2}$$

Thus, the map update can be applied to single cells instead of to the entire map. In particular, the mass function of each cell can be estimated recursively over time using

$$m(Y_i|x_{0:t},\bar{z}_{0:t}) = m(Y_i|x_{0:t-1},\bar{z}_{0:t-1}) \textcircled{\cap} m(Y_i|x_t,\bar{z}_t)\,, \tag{3}$$

for $1 \leq i \leq M$. The mass function $m(Y_i|x_{0:t-1},\bar{z}_{0:t-1})$ corresponds to the map belief for $t-1$ and $m(Y_i|x_t,\bar{z}_t)$ is the map belief resulting from the current measurement $\bar{z}_t$. The latter is computed using the inverse sensor model proposed by Clemens et al. (2016). Finally, the operator $\textcircled{\cap}$ denotes the conjunctive rule of combination (Smets, 1990). In contrast to other combination rules, $\textcircled{\cap}$ yields mass on $\emptyset$ and thus enables the identification of conflicts. Because of low vehicle speeds on a parking lot, consecutive laser scans have a large overlap. Thus, the map is not updated using every measurement but only when a certain distance (e.g., 0.25 m) has been driven.

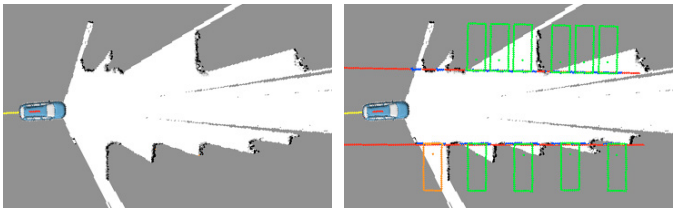
3.3 Obstacle Detection

The evidential grid map discussed above provides the basis for decisions on a longer time scale. However, since it is not updated with all measurement, it does not adapt very fast to changes in the environment and thus is only partially suited for making short-term decisions. For that reason, we compute a local *obstacle map* in order to retrieve up-to-date information regarding obstacles in the vicinity of the vehicle (see Fig. 4a). Here, a probabilistic grid map is used, since it is sufficient for that purpose. Each grid cell $\theta_i \in [0,1]$, $1 \leq i \leq N$ of this map indicates the probability of hitting an obstacle in the corresponding area. The entire map is given by $\theta \in [0,1]^N$ and N denotes the total number of cells, which is usually much smaller than the number of cells in the evidential grid map, i.e., $N \ll M$.

In contrast to the evidential grid map, the obstacle map is not computed for the entire sequence of measurements $\bar{z}_{0:t}$ but only for the latest measurements $\bar{z}_{t-d:t}$ with $d < t$. We solve the mapping problem using *forward model mapping* (Thrun et al., 2005, Sect. 9.4), which uses the forward sensor model $p(\bar{z}_t|\theta_i,x_t)$ instead of the inverse sensor model $p(\theta_i|x_t,\bar{z}_t)$. This is realized by applying maximum likelihood estimation (MLE) in order to obtain the most likely value θ_i^* for each cell, which results in

$$\theta_i^* = \underset{\theta_i \in [0,1]}{\arg\max} \prod_{j=t-d}^{t} p(\bar{z}_t|\theta_i,x_t)\,. \tag{4}$$

In general, this equation can only be applied to the entire trajectory slice $x_{t-d:t}$ and the entire measurement slice $\bar{z}_{t-d:t}$, i.e., the map cannot be estimated recursively over time. However, if high-precise sensors like laser scanners are used, (4) reduces to $\theta_i^* = k_i/(k_i+l_i)$, where k_i, l_i are counters, which indicate how often the corresponding cell was at the end-point of a laser beam and how often it was traversed by a laser beam without being hit, respectively. Accordingly, the map for $\bar{z}_{t-d:t}$ can be obtained from the map for $\bar{z}_{t-d-1:t-1}$ by decrementing and incrementing the counters according to $\bar{z}_{t-d-1}$ and $\bar{z}_t$, respectively, which makes the algorithm highly efficient.



(a) Obstacle map (b) Parking space detection

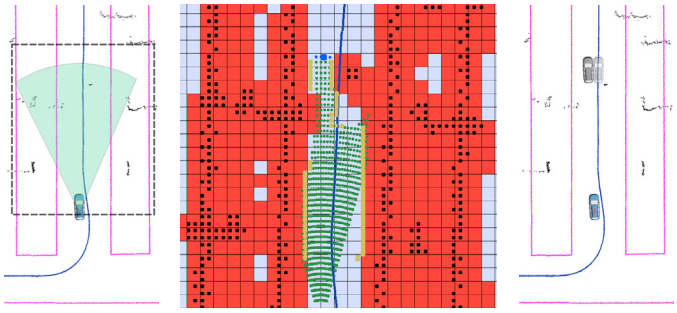
Fig. 4. An obstacle map and parking space detection based on that map. For a description of the elements see Sect. 4.1.

4. DECISION MAKING

The DM module decides on the behavior of the autonomous vehicle. It combines the current estimate of the vehicle's state and its environment with a priori knowledge and input from passengers. Within the exploration of a parking lot, the a priori knowledge relates to the geometry of the lot and the rough location of each parking space, while the input from passengers could be an area of interest or the command to park in the next possible parking space. In addition to setting the direction indicators or defining a suitable target speed depending on the current situation, DM is essentially responsible for two subtasks: (a) evaluating the fused sensor data to identify available parking spaces and (b) providing guidance that ensures continuous driving of the vehicle by defining desired target states at a high frequency (e.g., 50 Hz).

4.1 Parking Space Detection

Parking spaces are detected based on the obstacle map (see Sect. 3.3). The process is illustrated in Fig. 4b and consists of three steps. The first step aims to detect the rows of parking cars to the left and to the right of the vehicle. We assume that the other cars are aligned next to each other in approximately straight lines. Thus, for detecting the boundaries facing our vehicle, the obstacle map is transformed into the Hough space for lines (Duda and Hart, 1972) under consideration of the uncertainty associated with each grid cell. After that, one line at each side is selected based on certain criteria, e.g., distance and angle with respect to our vehicle (red lines in Fig. 4b). In the second step, we search along both lines for free spaces (blue dots in Fig. 4b). During this process, we do not only consider the grid cells located exactly on the lines but also take some grid cells next to the lines into account. The third step determines possible parking poses. For this, we fit rectangles into the free spaces obtained in the second step (green rectangles in Fig. 4b). Each rectangle has the size of our vehicle (plus a safety margin), while it is oriented orthogonally to the detected lines from the first step. Furthermore, the edge representing the rear of our vehicle is aligned with the corresponding line. If only one rectangle fits into a parking space, it is placed in the center of it (bottom row in Fig. 4b), while if multiple rectangles fit into a space, they are evenly distributed (top row in Fig. 4b). After that, one rectangle (e.g., the closest) is picked as initial parking pose (orange rectangle in Fig. 4b). Once the vehicle can observe the entire parking space, we optimize the final parking



(a) Situation (b) Evaluation (c) Result

Fig. 5. Evaluation of vehicle surrounding and reference trajectory (blue) by the guidance module. (a) Exemplary situation. Black dots: perceived obstacles. Magenta lines: a priori knowledge. Green area: evaluation in driving direction. Black dashed square: area of interest of the search algorithm. (b) Fast search algorithm. Light blue/red squares: Feasible/Infeasible positions. Blue dot: Result of forward searching. Yellow dots: Results of orthogonal searching. (c) Target computed by the search algorithm (left) and its counterpart given through the reference trajectory (right). Since the latter is too close to an obstacle the former is defined as the current target.

pose under consideration of its distance to all obstacles in the (updated) obstacle map as well as of the distance to the initial parking pose.

4.2 Guidance

The task of the guidance submodule is to continuously define new desired target states x_T at a predefined distance (e.g., 15 m). For this purpose, an optimal reference trajectory is computed once in a preliminary step. This is done according to the optimal control problem defined in Sect. 5 and solely based on a priori knowledge about the environment. Afterwards, the guidance will follow the corresponding positions and orientations if possible.

However, due to other vehicles or pedestrians, the reference trajectory might become infeasible, which is why an alternative target state is computed by an efficient fallback algorithm at every time step. Thereby, the estimate of the environment is evaluated through its conversion into a coarse obstacle grid (see Fig. 5b). Starting from the vehicles position, this is searched iteratively in the current direction of driving and orthogonally for feasible targets. If this algorithm is unable to find an unblocked lane, a feasible position is selected to let the CTRL module perform a stopping maneuver. In the case of multiple solutions, the result with the smallest deviation from the reference trajectory is selected.

The target states x_T resulting from consecutive computations are generally very close to each other. However, due to e.g. moving obstacles or when switching to a turning or parking maneuver, there might occur larger differences. In such situations, an interpolation between the previous and new solution is performed, which enables the CTRL module to use the preceding step as initial guess.

5. NONLINEAR MODEL PREDICTIVE CONTROL

In order to transfer the vehicle from an initial state to the desired target state given by DM, the CTRL module uses a nonlinear model predictive controller (NMPC). Thereby, optimal control problems are continuously defined and solved. Compared to common control approaches, NMPC is computationally expensive but offers great advantages since it takes nonlinear dynamics and path constraints into account (cf. Paden et al., 2016). In the automotive sector, there are some applications of (N)MPC. Most of them show their applicability in simulations (cf. Li et al., 2017; Weiskircher et al., 2017). Falcone et al. (2007) show the implementation of an MPC based control approach on a dSpace Autobox.

5.1 Optimal Control Problem

The planning of each maneuver can be considered as a search for an optimal trajectory and thus can be formulated as an optimal control problem given by

$$\begin{aligned} \min_{x,u,T} \quad & J(x,u,T) \\ \text{s.t. } & \dot{x}(t) = f(x(t),u(t)), \\ & x_{\min} \le x(t) \le x_{\max}, \\ & u_{\min} \le u(t) \le u_{\max}, \\ & \Psi_1(x(0),x(T)) = 0, \\ & \Psi_2(x(0),x(T)) \le 0, \\ & C(x(t),u(t),t) \le 0 \text{ for all } t \in [0,T]. \end{aligned} \tag{5}$$

The states x, controls u, as well as the process time T are determined in such a way that the objective function J is minimized. Every solution of (5) respects the dynamics given by f, the boundary conditions provided by Ψ_1 and Ψ_2, as well as further path constraints such as obstacle avoidance defined by the function C. The process time T can be either fixed or free. All states and controls are limited by physical boundaries $x_{\min}, x_{\max}$ and $u_{\min}, u_{\max}$, respectively.

In order to address optimal control problems of the form (5), we apply a *direct method* (Betts, 2010) by using the software framework TransWORHP (Knauer and Büskens, 2012). Thereby, the infinite dimensional optimal control problem is transformed into a *nonlinear optimization problem* (NLP), which is subsequently solved using the NLP solver WORHP (Büskens and Wassel, 2012).

5.2 Single Track Model

To describe the dynamic behavior of the vehicle, a kinematic single-track model is used, which provides sufficient accuracy due to low speeds on a parking lot and the correspondingly small external forces (Polack et al., 2017). Within this approach, the wheels are replaced by one virtual wheel per axle. The resulting model describes the x-y-position of the rear wheel, the vehicle's orientation ψ with respect to a reference frame, the steering angle β, the speed v and the acceleration a. To ensure the transmission of continuous and smooth controls, we add the derivatives of the acceleration and of the steering angle to the model. Thereby, the steering angle velocity ω_β is considered as an additional state variable and the controls consist of the jerk j and the steering angle acceleration a_β. Together with the wheelbase L, this leads to

$$\begin{aligned} \dot{r}_x &= v \cdot \cos(\psi), & \dot{\psi} &= v \cdot \frac{\tan(\beta)}{L}, \\ \dot{r}_y &= v \cdot \sin(\psi), & \dot{\beta} &= \omega_\beta, \\ \dot{v} &= a, & \dot{\omega}_\beta &= a_\beta, \\ \dot{a} &= j. \end{aligned} \tag{6}$$

Defining the state $x := (r_x, r_y, \psi, v, \beta, \omega_\beta, a)^\top \in \mathbb{R}^7$ and the control $u := (j, a_\beta)^\top \in \mathbb{R}^2$, the system (6) models the dynamics f in (5). States and controls are bounded by $x_{\min}, x_{\max} \in \mathbb{R}^7$ and $u_{\min}, u_{\max} \in \mathbb{R}^2$, respectively, due to physical limitations and comfort reasons in terms of smooth driving (Sommer et al., 2018).

5.3 Boundary Conditions and Obstacle Avoidance

The initial state $x(0)$ in (5) shall correspond to x_{est}, which is composed of the current state estimate x_t (see Sect. 3.1) and additional values, e.g., from the previous optimization step. Furthermore, it is required that $x(T)$ is located in the neighborhood of the target state x_T determined by DM (see Sect. 4.2). With $\varepsilon \in \mathbb{R}^7_{\geq 0}$ being small and the component-wise absolute value operator $|\cdot|$, these conditions can be expressed by Ψ_1 and Ψ_2 from (5) as

$$\begin{aligned} \Psi_1(x(0), x(T)) &:= x_{\text{est}} - x(0), \\ \Psi_2(x(0), x(T)) &:= |x_T - x(T)| - \varepsilon. \end{aligned} \tag{7}$$

In order to prevent the vehicle from colliding with obstacles, the function C forces the solution of (5) to maintain a minimum safety distance to all obstacles of the obstacle map (see Sect. 3.3). For this, an accessible area is specified by a closed polygonal chain, while the vehicle itself is approximated by overlapping circles (Sommer et al., 2018).

5.4 Objective Function

The objective function includes a weighted combination of terms consisting of the process time, controls, selected states, and penalty terms depending on the target state x_T provided by DM. Furthermore, the squared deviation from the desired speed v_{set} determined by DM is added. This yields

$$\begin{aligned} J(x,u,T) :=& w_0 T + \int_0^T w_1 {\omega_\beta}^2 + w_2 a^2 + w_3 j^2 + w_4 {a_\beta}^2 \mathrm{d}t \\ &+ \int_0^T w_5 (v - v_{\text{set}})^2 \mathrm{d}t \\ &+ w_6 (r_x(T) - r_{x,T})^2 + w_7 (r_y(T) - r_{y,T})^2 \\ &+ w_8 (\psi(T) - \psi_T)^2 + w_9 (\beta(T) - \beta_T)^2 . \end{aligned} \tag{8}$$

During exploration, $w_6, \ldots, w_9$ are set to zero and Ψ_2 in (7) shall hold. When entering a parking space, $w_6, \ldots, w_9 > 0$ are set and Ψ_2 is disregarded. The weighting can be adjusted adaptively to the current application, situation, and environment.

5.5 Implementation

An optimal control problem of the form (5) is set up with (6), (7), (8), and physical constraints (Sommer et al., 2018). As soon as (5) is solved, the first few control signals (acceleration and steering wheel angle, cf. Sect. 2.1) are transmitted to the vehicle. Therefore, the determined steering angle β is converted to the steering wheel angle. Afterwards (5) is set up with the new current estimated state and is solved again.

Even if NMPC adapts to a changing environment, there can be situations that require the vehicle to stop quickly. To ensure an immediate reaction, a high-priority *safety module* is used to initiate an emergency braking maneuver. Thereby, the minimum distance to obstacles in the direction of travel of the vehicle is monitored and evaluated with respect to the current speed and steering angle. In case of a potential collision, the module overwrites the acceleration value determined by NMPC and stops the vehicle.

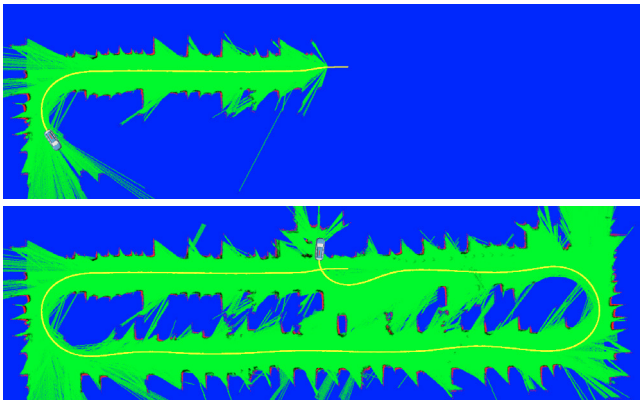
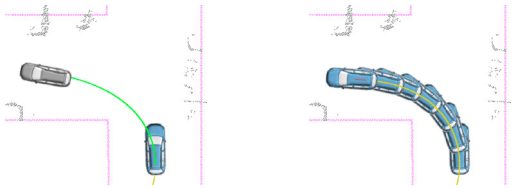


Fig. 6. Evidential grid map of the parking lot of the University of Bremen. In the map, red corresponds to mass on o, green corresponds to mass on f, blue corresponds to mass on Θ_Y, and consequently black corresponds to mass on $\emptyset$. Furthermore, the estimated driven path is shown in yellow.



(a) Optimal trajectory (b) Progress of turning

Fig. 7. Turning maneuver. Gray vehicle: target position. Blue vehicle: estimated position. Magenta lines: a priori knowledge of impassable areas. Black dots: detected obstacles. Yellow line: estimated driven path. Green line: optimal path determined by NMPC.

6. REAL WORLD DEMONSTRATION

To demonstrate the proposed system for autonomous driving, we present results for the exploration of a parking lot of the University of Bremen. We show how the self-driving vehicle explores the area performing lane keeping, dynamically taking obstacles into account, and finally executing a parking maneuver. Two corridors are selected as the area of interest and a set of parking spaces is provided from which the vehicle is supposed to find a free one.

Fig. 6 shows the evolving evidential grid map as described in Sect. 3.2. Parked vehicles and further obstacles detected during exploration are registered accordingly. Utilizing the information provided by FUS and a priori knowledge of the parking lot, DM determines a desired target state for CTRL. An exemplary situation is shown in Fig. 7, where the vehicle is guided through a left turn. It should be noticed that the posterior poses in Fig. 7b vary slightly from the path shown in Fig. 7a because DM continuously defines new target states and the used model is only an approximation of the reality.

When a free parking space is found, a maneuver to enter it is initiated by DM. Thereby, the desired target pose is moved into the parking space as described in Sect. 4.2. Exemplary parking trajectories and maneuvers have been shown by Sommer et al. (2018).

7. CONCLUSION

In this paper, a system for navigating and controlling an autonomous vehicle was presented, which was developed in the research project AO-Car. The goal of the latter was to transfer theories and methods for multi-sensor fusion and nonlinear model predictive control that were developed for deep space missions to this terrestrial application in the short time period of one year. Test drives on a real world vehicle showed that these methods yield a system that solves the challenging tasks of driving on a parking lot under consideration of unknown obstacles and of exploring it autonomously. Furthermore, free parking spaces are detected and a parking maneuver can be initiated on request. Altogether, our results show that approaches from space applications can be transferred to the terrestrial application of autonomous driving with comparably little effort and great success. One of the main advantages of the presented approach is its versatility, which allows extensions to further scenarios. In particular, we are continuing the work within the project OPA^{3}L, which focuses on urban traffic and posses the challenges of dealing with a highly-dynamic environment as well as of cooperating with other vehicles. Another topic is to perform a quantitative evaluation of the presented algorithms.

REFERENCES

Betts, J. (2010). *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Society for Industrial and Applied Mathematics, second edition.

Büskens, C. and Wassel, D. (2012). The ESA NLP Solver Worhp. In *Modeling and Optimization in Space Engineering*, 85–110. Springer.

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

Clemens, J., Meerpohl, C., Schwarting, V., Rick, M., Schill, K., and Büskens, C. (2018). Autonomous in-ice exploration of the Saturnian moon Enceladus. In *69th International Astronautical Congress (IAC)*.

Clemens, J., Reineking, T., and Kluth, T. (2016). An evidential approach to SLAM, path planning, and active exploration. *International Journal of Approximate Reasoning*, 73, 1–26.

Dickmanns, E.D. (1997). Vehicles capable of dynamic vision. In *Proceedings of the Fifteenth International Joint Conference on Artifical Intelligence*, volume 2, 1577–1592. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Duda, R.O. and Hart, P.E. (1972). Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15, 11–15.

Falcone, P., Borrelli, F., Asgari, J., Tseng, H.E., and Hrovat, D. (2007). Predictive active steering control for autonomous vehicle systems. *IEEE Transactions on Control Systems Technology*, 15(3), 566–580.

Ferguson, D., Howard, T., and Likhachev, M. (2008). Motion planning in urban environments. *Journal of Field Robotics*, 25(11–12), 939–960.

Hertzberg, C., Wagner, R., Frese, U., and Schröder, L. (2013). Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *Information Fusion*, 14(1), 57–77.

Knauer, M. and Büskens, C. (2012). From WORHP to TransWORHP. In *Proceedings of the 5th International Conference on Astrodynamics Tools and Techniques*.

Li, S.E., Guo, Q., Xu, S., Duan, J., Li, S., Li, C., and Su, K. (2017). Performance enhanced predictive control for adaptive cruise control system considering road elevation information. *IEEE Transactions on Intelligent Vehicles*, 2(3), 150–160.

Maurer, M., Gerdes, J.C., Lenz, B., and Winner, H. (eds.) (2016). *Autonomous Driving: Technical, Legal and Social Aspects*. Springer, Heidelberg.

Meerpohl, C., Flaßkamp, K., and Büskens, C. (2018). Optimization strategies for real-time control of an autonomous melting probe. In *American Control Conference (ACC)*.

Nakath, D., Clemens, J., and Schill, K. (2018). Multi-sensor fusion and active perception for autonomous deep space navigation. In *21th International Conference on Information Fusion (FUSION)*, 2596–2605. IEEE.

Paden, B., Čáp, M., Yong, S.Z., Yershov, D., and Frazzoli, E. (2016). A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1), 33–55.

Pagac, D., Nebot, E.M., and Durrant-Whyte, H. (1998). An evidential approach to map-building for autonomous vehicles. *IEEE Transactions on Robotics and Automation*, 14(4), 623–629.

Polack, P., Altché, F., d'Andréa Novel, B., and de La Fortelle, A. (2017). The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles? In *IEEE Intelligent Vehicles Symposium (IV)*, 812–818.

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

Reineking, T. and Clemens, J. (2014). Dimensions of uncertainty in evidential grid maps. In C. Freksa, B. Nebel, M. Hegarty, and T. Barkowsky (eds.), *Spatial Cognition IX*, volume 8684 of *Lecture Notes in Computer Science*, 283–298. Springer International Publishing.

Schattel, A. (2018). *Dynamic modeling and implementation of trajectory optimization, sensitivity analysis, and optimal control*. PhD thesis, University of Bremen.

Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, NJ.

Smets, P. (1990). The combination of evidence in the transferable belief model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5), 447–458.

Smets, P. and Kennes, R. (1994). The transferable belief model. *Artificial Intelligence*, 66(2), 191–234.

Sommer, L., Rick, M., Folkers, A., and Büskens, C. (2018). AO-Car: transfer of space technology to autonomous driving with the use of WORHP. In *Proceedings of the 7th International Conference on Astrodynamics Tools and Techniques*.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. MIT Press, Cambridge, MA.

Weiskircher, T., Wang, Q., and Ayalew, B. (2017). Predictive guidance and control framework for (semi-)autonomous vehicles in public traffic. *IEEE Transactions on Control Systems Technology*, 25(6), 2034–2046.

Wille, J.M. and Form, T. (2008). Realizing complex autonomous driving maneuvers. In *2008 IEEE International Conference on Vehicular Electronics and Safety*, 232–236.