Optimal Acceleration-Bounded Trajectory Planning in Dynamic Environments Along a Specified Path

Jeff Johnson and Kris Hauser

***Abstract*— Vehicles that cross lanes of traffic encounter the problem of navigating around dynamic obstacles under actuation constraints. This paper presents an optimal, exact, polynomial-time planner for optimal bounded-acceleration trajectories along a fixed, given path with dynamic obstacles. The planner constructs reachable sets in the path-velocity-time (PVT) space by propagating reachable velocity sets between obstacle tangent points in the path-time (PT) space. The terminal velocities attainable by endpoint-constrained trajectories in the same homotopy class are proven to span a convex interval, so the planner merges contributions from individual homotopy classes to find the exact range of reachable velocities and times at the goal. A reachability analysis proves that running time is polynomial given reasonable assumptions, and empirical tests demonstrate that it scales well in practice and can handle hundreds of dynamic obstacles in a fraction of a second on a standard PC.**

I. INTRODUCTION

We consider the problem of planning safe trajectories for a robot along a given path in the presence of dynamic obstacles. We are primarily motivated by scenarios in which the vehicle must cross multiple lanes of traffic in non-signalized intersections and left-hand turns, where the vehicle must plan its acceleration and braking in order to negotiate safely between gaps in traffic. Such scenarios are the second-highest cause of accidents for elderly drivers in the United States [1] and are a challenge for automated urban driving.

We present a polynomial-time, optimal, exact algorithm for the problem of planning acceleration and braking actions along a known path. Kant and Zucker (1986) showed how the version of this problem with velocity bounds but without acceleration constraints can be addressed by a visibility graph formulation in the two-dimensional path position/time (PT) space [2]. Our algorithm extends this approach to handle acceleration bounds. Like Kant and Zucker (1986) the nodes of the visibility graph correspond to obstacle vertices in PT space, but these points correspond to lines of variable velocity in the three-dimensional path/velocity/time (PVT) space [3]. To connect two nodes, *sets of reachable velocities* must be propagated from one line to another.

We show that when considering a set of homotopic trajectories within a "channel" of PT space, the set of reachable velocities at a terminal PT point is a convex interval. We also present a method to construct this interval exactly in $O(n^3)$ time. Our planner then generalizes this technique to many sets of homotopic trajectories, which requires considering the possibility that the set of reachable velocities at a PT point may be composed of multiple disjoint intervals. An analysis shows that the number of intervals is, for all practical purposes, bounded by $6n$.

Jeff Johnson and Kris Hauser are with the School of Informatics and Computing, Indiana University {jj56,hauserk}@indiana.edu

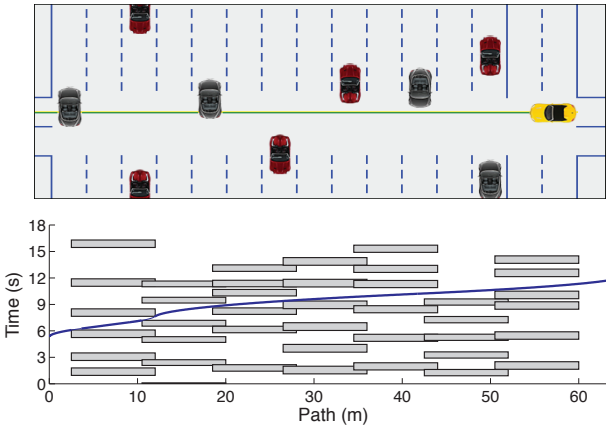

Fig. 1. Above: A lane crossing scenario where the vehicle (right) must follow the fixed path across 13 lanes of traffic with many oncoming vehicles at varying velocities. Below: A plot of the PT obstacles (shaded) and the optimal trajectory (curve) computed by the planner. The supplemental video contains an animation of the simulated trajectory.

For a problem with n obstacles the planner runs in worst case $O(n^4)$ time and $O(n^2)$ space. We have implemented the algorithm, and evaluations in Section V-C indicate that typical performance is closer to $O(n^3)$ time with a small leading coefficient and $O(n)$ space. We demonstrate the planner by computing the optimal trajectory for a robotic vehicle crossing 13 lanes of traffic with dozens of obstacles (Fig. 1). An implementation of the planner and simulation videos are available at http://www.indiana.edu/∼motion/pvtp/.

A. Related Work

Kant and Zucker (1986) presented a visibility graph formulation for time-optimal planning among dynamic obstacles and under velocity bounds, but allow the robot to change velocity instantaneously. Canny et al (1990) presented an exponential-time exact algorithm for planning optimal motions of a point in a plane amongst fixed obstacles under L_∞ acceleration bounds [4]. Optimal planning along a path with velocity and acceleration bounds along time-varying upper and lower bounds was solved in polynomial time by Ó'Dúnlaing (1987) [5], which is highly related to our "channel" construction of Section IV-B. To our knowledge our algorithm is the first exact algorithm to handle acceleration bounds as well as dynamic obstacles in polynomial time.

More recent techniques allow avoiding dynamic obstacles

without a fixed path, but sacrifice exactness and completeness to gain computational efficiency. A global, grid-based method for planning among moving obstacles and with acceleration constraints was presented in the velocity obstacles work of Fiorini and Shiller (1998) [6]. Other work in vehicle collision avoidance has used techniques like receding horizon control [7] and randomized kinodynamic planning [8], [9] to explore the vehicle's action/state space. Another approach is to consider a discrete lattice of precomputed dynamic maneuvers which allows for deeper exploration using classical search techniques [10], [11]. While these techniques are able to find paths as well as velocity profiles, they must pick a search resolution and/or time horizon in order to be computationally tractable.

The multi-robot coordination problem is also similar to ours, and in particular Peng and Akella (2005) considered multi-robot coordination along given paths with acceleration and velocity bounds [12]. A mixed integer programming formulation was presented for solving it. Our technique might also be extended to multi-robot coordination along given paths using the prioritized planning heuristic of Erdmann and Lozano-Perez (1987) [13].

II. Problem Definition

Suppose the robot has configuration space $\mathscr{C}$ and is given a fixed arc-length parameterized geometric path $q(p) : [0, p_{max}] \mapsto \mathscr{C}$. The dynamic planning problem occurs in the path-velocity-time (PVT) state space, in which states $x = (p, v, t)$ consist of a path position $p \in [0, p_{max}]$, velocity v, and time $t \in [0, t_{max}]$. In this paper we will also consider projections on the path-time (PT) plane as well as path-velocity (PV) slices at a particular time t.

The dynamics of the system must satisfy velocity and acceleration bounds:

$$\dot{p} = v \tag{1}$$

$$v \in [\underline{v}, \overline{v}] \tag{2}$$

$$\dot{v} \in [\underline{a}, \overline{a}] \tag{3}$$

with $\underline{v} \geq 0$ and $\underline{a} < 0 < \overline{a}$. A trajectory $x(t) = (p(t), v(t), t)$ defined over $t \in [a, b]$ in PVT space is *dynamically feasible* if and only if $\dot{p}(t) = v(t)$, $v(t) \in [\underline{v}, \overline{v}]$, and $\dot{v}(t) \in [\underline{a}, \overline{a}]$. We say a PVT state x_f is *dynamically reachable* from x_0 if there exists a dynamically feasible trajectory $x(t)$ over domain $[t_0, t_f]$ such that $x(t_0) = x_0$ and $x(t_f) = x_f$. Informally, we will say that $x(t)$ *connects* x_0 and x_f. Note that there is a one-to-one correspondence between $x(t)$ and its path function $p(t)$, and any function $p(t)$ gives rise to a trajectory $x(t) = (p(t), \dot{p}(t), t)$ as long as $\dot{p}(t)$ exists for all t. In this paper we will choose whichever representation is most convenient for a particular analysis.

Forbidden regions in the PT space correspond to the (p, t) points that would cause the robot to intersect obstacles at configuration $q(p)$ at time t [3]. In general it is challenging to determine an exact representation of the forbidden region, so we approximate it as a list of rectangles $O_i, i = 1, \ldots, n$ represented as $O_i = [\underline{p_{O_i}}, \overline{p_{O_i}}] \times [\underline{t_{O_i}}, \overline{t_{O_i}}]$, and future work will extend obstacle construction to arbitrary polygonal obstacles. These rectangles can be constructed conservatively to bound each obstacle, or approximately using cell decomposition or sampling. A trajectory $p(t)$ defined over $[a, b]$ is *collision free* if $(p(t), t) \notin O_i$ for all $i = 1, \ldots, n$ and $t \in [a, b]$.

The planning problem is stated as follows: find a dynamically feasible, collision-free trajectory $p(t)$ over a range $[0, t_f]$ with initial condition $p(0) = p_{start}$, $\dot{p}(0) = v_{start}$, and terminal condition $p(t_f) = p_{goal}$ and $\dot{p}(t_f) \in [\underline{v}_{goal}, \overline{v}_{goal}]$. The goal velocity is allowed to vary within a range rather than take on a specific value because a vehicle's speed is not precisely constrained upon exiting an intersection.

III. Reachable Set Analysis

Here we demonstrate several basic facts about dynamically reachable sets in PVT space in the absence of obstacles and present primitive subroutines that are used in our planner.

A. Dynamically Reachable Sets

Given an initial point $x_0 = (p_0, v_0, t_0)$, the dynamically reachable set of velocities $R(t; x_0)$ from x_0 at any point in time $t \geq t_0$ is the set of velocities attainable at t from x_0 via dynamically feasible trajectories. We show that $R(t; x_0)$ is a relatively simple convex region in the PV plane, bounded by at most six curves of parabolic or linear type [5]. There are in fact three types of region. The following discussion assumes the PV plane is oriented such that the path axis is horizontal, extending positively to the right, and the velocity axis is vertical, extending positively upwards.

Type I regions. Without loss of generality let $t_0 = 0$. Ignoring velocity constraints $v \in [\underline{v}, \overline{v}]$, the reachable set $R(t; x_0)$ is bounded on the left by a parabola corresponding to bang-bang trajectories at which acceleration is at a minimum for time t_s, and then switches to a maximum for time $t - t_s$. Let these be as P^-P^+ (parabolic down, parabolic up) trajectories. The right boundaries are likewise formed by P^+P^- trajectories. The parametric expressions $r(t, t_s)$ and $l(t, t_s)$ for the left and right bounds, respectively, are:

$$l(t, t_s) = (p_0 + tv_0 + \frac{1}{2}(\underline{a} - \overline{a})t_s^2 + \frac{1}{2}\overline{a}t^2, v_0 + t_s(\underline{a} - \overline{a}) + \overline{a}t) \tag{4}$$

$$r(t, t_s) = (p_0 + tv_0 + \frac{1}{2}(\overline{a} - \underline{a})t_s^2 + \frac{1}{2}\underline{a}t^2, v_0 + t_s(\overline{a} - \underline{a}) + \underline{a}t) \tag{5}$$

where t_s ranges from 0 to t. Note that traveling at constant maximum acceleration achieves the upper right vertex of $R(t; x_0)$, while traveling at minimum acceleration achieves the lower left vertex. Furthermore, both boundary curves are parabolic functions of v and are monotonically increasing along the domain. See Fig. 2a.

Type II regions. Now consider the case in which the reachable set exceeds a velocity maximum before time t. The reachable set boundary in this case adds two segments: a linear portion at $v = \overline{v}$ and an extra parabolic segment corresponding to P^+LP^- (parabolic-linear-parabolic) trajectories that reach $\overline{v}$, travel along it for some time, and then subsequently decelerate with minimum acceleration. This latter segment replaces the section of $r(t, t_s)$ constructed by

those P^+P^- trajectories that would otherwise exceed $\bar{v}$ at their switching time t_s.

The state arrives at the boundary by accelerating until $\bar{v}$ is reached at time t_v, progressing with constant velocity until time t_s, and then decelerating for time $t-t_s$. It can be shown that this construction maximizes the p value for a given v value given the constraints. The expression for a P^+LP^- boundary curve is:

$$p(t) = p_0 + t\bar{v} - \frac{(\bar{v}-v_0)^2}{2\bar{a}} + \frac{1}{2}(t-t_s)^2\underline{a} \quad (6)$$

$$v(t) = \bar{v} + (t-t_s)\underline{a} \quad (7)$$

which can be seen in Fig. 2b. A similar region utilizing a linear segment and P^-LP^+ curve is obtained at the velocity minimum.

Type III regions occur when both the velocity maximum and minimum are exceeded, and consist of six boundary curves: two *PP* segments, two linear, and two *PLP*-type segments. See Fig. 2c. (Let P without a superscript denote an arbitrary constant acceleration.)

A simple interpolation lemma proves that $R(t;x_0)$ is convex for all t and x_0.

Lemma 1: A convex combination $x(t) = \alpha x_a(t) + (1-\alpha)x_b(t)$, $\alpha \in [0,1]$ of dynamically feasible trajectories $x_a(t)$ and $x_b(t)$ is also dynamically feasible.

The proof is straightforward due to linearity of the differentiation operator and convexity of the velocity and acceleration bounds. The convexity of $R(t;x_0)$ follows easily. Later we will also make use of the following lemma.

Lemma 2: Any state that is dynamically reachable from x_0 can be reached by a feasible *PP* or *PLP* trajectory.

Proof: Note that all boundary curves of the parabolic type can be reached by *PLP* or *PP* trajectories. Let $x = (p,v,t)$ be dynamically reachable, that is $x \in R(t;x_0)$. The argument considers examining how $R(t;x_0)$ changes as the acceleration limits are scaled. Write this dependence explicitly as $R(t;x_0,\underline{a},\bar{a})$, and note that the dependence is continuous. Consider scaling the limits by a factor of γ from $\gamma = 1$ to 0 until x lies on a parabolic boundary segment of $R(t;x_0,\gamma\underline{a},\gamma\bar{a})$. At this value of γ, x can be reached from x_0 via a *PP* or *PLP* trajectory where the P segments accelerate at rate $\gamma\underline{a}$ or $\gamma\bar{a}$. ■

This result implies that only PP and PLP trajectory classes need be examined during reachability analysis.

B. Determining Reachable Velocities

Our planner uses a primitive operation $VReach(x_0,p_f,t_f)$ that produces the set of all terminal velocities at a PT state (p_f,t_f) among dynamically feasible trajectories starting at initial PVT state x_0. In other words, *VReach* computes the intersection $[\underline{v}_f,\bar{v}_f] \equiv R(t_f;x_0) \cap \{(p,v) \mid p = p_f\}$. As a side effect it also constructs example trajectories for each extremum of the interval, which will later be used by the planner. We describe in detail only how to find the maximum reachable velocity $\bar{v}_f$, because finding the minimum $\underline{v}_f$ is essentially symmetric.

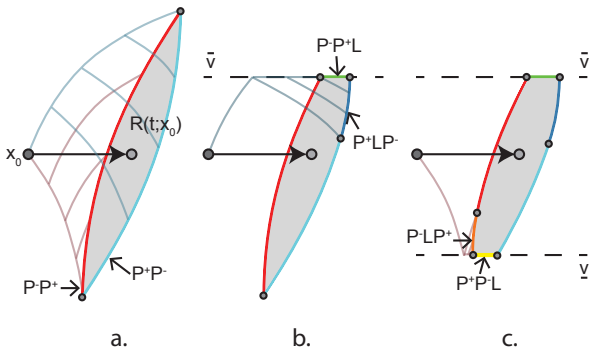

Fig. 2. Reachable sets (shaded) in the PV plane at a new path position from an initial state x_0 fall into Types I, II, and III. PP curves that remain within velocity limits form quadratic left and right boundary segments (left). The boundaries of Type II and III sets also contain horizontal linear segments formed by PPL curves and quadratic segments formed by PLP curves (middle and right).

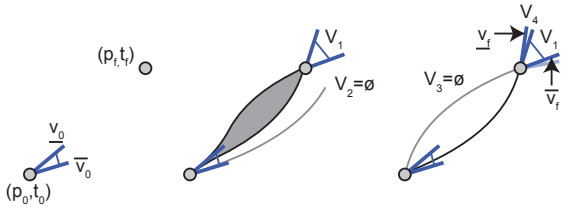

Fig. 3. Velocity interval propagation in the PT plane. Left: the *VIP* subroutine finds the interval of reachable terminal velocities at (p_f,t_f) starting from (p_0,t_0) given bounds $[\underline{v}_0,\bar{v}_0]$ on initial velocity. Middle: connecting the extrema of the initial velocities produces intervals V_1, V_2. Right: maximizing and minimizing goal velocities without imposing initial velocity constraint produces intervals V_3, V_4. The final interval $[\underline{v}_f,\bar{v}_f]$ bounds all four intervals.

$\bar{v}_f$ is defined at the intersection of a boundary curve of $R(t_f;x_0)$ with $p = p_f$. There are only two possible intersections due to the monotonicity of $R(t_f;x_0)$, and in fact only three types of boundary curves need be considered: P^-P^+, P^-LP^+, and the linear segment with $v = \bar{v}$. To handle the *PP* case, we determine the switching time variable t_s by substituting $p = p_f$ into (4) and solving the resulting quadratic equation. Similarly, we handle the *PLP* case by solving for t_s after substituting $p = p_f$ into (6). The linear segment case is solved using a PLP scaling method with unknown scale factor γ (see Lemma 2). After enumerating the three cases, we check velocity limits and validity of switching times, and if no trajectory can be found then (p_f,t_f) is declared unreachable.

C. Velocity Interval Propagation

Now we present the *velocity interval propagation* subroutine *VIP* that forms the backbone of our planner. It takes as input a PT point (p_0,t_0) and a range of starting velocities $[\underline{v}_0,\bar{v}_0]$, and a target point (p_f,t_f). It outputs the range of target velocities $[\underline{v}_f,\bar{v}_f]$ reachable from (p_0,t_0,v) for some $v \in [\underline{v}_0,\bar{v}_0]$ (Fig. 3, left). As a side effect, *VIP* also generates two example trajectories that reach the extrema of both input and output velocity intervals. The planner uses these trajectories later.

VIP proceeds by computing the terminal velocity intervals

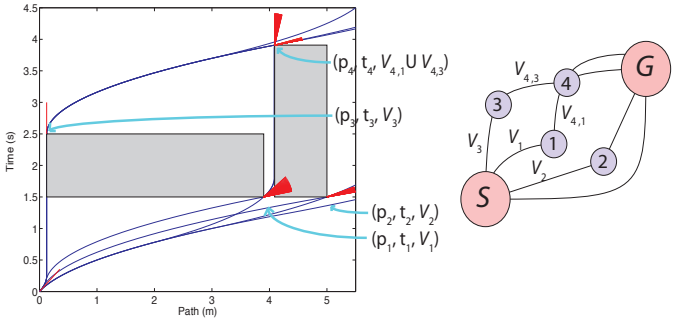

Fig. 4. Left: Example trajectories (curves) are computed by *VIP* to connect to obstacle (shaded) vertices, each marked by a PT coordinate and the set of reachable velocities V at that point, visualized as a wedge. Point 4 can be reached by a disjoint reachable set of velocities due to the large obstacle. Right: Reachability represented as a graph where nodes correspond to obstacle vertices, and edges indicate reachable velocity intervals.

V_1 and V_2 reachable, respectively, from minimum and maximum starting velocity: $V_1 = VReach((p_0,t_0,\underline{v}_0),p_f,t_f)$, and $V_2 = VReach((p_0,t_0,\overline{v}_0),p_f,t_f)$. Note that either V_1, V_2, or both may be empty (Fig. 3, middle). Next, *VIP* maximizes the terminal velocity without input velocity constraint. It does so by constructing a parabolic trajectory with acceleration $\overline{a}$ that interpolates (p_0,t_0) and (p_f,t_f). If the initial velocity v_0^p is valid ($v_0^p \in [\underline{v}_0,\overline{v}_0]$), it sets V_3 to the set containing the final velocity $\{v_f^p\}$, or $V_3 = \emptyset$ otherwise. The minimum ending velocity interval V_4 is similarly computed using a parabola of acceleration $\underline{a}$ (Fig. 3, right). Note that if V_1 is empty and (p_f,t_f) is indeed reachable, then V_3 must be nonempty. Likewise if V_2 is empty then V_4 must be nonempty. Finally *VIP* outputs $[\underline{v}_f,\overline{v}_f]$ as the smallest interval containing $V_1,\ldots,V_4$.

The planner also uses two variants of *VIP*. First, it uses the goal velocity interval propagation (*GVIP*) subroutine to connect a PT point (p_0,v_0) to the goal position p_{goal} with time and velocity constrained within intervals $[0,t_{max}]$ and $[\underline{v}_{goal},\overline{v}_{goal}]$ respectively. The second variant is the reverse velocity interval propagation (*RVIP*) subroutine used in the backward pass of the planner. *RVIP* determines the range of incoming velocities that reach an interval of terminal velocities. Both are very similar to *VIP*, so they will not be discussed in detail.

IV. Reachable Sets With Obstacles

Now we consider reachable sets in the presence of obstacles. In problems with $\underline{v} \geq 0$ and without acceleration constraints, it was shown that optimal collision-free trajectories either connect directly to the goal state or pass tangentially along either the upper-left or lower-right vertex of one or more PT obstacles [14]. This fact holds with acceleration constraints as well [5]. So our algorithm searches among collision-free trajectories that pass through combinations of upper-left and lower-right obstacle vertices. Here we present a recursive planner in a single homotopy class, or "channel," of PT space and prove its correctness; it forms the basis of the general formulation presented in the next section.

A. Reachable Sets With One Obstacle

Before considering multiple obstacles, we illustrate how to determine the set of velocities V_f attainable a target point (p_f,t_f) with a collision-free trajectory from an initial point (p_0,t_0) with initial velocity v_0 in $[\underline{v}_0,\overline{v}_0]$ when a single obstacle O intervenes between the start and target. We will show that V_f is the union of two velocity intervals each reachable via paths through one of the two homotopy classes (above or below the obstacle), and, furthermore, these intervals are determined by propagating reachable velocities through the vertices of the obstacle.

Let $V_{0\to f} = [\underline{v}_f,\overline{v}_f]$ denote the interval of dynamically reachable velocities at (p_f,t_f) from the initial point. Note that obviously $V_f \subseteq V_{0\to f}$. Let *VIP* compute $V_{0\to f}$ as well as two feasible example trajectories $U(t)$ terminating in $\overline{v}_f$ and $L(t)$ terminating in $\underline{v}_f$. We consider computing the sets of terminal velocities V_f^U and V_f^L that flow above and below the obstacle, respectively.

In computing the upper interval V_f^U there are three cases to consider.

1) O lies below $L(t)$ or above $U(t)$.
2) O does not lie below $L(t)$ and lies below $U(t)$.
3) O does not lie below $L(t)$ and intersects $U(t)$.

If O lies below $L(t)$, then $V_f^U = V_{0\to f}$, and if O lies above $U(t)$, then $V_f^U = V_{0\to f}$. In either case we are done. Otherwise O is either above $L(t)$ or intersects it, and hence we are unable to guarantee that $\underline{v}_f$ is reachable. Instead consider trajectories that pass through the upper left vertex of the obstacle (p_{ul},t_{ul}). In case 2, O does not intersect $U(t)$, and hence $\overline{v}_f$ is reachable through the collision-free trajectory $U(t)$. Thus we are simply left with the problem of minimizing the terminal velocity. Consider computing the set of reachable velocities at the obstacle vertex $V_{0\to ul} = VIP((p_0,t_0),V_0,(p_{ul},t_{ul}))$ and then propagating it to the terminal point $V_{ul\to f} = VIP((p_{ul},t_{ul}),V_{0\to ul},(p_f,t_f))$. It can be shown that $V_{ul\to f}$ is nonempty and contains the minimum reachable terminal velocity, and hence $V_f^U = [\min(V_{ul\to f}),\overline{v}_f$. In case 3, O intersects $U(t)$, and hence we are unable to guarantee that $\overline{v}_f$ is reachable. It can be shown that $V_f^U = V_{ul\to f}$ (which may or may not be empty) because any attainable terminal velocity can be also attained by a trajectory through the obstacle vertex. The proofs are special cases of Theorem 1 presented below.

The lower velocity interval V_f^L is computed similarly using the lower-right vertex, and the final reachable set is $V_f = V_f^U \cup V_f^L$, which may consist of up to two disjoint intervals, as seen in Figure 4.

B. Velocity Interval Propagation Through Channels

Now we generalize this argument to propagate reachable velocity intervals through an obstacle "channel," which is the PT space exactly containing a closed set of feasible homotopic trajectories and bounded by an arbitrary number of obstacles.

Let $(p_1,t_1),\ldots,(p_{m-1},t_{m-1})$ be the sequence of upper left and lower right obstacle vertices in the channel such that $t_0 \leq$

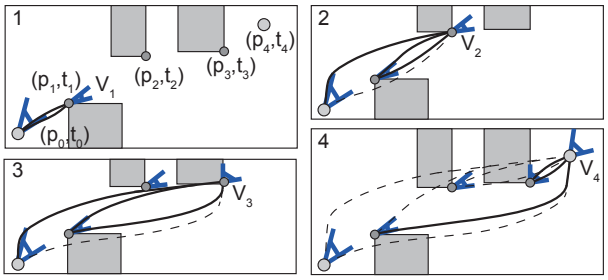

Fig. 5. Incremental construction of reachable sets along a three-obstacle channel using repeated calls of *VIP*. A dynamically-reachable velocity computed by *VIP* contributes to the reachable set if its example trajectory is collision free (solid curves) but can be ignored if it collides with obstacles (dotted curves). Theorem 1 proves this construction is correct.

$p_i \leq t_f$ for all i. Sort the vertices by increasing t coordinate and denote $(p_m,t_m) = (p_f,t_f)$. Define the following recursive quantities for all $0 < k \leq m$:

$$\overline{v}_{i\to k} = \begin{cases} \max(VIP((p_i,t_i),V_i,(p_k,t_k))) & \text{if } V_i \neq \emptyset \text{ and} \\ & Coll(U(t)) = 0 \\ -\infty & \text{otherwise,} \end{cases} \tag{8}$$

$$\underline{v}_{i\to k} = \begin{cases} \min(VIP((p_i,t_i),V_i,(p_k,t_k))) & \text{if } V_i \neq \emptyset \text{ and} \\ & Coll(L(t)) = 0 \\ \infty & \text{otherwise,} \end{cases} \tag{9}$$

$$\overline{v}_k = \max_{0\leq i<k} \overline{v}_{i\to k}, \tag{10}$$

$$\underline{v}_k = \min_{0\leq i<k} \underline{v}_{i\to k}, \tag{11}$$

and

$$V_k = \begin{cases} [\underline{v}_k, \overline{v}_k] & \text{if } \underline{v}_k \leq \overline{v}_k \\ \emptyset & \text{otherwise.} \end{cases} \tag{12}$$

Here $U(t)$ and $L(t)$ are the example trajectories computed by the immediately preceding *VIP* call, and $Coll(p(t))$ reports 0 only if the trajectory segment $p(t)$ is collision free over its domain. Fig. 5 illustrates this construction.

To connect to the goal define $\overline{v}_{i\to goal}$ and $\underline{v}_{i\to goal}$ similarly to (8) and (9) except that the call to *VIP* is replaced by *GVIP*. These quantities are used to define the set V_{goal} by application of (10), (11), and (12). We are now ready to state two theorems.

Theorem 1: $V_f = V_m$ is the set of velocities attainable at (p_f,t_f) over all feasible trajectories in the channel starting at (p_0,t_0).

Proof: The proof proceeds by induction. The base case is $m = 1$, in which case there are no obstacles and by inspection $V_m = VIP((p_0,t_0),V_0,(p_f,t_f)) = V_f$. Now consider the inductive case with $(p_m,t_m) = (p_f,t_f)$, and compare the true set of reachable velocities V_f against V_m given by the expression (12). Obviously $V_m \subseteq V_f$ because each of its endpoints is defined by a feasible trajectory; we will now prove the converse.

The inductive assumption states that the theorem holds for all possible $m-2$ obstacle vertices (p_i,t_i) with $i = 1,\ldots,m-2$ and boundary conditions. So V_i is the set of velocities reachable from (p_0,t_0) for each obstacle vertex (p_i,t_i) for $i = 1,\ldots,m-1$. Furthermore, for $i = 1,\ldots,m-1$ denote $V_{i\to m}$ as the set of velocities reachable at (p_m,t_m) starting at each obstacle vertex (p_i,t_i) with initial velocity interval V_i. Note that $\underline{v}_m$ defined by (11) is equivalent to the minimum of $\underline{v}_{0\to m}$ and $\min_{i=1,\ldots,m-1}\min(V_{i\to m})$, while $\overline{v}_m$ defined by (11) is the maximum of $\overline{v}_{0\to m}$ and $\max_{i=1,\ldots,m-1}\max(V_{i\to m})$.

Consider any feasible trajectory $p(t)$ connecting (p_0,t_0) to (p_m,t_m). We will prove that the terminal velocity $\dot{p}(t_f) \in V_m$ as defined by (12). Let $U(t)$ be the upper example trajectory for the connection between (p_0,t_0) and (p_m,t_m). If $U(t)$ is collision free, then $\overline{v}_{0\to m}$ would be the maximum achievable dynamically feasible velocity, and $\dot{p}(t_f)$ would satisfy $\dot{p}(t_f) \leq \overline{v}_{0\to m}$. If on the other hand, $U(t)$ is in collision with some obstacle, it is possible to construct a feasible trajectory $p_\gamma(t) = \gamma p(t) + (1-\gamma)U(t)$ that is a convex combination of $p(t)$ and $U(t)$ such that $p_\gamma(t)$ touches an obstacle vertex, say (p_i,t_i). Since p_γ is a feasible path starting at (p_i,t_i), we have that $\dot{p}_\gamma(t_f) \in V_{i\to k} \subseteq V_m$, and therefore $\dot{p}_\gamma(t_f) \leq \overline{v}_m$. Since $\dot{U}(t_f) \geq \overline{v}_m$, the fact that p_γ is a convex combination of p and U implies that $\dot{p}(t_f) \leq \overline{v}_m$. Using symmetry, can similarly prove $\dot{p}(t_f) < \underline{v}_m$. Hence $V_f \subseteq V_m$, and therefore $V_f = V_m$. By induction Theorem 1 holds for any $m \geq 1$. ■

Theorem 2: V_{goal} is the set of terminal velocities reached by feasible trajectories through the channel, and the time attained at the maximum velocity in V_{goal} is precisely the duration of the minimum-time trajectory.

The proof is similar to Theorem 1.

V. Polynomial-Time Optimal Planner

Theorems 1 and 2 give rise to a straightforward method for computing the optimal solution to the originally posed problem: enumerate all channels and compute the minimal time for each. The minimum over all channels produces the time of the optimal trajectory and its final velocity; the trajectory itself is recovered by a backwards search through obstacle vertices using *RVIP*. But enumeration is intractable in general because there are potentially 2^n distinct channels for n obstacles, and obstacle configurations that yield at or near this number of channels are not uncommon.

Instead our planner uses a polynomial-time algorithm that iteratively computes reachable sets over vertices. Each iteration accumulates terminal velocity intervals over all equivalent homotopy classes, and then *merges* the intervals before proceeding to the next vertex. The backwards search remains the same. Merging is key; we show that for any point there are far fewer merged velocity intervals ($O(n)$) than homotopy classes ($O(2^n)$), which leads to a running time of $O(n^4)$.

A. Algorithm

Let $(p_0,t_0) = (p_{start},t_{start})$ and $V_0 = \{v_{start}\}$. Let ε be a *resolution threshold* that allows the algorithm to discard tiny velocity intervals that cannot be reached with confidence; for example ε may indicate the measurement uncertainty of the speedometer. The importance of ε will become clear later.

First, sort the upper left and lower right obstacle vertices by increasing t coordinate into the list $(p_1,t_1),\ldots,(p_{2n},t_{2n})$.

For each $i \in [1,2n]$ store a set S_i of velocity intervals and singleton velocities, each annotated by a bit vector denoting the channel of the incoming trajectory. Initialize each S_i to an empty set.

Forward: computes the reachable velocities V_i at each point (p_i,t_i) and at the goal.
1. For $j = 1,\ldots,2n$:
2. For $i = 0,\ldots,j-1$:
3. For each disjoint interval $[a,b]$ in V_i:
4. Call **Propagate**$([a,b],i,j)$
5. $V_j \leftarrow$ **Merge**(S_j)
6. For each disjoint interval $[a,b]$ in V_j:
7. Call **PropagateGoal**$([a,b],j)$

Propagate$([a,b],i,j)$: computes velocities directly reachable at point j starting from point i with velocity $v \in [a,b]$, and adds them to the set S_j with annotation
1. $[a',b'] \leftarrow VIP((p_i,t_i),[a,b],(p_j,t_j))$
2. If $[a',b']$ is empty, return
3. Let $L(t)$ and $U(t)$ be the lower and upper examples
4. $B_L \leftarrow Channel(L(t))$, $B_U \leftarrow Channel(U(t))$
5. If $B_L = B_U \neq nil$, Insert$(([a',b'],B_L),S_j)$
6. Else
7. If $B_L \neq nil$, Insert$((\{a'\},B_L),S_j)$
8. If $B_U \neq nil$, Insert$((\{b'\},B_U),S_j)$

Merge(S): extends velocity intervals in S from same homotopy class, and outputs the union of the reachable velocities.
1. Repeat until S is unchanged:
2. If $\exists([a,b],B),([a',b'],B') \in S$ such that $Suffix?(B,B')$
3. Remove $([a,b],B)$ and $([a',b'],B')$ from S
4. Add $([\min(a,a'),\max(b,b')],B')$ to S
5. Output $V = \bigcup_{([a,b],B) \in S, b-a \geq \varepsilon} [a,b]$

PropagateGoal is very similar to **Propagate** except $GVIP$ is used instead of VIP. **Propagate** makes use of the subroutine $Channel(p(t))$ that returns nil if $p(t)$ collides, and otherwise outputs a bit vector $b = (b_i,\ldots,b_j)$ where b_k indicates 0 if $p(t_k) < p_k$, and is 1 otherwise; b acts as a signature for the channel. **Merge** uses the subroutine $Suffix?(B,B')$ that returns true iff B' is a suffix of B.

B. Complexity Analysis

First we consider the number of disjoint intervals in each V_k. A naïve bound is 2^k because there are at most 2^k distinct channels leading to k. A more involved analysis produces a linear bound after discarding intervals of width less than ε. First we will need a preliminary lemma.

Lemma 3: Consider two initial states $x = (p,v,t)$ and $x' = (p,v',t)$ that differ only in velocity with $v < v'$. Let $V = VReach(x,p',t')$ and $V' = VReach(x',p',t')$ for $t' > t$ and $p' \leq p + (t'-t)v$. If V and V' are nonempty and disjoint, then

$$|V|/2c \geq \sqrt{\sqrt{|V'|/2c} - |V'|/2c} - \sqrt{|V'|/2c} \qquad (13)$$

with $c = (t'-t)(\overline{a}-\underline{a})$ the range of reachable velocities from x at time t'. Furthermore the bounds $|V|/2c \leq 1/4$ and $|V'|/2c \leq (|V|\sqrt{2+\sqrt{2}}/2c)^4 \approx 11.65(|V|/2c)^4$ hold.

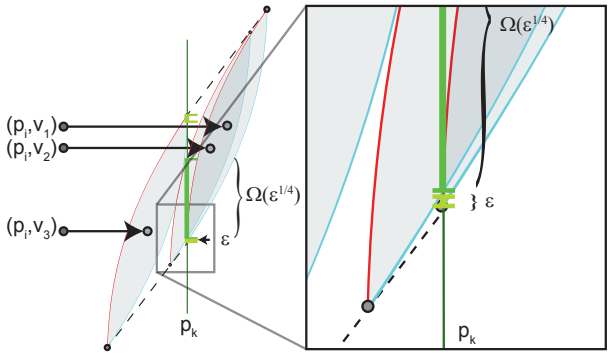

Fig. 6. It is difficult to construct a scenario in which $d > 2$ disjoint reachable velocities to continue to be disjoint after a propagation step. Here three initial velocities at p_i produce three disjoint velocity intervals at p_k. The region denoted with ε demonstrates that the space available for non-overlapping regions is tiny even with $d = 3$.

The proof is rather involved, so we only provide a sketch here. Up to velocity limits, $R(t';x)$ and $R(t';x')$ are identical except for an offset in the P and V axes (see Fig. 6). Consider the conditions under which $|V'|$ space lies between the right boundary of $R(t';x)$ and the line of slope $1/(t'-t)$ underneath (the lower dotted line in Fig. 6). Some algebra shows that to obtain this difference, p' must exceed the leftmost boundary of $R(t',x)$ by at least $2c^2(\sqrt{|V'|/2c} - |V'|/2c)$, and the bound in the theorem follows from algebraic manipulations.

Now we state the key theorem.

Theorem 3: Each V_i contributes $O(d)$ disjoint intervals of width at least ε to V_k, where $d = 2\log_4(-\log_2 \frac{\varepsilon}{2v_{max}}) + 2$ and $v_{max} = t_{max}(\overline{a}-\underline{a})$ bounds the range of achievable velocities.

Proof: Let $v_1,\ldots,v_r$ be a sorted list of velocities each from a single disjoint interval of V_i, such that $p_i + (t_k - t_i)v_q \geq p_k$ for all $q = 1,\ldots,r$. Let $V_1,\ldots,V_r$ be the sets of reachable velocities at (p_k,t_k) respectively starting at $(p_i,v_1,t_i),\ldots,(p_i,v_r,t_i)$. Also let $v_{ik} = (t_k - t_i)(\overline{a} - \underline{a})$ be the range of velocities attainable from a single PVT point after time $t_k - t_i$ has elapsed. By Lemma 3, $|V_{i+1}|/2v_{ik} \leq (|V_i|\sqrt{2+\sqrt{2}}/2v_{ik})^4$. Because $|V_1/2v_{ik}| \leq 1/4$ and $\sqrt{2+\sqrt{2}} < 2$, $|V_r|/2v_{ik} \leq (|V_1|/2v_{ik})^{4^{r-1}} \leq (1/2)^{4^{r-1}}$. So if $|V_r| > \varepsilon$, then

$$\varepsilon/2v_{ik} \leq 2^{-4^{r-1}} \qquad (14)$$

$$\Rightarrow r \leq \log_4(-\log_2(\varepsilon/2v_{max})) + 1 \qquad (15)$$

A similar bound applies for $p_i + (t_k - t_i)v_q \leq p_k$ as well. Since d is at most $2r$, this completes the proof. ■

Because of the double logarithm, the size of the smallest interval shrinks rapidly as d increases (Fig. 6). The 7th and 8th intervals are no larger than $10^{-26}v_{max}$, which implies that for all practical purposes no more than 6 significant disjoint intervals may remain after each propagation. Henceforth we will consider d as a constant.

With the new assumption the planner has $O(n^4)$ time complexity and $O(n^2)$ space complexity. Because each S_j contains $O(j^2)$ elements, steps 1-4 of **Forward** run in time $O(n^4)$. Using a trie data structure indexed by the reverse of

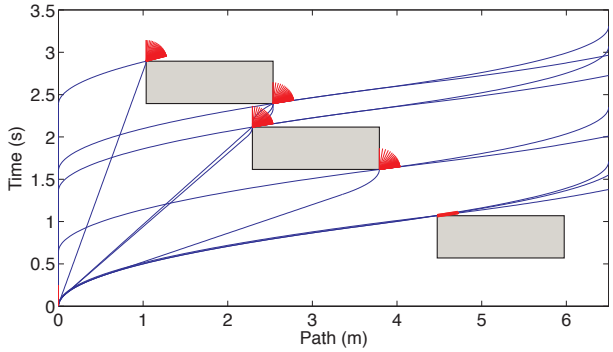

Fig. 7. Three obstacles are arranged randomly, with the reachable sets of velocities (wedges) at each upper-left and lower-right vertex. Here the planner found 21 feasible example trajectories.

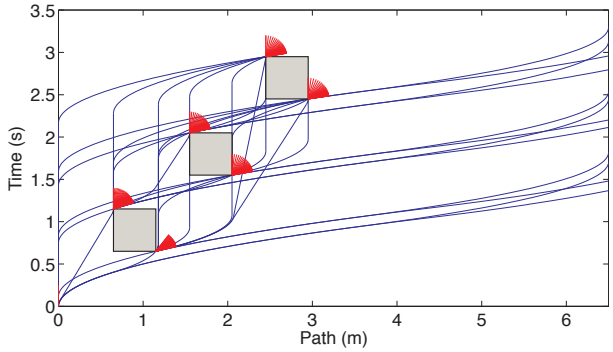

Fig. 8. Three obstacles are arranged in a staircase configuration, with the reachable sets of velocities (wedges) at each upper-left and lower-right vertex. Here the planner found 40 feasible example trajectories.

the bit vector B, **Merge** takes time $O(j^3)$, so the n merge steps of **Forward** take time $O(n^4)$ overall.

C. Experimental Results

We empirically evaluated the performance our implementation of the planner on random (Fig. 7) and the pathological case of the "staircase" configuration, where the obstacles are arranged increasingly in both path and time (Fig. 8). In both cases $O(n^2)$ velocity intervals must be propagated, but in the staircase configuration the visibility graph is almost complete while the in random configurations only 1/3 or 1/5 of the edges are present. Running times on a 2.4GHz PC for $n = 1, \ldots, 15$ obstacles are presented in Table I. In all cases the data were best interpolated by second- or third-degree polynomials with small leading coefficients, which is slightly better than the theoretically expected $O(n^4)$ performance. Our planner is able to handle 100 obstacles at 10Hz.

TABLE I

RUNNING TIMES FOR RANDOM AND STAIRCASE CONFIGURATIONS

Obstacles	1	5	10	15	~Growth
Random 1	0.233ms	1.66ms	4.21ms	7.98ms	$O(0.0002n^3)$
Random 2	0.206ms	1.36ms	3.59ms	5.47ms	$O(0.012n^2)$
Random 3	0.055ms	1.08ms	3.09ms	9.50ms	$O(0.007n^3)$
Staircase	0.244ms	2.77ms	16.6ms	55.9ms	$O(0.021n^3)$

VI. CONCLUSION

This paper presented an optimal, exact, polynomial-time planner for optimal bounded-acceleration trajectories along a fixed, given path with dynamic obstacles. The planner uses a velocity interval propagation technique to compute reachable velocity sets at obstacle vertices in the path-time space, while exploiting the convexity of reachable sets along path-time channels to make planning faster.

Future work may consider more complex vehicle dynamics models that include path-, velocity-, and history-dependent acceleration. We expect that our approach may straightforwardly extend as long as reachable sets are convex. Moreover, the assumption that the vehicle does not influence obstacle behavior is also not appropriate in certain settings like navigating amongst pedestrians at intersections. Some recent work has applied learning to pedestrian navigation scenarios [15] but incorporating vehicle dynamics is still an open problem. Finally, we plan to use a commercial driving simulator to observe how human drivers perform and respond to assisted driving using our system.

REFERENCES

[1] NHTSA, "Traffic safety facts 2008: Older population," Tech. Rep. DOT HS-811-161, National Highway and Safety Administration, 2008.

[2] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: the path-velocity decomposition," *Int. J. Rob. Res.*, vol. 5, pp. 72–89, September 1986.

[3] T. Fraichard, "Dynamic trajectory planning with dynamic constraints: a 'state-time space' approach," in *IEEE/RSJ Int. Conf. Intel. Rob. Sys.*, vol. 2, pp. 1393 – 1400, 1993.

[4] J. Canny, A. Rege, and J. Reif, "An exact algorithm for kinodynamic planning in the plane," in *Proc. 6th Annual Symposium on Computational geometry*, SCG '90, pp. 271–280, 1990.

[5] C. Ó'Dúnlaing, "Motion planning with inertial constraints," *Algorithmica*, vol. 2, no. 1–4, pp. 431–475, 1987.

[6] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *Int. Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[7] S. Anderson, S. Peters, K. Iagnemma, and T. Pilutti, "A unified approach to semi-autonomous control of passenger vehicles in hazard avoidance scenarios," in *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2009.

[8] S. Petti and T. Fraichard, "Safe motion planning in dynamic environments," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 3726–3731, 2005.

[9] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman, O. Koch, Y. Kuwata, D. Moore, E. Olson, S. Peters, J. Teo, R. Truax, and M. Walter, "A perception-driven autonomous urban vehicle," *Journal of Field Robotics*, vol. 25, no. 10, p. 148, 2008.

[10] M. Likhachev and D. Ferguson, "Planning long dynamically feasible maneuvers for autonomous vehicles," *The International Journal of Robotics Research*, vol. 28, no. 8, pp. 933–945, 2009.

[11] V. Delsart, T. Fraichard, and L. Martinez, "Real-time trajectory generation for car-like vehicles navigating dynamic environments," in *Proc. Int. Conf on Robotics and Automation*, pp. 2257–2262, 2009.

[12] J. Peng and S. Akella, "Coordinating multiple robots with kinodynamic constraints along specified paths," *Int. J. Robot. Res*, vol. 24, pp. 295–310, 2005.

[13] M. Erdmann and T. Lozano-Perez, "On multiple moving objects," *Algorithmica*, vol. 2, pp. 1419–1424, 1987.

[14] Y.-H. Liu and S. Arimoto, "Path planning using a tangent graph for mobile robots among polygonal and curved obstacles," *Int. J. Rob. Res.*, vol. 11, pp. 376–382, August 1992.

[15] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *Proc. IEEE/RSJ Int. Conf. on Intel. Robots and Systems (IROS)*, 2010.