Experimental Validation of Online Motion Planning for Semi-Autonomous Vehicles

Christoph Winter, Ricardo de Castro and Tilman Bünte

This work describes the optimization of paths and velocity profiles for the usage in semi-autonomous vehicles. The focus of the optimization lies on an energy efficient motion while considering the driver's longitudinal demands. The trade-off between travel time and energy consumption of the vehicle can be directly chosen by the driver. With help of a nonlinear model predictive control framework the global optimum of the velocity optimization with nonlinear constraints based on the vehicle's physical limits is found. The online path planner together with the online velocity planner provide the input to the underlying vehicle's control modules, as a basis for automated driving. The effectiveness of the proposed approach is shown via hardware-in-the-loop tests and real-world driving tests.

C. Winter, R. De Castro and T. Bünte, "Experimental Validation of Online Motion Planning for Semi-Autonomous Vehicles," 2022 IEEE Vehicle Power and Propulsion Conference (VPPC), 2022, pp. 1-7, doi: 10.1109/VPPC55846.2022.10003270.

Experimental Validation of Online Motion Planning for Semi-Autonomous Vehicles

Christoph Winter
Institute of System Dynamics and Control
German Aerospace Center (DLR)
Oberpfaffenhofen, Germany
christoph.winter@dlr.de

Ricardo de Castro
Department of Mechanical Engineering
University of California, Merced
Merced, United States of America
rpintodecastro@ucmerced.edu

Tilman Bünte
Institute of System Dynamics and Control
German Aerospace Center (DLR)
Oberpfaffenhofen, Germany
tilman.buente@dlr.de

***Abstract*— This work describes the optimization of paths and velocity profiles for the usage in semi-autonomous vehicles. The focus of the optimization lies on an energy efficient motion while considering the driver's longitudinal demands. The trade-off between travel time and energy consumption of the vehicle can be directly chosen by the driver. With help of a nonlinear model predictive control framework the global optimum of the velocity optimization with nonlinear constraints based on the vehicle's physical limits is found. The online path planner together with the online velocity planner provide the input to the underlying vehicle's control modules, as a basis for automated driving. The effectiveness of the proposed approach is shown via hardware-in-the-loop tests and real-world driving tests.**

***Keywords*—path planning, velocity planning, energy optimization, nonlinear MPC, dynamic programming**

I. Introduction

The main focus of this work consists in the optimal generation of reference paths and velocity profiles for semi-autonomous vehicles, e.g. the DLR's robotic electric vehicle research platform ROboMObil (ROMO) [1]. With increased automation of vehicles, new ways to interpret the longitudinal driving demands are emerging. Mostly visual, acoustic or haptic feedback for the driver is used to direct the driving behaviour towards a calculated optimal velocity setpoint [2]. We are particularly interested in an energy efficient motion while considering the driver's longitudinal demands, i.e. his preference on how fast the car should be driven along the reference path. A similar approach to let the driver choose between fast and slow trajectories was presented in [3]. There the driver influences the overall travel time which is incorporated as an additional constraint in the optimization problem. The travel time chosen by the driver thus indirectly influences the energy consumption. The resulting optimization problem from [3] is nonlinear and globally optimal solutions cannot be guaranteed. In [4] a complete convex-optimization-based method to solve speed planning problems for autonomous driving in both static and dynamic environments is presented, overcoming the problem of possibly suboptimal, i.e. only locally optimal solutions. However, the driver's demand is not incorporated into the velocity planning as proposed in our previous work [5], where the trade-off between time and energy optimality is explored for offline planned velocity profiles.

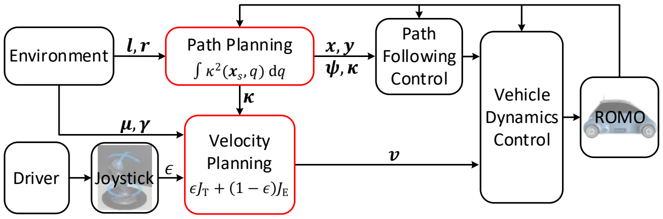
Environment
$\boldsymbol{l}, \boldsymbol{r}$
Path Planning
$\int \kappa^2(\boldsymbol{x}_s, q)\,\mathrm{d}q$
$\boldsymbol{x}, \boldsymbol{y}$
$\boldsymbol{\psi}, \boldsymbol{\kappa}$
Path Following Control
Vehicle Dynamics Control
ROMO
$\boldsymbol{\kappa}$
$\boldsymbol{\mu}, \boldsymbol{\gamma}$
Driver
Joystick
ϵ
Velocity Planning
$\epsilon J_{\mathrm{T}} + (1-\epsilon) J_{\mathrm{E}}$
$\boldsymbol{v}$

Fig. 1: Motion planning framework of the ROboMObil.

Fig. 1 shows a schematic of the motion planning framework of the ROboMObil, the prototype vehicle employed in this work. The planning framework is composed of two main modules: path planner [6] and velocity planner. The path planner receives environmental information in form of the left and right lane boundaries $\boldsymbol{l}$ and $\boldsymbol{r}$ and determines an energy efficient and comfortable path with coordinates $\boldsymbol{x}$ and $\boldsymbol{y}$ based on curvature minimization. The velocity planning determines the reference vehicle velocity $\boldsymbol{v}$ along this path, taking into account i) the driver's preference on the trade-off between travel time and energy consumption, captured via a normalized scalar $\epsilon \in [0,1]$ generated by a joystick input, ii) the sequence of path curvature $\boldsymbol{\kappa}$ from the path planner, and iii) environmental information, such as the tyre-road friction coefficient $\boldsymbol{\mu}$ and the slope angle of the path $\boldsymbol{\gamma}$. The reference path $(\boldsymbol{x}, \boldsymbol{y})$ and the corresponding orientation $\boldsymbol{\psi}$ and curvature $\boldsymbol{\kappa}$ is transmitted to the vehicle's path following control (PFC) [7], which tracks the reference signals through manipulation of low-level actuation signals in the vehicle dynamics control (VDC) [8]. The reference velocity $\boldsymbol{v}$ is directly provided to the VDC. The motion planning framework can also be seen as an advanced driver assistance system, which would prevent the driver from requesting unfeasible or potentially dangerous motion demands. It can also be used in fully autonomous driving as an interface to query the driver's velocity preferences which might vary over the journey.

The main contribution of this work consists in the development of a velocity planner strategy using nonlinear model predictive control (NMPC) techniques, which are widely used e.g. in [9], that is combined with the online path planner. This control framework allows us to optimize multiple performance metrics (energy and travel time), while taking into account actuation and safety constraints due to limited grip levels between the vehicle's tires and the road. To obtain global optimal solutions for the NMPC problem we investigate the deployment of dynamic programming solvers, which allows us to obtain global optimal solutions despite non-convex prediction models. Experimental validation demonstrates the ability of our approach to generate safe velocity profiles and fulfil real-time computational constraints.

II. Online Path Planning

This section provides an overview of an optimization-based planning algorithm [6] that was used to generated the reference path for the vehicle. The reference path is represented through two-dimensional cubic splines consisting of $n-1$ polynomials $\boldsymbol{o}_i(q)=\left(o_{i,x},o_{i,y}\right)\in\mathbb{R}^2$ describing positions x and y between n interpolation points as path representation as in [10] with spline parameter $q\in\mathbb{R}$ and interpolation points $q_i\in\mathbb{N}$ as given in:

$$\boldsymbol{o}_i(q)=\boldsymbol{a}_i(q-q_i)^3+\boldsymbol{b}_i(q-q_i)^2+\boldsymbol{c}_i(q-q_i)+\boldsymbol{d}_i\,. \tag{1}$$

The interpolation points $\boldsymbol{o}_i(q_i)$ can be moved on a line between the boundary nodes $\boldsymbol{l}(q_i)\in\mathbb{R}^2$ and $\boldsymbol{r}(q_i)\in\mathbb{R}^2$ depending on the scalar scaling factor $\alpha_i\in[0,1]$ as proposed in [11] and depicted in Fig. 2. This linear relationship is given by

$$\boldsymbol{o}_i(q_i)=\boldsymbol{l}(q_i)+\alpha_i\big(\boldsymbol{r}(q_i)-\boldsymbol{l}(q_i)\big)\,. \tag{2}$$

Thus, the number of optimization variables per spline segment can be reduced from eight polynomial coefficients (refer to (1)) to only three. The reduced set of variables for one spline segment $\boldsymbol{o}_i(q)$ are the two second degree polynomial coefficients $\boldsymbol{b}_i=\left(b_{i,x}\;b_{i,y}\right)$ and α_i. All other spline parameters $\boldsymbol{a}_i,\boldsymbol{c}_i,\boldsymbol{d}_i$ can be expressed depending on those three scalar variables [12].

The reference path is determined using the optimization problem (3). The cost function is chosen based on the quadratic path curvature κ^2, similarly to other path planning works [10], [11], [13]. The potentially energy saving capabilities of the curvature minimization result from this cost function, due to reduced deceleration and acceleration before and after sharp turns with a given maximum lateral acceleration. The equality constraints of the optimization problem represent equality of position, orientation, and curvature at the junction points and thus ensure a smooth transition between the spline segments. The inequality constraints keep the path within the road boundaries, described by splines $\boldsymbol{l}(q)$ and $\boldsymbol{r}(q)$. The optimization problem with the reduced set of spline coefficients $\boldsymbol{x}_s=\left(b_{1,x},b_{1,y},\alpha_1,\dots,b_{n,x},b_{n,y},\alpha_n\right)^T$ is given as

$$\begin{aligned}\boldsymbol{x}_s^*=\operatorname*{argmin}_{\boldsymbol{x}_s}\quad&\int_0^{q_n}\kappa^2(\boldsymbol{x}_s,q)\,\mathrm{d}q\\ \text{s.t.}\quad&\boldsymbol{A}\boldsymbol{x}_s=\boldsymbol{b}\\ &\boldsymbol{C}\boldsymbol{x}_s\le\boldsymbol{d}\,,\end{aligned} \tag{3}$$

with $\boldsymbol{A}\boldsymbol{x}_s=\boldsymbol{b}$ describing the equality constraints for a smooth transition between spline segments $\boldsymbol{o}_i$ and $\boldsymbol{C}\boldsymbol{x}_s\le\boldsymbol{d}$ representing the inequality constraints given by the road boundaries $\boldsymbol{l}(q)$ and $\boldsymbol{r}(q)$, for details refer to [12]. The artificial spline coefficients $\boldsymbol{b}_n$ are required to give the spline a defined boundary condition at the end. A real-time capable nonlinear gradient method similar to [14] is chosen as optimization algorithm to solve the minimization problem.

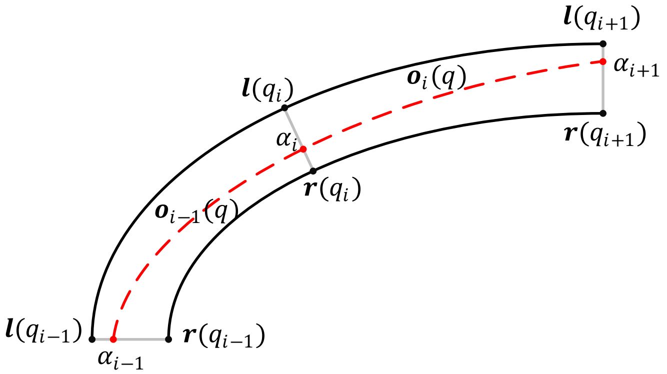
$\boldsymbol{l}(q_{i+1})$
α_{i+1}
$\boldsymbol{o}_i(q)$
$\boldsymbol{l}(q_i)$
α_i
$\boldsymbol{r}(q_{i+1})$
$\boldsymbol{r}(q_i)$
$\boldsymbol{o}_{i-1}(q)$
$\boldsymbol{l}(q_{i-1})$
$\boldsymbol{r}(q_{i-1})$
α_{i-1}

Fig. 2: Path representation with the sampling points of the spline $\mathbf{o}(q)$ placed with the scalar parameter α_i between the left and right path boundaries $\boldsymbol{l}(q_i)$ and $\boldsymbol{r}(q_i)$.

The path planning approach described here and developed in [6] is especially suitable for online usage, i.e. in real-time while the vehicle is driving. The boundary conditions of the spline representing the planned path enable a smooth transition, e.g. the same position, orientation, and curvature, between consecutive path segments. With a given bound on the maximum iterations of the nonlinear gradient method also an upper bound on the computation time is given, enabling a real-time implementation.

III. Online Velocity Planning using NMPC

This section describes the mathematical formulation of the NMPC-based velocity planning as further development of our work in [5]. We investigate an optimization problem based on two concurrent, yet rival performance metrics in the cost function: on the one hand the journey time J_T and on the other hand the energy consumption J_E.

A. *Mathematical Problem Formulation*

Regarding only minimum journey time, a concept for generating reference speed profiles based on ordinary physical considerations was proposed in [15]. While, for the sake of clarity, neglecting here some of its aspects like lateral road inclination, we adopt the foundations of this concept as a basis and extend it by energy saving considerations and optimization for finding a suitable trade-off. The vehicle's longitudinal dynamics in time-domain with the corresponding acceleration a_{long} is considered as

$$m\dot{v}=mu-F_{\mathrm{roll}}-F_{\mathrm{down}}-F_{\mathrm{air}}=ma_{\mathrm{long}}\,, \tag{4}$$

where m is the vehicle mass, v the longitudinal velocity, and u the acceleration part provided by the powertrain. Other contributing terms are the rolling resistance F_{roll}, the downhill force F_{down}, and the air drag F_{air}. Since the path description from section II as well as environmental information are described in path-domain also the velocity planning is described with respect to the arc length s hereafter.

The longitudinal motion of the vehicle is constrained by upper and lower limits of the achievable acceleration share by the drivetrain and the acting driving resistance, described as $u_{\min}(s)\le u(s)\le u_{\max}(s)$. Those limits also can be set lower than the physical limitation to satisfy comfort requirements. Combining the well-known friction circle and its relationship between lateral and longitudinal acceleration as well as the road friction coefficient μ and the slope angle γ with the lateral acceleration given as $a_{\mathrm{lat}}(s) = \kappa(s) \cdot v(s)^2$ results in the following velocity-dependent constraint:

$$(\kappa(s) \cdot v(s)^2)^2 \leq \Big(\mu(s) g \cos\big(\gamma(s)\big)\Big)^2 - u(s)^2 \,. \tag{5}$$

Additionally, further restrictions on the velocity due to traffic rules and comfort related criteria are given as $v_{\min}(s) \leq v(s) \leq v_{\max}(s)$ with the minimum velocity $v_{\min}$ and the maximum velocity $v_{\max}$. Especially important for the online usage of the velocity planning approach is the possibility to define suitable boundary conditions for the start and end values of the velocity profile. The initial speed is set to $v(0) = v_{\mathrm{start}}$ and the desired final speed $v(s_{\mathrm{f}})$ at arc length s_{f} is defined to be in the interval $v(s_{\mathrm{f}}) \in \left[v_{\mathrm{f,min}}, v_{\mathrm{f,max}}\right]$. For a smooth transition between consecutive planned velocity profiles the initial speed v_{start} of the current planning is set to the interpolated velocity at the corresponding location (arc length) of the actual reference path (i.e. the predecessor). Refer to [6] for further details.

The overall travel time will be a first summand of the total cost function (see (9) below) and is given as

$$J_{\mathrm{T}} = \int_0^{t_{\mathrm{f}}} 1 \,\mathrm{d}t = \int_0^{s_{\mathrm{f}}} \frac{1}{v} \mathrm{d}s \tag{6}$$

with the final time of the velocity profile t_{f} and the overall arc length $s(t_{\mathrm{f}}) = s_{\mathrm{f}}$. The energy consumption as the second part of the total cost function is calculated from

$$J_{\mathrm{E}} = \int_0^{s_{\mathrm{f}}} \frac{mu}{\tilde{\eta}(u,v)} \mathrm{d}s \,, \tag{7}$$

with

$$\tilde{\eta}(u,v) = \begin{cases} \eta(u,v) & \forall\, u \geq 0 \\ \dfrac{1}{\eta(u,v)} & \forall\, u < 0 \end{cases} \tag{8}$$

where $\eta \in [0,1]$ is the combined motor and battery efficiency depending on the desired acceleration u and the vehicle speed v. The case distinction in $\tilde{\eta}$ for $u < 0$ considers the reversed energy flow from the motors to the battery while braking. Note that the former quadratic acceleration term used in [5] to prevent chattering problems caused by the dynamic programming solver is abandoned. Instead the true physical energy balance in terms of the battery state of charge is considered in (7). Here, it is assumed that solely the electric motor is utilized for braking, no dissipative (friction) brake. In addition, for the sake of simplicity we assume a static control allocation (i.e. equal motor torques) between ROboMObil's four individual in-wheel motors. If a single electric motor is used in the drive train the energy cost function (7) can be applied directly without further assumptions.

The cost function portions (6) and (7) as well as the dynamic model equation (4) are discretized using a forward Euler approximation and a constant acceleration between arc length nodes is assumed. The resulting cost function can be formulated as follows:

$$J = \epsilon \frac{1}{\bar{J}_{\mathrm{T}}} J_{\mathrm{T}} + (1-\epsilon) \frac{1}{\bar{J}_{\mathrm{E}}} J_{\mathrm{E}} \tag{9}$$

and the discrete cost function results as

$$J \approx \sum_{l=0}^{N} \underbrace{\epsilon \frac{1}{\bar{J}_{\mathrm{T}}} \frac{1}{v_l} \Delta \mathrm{s} + (1-\epsilon) \frac{1}{\bar{J}_{\mathrm{E}}} \frac{m u_l}{\tilde{\eta}_l(u_l, v_l)} \Delta s}_{g_l(u_l, v_l)} \tag{10}$$

with the nominalization factors $\bar{J}_{\mathrm{T}}$ and $\bar{J}_{\mathrm{E}}$ and the cost function term $g_l(u_l, v_l)$ representing the driver defined trade-off:

$$g_l(u_l, v_l) = \epsilon \frac{1}{\bar{J}_{\mathrm{T}}} \frac{1}{v_l} \Delta \mathrm{s} + (1-\epsilon) \frac{1}{\bar{J}_{\mathrm{E}}} \frac{m u_l}{\tilde{\eta}_l(u_l, v_l)} \Delta s \,. \tag{11}$$

The discretized longitudinal vehicle dynamics model is given as

$$\begin{aligned} v_{l+1} &= v_l + \frac{\Delta s}{m v_l} \big(m u_l - F_{\mathrm{roll},l} - F_{\mathrm{down},l} - F_{\mathrm{air},l}\big) \\ &= f(u_l, v_l) \end{aligned} \tag{12}$$

where $v_l = v(s_l)$ represents the discretised variable at node l of the overall $N+1$ discrete arc length nodes with a spacing of Δs. Thereby, $f(u_l, v_l)$ describes the state space difference equation. The problem formulation is based on a single state variable v and only one input variable u. This reduced number of states and inputs is beneficial for real-time implementation.

B. NMPC Formulation

Minimizing the discrete cost function (10) with a given discrete dynamic model described by (12) over a moving horizon is well suited to be formulated as NMPC. The optimization problem considered in the NMPC is

$$\min_{u_{l|k}} \left(\sum_{l=0}^{N-1} g_{l|k}\big(u_{l|k}, v_{l|k}\big) \right) + g_{N|k}(v_{N|k}) \tag{13}$$

with $g_{N|k}$ later defined in (16) and the prediction model

$$v_{l+1|k} = f\big(u_{l|k}, v_{l|k}\big) \tag{14}$$

and the nonlinear constraints

$$\begin{gathered} u_{\mathrm{min},l|k} \leq u_{l|k} \leq u_{\mathrm{max},l|k} \,, \\ v_{\mathrm{min},l|k} \leq v_{l|k} \leq v_{\mathrm{max},l|k} \,, \\ \big(|\kappa_{l|k}| v_{l|k}^2\big)^2 \leq \Big(\mu_{l|k} g \cos\big(\gamma_{l|k}\big)\Big)^2 - u_{l|k}^2 \,, \\ v_{0|k} = v_{\mathrm{start}|k} \,, \end{gathered} \tag{15}$$

where the $l|k$ subindex indicates the predicted value of a given variable at the arc length node l in time instant $k \in \mathbb{N}^+$. Note that index l refers to a short and fixed arc length step size according to the discretization in section III.A and index k refers to a coarser grid with variable arc length step size but a fixed time step resulting from the motion planning being updated and replanned periodically. Hereby, we assume in accordance to [5] that

- future information about the curvature $\kappa_{l|k}$, friction coefficient $\mu_{l|k}$, and terrain slope angle $\gamma_{l|k}$ are available to the velocity planner, e.g. provided by the path planner, friction estimator [16], and digital map,
- the local minimum and maximum velocities along the path $v_{\mathrm{min},l|k}, v_{\mathrm{max},l|k}$ are available, e.g. based on digital maps, legal velocity limits or driver preferences,
- the initial speed $v_{\mathrm{start},k}$ of the current planning is set to the interpolated velocity at that point of the path of the previous planning result and

- the terminal cost $g_{N|k}(v_{N|k})$ encodes a preference for the desired velocity range at the end of the prediction horizon. It is defined as:

$$g_N(v) = \begin{cases} 0 & \text{if } v \in [v_{\text{f,min}}, v_{\text{f,max}}] \\ \infty & \text{otherwise} \end{cases}. \tag{16}$$

The implementation of the velocity planner follows a receding horizon strategy, i.e., at each time step k of the motion planning framework the optimal sequence $v_{l|k}$ minimizing J in (10) is computed and then the complete resulting velocity profile is provided as a demand input to the VDC, see Fig. 1.

C. Dynamic Programming

The NMPC formulation of the velocity planning problem, makes it well suitable to be solved by dynamic programming (DP) [17]. An important advantage of DP lies in its capability to handle nonlinear nonconvex cost-functions and constraints, while guaranteeing global optimal solutions of the individual optimization. On the other hand, high computational costs are a disadvantage of this approach. The dynamic optimization problem is solved backwards with the help of the following recursive equation starting from the end state $v_{N|k}$ and the index $l = N - 1, \dots, 0$

$$\begin{aligned} J_N(v_{N|k}) &= g_N(v_{N|k}) \\ J_l(v_{l|k}) &= \min_{\text{s.t. } (15)} \Big\{ g_l(u_{l|k}, v_{l|k}) \\ &\qquad + J_{l+1}\Big(f(u_{l|k}, v_{l|k})\Big)\Big\} \end{aligned} \tag{17}$$

for all $v_{l|k} \in \mathcal{V}$ and $u_{l|k} \in \mathcal{U}$, where $J_l(v_{l|k})$ is the cost-to-go function [17] and $\mathcal{V}$ is the space of discretized states and $\mathcal{U}$ is the space of discretized inputs defined in (15). Computing the cost-to-go function J_l yields the optimal acceleration $u_k^*(v_k)$ as the argument minimizing (17). The optimal velocity v_k^* is obtained by the longitudinal vehicle dynamics model (12).

The computational effort for solving the problem with dynamic programming strongly depends on the number of states and the number of nodes along the considered path that is optimized. For an efficient solving of the optimization problem a suitable size reduction is possibly required. The state space as well as the input space are innately constrained to their feasible values according to (15). Apart from that, computation time roughly depends linearly on the density of the state and input grid. Therefore, the grids can be chosen as dense as possible while keeping the computation time below the threshold discussed later in section IV.B.

IV. Experimental Results

This section describes the experimental validation by means of Hardware-in-the-Loop (HIL) simulations and real-world test drives with the ROboMObil for the combined online path planning and NMPC velocity planning approach. The focus of the experiments is on the tradeoff between energy savings and driving time, as well as on the online motion planning.

A. Description of the HIL Test Framework

To evaluate the real-time capabilities of the total online motion planning in a first step the HIL test bench of the ROboMObil [18] is used. For this purpose, the virtual design and test environment contains detailed multi-physics dynamics models based on the object-oriented modeling language Modelica. These realistic models perform real-time vehicle simulations that cover not only the multibody vehicle dynamics, but also the electro-mechanical actuators, the tires and all of ROboMObil's peripheral devices and sensors. An overview of the HIL setup is given in Fig. 3. The motion planning described in this paper is executed on the dSPACE AutoBox completed by PFC [7] and VDC modules. The abovementioned total model of the ROboMObil is simulated on a dSPACE SCALEXIO system and is shaded in blue in Fig. 3. The human machine interface to provide the driver's input is directly used in the ROboMObil, given that the ROboMObil hardware is part of the HIL test bench.

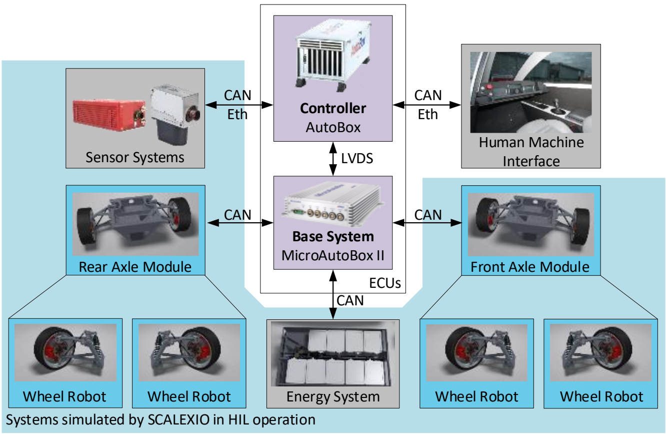

Fig. 3: Overview of ROboMObil's HIL setup with the motion planning running on the AutoBox. The components represented virtually by the real-time vehicle simulation on the SCALEXIO are shaded in blue.

The total HIL setup provides a realistic testing environment which reduces the development effort by facilitating the preparation of our algorithms already on the real-time hardware later used in the test drives.

B. Analysis of Online and Real-Time Capability

The aim is to have the path and velocity planning both running on the vehicle ("online") while it is following the currently planned motion, as introduced in section I. Therefore, the implementation of DP as a Matlab function [19] is adapted to run in real-time on the ROboMObil's rapid control prototyping hardware (AutoBox, cf. Fig. 3). That means the Matlab code was thinned out to fit the specific problem formulation of section III.C and rewritten to be suitable for code generation using the Matlab Coder and the `%#codegen` directive.

The assessment of a meaningful computation time threshold of the DP algorithm in [5] depending on the problem size and discretization resolution of the state space still holds true. This threshold is also constrained by the vision-based environment perception of the ROboMObil [20], providing information for potential path updates, e.g. due to obstacles. A faster update rate than the approximate 0.1 s or 10 Hz sample rate of the environmental perception would not bring any advantage, since no new information would be available during faster intermediate motion planning updates. Table 1 shows the turnaround (execution) times T for the motion planning framework obtained with the HIL setup, while varying the arc length step sizes Δs and resulting from a fixed horizon of 100 m also varying the number of arc length steps N accordingly. Thereby, $N_u = 25$ steps in the input grid and $N_x = 35$ steps in the state grid are used.

Table 1: Turnaround times of the real-time hardware AutoBox for different arc length discretization Δs with a horizon of 100 m and $N_u = 25$ steps in the input grid and $N_x = 35$ steps in the state grid of the DP formulation.

Arc Length Step Size Δs	**Number of Arc Length Steps N**	**Turnaround Time T**	**Valid**
0.1 m	1000	359 ms	Yes
0.2 m	500	225 ms	Yes
0.5 m	200	148 ms	Yes
1 m	100	122 ms	Yes
2 m	50	109 ms	No

On the one hand, a large arc length step size Δs is beneficial for a short computation time. On the other hand, a too coarse grid can violate the assumptions made in the model discretization of section III.A, such as a constant acceleration within one step. Since a short update time of the motion planning is preferred, an arc length step size of 1 m is chosen, which corresponds to a turnaround time of approximate 0.12 s still in a reasonable range of the above-mentioned threshold.

C. Path and Velocity Planning Profiles

In this section we show simulation results of the online motion planning using the previously described methods. A test path defined by a center line marked in blue and the dashed black boundaries, is shown on an aerial image in Fig. 4. The online path planning module from section II optimizes the path, resulting in the green line. The optimized path has an approximately 30 % lower curvature cost function (cf. (3)) compared to the original blue center line. This leads to a higher driving comfort due to reduced lateral accelerations and enables an energy reduction for fast velocity profiles as shown in [6].

The optimized path with its curvature profile as a function of the arc length serves as input for the subsequent velocity planning module described in section III. First, the stepwise results of the NMPC based velocity planning yielding the overall velocity $v_{l|k}$ is presented in Fig. 5. The individual optimal solutions $v_{l|1}, v_{l|2}$ etc. determined by the DP algorithm sectionally compose the resulting velocity profile $v_{l|k}$, acting as the input to the VDC, shown with a dashed black line. The parts of the velocity profile contributing to the overall velocity in each time step k are marked bold. In each replanning step only the first part of the NMPC solution is applied. As expected, the prediction in each step (cf. (14)) varies from the overall velocity for arc lengths further ahead.

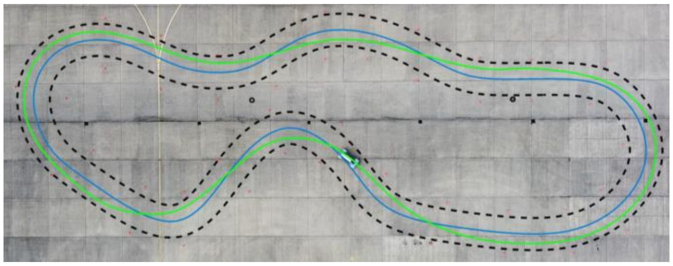

Fig. 4: Aerial image of the motion planning test drive with the ROboMObil where the overlayed blue line marks the center line of the path with its boundaries depicted as dashed black lines and the optimized path as green line.

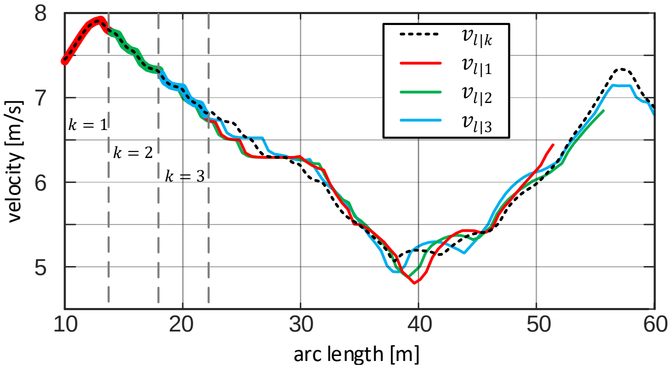

Fig. 5: Stepwise velocity profiles generated by the NMPC based velocity planning. The effective overall velocity profile $v_{l|k}$, is sectionally composed of the successive optimal solutions $v_{l|1}$, $v_{l|2}$ etc. at each time step k. In each replanning step only the first part of the NMPC solution is applied. Display in this figure is limited to the first three solutions $k = 1 \dots 3$.

Second, to show the impact of the trade-off parameter ϵ, various resulting velocity profiles for the same test path are compared while the trade-off parameter is varied in the interval $\epsilon \in [0,1]$. Fig. 6 shows the resulting family of velocity profiles. The lower and upper bounds of the velocity are marked in gray and the same start velocity of $v_{\text{start}} = 5$ m/s for all profiles is chosen. As it can be seen, all velocity constraints of the problem are satisfied. Fig. 7 depicts the corresponding accelerations and their upper and lower limits for the trade-off parameter $\epsilon = 0.1$ in the upper plot and $\epsilon = 1$ in the lower plot, showing that the acceleration constraints of the problem, marked as black lines, are satisfied. For energy saving motion ($\epsilon = 0.1$) long phases with zero acceleration from the traction motors are preferred and only the driving resistances are acting, whereas for short travel time ($\epsilon = 1$) the acceleration limits determine the resulting motion.

The combined path and velocity information along the test path in each replanning step is passed to the PFC and VDC as demand.

D. Energy Consumption

The energy consumption as the main evaluation criterion is considered during real-world tests. The test drives of the ROboMObil in this section compare offline planned velocity profiles with profiles planned using the NMPC approach and

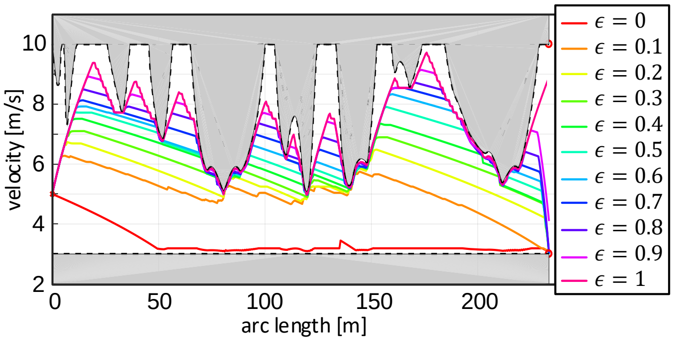

Fig. 6: Velocity profiles for a varying trade-off parameter ϵ within the constrained state-space generated by the velocity planning framework for the test path shown in Fig. 4.

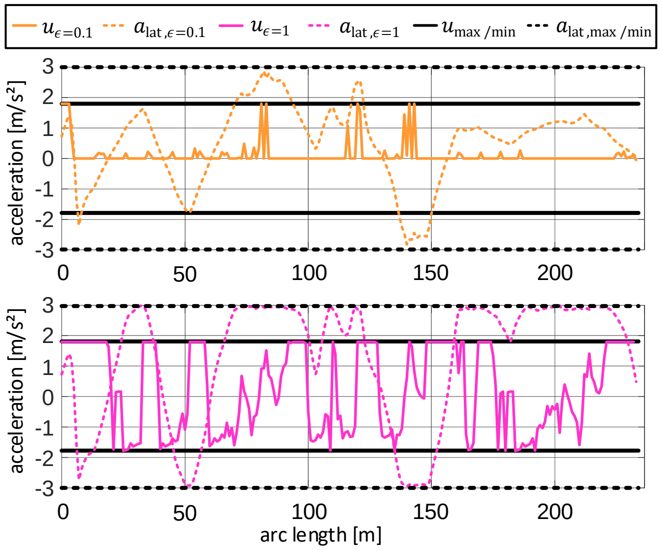

Fig. 7: Longitudinal and lateral acceleration profiles for a trade-off parameter $\epsilon = 0.1$ in the upper plot and $\epsilon = 1$ in the lower plot within the acceleration constrains generated by the velocity planning framework for the test path shown in Fig. 4.

investigate the benefit of the path optimization. An impression of the test drive can be seen in the video [21]. Note that the implementation of the NMPC based DP function fulfills the requirements for code generation as described in section IV.B, but shows discrepancies after compilation on the target system. Therefore, and to reflect here precisely the conditions of the experiments: The method for sequential online velocity planning described in section III was used to determine a velocity profile prior the test drive rather than online. The energy consumption and drive time for multiple planned motions are compared hereafter, refer to Table 2 for an overview. The middle of the track, i.e. no path optimization, acts as reference path. The optimized paths are approximately 3 m longer than the reference path, since less curvature results in higher curve radii.

Table 2: Overview of the energy consumption and driving times of real-world test drives on the test track depicted in Fig. 4 for different trade-off factors and different setups of path and velocity optimization.

Setup		**Trade-Off ϵ**	**Energy**	**Drive Time**
Path Opt.	***Velocity Opt.***			
-	offline	1	149.5 kJ	37.1 s
-	offline	0.1	74.6 kJ	47.1 s
-	offline	0	67.1 kJ	76.5 s
offline	offline	1	141.6 kJ	33.6 s
offline	offline	0.1	72.9 kJ	41.3 s
offline	offline	0	68.7 kJ	77.4 s
online	NMPC	1	129.2 kJ	33.0 s
online	NMPC	0.1	69.1 kJ	42.5 s
online	NMPC	0	66.0 kJ	77.5 s

First, it can be seen that the velocity planning and the trade-off factor ϵ chosen by the driver have the most impact on the resulting energy consumption and drive times, as expected. The energy saving effect of the path optimization shows when comparing the reference path and the optimized paths for $\epsilon = 1$, where the assumption that a reduced curvature leads to reduced deceleration and acceleration before and after sharp turns holds true. This results in up to 13 % in energy savings while also reducing the drive time up to 11 % between the reference and the online optimized path. Due to reduced lateral acceleration also a higher driving comfort can be achieved. The shorter drive time is the outcome of higher velocities due to minimized curvature. In contrast to the time-optimal profile, the path optimization has no benefit for the velocity profile with $\epsilon = 0$ where a constant slow velocity is used.

Second, the DP-based velocity planning results in significant energy savings while maintaining reasonable driving times for all motion planning setups investigated in the experiments by choosing a low trade-off factor of $\epsilon = 0.1$. The effect is similar for all setups and shows a 46 % energy reduction in exchange for a 29 % longer driving time for the online planning setup. Focusing the optimization only on the energy cost function with $\epsilon = 0$ yields a further decreased energy consumption with a 49 % reduction compared to the reference but an impractical increase of the drive time of 135 %.

Last, the results of the online and offline planning are investigated. Comparing the time and energy optimal test drives of the offline and online planned motions, it seems counter intuitive that the online planning achieves a slightly shorter drive time for $\epsilon = 1$ and a lower energy consumption for $\epsilon = 0$. Since the curvature of the path is the basis for the velocity planning, in Fig. 8 we compare the offline (blue line) and online (orange line) planned curvature and corresponding velocity profiles for $\epsilon = 1$ to understand the above-mentioned effect. The most noticeable difference between the two motion profiles can be seen in the beginning of the path where the online planning yields a lower curvature compared to the offline planning. Other than that, the curvature and velocity profiles are nearly identical. The reason for the higher curvature peak from the offline planning could be tensions in the spline describing the path from not ideal boundary conditions at the beginning of the path. The online planning on the other hand benefits from its continues planning where the boundary conditions are given by the previous planned path segment, resulting in a smoother motion. Furthermore, the in (3) described optimization problem in the path planning framework is solved with a nonlinear gradient decent, not leading to global optimal solutions. The lower curvature of the online planning allows for a continuous high velocity at the beginning of the path and therefore the shorter

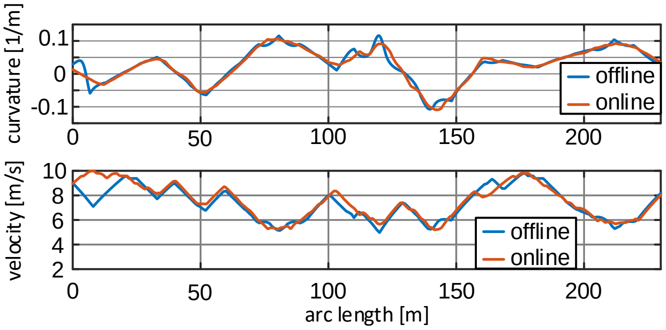

Fig. 8: Comparison of the curvature and the corresponding velocity profile for the offline and online planned motion for $\epsilon = 1$.

overall drive time. The planned path itself in the x and y plane is nearly identical for the offline and online path optimization.

In summary, the online motion planning achieves similar results to the offline planning on the investigated test track, whereby a relatively long horizon of the online planning of 100 m in comparison to the overall track length of 230 m is beneficial.

V. Conclusion and Future Work

We proposed a nonlinear model predictive control framework for online velocity planning of semi-autonomous vehicles. This framework allows trade-offs between time optimal and energy optimal velocity profiles. This trade-off can be chosen by the driver by means of a corresponding user-defined parameter. The online velocity planning is integrated in the online motion planning framework of the ROboMObil along with the online path planning. The motion planning is able to take environmental conditions (possibly changing along the way) into account by incorporating them into the optimization constraints. Furthermore, the implementation of the motion planning algorithms on the rapid control prototyping system of the ROboMObil and their experimental validation was shown.

Future work will include the combination with environmental perception information for a reactive path planning and the coupling with reinforcement learning based path following control [22]. Moreover, solvers for the velocity planning with less computational effort compared to dynamic programming will be investigated.

References

[1] Jonathan Brembeck et al., "ROMO – The Robotic Electric Vehicle," in *International Symposium on Dynamics of Vehicle on Roads and Tracks*, 2011.

[2] Carina Fors, Katja Kircher, and Christer Ahlström, "Interface design of eco-driving support systems – Truck drivers' preferences and behavioural compliance," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 706 - 720, 2015.

[3] Thijs van Keulen, Bram de Jager, Darren Foster, and Maarten Steinbuch, "Velocity trajectory optimization in hybrid electric trucks," in *American Control Conference (ACC), 2010*, 2010, pp. 5074-5079.

[4] Yu and Chen, Huiyan and Waslander, Steven L. Zhang, Tian Yang, Sheng Zhang, Guangming Xiong, and Kai Liu, "Toward a More Complete, Flexible, and Safer Speed Planning for Autonomous Driving via Convex Optimization," *Sensors*, vol. 18, no. 7, 2018.

[5] Christoph Winter and Ricardo de Castro, "Optimal Velocity Profile Generation for Semi-Autonomous Vehicles," *AUTOREG 2017*, vol. 2292, pp. 129-141, 2017.

[6] Christoph Winter, Peter Ritzer, and Jonathan Brembeck, "Experimental Investigation of Online Path Planning for Electric Vehicles," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, Brazil, 2016, pp. 1403-1409.

[7] P. Ritzer, C. Winter, and J. Brembeck, "Advanced path following control of an overactuated robotic vehicle," in *IEEE Intelligent Vehicles Symposium*, 2015, pp. 1120-1125.

[8] Ricardo de Castro, Tilman Bünte, and Jonathan Brembeck, "Design and Validation of the Second Generation of the Robomobil's Vehicle Dynamics Controller," in *Proceedings of the 24th Symposium of the International Association for Vehicle System Dynamics (IAVSD 2015)*, Graz, 2016.

[9] Robin and De Bruyne, Stijn and Zanon, Mario and Frasch, Janick V. and Diehl, Moritz Verschueren, "Towards time-optimal race car driving using nonlinear MPC in real-time," in *53rd IEEE Conference on Decision and Control*, Los Angeles, CA, USA, 2014, pp. 2505-2510.

[10] J. Daniel, A. Birouche, J.-P. Lauffenburger, and M. Basset, "Navigation-based constrained trajectory generation for advanced driver assistance systems," *International Journal of Vehicle Autonomous Systems*, vol. 9, no. 3-4, pp. 269-296, 2011.

[11] F Braghin, F Cheli, S Melzi, and E Sabbioni, "Race Driver Model," *Computers and Structures*, vol. 86, no. 13-14, pp. 1503-1516, 2008.

[12] Jonathan Brembeck and Christoph Winter, "Real-time capable path planning for energy management systems in future vehicle architectures," in *IEEE Intelligent Vehicles Symposium*, 2014, pp. 599-604.

[13] Liang Ma, Jing Yang, and Meng Zhang, "A Two-level Path Planning Method for On-road Autonomous Driving," in *International Conference on Intelligent System Design and Engineering Application*, 2012.

[14] J. Rosen, "The Gradient Projection Method for Nonlinear Programming. Part I. Linear Constraints," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 1, pp. 181-217, 1960.

[15] Tilman Bünte and Emmanuel Chrisofakis, "A Driver Model for Virtual Drivetrain Endurance Testing," in *8th International Modelica Conference*, Dresden, Germany, 2011.

[16] R. de Castro, R.E. Araújo, and D. Freitas, "Real-time estimation of tyre–road friction peak with optimal linear parameterisation," *IET Control Theory & Applications*, vol. 6, no. 14, pp. 2257-2268, September 2012.

[17] Richard Bellman and Stuart Dreyfus, *Applied Dynamic Programming*. Princeton: Princeton Univ. Press, 1962.

[18] Jonathan Brembeck, Michael Panzirsch, and Peter Ritzer, "Robotic Motion," *dSPACE Magazine*, no. 1, pp. 52-57, May 2016. [Online]. https://www.dspace.com/en/pub/home/applicationfields/stories/dlr-robotic-motion.cfm

[19] Olle Sundstrom and Lino Guzzella, "A generic dynamic programming Matlab function," in *2009 IEEE Control Applications, (CCA) Intelligent Control, (ISIC)*, 2009, pp. 1625-1630.

[20] Alexander Schaub, Juan Carlos Ramirez de la Cruz, and Darius Burschka, "Autonomous Parking using a Highly Maneuverable Robotic Vehicle," in *IFAC World Congress*, vol. 19, 2014, pp. 2640-2645.

[21] Christoph Winter. (2022, April) YouTube, ROboMObil - Motion Planning. [Online]. https://www.youtube.com/watch?v=XElxGrBtcMs

[22] Johannes Ultsch, Jonas Mirwald, Jonathan Brembeck, and Ricardo de Castro, "Reinforcement Learning-based Path Following Control for a Vehicle with Variable Delay in the Drivetrain," in *31st IEEE Intelligent Vehicles Symposium*, Las Vegas, NV, USA, 2020, pp. 532-539.