

Positive Invariant Sets for Safe Integrated Vehicle Motion Planning and Control

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

TR2018-172 December 29, 2018

Abstract

This paper describes a method for real-time integrated motion planning and control of autonomous vehicles. Our method leverages feedback control, positive invariant sets, and equilibrium trajectories of the closed-loop system to guarantee collision-free closed-loop trajectory tracking. Our method jointly steers the vehicle to a target region and controls the velocity while satisfying constraints associated with the future motion of the obstacles with respect to the vehicle. We develop a receding-horizon implementation and verify the method in a simulated road scenario. The results show that our method generates safe dynamically feasible trajectories while accounting for obstacles in the environment and modeling errors. In addition, the computation times indicate that the method is sufficiently efficient for real-time implementation.

IEEE Annual Conference on Decision and Control (CDC)

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Positive Invariant Sets for Safe Integrated Vehicle Motion Planning and Control

Karl Berntorp, Claus Danielson, Avishai Weiss, and Stefano Di Cairano

Abstract—This paper describes a method for real-time integrated motion planning and control of autonomous vehicles. Our method leverages feedback control, positive invariant sets, and equilibrium trajectories of the closed-loop system to guarantee collision-free closed-loop trajectory tracking. Our method jointly steers the vehicle to a target region and controls the velocity while satisfying constraints associated with the future motion of the obstacles with respect to the vehicle. We develop a receding-horizon implementation and verify the method in a simulated road scenario. The results show that our method generates safe dynamically feasible trajectories while accounting for obstacles in the environment and modeling errors. In addition, the computation times indicate that the method is sufficiently efficient for real-time implementation.

I. INTRODUCTION

Autonomous vehicle guidance and control is commonly divided into trajectory generation (motion planning) and trajectory tracking (vehicle control) [1], [2]. Trajectory generation is often performed using either sampling-based methods such as rapidly-exploring random trees (RRTs) [3]–[5] or graph-search methods [6], [7]. Trajectory tracking in automotive is frequently done by conventional (e.g., PID) or more advanced control algorithms (e.g., model-predictive control (MPC) [8], [9]). Viewing the trajectory generation and tracking problems as decoupled is appealing as it simplifies the problem. This is the dominant approach in the robotics community [3] and is also frequently used in automotive applications [1]. However, the time scales, dynamics, and stringent performance and driving requirements that are present in automotive systems motivate a more integrated approach to planning and control than in traditional robotics. It may be advantageous to not consider planning and control as isolated parts of the autonomous vehicle, but rather as interacting components. Thus, an important question is how to connect the motion planning and vehicle control to ensure performance and safety of the vehicle [8], [10].

This paper builds on our previous work in [11], in which we developed a method for motion planning and tracking that enabled lane-change maneuvers be performed on structured road networks. Our approach in [11] assumed a constant velocity over the planning horizon. It used a graph search over a finite set of lateral displacements on the road to determine a trajectory from initial state to desired lane, where a constraint admissible positive invariant set was associated with each lateral displacement. that starts inside the set remains in the set for all future times as long as the setpoint is unchanged.

The authors are with Mitsubishi Electric Research Laboratories (MERL), 02139 Cambridge, MA, USA. Email: karl.o.berntorp@ieee.org

This work extends the method to allow variable velocity within the planning horizon. Our method steers the vehicle and controls the velocity using state-feedback controllers, while satisfying input and state constraints associated with the future motion of the obstacles with respect to the vehicle. Instead of using lateral displacements, to account for the velocity variations we define reference paths on the road, with respect to the local vehicle frame, that are associated with equilibrium trajectories of the closed-loop system. Our method uses a graph search over reference paths, to find a path from the initial state to the desired region, where a constraint admissible positive invariant set is associated with each reference path.

We formulate the planning and tracking problem in error coordinates of the vehicle with respect to the road-aligned coordinate frame. This reduces the dimensionality of the graph-search problem to an extent that computational times become suitable for real-time execution. Compared to methods for lane-change maneuvers and overtaking in automotive based on MPC (e.g., [12], [13]), and algorithms connecting MPC and set invariance for obstacle avoidance in robotic systems (e.g., [14], [15]), our method does most of the computations offline. MPC relies on solving constrained optimization problems in real time, whereas our approach solves a low-dimensional graph-search problem that accounts for a model of the closed-loop system and uses predesigned unconstrained linear quadratic controllers to track the reference path generated by the graph search. 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.

Notation: We denote vectors in lower-case bold font as \mathbf{x} and matrices with \mathbf{X} , and $\hat{\mathbf{x}}$ denotes the estimate of \mathbf{x} . A set \mathcal{O} is positive invariant for the system $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k)$ if $\forall \mathbf{x} \in \mathcal{O}$ we have $\mathbf{f}(\mathbf{x}) \in \mathcal{O}$. If $V(\mathbf{x})$ is a Lyapunov function for the stable system $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k)$, then any level set $\mathcal{O}\{\mathbf{x} \in \mathbb{R}^n : V(\mathbf{x}) \leq \rho\}$ is positive invariant since $V(\mathbf{f}(\mathbf{x})) \leq V(\mathbf{x})$, and we write $\mathbf{x}_{0:k} = \{\mathbf{x}_0, \dots, \mathbf{x}_k\}$. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined by a set of vertices \mathcal{V} and a list of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. Two vertices $i, j \in \mathcal{V}$ are adjacent if $(i, j) \in \mathcal{E}$. A path is a sequence of adjacent vertices, and two vertices $i, j \in \mathcal{E}$ are connected if there exists a path connecting them.

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). 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, with the origin at the start of each planning step fixed to the middle of the road. We introduce the following assumptions.

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

These can be measured and estimated by cameras, Lidars, radars, and/or ultrasound sensors attached to the EV. The future states of the OVs over the planning horizon are not assumed to be known a priori in the method we propose.

Assumption 2: The road geometry, number of lanes, and the direction of travel in each lane is known.

These quantities are usually known over the ROI from maps and onboard cameras.

A. Vehicle Model

We introduce the following assumption on the driving behavior.

Assumption 3: The planner operates regular driving maneuvers, while emergency braking and aggressive evasive maneuvers are handled by a separate control system.

Due to Assumption 3, the planner only operates in normal driving conditions where the lateral dynamics are well represented by 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 $F_{y,f} \approx C_f \alpha_f$, $F_{y,r} \approx C_r \alpha_r$, where C_f , C_r are the front and rear lateral tire stiffness coefficients. The slip angles α_f , α_r can be approximated as $\alpha_f \approx \delta - \frac{v_y + l_f \dot{\psi}}{v_x}$, $\alpha_r \approx \frac{l_r \dot{\psi} - v_y}{v_x}$, where δ is the steering angle of the front wheel (a 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 lateral dynamics state vector

$$\mathbf{x}_{\text{lat}} = [e_y \quad \dot{e}_y \quad e_\psi \quad \dot{e}_\psi]^T, \quad (1)$$

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.

The model of the lateral dynamics in the error coordinates (1) can be written as the linear system [16]

$$\dot{\mathbf{x}}_{\text{lat}} = \mathbf{A}_e \mathbf{x}_{\text{lat}} + \mathbf{B}_e \delta + \mathbf{D}_e \dot{\psi}_d. \quad (2)$$

For the longitudinal dynamics, from Assumption 3 we use the approximation $a_x = \dot{v}_x + v_y \dot{\psi} \approx \dot{v}_x$, since in normal driving conditions $v_y \dot{\psi} \ll \dot{v}_x$. Furthermore, again motivated by Assumption 3, we assume that the longitudinal slip is sufficiently small such that the contribution of the longitudinal force to the lateral motion of the vehicle will be negligible compared to the lateral force. This results in

$$\dot{e}_x = v_x - v_{x,\text{nom}}, \quad (3a)$$

$$m \dot{v}_x = \sum_{i=f,r} R \tau_i, \quad (3b)$$

where e_x is the longitudinal position in the road-aligned coordinate frame, with respect to a nominal reference path with nominal velocity $v_{x,\text{nom}}$, m is the mass, and R is the wheel radius. We model the total actuation torque $\tau = \sum_{i=f,r} \tau_i$ as a first-order system from input torque τ to acceleration with time constant T_c , which gives the longitudinal dynamics

$$\dot{e}_x = v_x - v_{x,\text{nom}}, \quad (4a)$$

$$\dot{v}_x = -\frac{T_c}{m} v_x + T_c u_a, \quad (4b)$$

where we have introduced the control input $u_a = \tau R/m$, and the nominal desired velocity $v_{x,\text{nom}}$ is assumed constant over the planning horizon. The combined continuous-time nonlinear lateral and longitudinal dynamics are described by (2) and (4), with input vector $\mathbf{u} = [\delta \ u_a]^T$. Next, we convert the continuous-time dynamics to discrete time with sampling period Δt , which results in the system

$$\mathbf{x}_{\text{lat},k+1} = \mathbf{A}(v_{x,k}) \mathbf{x}_{\text{lat},k} + \mathbf{b} \delta_k + \mathbf{d}(v_{x,k}) d_k, \quad (5a)$$

$$\mathbf{x}_{\text{lon},k+1} = \mathbf{F} \mathbf{x}_{\text{lon},k} + \mathbf{g} u_{a,k} + \mathbf{h}(v_{x,k}) w_k, \quad (5b)$$

$$\mathbf{y}_{\text{lat},k} = \mathbf{C} \mathbf{x}_{\text{lat},k}, \quad (5c)$$

$$\mathbf{y}_{\text{lon},k} = \mathbf{E} \mathbf{x}_{\text{lon},k}, \quad (5d)$$

where $\mathbf{x}_{\text{lon}} = [e_x \ v_x]^T$, k is the time index corresponding to time t_k , $d = \dot{\psi}_d$ is the disturbance term on the lateral dynamics and $w = v_{x,\text{nom}}$ is the nominal velocity we wish to track. Eqs. (5c), (5d) model the outputs we wish to plan a trajectory for and, subsequently, control.

B. System Constraints

We impose various constraints on the vehicle states and inputs. The input vector is subject to symmetric constraints on acceleration and steering angle, which can be written as

$$\mathcal{U} = \{\mathbf{u}_k : \mathbf{u}_{\min} \leq \mathbf{u}_k \leq \mathbf{u}_{\max}\}. \quad (6)$$

These constraints are determined by the physical limitations of the vehicle, induced by ensuring that the assumptions made for deriving (2) hold, or determined as a tradeoff between the level of aggressiveness and driving comfort.

The output $\mathbf{y}_k = [\mathbf{y}_{\text{lat},k} \ \mathbf{y}_{\text{lon},k}]^T$ is constrained as $\mathbf{y}_k \in \mathcal{Y}_k \in \mathbb{R}^m$, 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 on the lateral position of the EV. The term ψ_d associated with the curvature of the road in the global frame, together with bounds on the allowed lateral accelerations, also give rise to constraints, and limitations on local lateral velocity error can also be set. These constraints can compactly be written as

$$\mathcal{Y}_k = \{\mathbf{y}_k : \mathbf{H}_k \mathbf{y}_k \leq \mathbf{K}_k\} \quad (7)$$

for appropriately defined matrices \mathbf{H}_k and \mathbf{K}_k . In this paper, (7) only treats the lateral dynamics, since the longitudinal dynamics (5b) is completely governed by the input constraint (6) and the nominal velocity $v_{x,\text{nom}}$, which is a known parameter, as will be described in detail in Sec. III. The velocity is a setpoint and the overshoot when converging to the setpoint can be adjusted by tuning of the controller response

to the dynamics (4). In this paper (7) is time invariant and hence can be determined offline. In general (7) can be time varying, which, however increases the computational burden since the invariant set computation depends on (7).

The spatial extent of the collision area of the EV around the j th OV is denoted with \mathcal{B}^j . The longitudinal and lateral position and the velocity of each OV relative to the EV are included in the state vector \mathbf{x}_{OV} and add further time-varying constraints on the outputs of the EV. We define the (deterministic or probabilistic) j th obstacle set at time step k as $\mathcal{D}(\hat{\mathbf{x}}_{OV,k}^j, \mathcal{B}^j)$, which is a function of the predicted OV state vector and the spatial extent. Denote the planning horizon with N_p . Then, the predicted set of the j th obstacle for each $k \in [0, N_p]$ is

$$\mathcal{S}_k^j = \mathcal{D}(\hat{\mathbf{x}}_{OV,k}^j, \mathcal{B}^j). \quad (8)$$

The area the motion planner should avoid up until time index k is computed as the union over all OV trajectory sets (8),

$$\mathcal{S}_k = \bigcup_{j=1}^M \mathcal{S}_k^j. \quad (9)$$

C. Problem Statement

The objective of the integrated motion-planning and vehicle control approach developed in this paper is to generate an input trajectory $\mathbf{u}_k \in \mathcal{U}$ for all $k \in [0, N_p]$ over the planning horizon N_p , such that the resulting trajectory obtained from (5) satisfies the constraints (7), avoids the obstacle set (9), and reaches a given goal region $\mathcal{X}_{\text{goal}}$, that is, $\mathbf{x}_{N_p} \in \mathcal{X}_{\text{goal}}$.

III. SAFE MOTION PLANNING USING POSITIVE INVARIANT SETS

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

A. Feedback Control and Offline Graph Construction

In our previous approach using positive invariant sets for motion planning [11], we generated the paths by connecting equilibrium points that correspond to lateral positions on the road, where each equilibrium was associated with an invariant set. However, to allow variable velocity we must include velocity information into the equilibria.

We formulate the path-planning problem as a graph-search problem over the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of vertices \mathcal{V} and edges \mathcal{E} incorporating the closed-loop dynamics. The planning horizon N_p is constructed with sampling period T_s , where $T_s = \ell \Delta t$ for a positive integer ℓ is a multiple of the sampling period Δt of the vehicle dynamics model. At each time step $k \in [0, N_p]$ in the planning phase we define a set of candidate equilibrium references

$$\mathcal{R} = \{\mathbf{r}_k^j\}_{j=1}^R \subset \mathbb{R}^2, \quad (10)$$

where each equilibrium reference point

$$\mathbf{r}_k^j = \begin{bmatrix} v_{x,\text{nom}}^j & \bar{e}_{y,k}^j \end{bmatrix}^T \quad (11)$$

is modeled in the local vehicle frame relative to the position and velocity at the beginning of the planning phase. Note also that with the sampling time Δt of the vehicle dynamics, the nominal longitudinal relative position reference $e_{x,k}$ is also defined. The set of lateral reference points $\{\bar{e}_y^i\}_{i=1}^{n_y}$ define a grid on the road and includes the middle of each lane. For each candidate nominal velocity $\{v_{x,\text{nom}}^j\}_{j=1}^{n_v}$ the relative motion of the OVs to the EV will be different, and by checking for feasibility in the set of candidate reference velocities, the graph search will eventually find a collision-free closed-loop trajectory for a given reference velocity $v_{x,\text{nom}}^j$. The number of reference paths in (10) is $R = n_y n_v$ and constitute reference paths over the planning horizon N_p .

The motion planner integrates trajectory generation and trajectory tracking by exploiting state-feedback controllers for both longitudinal and lateral motion. The longitudinal dynamics (5b) is linear, and we are interested in tracking the nominal velocity v_{nom} . We enforce integral action [17] by adding $\epsilon_{x,k} = \epsilon_{x,k-1} + \Delta t \dot{e}_x$ to (5b) and design a state-feedback controller

$$a_{x,k} = -\boldsymbol{\kappa}_x^T (\mathbf{r}_{x,k} - [\mathbf{x}_{\text{lat},k} \quad \epsilon_{x,k}]^T), \quad (12)$$

where $\mathbf{r}_{x,k} = [0 \ v_{x,\text{nom}} \ 0]^T$ and $\boldsymbol{\kappa}_x$ is the feedback gain. The lateral dynamics (5a) is nonlinear in v_x . We linearize the dynamics (5a) about the nominal reference velocity $v_{x,\text{nom}}$, which gives a locally linear model

$$\mathbf{x}_{\text{lat},k+1} = \tilde{\mathbf{A}} \mathbf{x}_{\text{lat},k} + \mathbf{b} \delta_k + \tilde{\mathbf{d}} d_k. \quad (13)$$

We use a local state-feedback controller and again enforce integral action, $\epsilon_{y,k} = \epsilon_{y,k-1} + \Delta t e_y$, which results in

$$\delta_k = -\boldsymbol{\kappa}_y^T (\mathbf{r}_{y,k} - \mathbf{x}_{\text{lat},k}^a) + \delta_k^{ff} \quad (14)$$

for the augmented system with state $\mathbf{x}_{\text{lat},k}^a = [\mathbf{x}_{\text{lat},k}^T \quad \epsilon_{y,k}]^T$, and where $\mathbf{r}_{y,k} = [\bar{e}_{y,k} \ 0 \ 0 \ 0 \ 0]^T$. The feedforward term δ_k^{ff} corrects for the disturbance d_k due to the curvature of the road. We design the feedback gain $\boldsymbol{\kappa}_x$ to globally asymptotically stabilize $v_{x,\text{nom}}$ and $\boldsymbol{\kappa}_y$ to locally stabilize the lateral reference point (11), and construct a family of positive invariant sets $\{\mathcal{O}_i\}_{i=1}^{n_o} \subseteq \mathbb{R}^5$ of states $\mathbf{x}_{\text{lat}}^a \in \mathbb{R}^5$. Each positive invariant set \mathcal{O}_i is a level set of the quadratic Lyapunov function $V(\mathbf{x}_{\text{lat}}^a - \mathbf{r}_y^i)$ associated with the system (5a) in closed-loop with the controller $\boldsymbol{\kappa}_y$. The i th positive invariant set is

$$\mathcal{O}_i = \{\mathbf{x}_{\text{lat}}^a \in \mathbb{R}^5 : (\mathbf{x}_{\text{lat}}^a - \mathbf{r}_y^i)^T \mathbf{P} (\mathbf{x}_{\text{lat}}^a - \mathbf{r}_y^i) \leq \rho_i\}, \quad (15)$$

where \mathbf{P} is a symmetric matrix associated with the Lyapunov function $V(\cdot)$ [11]. Although each reference equilibrium point \mathbf{r}_y^i has an associated positive invariant set \mathcal{O}_i , storage-wise typically $n_o \neq n_r$ since the scale factor ρ_i is the same for multiple invariant sets. Since $V(\cdot)$ is a Lyapunov function associated with the feedback gain $\boldsymbol{\kappa}_y$, any state trajectory that is initially inside \mathcal{O}_i will remain inside \mathcal{O}_i for all $k > 0$ if \mathbf{r}_y^i

is unchanged. The scale factor ρ_i is determined as the largest value such that \mathcal{O}_i does not violate the input constraints (6) and the static output constraints (7). Determining ρ_i is in general a nonconvex optimization problem. However, for our locally linear dynamics (5a) and constraints (6), (7), the problem has a closed-form solution [18].

Each vertex $v \in \mathcal{V}$ of the graph includes the lateral equilibrium point, the state-feedback controller that stabilizes the equilibrium point, and the safe set associated with the state-feedback controller. The edges \mathcal{E} indicate which of the setpoints are connected by safe trajectories. An equilibrium point $\mathbf{r}_{y,i}$ with positive invariant set \mathcal{O}_i is connected to $\mathbf{r}_{y,j}$ with positively invariant set \mathcal{O}_j in ℓ time steps (i.e., in one planning step) if \mathcal{O}_i is contained in $\bar{\mathcal{O}}_j^\ell$ [19],

$$\mathcal{O}_i \subseteq \bar{\mathcal{O}}_j^\ell, \quad (16)$$

where

$$\bar{\mathcal{O}}_j^\ell = \{\mathbf{x} \in \mathbb{R}^5 : (\mathbf{x}_{\text{lat}}^a - \mathbf{r}_y^j)^\top \bar{\mathbf{P}}(\mathbf{x}_{\text{lat}}^a - \mathbf{r}_y^j) \leq \rho_j\}, \quad (17)$$

where $\bar{\mathbf{P}} = (\bar{\mathbf{A}}^{a,\ell})^\top \mathbf{P} \bar{\mathbf{A}}^{a,\ell}$, $\bar{\mathbf{A}}^a = \bar{\mathbf{A}} - \mathbf{b}\kappa_y$. Evaluating (16) exactly requires solving a nonconvex quadratically-constrained quadratic program. However, a sufficient condition for (16) to hold is

$$(\mathbf{r}_y^i - \mathbf{r}_y^j)^\top \bar{\mathbf{P}}(\mathbf{r}_y^i - \mathbf{r}_y^j) \leq \rho_j - \rho_i \|\mathbf{P}^{-1/2} \bar{\mathbf{A}}^a \mathbf{P}^{1/2}\|_F, \quad (18)$$

where $\|\cdot\|_F$ is the Frobenius norm. Checking for connectivity using (18) is not exact, however, conservative.

Remark 1: We focus on the lateral dynamics when designing the invariant sets. For driving maneuvers where Assumption 3 holds, the longitudinal dynamics are approximately decoupled from the lateral dynamics, as indicated by (5b).

B. Obstacle Avoidance

The connectivity test (18) between equilibrium points is done offline and in absence of any OVs. While it is possible to change online the size of ρ_i and ρ_j in (15) and (17), respectively, depending on the obstacle constraints \mathcal{S}_k , the computational cost will be too high for implementation on an embedded platform for automotive applications. Instead, during runtime, we check for intersection of the invariant sets with the obstacle set (9),

$$\mathcal{O}_i \cap \mathcal{S}_k \neq \emptyset. \quad (19)$$

In practice, we introduce a safety time of N_s time steps and a lateral safety margin w to account for sensing and estimation errors with respect to the OVs. The equilibrium point \mathbf{r}_i associated with \mathcal{O}_i is marked as unsafe and eliminated from the graph search between $k - N_s$ and $k + N_s$. As we traverse the different reference velocities, the relative motion of the OVs will be different and therefore different vertices will in general be connected for the different reference velocities. The method is compatible with motion-prediction and threat-assessment methods proposed in literature [20].

C. Online Graph Search and Reference Tracking

With all edges between the vertices determined, we construct a weighted adjacency matrix \mathbf{M} between all vertices 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. Starting with the first candidate velocity reference $v_{x,\text{nom}}^1$, we traverse the set of candidate velocities until a solution has been found. When the graph search is completed and the reference path has been found, this path is submitted to the controllers (12), (14) for execution.

IV. IMPLEMENTATION CONSIDERATIONS

The size of the invariant sets will change with the velocity setpoint we linearize around to get the linear model (13). In practice, we design a lateral controller for a subset of the range of velocities such that connectivity and stability in this subset is guaranteed.

In the adjacency matrix \mathbf{M} it is possible to encode [19] a minimum length of the planning horizon. The nominal planning horizon is N_p long, but it may be possible to reach the target region $\mathcal{X}_{\text{goal}}$ in fewer steps, and due to the stage costs associated with traversing the nodes, the graph search will favor solutions that have fewer steps. As solutions that are too short may be overly aggressive for passengers, we encode in \mathbf{M} that the path must be at least N_m steps long, where $0 < N_m \leq N_p$.

The vehicle model (2) assumes knowledge of the disturbance ψ_d , which has to be estimated. The disturbance can be written as $\psi_d(t) = v_x c(t)$, where $c(t)$ is the road curvature, which is an unknown function of time. However, it is possible to point-wise estimate the curvature given data points of either the road boundary or the lane markers, or from a map.

The computed steering inputs and corresponding trajectory are implemented as a receding horizon strategy. The computed trajectory is N_p steps long but is only applied for a portion N_c of the whole plan. This ensures that feedback is not only imposed during the trajectory tracking but also in the planning stage.

The vehicle dynamics sampling period Δt is typically determined by the update rate of the sensors or the available computing power and can be considered independent of the other parameters. However, n_r and T_s that indicate the number of setpoints and the sampling-period in the planner, respectively, are tightly connected. For instance, a small T_s means that a large number of setpoints n_r is needed.

The proposed 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 4) and the intersection test (Line 5). Line 5 scales linearly with the number of obstacles and the collision checks are quadratic in the output dimension. The graph search (Line 7) is computationally fast, since the graph matrix \mathbf{M} is upper block-diagonal and sparse and applying standard state-feedback control (Line 14) is computationally inexpensive. The computational cost depends on the number of obstacles in the region of interest and how many reference velocities the method needs to traverse before finding a solution.

Algorithm 1 Proposed method

Offline: Compute \mathcal{O}_i using (15) $\forall i \in [1, \dots, n_r]$ for different velocity setpoints and construct adjacency matrices \mathbf{M} by determining (16) using (18).

- 1: **Input:** $\hat{\mathbf{x}}_0, \{\hat{\mathbf{x}}_{ov,0}\}_{j=1}^M, \mathcal{X}_{\text{goal}}$.
- 2: Predict obstacle set (9).
- 3: **for** $v \in \{v_{x,\text{nom}}^i\}_{i=1}^{n_v}$ **do**
- 4: Closed-loop prediction of EV using v .
- 5: Check for intersection using (19) and remove corresponding edges in \mathbf{M} .
- 6: Determine the setpoint \mathbf{x}_0 belongs to.
- 7: Perform a graph search to find a reference path $\bar{\mathbf{r}}_{0:N}, N \in [N_m, N_p]$, where $\mathbf{r}_N \in \mathcal{X}_{\text{goal}}$.
- 8: **if** Solution found **then**
- 9: Break loop.
- 10: **end if**
- 11: **end for**
- 12: **for** $k = 1$ **to** N_c **do**
- 13: Estimate $\dot{\psi}_d$.
- 14: Control the vehicle using (12), (14) with setpoint $\bar{\mathbf{r}}_k$.
- 15: **end for**
- 16: Go to Line 1.

V. SIMULATION STUDY

We consider an autonomous vehicle (EV) that travels on the outer ring test track of the Japanese Automotive Research Institute. In the simulation, the obstacle set is predicted by designing lane-tracking controllers that control the OVs assuming a fixed lane over the planning horizon N_p . The desired velocity is $v_x = 20$ m/s. Hence, this is the first candidate reference velocity the planner tries to find a collision-free trajectory for. The gridding of the velocity setpoints is done in decrements of 2 m/s down to 10 ms/s, that is, using five reference velocity setpoints. The goal region is chosen such that a path is considered to have reached the region if the endpoint is at least N_m steps long and is in the middle of either of the lanes.

The planning is done in the road-aligned, local coordinate frame. However, in the simulation, the computed control inputs are used in a vehicle modeled in the global coordinate frame. Furthermore, the disturbance $\dot{\psi}_d$ due to the road curvature nor the true motion of the vehicles are known to the planner. The disturbance is estimated online by fitting a circle segment to the data points. The obstacle set (9) is determined from obstacle predictions in the local frame, by using the position and velocity of each OV at time instant corresponding to the beginning of each planning phase. The

TABLE I
PARAMETER VALUES USED FOR THE SIMULATION STUDY.

Parameter	value	Unit	Description
Δt	0.1	s	Sampling period vehicle dynamics
T_s	0.5	s	Sampling period in planner
N_p	20 (10)	steps (s)	Nominal planning horizon
N_m	10 (5)	steps (s)	Minimum planning horizon
N_c	5 (0.5)	steps (s)	Control horizon
n_r	36	-	# road discretization points

simulation therefore also gives indications on how robust is the planner to these uncertainties.

Table I shows the algorithm parameters. These values correspond to a weighted adjacency matrix $\mathbf{M} \in \mathbb{R}^{758 \times 758}$, out of which approximately 3100 elements are nonzero (i.e., about 0.5%). We design one set of state-feedback controllers (12), (14) for the entire range [20, 10] of reference velocities and construct the adjacency matrix \mathbf{M} using the connectivity test for $v_{x,\text{nom}}^5 = 10$ m/s such that we ensure connectivity for the same vertices for all $v_{x,\text{nom}}^j > v_{x,\text{nom}}^5$.

Fig. 1 shows five snapshots of a situation where the EV catches up with two slower moving OVs, one in each lane. Eventually, there is no collision-free trajectory for the preferred velocity v_{nom}^1 , so the planner tests the different candidate velocities in decreasing order until a solution is found. In the figure, the time at which switching between setpoints occur can be seen in the second to fourth subplots. The positive invariant sets projected on the road are shown in green. When the switching between different setpoints is initiated (e.g., the second plot from the left), the contraction of the invariant sets due to (18) is noticeable. Fig. 2 displays the velocity profile for the time period corresponding to the snapshots. The time instants when the different snapshots occur are indicated by dashed lines.

Fig. 3 shows the computation time for the planning steps across the scenario. The nominal computation time is on average less than 25 ms. A slight increase can be seen for the time period when the velocity is decreased (c.f. Fig. 2), however, the increase is minor. The complexity grows linearly with the number of elements in the adjacency matrix [11] and the number of obstacles.

VI. CONCLUSION

This paper presented a method for integrating motion planning and vehicle control by exploiting positive invariant sets. The method enforces the vehicle to satisfy constraints on the vehicle motion as well as avoiding collisions. The method uses a graph search over lateral displacements on the road for different reference velocity setpoints, and then executes state-feedback controllers combined to track the reference path resulting from the graph search. The simulation study showed that the method can safely navigate the vehicle through tight passages where combined slow-down and lane change is needed, and the online computation requirements are modest.

REFERENCES

- [1] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving

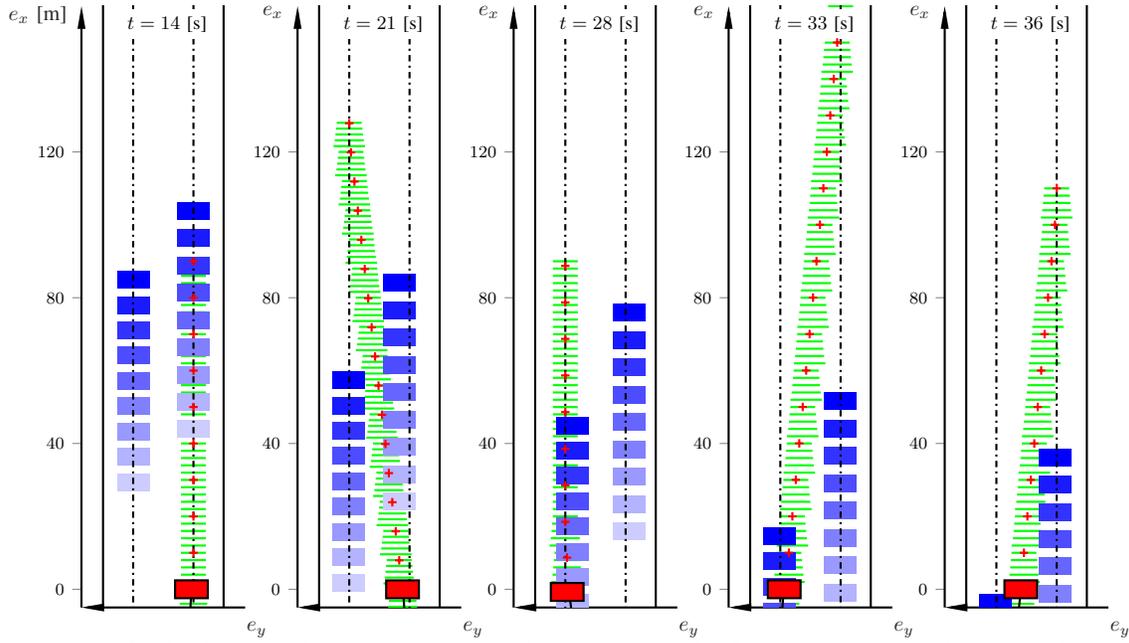


Fig. 1. Five snapshots of a situation where the EV (red) catches up with two slower moving OV's (blue), one in each lane. The desired path is indicated with red crosses and the invariant sets projected onto the road are in green. Resulting trajectory is in black and the corresponding time the EV reaches a particular point on the trajectory is enumerated to the left in each snapshot. In each snapshot, snapshots of the OV's every 0.5 s are shown in increasing color. The snapshots of the OV's correspond to the point at the time of the reference equilibrium points.

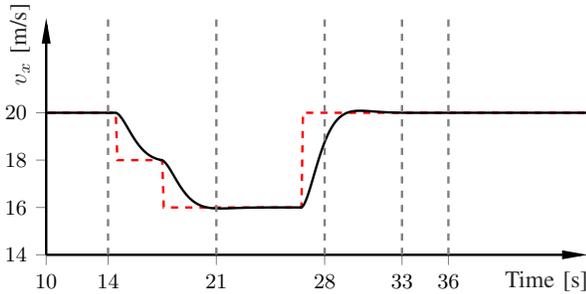


Fig. 2. The resulting velocity (black) and reference velocity setpoints (red dashed) as computed by the planner over the time span in Fig. 1. The dashed lines correspond to the snapshots in Fig. 1, numbered from left to right.

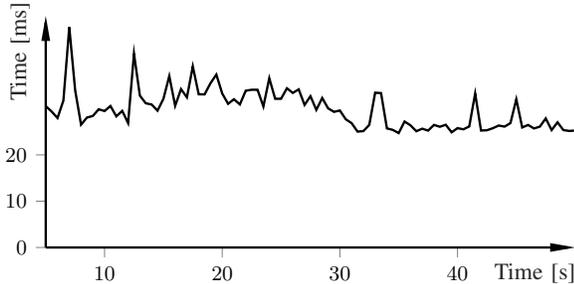


Fig. 3. Computation time over the time steps for the driving scenario (Fig. 1). The implementation is in MATLAB on a 2014 i5 2.8GHz laptop.

urban vehicles," *IEEE Trans. Intell. Veh.*, vol. 1, no. 1, pp. 33–55, 2016.

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

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

[4] 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.

[5] K. Berntorp and S. Di Cairano, "Particle filtering for online motion planning with task specifications," in *Amer. Control Conf.*, Boston, MA, July 2016.

[6] 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 R.*, vol. 25, no. 8, pp. 425–466, 2008.

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

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

[9] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 3, pp. 566–580, 2007.

[10] A. Carvalho, S. Lefèvre, G. Schildbach, J. Kong, and F. Borrelli, "Automated driving: The role of forecasts and uncertainty - a control perspective," *Eur. J. Control*, vol. 24, pp. 14–32, 2015.

[11] K. Berntorp, A. Weiss, C. Danielson, I. Kolmanovsky, and S. Di Cairano, "Automated driving: Safe motion using positively invariant sets," in *Int. Conf. Intell. Transp. Syst.*, Yokohama, Japan, Oct. 2017.

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

[13] 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 *Int. Conf. Intell. Transp. Syst.*, The Hague, Netherlands, 2013.

[14] S. Singh, A. Majumdar, J. J. Slotine, and M. Pavone, "Robust online motion planning via contraction theory and convex optimization," in *Int. Conf. Robot. Autom.*, Singapore, May 2017.

[15] G. Franzè and W. Lucia, "A receding horizon control strategy for autonomous vehicles in dynamic environments," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 2, pp. 695–702, 2016.

[16] R. Rajamani, *Vehicle Dynamics and Control*. Springer-Verlag, 2006.

[17] K. J. Åström and B. Wittenmark, *Computer-Controlled Systems*. Dover, 2011.

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

[19] 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.

[20] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *Robomech J.*, vol. 1, no. 1, p. 1, 2014.