Preprints, 10th IFAC International Symposium on
Advances in Automotive Control
Columbus, Ohio, USA, August 28-30, 2022

Dynamics-Based Optimal Motion Planning of Multiple Lane Changes using Segmentation ⋆

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

[*] *Division of Vehicular Systems, Linköping University, Linköping, Sweden.*
[**] *Department of Automatic Control, Lund University, Lund, Sweden.*
[***] *Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden (e-mail: pavel.anistratov@chalmers.se).*

Abstract: Avoidance maneuvers at normal driving speed or higher are demanding driving situations that force the vehicle to the limit of tire–road friction in critical situations. To study and develop control for these situations, dynamic optimization has been in growing use in research. One idea to handle such optimization computations effectively is to divide the total maneuver into a sequence of sub-maneuvers and to associate a segmented optimization problem to each sub-maneuver. Here, the alternating augmented Lagrangian method is adopted, which like many other optimization methods benefits strongly from a good initialization, and to that purpose a method with motion candidates is proposed to get an initially feasible motion. The two main contributions are, firstly, the method for computing an initially feasible motion that is found to use obstacle positions and progress of vehicle variables to its advantage, and secondly, the integration with a subsequent step with segmented optimization showing clear improvements in paths and trajectories. Overall, the combined method is able to handle driving scenarios at demanding speeds.

1. INTRODUCTION

Motion planning is generally a challenging task with no universal solution suited for all applications, and various methods have been proposed to address different planning problems (LaValle, 2006; Paden et al., 2016). One class of approaches uses optimization-based methods (Limebeer and Rao, 2015), which are well suited for dynamic models used in autonomous-driving applications. Such methods compute an optimal maneuver with respect to a criterion, and straightforwardly allow incorporation of constraints in the formulation and the subsequent solution. The optimization is in this paper done with the view that a complete maneuver is a sequence of sub-maneuvers, which is inspired by the use of motion primitives in motion-planning problems (Pivtoraiko and Kelly, 2011; Bergman et al., 2019). Such planning has successfully been applied to situations where a kinematic model is feasible to use. However, for a critical maneuver at normal or high driving speeds, it is necessary to consider the dynamics of the vehicle, perhaps even at the limit of friction. Because of such dynamics, there is a combinatorial explosion of possible motions compared to the kinematic case, and it is challenging to precompute and store sufficiently many primitives. In this paper, another line of thought is used for dynamics-based motion planning. It leads to: computation of a scenario division based on obstacle positions, computation of motion candidates and an initially feasible motion, scenario segmentation based on extremum values of vehicle motion, and finally segmented optimization.

1.1 Previous Research

This paper presents and investigates a motion-planning approach that uses segmentation and the alternating augmented Lagrangian method (Galvan et al., 2019), and it generalizes the previously proposed segment-based optimal motion-planning approach in (Anistratov et al., 2020), which considered avoidance of one single obstacle on the road by a double lane-change maneuver, whereas one extension in this paper is to avoid several consecutive obstacles, and this of course causes increased interaction between sub-maneuvers. The main example used is a scenario with two consecutive double lane changes with dimensions inspired by the standard ISO 3888-2:2011 (2011).

Methods for motion planning using search and state-space discretization, e.g., lattice-based trajectory planners (Likhachev and Ferguson, 2009; Pivtoraiko and Kelly, 2011; Bergman et al., 2019), find an approximate solution where the quality is dependent on the resolution of the state-space discretization, which in turn is limited by the size and the time for pre-computation of the state lattice and motion primitives. This often restricts the state-space discretization methods to kinematic models with a limited number of discretization points for each state. Nevertheless, for special classes of actions, motion primitives have been used for dynamic models as well (Frazzoli et al., 2002; Gray et al., 2012; Bergman et al., 2020b; Ajanovic et al., 2020).

To improve computational performance and solution quality, two main ideas have been adopted, segmentation and two-step calculations. When solving an optimization problem, decomposition techniques have been used (see, e.g., (Boyd et al., 2011)), even down to each control interval (O'Donoghue et al., 2013), or using a linearized problem at each step (Sindhwani et al., 2017). Optimization convergence often benefits largely from a good initialization, so the other main idea is the two-step approach where the first step finds an initialization and the second step is the optimization, see (Diehl et al., 2005; Svensson et al., 2019; Bergman et al., 2020a; Ljungqvist et al., 2020; Bergman et al., 2021) for some examples.

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

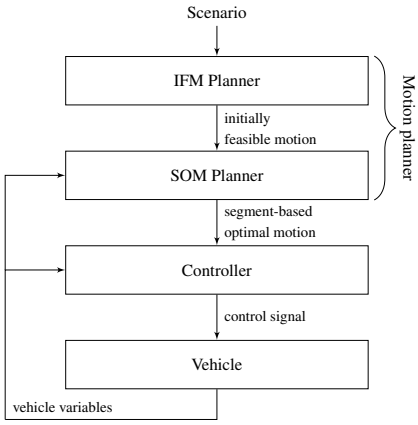

Fig. 1. Overall layout of the motion planner and its interaction with the control and the vehicle.

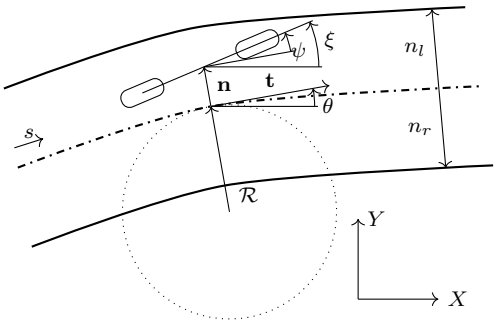

Fig. 2. Description of vehicle motion using road coordinates adopted from (Limebeer and Rao, 2015).

1.2 Contributions and Paper Outline

This paper aims for solving the complete optimal control problem achieving both path and trajectory planning even for high performance, at-the-limit maneuvers. This is a challenging computational task, and to handle it a number of new ideas are contributed. One is to integrate and further develop the segmentation technique in the optimization problem itself from (Anistratov et al., 2020) into a complete motion planner. Another important new part is the first step of finding an initially feasible motion (IFM). These main ideas and contributions of this paper are captured in Figures 1, 3, and 4. Figure 1 gives a high-level overview of the whole motion-planning approach. The motion-planning part consists of two main blocks, and it computes inputs to the controller by utilizing situational awareness about the scenario. The Initially Feasible Motion Planner (IFM Planner), see also Figure 3, is responsible for generating an initially feasible motion, and it is detailed in Section 3. The Segment-Based Optimal Motion Planner (SOM Planner), see also Figure 4, outputs a segment-based optimal motion for the maneuver by using optimization and segmentation techniques. The SOM Planner takes updated vehicle variables into account to compute updated motion plans. Notice that Figure 1 is one integrated computational scheme. Thus, it is not so that a path is planned for a path tracker to follow, but rather that a complete motion is computed in two main steps. The details of this planner are covered in Section 4. Parameters and implementation are treated in Section 5, and the results are presented in Section 6. Section 7 gives the conclusions.

2. MODELING

The single-track model is here describing the vehicle dynamics using road coordinates (Limebeer and Rao, 2015; Anistratov et al., 2020). The derivatives of the steering angle and the longitudinal tire forces are considered as inputs, to allow having constraints on these variables in the optimization formulation. The vehicle position is characterized by the distance s traveled along the center of the road and position $n(s)$ along the vector $\mathbf{n}(s)$ perpendicular to the track tangent $\mathbf{t}(s)$ (see Figure 2). Using the curvature $\mathcal{C}(s)$ of the road (i.e., the inverse of $\mathcal{R}$ in Figure 2), the transformation factor to change the independent variable in the model from time t to distance s is given by (Limebeer and Rao, 2015)

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

where v_x, v_y are the longitudinal and lateral velocities at the center of gravity, respectively, and ψ is the vehicle orientation in the road frame. Using the transformation factor S_f, the model equations are

$$mv_x' = (F_{x,f}\cos(\delta) + F_{x,r} - F_{y,f}\sin(\delta) + mv_y r)\, S_f, \tag{2a}$$
$$mv_y' = (F_{y,f}\cos(\delta) + F_{y,r} + F_{x,f}\sin(\delta) - mv_x r)\, S_f, \tag{2b}$$
$$I_Z r' = (l_f F_{y,f}\cos(\delta) - l_r F_{y,r} + l_f F_{x,f}\sin(\delta))\, S_f, \tag{2c}$$
$$\psi' = rS_f - \mathcal{C},\ n' = (v_x \sin(\psi) + v_y \cos(\psi))S_f, \tag{2d}$$
$$F_{x,f}' = u_{F_{x,f}} S_f,\ F_{x,r}' = u_{F_{x,r}} S_f,\ \delta' = u_\delta S_f, \tag{2e}$$

where $(\,)'$ denotes derivative with respect to s, and 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, respectively, for the front and rear wheels, m is the vehicle mass, I_Z is the vehicle chassis inertia along the yaw direction, and l_f, l_r are lengths to the center of gravity from the front and rear wheels, respectively. The state vector $\bar{x}$ and the input vector $\bar{u}$ are

$$\bar{x} = (v_x, v_y, r, \psi, n, F_{x,f}, F_{x,r}, \delta),\ \bar{u} = (u_{F_{x,f}}, u_{F_{x,r}}, u_\delta).$$

The tire forces are described using a tire model from (Pacejka, 2006). The lateral forces are given by

$$F_{y,i} = -C_{\alpha,i}\alpha_i, \quad i \in \{f, r\}, \tag{3}$$

where α_i is the slip angle and $C_{\alpha,i}$ is the lateral tire stiffness. To capture saturation of the tire behavior in the formulation of the motion planner, the combined longitudinal and lateral forces are constrained by the friction ellipse (11c)–(11d).

3. IFM PLANNER

The upper block in the motion-planning architecture (IFM Planner in Figure 1) is considered first. This block is in turn divided into four blocks, see Figure 3, and the overall purpose of the block is to compute an initially feasible motion for the vehicle in a given scenario.

3.1 Scenario Division

A scenario is here defined between the coordinates s_s and s_f, and it includes information about L box-shaped obstacles along the road in terms of their center $s_{l,c}$ and length $s_{l,w}$ along the s coordinate, where $l \in \{1, \ldots, L\}$ is used for indexing obstacles throughout this section. In the Scenario Division block, a simple strategy to divide the scenario is considered; the choice of s-points to use for the division is determined using the obstacle

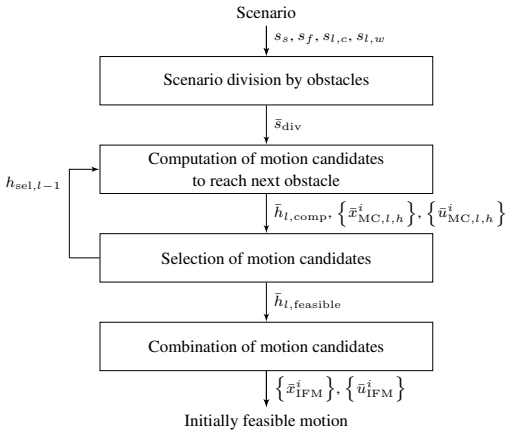

Fig. 3. Layout of the IFM Planner block (see Figure 1 for the overall layout of the motion planner).

positions and lengths. The total vector $\bar{s}_{\mathrm{div}}$ of s-coordinates to use for the scenario division is the following

$$\bar{s}_{\mathrm{div}} = \left(s_s, s_{1,c} - \frac{1}{4}s_{1,w}, \ldots, s_{L,c} - \frac{1}{4}s_{L,w}, s_f \right), \quad (4)$$

which means that each motion segment leads the vehicle to reach the s-coordinate corresponding to $1/4$ of the length of each obstacle. Such a choice of $\bar{s}_{\mathrm{div}}$ may imply that the computed motion candidates have different length in s, depending on the obstacle placement, but this is straightforward to handle in the subsequent blocks.

A summary of the steps of the Scenario Division block is:

(1) Define s_s and s_f, and set $s_{l,c}$ and $s_{l,w}$ of the L obstacles.
(2) Compute $\bar{s}_{\mathrm{div}}$ in (4), which in total contains $L + 2$ points.

3.2 Computation of Motion Candidates

After the scenario has been divided according to the previous subsection, the vector $\bar{s}_{\mathrm{div}}$ is known. The scenario in addition defines the road boundaries $n_{\min}$ and $n_{\max}$ and the upper $n_{l,u}$ and lower $n_{l,d}$ boundaries of the n-coordinate for the L box-shaped obstacles. To the purpose of presentation, the road space occupied by the obstacles is denoted as $\mathcal{O}_{\mathrm{obs}}$, and the obstacle-free space is $\mathcal{O}_{\mathrm{free}}$. The nominal longitudinal velocity is $v_{x,\mathrm{in}}$.

The vehicle-dynamics model (2) with included relation for the lateral forces (3) is discretized using the Runge-Kutta method RK4 (Ascher and Petzold, 1998) and a fixed step size Δs as $\bar{x}^{i+1} = \mathcal{F}(\bar{x}^i, \bar{u}^i)$.

In the Computation of Motion Candidates block, there is a need to determine the coordinates for the motion candidates to aim for along the $\mathbf{n}$-axis. In total H different n-coordinates indexed with h are used when computing motion candidates, and these are located on a grid with a spacing of Δn according to

$$n_h = n_{\min} + h\Delta n, \quad h \in \{0, \ldots, H\}. \quad (5)$$

Using this approach, which is a special case of state-space sampling (Howard et al., 2008), and a trajectory rollout method (Svensson et al., 2019), motion candidates are generated in simulation using a controller based on an infinite-horizon linear-quadratic regulator (LQR) (Anderson and Moore, 1989) for the vehicle model. The control gain matrix F is computed once for the scenario for a quadratic criterion minimizing the weighted sum of the deviation of the state trajectory from the input trajectory $\bar{x}^i_{\mathrm{in}}$ and the use of actuation with weighting matrices $\breve{Q}$ and $\breve{R}$, respectively. The gain matrix is obtained by solving the discrete-time algebraic Riccati equation (Anderson and Moore, 1989).

As feedback from the Selection of Motion Candidates block (described in Section 3.3), the index $h_{\mathrm{sel},l-1}$ of the previously selected motion candidate is known. The motion candidate to reach obstacle l at the position n_h along the n-coordinate starts from the end state of the previously selected ($h_{\mathrm{sel},l-1}$) motion candidate (or the start of the scenario $\bar{x}_0$ to reach the first obstacle). The motion candidates are obtained by iteratively computing

$$\bar{x}^{i+1}_{\mathrm{MC},l,h} = \mathcal{F}(\bar{x}^i_{\mathrm{MC},l,h}, \bar{u}^i_{\mathrm{MC},l,h}), \quad i \in \{1, \ldots, N_l\}, \quad (6a)$$

$$\bar{u}^i_{\mathrm{MC},l,h} = -F \cdot (\bar{x}^i_{\mathrm{MC},l,h} - \bar{x}^i_{\mathrm{in},l,h}), \quad (6b)$$

where, for each obstacle l, the number of discretization points is N_l, the initial state $\bar{x}^1_{\mathrm{MC},l,h} = \bar{x}^{N_l+1}_{\mathrm{MC},l-1,h_{\mathrm{sel},l-1}}$, and $\bar{x}^i_{\mathrm{in},l,h}$ is the input trajectory for that motion candidate to the LQR. The input longitudinal-velocity component of $\bar{x}^i_{\mathrm{in},l,h}$ is set to the nominal longitudinal velocity for all motion candidates in the scenario. The input path to the LQR is in the simulation defined using a fifth-order polynomial. Adopting the path polynomial from (He et al., 2018), the input lateral coordinate $n_{\mathrm{in},l,h}$ for each motion candidate is then formulated in a continuous form as

$$n_{\mathrm{in},l,h}(s) = n_{s,l} + (n_h - n_{s,l})\left(10\tilde{s}^3 - 15\tilde{s}^4 + 6\tilde{s}^5\right), \quad (7a)$$

$$\tilde{s} = \frac{s - s_{\mathrm{div},l-1}}{s_{\mathrm{div},l} - s_{\mathrm{div},l-1}}, \quad (7b)$$

where $n_{s,l}$ is the n-coordinate at the beginning of the current motion candidate, which is equal to the final n-coordinate of the previously selected motion candidate, or the component n_0 of $\bar{x}_0$ for the motion candidates computed to reach the first obstacle. Such a construction of the input path means that the computation of the next motion candidate does not start at the actual n-coordinate of the previous motion candidate, but rather at the final n-coordinate of the input path for the previous motion candidate. It also makes the complete input path for the maneuver to be continuous. The input orientation $\psi_{\mathrm{in},l,h}(s)$ to the LQR is for each motion candidate obtained from computing the first-order derivative of $n_{\mathrm{in},l,h}(s)$ in (7) with respect to s and then combining it with (1) and (2d) under the approximations $v_y \approx 0$ and $n\mathcal{C}(s) \approx 0$, which gives $\psi_{\mathrm{in},l,h}(s) =$

$$\arctan\left(\frac{n_h - n_{s,l}}{s_{\mathrm{div},l} - s_{\mathrm{div},l-1}}\left(30\tilde{s}^2 - 60\tilde{s}^3 + 30\tilde{s}^4\right)\right). \quad (8)$$

To save computational time, motion candidates are not computed for all values of h. A simple collision check is first performed to make sure that the end-point of an input path is in $\mathcal{O}_{\mathrm{free}}$. The top plot in Figure 5 shows an example scenario, where the dotted lines are the input paths n_{in} for the computed motion candidates.

The following steps define the computational procedure for the Computation of Motion Candidates block:

(1) Define $n_{\min}$, $n_{\max}$, $v_{x,\mathrm{in}}$, and $n_{l,u}$, $n_{l,d}$ for the L obstacles.
(2) Compute the grid points n_h in (5).
(3) Compute the LQR gain matrix F for the complete scenario.
(4) From Selection of Motion Candidates, get $h_{\mathrm{sel},l-1}$.
(5) Determine the vector $\bar{h}_{l,\mathrm{comp}}$ with all h fulfilling the collision condition $h \in \mathcal{O}_{\mathrm{free}}$.
(6) For h in $\bar{h}_{l,\mathrm{comp}}$, compute $\bar{x}^i_{\mathrm{in},l,h}$ using $v^i_{x,\mathrm{in},l,h}$, $n_{\mathrm{in},l,h}(s)$, and $\psi_{\mathrm{in},l,h}(s)$.
(7) For h in $\bar{h}_{l,\mathrm{comp}}$, compute motion candidates as sequences of $\{\bar{x}^i_{\mathrm{MC},l,h}\}$ and $\{\bar{u}^i_{\mathrm{MC},l,h}\}$ according to the relations (6).

3.3 Selection of Motion Candidates

In the Selection of Motion Candidates block, the complete computed motion candidates are checked to be in $\mathcal{O}_{\text{free}}$ and not just the end-point of the input paths as in Section 3.2. For each h in $\bar{h}_{l,\text{comp}}$, the n-coordinates in $\{\bar{x}^i_{\text{MC},l,h}\}$ should be in $\mathcal{O}_{\text{free}}$. The check could be extended to also verify actuation utilization, but this is not further pursued here. If multiple motion candidates are feasible, the motion candidate with the lowest change in the n-coordinate between its start and end is selected. If none of the motion candidates is feasible, the algorithm may choose an infeasible motion candidate, and then the algorithm relies on the ability of the SOM planner not only to improve the planned trajectory with the respect to the criterion but also to make it feasible with respect to the constraints.

The steps of the Selection of Motion Candidates block are:

(1) Compute $\bar{h}_{l,\text{feasible}}$ with indices h from $\bar{h}_{l,\text{comp}}$ fulfilling the complete collision check $\{\bar{x}^i_{\text{MC},l,h}\} \in \mathcal{O}_{\text{free}}$.
(2) Set $h_{\text{sel},l}$ according to h from $\bar{h}_{l,\text{feasible}}$ of the motion candidate that has the lowest change $n_h - n_{s,l}$.

3.4 Combination of Motion Candidates

When all motion candidates are computed, the indices of the selected motion candidates $h_{\text{sel},l}$ for all L obstacles are known. The corresponding sequences of states $\{\bar{x}^i_{\text{MC},l,h}\}$ and inputs $\{\bar{u}^i_{\text{MC},l,h}\}$ of $L+1$ motion candidates are combined in the Combination of Motion Candidates block into the initially feasible motion given by

$$\{\bar{x}^i_{\text{IFM}}\},\ i \in \{1,\dots,N+1\} \text{ and } \{\bar{u}^i_{\text{IFM}}\},\ i \in \{1,\dots,N\},$$

where $N = -L + \sum_{l=1}^{L} N_l$ and $N+1$ is the total number of discretization points. The used computational approach ensures that the computed motion is feasible with respect to the combined vehicle model (2).

4. SOM PLANNER

The second motion-planning block, the SOM Planner in Figure 1, is in turn described by five blocks, see Figure 4.

4.1 Scenario Segmentation

The scenario is divided into a number of segments to allow parallel computations at later stages of the planning. Note that this division is done in a different way than in the IFM Planner. The approach used in the SOM Planner builds on the previous research in (Anistratov et al., 2018, 2020), namely a segmentation from a vehicle-dynamics perspective. More specifically, a vector $\bar{s}_{\text{extrema}}$ with s-coordinates of extrema of the vehicle orientation ψ and the yaw rate r is computed in the Scenario Segmentation block. Information about these variables is available from the initially feasible motion, which is an output from the IFM Planner block (see Section 3). Segmentation points are represented as a vector $\bar{s}_{\text{seg}}$, which includes the s-coordinates

$$\bar{s}_{\text{seg}} = (s_0, s_{\text{extrema},j}, \dots, s_f), \tag{9}$$

where the boundary points s_0, s_f and a selection of the extremum points $s_{\text{extrema},j}$, $j \in \bar{j}_{\text{select}}$, are included. Different selection approaches for the number of extremum points are possible. One approach is based on selecting the points in such a way that segments are approximately of the same size. The

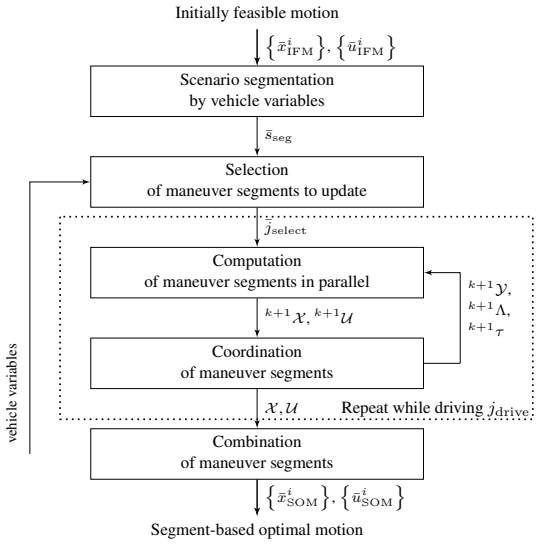

Fig. 4. Layout of the SOM Planner block (see Figure 1 for the overall layout of the motion planner).

length of the vector $\bar{s}_{\text{seg}}$ is $M+1$, where M is the number of segments, and the variable j is used for indexing.

The computations of the Scenario Segmentation block are:

(1) Find $\bar{s}_{\text{extrema}}$ using the initially feasible motion $\{\bar{x}^i_{\text{IFM}}\}$.
(2) Define the segmentation points $\bar{s}_{\text{seg}}$ in (9).
(3) Set M to the number of segments.

4.2 Selection of Maneuver Segments to Update

By using the observation that the maneuver segments are followed sequentially, the segment-based optimal motion is updated in a receding-horizon fashion using the current position of the vehicle in the Selection of Maneuver Segments to Update block and provided to the Controller block. The main idea used here is that before the vehicle reaches a particular maneuver segment, it is possible to update that maneuver segment as well as the following maneuver segments. Such a strategy allows reducing the time before the vehicle is able to start following the maneuver, at the expense of a potentially suboptimal motion at the beginning of the maneuver. While driving through maneuver segment j_{drive}, the maneuver segments selected for update are

$$\bar{j}_{\text{select}} = \{j_{\text{drive}}+1, \dots, M\}. \tag{10}$$

Referring to Figure 4, this implies that the blocks Computation of maneuver segments in parallel and Coordination of maneuver segments are repeated several times before the end of the maneuver segment j_{drive} is reached and $\bar{j}_{\text{select}}$ is updated.

4.3 Computation of Maneuver Segments in Parallel

To provide a basis for the description of the Computation of Maneuver Segments in Parallel block responsible for computing maneuver segments in parallel, the optimal control problem for the complete scenario is first formulated, and then it is reformulated to allow parallel computations of maneuver segments.

Optimal Control Problem To formulate the optimal control problem for the complete scenario, functions penalizing driving

closer than the distance n_m to the road boundaries are formulated for the left boundary n_l and the right boundary n_r as

$p_l^i = \max(n^i - n_l^i + n_m, 0),\ p_r^i = \max(n_r^i - n_i + n_m, 0),$

for $i \in \{1, \dots, N+1\}$. Then the optimal control problem with N piecewise constant control inputs is formulated as

$$\min_{\bar{x},\bar{u}} \quad \sum_{i=1}^{N+1} \Big((c_v(v_x - v_{x,\mathrm{in}})^2 + c_{n,b}((p_l^i)^2 + (p_r^i)^2) + c_\delta(\delta^i)^2 \Big) \Delta s + \sum_{i=1}^{N} (\bar{u}^i)^{\mathrm{T}} R \bar{u}^i \Delta s \tag{11a}$$

$$\text{s.t.} \quad \bar{x}^1 = \bar{x}_0,\ \left|\delta^i\right| \le \delta_{\max},\ \bar{u}_{\min} \le \bar{u}^i \le \bar{u}_{\max}, \tag{11b}$$

$$(F_{x,f}^i)^2 + (\eta F_{y,f}^i)^2 \le (\mu m g l_r/(l_f + l_r))^2, \tag{11c}$$

$$(F_{x,r}^i)^2 + (\eta F_{y,r}^i)^2 \le (\mu m g l_f/(l_f + l_r))^2, \tag{11d}$$

$$n_l^i \le n^i \le n_r^i,\ \bar{x}^{i+1} = \mathcal{F}(\bar{x}^i, \bar{u}^i), \tag{11e}$$

where $i \in \{1, \dots, N\}$, the constant matrix R and the constants c_v, $c_{n,b}$, and c_δ are weighting factors, the absolute value of the steering angle is limited by $\delta_{\max}$, the inputs are constrained by $\bar{u}_{\max}$ and $\bar{u}_{\min}$, and the forces for each tire are bounded by the friction ellipse (Pacejka, 2006), where η is the ellipse parameter.

Separable Control Problem and Parallel Computations To separate the optimal control problem (11) into maneuver segments for parallel computations, the procedure from (Anistratov et al., 2020) is used, which in turn is built on the approach in (Galvan et al., 2019). The problem (11) can compactly be represented with a sum of the terms $\mathcal{J}(\bar{x}^i, \bar{u}^i)$, $i \in \{1, \dots, N+1\}$, for the objective function (11a) and with $\mathcal{G}(\bar{x}^i, \bar{u}^i) \le 0$ and $\bar{x}^{i+1} = \mathcal{F}(\bar{x}^i, \bar{u}^i)$, $i \in \{1, \dots, N\}$, for the constraints (11b)–(11e). This problem is subsequently reformulated to allow splitting it into M subproblems. This is achieved by dividing the state variables $\bar{x}$ and control inputs $\bar{u}$ into the M maneuver segments and by introducing additional equality constraints with auxiliary variables $\bar{y}$ to make the new problem to be equivalent to (11). The new segmented state and control variables are denoted as $\bar{x}_j$ and $\bar{u}_j$, consisting of $p_j + 1$ and p_j vectors, respectively, representing p_j steps in s. For compact notation, the compositions of the new segmented vectors and the composition of multipliers, to be used in the subsequently defined augmented Lagrangian, are introduced as

$$\mathcal{X} = \{\bar{x}_1, \dots, \bar{x}_M\},\ \mathcal{U} = \{\bar{u}_1, \dots, \bar{u}_M\}, \tag{12a}$$

$$\mathcal{Y} = \{\bar{y}_1, \dots, \bar{y}_{M+1}\},\ \Lambda = \{\lambda_1^0, \lambda_1^f, \dots, \lambda_M^0, \lambda_M^f\}. \tag{12b}$$

The augmented Lagrangian corresponding to the j:th of the M subproblems is given by

$$\mathcal{L}_{j,\tau}(\bar{x}_j, \bar{u}_j, \bar{y}_j, \bar{y}_{j+1}, \lambda_j^0, \lambda_j^f) = \lambda_j^0 \left(\bar{x}_j^1 - \bar{y}_j\right) + \frac{\tau}{2} \left\|\bar{x}_j^1 - \bar{y}_j\right\|^2 + f_j^0 \mathcal{J}\left(\bar{x}_j^1, \bar{u}_j^1\right) + \sum_{i=2}^{p_j} \mathcal{J}\left(\bar{x}_j^i, \bar{u}_j^i\right) + f_j^f \mathcal{J}\left(\bar{x}_j^{p_j+1}, \bar{u}_j^{p_j+1}\right) + \lambda_j^f \left(\bar{x}_j^{p_j+1} - \bar{y}_{j+1}\right) + \frac{\tau}{2} \left\|\bar{x}_j^{p_j+1} - y_{j+1}\right\|^2, \tag{13}$$

where τ is a penalty parameter and the factors f_j^0 and f_j^f are introduced to consider that the state values $\bar{x}_j^{p_j+1}$ and $\bar{x}_{j+1}^1$ correspond to the same traveled distance s along the center of the road, i.e., $f_1^0 = f_M^f = 1$ and 0.5 in all other cases. More details about the augmented Lagrangian formulation can be found in (Anistratov et al., 2020).

Given the current iterate $\left({}^k\mathcal{X},\ {}^k\mathcal{U},\ {}^k\mathcal{Y}, {}^k\Lambda, {}^k\tau\right)$, the computational steps are the following. For fixed ${}^k\mathcal{Y}$, ${}^k\Lambda$, and ${}^k\tau$, the values of ${}^{k+1}\mathcal{X}$ and ${}^{k+1}\mathcal{U}$ are obtained by finding their components ${}^{k+1}\bar{x}_j$, ${}^{k+1}\bar{u}_j$ from solving subproblems for each $j \in \bar{j}_{\mathrm{select}}$, with $i \in \{1, \dots, p_j\}$, as

$$\min_{\bar{x}_j, \bar{u}_j} \quad \mathcal{L}_{j,\tau}(\bar{x}_j, \bar{u}_j, {}^k\bar{y}_j, {}^k\bar{y}_{j+1}, {}^k\lambda_j^0, {}^k\lambda_j^f) \tag{14a}$$

$$\text{s.t.} \quad \mathcal{G}(\bar{x}_j^i, \bar{u}_j^i) \le 0,\ \bar{x}_j^{i+1} = \mathcal{F}(\bar{x}_j^i, \bar{u}_j^i). \tag{14b}$$

These subproblems are independent of each other and can thus be solved in parallel on different computational units.

The steps of the Computation of Maneuver Segments in Parallel block are:

(1) Setup M subproblems of (14), set ${}^k\mathcal{X}$, ${}^k\mathcal{U}$, and ${}^k\mathcal{Y}$ using the initially feasible motion $\left\{\bar{x}_{\mathrm{IFM}}^i\right\}$ and $\left\{\bar{u}_{\mathrm{IFM}}^i\right\}$.
(2) Get ${}^{k+1}\mathcal{X}$ and ${}^{k+1}\mathcal{U}$ by solving (14) for $\bar{j}_{\mathrm{select}}$.

4.4 Coordination of Maneuver Segments

For parallel computations of the maneuver segments, these segments are coordinated in the Coordination of Maneuver Segments block using the variables $\mathcal{Y}$, Λ, and τ. For fixed ${}^{k+1}\mathcal{X}$, ${}^{k+1}\mathcal{U}$, ${}^k\Lambda$, and ${}^k\tau$, the components of the new iterate ${}^{k+1}\mathcal{Y}$ are obtained by analytic expressions (derived in Anistratov et al. (2020))

$${}^{k+1}y_1 = \frac{{}^k\lambda_1^0}{{}^k\tau} + {}^{k+1}x_1^0,\ {}^{k+1}y_{M+1} = \frac{{}^k\lambda_M^f}{{}^k\tau} + {}^{k+1}x_M^f, \tag{15a}$$

$${}^{k+1}y_j = \frac{{}^k\lambda_{j-1}^f + {}^k\lambda_j{}^0}{2 \cdot {}^k\tau} + \frac{{}^{k+1}x_{j-1}^{p_j+1} + {}^{k+1}x_j^0}{2}, \tag{15b}$$

where $j \in \{1, \dots, M\}$. However, in this paper, the auxiliary variables $\bar{y}_j$ for the first maneuver segment j possible to update are set equal to the state values at the end of the previous maneuver segment $\bar{x}_{j-1}^{p_j+1}$. New multipliers ${}^{k+1}\Lambda$ and the penalty parameter τ are updated as in (Galvan et al., 2019; Anistratov et al., 2020).

The steps of the Coordination of Maneuver Segments block are:

(1) Compute ${}^{k+1}\mathcal{Y}$ in (15).
(2) Compute ${}^{k+1}\Lambda$ and ${}^{k+1}\tau$.

4.5 Combination of Maneuver Segments

When the parallel computations and the coordination processes are completed (e.g., the vehicle reaches the end of the maneuver segment j_{drive}), the segment vectors in $\mathcal{X}$ and $\mathcal{U}$ are combined in the Combination of Maneuver Segments block into the segment-based optimal motion given by

$\left\{\bar{x}_{\mathrm{SOM}}^i\right\},\ i \in \{1, \dots, N+1\}$ and $\left\{\bar{u}_{\mathrm{SOM}}^i\right\},\ i \in \{1, \dots, N\}$, which is passed to the Controller block in Figure 1.

5. PARAMETERS AND IMPLEMENTATION

In this section, the parameters used to define the model and the optimal control problems are defined. Moreover, implementation aspects are discussed.

5.1 Obstacles and Road Boundaries

Two consecutive double lane-changes are considered as the scenario, where the dimensions for one double lane-change are inspired by the ISO 3888-2 double lane-change test. One

double lane-change setup is defined using three obstacles, and for the second double lane-change setup, these three obstacles are repeated, but shifted along the s-axis. The top subplot in Figure 5 visualizes the setup, and the corresponding lateral road limits along the $\mathbf{n}$-axis in Figure 6 are

$$n_l(s) = \begin{cases} 0.7, & s \in [0, 12] \cup [49, 73] \cup [110, 122] \\ 3.5, & \text{otherwise} \end{cases} \tag{17a}$$

$$n_r(s) = \begin{cases} 2.5, & s \in [25.5, 36.5] \cup [76.5, 97.5] \\ -0.7, & \text{otherwise} \end{cases} \tag{17b}$$

5.2 Model and Problem Parameters

The vehicle model and tire parameters are chosen as in (Anistratov et al., 2020), and they are shown in Table 1. For the IFM Planner, $\Delta n = 0.35$ m and $\Delta s = 0.25$ m. The weighting parameters for the performance criterion of the LQR are

$$\breve{Q} = \text{diag}(100, 0, 100, 0, 1000, 10, 10, 0), \tag{18a}$$

$$\breve{R} = \text{diag}(20, 20, 20), \tag{18b}$$

where $\text{diag}(\cdot)$ creates a square diagonal matrix with the elements of the provided vector along the main diagonal.

For the SOM Planner, the weighting parameters in the objective function (11a) are set to $c_v = 0.05$, $c_{n,b} = 10$, $n_m = 0.5$ m, $c_\delta = 1$, and $R = \text{diag}(0.1, 0.1, 0.01)$. The number of piecewise constant control inputs N is set such that the step size is $\Delta s = 0.25$ m. The vehicle initial state is $x_0 = (60/3.6, 0, 0, 0, 0, 0, 0, 0)$, and the corresponding target longitudinal velocity is $v_{x,\text{in}} = 60/3.6$ m/s. For the vehicle steering angle, the limit is $\delta_{\max} = \pi/3$, and the control-input limits are $\bar{u}_{\max} = -\bar{u}_{\min} = (2.0, 2.0, 5.0)$. The parameters of the update rule for τ are chosen as ${}^1\tau = 2.5$, $\sigma = 0.95$, and $\alpha = 1.02$.

5.3 Comments on Implementation

The IFM and SOM Planners were implemented using the Python 3.8 programming language. The optimization problems 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, 2022). The longitudinal tire forces $F_{x,f}$ and $F_{x,r}$ and the inputs $u_{F_{x,f}}$ and $u_{F_{x,r}}$ are scaled by a factor $1/1000$ for improved numerical performance in the solver. When showing the final solution, the inverse scaling is applied on the affected variables. The presented performance results are computed under the assumption that the segmented problems are computed in parallel threads with access to a shared memory. It means that the time for each iteration is the maximum computational time over all maneuver segments. To decrease the computational time, 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 120. 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 warm-start related parameters of the solver IPOPT are set to 10^{-9}.

Table 1. Vehicle parameters for the model.

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	[-]
g	9.82	ms^{-2}			

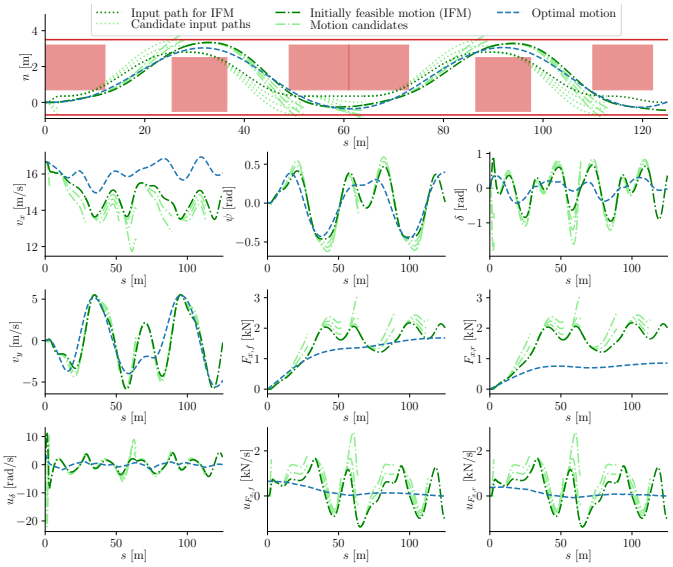

Fig. 5. Motion candidates computed by the IFM Planner. Obstacles and road boundaries are shown with red rectangles and lines, respectively.

6. RESULTS

The motion-planning approach is evaluated to demonstrate its performance. First, the results from the IFM Planner are shown in a straight-road scenario. Second, the complete planner is evaluated. For comparison, optimal-motion results without segmentation are computed as well.

6.1 Motion Candidates and Initially Feasible Motion

For a straight-road scenario ($\mathcal{C}(s) = 0$), where the number of obstacles is $L = 6$, the computed motion candidates in the IFM Planner are illustrated in Figure 5 with the dash-dotted lines. In the top subplot, it is visualized how several motion candidates are computed to reach each obstacle, and the corresponding number of input paths are shown with the dotted lines in the figure as well. The input paths are computed using $n_{\text{in},l,h}(s)$ in (7) for $l \in \{1, \ldots, L+1\}$ and the values of h that pass the simple collision check (Section 3.2). This collision check ensures that only motion candidates for input paths with obstacle-free start and end points are computed (see the position at $1/4$ of the distance in s for each obstacle), thus saving computational time.

The input paths determined using the polynomial (7) aim to guide in the computation of an initially feasible motion, and are thus not necessarily to be followed as close as possible. This is why deviations between an input path and a motion candidate in Figure 5 are not of particular importance here. A subset of the computed motion candidates are leaving the road boundaries. However, the algorithm does not select them for the initially feasible motion, since they are ruled out when running the complete collision check (Section 3.3). The other subplots in Figure 5 show selected vehicle variables and the inputs.

The selected motion candidates (based on $h_{\text{sel},l}$ in Section 3.3) give the initially feasible motion, and they are shown with the dash-dotted dark green lines in Figure 5. The resulting initially feasible motion has continuous inputs, see u_δ, $u_{F_{x,f}}$, and $u_{F_{x,r}}$,

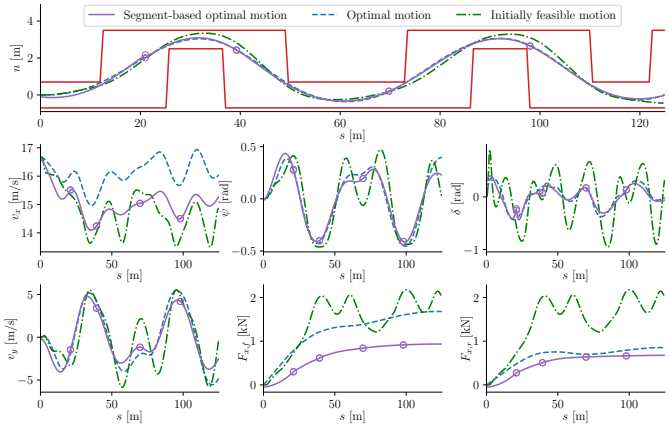

Fig. 6. Solution comparison between the segment-based optimal motion, the optimal motion, and the initially feasible motion.

which is achieved by construction in the computational approach in the IFM Planner (see Section 3). The input trajectory used to drive the simulation of the motion candidates is not feasible *per se*. However, since the motion candidates are computed using the same model (2) as is used to compute the segment-based optimal motion in the SOM Planner, the initially feasible motion is feasible with respect to that model. This facilitates computations in the subsequent SOM Planner. As a result of the yaw dynamics, the vehicle shows decreases in the longitudinal velocity v_x even though the tire longitudinal forces $F_{x,f}$ and $F_{x,r}$ are always positive in the example considered.

Figure 5 also compares the initially feasible motion with the optimal motion, which is computed by solving the optimization problem (11) for the complete scenario. The solutions have some important similar characteristics such as position of extremum points, see, e.g., the lateral position n, the lateral velocity v_y, and the vehicle orientation ψ. Since the vehicle orientation ψ is used to find segmentation points for the SOM Planner, such similarities are important, since it allows finding segmentation points using the initially feasible motion trajectory close to the *a priori* unknown segmentation points of the optimal trajectory. The optimal motion uses noticeably less actuation in u_δ, $u_{F_{x,r}}$, and $u_{F_{x,f}}$, compared to the output from the IFM Planner. Because of differences in the excitation of the yaw dynamics, visible, e.g., in the differences in ψ for $s = 60$–90 m, the longitudinal forces $F_{x,f}$ and $F_{x,r}$ are lower for the optimal motion even though the longitudinal velocity v_x decreases less.

6.2 SOM Planner

The segment-based optimal motion, i.e., the output from the SOM Planner, is compared with the optimal motion in Figure 6 for the straight-road scenario. Initial values for the state and control variables are set using the initially feasible motion, i.e., the output from the IFM Planner. The scenario is segmented into five segments ($M = 5$), where the segmentation points are found as extrema of the vehicle orientation ψ, while omitting points that are too close to other points; in Figure 6, the points at $s = 57.75$ m and $s = 82$ m are omitted.

The final solution for the segment-based optimal motion is shown in Figure 6, where maneuver segments that are not yet reached by the vehicle are continued to be updated as the vehicle progresses along the road. It means that the first maneuver segment (starting at $s = 0$ m) is initialized and then updated only three times before the vehicle starts to follow that maneuver segment, while the remaining maneuver segments are updated three times in parallel during the same time. This gives a computational time of about 0.46 s. While the vehicle is following the first maneuver segment (about 1.3 s), maneuver segments two to five are updated in parallel (ten times in total in the example scenario considered). The same strategy is applied until the vehicle reaches the final maneuver segment. Figure 6 illustrates a substantial improvement of the initially feasible motion, and is similar to the optimal motion in many respects. Compare, e.g., the vehicle lateral position n, orientation ψ, and lateral velocity v_y for the respective case. This concerns in particular the motion at the later stages of the maneuver. This shows that the developed planning approach is performing well for the considered scenario.

The usage of the steering angle δ is drastically reduced for the segment-based optimal motion compared to the initially feasible motion in Figure 6. The longitudinal forces $F_{x,f}$ and $F_{x,r}$ of the segment-based optimal motion have a similar trend compared to the optimal motion, and the longitudinal velocity v_x shows a similar variation for both solutions, even though the values are not matching completely. A possible explanation of this difference is a low parameter weight on the velocity deviations c_v relative to the other parameters in the objective function (11a).

In this example scenario, no additional segments are added at the end. The approach could, however, be extended such that new segments are included to cover an extended distance beyond the current final target. The maneuver segments are recomputed with a period of approximately 0.13 s. Here, stationary obstacles are considered. It would also be possible to take updated obstacle information into account, before the vehicle reaches the corresponding maneuver segment and commits to follow a particular motion plan provided by the SOM Planner.

A curved-road scenario was also used for evaluating the motion planner. The road started straight and then turned left or right, with a curvature of ± 0.01 m^{-1} after 10 m, i.e.,

$$\mathcal{C}(s) = \begin{cases} 0, & \text{if } s < 10 \\ \pm 0.01, & \text{if } s \geq 10 \end{cases} \tag{19}$$

The IFM Planner is aware of the road curvature and provides an initially feasible motion for this scenario as well, even though the input orientation ψ_{in} in (8) is computed under the assumption of zero curvature (see Section 3.2). The SOM Planner computes the segment-based optimal motion, which, as for the straight-road scenario, improves the initially feasible motion and computes a motion plan close to the optimal motion. Additional analysis of this scenario is provided in the thesis (Anistratov, 2021).

7. CONCLUSIONS

Avoidance maneuvers at normal driving speed or higher are demanding driving situations that force the vehicle to the limit of tire–road friction in critical situations. A complete dynamics-based two-step algorithm has been developed to handle long scenarios involving multiple lane changes, and the main ideas are captured in Figures 1, 3, and 4. The first main step, the IFM Planner (see Figure 3), finds an initially feasible motion. The first step within this planner is scenario division based on obstacle locations, which is followed by the major steps of computing, selecting, and combining dynamics-based motion candidates. The second main step is the SOM Planner (see Figure 4). In that

part, a scenario segmentation is first performed using a vehicle-dynamics perspective, i.e., using vehicle variables such as the vehicle orientation and yaw rate, to find favorable segmentation points. Optimization is then performed for each of the segments individually, under coordination using the alternating augmented Lagrangian method. This combines the advantages of IPOPT in terms of its efficient implementation and the convergence of the interior-point method together with the alternating augmented Lagrangian method, allowing parallel computation of maneuver segments. The segmented optimization approach lends itself well to computing the segments in a receding-horizon fashion, where the motion plans for not yet reached parts are kept being updated as long as the computational time allows it.

The complete algorithm was applied to a scenario with two consecutive double lane-changes, and this was done for driving speeds higher than where the kinematic-model assumption is valid. It was shown that the IFM Planner was able to compute dynamics-based motion candidates, and also to effectively find an initially feasible motion. It was then shown that the SOM Planner gave significant improvement of the motion by its use of segmented optimization. Overall, the results show that the complete algorithm is successful in finding a motion that handles challenging scenarios at high speed.

REFERENCES

Ajanovic, Z., Regolin, E., Stettinger, G., Horn, M., and Ferrara, A. (2020). Search-based motion planning for performance autonomous driving. In *Advances in Dynamics of Vehicles on Roads and Tracks*, 1144–1154. Springer International Publishing, Cham, Germany.

Anderson, B. and Moore, J. (1989). *Optimal control: Linear quadratic methods*. Prentice-Hall International, Inc, Englewood Cliffs, NJ, USA.

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. (2021). *Autonomous Avoidance Maneuvers for Vehicles using Optimization*. Ph.D. thesis, Department of Electrical Engineering, Linköping University. Dissertation No. 2162.

Anistratov, P., Olofsson, B., Burdakov, O., and Nielsen, L. (2020). Autonomous-vehicle maneuver planning using segmentation and the alternating augmented Lagrangian method. *IFAC-PapersOnLine*, 53(2), 15558–15565.

Anistratov, P., Olofsson, B., and Nielsen, L. (2018). 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.

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.

Bergman, K., Ljungqvist, O., and Axehill, D. (2019). Improved optimization of motion primitives for motion planning in state lattices. In *IEEE Intelligent Vehicles Symposium (IV)*, 2307–2314. Paris, France.

Bergman, K., Ljungqvist, O., and Axehill, D. (2021). Improved path planning by tightly combining lattice-based path planning and optimal control. *IEEE Transactions on Intelligent Vehicles*, 6(1), 57–66.

Bergman, K., Ljungqvist, O., Glad, T., and Axehill, D. (2020a). An optimization-based receding horizon trajectory planning algorithm. *IFAC-PapersOnLine*, 53(2), 15550–15557.

Bergman, K., Ljungqvist, O., Linder, J., and Axehill, D. (2020b). An optimization-based motion planner for autonomous maneuvering of marine vessels in complex environments. In *59th IEEE Conference on Decision and Control (CDC)*, 5283–5290. Jeju Island, South Korea.

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.

Diehl, M., Bock, H.G., and Schlöder, J.P. (2005). A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM Journal on Control and Optimization*, 43(5), 1714–1736.

Frazzoli, E., Dahleh, M.A., and Feron, E. (2002). Real-time motion planning for agile autonomous vehicles. *Journal of Guidance, Control, and Dynamics*, 25(1), 116–129.

Galvan, G., Lapucci, M., Levato, T., and Sciandrone, M. (2019). An alternating augmented Lagrangian method for constrained nonconvex optimization. *Optimization Methods and Software*, 35(3), 502–520.

Gray, A., Gao, Y., Lin, T., Hedrick, J.K., Tseng, H.E., and Borrelli, F. (2012). Predictive control for agile semi-autonomous ground vehicles using motion primitives. In *American Control Conference (ACC)*. Montreal, QC, Canada.

He, X., Liu, Y., Lv, C., Ji, X., and Liu, Y. (2018). Emergency steering control of autonomous vehicle for collision avoidance and stabilisation. *Vehicle System Dynamics*, 57(8), 1163–1187.

Howard, T.M., Green, C.J., Kelly, A., and Ferguson, D. (2008). State space sampling of feasible motions for high-performance mobile robot navigation in complex environments. *Journal of Field Robotics*, 25(6-7), 325–345.

HSL (2022). A collection of Fortran codes for large scale scientific computation. `http://www.hsl.rl.ac.uk`. (Date accessed: 18.02.2022).

ISO 3888-2:2011 (2011). Passenger cars — Test track for a severe lane-change manoeuvre — Part 2: Obstacle avoidance. Technical report, International Organization for Standardization (ISO). Geneva, Switzerland.

LaValle, S.M. (2006). *Planning Algorithms*. Cambridge University Press, New York, NY, USA, first edition.

Likhachev, M. and Ferguson, D. (2009). Planning long dynamically feasible maneuvers for autonomous vehicles. *The International Journal of Robotics Research*, 28(8), 933–945.

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

Ljungqvist, O., Bergman, K., and Axehill, D. (2020). Optimization-based motion planning for multi-steered articulated vehicles. *IFAC-PapersOnLine*, 53(2), 15580–15587.

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.

Pivtoraiko, M. and Kelly, A. (2011). Kinodynamic motion planning with state lattice motion primitives. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2172–2179. San Francisco, CA, USA.

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.

Svensson, L., Bujarbaruah, M., Kapania, N.R., and Törngren, M. (2019). Adaptive trajectory planning and optimization at limits of handling. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3942–3948. Macau, China.

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.