MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Automated Driving: Safe Motion Planning Using Positively Invariant Sets

Berntorp, K.; Weiss, A.; Danielson, C.; Di Cairano, S.

TR2017-130 October 2017

Abstract

This paper develops a method for safe lane changes. We leverage feedback control and constraint-admissible positively invariant sets to guarantee collision-free closed-loop trajectory tracking. Starting from an initial state of the vehicle and obstacles in the region of interest, our method steers the vehicle to the desired lane while satisfying constraints associated with the future motion of the obstacles with respect to the vehicle. We connect the initial state with the desired lane using equilibrium points and associated positively invariant sets of the vehicle dynamics, where the positively invariant sets are used to guarantee safe transitions between the equilibrium points. An autonomous highway-driving example with a receding-horizon implementation shows that our method is capable of generating safe dynamically feasible trajectories in real-time while accounting for obstacles in the environment and modeling errors.

International IEEE Conference on Intelligent Transportation Systems (ITSC)

Automated Driving: Safe Motion Planning Using Positively Invariant Sets

Karl Berntorp, Avishai Weiss, Claus Danielson, Ilya V. Kolmanovsky, and Stefano Di Cairano

***Abstract*—This paper develops a method for safe lane changes. We leverage feedback control and constraint-admissible positively invariant sets to guarantee collision-free closed-loop trajectory tracking. Starting from an initial state of the vehicle and obstacles in the region of interest, our method steers the vehicle to the desired lane while satisfying constraints associated with the future motion of the obstacles with respect to the vehicle. We connect the initial state with the desired lane using equilibrium points and associated positively invariant sets of the vehicle dynamics, where the positively invariant sets are used to guarantee safe transitions between the equilibrium points. An autonomous highway-driving example with a receding-horizon implementation shows that our method is capable of generating safe dynamically feasible trajectories in real-time while accounting for obstacles in the environment and modeling errors.**

I. INTRODUCTION

Autonomous vehicles are complex decision-making systems that integrate advanced and interconnected sensing and control components. While autonomous vehicles increasingly begin testing on public roads, production vehicles are more commonly being equipped with advanced driver-assistance systems (ADAS) such as adaptive cruise control and lane-change assist. This is driven by both safety and economic aspects such as the high number of traffic accidents associated with overtaking and lane-change maneuvers and potential fuel savings [1].

A fundamental component in the control and guidance layer of an autonomous vehicle control system or in a sophisticated ADAS is the motion planner, which produces a desired trajectory that the vehicle should follow based on the output from a sensing & mapping module. Autonomous vehicle control is commonly divided into trajectory generation (motion planning) and trajectory tracking (vehicle control) [2]. Trajectory generation is often performed using either sampling-based methods such as rapidly-exploring random trees (RRTs) [3], [4] or graph-search methods [5]. Trajectory tracking in automotive is frequently done by leveraging classical control methods, for example, PID control, or more advanced methods such as model-predictive control (MPC) [6]. Integrating the planning and control into a single module has also been considered [7], however, in such a case a more complex and difficult problem needs to be solved. Decoupling is appealing as it simplifies the problem and makes it more tractable. This is the dominant approach in the robotics community [8] and is also used in automotive applications [9], [10]. However, the time scales, dynamics, and stringent performance and driving requirements that are present in automotive systems motivate to pursue a more integrated approach to planning and control than in traditional robotics. Thus, an important question is how to connect the motion planning and vehicle control to ensure performance and safety of the vehicle [6], [11]. There are methods for unifying motion planning and real-time vehicle control into one module [7], [12]–[14]. However, this often leads to nonconvex problems [13].

This paper develops a method for motion planning and tracking which enables overtaking and lane-change maneuvers. Our method steers the vehicle to the desired lane using state-feedback controllers, while satisfying input and state constraints associated with the future motion of the obstacles with respect to the vehicle. Our method uses a graph search over a finite set of lateral displacements on the road to find a path from initial state to desired lane, where a constraint admissible positively invariant set is associated with each lateral displacement. The positively invariant sets determine a safe trajectory to reach the desired lane, thereby integrating motion planning and vehicle control in a systematic manner.

We leverage a similar method as in [15], [16], which addressed constrained spacecraft relative motion control problems. We extend the applicability of that approach to road vehicles on multi-lane roads including moving obstacles. We formulate the planning and tracking problem in the error coordinates of the vehicle with respect to the road-aligned coordinate frame. We therefore reduce the dimensionality of the graph-search problem such that computational times become suited for real-time execution. Our results demonstrate the ability to perform efficient and safe maneuvers with this approach. Compared to methods for lane-change maneuvers based on MPC (e.g., [12]–[14]), our method does most computations offline. MPC relies on solving constrained optimization problems in real time, whereas our approach solves a low-dimensional graph-search problem and uses a computationally inexpensive unconstrained linear quadratic controller. We exploit a receding-horizon implementation, which provides feedback both in planning stage and in the vehicle-control stage; obstacle avoidance and constraint-satisfaction are accounted for already at the planning stage and the state-feedback control takes care of the remaining modeling errors.

II. MODELING AND PROBLEM STATEMENT

We refer to the automated vehicle as the ego vehicle (EV), whereas other moving entities in the region of interest (ROI) of the EV are designated as other vehicles (OV). Note that

The first three and the fifth author are with Mitsubishi Electric Research Laboratories (MERL), 02139 Cambridge, MA, USA. Ilya V. Kolmanovsky is with the University of Michigan, Ann Arbor, MI, USA. Email: karl.o.berntorp@ieee.org

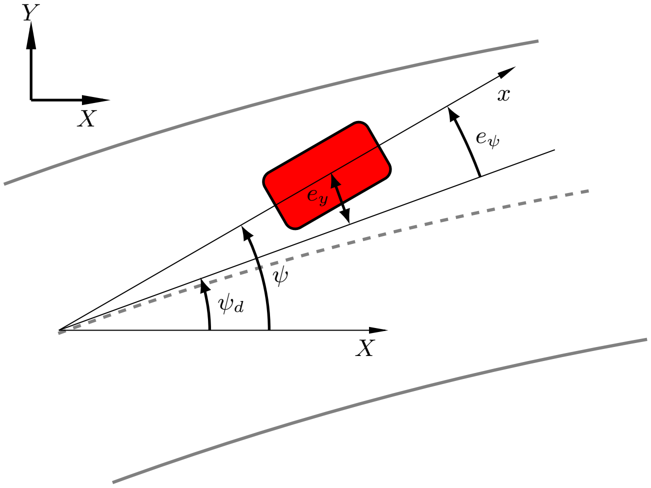

Fig. 1. Definition of the coordinate frames and related notation. The origin in the road-aligned frame is located at a point in the middle of the road corresponding to the EV at the start of determining the motion plan.

the OVs can be either in autonomous or manual mode. The modeling of the EV is done in the local error coordinates with respect to a road-aligned frame (Fig. 1), with the origin at the start of each planning step fixed to the middle of the road. We introduce the following assumption.

Assumption 1: Positions and velocities of the OVs relative to the EV at the current time are known.

In practice these can be measured and estimated by cameras, lidars, radars, and/or ultrasound sensors attached to the EV. Note that the future states of the OVs over the planning horizon are not assumed to be known a priori in the method we propose in this paper. We introduce the following assumption on the driving behavior.

Assumption 2: The driving behavior is typical of highway driving. The velocity is approximately constant over the planning horizon and the steering angle adheres to $\sin \delta \approx \delta$.

According to Assumption 2, emergency braking and/or aggressive evasive maneuvers are handled by a different layer in the decision logic. Motivated by this, we focus on the lateral dynamics of the vehicle, which we model using a planar single-track model with lumped right and left wheels on each axle, where the lateral tire forces are modeled by the linear approximations $F_f \approx C_f \alpha_f$, $F_r \approx C_r \alpha_r$, and C_f, C_r are the front and rear lateral tire stiffness coefficients. Under assumption 2 on moderate steering the slip angles α_f, α_r can be approximated as [17] $\alpha_f \approx \delta - \frac{v_y + l_f \dot{\psi}}{v_x}$, $\alpha_r \approx \frac{l_r \dot{\psi} - v_y}{v_x}$, where δ is the steering angle of the front wheel, which is the control input, v_x and v_y are the longitudinal and lateral velocity of the vehicle, respectively, $\dot{\psi}$ is the yaw rate, and l_f, l_r are the distances from the center of mass to the front and rear wheel base. We introduce the state vector

$$\boldsymbol{x} = \begin{bmatrix} e_y & \dot{e}_y & e_\psi & \dot{e}_\psi \end{bmatrix}^{\mathrm{T}}, \tag{1}$$

where e_y and $e_\psi = \psi - \psi_d$ denote the lateral position and vehicle orientation, respectively, in the road-aligned coordinate frame, and ψ_d is the angle of the tangent of the road with respect to the inertial frame, as defined in Fig. 1.

The vehicle model in the error coordinates (1) can be written on the form [18], [19]

$$\dot{\boldsymbol{x}} = \boldsymbol{A}_e \boldsymbol{x} + \boldsymbol{B}_e \delta + \boldsymbol{D}_e \dot{\psi}_d. \tag{2}$$

Remark 1: The term $\dot{\psi}_d$ acts as a disturbance on the vehicle dynamics and arises because we model the vehicle motion in the noninertial road-aligned frame. The model is consistent with [14], [18], but is an approximation because we ignore higher-order effects. We consider lane-change maneuvers with moderate steering and approximately constant velocity over the planning horizon, where this approximation is reasonable. However, for high-performance maneuvering, such as during emergency collision avoidance, higher-order terms may be needed. Note that $\dot{\psi}_d$ is not considered known, but is estimated during driving (see Sec. III-C).

A. Problem Statement

This paper focuses on generating safe overtaking and lane-change maneuvers for normal driving scenarios. We consider a discrete-time linear model in the form

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\delta_k + \boldsymbol{D}d_k, \tag{3a}$$

$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k, \tag{3b}$$

where k is the time index corresponding to time t_k and $d = \dot{\psi}_d$ is the disturbance disturbance term. We obtain (3a) by discretizing the continuous-time system (2) using a zero-order hold on the input with sampling time Δt. Eq. (3b) models the outputs of the system (3a) that we wish to plan a trajectory for and, subsequently, control. The input δ is subject to constraints $\delta_{\min} \leq \delta_k \leq \delta_{\max}$. These constraints are determined by the physical limitations of the vehicle (i.e., maximum steering angle) or induced by ensuring that the assumptions made for deriving (2) hold. For instance, the linear single-track model is valid for lateral accelerations up to approximately $0.4g$ on dry asphalt, where g is the gravitational acceleration. Hence, limitations on δ can be set by the relation to $\dot{\psi}$ for steady-state cornering [18].

The output $\boldsymbol{y}_k$ is constrained as $\boldsymbol{y}_k \in \mathcal{Y}_k \in \mathbb{R}^m$, where the output set $\mathcal{Y}_k$ can be time-varying and is determined from different constraints. The road boundaries in the road-aligned frame impose constraints $-e_{y,\max} \leq e_{y,k} \leq e_{y,\max}$ on the lateral position of the EV. The term $\dot{\psi}_d$ associated with the curvature of the road in the global frame, together with bounds on the allowed lateral accelerations, gives the constraints $\dot{e}_{\psi,\min} \leq \dot{e}_{\psi,k} \leq \dot{e}_{\psi,\max}$. Limitations on local lateral velocity error can also be set, leading to bounds $\dot{e}_{y,\min} \leq \dot{e}_{y,k} \leq \dot{e}_{y,\max}$. The constraints can compactly be written as

$$\mathcal{Y}_k = \{\boldsymbol{y}_k : \boldsymbol{H}_k \boldsymbol{y}_k \leq \boldsymbol{K}_k\} \tag{4}$$

for appropriately defined matrices $\boldsymbol{H}_k$ and $\boldsymbol{K}_k$. In this paper, (4) is determined offline, hence do not depend on time.

If the motion of the OVs is estimated by means of statistical methods, a natural choice is to model the OVs as ellipsoids. Alternatively, the vehicles can be modeled as rectangular, which is rather common [13]. The positions and velocities of the OVs relative to the EV inform additional time-varying constraints on the outputs of the EV. Let $y_{\mathrm{ov},T_{p1}}$ denote the predicted lateral position of an OV relative to the EV at prediction time T_{p1} and introduce the safety time t_s. Then, the EV is prevented to enter the critical zone $[y_{\mathrm{ov},T_{p1}} - w, y_{\mathrm{ov},T_{p1}} + w]$ in the time interval $[T_{p1} - t_s, T_{p1} + t_s]$, where w denotes a lateral safety margin. The safety time t_s is introduced to account for sensing and estimation errors with respect to the OVs. The predicted sets $\mathcal{S}_{k,i}$ of the ith of M OVs over the planning horizon $[t_k, t_k + T_p]$ are denoted with

$$\mathcal{S}_k = \bigcup_{i=1}^{M} \mathcal{S}_{k,i}(t_k, t_k + T_f), \tag{5}$$

where $\mathcal{S}_k$ is the combination of all obstacles over the time horizon T_f over which the motion plan is to be determined, which can be constrained by the range of the sensors (i.e., the ROI). Hence, given the vehicle dynamics (3), the goal is to generate an input trajectory satisfying the input constraints that when applied to (3) leads to a trajectory that satisfies (4) and avoids the obstacle set (5) for all $T_p \in [t_k, t_k + T_f]$.

III. Safe Motion Planning Using Positively Invariant Sets

In this section we describe our method for solving the motion-planning problem for safe overtaking and lane-change maneuvers. Our focus is real-time motion planning when computing capabilities are scarce. This implies that the main objective is to quickly find smooth, drivable motion profiles that avoid collision, rather than searching for the optimal one.

The main idea of the method is that we determine regions on the road where it is safe to travel, each associated with the controller that makes such region invariant. Then we compute the trajectory to navigate the road by means of graph search to find a safe path through the regions with associated tracking control for computing the trajectories.

A. Feedback Control and Positively Invariant Sets

Fig. 2 shows the proposed control architecture. The motion planner uses the steering system as the actuator. The velocity control is handled outside of the motion planner, thus in effect decoupling velocity and steering control. This is rather common (see, e.g., [6], [14]) and is reasonable for maneuvers with moderate steering and acceleration. The upper-most block in Fig. 2 generates the commanded longitudinal velocity profile to control the vehicle velocity, sent to the actuator control block and the motion-planning block.

The motion planner integrates trajectory generation and trajectory tracking by exploiting state-feedback steering controllers of the form

$$\delta_k = -\boldsymbol{L}(\boldsymbol{r}_i - \boldsymbol{x}_k) + \delta_k^{ff}, \tag{6}$$

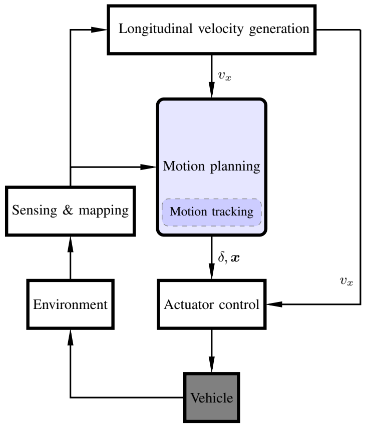

Fig. 2. Structure of the control architecture.

where $\boldsymbol{L}$ is the feedback gain and $\boldsymbol{r}_i$ is the setpoint that the vehicle should reach. The term δ_k^{ff} is a feedforward term that corrects for the disturbance d_k due to the curvature of the road. The setpoint location $\boldsymbol{r}_i$ is chosen to be an equilibrium point of the system (3) that satisfies $\bar{\boldsymbol{y}} \in \mathcal{Y}_k$. With the feedforward term δ_k^{ff}, from (3a) it follows that an equilibrium point satisfies $\bar{\boldsymbol{x}} = [\bar{e}_y \ 0 \ \bar{e}_\psi \ 0]^{\mathrm{T}}$, which means that an equilibrium point satisfies (4). We have a collection of setpoints $\{\boldsymbol{r}_i\}_{i=1}^{n_r}$,

$$\boldsymbol{r}_i = \bar{\boldsymbol{x}}_i, \quad i = 1, \dots, n_r, \tag{7}$$

where the middle of the lanes are included in (7). We design the feedback gain $\boldsymbol{L}$ such that the closed-loop system, when inserting (6) into (3a) is asymptotically stable. Since the control (6) asymptotically stabilizes the vehicle with the model (3a), at the target equilibrium point corresponding to $\boldsymbol{r}_i$ we can associate a quadratic Lyapunov function with $\boldsymbol{L}$,

$$V(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{r}_i)^{\mathrm{T}} \boldsymbol{P} (\boldsymbol{x} - \boldsymbol{r}_i) \tag{8}$$

where $\boldsymbol{P}$ is a symmetric matrix that fulfills $\bar{\boldsymbol{A}}^{\mathrm{T}} \boldsymbol{P} \bar{\boldsymbol{A}} - \boldsymbol{P} < \boldsymbol{0}$, and where $\bar{\boldsymbol{A}} = \boldsymbol{A} - \boldsymbol{B}\boldsymbol{L}$. Each level set of the Lyapunov function (8) defines a positively invariant set; that is, a set that guarantees that whenever the initial condition is in this set, the ensuing trajectory will remain in that set. We rely on positively invariant sets to guarantee that the closed-loop trajectory satisfies the input and output constraints. For a setpoint $\boldsymbol{r}_i$ and a Lyapunov function (8) associated with the feedback gain $\boldsymbol{L}$, a positively invariant set $\mathcal{O}_i$ is the set

$$\mathcal{O}_i = \{\boldsymbol{x} \in \mathbb{R}^4 : (\boldsymbol{x} - \boldsymbol{r}_i)^{\mathrm{T}} \boldsymbol{P} (\boldsymbol{x} - \boldsymbol{r}_i) \leq \rho_i\}. \tag{9}$$

Since $V(\boldsymbol{x})$ is a Lyapunov function associated with the feedback gain $\boldsymbol{L}$, any state trajectory that initially satisfies $\boldsymbol{x}_{k_0} \in \mathcal{O}_i$ at a time index k_0 will remain inside $\mathcal{O}_i$ for all $k > k_0$ if $\boldsymbol{r}_i$ is not changed. The scale factor ρ_i is determined as the largest value such that $\mathcal{O}_i$ does not violate the input constraints and the static output constraints (4). The determination of ρ_i is posed as an optimization problem, which for our constraints and equilibrium points has an analytic solution (see [16]).

Remark 2: Note that δ_k^{ff} depends on the road curvature, so ρ_i should ideally be determined online. It is possible to set offline bounds on the maximum curvature, for example, by utilizing knowledge of the maximum curvature on regular highways. Then it is possible to precompute ρ_i offline, and the difference for low-curvature roads where limited actuation is used is negligible.

Remark 3: The positively invariant sets do not depend on the OV locations. However, given a prediction of the OVs over the planning horizon, it is possible to determine (see Sec. III-B1) whether any of the OVs (5) intersect with (9). If not, then the closed-loop trajectory starting in $\mathcal{O}_i$ will also satisfy the constraints (5) imposed by the OVs over the planning horizon, provided the setpoint $\boldsymbol{r}_i$ is maintained.

B. Determining the Connectivity Graph and Graph Search

To determine a safe trajectory that avoids the obstacles and satisfies the input and output constraints, we find a sequence of setpoints $\boldsymbol{r}_i$ that, when tracked using (6), steers the vehicle from the initial state $\boldsymbol{x}_k$ at time t_k to the setpoint corresponding to the desired lane before the end of the planning horizon T_f. The initial setpoint is determined by finding which invariant sets the initial state $\boldsymbol{x}_k$ is contained in.

We formulate the problem as a graph-search problem over the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of vertices $\mathcal{V}$ and edges $\mathcal{E}$, and divide the planning horizon into N steps with discretization time T_s, where $T_s = \ell\Delta t$ for a positive integer ℓ is a multiple of the discretization time Δt of the vehicle dynamics. The vertices are the indices (m, i) corresponding to the setpoints $\{\{\boldsymbol{r}_i^m\}_{i=1}^{n_r}\}_{m=0}^{N} \in \mathcal{V}$ for the different time steps m, where $m = 0$ corresponds to the initial time t_k. Then the graph-search problem consists of finding a sequence of setpoints that can be used to safely navigate the vehicle from the initial state to the middle of any of the lanes.

The edges $\mathcal{E}$ indicate which of the setpoints are connected by safe trajectories. An equilibrium point $\boldsymbol{r}_i$ with positively invariant set $\mathcal{O}_i$ is connected to $\boldsymbol{r}_j$ with positively invariant set $\mathcal{O}_j$ in l time steps (i.e., in one planning step) if $\mathcal{O}_i$ is contained in $\bar{\mathcal{O}}_j^\ell$,

$$\mathcal{O}_i \subseteq \bar{\mathcal{O}}_j^\ell, \tag{10}$$

where

$$\bar{\mathcal{O}}_j^\ell = \{\boldsymbol{x} \in \mathbb{R}^4 : (\boldsymbol{x} - \boldsymbol{r}_j)^{\mathrm{T}}(\bar{\boldsymbol{A}}^\ell)^{\mathrm{T}}\boldsymbol{P}\bar{\boldsymbol{A}}^\ell(\boldsymbol{x} - \boldsymbol{r}_j) \leq \rho_j\}. \tag{11}$$

Eq. (10) and the corresponding connectivity can be interpreted as in Fig. 3 for a case where $l = 3$. At $m = 1$, no states from $\mathcal{O}_1$ can reach $\mathcal{O}_2$. At $m = 2$, $\mathcal{O}_1$ and $\bar{\mathcal{O}}_2^2$ intersect but not all points in $\mathcal{O}_1$ can reach $\mathcal{O}_2$. However, at $m = 3$, $\mathcal{O}_1$ is covered in the interior of $\bar{\mathcal{O}}_2^3$. This ensures that all points in $\mathcal{O}_1$ can reach $\mathcal{O}_2$ in $m = \ell$ time steps; hence, $\boldsymbol{r}_1^0$ is connected to $\boldsymbol{r}_2^1$.

Suppose that the vehicle state $\boldsymbol{x}_m$ is in the positively invariant set $\mathcal{O}_i$ associated with the setpoint $\boldsymbol{r}_i^m$ at planning index m. If there exists an edge between $\boldsymbol{r}_i^m$ and $\boldsymbol{r}_j^{m+1}$, it is safe to move from $\boldsymbol{x}_m$ to $\boldsymbol{r}_j$ over the time period T_s. Exact evaluation of (10) requires solving a nonconvex quadratically-constrained quadratic program. However, it can be shown that a sufficient condition for (10) to hold can be evaluated analytically through

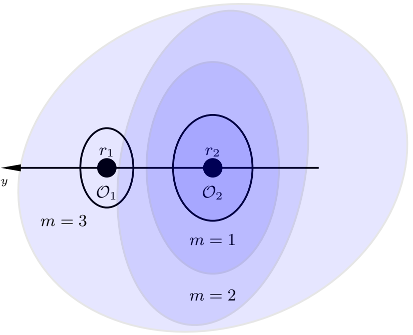

Fig. 3. A schematics of how positively invariant sets are connected, illustrated by iterative use of (11) for successively increasing m for a case where $l = 3$. The states that can reach $\mathcal{O}_j$ in $m = 1, 2, 3$ timesteps are shown as ellipses with increasing size. The setpoint $\boldsymbol{r}_1^0$ is connected to $\boldsymbol{r}_2^1$ because all states in $\mathcal{O}_1$ can reach $\mathcal{O}_2$ in $m = l$ timesteps.

$$(\boldsymbol{r}_i - \boldsymbol{r}_j)^{\mathrm{T}}(\bar{\boldsymbol{A}}^l)^{\mathrm{T}}\boldsymbol{P}\bar{\boldsymbol{A}}^l(\boldsymbol{r}_i - \boldsymbol{r}_j) \leq \rho_j - \rho_i\big\|\boldsymbol{P}^{-1/2}\bar{\boldsymbol{A}}\boldsymbol{P}^{1/2}\big\|_F, \tag{12}$$

where $\|\cdot\|_F$ is the Frobenius norm. Checking for connectivity using (12) may be conservative but has low computational cost and is an offline procedure if the bound determination in Remark 2 is used.

1) Obstacle Avoidance: The connectivity test (12) between equilibrium points is done offline and in absence of any OVs. While it is possible to online change the size of ρ_i and ρ_j in (9) and (4), respectively, depending on the obstacle constraints $\mathcal{S}_k$, the computational cost will be high. Instead, suppose that the OVs have been predicted over the time of the planning horizon, leading to the obstacle set (5). In general, if

$$\mathcal{O}_i \bigcap \mathcal{S}_k(T_{p1}, T_{p2}) \neq \emptyset, \tag{13}$$

the equilibrium point $\boldsymbol{r}_i$ associated with $\mathcal{O}_i$ is marked as unsafe and eliminated from the graph search between T_{p1} and T_{p2}. In practice, (13) amounts to checking for intersection between EV and OVs in the lateral position coordinate and by accounting for the safety time t_s introduced in Sec. II-A. The method is compatible with motion-prediction and threat-assessment methods that have been proposed in literature but further elaboration on this is out of the scope of the paper.

2) Graph Search: With all edges between the vertices determined, we construct a weighted adjacency matrix $\boldsymbol{M}$ between all $\{\{\boldsymbol{r}_i^m\}_{i=1}^{n_r}\}_{m=0}^{N} \in \mathcal{V}$. If two vertices are not connected, the corresponding edge weight is set to ∞. A connection between two vertices is indicated by setting the edge weight to the cost of moving between the different vertices. For instance, edges corresponding to transitions between the middle of either of the lanes might have a low cost, whereas transitions close to the road boundaries may have larger cost. Because of time-causality and size limitations of the positively invariant sets, the matrix will be upper-block diagonal and very sparse.

We employ Dijkstra's algorithm for the graph search in our numerical experiments. To efficiently solve the corresponding graph-search problem there exist a variety of graph-search methods [8]. When the graph search is completed and the setpoints that should be tracked have been found, the controller (6) tracks these points and switches between them until the setpoint corresponding to the target point has been reached.

C. Implementation Aspects

The vehicle model (2) assumes knowledge of the disturbance $\dot{\psi}_d$, which has to be estimated. The disturbance can be written as $\dot{\psi}_d(t) = v_x c(t)$, where $c(t)$ is the road curvature, which is unknown. However, it is possible to point-wise estimate the curvature given data points of the road boundary or the lane markers. We fit a circle segment to the data points at each time step and from that estimate the radius of curvature.

The steering inputs and corresponding trajectory are implemented as a receding horizon strategy; that is, the computed trajectory is T_f s long but is only applied for $T_r \leq T_f$ s. This ensures that feedback is not only imposed during the trajectory tracking but also in the planning stage. To not switch between trajectories excessively, the cost associated with the previously determined sequence of setpoints is decreased in the next graph search to favor the same solution.

The dynamics is discretized assuming a sampling time Δt, which is typically determined by the update rate of the sensor or the available computing power. However, n_r and T_s that indicate the number of setpoints and the discretization time in the planner, respectively, are tightly connected. For instance, a small T_s means that a large number of setpoints n_r is needed.

D. Algorithm Summary

One step of the complete algorithm is summarized in Algorithm 1. Most of the computations are done offline. Online, the most demanding task is to perform the prediction of obstacles (Line 2). The computational complexity depends on the number of obstacles in the region of interest and the method employed. The graph search (Line 5) is computationally fast, since the graph matrix $\boldsymbol{M}$ is upper block-diagonal and sparse. On Line 5, $\mathcal{I}_r$ is the set of indices found in the graph search to reach the target setpoint from the initial state. Tracking of the sequence of setpoints (Lines 7 and 8) is a matter of trigonometry (Line 7) and applying standard state-feedback control (Line 8) is computationally inexpensive.

IV. Simulation Study

We consider an autonomous vehicle (EV) that travels on a single-directed two-lane road. The road coordinates are taken from a real test track (the JARI test track), and the road includes both straight-line driving and curved segments. The desired velocity is $v_x = 20$ m/s. There are surrounding vehicles in both lanes with velocities between $9 - 18$ m/s. The EV tries to maintain the right lane whenever possible, but otherwise changes lane. The planning is done in the road-aligned, local frame. However, in the simulation, the control inputs δ are used in a vehicle modeled in the global frame.

Algorithm 1 Proposed method

```
    Offline: Compute $\mathcal{O}_i$ using (9) $\forall i \in [1, \dots, n_r]$.
    Offline: Construct adjacency matrix $\boldsymbol{M}$ between all set-
    points $\boldsymbol{r}_i$ by determining (10) using (12).
 1: Input: $\boldsymbol{x}_k$, estimates of OVs, target lane.
 2: Predict obstacle set (5).
 3: Check for intersection using (13) and remove corresponding edges in $\boldsymbol{M}$.
 4: Determine the setpoint $\boldsymbol{x}_k$ belongs to.
 5: Perform a graph search to find a sequence $\{\boldsymbol{r}_i\}_{i \in \mathcal{I}_r}$.
 6: for $t = \{t_k, t_{k+1}, \dots, t_k + T_r\}$ do
 7:     Estimate $\dot{\psi}_d$.
 8:     Control the vehicle using (6) with setpoint $\mathcal{I}_r(t)$.
 9: end for
10: Go to Line 1.
```

TABLE I
PARAMETER VALUES USED FOR THE SIMULATION STUDY.

Parameter	value	Unit	Description
Δt	0.1	s	Sampling time Vehicle dynamics
T_f	10	s	Planning horizon
T_s	0.5	s	Sampling time in planner
N	20	-	T_f/T_s
T_r	0.5	s	Control horizon
n_r	15	-	# road discretization points

The vehicle parameters are those of a real mid-size SUV, see [20] for more details. Table I provides the set of algorithm parameters. These values correspond to a weighted adjacency matrix $\boldsymbol{M} \in \mathbb{R}^{485 \times 485}$, out of which approximately 1600 elements are nonzero (i.e., less than 1%).

Fig. 4 shows snapshots of the planning for four timesteps. In the figure, the time at which switching between setpoints occur can be seen in the second to fourth subplots. Eq. (12) is the magnified positively invariant set when the switching between setpoints is initiated (c.f. Fig. 3).

The computation time of the proposed approach is low enough for real-time implementation. Fig. 5 displays how the computation time varies as the number of elements in the weighted adjacency matrix change, that is, the product Nn_r. The complexity grows linearly with the number of elements and the computation times are competitive when comparing with recent lane-change approaches based on MPC.

V. Conclusion

We presented a method for lane-change maneuvers that connects motion planning and vehicle control by exploiting positively invariant sets. A positively invariant set guarantees that whenever a state trajectory is inside that set, the state trajectory will remain in that set. By leveraging this, we developed a method that can enforce the vehicle to satisfy constraints on the vehicle motion as well as avoiding collisions. The method uses a graph search to determine the full lane change, and then executes state-feedback control combined with setpoint switching to perform the lane change.

Our method assumes that the longitudinal velocity is constant over the planning horizon. We will in future work relax this limitation.

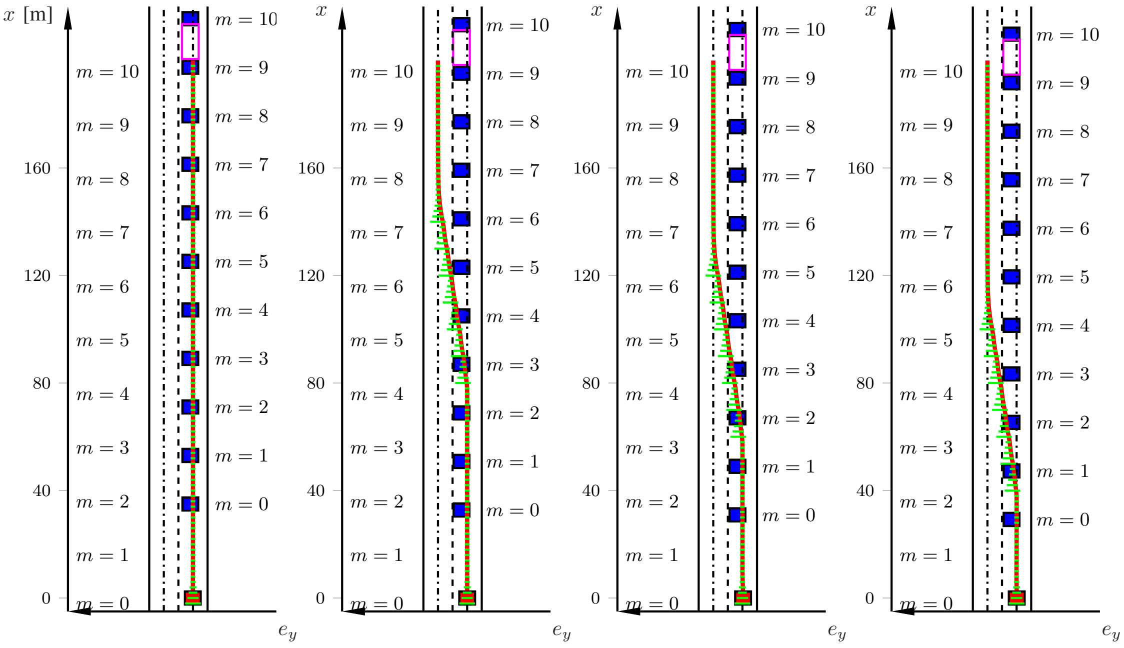

Fig. 4. Four snapshots with one second between each snapshot. EV in red and OVs in blue. Resulting trajectory for the planning horizon is in red and the corresponding time the EV reaches a particular point on the trajectory is enumerated to the left in each snapshot. The positively invariant sets corresponding to the tracked setpoints are shown in green, projected on the road. The OV present in the ROI is measured at $m = 0$, and the predictions over the planning horizon are also visualized. In the left-most figure, no OVs are predicted to intersect with the positively invariant sets over the planning horizon ($T_f = 10$ s). One second later, the OV is predicted to be within the safety margin (shown in magenta) and the planner therefore decides to initiate an overtaking maneuver.

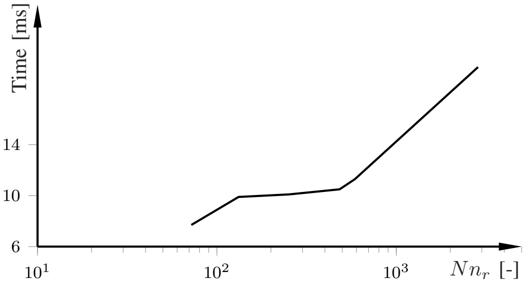

Fig. 5. Computation time as function of the number of elements in the weighted adjacency matrix. The implementation is done in MATLAB and the results are obtained on a standard 2014 i5 2.8GHz laptop.

REFERENCES

[1] S. M. Broek, E. van Nunen, and H. Zwijnenberg, "Definition of necessary vehicle and infrastructure sys-tems for automated driving," European Commission, Tech. Rep. 2010/0064, June 2011.

[2] K. Berntorp, "Path planning and integrated collision avoidance for autonomous vehicles," in *Amer. Control Conf.*, Seattle, WA, May 2017.

[3] Y. Kuwata, S. Karaman, J. Teo, E. Frazzoli, J. How, and G. Fiore, "Real-time motion planning with applications to autonomous urban driving," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 5, pp. 1105–1118, 2009.

[4] K. Berntorp and S. Di Cairano, "Joint decision making and motion planning for road vehicles using particle filtering," in *IFAC Symp. Advances in Automotive Control*, Kolmården, Sweden, June 2016.

[5] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, *et al.*, "Junior: The Stanford entry in the Urban challenge," *J. Field Robotics*, vol. 25, no. 9, pp. 569–597, 2008.

[6] S. Di Cairano, U. V. Kalabic, and K. Berntorp, "Vehicle tracking control on piecewise-clothoidal trajectories by MPC with guaranteed error bounds," in *IEEE Conf. Decision and Control*, Las Vegas, NV, Dec. 2016.

[7] J. Funke, M. Brown, S. M. Erlien, and J. C. Gerdes, "Collision avoidance and stabilization for autonomous vehicles in emergency scenarios," *IEEE Trans. Control Syst. Technol.*, vol. PP, no. 99, pp. 1–13, 2016.

[8] S. M. LaValle, *Planning Algorithms*. Cambridge, UK: Cambridge University Press, 2006.

[9] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Veh.*, vol. 1, no. 1, pp. 33–55, 2016.

[10] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, *et al.*, "Autonomous driving in urban environments: Boss and the Urban challenge," *J. Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[11] A. Carvalho, S. Lefévre, G. Schildbach, J. Kong, and F. Borrelli, "Automated driving: The role of forecasts and uncertainty—A control perspective," *European J. Control*, vol. 24, pp. 14–32, 2015.

[12] J. Nilsson, P. Falcone, M. Ali, and J. Sjöberg, "Receding horizon maneuver generation for automated highway driving," *Control Eng. Pract.*, vol. 41, pp. 124–133, 2015.

[13] N. Murgovski and J. Sjöberg, "Predictive cruise control with autonomous overtaking," in *IEEE Conf. Decision and Control*, Osaka, Japan, 2015.

[14] V. Turri, A. Carvalho, H. E. Tseng, K. H. Johansson, and F. Borrelli, "Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads," in *IEEE Int. Conf. Intell. Transp. Syst.*, The Hague, Netherlands, 2013.

[15] A. Weiss, C. Petersen, M. Baldwin, R. S. Erwin, and I. Kolmanovsky, "Safe positively invariant sets for spacecraft obstacle avoidance," *J. Guidance, Control, and Dynamics*, vol. 38, no. 4, pp. 720–732, 2014.

[16] C. Danielson, A. Weiss, K. Berntorp, and S. Di Cairano, "Path planning using positive invariant sets," in *IEEE Conf. Decision and Control*, Las Vegas, NV, Dec. 2016.

[17] H. B. Pacejka, *Tire and Vehicle Dynamics*, 2nd ed. Oxford, United Kingdom: Butterworth-Heinemann, 2006.

[18] R. Rajamani, *Vehicle Dynamics and Control*. Berlin Heidelberg: Springer-Verlag, 2006.

[19] A. Gray, M. Ali, Y. Gao, J. K. Hedrick, and F. Borrelli, "Integrated threat assessment and control design for roadway departure avoidance," in *IEEE Int. Conf. Intell. Transp. Syst.*, Anchorage, AK, 2012.

[20] K. Berntorp and S. Di Cairano, "Tire-stiffness estimation by marginalized adaptive particle filter," in *IEEE Conf. Decision and Control*, Las Vegas, NV, Dec. 2016.