Autonomous-Vehicle Maneuver Planning Using Segmentation and the Alternating Augmented Lagrangian Method [⋆]

Pavel Anistratov [*] **Björn Olofsson** [*,***] **Oleg Burdakov** [**] **Lars Nielsen** [*]

[*] *Division of Vehicular Systems, Department of Electrical Engineering, Linköping University, Sweden, (e-mail: pavel.anistratov@liu.se, bjorn.olofsson@liu.se, lars.nielsen@liu.se).*
[**] *Division of Optimization, Department of Mathematics, Linköping University, Sweden, (e-mail: oleg.burdakov@liu.se).*
[***] *Department of Automatic Control, Lund University, Sweden.*

Abstract: Segmenting a motion-planning problem into smaller subproblems could be beneficial in terms of computational complexity. This observation is used as a basis for a new sub-maneuver decomposition approach investigated in this paper in the context of optimal evasive maneuvers for autonomous ground vehicles. The recently published alternating augmented Lagrangian method is adopted and leveraged on, which turns out to fit the problem formulation with several attractive properties of the solution procedure. The decomposition is based on moving the coupling constraints between the sub-maneuvers into a separate coordination problem, which is possible to solve analytically. The remaining constraints and the objective function are decomposed into subproblems, one for each segment, which means that parallel computation is possible and beneficial. The method is implemented and evaluated in a safety-critical double lane-change scenario. By using the solution of a low-complexity initialization problem and applying warm-start techniques in the optimization, a solution is possible to obtain after just a few alternating iterations using the developed approach. The resulting computational time is lower than solving one optimization problem for the full maneuver.

Keywords: trajectory and path planning, motion planning, optimal control, problem decomposition, vehicle safety maneuvers.

1. INTRODUCTION

Efficient motion planning is an essential component in autonomous vehicles to allow safe and reliable operation under various conditions, including time and safety critical traffic situations. There are different ways to approach a motion-planning problem (see, *e.g.*, Paden et al. (2016)). Here, we consider the method where a motion-planning problem is formulated and subsequently solved as an optimization problem (see, *e.g.*, (Kelly, 2017)), and the focus is on investigating a possibility to combine the method with a new segmentation (decomposition) strategy. Formulating a motion-planning problem as an optimization problem brings many advantages (Sharp and Peng, 2011; Limebeer and Rao, 2015); it provides a mathematical framework allowing inclusion of various dynamic constraints for complex vehicle and tire–road interaction models (Berntorp et al., 2014), and it allows formulating limits on the state variables reflecting the current driving situation. Important tasks for the motion planning and control components are to prevent the vehicle from colliding with obstacles (Subosits and Gerdes, 2019), or leaving the road (Fors et al., 2019; Gao and Gordon, 2019).

There are often computational challenges originating from solving optimization problems related to motion-planning problems, in particular in online scenarios. Previous research has been presented that utilizes decomposition techniques to improve the computational performance for determining a solution of an optimization problem. The dual ascent method taking advantage of decomposability of the dual problem for a class of optimization formulations is presented in (Lasdon, 1968). The alternating direction method of multipliers (ADMM) (see, *e.g.*, (Boyd et al., 2011)) was proposed to combine the decomposability of the dual-ascent method with the improved convergence of the augmented Lagrangian method. In ADMM, an optimization problem is decomposed into smaller parts, and then iterations are performed that alternate between solving the subproblems and an overall coordination problem. ADMM for linear-convex optimal control problems is proposed in (O'Donoghue et al., 2013), where a decomposition approach down to each control interval is presented. Sindhwani et al. (2017) have approached constrained nonlinear optimal control problems by combining a trust-region strategy and ADMM such that a linearized problem is solved at each step. An alternative approach to adopt decomposition

[⋆] This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

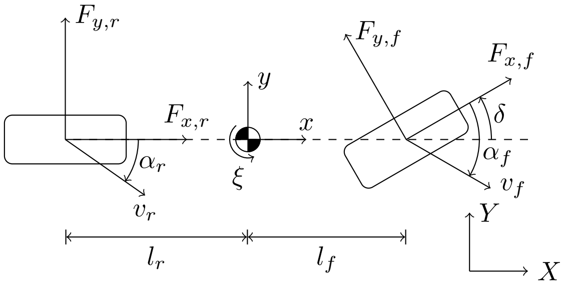

Fig. 1. The single-track model.

for nonlinear optimization problems is to distribute the problem constraints between several subproblems, and then search for a solution by alternating between these subproblems as proposed in (Galvan et al., 2019).

In this paper, we consider splitting of the motion-planning problem (*i.e.*, the corresponding optimization problem) for the full maneuver into several subproblems, each constituting one part of the full maneuver. The splitting is done from the vehicle-dynamics perspective as presented in (Anistratov et al., 2018b). By coupling the subproblems in a structured way, it is possible to solve them in parallel, thus allowing utilization of the computational power in modern multi-core platforms. An approach to decompose optimization problems for vehicle maneuvers using a duality-based decomposition method by relaxing a subset of the coupling constraints between the segments was studied in (Anistratov et al., 2019) for a double lane-change maneuver. The remaining coupling constraints were thereby substituted by state values assumed to be available *a priori* from offline computations. It is desirable to eliminate the dependence on pre-computed data in the method. In this paper, it is therefore considered to move all coupling constraints into another high-level coordination problem of low complexity by adopting and leveraging on the alternating augmented Lagrangian method proposed in (Galvan et al., 2019). The difference to the method in (O'Donoghue et al., 2013) is that the approach in this paper is applicable to nonlinear problems. Compared to (Sindhwani et al., 2017), the approach developed here does not require linearization at each step. As in comparison to the previously studied method (Anistratov et al., 2019), no pre-computed data are needed in the method in this paper. Initialization values are obtained by solving a highly reduced optimization problem, where the number of optimization variables are approximately one magnitude smaller than in the original problem.

2. VEHICLE MODEL

In this section, the vehicle model is first presented as a system of differential equations in the time domain and is then reformulated in terms of distance traveled along the road center lane.

2.1 Time Domain

The single-track model in the time domain (see, *e.g.*, Wong (2008)) is describing the vehicle dynamics by

$$m\dot{v}_x = F_{x,f}\cos(\delta) + F_{x,r} - F_{y,f}\sin(\delta) + mv_y r, \tag{1}$$

$$m\dot{v}_y = F_{y,f}\cos(\delta) + F_{y,r} + F_{x,f}\sin(\delta) - mv_x r, \tag{2}$$

$$I_Z\dot{r} = l_f F_{y,f}\cos(\delta) - l_r F_{y,r} + l_f F_{x,f}\sin(\delta), \tag{3}$$

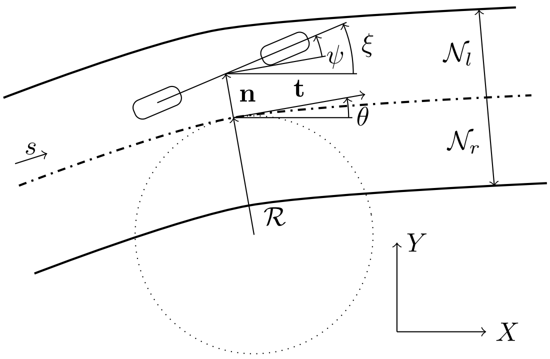

Fig. 2. Road coordinates-based description adopted from (Limebeer and Rao, 2015).

where v_x, v_y are the longitudinal and lateral velocities at the center-of-gravity, respectively, r is the yaw rate, δ is the steering angle, $F_{x,i}$, $F_{y,i}$, $i \in \{f, r\}$, are the longitudinal and lateral forces for the front and the rear wheels, respectively, m is the vehicle mass, I_Z is the vehicle chassis inertia in the yaw direction, and l_f, l_r are defined in Fig. 1. The vehicle global position XY and orientation ξ are determined by

$$\dot{X} = v_x\cos(\xi) - v_y\sin(\xi), \tag{4}$$

$$\dot{Y} = v_x\sin(\xi) + v_y\cos(\xi), \tag{5}$$

$$\dot{\xi} = r. \tag{6}$$

2.2 Road Coordinates

The road coordinates-based description is adopted from (Limebeer and Rao, 2015). The vehicle position is characterized by $s(t)$ (see Fig. 2), the distance traveled along the center of the road, and position $n(s(t))$ along the vector $\mathbf{n}(s(t))$ perpendicular to the track tangent $\mathbf{t}(s(t))$. It is assumed that $s(t)$ is an increasing function of time. The road is set between $N_l(s)$ and $N_r(s)$ along the vector $\mathbf{n}(s(t))$.

The vehicle dynamics is reformulated to depend on the distance traveled along the road center line by adopting the approach suggested in (Limebeer and Rao, 2015), where it is shown that such a reformulation reduces the number of problem state variables by one. It also allows to later formulate the optimal control problem with a fixed horizon (instead of having the final time as a free variable) and improves the computation time for solving the subproblems.

The time element $\mathrm{d}t$ is expressed in terms of a distance element $\mathrm{d}s$ by

$$\mathrm{d}t = \frac{\mathrm{d}t}{\mathrm{d}s}\mathrm{d}s = S_f(s)\mathrm{d}s, \tag{7}$$

where the transformation factor S_f from (Limebeer and Rao, 2015), using the curvature $\mathcal{C}(s)$ of the road (inverse of $\mathcal{R}$ in Fig. 2), is given by

$$S_f = \left(\frac{\mathrm{d}s}{\mathrm{d}t}\right)^{-1} = \frac{1 - n\mathcal{C}(s)}{v_x\cos(\psi) - v_y\sin(\psi)}. \tag{8}$$

Using the transformation factor S_f, the model equations (1)–(6) are represented as follows

$$mv_x' = (F_{x,f}\cos(\delta) + F_{x,r} - F_{y,f}\sin(\delta) + mv_y r)\, S_f, \quad (9)$$
$$mv_y' = (F_{y,f}\cos(\delta) + F_{y,r} + F_{x,f}\sin(\delta) - mv_x r)\, S_f, \quad (10)$$
$$I_Z r' = (l_f F_{y,f}\cos(\delta) - l_r F_{y,r} + l_f F_{x,f}\sin(\delta))\, S_f, \quad (11)$$
$$\psi' = rS_f - \mathcal{C}, \quad (12)$$
$$n' = (v_x\sin(\psi) + v_y\cos(\psi))S_f, \quad (13)$$

where $(\ )'$ denotes the derivative with respect to s and ψ is the vehicle orientation in the road frame.

The longitudinal forces and the steering angle are considered as inputs

$$u = \{F_{x,f}, F_{x,r}, \delta\}, \quad (14)$$

and the state vector is

$$x = \{v_x, v_y, r, \psi, n\}. \quad (15)$$

2.3 Tire Model

Since the focus of this paper is to illustrate the considered segmentation method, a comparably simple linear tire model from (Pacejka, 2006) is used, such that the slip angles for the front and rear wheel are defined as

$$\alpha_f = \frac{v_y + l_r r}{v_x} - \delta, \quad \alpha_r = \frac{v_y - l_r r}{v_x}, \quad (16)$$

and the lateral tire forces are

$$F_{y,i} = -C_{\alpha,i}\alpha_i, \ i \in \{f, r\}. \quad (17)$$

2.4 Lane-Deviation Penalty Function

Using a smooth approximation of the Heaviside step function with an offset a_o and a rising distance a_r

$$\widetilde{H}^{a_\mathrm{r}}_{a_\mathrm{o}}(a) = \frac{1}{2} + \frac{1}{2}\tanh\left(\frac{\pi}{a_\mathrm{r}}(a - a_\mathrm{o})\right), \quad (18)$$

the lane-deviation penalty (LDP) function from (Anistratov et al., 2018a), penalizing deviations from the own driving lane of the vehicle, is transformed to the road-coordinate formulation

$$H(n(s)) = \widetilde{H}^{n_\mathrm{r}}_{n_\mathrm{o}}(n(s)), \quad (19)$$

using the parameters n_o and n_r.

3. SEPARABLE OPTIMAL CONTROL PROBLEM

The motion-planning problem to compute a double lane-change maneuver is formulated as an optimal control problem. The problem is first presented in continuous infinite-dimensional form and then subsequently reformulated to a discretized version allowing separation into subproblems.

3.1 Continuous Formulation

The objective function is chosen as the integral of the weighted sum of the LDP function (19) and the squared value of the velocity deviation from the target velocity $v_{x,0}$. By substituting the algebraic relations (16)–(17) into the vehicle dynamics (9)–(13), the latter is formulated as the constraint $x' = G(x, u)$ in the optimal control problem. The optimal control problem between s_0 and s_f for starting state x_0 and final state x_f is

$$\begin{aligned}
\min_{x,u}.\quad & \int_{s_0}^{s_f} H(n(s)) + \gamma(v_x(s) - v_{x,0})^2 \mathrm{d}s \\
\text{s. t.}\quad & x(s_0) = x_0,\ x(s_f) = x_f,\ |\delta| \le \delta_{\max}, \\
& F_{x,f}^2 + (\eta F_{y,f})^2 \le (\mu m g l_r/L)^2,\ F_{x,f} \le 0, \\
& F_{x,r}^2 + (\eta F_{y,r})^2 \le (\mu m g l_f/L)^2,\ F_{x,r} \le 0, \\
& N_l(s) \le n(s) \le N_r(s),\ x' = G(x, u),
\end{aligned} \quad (20)$$

where γ is the weighting factor, the absolute value of the steering angle is limited by $\delta_{\max}$, $L = l_f + l_r$, and the forces for each tire are bounded by the friction ellipse (Pacejka, 2006), where η is the parameter of the ellipse. The longitudinal forces are non-positive as is common in double lane-change tests.

3.2 Discretization of Cost Function and Vehicle Dynamics

The vehicle dynamics is discretized for N control intervals using the multiple-shooting method (Bock and Plitt, 1984) with the Runge-Kutta method (RK4) (see, *e.g.*, (Ascher and Petzold, 1998)) as $x^{i+1} = \mathcal{F}(x^i, u^i)$. The resulting optimization problem with piecewise constant control inputs is

$$\begin{aligned}
\min_{x,u}.\quad & \sum_{i=1}^{N+1} \left(H(n^i) + \gamma(v_x^i - v_{x,0})^2\right)\Delta s \\
\text{s. t.}\quad & x^1 = x_0,\ x^{N+1} = x_f,\ \left|\delta^i\right| \le \delta_{\max}, \\
& (F_{x,f}^i)^2 + (\eta F_{y,f}^i)^2 \le (\mu m g l_r/L)^2,\ F_{x,f}^i \le 0, \\
& (F_{x,r}^i)^2 + (\eta F_{y,r}^i)^2 \le (\mu m g l_f/L)^2,\ F_{x,r}^i \le 0, \\
& N_l^i \le n^i \le N_r^i, \\
& x^{i+1} = \mathcal{F}(x^i, u^i),\ i \in \{1, \ldots, N\}.
\end{aligned} \quad (21)$$

A compact version of (21) could be formulated by representing its objective-function terms as $\mathcal{J}(x^i)$ and all inequality constraints as $\mathcal{G}(x^i, u^i) \le 0$ in the following format

$$\begin{aligned}
\min_{x,u}.\quad & \sum_{i=1}^{N+1} \mathcal{J}(x^i) \\
\text{s. t.}\quad & x^1 = x_0,\ x^{N+1} = x_f, \\
& \mathcal{G}(x^i, u^i) \le 0, \\
& x^{i+1} = \mathcal{F}(x^i, u^i),\ i \in \{1, \ldots, N\}.
\end{aligned} \quad (22)$$

3.3 Separable Discretized Formulation

The optimization problem (22) is reformulated to allow splitting it into M subproblems. This is achieved by dividing the state variables x and control inputs u in (22) into M segments and by introducing extra equality constraints with auxiliary variables y to make the new problem to be equivalent to (22). The new segmented state and control variables are denoted as x_j and u_j consisting of $p_j + 1$ and p_j vectors, respectively, representing p_j time steps. For compact notation, the compositions of the new segmented vectors are defined as

$$\mathcal{X} = \{x_1, \ldots, x_M\}, \quad (23)$$
$$\mathcal{U} = \{u_1, \ldots, u_M\}, \quad (24)$$
$$\mathcal{Y} = \{y_1, \ldots, y_{M+1}\}. \quad (25)$$

An equivalent formulation of (22), allowing later splitting into M subproblems, is the following

$$\begin{aligned} \min_{\mathcal{X},\mathcal{U},\mathcal{Y}} \quad & \sum_{j=1}^{M}\left\{f_j^0\mathcal{J}\left(x_j^1\right)+\sum_{i=2}^{p_j}\mathcal{J}\left(x_j^i\right)+f_j^f\mathcal{J}\left(x_j^{p_j+1}\right)\right\}\\ \text{s. t.} \quad & \mathcal{G}(x_j^i,u_j^i)\le 0,\ x_j^{i+1}=\mathcal{F}(x_j^i,u_j^i),\\ & x_j^1-y_j=0,\ x_j^{p_j+1}-y_{j+1}=0,\\ & i\in\{1,\dots,p_j\},\ j\in\{1,\dots,M\},\end{aligned}\tag{26}$$

where the factors f_j^0 and f_j^f are introduced in the objective function to take into account that the state values $x_j^{p_j+1}$ and x_{j+1}^1 correspond to the same traveled distance s along the center of the road. These factors are equal to one at the points corresponding to s_0 and s_f ($f_1^0 = f_M^f = 1$) and equal to 0.5 in all other cases. The coupling constraints involving the extra variables y_j are introduced to connect the adjacent segments with each other and to have the same initial and final states as in (22). For the latter, the variables y_1 and y_{M+1} are set to x_0 and x_f, respectively. All coupling constraints in (26) are compactly denoted as $\mathcal{E}$ for later use.

3.4 Applying Alternating Augmented Lagrangian Method

Seeking stationary points of the Lagrangian function is an approach to search for solutions of a constrained optimization problem (Nocedal and Wright, 2006). To increase robustness of the solution process, the augmented Lagrangian methods were developed (Boyd et al., 2011). Define the augmented Lagrangian (see, *e.g.*, (Galvan et al., 2019)) of (26) for the combined vector of multipliers

$$\Lambda = \{\lambda_1^0, \lambda_1^f, \dots, \lambda_M^0, \lambda_M^f\}, \tag{27}$$

and a penalty parameter $\tau \ge 0$ as

$$\begin{aligned} &\mathcal{L}_\tau(\mathcal{X},\mathcal{U},\mathcal{Y},\Lambda) = \\ &\quad \sum_{j=1}^{M}\left\{ f_j^0\mathcal{J}\left(x_j^1\right) + \sum_{i=2}^{p_j}\mathcal{J}\left(x_j^i\right) + f_j^f\mathcal{J}\left(x_j^{p_j+1}\right)\right.\\ &\qquad +\lambda_j^0\left(x_j^1-y_j\right)+\lambda_j^f\left(x_j^{p_j+1}-y_{j+1}\right)\\ &\qquad \left. + \frac{\tau}{2}\left\|x_j^1-y_j\right\|^2 + \frac{\tau}{2}\left\|x_j^{p_j+1}-y_{j+1}\right\|^2\right\}. \end{aligned}\tag{28}$$

The alternating augmented Lagrangian method in (Galvan et al., 2019) is adopted to move the constraints involving variables $\mathcal{Y}$ in (26) into a separable problem. Taking advantage of the special structure in (26), the rest of the problem is decomposed into M subproblems. Given the current iterate (${}^k\mathcal{X}$, ${}^k\mathcal{U}$ ${}^k\mathcal{Y}$,${}^k\Lambda$,${}^k\tau$), the steps are the following. For fixed ${}^k\mathcal{Y}$, ${}^k\Lambda$ (defined in (25), (27)), and ${}^k\tau$, the values of ${}^{k+1}\mathcal{X}$ and ${}^{k+1}\mathcal{U}$ are obtained by finding their components ${}^{k+1}x_j$, ${}^{k+1}u_j$ from solving M subproblems for each $j \in \{1,\dots,M\}$ as

$$\begin{aligned} \min_{x_j,u_j} \quad & f_j^0\mathcal{J}\left(x_j^1\right)+\sum_{i=2}^{p_j}\mathcal{J}\left(x_j^i\right)+f_j^f\mathcal{J}\left(x_j^{p_j+1}\right)\\ & + {}^k\lambda_j^0\left(x_j^1-{}^ky_j\right)+{}^k\lambda_j^f\left(x_j^{p_j+1}-{}^ky_{j+1}\right)\\ & + \frac{{}^k\tau}{2}\left\|x_j^1-{}^ky_j\right\|^2+\frac{{}^k\tau}{2}\left\|x_j^{p_j+1}-{}^ky_{j+1}\right\|^2\\ \text{s. t.} \quad & \mathcal{G}(x_j^i,u_j^i)\le 0,\ x_j^{i+1}=\mathcal{F}(x_j^i,u_j^i),\\ & i\in\{1,\dots,p_j\}. \end{aligned}\tag{29}$$

These M subproblems are independent of each other and can be solved in parallel. After that, for fixed ${}^{k+1}\mathcal{X}$, ${}^{k+1}\mathcal{U}$, ${}^k\Lambda$, and ${}^k\tau$, the new iterate ${}^{k+1}\mathcal{Y}$ is obtained by solving the problem

$$\begin{aligned} \min_{\mathcal{Y}} \quad & \sum_{j=1}^{M}\left\{{}^k\lambda_j^0\left({}^{k+1}x_j^1-y_j\right)+{}^k\lambda_j^f\left({}^{k+1}x_j^{p_j+1}-y_{j+1}\right)\right.\\ & \left. +\frac{{}^k\tau}{2}\left\|{}^{k+1}x_j^1-y_j\right\|^2+\frac{{}^k\tau}{2}\left\|{}^{k+1}x_j^{p_j+1}-y_{j+1}\right\|^2\right\}. \end{aligned}\tag{30}$$

This problem is possible to solve analytically to obtain the components of ${}^{k+1}\mathcal{Y}$ as

$$\begin{aligned} &{}^{k+1}y_1 = \frac{{}^k\lambda_1^0}{{}^k\tau} + {}^{k+1}x_1^0, \quad {}^{k+1}y_{M+1} = \frac{{}^k\lambda_M^f}{{}^k\tau} + {}^{k+1}x_M^f,\\ &{}^{k+1}y_j = \frac{{}^k\lambda_{j-1}^f + {}^k\lambda_j{}^0}{2\cdot{}^k\tau} + \frac{{}^{k+1}x_{j-1}^{p_j+1} + {}^{k+1}x_j^0}{2},\\ &\qquad j \in \{1,\dots,M\}. \end{aligned}\tag{31}$$

It should be noted that for the fixed components in x_0 and x_f, updates in the respective components of y_1 and y_{M+1} are not performed.

New multipliers ${}^{k+1}\Lambda$ are computed as follows (Galvan et al., 2019)

$${}^{k+1}\lambda_j^0 = {}^k\lambda_j^0 + {}^k\tau\left({}^{k+1}x_j^1 - {}^{k+1}y_j\right), \tag{32}$$

$${}^{k+1}\lambda_j^f = {}^k\lambda_j^f + {}^k\tau\left({}^{k+1}x_j^{p_j+1} - {}^{k+1}y_{j+1}\right), \tag{33}$$

$$j \in \{1,\dots,M\}. \tag{34}$$

The update rule for the penalty parameter τ is adopted from (Galvan et al., 2019) and is given by

$${}^{k+1}\tau = \begin{cases} {}^k\tau & \text{if } \left\|{}^{k+1}\mathcal{X} - {}^{k+1}\mathcal{Y}\right\| \le \sigma\left\|{}^k\mathcal{X} - {}^k\mathcal{Y}\right\|, \\ \alpha\cdot{}^k\tau & \text{otherwise,} \end{cases} \tag{35}$$

where α and σ are tuning parameters.

4. PARAMETERS AND IMPLEMENTATION

This section describes parameters used to formulate the optimal control problem for a double lane-change maneuver. Implementation aspects are also discussed.

4.1 Road Definition and Parameters

The road right-hand side $N_r(s)$ is defined, adapting the approach in (Anistratov et al., 2018a), using (18) by

$$N_r(s(t)) = N_{r,1}(\widetilde{H}^{s_\mathrm{r}}_{s_\mathrm{ou}}(s(t)) - \widetilde{H}^{s_\mathrm{r}}_{s_\mathrm{od}}(s(t))) + N_{r,2}, \tag{36}$$

where the parameters of the function are set to: $N_{r,1} = 2.5$ m, $N_{r,2} = -0.7$ m, $s_\mathrm{r} = 2$ m, $s_\mathrm{ou} = 23.5$ m, $s_\mathrm{od} = 36.5$ m. The function (36) is illustrated with the bottom red line in Fig. 3. The road left-hand side $N_l(s)$ is defined to have a constant value of 3.5 m. The road curvature $\mathcal{C}(s) = 0$.

4.2 Model and Problem Parameters

The vehicle model and tire parameters are shown in Table 1. The acceleration due to gravity is $g = 9.82\ \mathrm{ms}^{-2}$. The parameters in the LDP function (19) are set to $n_0 = 2$ and $n_\mathrm{r} = 2$; the weighting factor in the objective function (20) is

$\gamma = 0.2$. The vehicle steering angle is limited by $\delta_{\max} = \pi/3$. The vehicle initial state is $x_0 = \{60/3.6, 0, 0, 0, 0\}$ and the final state is $x_f = \{-, 0, 0, 0, 0\}$, where "$-$" represents no constraint. The problem is formulated between $s_0 = 0$ m and $s_f = 60$ m with the target velocity $v_{x,0} = (60/3.6)$ m/s. For the optimization of the full maneuver (21), $N = 100$. The parameters of the update rule are chosen as ${}^{1}\tau = 35$, $\sigma = 0.95$, and $\alpha = 1.02$.

4.3 Comments on Implementation

The method in Section 3 was implemented using the Python 3.7 programming language. Problems (21) and (29) are declared using the `nlpsol` interface in the framework CasADi (Andersson et al., 2019) and subsequently solved by IPOPT (Wächter and Biegler, 2005), together with the MA57 linear solver (HSL, 2019). The presented performance results are computed under the assumption that the segmented problems are computed in parallel threads with access to the shared memory. It means that the time for each iteration is the maximum computational time of the segments. The common implementation of Python (CPython) does not allow to execute threads in parallel, even when a multi-core processor is available. The use of the `Pool` class representing a pool of worker processes, provided by the `multiprocessing` library (Python Standard Library, 2019), results in overhead to transmit data to and from the process for each subproblem. A discussion on estimated overhead for both the sequential and the parallel case is given in Section 5.2.

4.4 Initialization

To improve performance of the investigated approach and to give initial values for the variables in the subproblems (29), an initialization strategy is developed. It is based on solving one optimization problem (21) with a loose tolerance bound and small N, which is typically one magnitude smaller than for the main problem. To initialize the solver for this problem, the longitudinal velocity v_x for the complete maneuver is set to the initial velocity and other variables are set to zero. Having a solution of this small problem, it is possible to initialize the variables $\mathcal{X}$, $\mathcal{U}$, and $\mathcal{Y}$ for the considered subproblems using linear interpolation of the resulting x and u. For each constraint of problem (21), the used solver outputs the associated dual variables. By considering the dual variables of the constraints $x^{i+1} = \mathcal{F}(x^i, u^i)$ to be associated with point s^{i+1}, a linear-interpolation approach is applied to obtain Λ at the corresponding points of s. The interpolation method for the dual variables of the constraints $x^{i+1} = \mathcal{F}(x^i, u^i)$ is also applied to initialize the respective constraints at the discretization points of s in the subproblems (29).

Table 1. Chassis and tire parameters.

Notation	Value	Unit	Notation	Value	Unit
l_f	1.3	m	$C_{\alpha,f}$	$17 \cdot 10^3$	N
l_r	1.5	m	$C_{\alpha,r}$	$20 \cdot 10^3$	N
m	2100	kg	μ	0.8	[-]
I_Z	3900	kgm^2	η	1	[-]

4.5 Scaling

Using the results from the initial solution, the vehicle orientation ψ is scaled by the ratio involving multipliers associated with the discretization constraint for the ψ dynamics (denoted as $\widetilde{\psi}^i$) and for the other state variables, whose multipliers are denoted by $\widetilde{\Upsilon}^i = \{\widetilde{v}_x^i, \widetilde{v}_y^i, \widetilde{r}^i, \widetilde{n}^i\}$, as

$$\beta_\psi = \frac{\max_i(\widetilde{\Upsilon}^i) - \min_i(\widetilde{\Upsilon}^i)}{\max_i(\widetilde{\psi}^i) - \min_i(\widetilde{\psi}^i)}, \tag{37}$$

where $i \in \{1, \dots, N\}$. The effect of the ψ-scaling is discussed in Section 5.3. Also the tire forces are scaled by $\beta_F = 1/1000$ for improved performance in the solver. The inverse scaling is applied on the affected variables when visualizing the final solution.

4.6 Solver Configuration

To decrease the computation time for the subproblems (29) and since an accurate solution at each update step is not required, strategies for the interior-point solver configuration are adopted from (Wang and Boyd, 2010). The maximum number of iterations for the IPOPT solver is chosen as 12. A warm-start approach is used, with the target value and the minimum value of the barrier parameter μ in the solver set to 0.01, and all the warm-start related parameters of the solver (*i.e.*, how to change different boundaries before a new run) are set to 10^{-9}.

5. RESULTS

The method in Section 3 is illustrated for a double lane-change maneuver. An analysis of the effects of variable scaling, parallel computation, and segmentation with different number of segments is also presented.

5.1 Solution of the Decomposed Problem

The problem is divided into three segments (three subproblems on the format (29)) with $p_1 = 25$, $p_2 = 51$, and $p_3 = 24$. These points are chosen, such that segmentation points correspond to extrema of the vehicle orientation ψ and the yaw rate r as developed in (Anistratov et al., 2018b). The first segmentation point is the extremum of the vehicle orientation before the obstacle. The second segmentation point is the extremum of the yaw rate after the obstacle. Information about these variables is available from the initial solution (where $N = 10$) and quadratic spline interpolation is applied to find the nearest points for the case of $N = 100$.

The solution of the decomposed problem and the solution of the full problem are shown in Fig. 3. The variables for the vehicle orientation and the tire forces in Fig. 3 (and in Figs. 5–6 in Section 5.2) are scaled back to the original ranges (see Section 4.5). The solution is obtained for a fixed number of alternating iterations, here 30. The solution of the decomposed problem is overlaying the solution of the full problem most of the time. A small difference is observed for the longitudinal velocity v_x towards the end of the maneuver. The discrepancy could be explained by a comparably low penalty on the velocity relative to the penalty on the position of the vehicle (the vehicle position

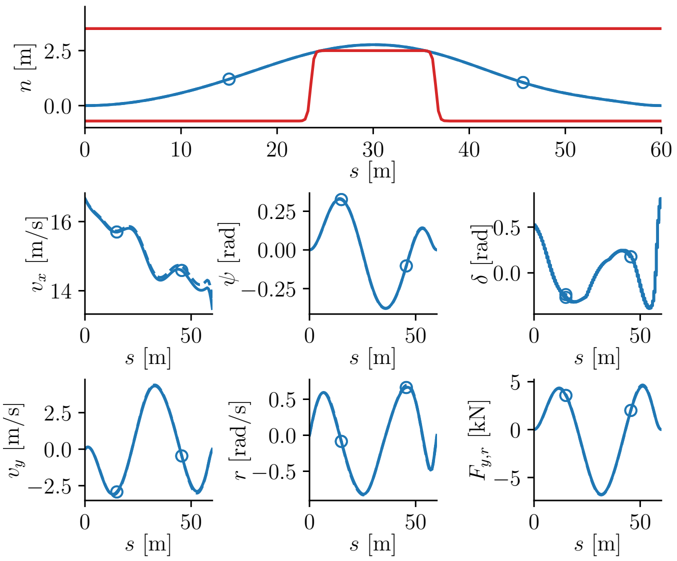

Fig. 3. The solution of the decomposed problem (solid lines) overlaying the solution of the full problem (dashed lines). Circles are shown at the segmentation points.

n is in good agreement in Fig. 3) in the objective function in (20). Circles are shown for the state variables of the respective segment at each segmentation point; the circles are overlaying meaning that the variables are connected between segments. However, for the steering-angle input δ, there is a small difference at the first segmentation point because there are no constraints on the piecewise constant inputs between control intervals in (21) and consequently not in (29).

Figure 4 shows the evolution for a number of variables and progress measures. Each point in the figure is one iteration of the algorithm shown against the solution time (measured in the implementation as described in Section 4.3). The time to setup the problem internally in CasADi is not taken into account here and this is further discussed in Section 5.2. From left to right, top to bottom the figure shows:

- The error-norm of the coupling constraints, $\|\mathcal{E}\|$.
- Penalty parameter τ.
- Multipliers for the coupling constraint $x_1^{p_1+1} - y_2 = 0$ of the first segment, λ_1^f.
- Errors for each state variable between the first and the second segment at the first segmentation point, $x_2^1 - x_1^{p_1+1}$.
- Multipliers for the coupling constraint $x_2^{p_2+1} - y_3 = 0$ of the second segment, λ_2^f.
- Errors for each state variable between the second and the third segment at the second segmentation point, $x_3^1 - x_2^{p_2+1}$.

In the plots of multipliers in Fig. 4, the black circles show the values of the multipliers obtained for the solution of the full problem. The values are shown for reference purpose only, and they are not available during the solution process of the decomposed problem. Figure 4 does not show the multipliers λ_2^0 and λ_3^0 since in the solution they are just equal to the negative values of the shown multipliers λ_1^f and λ_2^f, respectively. The deviations of the coupling constraints at the beginning $(x_1^1 - y_1)$ and at the end $(x_3^{p_3+1} - y_4)$ of the maneuver are not shown in the figures. However, they are still accounted for in the coupling error $\|\mathcal{E}\|$.

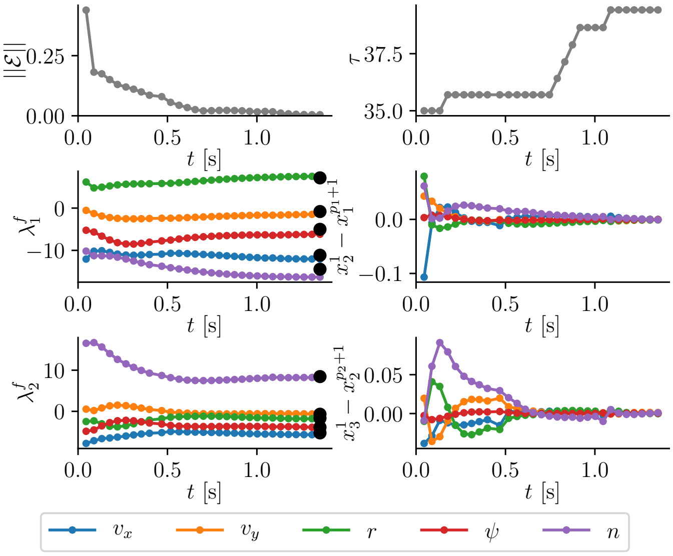

Fig. 4. Variables and progress measures for the solution of the segmented problem. See the data description in Section 5.1. Note that the legend colors for the plots in the left column refer to corresponding multipliers.

Compared to (O'Donoghue et al., 2013; Sindhwani et al., 2017), the investigated decomposition method has a lower number of decomposed subproblems relative to the number of control segments ($M \ll N$). The lower number of decomposed subproblems is chosen under considerations of the typical situation when a limited number of computation cores are available and to decrease the overhead time, while taking benefit of already available efficient interior-point solvers for the subproblems.

5.2 Time Performance

Performance evaluations for the implementation (see Section 4.3 for details) were obtained on a laptop computer with an Intel i7-8550U processor. Averaging over 50 executions, it takes 0.13 s to solve the optimization problem for the full maneuver. The initialization problem of the method in Section 3 takes 0.005 s to solve. An average solution time of one iteration of the decomposed problem (with the assumption of parallel computation of the subproblems) is approximately 0.04 s. With a lower number of alternating iterations, it is possible to obtain a solution (acceptable for certain applications) of the decomposed problem faster than solution of the original problem. Figure 5 shows the solution after two alternating iterations. The solution time including the initialization (see Section 3) is 0.08 s. Comparing Fig. 3 (after 30 alternating iterations) with Fig. 5, it is visible that the agreement between the solution of the decomposed problem and the solution of the full problem is lower for the latter case. The solution, however, is obtained much faster.

For certain applications, not only the solution time, but also the time required to instantiate a solver for the optimization

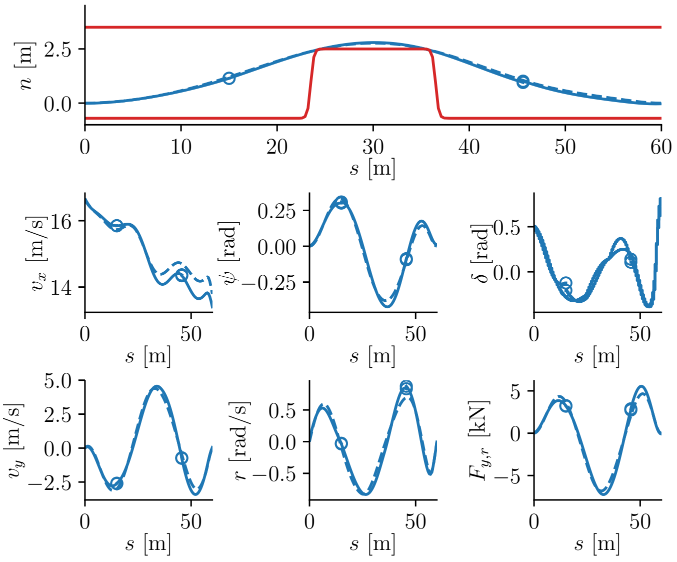

Fig. 5. The solution of the decomposed problem after two alternating iterations (solid lines) almost overlaying the solution of the full problem (dashed lines).

problem is of importance. For the considered scenario, it takes 0.44 s to instantiate the solver for the full maneuver in CasADi, while it takes 0.04 s and 0.22 s for the initialization and subproblems, respectively. With the assumption on parallel computations, it is thus faster to formulate the decomposed problem.

As described in Section 4.3, an implementation of the described method with parallel computations of the subproblems has been made. For 30 alternating iterations, the average computational overhead for each alternating iteration is 5.6 % of the maximum solution time of the subproblems. When the subproblems are actually solved in parallel, the average solution time per iteration increases by 19 % in the current implementation. The overhead connected with updating the multipliers and auxiliary variables together with the overhead of data transfer to a parallel process is 14 % of the maximum solution time of the subproblems.

The method is flexible in terms of number of segments. By increasing the number of segments, each associated subproblem gets smaller. It means that the subproblem is faster to initialize and to solve. By increasing the number of segments to seven equally long subproblems in distance s, the average solution time for one iteration is 0.012 s (0.04 s for the segmentation into three segments). For eleven segments, an average solution time for one iteration is 0.009 s. Figure 6 shows the solution obtained after two alternating iterations (0.023 s of solution time including the initialization problem). For increased number of segments, a solution could be obtained faster, if there are sufficiently many processor cores available for computations.

5.3 Importance of Scaling

It turned out that scaling of the vehicle orientation ψ was important for rapid convergence of the multipliers Λ. The evolution of the problem variables when the scaling is not applied is shown in Fig. 7. When it is compared to Fig. 4, it is visible that the multiplier λ_1^f corresponding to the vehicle orientation ψ (red line in the λ_1^f plot in the figures) does not converge to the values obtained for the solution of the full problem even after 30 iterations (60 iterations are needed instead with the maximum number of iterations for the IPOPT solver increased to 20). The discrepancy between the first and the second segments for ψ in Fig. 7 is slow to decrease (red line for $x_2^1 - x_1^{p_1+1}$ in the figures), as well as the coupling error $\|\mathcal{E}\|$. Without scaling, more computation time is needed to obtain a solution with the same quality as in the case with scaling. The observation could be related to the dual variables of the dynamic constraints obtained from the solution of the full maneuver (Fig. 8). The dual variables associated with ψ change over a much wider range

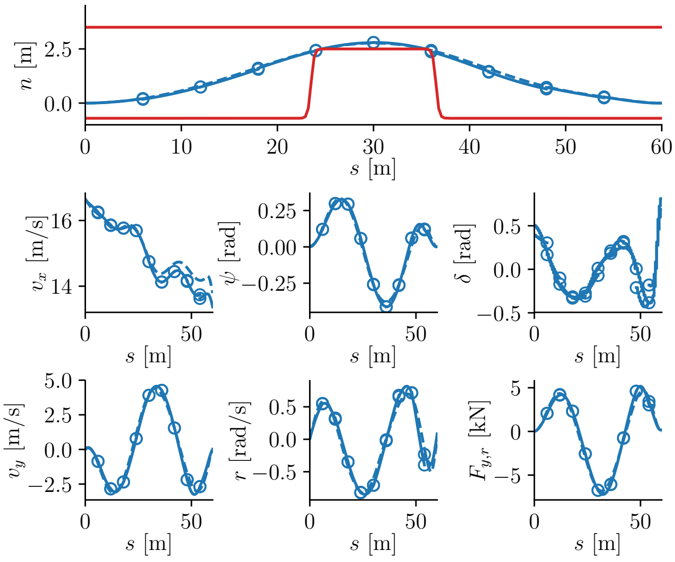

Fig. 6. The solution of the decomposed problem (solid lines, eleven segments) almost overlaying the solution of the full problem (dashed lines).

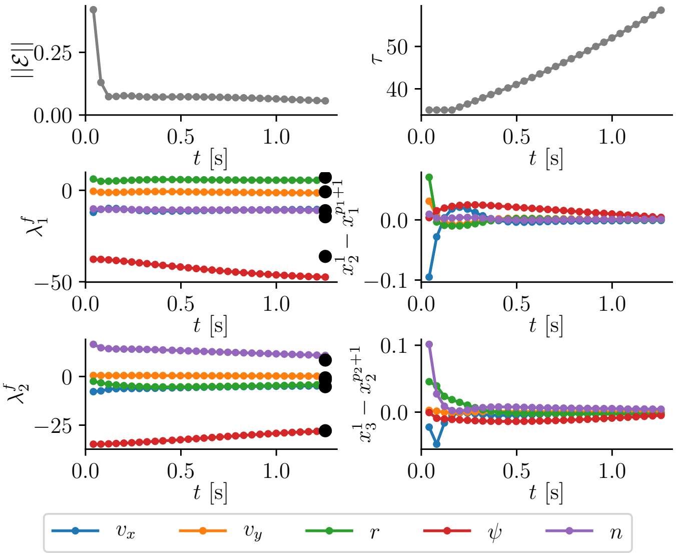

Fig. 7. Variables and progress measures for the segmented problem. See the data description in Section 5.1. Note that the legend colors for the plots in the left column refer to corresponding multipliers. Scaling for ψ is not performed here.

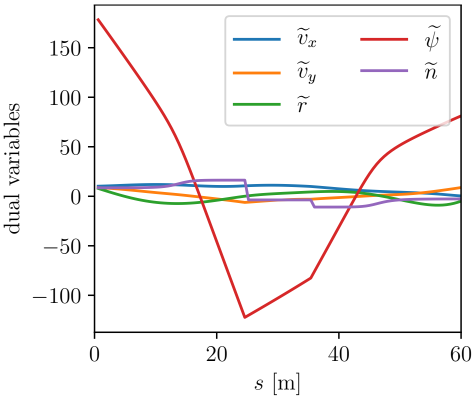

Fig. 8. Dual variables associated with the dynamic constraints in the solution of the full maneuver.

relative to the other dual variables over the path. By scaling the ψ variable, such that the associated dual variables are changing over the same range as the other dual variables, the convergence characteristics are improved.

6. CONCLUSION

The presented decomposition method for the optimal control problem arising in motion planning of a double lane-change maneuver was shown to be effective in the considered scenario. The solution is computed after just a few alternating iterations, with the first iteration initialized using the results of a particular computational procedure. The scaling of the state variable representing vehicle orientation in the problem was noted to be important for achieving rapid convergence of the method. The resulting maneuver was compared to the results obtained for solving the full maneuver in one step, and the correspondence turned out to be high already after a few iterations resulting in lower computational times. The results indicate the usefulness of the decomposition method.

REFERENCES

Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., and Diehl, M. (2019). CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1), 1–36.

Anistratov, P., Olofsson, B., and Nielsen, L. (2018a). Lane-deviation penalty for autonomous avoidance maneuvers. In *Proceedings of the 14th International Symposium on Advanced Vehicle Control*. Beijing, China.

Anistratov, P., Olofsson, B., and Nielsen, L. (2018b). Segmentation and merging of autonomous at-the-limit maneuvers for ground vehicles. In *Proceedings of the 14th International Symposium on Advanced Vehicle Control*. Beijing, China.

Anistratov, P., Olofsson, B., and Nielsen, L. (2019). Efficient motion planning for autonomous vehicle maneuvers using duality-based decomposition. In *9th IFAC Symposium on Advances in Automotive Control*, 78–84. Orléans, France.

Ascher, U.M. and Petzold, L.R. (1998). *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM: Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Berntorp, K., Olofsson, B., Lundahl, K., and Nielsen, L. (2014). Models and methodology for optimal trajectory generation in safety-critical road–vehicle manoeuvres. *Vehicle System Dynamics*, 52(10), 1304–1332.

Bock, H.G. and Plitt, K.J. (1984). A multiple shooting algorithm for direct solution of optimal control problems. *IFAC Proceedings Volumes*, 17(2), 1603–1608.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3, 1–122.

Fors, V., Olofsson, B., and Nielsen, L. (2019). Yaw-moment control at-the-limit of friction using individual front-wheel steering and four-wheel braking. In *9th IFAC Symposium on Advances in Automotive Control*, 458–464. Orléans, France.

Galvan, G., Lapucci, M., Levato, T., and Sciandrone, M. (2019). An alternating augmented Lagrangian method for constrained nonconvex optimization. *Optimization Methods and Software*. doi:10.1080/10556788.2019.1576177.

Gao, Y. and Gordon, T. (2019). Optimal control of vehicle dynamics for the prevention of road departure on curved roads. *IEEE Transactions on Vehicular Technology*, 68(10), 9370–9384.

HSL (2019). A collection of Fortran codes for large scale scientific computation. `http://www.hsl.rl.ac.uk`. (Date accessed: 28.10.2019).

Kelly, M. (2017). An introduction to trajectory optimization: How to do your own direct collocation. *SIAM Review*, 59(4), 849–904.

Lasdon, L. (1968). Duality and decomposition in mathematical programming. *IEEE Transactions on Systems Science and Cybernetic*, 4(2), 86–100.

Limebeer, D. and Rao, A. (2015). Faster, higher, and greener: Vehicular optimal control. *Control Systems Magazine*, 35(2), 36–56.

Nocedal, J. and Wright, S. (2006). *Numerical Optimization (Springer Series in Operations Research and Financial Engineering)*. Springer, New York, NY, second edition.

O'Donoghue, B., Stathopoulos, G., and Boyd, S. (2013). A splitting method for optimal control. *IEEE Transactions on Control Systems Technology*, 21(6), 2432–2442.

Pacejka, H. (2006). *Tyre and vehicle dynamics*. Butterworth-Heinemann, Oxford, UK, second edition.

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.

Python Standard Library (2019). multiprocessing – Process-based parallelism – Python 3.7.5 documentation. `https://docs.python.org/3.7/library/multiprocessing.html`. (Date accessed: 24.10.2019).

Sharp, R.S. and Peng, H. (2011). Vehicle dynamics applications of optimal control theory. *Vehicle System Dynamics*, 49(7), 1073–1111.

Sindhwani, V., Roelofs, R., and Kalakrishnan, M. (2017). Sequential operator splitting for constrained nonlinear optimal control. In *American Control Conference*, 4864–4871. Seattle, WA, USA.

Subosits, J. and Gerdes, J.C. (2019). From the racetrack to the road: Real-time trajectory replanning for autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 4(2), 309–320.

Wächter, A. and Biegler, L.T. (2005). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1), 25–57.

Wang, Y. and Boyd, S. (2010). Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, 18(2), 267–278.

Wong, J.Y. (2008). *Theory of Ground Vehicles*. Wiley, Hoboken, NJ, fourth edition.