Comparison between mixed-integer and second order cone programming for autonomous overtaking

Johan Karlsson, Nikolce Murgovski and Jonas Sjöberg

***Abstract*— This paper concerns optimally controlling an autonomous vehicle to perform safe and comfortable overtaking of a slower moving leading vehicle. The contribution is an analysis of the comparisons between a convex relaxation with the standard mixed integer quadratic program. The main difference between the formulations is that the sampling is performed in the temporal domain in the standard formulation, but in the spatial domain for the convex relaxation. The case of varying lateral position and longitudinal velocity is studied for both algorithms and the solution quality and computational effort are discussed. The results are given in a case study where an ego vehicle is forced to accelerate when overtaking a leading vehicle, due to the presence of an oncoming vehicle. The results illustrate that the temporal and spatial formulation's yield similar solutions but obtaining them require less computational effort in the spatial domain.**

I. Introduction

Aims such as decreasing the number of traffic accidents and traffic congestions motivate the introduction of partially or fully autonomous vehicles [1], [2]. In today's market, many major car companies work on partially or fully autonomous vehicles [3], in which automated systems, such as adaptive cruise control [1] and automatic parking [4] are in standard production. Fully autonomous driving, however, introduces challenges. One such challenge, which has drawn much attention recently, is the introduction of autonomous overtaking maneuvers, where the vehicle needs to perform a safe overtaking of a slow moving leading vehicle.

The vehicle overtaking procedure can crudely be divided into four steps: 1) detection of all surrounding obstacles and vehicles, (including the leading vehicle) using sensors (such as light detection and ranging sensors); 2) deciding whether overtaking should be performed; 3) trajectory planning, if overtaking should be performed; 4) trajectory tracking. Of course, this structure is not unique to the overtaking case but a standard way of developing a variety of autonomous features, see for example [5].

This paper studies the third step, trajectory generation in the case of autonomous overtaking, with the goal of obtaining position and velocity trajectories which the vehicle can follow in a way that is *i)* safe, *ii)* comfortable, and, *iii)* deviates as little as possible from the preferred longitudinal speed and lateral position. This should be done using an algorithm that, *a)* does not rely heavily on heuristics, and, *b)* should be able to react to unforeseen events, e.g., a vehicle suddenly appearing in the adjacent lane due to late detection or sensor failure. There exist several methods for solving the overtaking problem described above. For example, in [6], the overtaking is made using a so-called shadow vehicle and the Rendezvous-guidance technique, while in [7] by approximating lane change maneuvers using fifth order minimal jerk trajectories. Other approaches include grid/graph based search [8], [9], but these methods rely on the availability of good heuristics, i.e., does not fulfill requirement *a*.

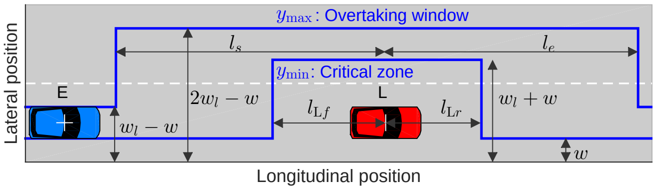

Fig. 1. Scenario where the ego vehicle (E) is overtaking a leading vehicle (L) on a road with two lanes. The CoG of the ego vehicle is allowed to reside between the limits $y_{\min}$ and $y_{\max}$, depicted by the thick solid lines.

Considering the goals *i-iii* one approach is to solve the problem using optimal control. Optimal control involves modelling the overtaking problem as an optimization problem, where constraints are used to enforce goal *i* and the cost function enforces goals *ii* and *iii*. However, optimal control has the disadvantage that it cannot respond to surprises, i.e., does not fulfill requirement *b*, since it is solved only once. One way of fulfilling requirement *b* is to solve the optimal control problem iteratively. This can be done using model predictive control (MPC), meaning that the optimal control problem is used to obtain trajectories for a finite time horizon, which are then tracked by the vehicle for a few time steps before the optimal control problem is remodelled (taking any changes in the environment and traffic into account) and re-solved for the shifted finite time horizon.

The straightforward way of formulating the optimal control problem is in the time domain using both continuous and binary variables to model the goals *i-iii*, i.e., a non-convex mixed integer program (MIP), [10]. The computation time of MIPs is highly dependent on choosing a feasible starting point, which might be difficult to find in complex traffic situations. A way around this issue is to recast the problem as a convex program, since convex programs can be solved efficiently using a variety of solvers, e.g., CPLEX, GUROBI and MOSEK, to name a few.

However, most attempts of introducing convex formulations of the optimal overtaking problem come with drawbacks. For example, in [11], two different convex formulations were considered. For both formulations, the drawbacks stems from the fact that the overtaking needs to be safe, i.e.,

This work was partially supported by the Wallenberg Autonomous Systems and Software Program (WASP) and partially by the European Commission Seventh Framework Program under the project AdaptIVe, grant agreement number 610428.

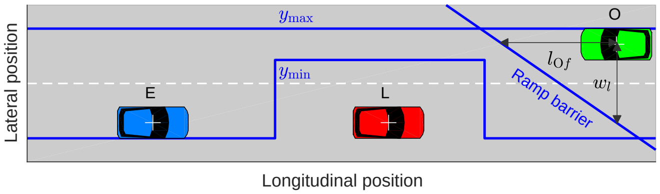

Fig. 2. Illustration of an overtaking scenario with an oncoming vehicle (O). During the overtaking maneuver, the vehicle's CoG is allowed to reside between the limits $y_{\min}$, $y_{\max}$ and the ramp barrier of the oncoming vehicle.

fulfill goal i. In one case, the safety constraints include slack variables whose weights needs to be chosen carefully in the objective function for the program to yield good solutions. To avoid this tuning problem a second approach is tried in which restrictions are implemented on the slack variables. However, this leads to the drawback that a full overtaking trajectory cannot be calculated in one MPC cycle. Since, both these issues stem from the need for slack variables a formulation where these are not needed was introduced in [12], [13] by sampling in the relative longitudinal distance instead of time.

Sampling in the spatial domain is not a new idea. It has been used in various applications, such as minimizing the lap time for a race car, see, e.g., [14] and references therein. As in [14] sampling is usually done in absolute space instead of relative space as is suggested in [12], [13]. In the setting of overtaking the main advantage of the relative spatial formulation is that it is easy to convexify the overtaking problem in the presence of a leading vehicle while still being able to calculate the full trajectory. In this way, all the requirements and goals stated are achieved. For the remainder of the paper the relative spatial domain will be referred to simply as the spatial domain.

The focus of this paper is to compare the optimal solution and the computational effort of the spatial and the traditional temporal formulation. The contribution is two-fold. Building on the work presented in [12], [13] a new version of the spatial formulation is introduced where kinetic energy is used instead of speed, which, as we will see, removes non-convexity when bounding the longitudinal acceleration. Secondly, this paper provides a comparison of solution times between the integer formulation and the convex formulation. This is important since fast solutions are necessary to be able to respond to sudden changes in the traffic environment.

The paper is organised as follows. In Section II the standard temporal formulation of the overtaking problem is formulated as a mixed integer program (MIP), which in Section III is transformed into a non-linear continuous program (NLP) by sampling in the relative space domain. In Section IV the objective functions are introduced. In Section V the NLP is transformed into a second order cone problem (SOCP), [15], and an algorithm is presented to solve the problem sequentially. Finally, Section VI and VII contain the results and conclusions of the paper.

II. Problem formulation in the temporal domain

Letting the autonomous vehicle be controlled using MPC, the focus is to study the optimal control problem solved during one iteration of the MPC. A scenario is considered where the ego vehicle (E) is travelling on a two-lane road, with the goal of controlling speed and steering, to overtake a leading vehicle (L) travelling with constant speed v_{L} and constant lateral position y_{L}, in the presence of an oncoming vehicle (O) with constant longitudinal velocity v_{O} and constant lateral position y_{O}. The assumption of constant velocity is made here to simplify the math, but it is possible to derive similar algorithms for the case of non-constant, but known, longitudinal velocity of the surrounding vehicles. In this section, an MIP formulation of the optimal control problem is introduced, which generates smooth and safe trajectories for the ego vehicle to track when overtaking.

A. Vehicle dynamics

Let

$$\mathbf{x}_{\mathrm{E}}(t) = [x_{\mathrm{E}}(t), v_{\mathrm{Ex}}(t), y_{\mathrm{E}}(t)]^T \qquad \mathbf{u}_{\mathrm{E}}(t) = [a_{\mathrm{Ex}}(t), v_{\mathrm{Ey}}(t)]^T,$$

be the state and control vectors of the ego vehicle, respectively. Here, x_{E} and $v_{\mathrm{E}x}$ denote the longitudinal position and velocity of the ego vehicle, respectively, y_{E} and $v_{\mathrm{E}y}$ the lateral position and velocity, respectively, and $a_{\mathrm{E}x}$ the longitudinal acceleration. By modelling the vehicles as point mass systems, the ego vehicle can be described by the state space model

$$\dot{\mathbf{x}}_{\mathrm{E}}(t) = A\mathbf{x}_{\mathrm{E}}(t) + B\mathbf{u}_{\mathrm{E}}(t)$$

with

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{1}$$

B. Safety constraints

To avoid colliding with other vehicles and surrounding obstacles, safety constraints need to be introduced. The constraint for not colliding with the leading vehicle takes the form of a rectangular zone, called the critical zone (CZ), see Fig. 1. Additionally, constraints are added to prevent the ego vehicle from overtaking too early (and too late), i.e., an overtaking window (OW) is formulated, in which overtaking of the leading vehicle is allowed, see Fig. 1. There are multiple ways of modeling these zones using a mixed integer approach (see for instance [10]). Here, the following binary variables are introduced for modelling the CZ and OW

Variables:	Description:
$c_1^{\mathrm{cz}}(t) \in \{0,1\}$	1 if E's position is farther than $x_{\min}^{\mathrm{cz}}(t)$
$c_2^{\mathrm{cz}}(t) \in \{0,1\}$	1 if E's position is nearer than $x_{\max}^{\mathrm{cz}}(t)$
$c_1^{\mathrm{ow}}(t) \in \{0,1\}$	1 if E's position is farther than $x_{\min}^{\mathrm{ow}}(t)$
$c_2^{\mathrm{ow}}(t) \in \{0,1\}$	1 if E's position is nearer than $x_{\max}^{\mathrm{ow}}(t)$

where the constants $x_{\min}^{\mathrm{cz}}(t) = x_{\mathrm{L}}(t) - l_{\mathrm{Lf}}$, $x_{\max}^{\mathrm{cz}}(t) = x_{\mathrm{L}}(t) + l_{\mathrm{Lr}}$, $x_{\min}^{\mathrm{ow}}(t) = x_{\mathrm{L}}(t) - l_{\mathrm{s}}$ and $x_{\max}^{\mathrm{ow}}(t) = x_{\mathrm{L}}(t) + l_{\mathrm{e}}$ represent the starting and ending of the CZ and OW. Further, l_{Lf}, l_{Lr}, l_{s} and l_{e} denote the corresponding lengths to the center of gravity (CoG) of the leading vehicle, see Fig. 1, and x_{L} is the longitudinal position of the leading vehicle. This means that if $c_1^{\mathrm{cz}}(t)$ and $c_2^{\mathrm{cz}}(t)$ are both equal to 1 for a timestep, the ego vehicle is within the longitudinal safety margin and if $c_1^{\text{ow}}(t)$ and $c_2^{\text{ow}}(t)$ are both 1 then the ego vehicle is within overtaking range. Thus, for a safe overtaking, the CZ and OW pair of binary variables need to belong to the set

$$\begin{aligned}\mathcal{B}_t^i(x_{\text{E}}(t)) = \{(c_1^i(t), c_2^i(t)) \in \{0,1\} \times \{0,1\} :\\ x_{\text{E}}(t) - x_{\text{min}}^i(t) \leq c_1^i(t)M,\\ x_{\text{E}}(t) - x_{\text{min}}^i(t) \geq (c_1^i(t) - 1)M\\ x_{\text{max}}^i(t) - x_{\text{E}}(t) \leq c_2^i(t)M\\ x_{\text{max}}^i(t) - x_{\text{E}}(t) \geq (c_2^i(t) - 1)M\\ c_1^i(t) + c_2^i(t) \geq 1\},\end{aligned}$$

where $i = \{\text{cz, ow}\}$ and M is a large positive real number. The first row in the set ensures the variables are binary, the second to fourth rows enforces the pair of binary variables to both be 1 when the ego vehicle is within the CZ or OW respectively, while the last row forbids the ego vehicle to be on both sides of a zone at once. The last constraint is included here to reduce the search space, which often reduces the computation time of MIPs [16]. Using the binary variables, the lateral limits are

$$\begin{aligned}y_{\text{E}}(t) &\geq w + (c_1^{\text{cz}}(t) + c_2^{\text{cz}}(t) - 1)w_l,\\ y_{\text{E}}(t) &\leq w_l - w + (c_1^{\text{ow}}(t) + c_2^{\text{ow}}(t) - 1)w_l,\end{aligned}$$

where w_l is the lane width and w is the lateral safety margin. Here, the binary variables are used as a means of deciding when the switch of the lateral limits occur, while the set $\mathcal{B}_t^i(x_{\text{E}}(t))$ ensures these switches occur at the correct times. Linearly modelling a sudden change in constraint parameters this way is common in an MIP framework, [17].

Further, a safety constraint for the oncoming vehicle is needed. Instead of modelling this as a rectangular zone, the constraint is modeled as a ramp barrier, see Fig. 2, which is only active when the ego vehicle is present in the OW. The reason for choosing the ramp barrier over a rectangular zone is that it requires less constraints and does not involve any additional binary variables. The ramp barrier is expressed as

$$\begin{gathered}\frac{x_{\text{E}}(t) - (x_{\text{O0}}(t) - v_{\text{O}}t)}{l_{\text{Of}}} + \frac{y_{\text{E}}(t) - y_{\text{O}}}{w_l} +\\ (c_1^{\text{ow}}(t) + c_2^{\text{ow}}(t) - 2)M \leq -1\end{gathered}$$

where x_{O0} is the initial longitudinal position of the oncoming vehicle and $l_{\text{O}f}$ is a longitudinal length computed as a function of the difference between the mean reference speed of the ego and the speed of the oncoming , [13].

C. Vehicle limitations

The following constraints describe actuator and state limits

$$\begin{aligned}&\mathbf{x}_{\text{E}}(t) \in [\mathbf{x}_{\text{min}}(\cdot), \mathbf{x}_{\text{max}}(\cdot)],\\ &a_{\text{Ex}}(t) \in [a_{x\text{min}}(t), a_{x\text{max}}(t)],\\ &v_{\text{Ey}}(t) \in [s_{\text{min}}, s_{\text{max}}]v_{\text{Ex}}(t),\end{aligned}$$

where the last constraint requires a non-zero longitudinal motion in order to perform a lateral motion. This constraint is modeled via the slip angle β, through $s_{\text{min}} = -\arctan(\beta)$ and $s_{\text{max}} = \arctan(\beta)$, [11]. The state limits are given by

$$\begin{aligned}\mathbf{x}_{\text{min}}(\cdot) &= [0, v_{\text{L}} + \epsilon, \text{free}]^T,\\ \mathbf{x}_{\text{max}}(\cdot) &= [\text{free}, v_{\text{xmax}}, \text{free}]^T,\end{aligned}$$

where ϵ is a small real positive number guaranteeing that the ego vehicle is travelling faster than the leading vehicle, which is necessary to make the overtaking maneuver possible.

D. Full problem

Thus, if an objective function $J(\mathbf{x}_{\text{E}}(t), \mathbf{u}_{\text{E}}(t), \dot{\mathbf{u}}_{\text{E}}(t))$, to be detailed in Section IV, is attached, the complete MIP problem reads

$$\underset{\mathbf{u}_{\text{E}}(t)}{\text{minimize}}\ J(\mathbf{x}_{\text{E}}(t), \mathbf{u}_{\text{E}}(t), \dot{\mathbf{u}}_{\text{E}}(t)) \tag{2a}$$

subject to

$$\dot{\mathbf{x}}_{\text{E}}(t) = A\mathbf{x}_{\text{E}}(t) + B\mathbf{u}_{\text{E}}(t) \tag{2b}$$

$$\mathbf{x}_{\text{E}}(t) \in [\mathbf{x}_{\text{min}}(\cdot), \mathbf{x}_{\text{max}}(\cdot)] \tag{2c}$$

$$a_{\text{Ex}}(t) \in [a_{\text{xmin}}, a_{\text{xmax}}] \tag{2d}$$

$$v_{\text{Ey}}(t) \in [s_{\text{min}}, s_{\text{max}}]v_{\text{Ex}}(t) \tag{2e}$$

$$y_{\text{E}}(t) \geq w + (c_1^{\text{cz}}(t) + c_2^{\text{cz}}(t) - 1)w_l \tag{2f}$$

$$y_{\text{E}}(t) \leq w_l - w + (c_1^{\text{ow}}(t) + c_2^{\text{ow}}(t) - 1)w_l \tag{2g}$$

$$\begin{gathered}\frac{x_{\text{E}}(t) - (x_{\text{O0}}(t) - v_{\text{O}}t)}{l_{\text{Of}}} + \frac{y_{\text{E}}(t) - y_{\text{O}}}{w_l} +\\ + (c_1^{\text{ow}}(t) + c_2^{\text{ow}}(t) - 2)M \leq -1\end{gathered} \tag{2h}$$

$$\mathbf{x}_{\text{E}}(0) = \mathbf{x}_{\text{E0}} \tag{2i}$$

$$(c_1^i(t), c_2^i(t)) \in \mathcal{B}_t^i(x_{\text{E}}(t)),\ i = \{\text{cz}, \text{ow}\} \tag{2j}$$

where $\mathbf{x}_{\text{E0}} = [x_{\text{E0}}, v_{\text{E0}}, y_{\text{E0}}]^T$ denote the initial state values and all constraints, except (2i), should hold for all $t \in [0, t_f]$.

III. Continuous modelling in the spatial domain

In this section the program (2) is reformulated as a continuous optimization program. This is done in two steps, in the same fashion as in [12], [13], i.e., by first changing the reference frame to relative velocity and then sampling in the distance instead of absolute time.

A. Change of reference frame

In order to switch the reference frame of the ego vehicle to velocity relative to the leading vehicle, introduce the vector $\mathbf{p_L}(t) = [v_{\text{L}}t, v_{\text{L}}, 0]^T$. Thus, the relative control and state vectors read

$$\begin{aligned}&\tilde{\mathbf{x}}_{\text{E}}(t) = \mathbf{x}_{\text{E}}(t) - \mathbf{p_L}(t) = [\tilde{x}_{\text{E}}(t), \tilde{v}_{\text{Ex}}(t), y_{\text{E}}(t)]^T,\\ &\mathbf{u}_{\text{E}}(t) = [a_{\text{Ex}}(t),\ v_{\text{Ey}}(t)]^T.\end{aligned}$$

The new state space model reads

$$\dot{\tilde{\mathbf{x}}}_{\text{E}}(t) = A\tilde{\mathbf{x}}_{\text{E}}(t) + B\mathbf{u}_{\text{E}}(t),$$

with A and B defined as in (1).

B. Change of independent variable

Now, the sampling variable is changed from time t to relative longitudinal distance, $\tilde{x}$. A consequence of sampling in the spatial domain is that the position of the ego vehicle can be removed from the state vector. However, travel time, $\tilde{t}_{\mathrm{E}}$, is introduced as an additional state in the problem, for which it holds that $\tilde{t}'_{\mathrm{E}}(\tilde{x}) = 1/\dot{\tilde{x}}_{\mathrm{E}}(\tilde{x})$. Here, $(\cdot)'$ denotes the derivative with respect to relative distance, i.e., $y' = dy/d\tilde{x}$. The new state and control vectors are

$$\begin{aligned}\tilde{\mathbf{x}}_{\mathrm{E}}(\tilde{x}) &= [\tilde{v}_{\mathrm{Ex}}(\tilde{x}), y_{\mathrm{E}}(\tilde{x}), \tilde{t}_{\mathrm{E}}(\tilde{x})]^T,\\ \tilde{\mathbf{u}}_{\mathrm{E}}(\tilde{x}) &= [\tilde{a}_{\mathrm{Ex}}(\tilde{x}), v_{\mathrm{E}y}(\tilde{x})]^T.\end{aligned}$$

Further, since the positions of all the vehicles are now known at all samples, the binary variables are no longer needed. Hence, the constraint (2j) is removed, while the lateral limits and ramp barrier can be expressed as

$$y(\tilde{x}) \geq y_{\min}(\tilde{x}) = \begin{cases} w_l + w, & \tilde{x} \in x_{\mathrm{L}0} + [-l_{\mathrm{L}f}, l_{Lr}] \\ w, & \text{otherwise} \end{cases}$$

$$y(\tilde{x}) \leq y_{\max}(\tilde{x}) = \begin{cases} 2w_l - w, & \tilde{x} \in x_{\mathrm{L}0} + [-l_s, l_e] \\ w_l - w, & \text{otherwise} \end{cases}$$

$$\frac{\tilde{x} - x_{\mathrm{O}0} - (v_{\mathrm{O}} - v_{\mathrm{L}})\tilde{t}_{\mathrm{E}}(\tilde{x})}{l_{\mathrm{O}f}} + \frac{y_{\mathrm{E}}(\tilde{x}) - y_{\mathrm{O}}}{w_l} \leq -1.$$

However, notice that the first control input has been changed (in order to preserve linear dynamics) to $\tilde{a}_{\mathrm{Ex}}(\tilde{x}) = v_{\mathrm{Ex}}(\tilde{x})v'_{\mathrm{Ex}}(\tilde{x})$, which leads to the non-convex acceleration limits $\tilde{a}_{\mathrm{E}x}(\tilde{x}) \in [a_{\mathrm{xmin}}, a_{\mathrm{xmax}}]/\tilde{v}_{\mathrm{Ex}}(\tilde{x})$. This non-convexity is addressed in the next section.

C. Change of optimization variables

Replace the longitudinal velocity, $\tilde{v}_{\mathrm{Ex}}(\tilde{x})$, with the kinetic energy, $\tilde{E}_{\mathrm{Ex}}(\tilde{x}) = m\tilde{v}_{\mathrm{Ex}}(\tilde{x})^2/2$. The state and control vectors are then expressed as

$$\hat{\mathbf{x}}_{\mathrm{E}}(\cdot) = [\tilde{E}_{\mathrm{Ex}}(\tilde{x}), y_{\mathrm{E}}(\tilde{x}), \tilde{t}_{\mathrm{E}}(t)]^T,\ \hat{\mathbf{u}}_{\mathrm{E}}(\cdot) = [a_{\mathrm{Ex}}(\tilde{x}),\ v_{\mathrm{E}y}(\tilde{x})]^T.$$

and the state space model becomes

$$\hat{\mathbf{x}}'_{\mathrm{E}}(\tilde{x}) = \left[m a_{\mathrm{E}x}(\tilde{x})\ \ v_{\mathrm{E}y}(\tilde{x})\ \ \sqrt{\tfrac{m}{2\tilde{E}_{\mathrm{Ex}}(\tilde{x})}} \right]^T, \tag{3}$$

where we have used that $\tilde{E}'_{\mathrm{E}x}(\tilde{x}) = m\tilde{v}'_{\mathrm{Ex}}(\tilde{x})\tilde{v}_{\mathrm{Ex}}(\tilde{x})$ and $a_{\mathrm{E}x}(\tilde{x}) = \tilde{v}_{\mathrm{Ex}}(\tilde{x})\tilde{v}'_{\mathrm{Ex}}(\tilde{x})$. The state constraint $\hat{\mathbf{x}}_{\mathrm{E}}(\tilde{x}) \in [\hat{\mathbf{x}}_{\min}(\tilde{x}), \hat{\mathbf{x}}_{\max}(\tilde{x})]$ has the limits

$$\begin{aligned}\hat{\mathbf{x}}_{\min}(\tilde{x}) &= [m\epsilon^2/2, y_{\min}(\tilde{x}), \text{free}]^T,\\ \hat{\mathbf{x}}_{\max}(\tilde{x}) &= [m(v_{\mathrm{xmax}} - v_{\mathrm{L}})^2/2, y_{\max}(\tilde{x}), \text{free}]^T.\end{aligned}$$

Further, the acceleration bounds and lateral slip constraints translate to

$$\begin{aligned}& a_{\mathrm{E}x}(\tilde{x}) \in [a_{\min}, a_{\max}],\\ & v_{\mathrm{E}y}(\tilde{x}) \in [s_{\min}, s_{\max}]\left(1 + \sqrt{\frac{m v_{\mathrm{L}}^2}{2\tilde{E}_{\mathrm{Ex}}(\tilde{x})}}\right).\end{aligned}$$

Still the problem is non-convex due to the travel time dynamics and the slip constraints. In the next section, these will be convexified.

D. Convex relaxation

The time dynamics constraint in (3) could be convexified in several ways. Instead of standard linearization techniques, the following relaxation is proposed

$$\tilde{t}'_{\mathrm{E}}(\tilde{x}) \geq \sqrt{m/(2\tilde{E}_{\mathrm{E}x}(\tilde{x}))},$$

the validity of which is discussed in Section III-E, after the full convexified problem has been formulated.
On the other hand, a standard linearization technique is applied for the slip constraints. Linearizing $1/\tilde{E}_{\mathrm{E}x}(\tilde{x})^{(1/2)}$ around the reference energy $\tilde{E}_{\mathrm{lin}}(\tilde{x})$ yields

$$1/\sqrt{\tilde{E}_{\mathrm{Ex}}(\tilde{x})} \approx f_{\mathrm{lin}}(\tilde{E}_{\mathrm{lin}}(\tilde{x}), \tilde{E}_{\mathrm{E}x}(\tilde{x})).$$

E. Convex formulation

If the cost function $\hat{J}$ is assumed convex, the convex version of problem (2) reads

$$\underset{\hat{\mathbf{u}}_{\mathrm{E}}(\tilde{x})}{\text{minimize}}\ \hat{J}(\hat{\mathbf{x}}_{\mathrm{E}}(\tilde{x}), \hat{\mathbf{u}}_{\mathrm{E}}(\tilde{x}), \hat{\mathbf{u}}'_{\mathrm{E}}(\tilde{x})) + \varepsilon\tilde{t}_{\mathrm{E}}(\tilde{x}_f) + \mathcal{Q}(\cdot) \tag{4a}$$

subject to

$$\left[\tilde{E}'_{\mathrm{E}x}(\tilde{x})\ y'_{\mathrm{E}}(\tilde{x})\right]^T = \left[m a_{\mathrm{E}x}(\tilde{x})\ v_{\mathrm{E}y}(\tilde{x})\right]^T \tag{4b}$$

$$\tilde{t}'_{\mathrm{E}}(\tilde{x}) \geq \sqrt{\frac{m}{2\tilde{E}_{\mathrm{Ex}}(\tilde{x})}} \tag{4c}$$

$$\hat{\mathbf{x}}_{\mathrm{E}}(\tilde{x}) \in [\hat{\mathbf{x}}_{\min}(\tilde{x}), \hat{\mathbf{x}}_{\max}(\tilde{x})] \tag{4d}$$

$$a_{\mathrm{E}x}(\tilde{x}) \in [a_{\mathrm{xmin}}(\tilde{x}), a_{\mathrm{xmax}}(\tilde{x})] \tag{4e}$$

$$v_{\mathrm{E}y}(\tilde{x}) \in [s_{\min}, s_{\max}]\left(1 + \sqrt{\frac{m v_{\mathrm{L}}^2}{2}} f_{\mathrm{lin}}(\tilde{E}_{\mathrm{lin}}, \tilde{E}_{\mathrm{E}x})\right) \tag{4f}$$

$$\hat{\mathbf{x}}_{\mathrm{E}}(0) = \hat{\mathbf{x}}_{\mathrm{E}0} \tag{4g}$$

$$\frac{\tilde{x} - x_{\mathrm{O}0} - (v_{\mathrm{O}} - v_{\mathrm{L}})\tilde{t}_{\mathrm{E}}(\tilde{x})}{l_{\mathrm{O}f}} + \frac{y_{\mathrm{E}}(\tilde{x}) - y_{\mathrm{O}}}{w_l} \leq -1 \tag{4h}$$

where the constraints (4b)-(4f) are imposed for all $\tilde{x} \in [0, \tilde{x}_f]$ while (4h) holds for all $\tilde{x} \in x_{\mathrm{L}0} + [-l_s, l_e]$. The extra term in the objective, $\varepsilon\tilde{t}_{\mathrm{E}}(\tilde{x}_f)$, is added to ensure that the inequality (4c) is tight at the optimum, see [13] for a proof. The additional term $\mathcal{Q}$ will be introduced in Section V.

IV. Objective functions

A quadratic cost function is used to punish states from deviating from the reference trajectory $\mathbf{x}_r(t) = [0, v_r(t), y_r(t)]^T$, changes in control actions and changes in the derivative of the control actions,

$$J = \int_0^{t_f} \|\mathbf{x}_{\mathrm{E}}(t) - \mathbf{x}_{\mathrm{r}}\|_Q^2 + \|\mathbf{u}_{\mathrm{E}}(t)\|_R^2 + \|\dot{\mathbf{u}}_{\mathrm{E}}(t)\|_S^2 dt, \tag{5}$$

where Q, R and S are positive semidefinite weighting matrices, and the norm is defined as $\|a\|_A^2 := a^T A a$. The corresponding objective function in the spatial domain punishes deviation from $\hat{\mathbf{x}}_{\mathrm{r}}(\tilde{x}) = [\tilde{E}_r(\tilde{x}), y_r(\tilde{x}), 0]^T$:

$$\hat{J}(\cdot) = \int_0^{\tilde{x}_f} V(\hat{\mathbf{x}}_{\mathrm{E}}, \hat{\mathbf{u}}_{\mathrm{E}}, \hat{\mathbf{u}}'_{\mathrm{E}}) d\tilde{x} \tag{6}$$

where $V = \|\hat{\mathbf{x}}_{\mathrm{E}}(\tilde{x}) - \hat{\mathbf{x}}_{\mathrm{r}}\|_{\tilde{Q}}^2 + \|\hat{\mathbf{u}}_{\mathrm{E}}(\tilde{x})\|_{\tilde{R}}^2 + \|\hat{\mathbf{u}}'_{\mathrm{E}}(\tilde{x})\|_{\tilde{S}}^2$ and $\tilde{Q}$, $\tilde{R}$ and $\tilde{S}$ are positive semidefinite weighting matrices.

The cost functions chosen for the temporal and spatial formulations are not equal, since the spatial and temporal derivatives are different. However, it is possible to make a direct translation of the temporal cost function (5) into the spatial domain, see [12]. This direct translation makes it possible to, approximately, translate the weights used in the temporal formulation (5) to the spatial formulation (6), i.e., one can choose which of the formulations one prefers to tune and then directly translate the weights into the other formulation. This is important, since the tuning of the more unusual objective function (6) may be seen as a liability.

V. Sequential SOCP

While the temporal program (2) will be solved using a standard mixed integer quadratic program (MIQP) solver, the spatial program, (4), will be solved using the sequential SOCP method, which is related to the sequential QP method [18]. The term $\mathcal{Q}$ is chosen as

$$\mathcal{Q} = \frac{1}{2}\nabla^2_{\tilde{E}_{\mathrm{E}x}} \mathcal{L}(E_{\mathrm{lin}}, \boldsymbol{\lambda}) \Delta \tilde{E}(\tilde{x})^2 + \nabla V^T \Delta \hat{\mathbf{x}} \tag{7}$$

where ∇ is the gradient, ∇^2_x is the Hessian with respect to x, $\mathcal{L}$ is the lagrangian when relaxing the slip constraints, and $\lambda = [\lambda_1, \ \lambda_2]^T$ are the dual variables of the slip constraints, respectively, and $\Delta\hat{\mathbf{x}}^T = [\Delta\hat{\mathbf{x}}_{\mathrm{E}}^T, \Delta\hat{\mathbf{u}}_{\mathrm{E}}^T, \Delta\hat{\mathbf{u}}_{\mathrm{E}}'^T]^T$. With the objective function (4a), where $\hat{J}$ and $\mathcal{Q}$ are chosen as in (6) and (7), the problem (4) is an SOCP. This program is solved sequentially and updating is done by moving in the direction of the current optimal solution

$$\hat{\mathbf{x}}^{k+1} = \hat{\mathbf{x}}^k + \alpha \Delta \hat{\mathbf{x}}^k$$

where $\alpha \in (0, 1]$ is the step length and k is the current iteration. This procedure continues until the change in linearization between iterations fulfill a stopping criteria, provided by the user. The algorithm is outlined in Algorithm V.1. Since, the linearizations and relaxation are inner approximations, each iteration in Algorithm V.1 yields a feasible solution to the non-relaxed spatial program. The advantage of this is that, when seen as the optimal control problem in MPC, the sequential SOCP need not converge to optimality in each MPC cycle. Instead, it is enough to solve one SOCP in each iteration of the MPC cycle. Thus, the solution in the initial iterations will be feasible, but might be suboptimal. This procedure is called real time iterations (RTI) [19].

Algorithm V.1.

```
Step 1: Choose a linearization $\hat{\mathbf{x}}_{\mathrm{lin}}^1$ and initialize $\lambda_1^1, \lambda_2^1$.
Step 2: Solve (4) for $\Delta\hat{\mathbf{x}}^k$ around $\hat{\mathbf{x}}_{\mathrm{lin}}^k$, $\lambda_1^k$, $\lambda_2^k$.
Step 3: Check if $\|\Delta\hat{\mathbf{x}}^k\|_2 < \delta$. If yes, stop. Otherwise,
        update $\hat{\mathbf{x}}_{\mathrm{lin}}^{k+1} = \hat{\mathbf{x}}_{\mathrm{lin}}^k + \Delta\hat{\mathbf{x}}^k$.
Step 4: Set $\lambda_1^{k+1}$, $\lambda_2^{k+1}$ to the dual variables of their
        respective constraint and go to step 2.
```

VI. Case study

The case study compares the solutions and computational efforts of the two programs. To compare computational efforts, the overtaking scenario is solved for several different partitions of the temporal and spatial horizons, and in the presence or absence of an oncoming vehicle.

The problems are discretized and then implemented in MATLAB using the software YALMIP [20]. The commercial solver GUROBI is used to solve the MIQP, while the SOCP formulation is solved using ECOS [21].

TABLE I

Problem parameters.

Ego vehicle	Other vehicles	Problem specific parameters
x_{E0} $=0\,\mathrm{m}$	x_{L0} $=75\,\mathrm{m}$	$\tilde{x}_f$ $=150\,\mathrm{m}$
y_{E0} $=2.5\,\mathrm{m}$	y_{L0} $=2.5\,\mathrm{m}$	t_f $=27\,\mathrm{s}$
v_r $=70\,\mathrm{km/h}$	$v_{\mathrm{L}x}$ $=50\,\mathrm{km/h}$	Q $=\mathrm{diag}(0, 0.06, 0.6)$
v_{x0} $=70\,\mathrm{km/h}$	$l_{\mathrm{L}r}$ $=12.3\,\mathrm{m}$	R $=\mathrm{diag}(0.4, 3.6)$
v_{y0} $=0\,\mathrm{km/h}$	$l_{\mathrm{L}f}$ $=15\,\mathrm{m}$	S $=\mathrm{diag}(0.6, 2)$
$v_{x\mathrm{min}}$ $=0\,\mathrm{km/h}$	l_s $=40\,\mathrm{m}$	$\tilde{Q}$ $=\mathrm{diag}(37\,\mathrm{psm}^{-1}\mathrm{J}^{-2}, 0.1)$
$v_{x\mathrm{max}}$ $=80\,\mathrm{km/h}$	l_e $=37.3\,\mathrm{m}$	$\tilde{R}$ $=\mathrm{diag}(0.065, 50)$
$a_{x\mathrm{min}}$ $=-4\,\mathrm{m/s^2}$	x_{O0} $=650\,\mathrm{m}$	$\tilde{S}$ $=\mathrm{diag}(3.24, 800)$
$a_{x\mathrm{max}}$ $=1\,\mathrm{m/s^2}$	y_{O0} $=7.5\,\mathrm{m}$	M $=1000$
β $=10°$	$v_{\mathrm{O}x}$ $=-70\,\mathrm{km/h}$	w $=1.5\,\mathrm{m}$
ϵ $=0.01$	$l_{\mathrm{O}f}$ $=48.4\,\mathrm{m}$	w_l $=5\,\mathrm{m}$

A. Quality of solutions

The parameters used in the case study are listed in Table I. All the parameters are used in both formulations except for $t_f, Q, R, S,$ which are needed in the temporal formulation, and $\tilde{x}_f, \tilde{Q}, \tilde{R}, \tilde{S}$ which are needed in the spatial formulation. Note that, to be able to make a fair comparison, the length $\tilde{x}_f$ of the prediction horizon in the spatial formulation was chosen as the distance travelled by the ego vehicle in the temporal formulation. Fig. 3 illustrates the optimal lateral trajectory generated with the temporal formulation and it also depicts vehicles' locations at chosen time instances. Further, from Fig. 4, which depicts the optimal state and control trajectories for both the temporal and spatial formulation, it can be seen that the formulations both offers smooth solutions. The differences in the solutions is caused by the fact that the cost functions in the programs are not identical.

B. Computational effort

TABLE II

Time spent solving the temporal and spatial programs for the scenario in Table 1, with or without (w-o) oncoming vehicle. H-S stands for horizon samples.

H-S	time w-o O	space w-o O	time w O	space w O
41	1.20 s	0.015 s	3 s	0.16 s
51	1.92 s	0.015 s	1.4 s	0.19 s
61	5.93 s	0.024 s	1.28 s	0.16 s
71	4.69 s	0.055 s	8.52 s	0.19 s

As one would expect with the MIQP, the temporal program (2), is less computationally efficient than the SOCP, the spatial program (4) , see Table II. In an attempt to speed up the temporal formulation, we have provided it with the reference trajectories as initial guesses. As can be seen the spatial formulation solves the problem, when no oncoming vehicle is present, in milliseconds while it takes seconds for the temporal formulation. In the case of an oncoming vehicle present, it takes a bit longer for both algorithms, but the spatial formulation is still significantly faster. Further,

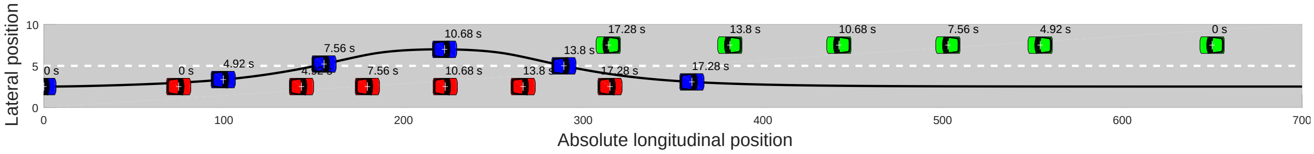

Fig. 3. Illustration of time formulation solution of the ego vehicle (blue) overtaking a leading vehicle (red) in the presence of an oncoming vehicle (green). The black line is the travelling trajectory of the CoG of the ego vehicle and the black numbers indicate time instances.

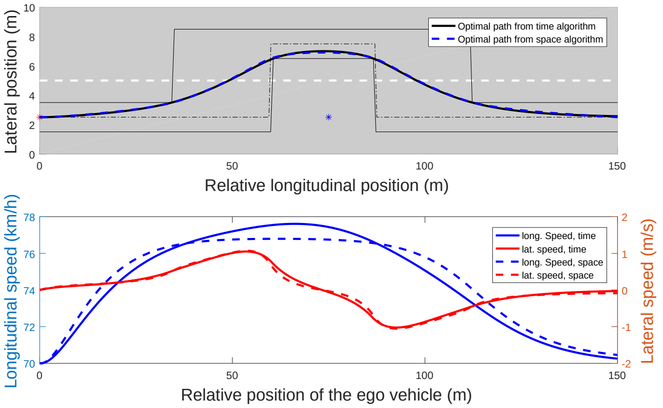

Fig. 4. Scenario where the ego vehicle is overtaking a leading vehicle on a road with two lanes while increasing its velocity to avoid colliding with an oncoming vehicle. The CoG of the ego vehicle is allowed to reside between the limits $y_{\min}$ and $y_{\max}$, depicted with solid lines. In the top plot the dashed-dotted line is the reference position of the ego vehicle.

remember that the algorithm used to solve the spatial formulation is iterative, and that in Table II the full solving time is presented. However, in practice, on might solve only the first iteration in the spatial formulation and then carry that solution over to the next cycle in the MPC, which would decrease the computational effort even further. For all the scenarios listed in Table II the sequential SOCP required two to five iterations to converge. However, already after the second iteration the cost is equal to the optimal cost with a precision of four decimals.

VII. Conclusion

A new spatial formulation of the overtaking problem was introduced, involving a variable change from velocity to kinetic energy. The reason for this variable change was to convexify constraints on the acceleration without using linearization. The case study revealed that the spatial formulation is more computationally efficient than the time formulation. This was expected, since SOCPs are generally more computationally efficient than MIQPs. The resulting longitudinal velocity for the spatial domain has a smoother appearance than that of the time domain. However, this difference could be decreased by tuning the weighting matrices in the objective function of the temporal formulation.

References

[1] M. van Schijndel-de Nooij, B. Krosse, T. van den Broek, S. Maas, E. van Nunen, H. Zwijnenberg, A. Schieben, H. Mosebach, N. Ford, M. McDonald, D. Jeffery, J. Piao, and J. Sanchez, “Definition of necessary vehicle and infrastructure systems for automated driving,” European Commission, Tech. Rep. 2010/0064, 2011.

[2] D. J. Fagnant and K. Kockelman, “Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations,” *Transportation Research Part A: Policy and Practice*, vol. 77, pp. 167 – 181, 2015.

[3] T. Victor, M. Rothoff, E. Coelingh, A. Ödblom, and K. Burgdorf, *When Autonomous Vehicles Are Introduced on a Larger Scale in the Road Transport System: The Drive Me Project*. Cham: Springer International Publishing, 2017, pp. 541–546.

[4] T. Litman, “Autonomous vehicle implementation predictions,” *Victoria Transport Policy Institute*, vol. 28, 2014.

[5] J. Nilsson, J. Silvin, M. Brännström, E. Coelingh, and J. Fredriksson, “If, when, and how to perform lane change maneuvers on highways,” *Intelligent transportation systems magazine*, vol. 8, 2016.

[6] G. Usman and F. Kunwar, “Autonomous vehicle overtaking- an online solution,” in *International conference on Automation and Logistics*, Shenyang, China, 2009.

[7] T. Shamir, “How should an autonomous vehicle overtake a slower moving vehicle: Design and analysis of an optimal trajectory,” *IEEE Transactions on optimal control*, vol. 49, pp. 607–610, 2004.

[8] Y. Kuwata, G. Fiore, J. Teo, E. Frazzoli, and J. How, “Motion planning for urban driving using RRT,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 1681–1686.

[9] J. Ziegler, M. Werling, and J. Schröder, “Navigating car-like robots in unstructured environments using an obstacle sensitive cost function,” in *IEEE Intelligent Vehicles Symposium*, 2008, pp. 787–791.

[10] F. Molinari, N. N. Anh, and L. D. Re, “Efficient mixed integer programming for autonomous overtaking,” in *American control conference (ACC)*, Seattle, USA, 2017, pp. 2303–2308.

[11] J. Nilsson, M. Ali, P. Falcone, and J. Sjöberg, “Predictive cruise control with autonomous overtaking,” in *Intelligent Transportation Systems (ITSC)*, The Hague, Netherlands, 2013.

[12] J. Karlsson, N. Murgovski, and J. Sjöberg, “Temporal vs. spatial formulation of autonomous overtaking algorithms,” in *ITSC*, Rio, Brazil, 2016.

[13] N. Murgovski and J. Sjöberg, “Predictive cruise control with autonomous overtaking,” in *Decision and Control (CDC)*, Osaka, Japan, 2015.

[14] D. Casanova, “On minimum time vehicle manoeuvring: The theoretical optimal lap,” 2000.

[15] S. Boyd and L. Vanderberghe, *Convex optimization*, 1st ed. Cambridge University Press, 2004.

[16] Y. Puranik and N. V. Sahinidis, “Domain reduction techniques for global NLP and MINLP optimization,” *Constraints*, vol. 22, 2017.

[17] S. Gerard, “Linear and integer programming: theory and practice,” 2002.

[18] J. Nocedal and S. J. Wright, *Numerical optimization*, 2nd ed. Springer, 2000.

[19] M. Diehl, “Real-time optimization for large scale nonlinear processes,” Ph.D. dissertation, 2001.

[20] J. Löfberg, “YALMIP: A toolbox for modeling and optimization in matlab,” in *International Symposium on Computer Aided Control Systems Design*, Zürich, Switzerland, 2004.

[21] A. Domahidi, E. Chu, and S. Boyd, “ECOS: An SOCP solver for embedded systems,” in *European Control Conference (ECC)*, Zürich, Switzerland, 2013.