



Optimal Trajectory Planning and Decision Making in Lane Change Maneuvers Near a Highway Exit

Downloaded from: <https://research.chalmers.se>, 2021-05-10 05:59 UTC

Citation for the original published paper (version of record):

Karlsson, J., Murgovski, N., Sjöberg, J. (2019)

Optimal Trajectory Planning and Decision Making in Lane Change Maneuvers Near a Highway Exit
European control conference

N.B. When citing this work, cite the original published paper.

Optimal trajectory planning and decision making in lane change maneuvers near a highway exit

Johan Karlsson, Nikolce Murgovski and Jonas Sjöberg

Abstract—This paper presents a control architecture for autonomous driving near a highway exit. In particular, detailed descriptions are provided for the two main blocks in the proposed architecture; a novel trajectory planner for lateral and longitudinal motion, and a novel decision manager for choosing the optimal target lane. We show that the trajectory planning can be realized by executing two sequential quadratic programs as two model predictive controllers, where one is planning the optimal trailing of a leading vehicle and the other is planning a lane change maneuver. Having the two optimal trajectories and a scalarized objective that considers safety, comfort, reference tracking and distance to the exit, the decision manager chooses one of the planned trajectories that is sent to the vehicle. A case study is provided showing the effectiveness of the proposed control architecture.

I. INTRODUCTION

A large number of traffic accidents happening on the roads today are due to human factors such as fatigue, distracted driving and speeding [1], [2]. To reduce the number of accidents, a surge towards automated driving has been seen in recent years. This surge has included driving assistance systems such as adaptive cruise control (ACC), cooperative ACC (CACC) [3], collision avoidance systems [4] and lane keeping assistance [5]; many of which are already standard in many production vehicles today [6]. To further improve traffic safety and lower energy consumption, research is being done towards fully autonomous vehicles, where a vast number of possible driving scenarios are studied to address traffic regulations, safety, energy efficiency and driver comfort, [7], [8].

One of the most studied scenarios for autonomous vehicles is highway driving. This is because accidents tend to be severe, due to the high velocity of the vehicles and that highway travel tends to be long distance which increase the risk of driver fatigue or loss of attention. In this paper we study a particular highway scenario, where the ego vehicle intends to exit the highway, either to the left or right. Naturally, if the ego vehicle is in the same lane as the upcoming exit, no action is needed besides that of staying in the lane and avoiding collision with potential vehicles in front, from now on referred to as trailing. However, if the exit is in the adjacent lane, a lane change needs to be performed, well in time before the exit is reached. To solve this problem, we propose a control architecture, which consists of two main parts, a novel trajectory planner and a novel decision maker.

The authors would like to thank the master student Gudrun Dovner for her development of the decision manager and the implementation of the architecture presented in this paper.

The trajectory planner includes calculation of two candidate trajectories, one for trailing a leading vehicle in the same lane, and another for performing a lane change. For the trajectory calculation, we propose a combination of two model predictive controllers (MPCs), one optimizing in a temporally-defined receding horizon and another in a spatially-defined horizon. The combination of the two MPCs allows for different implementation of the collision avoidance constraints, which makes it possible to formulate both the trailing and lane change programs as quadratic programs (QPs). This is important since QPs can be solved efficiently; efficiency is necessary since two optimization programs are solved at each instance, one for trailing and one for lane change planning.

For the trailing MPC, we propose a temporal formulation which is a combination of ACC and lane keeping control, for both longitudinal and lateral vehicle motion, while considering ramp barrier constraints for surrounding vehicles, similarly as proposed by [9]. For the lane change MPC, we propose a combination of a temporal and a spatial formulation which are triggered depending on the speed difference of the ego and leading vehicle. We show that for the scenario with a slow moving leading vehicle, the spatial QP formulation allows usage of a box constraint for preventing collision with the leading vehicle [10], instead of a ramp barrier. This is advantageous since a box constraint allows for planning beyond the position of the leading vehicle, which a ramp barrier does not.

The paper is organized as follows. A description of the problem is provided in Section II. An overview of the full architecture is then presented in Section III. In Sections IV and V the trajectory planner and decision manager are introduced, respectively. Finally, a case study is presented in Section VI before conclusions are drawn in Section VII.

II. PROBLEM FORMULATION

We study the problem of controlling an autonomous vehicle, from here on referred to as the ego vehicle (E), driving on a straight, two-lane highway. Three surrounding vehicles are considered: the leading vehicle (L) starting in front of and in the same lane as the ego vehicle, two adjacent vehicles (R) and (F), where (F) is in front of, and (R) is behind, (E), see Fig. 1. Specifically, we consider the case when the ego vehicle's goal is to exit the highway. The task of the controller is to calculate two candidate paths, one for trailing, based on the reference velocity $v_r(t)$ and lateral reference position $y_r^l(t)$, and one for lane change, based on the reference velocity $v_r(t)$ and lateral reference

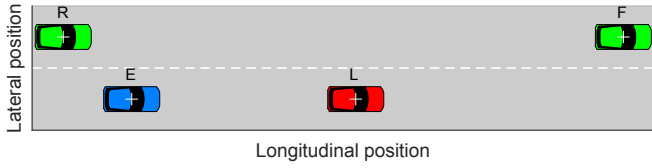


Fig. 1: Straight highway with two lanes and four vehicles: the ego vehicle (E), the leading vehicle (L) and the adjacent vehicles (R) and (F).

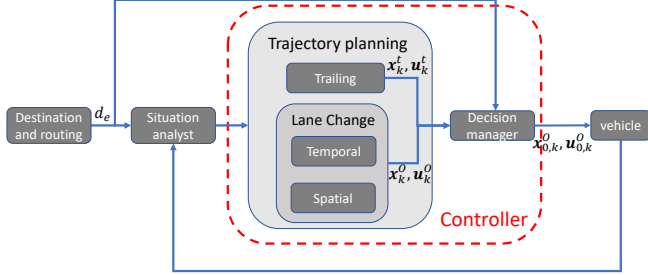


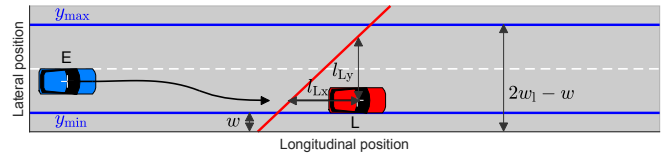
Fig. 2: The control architecture consists of four parts, from left to right: destination and routing, which sets the goal of the maneuver, the situation analyst which analyses the current traffic situation, the controller which calculates optimal trajectories and decides whether lane change or trailing should be performed and finally the vehicle itself.

position $y_r^{ls}(t)$, and then choose which of these trajectories is safest and most comfortable while bringing us closer to the overall goal.

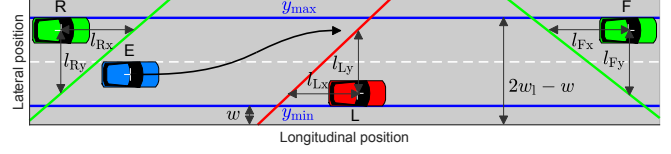
III. ARCHITECTURE

We address the problem by proposing a feedback control architecture, as illustrated in Fig. 2.

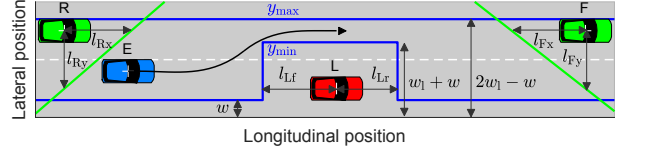
The destination and routing block decide which exit to take and where it is located, for a horizon of up to several kilometers. The situation analyst gathers data in a small neighborhood around the ego vehicle, which is relevant for planning the trajectory of the ego vehicle within the MPC horizon. This includes information on the current state of the ego vehicle, a prediction of the future trajectories of the surrounding vehicles and parameters needed for modeling the safety constraints. This data is then sent to the controller, which consists of two parts, a trajectory planner and a decision manager. In every MPC update k the trajectory planner calculates two candidate trajectories, one trailing trajectory $(\mathbf{x}_k^l, \mathbf{u}_k^l)$ and one lane change trajectory $(\mathbf{x}_k^l, \mathbf{u}_k^l)$, both consisting of a state vector \mathbf{x} and an input vector \mathbf{u} , which will be introduced in Section IV-A and IV-B, respectively. These trajectories are then sent to the decision manager, that decides which of these two trajectories should be sent to the local controllers of the ego vehicle. The output of the decision manager, which is also the output of the controller, is the first state and control of the chosen trajectory $(\mathbf{x}_k^o, \mathbf{u}_k^o)$, labelled $(x_{k,0}^o, u_{k,0}^o)$. The input and state $(x_{k,0}^o, u_{k,0}^o)$ is then applied to the vehicle. Further discussion on the trajectory planning and the decision manager is provided in Sections IV and V. The situation analyst, the local controllers and the destination and routing are outside the scope of this paper and are not discussed further.



(a) Trailing a leading vehicle. The safety constraint preventing a collision with the leading vehicle is modeled as a ramp barrier.



(b) Lane change in the temporal domain. The safety constraint preventing a collision with the surrounding vehicles are modeled as ramp barriers.



(c) Lane change in the spatial domain. The safety constraint preventing a collision with the leading vehicle is modeled with a critical zone.

Fig. 3: Illustration of traffic scenario on a highway with two adjacent vehicles (R and F) moving in the same direction as the ego vehicle. During the lane change maneuver, the ego vehicle's center of gravity is allowed to reside between the limits y_{\min} , y_{\max} and the ramp barriers of the surrounding vehicles.

IV. TRAJECTORY PLANNING

The trajectory planner solves two optimization programs, one to calculate a trailing trajectory, see Fig. 3a and one to calculate a lane change trajectory.

For the lane change planning, two cases are considered, depending on the speed of the leading vehicle. These two cases, which are described in more detail in Section IV-B, results in two problem formulations, one temporal formulation and one relative spatial formulation. As can be seen in Fig. 3 collision avoidance is modeled via ramp barriers or rectangular critical zones where w_l is the lane width, w is the road safety distance and the longitudinal parameters of the ramp barriers and critical zone are calculated by

$$l_{Li} = \frac{l_L + l_E}{2} + \theta_{Li}(v_r(t) - v_L(t)) + \tilde{\theta}_{Li}v_r(t), \quad i = \{x, f, r\},$$

$$l_{jx} = \frac{l_j + l_E}{2} + \theta_j(v_r(t) - v_j(t)) + \tilde{\theta}_jv_r(t), \quad j = \{F, R\},$$

where $v_r(t)$ is the longitudinal reference speed of the ego vehicle, $v_L(t)$ is the longitudinal velocity of the leading vehicle, v_F is the longitudinal velocity of (F), v_R is the longitudinal velocity of (R), θ_{Li} is the desired time gap between the leading and ego vehicle, θ_j is the desired time gap between vehicle j and the ego vehicle, while $\tilde{\theta}_{Li}$ and $\tilde{\theta}_j$ is a time gap increasing the safety distance with increasing velocity of the ego vehicle. A typical choice for the lateral parameters of the ramp barriers would be $l_{Ly} = l_{Ry} = l_{Fy} = w_l$, [10].

The trailing and lane change trajectory planning problems

are formulated as QPs of the form

$$\min_{\mathbf{u}_k^j(\tau)} \int_{\tau_{0,k}^j}^{\tau_f^j} \|\mathbf{x}_k^j(\tau) - \mathbf{x}_{r,k}^j(\tau)\|_{Q^j}^2 + \|\mathbf{u}_k^j(\tau)\|_{R^j}^2 + \|\frac{d}{d\tau}\mathbf{u}_k^j(\tau)\|_{S^j}^2 d\tau \quad (1a)$$

subject to

$$\frac{d}{d\tau}\mathbf{x}_k^j(\tau) = A^j\mathbf{x}_k^j(\tau) + B^j\mathbf{u}_k^j(\tau), \quad (1b)$$

$$\mathbf{x}_k^j(\tau) \in [\mathbf{x}_{\min,k}^j, \mathbf{x}_{\max,k}^j], \quad (1c)$$

$$\mathbf{u}_k^j(\tau) \in [\mathbf{u}_{\min,k}^j, \mathbf{u}_{\max,k}^j], \quad (1d)$$

$$C_k^j\mathbf{x}_k^j(\tau) + D_k^j\mathbf{u}_k^j(\tau) \leq \mathbf{g}_k^j, \quad (1e)$$

$$\mathbf{x}_k^j(0) = \mathbf{x}_{0,k}^j, \quad (1f)$$

where k is the k th step in the MPC-loop, τ is the sampling variable going from $[0, \tau_f]$, $\mathbf{x}_k(\tau)$ and $\mathbf{u}_k(\tau)$ are the state and input vectors, respectively, i.e., decision variables, $\mathbf{x}_{r,k}^j$ is the reference vector and $\mathbf{x}_{0,k}^j$ is the vector of initial states. The constraint (1b) describes the dynamics of the vehicle via the matrices A^j and B^j , (1c) and (1d) are box constraints on the states and inputs, stemming from physical restrictions of the ego vehicle and restrictions imposed by traffic rules, (1e) are general linear constraints on states and inputs modeled by the matrices $C_k^j, D_k^j, \mathbf{g}_k^j$ and (1f) are the initial state conditions. Further, $j \in \{t, lt, ls\}$ indicates which program we are currently solving, trailing $j = t$, temporal lane change program $j = lt$ or spatial lane change program $j = ls$.

A. Trailing program

Similarly as in [9], the ego vehicle is modeled with simple point mass dynamics, using the state and control vectors

$$\mathbf{x}_k^t(t) = [x_k^t(t), v_{x,k}^t(t), y_k^t(t), v_{y,k}^t(t)]^T, \quad (2)$$

$$\mathbf{u}_k^t(t) = [a_{x,k}^t(t), a_{y,k}^t(t)]^T, \quad (3)$$

where $x_k^t(t), y_k^t(t)$ is the longitudinal and lateral position of the ego vehicle at time t in MPC loop k , respectively, $v_{x,k}^t(t), v_{y,k}^t(t)$ is the corresponding velocities and $a_{x,k}^t(t), a_{y,k}^t(t)$ are the accelerations. The matrices of (1b) are thus given by

$$A^t = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B^t = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (4)$$

The trailing program calculates a candidate trajectory that keeps the ego vehicle in the current lane, see Fig. 3a, by trailing the leading vehicle while penalizing deviation from a given reference trajectory

$$\mathbf{x}_r^t = [0, v_r^t(t), y_r^t(t), 0], \quad (5)$$

where $y_r^t(t)$ is a lateral reference path, and actuator limits with the weighting matrices

$$Q^t = \text{diag}(0, q_x^t, q_y^t, 0), \quad (6)$$

$$R^t = \text{diag}(r_x^t, r_y^t), \quad (7)$$

$$S^t = \text{diag}(s_x^t, s_y^t). \quad (8)$$

Further, the right hand side of the box constraints (1c) and (1d) are given by

$$\mathbf{x}_{\min,k}^t = [0, w_E/2, 0, v_{y,k}^{\min}], \quad (9)$$

$$\mathbf{x}_{\max,k}^t = [\text{free}, 2w_l - \frac{w_E}{2}, v_{x,k}^{\max}, v_{y,k}^{\max}]^T. \quad (10)$$

$$\mathbf{u}_{\min,k}^t = [a_{x,k}^{\min}, a_{y,k}^{\min}], \quad (11)$$

$$\mathbf{u}_{\max,k}^t = [a_{x,k}^{\max}, a_{y,k}^{\max}], \quad (12)$$

where w_E is the width of the ego vehicle, $x_L(t) = v_L t + x_{L0}$ is the position of the leading vehicle, x_{L0} the initial position of the leading vehicle and w_l is the lane width. The speed and acceleration limits satisfy $v_{y,k}^{\min}(t), a_{y,k}^{\min}(t), a_{x,k}^{\min}(t) < 0$, $v_{y,k}^{\max}(t), a_{x,k}^{\max}(t), a_{y,k}^{\max}(t) > 0$. Additionally, for the trailing program, the general constraints (1e) contain three rows, the first two model slip constraints and the third the collision avoidance with the leading vehicle

$$C_k^t = \begin{bmatrix} 0 & 0 & s & 1 \\ 0 & 0 & s & -1 \\ \frac{1}{l_{Lx}} & \mp \frac{1}{l_{Ly}} & 0 & 0 \end{bmatrix}, \quad D_k^t = \mathbf{0}, \quad (13)$$

$$\mathbf{g}_k^t = \left[0, 0, -1 + \frac{x_L(t)}{l_{Lx}} \mp \frac{y_L(t)}{l_{Ly}} \right]^T, \quad (14)$$

where s is the maximum slip, calculated via the maximum slip angle β as $s = \arctan(\beta)$. Since, it is not allowed to switch lanes during trailing, vehicles in the adjacent lane are not considered.

The trailing program, i.e., the optimization (1) with the problem data (2)-(14), is a continuous, convex QP.

B. Lane change programs

The lane change maneuver involves collision avoidance modeling for, not only the leading vehicle, but also for the vehicles in the target lane δ_t . The natural extension to the lane change program from the trailing program would be to model collision avoidance of all vehicles via ramp barriers in the temporal domain, as is illustrated in Fig. 3b. However, as can be seen in Fig. 3b, modeling the collision avoidance of the ramp barrier cuts out the half-plane of the leading vehicle, prohibiting the ego vehicle from passing. This might lead to unnecessary deceleration of the ego vehicle, if the leading vehicle is traveling slower than the reference speed of the ego vehicle. This deceleration may render the lane change maneuver a less viable option for the decision manager, which will decide to trail instead. To resolve this, collision avoidance of the leading vehicle is modeled as a ramp barrier if the speed of the leading vehicle is higher than the reference speed of the ego vehicle. On the other hand, if the speed of the leading vehicle is lower than the reference speed of the ego vehicle, the leading vehicle is modeled using a rectangular critical zone, see Fig. 3c. To preserve the program structure (1) when modeling with a critical zone sampling is done in the relative spatial domain and the inverse of the relative velocity between the ego and leading vehicle is used as a state. This prevents the use of the spatially-defined overtaking when the ego and leading vehicle are driving with the same speed.

For the temporally defined MPC, i.e., when $v_L > v_r$ with the leading vehicle collision avoidance modeled as a ramp barrier, the data \mathbf{x}^{lt} , \mathbf{u}^{lt} , \mathbf{x}_r^{lt} , Q^{lt} , R^{lt} , S^{lt} , A^{lt} , B^{lt} , $\mathbf{x}_{\min,k}^{\text{lt}}$, $\mathbf{x}_{\max,k}^{\text{lt}}$, $\mathbf{u}_{\min,k}^{\text{lt}}$ and $\mathbf{u}_{\max,k}^{\text{lt}}$, is identical to the trailing data presented in (2)-(12), with the only difference that the lateral reference position is now in the adjacent lane. The matrices of the general linear constraints (1e) can be written as

$$C_k^{\text{lt}} = \begin{bmatrix} 0 & 0 & s & 1 \\ 0 & 0 & s & -1 \\ 1/l_{\text{Lx}} & \mp 1/l_{\text{Ly}} & 0 & 0 \\ 1/l_{\text{Fx}} & \pm 1/l_{\text{Fy}} & 0 & 0 \\ -1/l_{\text{Rx}} & \pm 1/l_{\text{Ry}} & 0 & 0 \end{bmatrix}, D_k^{\text{lt}} = \mathbf{0},$$

while the right hand side is given by

$$g_k^{\text{lt}} = [0, 0, -1 + x_L(t)/l_{\text{Lx}} \mp y_L(t)/l_{\text{Ly}}, \\ -1 + x_F(t)/l_{\text{Fx}} \pm y_F(t)/l_{\text{Fy}}, \\ -1 - x_R(t)/l_{\text{Rx}} \pm y_R(t)/l_{\text{Ry}}]^T.$$

where $x_R(t)$ and $x_F(t)$ are the longitudinal positions of the adjacent vehicles R and F, respectively, and $y_R(t)$ and $y_F(t)$ are the corresponding lateral positions. The top signs in g_k^{lt} and C_k^{lt} are used if lane change is done from the right lane to the left, and the lower sign is used if the lane change is from left to right. For detailed, derivation of this optimization program, see [11]. This temporal lane change program is a convex QP.

When $v_L \leq v_r$, the collision avoidance with respect to the leading vehicle is achieved using a rectangular zone instead of a ramp barrier, as in Fig. 3c. If modeling was to be done in the temporal domain we would end up with a mixed integer formulation, since the time we reach the rectangular zone is unknown (due to the fact that the position and speed of the ego vehicle are decision variables). However, as mixed-integer programs are computationally difficult to solve, we apply the method proposed in [10], that transforms the program to a smooth sequential QP that can be solved efficiently. The method involves a change of independent variables from time t to relative distance $\tilde{x} = x - (v_{L0} + tv_L)$ where v_{L0} denotes the initial speed of the leading vehicle. Further, the state and control vectors are replaced by

$$\hat{\mathbf{x}}_k^{\text{ls}}(\tilde{x}) = [\tilde{t}_k(\tilde{x}), y_k(\tilde{x}), z_k(\tilde{x}), \tilde{v}_{y,k}(\tilde{x})]^T, \\ \mathbf{u}_k^{\text{ls}}(\tilde{x}) = [u_{x,k}(\tilde{x}), u_{y,k}(\tilde{x})]^T,$$

where $\tilde{v}_{x,k} = v_{x,k}(\tilde{x}) - v_{L,k}(\tilde{x})$, $z_k(\tilde{x}) = \tilde{v}_{x,k}^{-1}$ is the lethargy, $\tilde{t}_k(\tilde{x})$ is the travelling time fulfilling $\tilde{t}'_k(\tilde{x}) = z_k(\tilde{x})$, $\tilde{v}_{y,k}(\tilde{x}) = v_{y,k}(\tilde{x})\tilde{v}_{x,k}(\tilde{x})$, $u_{x,k}(\tilde{x}) = -z_k^3(\tilde{x})\tilde{a}_{x,k}(\tilde{x})$ and $u_{y,k}(\tilde{x}) = a_{y,k}(\tilde{x})z_k^2(\tilde{x}) - v_{y,k}(\tilde{x})z_k^3(\tilde{x})\tilde{a}_{x,k}(\tilde{x})$. This gives $A^{\text{ls}} = A^{\text{t}}$ and $B^{\text{ls}} = B^{\text{t}}$. Further, the box constraints are given by

$$\mathbf{x}_{\min,k}^{\text{ls}}(\tilde{x}) = [0, y_k^{\min}(\tilde{x}), (v_{x,k}^{\max} - v_L(\tilde{x}))^{-1}, \text{free}]^T, \\ \mathbf{x}_{\max,k}^{\text{ls}}(\tilde{x}) = [\text{free}, y_k^{\max}(\tilde{x}), 1/\epsilon, \text{free}]^T,$$

where ϵ is a small positive number, representing a non-zero

minimum speed of the ego vehicle. Further, we have

$$\mathbf{x}_r^{\text{ls}} = [0, y_r^{\text{ls}}(\tilde{x}), z_r^{\text{ls}}(\tilde{x}), 0], \\ Q^{\text{ls}} = \text{diag}(0, q_z^{\text{ls}}, q_y^{\text{ls}}(\tilde{x}), 0), \\ R^{\text{ls}} = \text{diag}(r_z^{\text{ls}}, r_y^{\text{ls}}), \\ S^{\text{ls}} = \text{diag}(s_z^{\text{ls}}, s_y^{\text{ls}}),$$

where $z_r(\tilde{x}) = 1/v_r(\tilde{x})$. The general linear constraint matrices read

$$C_k^{\text{ls}} = \begin{bmatrix} 0 & 0 & -sv_{\text{Lx}}(\tilde{x}) & 1 \\ 0 & 0 & -sv_{\text{Lx}}(\tilde{x}) & -1 \\ 0 & 0 & 3z_r^2 a_{x,k}^{\min} & 0 \\ 0 & 0 & -3z_r^2 a_{x,k}^{\max} & 0 \\ 0 & 0 & 2z_r a_{y,k}^{\min} & 0 \\ 0 & 0 & -2z_r a_{y,k}^{\max} & 0 \\ \frac{v_L - v_F}{l_{\text{Fx}}} & \pm \frac{1}{w_1} & 0 & 0 \\ -\frac{v_L - v_R}{l_{\text{Rx}}} & \pm \frac{1}{w_1} & 0 & 0 \end{bmatrix}, \\ D_k^{\text{ls}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -1 & 0 \\ 1 & 0 \\ 0 & -1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, g_k^{\text{ls}} = \begin{bmatrix} s \\ s \\ 2z_r^3 a_{x,k}^{\min} \\ -2z_r^3 a_{x,k}^{\max} \\ z_r^2 a_{y,k}^{\min} \\ -z_r^2 a_{y,k}^{\max} \\ -1 + \frac{x_F(\tilde{x}) - \tilde{x}}{l_{\text{Fx}}} \pm \frac{y_F(\tilde{x})}{l_{\text{Fy}}} \\ -1 - \frac{x_R(\tilde{x}) - \tilde{x}}{l_{\text{Rx}}} \pm \frac{y_R(\tilde{x})}{l_{\text{Ry}}} \end{bmatrix},$$

The spatial formulation does not include box constraints on the input, since these have been moved to the general constraints (1e), due to their dependence on the lethargy $z_k(\tilde{x})$. Further, the box constraints on the input depended nonlinearly on $z_k(\tilde{x})$, so to write the constraints on the form in (1e), they have been linearized around a reference trajectory $z_r(\tilde{x})$. Thus, a nonlinear program has been cast into a sequential smooth quadratic program which, to remove linearization errors, can be solved using sequential quadratic programming (SQP), [12]. The first two rows of the general constraints (1e) define the slip constraints, row three to six are the transformed box constraints on the inputs and the last two rows are the ramp barriers for the adjacent vehicles. The collision avoidance of the leading vehicle is modeled in y_k^{\min} , as can be seen in 3c. For mathematical details, see for example [10].

Given a solution for the spatially-defined MPC, the temporally defined state and control values, corresponding to (2) and (3), can be recovered via

$$\begin{bmatrix} x_k(\tilde{x}) \\ y_k(\tilde{x}) \\ v_{x,k}(\tilde{x}) \\ v_{y,k}(\tilde{x}) \end{bmatrix} = \begin{bmatrix} \tilde{x} + v_L t \\ y_k(\tilde{x}) \\ \frac{1}{z_k(\tilde{x})} + v_L(\tilde{x}) \\ \tilde{v}_{y,k}(\tilde{x})\tilde{v}_{x,k}(\tilde{x}) \end{bmatrix}, \\ \begin{bmatrix} a_{x,k}(\tilde{x}) \\ a_{y,k}(\tilde{x}) \end{bmatrix} = \begin{bmatrix} \frac{u_{x,k}(\tilde{x})}{z_k^3(\tilde{x})} \\ \frac{u_{y,k}(\tilde{x})}{z_k^2(\tilde{x})} + v_{y,k}(\tilde{x})a_{x,k}(\tilde{x})z_k^2(\tilde{x}) \end{bmatrix},$$

and then sampled with the same time interval as the trailing program, using interpolation.

V. DECISION MANAGER

The trajectory planner deliver two solutions: the trailing solution, Section IV-A, $(\mathbf{x}_k^t, \mathbf{u}_k^t)$ given the reference trajectory x_r^t and the lane change solution, Section IV-B, $(\mathbf{x}_k^l, \mathbf{u}_k^l)$ given the trajectory x_r^l . The decision manager selects the one which gives best performance based on four factors. comfort, reference tracking, distance to the (potential) exit and avoidance of frequent switching between trailing and lane change. This is done via the criterion

$$J(\cdot) = q_c J_c(\cdot) + q_e J_e(\cdot) + q_s J_s(\cdot), \quad (15)$$

where q_c , q_e and q_s are positive weights which determine how important the different factors are, [11].

The first two factors, comfort and reference tracking, is evaluated via the first term of (15), which is

$$J_c(\mathbf{x}, \mathbf{x}_r, \mathbf{u}) = \int_{t_0}^{t_f} \|\mathbf{x}(t) - \mathbf{x}_r(t)\|_{Q^c}^2 + \|\mathbf{u}(t)\|_{R^c}^2 dt,$$

where Q^c and R^c are weighting matrices with the same sparsity pattern as the trailing matrices in (6) and (7).

The second term of (15) considers the distance to the exit, using the function

$$J_e = \begin{cases} (1 - (\frac{d_e - x_k(t)}{d_{\max}})^\alpha) |\delta_i(y) - \delta_e|, & x_k(t) \in d_e - [0, d_{\max}] \\ 0, & \text{otherwise} \end{cases}$$

where d_e is the distance from the initial position of the ego vehicle to the exit, d_{\max} is the largest distance at which the exit affects the decision making, α is a real number between 0 and 1, and δ_e is the lane containing the exit. The target lane $\delta_i(y)$ for the trajectory currently being evaluated can be found via

$$\delta_i(y) = \text{floor} \left(\frac{y(t_f)}{w_1} \right) + 1 = \begin{cases} 1, & 0 \leq y(t_f) < w_1, \\ 2, & w_1 \leq y(t_f) \leq 2w_1, \end{cases}$$

where 1 is the right lane, 2 is the left lane and floor is an operator that rounds a number down to its nearest integer.

Lastly, the third term of (15) is aimed at avoiding frequent changes in decisions between trailing and lane change in the following way

$$J_s = \sum_{k=1}^n \rho^k |\delta_{\text{prev}}(k) - \delta_0|,$$

where $\delta_{\text{prev}}(k)$ is the decision made k steps ago, n is the number of previous decisions considered and δ_0 is the decision currently being evaluated. Hence, the decision is the solution to

$$\begin{aligned} (\mathbf{x}_k^O, \mathbf{u}_k^O) = \\ = \text{argmin}(J(\mathbf{x}_k^t, \mathbf{u}_k^t, \mathbf{x}_r^t, \delta_{\text{current}}^t), J(\mathbf{x}_k^l, \mathbf{u}_k^l, \mathbf{x}_r^l, \delta_{\text{current}}^l)) \end{aligned}$$

If the trailing or lane change program is infeasible the vectors $\mathbf{x}_k^t, \mathbf{u}_k^t$ or $\mathbf{x}_k^l, \mathbf{u}_k^l$ do not exist. When a solution does not exist, the decision manager assigns it a value of infinity to make sure the feasible solution is chosen. However, if the trailing and lane change programs are solved using hard constraints both problems might be infeasible at the same time. In this

case one could use a heuristic backup controller to make sure that there is always a solution available. As an alternative, one could formulate the trailing and lane change problems using soft constraints, this would guarantee feasibility but introduce additional tuning parameters.

VI. CASE STUDY

In this case study the vehicle model is the same point mass model as we used in the optimal control programs, i.e., there is no modeling error between the vehicle model and the modeling in the optimization program, so the output of the controller can be directly fed to the vehicle. Further, it is assumed that the sensors provide perfect readings and the predictions are calculated via the deterministic equations

$$\begin{aligned} x_i(\tau) &= x_{i,0} + v_{i,0}\tau, \\ v_i(\tau) &= v_{i,0} + a_{i,0}\tau, \end{aligned}$$

for all $i = \{E, L, F, R\}$ and all $\tau_0 \leq \tau \leq \tau_f$ in each MPC-cycle.

The architecture is fully implemented in MATLAB and the optimization control programs are transferred into a discrete form using first order Euler approximation with a sampling interval of 0.1s and horizon $t_f = 10$ s for the temporal programs, and a sampling interval of 1m and horizon of $\tilde{x}_f = 50$ m for the spatial program. The discretized problems are then solved using the solver HPIPM [13] on a laptop with Intel Core i7-5600 CPU at 2.60 GHZ with 16 GB RAM. New trajectories are planned each 1s and the simulation is stopped after 250s.

The architecture was evaluated on three different scenarios. In all three scenarios $v_r = 80$ km/h, the ego vehicle is heading for an exit in the left lane while starting in the right lane. The rest of the problem data that is identical for all three scenarios have been gathered in Table I.

A. Scenario I

In the first scenario the ego vehicle is driving behind a slow moving leading vehicle, with a constant velocity of $v_L = 60$ km/h, in the presence of two surrounding vehicles, maintaining the speed $v_F = v_R = v_r$ throughout the simulation and the distance to the exit is $d_e = 1100$ m.

The results for scenarios I can be seen in the first column of Fig. 4, where the top plot depicts the position of all vehicles, the middle plot the longitudinal and lateral velocity of the ego vehicle, along with its limits and the bottom plot depicts the target lane of the controller in each MPC update. The thin lines in Figs. 4a depict the trajectories chosen by the decision manager in each MPC iteration. The thin red lines indicate candidate trajectories that end up in the right lane and black thin lines represent candidate trajectories that end up in the left lane. It can be seen that lane change is done as soon as possible even though the distance to the exit is large. This is because it is more beneficial to drive between the fast-moving vehicles in the adjacent lane instead of behind the slow-moving leading vehicle.

TABLE I: Problem data common for the three scenarios

$l_R = 4.5$ m	$l_F = 4.5$ m	$l_L = 4.5$ m
$l_E = 12$ m	$\theta_L = 1$ s	$\tilde{\theta}_L = 0.5$ s
$\theta_F = 0.5$ s	$\tilde{\theta}_F = 0.5$ s	$\theta_R = 0.5$ s
$\tilde{\theta}_R = 0.5$ s	$l_{Ly} = 2$ m	$l_{Ry} = 2$ m
$l_{Fy} = 2$ m	$w_E = 2.55$ m	$w = 1.275$ m
$w_l = 3.2$ m	$v_r = 80$ km/h	$v_{x,k}^{\min} = 0$ km/h
$v_{x,k}^{\max} = 90$ km/h	$v_{y,k}^{\min} = -4$ m/s	$v_{y,k}^{\max} = 4$ m/s
$a_{x,k}^{\min} = -4$ m/s ²	$a_{x,k}^{\max} = 4$ m/s ²	$a_{y,k}^{\min} = -1$ m/s ²
$a_{y,k}^{\max} = 1$ m/s ²	$\theta_t = 0.25$ s	$Q^t = \text{Diag}(0, 2, 1, 4)$
$R^t = \text{Diag}(4, 4)$	$Q^{lt} = Q^t$	$R^{lt} = R^t$
$Q^{ls} = \text{Diag}(0, 2, 4, 4)$	$R^{ls} = \text{Diag}(3, 4)$	$Q^c = Q^t$
$R^c = R^t$	$q_c = 1$	$q_e = 600$
$q_s = 30$	$s = 0.18$	$t_f = 10$ s
$\tilde{x}_f = 50$ m	$x_{0,k} = 45$ m	$v_{x0,k} = v_r$
$v_{L0} = 50$ km/h	$v_F = v_r$	$v_R = v_r$

B. Scenario II

The second scenario is similar to the first, the only difference being that the leading vehicle is driving with $v_L = 70$ km/h and the adjacent vehicles are decelerating from their initial velocity v_r to 60 km/h

The results for scenario II can be seen in the second column of Fig. 4. In this case, the lane change is started significantly later (about 180 m later) than in Scenario I. This is expected since it is no longer beneficial to drive between the two adjacent vehicles, which are slowing down. Therefore, the architecture decides to instead stay in the right lane, see Fig. 4h, and trail the leading vehicle until it has passed the decelerating adjacent vehicle before performing the lane change, allowing the ego vehicle to keep a longitudinal velocity close to the reference, see Fig. 4e.

C. Scenario III

The third scenario is identical to the second, except that the distance to the exit is now 500 m instead of 1100 m.

In contrast to Scenario II, the ego vehicle is now so close to the exit that the decision manager prompts an early lane change, see 4c, and drive between the decelerating adjacent vehicles to minimize the risk of missing the exit. However, this yield a significant drop in the longitudinal velocity as can be seen in Fig. 4f.

VII. CONCLUSIONS

In this paper we have presented an architecture for controlling an autonomous vehicle near a highway exit. As part of the control architecture, a trajectory planner is proposed where two model predictive controllers are evaluated, one which suggests an optimal lane change trajectory and one which suggests an optimal trailing trajectory. Based on safety, comfort, reference tracking and distance to the exit, a decision manager chooses one of the optimal trajectories to be sent to the vehicle. This approach allows for lane changes, simple trailing (similar to adaptive cruise control) and abortion of initialized lane changes. Further, two versions of the lane change program are implemented. The first is in the temporal domain and is applied when the leading vehicle is driving faster than the desired velocity of the ego vehicle.

The second is in the spatial domain and is applied when the leading vehicle is driving slower than the desired velocity of the ego vehicle. This approach combines the strengths of these two approaches, allowing us to efficiently compute optimal lane change trajectories for all speeds of the ego and leading vehicle. A case study was made that showed that both the trailing and lane change before and after decision making provide smooth trajectories and that the decision manager adjusts its decisions based on factors such as distance to the exit and velocity changes of the surrounding vehicles. How to weight the different factors in the decision making is an open question that could be subject to future research. An idea would be to try and fine tunings corresponding to different traffic rules.

REFERENCES

- [1] S. Singh, "Critical reasons for crashes investigated in the national motor vehicle crash causation survey," <http://www-nrd.nhtsa.dot.gov/Pubs/812115.pdf>, U.S. Department of Transportation: National Highway Traffic Safety Administration, Tech. Rep., 2015.
- [2] P. Thomas, A. Morris, R. Talbot, and H. Fagerlind, "Identifying the causes of road crashes in europe," in *Annals of advances in automotive medicine*, 2015, pp. 13–22.
- [3] V. Milanés, S. E. Shladover, J. Spring, Christopher, H. Kawazoe, and M. Nakamura, "Cooperative adaptive cruise control in real traffic situations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 296–305, 2013.
- [4] A. Vahidi and A. Eskandarian, "Research advances in intelligent collision avoidance and adaptive cruise control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, no. 3, pp. 143–153, 2003.
- [5] J. Pohl, W. Birk, and L. Westervall, "A driver-distraction-based lane-keeping assistance system," *J. Systems and Control Engineering*, vol. 221, pp. 541–552, 2007.
- [6] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, and C. Stiller, "Three decades of driving assistance systems: Review and future perspective," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 4, pp. 6–22, 2014.
- [7] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, "Towards fully autonomous driving: Systems and algorithms," in *IEEE Intelligent vehicle symposium*, Baden-Baden, Germany, 2011.
- [8] M. Campbell, M. Egerstedt, J. P. How, and R. M. Murray, "Autonomous driving in urban environments: approaches, lessons and challenges," *Philosophical transactions of the royal society*, vol. 368, no. 1928, pp. 4649–4672, 2010.
- [9] P. F. Julia Nilsson, Mohammed Ali and J. Sjöberg, "Predictive cruise control with autonomous overtaking," in *Intelligent Transportation Systems (ITSC)*, The Hague, Netherlands, 2013.
- [10] N. Murgovski and J. Sjöberg, "Predictive cruise control with autonomous overtaking," in *Decision and Control (CDC)*, Osaka, Japan, 2015.
- [11] G. Dovner, "Path planning and decision making in a highway exit situation," Master's thesis, Chalmers University of Technology, Sweden, 2018.
- [12] J. Nocedal and S. J. Wright, *Numerical optimization*, 2nd ed. Springer, 2000.
- [13] G. Frison, "Algorithms and methods for fast model predictive control," 2016, script of talk.

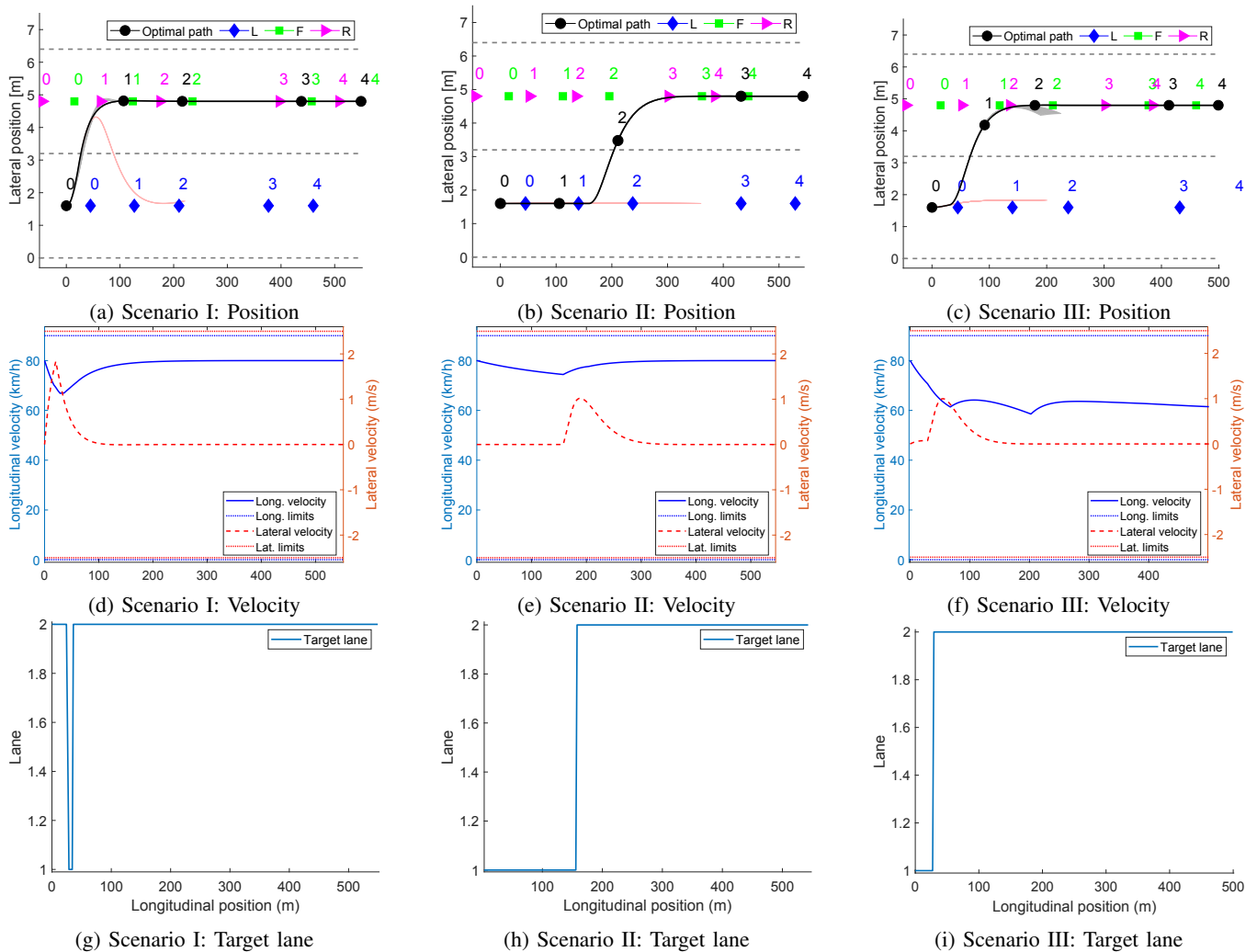


Fig. 4: a-c) The resulting position, velocity and target lane for scenario I. d-f) The resulting position, velocity and target lane for scenario II. g-i) The resulting position, velocity and target lane for scenario III. The thick black line in the position plots in the first row depict the position of the ego vehicle. The thin lines depict the trajectories chosen by the decision manager, in each MPC update. The plots in third row depict the target lane the architecture is aiming for at each MPC update.