A Constant-Time Algorithm for Checking Reachability of Arrival Times and Arrival Velocities of Autonomous Vehicles

Ty Nguyen[1] and Tsz-Chiu Au[1]

***Abstract*— A fast algorithm for checking whether an autonomous vehicle can arrive at a position at a given arrival time *and* velocity is the key to Autonomous Intersection Management (AIM). This paper presents a complete set of closed form equations that fully describes the set of all reachable arrival configurations in longitudinal motion planning if the vehicle's controller is a double integrator with bounded acceleration. This result improves the running time of the algorithm for checking the reachability of an arrival configuration from logarithmic time to constant time. We also apply the result to check the reachability in a segmented road and discuss how the algorithm can be applied to real vehicles.**

I. INTRODUCTION

Motion planning is PSPACE-hard in general [1], [2]. Hence, it is unlikely to find a complete algorithm that guarantees to find a solution in a reasonable time. Most practical motion planning algorithms are sampling-based algorithms that are incomplete (e.g., probabilistic roadmap methods (PRM) [3] and rapidly-exploring random trees (RRT) [4]). However, a guarantee in finding a motion plan if one exists can be useful in certain applications. For example, Au and Stone [5] showed that the ability for an autonomnous vehicle to arrive at an intersection at a given time and velocity is crucial to autonomous intersection management (AIM), a new intersection control protocol that intends to replace traffic signals when most vehicles are autonomous in the future. More specifically, they showed that if vehicles can determine their unreachability to arrive at an intersection at a specific time and at a specific velocity as early as possible, the efficiency of AIM can be greatly improved [6].

Deciding the existence or non-existence of a motion plan is difficult because this often requires an exhaustive exploration of the entire configuration space. An even more difficult problem is to describe the set of *all* configurations in a configuration space that are reachable by some motion plans. Let us call a set of all reachable configurations a *reachable set*. Computing reachable sets can be very useful because a mathematical description of the set of all reachable configurations eliminates the need to search for reachable configurations to check the existence or non-existence of a motion plan. Furthermore, if we can describe a reachable set geometrically in a configuration space by a set of closed-form equations, we can perform geometrical operations on the *entire* set of reachable configurations, such as checking whether two reachable sets intersect. However, it is not easy to come up with these equations because, apart from the fact that the equations can be quite complicated, the "shape" of a reachable set also depends on parameters in the planning problem. Different parameter values can lead to different reachable sets whose equations are totally different, and there can be potentially an infinite number of possible values of the parameters in a planning problem.

In this paper, we address this challenge and provide a complete set of equations describing the set of all possible arrival configurations when controlling an autonomous vehicle to arrive at a specific position on a trajectory. Here the configuration space is the pairs of arrival time and arrival velocity at the destination. The immediate corollary of this result is the reduction of the running time of the algorithm for checking the reachability of an arrival configuration from logarithmic time [7] to constant time. This speed improvement is important because this checking procedure is repeatedly run every few milliseconds to check whether the arrival time and velocity to an intersection remains reachable. If the arrival time and velocity becomes unreachable due to the control noise when driving on a bumpy road, the vehicle should not enter the intersection to avoid collision. A constant time algorithm allows a much higher frequency of checking, thus enhancing safety. The algorithm can also be extended to deal with multiple road segments, unlike the previous algorithm that only works for one road segment.

Some previous motion planning algorithms are also based on reachable sets for autonomous vehicles [8], robot arms [9], and humanoid robots [10]. Among these works, few consider meeting both time and velocity objectives simultaneously. One of these works is the path-velocity-time planner proposed by Johnson and Hauser, which computes a reachable set that denotes the range of reachable velocities after the vehicle travels for a specific time in longitudinal motion [8], [11]. However, their works are different from ours since their algorithm (more specifically, the velocity interval propagation subroutine) assumes the arrival times are fixed and outputs a range of target velocities reachable from an initial point (p_0, t_0) to a target point (p_f, t_f) in a path-time (PT) plane. Notice that a reachable set of velocities in PVT diagrams (positions, velocity, and time) in [8] is different from a reachable set of time-velocity pairs in VT diagrams (velocity and time). Unlike reachable sets of velocities in PVT diagrams, reachable sets of time-velocity pairs are usually non-convex and hence they are far more complicated than the reachable sets in [8]. Since the arrival times are fixed, the planners based on the reachable sets defined in [8] and [11] cannot directly be used to solve our problem.

[1]School of Electrical and Computer Engineering, Ulsan National Institute of Science and Technology (UNIST), Ulsan, South Korea. {tynguyen,chiu}@unist.ac.kr

II. RELATED WORK

Recent work on motion planning focuses on planning in limited domains or exploiting the problem structure. Švestka and Vleugels [12] gave an exact motion planning algorithm for tractor-trailer robot in the absence of obstacles. Halperin [13] concerns with the geometric algorithms for robust primitives for complete motion planning. Varadhan et al. [14] described a complete algorithm for motion planning of translating polyhedral robots in 3D. Most of these planners concern with checking whether a robot can arrive at a position with a correct final orientation. However, since these algorithms aims to solve motion planning problems in general, they run in exponential time in the worst cases.

Probabilistic roadmap methods (PRM) [3] and rapidly-exploring random trees [15] are both widely used, sampling-based algorithms. While these algorithms are probabilistically complete under very general conditions [4], they are actually incomplete algorithms because there is no guarantee that they find a solution if one exists. Hirsch and Halperin [16] proposed a hybrid motion planner that generates complete solutions with PRM. However, these modified algorithms suffer from inefficiency due to their completeness.

Longitudinal control of autonomous/semi-autonomous vehicles has been widely studied since the 1960's, in particular in platooning in automated highway systems [17]. These studies mainly focus on car following in a platoon [18], but our approach is more suitable for point following [19]. Most work on motion planning for autonomous vehicles (e.g., [20]) treated the arrival time and velocity requirements as secondary. But finding optimal arrival times and velocities is an important issue in some applications [5], [6]. Au and Stone studied the longitudinal control problem on real vehicles [7], but they assumed that the maximum and minimum accelerations remain constant all the time [5].

Some previous works on motion planning focus exclusively one-dimensional trajectories. For example, Bobrow et al. [21] dealt with the minimum-time manipulator control problem, which is about controlling a robot manipulator to move along a specified path in a time-optimal manner, subject to the actuator torque constraints. Kunz and Stilman [22] dealt with a similar problem using the same approach with a path preprocessing step. Some general-purpose motion planners are capable of satisfying *both* the arrival time and arrival velocity requirements. For example, Johnson and Hauser [8] presented a polynomial-time, complete planner that computes collision-free, time-optimal, longitudinal control sequences for meeting arrival time and velocity requirements, via the computation of the reachable sets in the path-velocity-time space. As discussed in Section I, their problem as well as their solutions are different from ours as velocity interval propagation in their works assumes arrival times are fixed, and their planner has to enumerate arrival times using the problem structure. By contrast, we directly compute the reachable sets of both arrival times and arrival velocities simultaneously. Johnson and Hauser [11] improved their previous work by allowing non-rectangular obstacles in position-velocity-time space. [23] and [24] also considered the computation of reachable sets for double integrators for motion planning for manipulators.

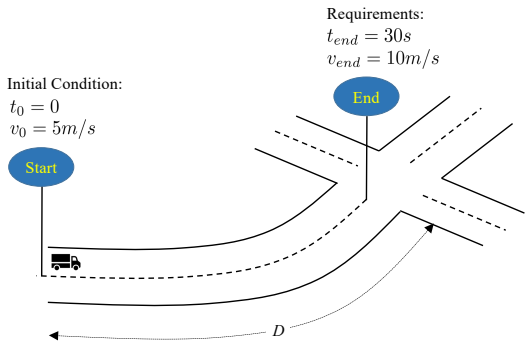

Fig. 1. The longitudinal motion planning problem

III. LONGITUDINAL MOTION PLANNING

We define our problem as a longitudinal motion planning problem of autonomous vehicles, even though our problem formulation is applicable to any motion planning of mobile robots on 1-D trajectories. In Fig. 1, a vehicle is located at the starting position at time $t_0 = 0$ and has an *initial velocity* v_0. Our objective is to control the vehicle to reach the destination (the "End" sign) at a given *arrival time* t_{end} and at a given *arrival velocity* v_{end}, subject to the speed limits and the vehicle's acceleration constraints. The distance between the starting position and the destination is D. Let v_{max} be the speed limit such that the velocity of the vehicle cannot exceed v_{max} at any point in time. Let a_{max} and a_{min} be the absolute values of the maximum acceleration and the maximum deceleration, respectively, such that the vehicle cannot accelerate more than a_{max} or decelerate more than a_{min} at any point on the road.

We define 1) the *initial configuration* as (t_0, v_0), 2) the *arrival configuration* as $(t_{\mathsf{end}}, v_{\mathsf{end}})$, and 3) the *road configuration* as $(D, a_{\mathsf{max}}, a_{\mathsf{min}}, v_{\mathsf{max}})$. A *longitudinal motion planning problem* $\mathscr{P}_{\mathsf{valid}}$ is a 3-tuple $\langle (t_0, v_0), (t_{\mathsf{end}}, v_{\mathsf{end}}), (D, a_{\mathsf{max}}, a_{\mathsf{min}}, v_{\mathsf{max}}) \rangle$, where $t_0 = 0$, $0 \leq v_0 \leq v_{\mathsf{max}}$, $0 < t_{\mathsf{end}}$, $0 \leq v_{\mathsf{end}} \leq v_{\mathsf{max}}$, $0 < D$, $a_{\mathsf{max}} \geq 0$, $a_{\mathsf{min}} \geq 0$, and $0 < v_{\mathsf{max}}$. Our task is to generate a sequence of *control signals* such that if the vehicle follows the sequence exactly, it will reach the destination while satisfying all requirements and constraints. There are many different type of vehicle controllers, but for *velocity-based* controllers, the sequence of control signals is a *velocity function* $v(\cdot)$, such that the controller will set the velocity of the vehicle according to $v(\cdot)$ over time. We are only interested in non-negative velocity functions because we forbid a vehicle to move backward. We say $v(\cdot)$ is *reachable* if it satisfies the following constraints:

C1) $v(t_0) = v(0) = v_0$;
C2) $v(t_{\mathsf{end}}) = v_{\mathsf{end}}$;
C3) $0 \leq v(t) \leq v_{\mathsf{max}}$ for $t_0 \leq t < t_{\mathsf{end}}$ (i.e., the velocity cannot exceed the speed limit or be negative at any point in time);
C4) $\int_{t_0}^{t_{\mathsf{end}}} v(t)\,dt = D$ (i.e., the distance traveled must be D_i); and

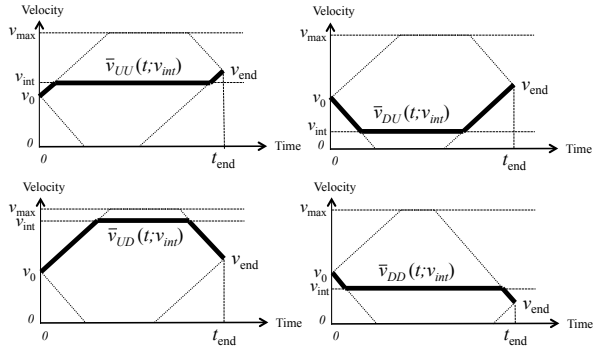

Fig. 2. The four canonical velocity functions.

C5) $-a_{\mathsf{min}} \leq v'(t^-) \leq a_{\mathsf{max}}$ and $-a_{\mathsf{min}} \leq v'(t^+) \leq a_{\mathsf{max}}$, where $v'(t^-)$ is the left derivative of $v(\cdot)$ at t, $v'(t^+)$ is the right derivative of $v(\cdot)$ at t, and $t_0 < t < t_{\mathsf{end}}$ (i.e., the acceleration and the deceleration must be within the limitations when moving on the road).

The objective of a longitudinal motion planning problem $\mathscr{P}_{\mathsf{valid}}$ is to check whether a reachable velocity function $v(\cdot)$ exists. In other words, $\mathscr{P}_{\mathsf{valid}}$ is called an instance of the *validation problem*, in which we want to *validate* the given arrival configuration $(t_{\mathsf{end}}, v_{\mathsf{end}})$ by checking whether $(t_{\mathsf{end}}, v_{\mathsf{end}})$ is reachable by a reachable velocity function.

IV. REACHABLE SETS

Let us first consider the case of one road segment: given an initial configuration (t_0, v_0) and a road segment configuration $(D, a_{\mathsf{max}}, a_{\mathsf{min}}, v_{\mathsf{max}})$, we want to find the set F of all reachable arrival configurations. We simply call F a *reachable set*. Previously, Au and Stone have provided a validation procedure for problems with exactly one road segment [5]. However, their algorithm can only check whether the arrival configuration is a member of the reachable set. By contrast, we want to find not just one but *all* members in F. Nonetheless, Au and Stone provided a hint for us to construct a reachable set [5]. Fig. 2 shows four velocity functions: $\bar{v}_{UU}(t;v_{\mathsf{int}})$, $\bar{v}_{DU}(t;v_{\mathsf{int}})$, $\bar{v}_{UD}(t;v_{\mathsf{int}})$, and $\bar{v}_{DD}(t;v_{\mathsf{int}})$. We will call them *canonical* velocity functions. $\bar{v}_{UD}(t;v_{\mathsf{int}})$ is also called an "up-down" velocity function, which instructs the vehicle to immediately accelerate to an *intermediate velocity* v_{int} at t_0 using the maximum acceleration a_{max}, and then maintain the velocity at v_{int} until the last moment at which the vehicle can decelerate using a_{min} to reach the destination at the given t_{end} and v_{end}. Likewise, each of $\bar{v}_{UU}(t;v_{\mathsf{int}})$, $\bar{v}_{DU}(t;v_{\mathsf{int}})$, and $\bar{v}_{DD}(t;v_{\mathsf{int}})$ has one parameter—the intermediate velocity v_{int}—and will instruct the vehicle to accelerate or decelerate to v_{int} and maintain the speed as long as possible. These canonical functions are significant due to Theorem 1.

Theorem 1: If a reachable velocity function $v(\cdot)$ exists for a validation problem with one road segment only, there exists a canonical velocity function $\bar{v}(\cdot;v_{\mathsf{int}})$ for some intermediate velocity v_{int} such that $\bar{v}(\cdot;v_{\mathsf{int}})$ is also reachable.

An informal proof of Theorem 1 is given in [5], though [5] has not stated this theorem formally. The theorem implies that there is no need to check all possible reachable velocity functions in the validation problem—it is sufficient to check whether one of the four canonical velocity functions exists and is reachable. This result significantly reduces the complexity of the validation problem, as discussed in [5].

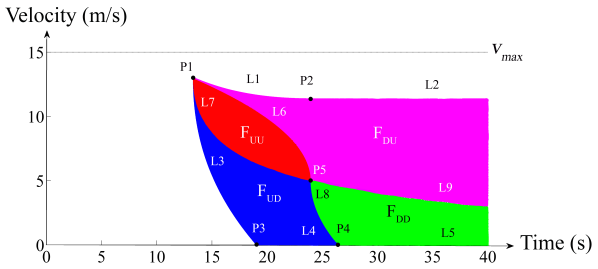

Fig. 3. The reachable set $F = (F_{\mathsf{UU}} \cup F_{\mathsf{UD}} \cup F_{\mathsf{DU}} \cup F_{\mathsf{DD}})$.

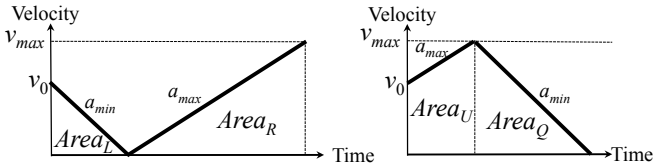

Fig. 4. Special values that dictates the boundary of F.

First of all, we identify some interesting structures in the reachable set based on Theorem 1. Let F_{UU} be the reachable set of arrival configurations that are reachable by using $\bar{v}_{UU}(t;v_{\mathsf{int}})$ only. Similarly, let F_{UD}, F_{DU}, and F_{DD} be the reachable sets using $\bar{v}_{UD}(t;v_{\mathsf{int}})$, $\bar{v}_{DU}(t;v_{\mathsf{int}})$, and $\bar{v}_{DD}(t;v_{\mathsf{int}})$, respectively. Theorem 1 infers that F is the union of the four reachable sets: $F = (F_{\mathsf{UU}} \cup F_{\mathsf{UD}} \cup F_{\mathsf{DU}} \cup F_{\mathsf{DD}})$.

Fig. 3 is a time-velocity diagram showing the reachable set F when $v_0 = 5m/s$, $t_{\mathsf{end}} = 40s$, $D = 120m$, $a_{\mathsf{min}} = 1m/s^2$, $a_{\mathsf{max}} = 0.6m/s^2$, and $v_{\mathsf{max}} = 15m/s$. As can be seen, F can be divided into four regions: F_{UU}, F_{UD}, F_{DU}, and F_{DD}. All four regions meet at a point P5, which is reachable by a constant velocity function $v(t) = 5$ for $24s$. P5 exists as long as $v_0 > 0$; when $v_0 = 0$, the vehicle cannot start to decelerate as negative velocity is prohibited, and F_{DU} and F_{DD} do not exist. Lines L6, L7, L8 and L9, which radiate from P5, are sets of arrival configurations shared by the adjacent reachable sets. F_{UU}, F_{UD}, F_{DU}, and F_{DD} overlap only on these lines. For example, $F_{\mathsf{UU}} \bigcap F_{\mathsf{DU}} = L6$. The interiors of F_{UU}, F_{UD}, F_{DU}, and F_{DD}, together with L6, L7, L8, and L9, form a subdivision of F.

We are interested in the boundary of F, which encloses all sets of reachable configurations. The fact that the boundary of F is a combination of some boundaries of F_{UU}, F_{UD}, F_{DU}, and F_{DD} can be used to deduce the closed-form expressions of the set of equations describing the boundary of F. However, the shape of F depends on the initial configuration and the road segment configuration, and in some cases F is infinite. It is necessary to enumerate all possible cases in which the set of equations differ from each other. Let consider the following four areas that are visualized in Fig. 4:

- $Area_L$ is the distance the vehicle travels from v_0 with a deceleration of a_{min} until a complete stop;
- $Area_R$ is the distance the vehicle travels from a complete stop with an acceleration of a_{max} until it hits the speed

Case 1: $D \le Area_L$ **and** $D \le Area_U$

t	$t < t_2^d$	$t_2^d \le t \le t_1^d$	$t_1^d < t$
$\Omega^{\mathsf{upper}}(t)$		$h_1^d(t)$	

t	$t < t_2^d$	$t_2^d \le t \le t_1^d$	$t_1^d < t$
$\Omega^{\mathsf{lower}}(t)$		$g_1^u(t)$	

Case 2: $D \le Area_L$ **and** $D > Area_U$

t	$t < t_2^u$	$t_2^u \le t \le t_3^d$	$t_3^d \le t \le t_1^d$	$t_1^d < t$
$\Omega^{\mathsf{upper}}(t)$		v_{max}	$h_1^d(t)$	

t	$t < t_2^u$	$t_2^u \le t \le t_1^u$	$t_1^u \le t \le t_1^d$	$t_1^d < t$
$\Omega^{\mathsf{lower}}(t)$		$g_2^u(t)$	$g_1^u(t)$	

Case 3: $D \ge Area_L$ **and** $D \le Area_U$

t	$t < t_2^d$	$t_2^d \le t \le t_5^d$	$t_5^d \le t$
$\Omega^{\mathsf{upper}}(t)$		$h_1^d(t)$	$h_2^d(t)$

t	$t < t_2^d$	$t_2^d \le t \le t_4^u$	$t_4^u \le t$
$\Omega^{\mathsf{lower}}(t)$		$g_1^u(t)$	0

Case 4: $D \ge \max\{Area_L,\ Area_U\}$ **and** $D \le \min\{Area_L + Area_R,\ Area_U + Area_Q\}$

t	$t < t_2^u$	$t_2^u \le t \le t_3^d$	$t_3^d \le t \le t_5^d$	$t_5^d \le t$
$\Omega^{\mathsf{upper}}(t)$		v_{max}	$h_1^d(t)$	$h_2^d(t)$

t	$t < t_2^u$	$t_2^u \le t \le t_1^u$	$t_1^u \le t \le t_4^u$	$t_4^u \le t$
$\Omega^{\mathsf{lower}}(t)$		$g_2^u(t)$	$g_1^u(t)$	0

Case 5: $D \ge Area_L + Area_R$ **and** $D \le Area_U + Area_Q$

t	$t < t_2^u$	$t_2^u \le t$
$\Omega^{\mathsf{upper}}(t)$		v_{max}

t	$t < t_2^u$	$t_2^u \le t \le t_1^u$	$t_1^u \le t \le t_4^u$	$t_4^u \le t$
$\Omega^{\mathsf{lower}}(t)$		$g_2^u(t)$	$g_1^u(t)$	0

Case 6: $D \le Area_L + Area_R$ **and** $D \ge Area_U + Area_Q$

t	$t < t_2^u$	$t_2^u \le t \le t_3^d$	$t_3^d \le t \le t_5^d$	$t_5^d \le t$
$\Omega^{\mathsf{upper}}(t)$		v_{max}	$h_1^d(t)$	$h_2^d(t)$

t	$t < t_2^u$	$t_2^u \le t \le t_5^u$	$t_5^u \le t$
$\Omega^{\mathsf{lower}}(t)$		$g_2^u(t)$	0

Case 7: $D \ge \max\{Area_L + Area_R,\ Area_U + Area_Q\}$

t	$t < t_2^u$	$t_2^u \le t$
$\Omega^{\mathsf{upper}}(t)$		v_{max}

t	$t < t_2^u$	$t_2^u \le t \le t_5^u$	$t_5^u \le t$
$\Omega^{\mathsf{lower}}(t)$		$g_2^u(t)$	0

Fig. 5. The upper boundaries and the lower boundaries of the reachable sets in seven different cases.

limit v_{max};

- $Area_U$ is the distance the vehicle travels from v_0 with an acceleration of a_{max} until it hits the speed limit v_{max}; and
- $Area_Q$ is the distance the vehicle travels from v_{max} with a deceleration of a_{min} until a complete stop.

It turns out that these values play a critical role in identifying different cases as their combinations serve as interval values that shape the set of all reachable points of each *canonical* velocity function. For example, if the road distance D is less that $Area_L$, there is no reachable velocity function $\bar{v}_{DU}(t; v_{\mathsf{int}})$ that can end up with the arrival velocity $v_{\mathsf{end}} = 0$. Similarly, $Area_U$ is the minimum value of the road segment such that there exists a velocity function $\bar{v}_{UD}(t; v_{\mathsf{int}})$ ending up with the arrival velocity $v_{\mathsf{end}} = v_{\mathsf{max}}$. Likewise, $Area_L + Area_R$ is the lower bound condition for the existence of $\bar{v}_{UD}(t; 0)$ ending at $v_{\mathsf{end}} = v_{\mathsf{max}}$, and $Area_U + Area_Q$ is the lower bound condition for the existence of $\bar{v}_{UD}(t; v_{\mathsf{max}})$ ending at $v_{\mathsf{end}} = 0$.

We need to identify all possible relationships between D, $Area_L$, $Area_U$, $Area_L + Area_R$ and $Area_U + Area_Q$. In general, the number of possible ways to compare the 5 values is $5! = 120$. Fortunately, from Fig. 4 we identify some additional relationships among those four values: 1) $Area_L \le Area_L + Area_R$; 2) $Area_U \le Area_U + Area_Q$; 3) $Area_L \le Area_Q \le Area_U + Area_Q$; and 4) $Area_U \le Area_R \le Area_L + Area_R$. Basically, these relationships imply that both $Area_L$ and $Area_U$ are less than or equal to $Area_L + Area_R$ and $Area_U + Area_Q$. This helps us reduce the number of different cases to seven:

- Case 1: $D \le Area_L$ and $D \le Area_U$;
- Case 2: $D \le Area_L$ and $D \ge Area_U$;
- Case 3: $D \ge Area_L$ and $D \le Area_U$;
- Case 4: $D \ge \max\{Area_L,\ Area_U\}$ and $D \le \min\{Area_L + Area_R,\ Area_U + Area_Q\}$;
- Case 5: $D \ge Area_L + Area_R$ and $D \le Area_U + Area_Q$;
- Case 6: $D \le Area_L + Area_R$ and $D \ge Area_U + Area_Q$; and
- Case 7: $D \ge \max\{Area_L + Area_R,\ Area_U + Area_Q\}$.

We derived the equations for the upper bound $\Omega^{\mathsf{upper}}(t)$ and the lower bound $\Omega^{\mathsf{lower}}(t)$ of the reachable set in these seven cases. These equations, shown in Fig. 5 and 6, are inferred from an analysis of the aforementioned canonical velocity functions, but due to the lack of space we omit the analysis.

$$h_1^d(t) = v_0 - a_{\mathsf{min}} t + \sqrt{(a_{\mathsf{min}} + a_{\mathsf{max}})(a_{\mathsf{min}} t^2 - 2v_0 t + 2D)}$$

$$h_2^d(t) = \sqrt{\left(\frac{a_{\mathsf{max}}}{a_{\mathsf{min}}}\right)\left(2a_{\mathsf{min}} D - v_0^2\right)}$$

$$g_1^u(t) = v_0 + a_{\mathsf{max}} t - \sqrt{(a_{\mathsf{min}} + a_{\mathsf{max}})(a_{\mathsf{max}} t^2 + 2v_0 t - 2D)}$$

$$g_2^u(t) = v_{\mathsf{max}} - \sqrt{\left(\frac{a_{\mathsf{min}}}{a_{\mathsf{max}}}\right)[2a_{\mathsf{max}} v_{\mathsf{max}} t - (v_{\mathsf{max}} - v_0)^2 - 2a_{\mathsf{max}} D]}$$

$$t_1^d = \frac{v_0 - \sqrt{v_0^2 - 2a_{\mathsf{min}} D}}{a_{\mathsf{min}}}; t_2^d = \frac{-v_0 + \sqrt{v_0^2 + 2a_{\mathsf{max}} D}}{a_{\mathsf{max}}}$$

$$t_3^d = \frac{v_0}{a_{\mathsf{min}}} + \frac{v_{\mathsf{max}}}{a_{\mathsf{max}}} - \sqrt{\left(\frac{a_{\mathsf{min}} + a_{\mathsf{max}}}{a_{\mathsf{min}}^2 a_{\mathsf{max}}^2}\right)\left(a_{\mathsf{max}} v_0^2 + a_{\mathsf{min}} (v_{\mathsf{max}})^2 - 2a_{\mathsf{min}} a_{\mathsf{max}} D\right)}$$

$$t_5^d = \frac{v_0}{a_{\mathsf{min}}} + \sqrt{\frac{2a_{\mathsf{min}} D - v_0^2}{a_{\mathsf{min}} a_{\mathsf{max}}}}; t_2^u = \frac{(v_{\mathsf{max}} - v_0)^2 + 2a_{\mathsf{max}} D}{2a_{\mathsf{max}} v_{\mathsf{max}}}$$

$$t_1^u = \left(\frac{1}{a_{\mathsf{min}}} + \frac{1}{a_{\mathsf{max}}}\right) v_{\mathsf{max}} - \frac{v_0}{a_{\mathsf{max}}} - \sqrt{\frac{(a_{\mathsf{min}} + a_{\mathsf{max}})(v_{\mathsf{max}})^2 - a_{\mathsf{min}} v_0^2 - 2a_{\mathsf{min}} a_{\mathsf{max}} D}{a_{\mathsf{min}}^2 a_{\mathsf{max}}}}$$

$$t_4^u = -\frac{v_0}{a_{\mathsf{max}}} + \sqrt{\left(\frac{a_{\mathsf{min}} + a_{\mathsf{max}}}{a_{\mathsf{min}} a_{\mathsf{max}}^2}\right)\left(v_0^2 + 2a_{\mathsf{max}} D\right)}$$

$$t_5^u = \left(\frac{1}{a_{\mathsf{min}}} + \frac{1}{a_{\mathsf{max}}}\right)\left(\frac{v_{\mathsf{max}}}{2}\right) - \frac{v_0}{v_{\mathsf{max}}} + \frac{v_0^2 + 2a_{\mathsf{max}} D}{2a_{\mathsf{max}} v_{\mathsf{max}}}$$

Fig. 6. The equations in Fig. 5.

V. Improved Validation Algorithms

This section discusses two applications in which the exact equations of reachable sets are useful.

A. A Constant-Time Algorithm for the Validation Problem

Given a road segment's configuration $(D, a_{\mathsf{max}}, a_{\mathsf{min}}, v_{\mathsf{max}})$, an initial configuration (t_0, v_0), and an arrival configuration $(t_{\mathsf{end}}, v_{\mathsf{end}})$, we can use the tables in Fig. 5 to check whether $(t_{\mathsf{end}}, v_{\mathsf{end}})$ is reachable in constant time, which is faster than the bisection method proposed in [7]. First, we determine which case it belongs to and find the equations of $\Omega^{\mathsf{upper}}(t)$ and $\Omega^{\mathsf{lower}}(t)$ in the tables in Fig. 5 and 6. For instance, take a look at the example in Fig. 3. We have $Area_L = \frac{v_0^2}{2a_{\mathsf{min}}} =$

$12.5m$; $Area_R = \frac{v_{\max}^2}{2a_{\max}} = 187.5m$; $Area_Q = \frac{v_{\max}^2}{2a_{\min}} = 112.5m$; and $Area_U = \frac{v_{\max}^2 - v_0^2}{2v_{\max}} = 166.7m$. Hence this is Case 3 since $D \geq Area_L$ and $D \leq Area_U$. The equation of the upper bound is

$$\Omega^{\text{upper}}(t) = \begin{cases} \text{undefined} & \text{if } t < 13.3 \\ g_1(t) & \text{if } 13.3 \leq t \leq 23.9 \\ 11.4 & \text{if } 23.9 \leq t \end{cases}$$

where $g_1(t) = 5 - t + \sqrt{1.6t^2 - 16t + 384}$. This corresponds to L1 and L2 in Fig.3. We also have

$$\Omega^{\text{lower}}(t) = \begin{cases} \text{undefined} & \text{if } t < 13.3 \\ g_2(t) & \text{if } 13.3 \leq t \leq 19.1 \\ 0 & \text{if } 19.1 \leq t \end{cases}$$

where $g_2(t) = 5 + 0.6t - \sqrt{0.96t^2 + 16t - 384}$. This corresponds to L3 and L4 $\cup$ L5 in Fig. 3.

Second, since all reachable configurations are enclosed between $\Omega^{\text{upper}}(t)$ and $\Omega^{\text{lower}}(t)$, we can plug t_{end} into $\Omega^{\text{upper}}(t)$ and $\Omega^{\text{lower}}(t)$ to compute the valid range of v_{end}. For example, if $t_{\text{end}} = 18$ and $v_{\text{end}} = 5$, $\Omega^{\text{lower}}(t_{\text{end}}) = g_2(t_{\text{end}}) = 1.14$ and $\Omega^{\text{upper}}(t_{\text{end}}) = g_1(t_{\text{end}}) = 11.79$, and hence the arrival configuration $(18,5)$ is reachable. In short, $(t_{\text{end}}, v_{\text{end}})$ is reachable if and only if $\Omega^{\text{lower}}(t_{\text{end}}) \leq v_{\text{end}} \leq \Omega^{\text{upper}}(t_{\text{end}})$. The running time for checking this condition is $O(1)$.

B. Validation for Two Road Segments

When the road segment before an intersection is too short, a vehicle may have to start validating the reachability of the arrival time and velocity on the previous road segment. Let us consider the validation for two consecutive road segments. Suppose a road consists of two segments R_1 and R_2. Let F_1 be the set of all reachable configurations at the end of the first road segment R_1 given that the initial configuration is (t_0, v_0). Let F_2' be the set of all starting configurations at the beginning of the second road segment R_2 from which there exists a velocity function to reach the end of the second road segment with the arrival configuration $(t_{\text{end}}, v_{\text{end}})$.

Theorem 2: Given two road segments R_1 and R_2, a reachable velocity function exists *if and only if* F_1 intersects F_2' (i.e., $F_1 \cap F_2' \neq \emptyset$).

Fig. 7 gives an example showing the intersection of F_1 and F_2'. Here is the reason why this theorem is correct: if F_1 intersects with F_2', we can pick one point (t', v') in $(F_1 \cap F_2')$ and construct a reachable velocity function by connecting $(0, t_1)$ to $(t_{\text{end}}, v_{\text{end}})$ via (t', v') using two velocity functions (e.g., find two canonical velocity functions, one connecting $(0, t_1)$ to (t', v') and the other connecting (t', v') to $(t_{\text{end}}, v_{\text{end}})$, using the method outlined in in [5]). If F_1 does not intersect F_2', there is *no* reachable velocity function that can connect $(0, t_1)$ to $(t_{\text{end}}, v_{\text{end}})$, because F_1 and F_2' are *complete* sets of reachable configurations—all configurations outside F_1 or F_2' are *not* reachable from $(0, t_1)$ or $(t_{\text{end}}, v_{\text{end}})$, respectively.

The boundary of F_2' can be computed by assuming the vehicle is moving backward in time from the destination to the beginning of R_2. First, set the initial configuration to $(0, v_{\text{end}})$ and swap the roles of $a_{\max}$ and $a_{\min}$. Second, obtain the boundaries of F_2'' from the tables in Fig. 5. Third, replace t with $t_{\text{end}} - t$ in the equations of the boundaries of F_2'' to obtain a new set of equations of the boundaries of F_2'.

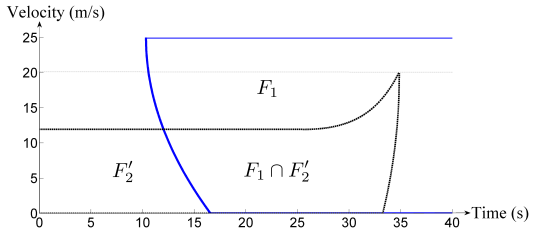

Fig. 7. Reachable sets for two segments.

Let $\Omega_1^{\text{upper}}(t)$ and $\Omega_1^{\text{lower}}(t)$ be the upper bound and the lower bound of F_1, respectively. Let $\Omega_2^{\text{upper}}(t)$ and $\Omega_2^{\text{lower}}(t)$ be the upper bound and the lower bound of F_2', respectively. F_1 and F_2' can intersect only in one of the following ways: either 1) one of them is enclosed entirely in another, or 2) their boundaries intersect. In the former case, we can simply pick one point in F_1 and check whether it is bounded between $\Omega_2^{\text{upper}}(t)$ and $\Omega_2^{\text{lower}}(t)$. If not, pick another point in F_2 and check whether it is bounded between $\Omega_1^{\text{upper}}(t)$ and $\Omega_1^{\text{lower}}(t)$.

In the latter case, we need to find an intersection point at t such that either $\Omega_1^{\text{upper}}(t) = \Omega_2^{\text{upper}}(t)$, $\Omega_1^{\text{upper}}(t) = \Omega_2^{\text{lower}}(t)$, $\Omega_1^{\text{lower}}(t) = \Omega_2^{\text{upper}}(t)$, or $\Omega_1^{\text{lower}}(t) = \Omega_2^{\text{lower}}(t)$. If we do not need to construct a reachable velocity function, we can check the existence of an intersection point non-constructively using the intermediate value theorem: check whether the end points of one of the following equations have different sign: $f_1(t) = \Omega_1^{\text{upper}}(t) - \Omega_2^{\text{upper}}(t)$, $f_2(t) = \Omega_1^{\text{upper}}(t) - \Omega_2^{\text{lower}}(t)$, $f_3(t) = \Omega_1^{\text{lower}}(t) - \Omega_2^{\text{upper}}(t)$, and $f_4(t) = \Omega_1^{\text{lower}}(t) - \Omega_2^{\text{lower}}(t)$.

One way to find an intersection point is to find the roots of $f_1(t)$, $f_2(t)$, $f_3(t)$, $f_4(t)$ using some numerical algorithms such as Newton's method. However, Newton's method converges under certain conditions. In general, if the functions are monotonically decreasing functions, Newton's method will not get stuck at local minima and can converge quickly. Theorem 3 shows that all $\Omega^{\text{upper}}(t)$ and $\Omega^{\text{lower}}(t)$ are decreasing functions.

Theorem 3: Both $\Omega^{\text{upper}}(t)$ and $\Omega^{\text{lower}}(t)$ are decreasing functions.

Since $\Omega_1^{\text{upper}}(t)$ and $\Omega_1^{\text{lower}}(t)$ are decreasing while $\Omega_2^{\text{upper}}(t)$ and $\Omega_2^{\text{lower}}(t)$ are increasing (since the boundary equations of F_2' are reversed in time), $f_1(t)$, $f_2(t)$, $f_3(t)$ and $f_4(t)$ are decreasing functions. However, these functions are not necessarily monotonically decreasing, and the Newton's method needs to use some special procedures to handle the situation in with the derivative is zero. This approach can find the intersection points effectively.

VI. REACHABLE SETS FOR REAL VEHICLES

We also studied how to utilize reachable sets in the control of a real vehicle. The computation of a reachable set is based on a simplified vehicular model with bounded acceleration and deceleration, but the actual behavior of a real vehicle is more complicated than that, and hence the reachable set of a real vehicle is different from the one in the tables in

Fig. 5. Nonetheless, the actual behavior will only impose additional constraints to the computation of the reachable set, and hence *the reachable set of a real vehicle must be a subset of a reachable set in the tables in Fig. 5*. Thus, the reachable sets as described in Fig. 5 are still useful in identifying reachable arrival configurations and eliminating arrival configurations that are impossible to reach.

To illustrate this usage, we conducted an experiment with two road segments, one is a flat road and the other is a slope. We want to control a physical vehicle to arrive at the top of the slope at certain time and at certain velocity. The key question is to find a suitable arrival time and velocity at the junction of the two road segments. First, we used the tables in Fig. 5 to compute a reachable "superset" at the junction, based on the measurement of maximum acceleration and deceleration. Then we randomly chose a configuration in the superset and then used the bisection method in [7] to check whether there is a setpoint schedule to control the vehicle to arrive at the junction at the chosen configuration. If not, then we chose another configuration in the superset and tested again. Otherwise, we used the bisection method to check whether it is possible to meet the arrival requirement at the top of the slope starting from the chosen configuration. We repeated the process until we find a configuration at the junction such that reachable setpoint schedules exist on both road segments. Then we controlled the vehicle to run according to the two setpoint schedules, and measured the errors in the arrival time and the arrival velocity. We repeated the procedure 32 times with randomly chosen arrival configurations at the top of the slope. The errors in the arrival time and velocity are $-1.42 \pm 0.57s$ and $-0.16 \pm 0.16m/s$, respectively. These errors are probably due to the model error in performance profiling of the vehicle, as well as the sudden slowdown when the vehicle hit the slope at the junction. In the future, we will investigate how to reduce these errors.

VII. SUMMARY AND DISCUSSION

In this paper, we considered the problem of deciding whether a motion plan exists such that an autonomous vehicle can arrive at a specific position on a trajectory at a given time and velocity. Our main contribution is to improve the running time of the validation algorithm in [7] from logarithmic time to constant time. Our approach is based on the computation of reachable sets describing all reachable arrival configurations by closed form equations for any parameters in the problem. To illustrate the usage of these equations, we presented 1) a constant-time algorithm for the validation problem, and 2) the validation procedure for two road segments. We also discussed how these equations can be useful when applying to real autonomous vehicles. AIM is an important domain in which vehicles have to move to certain positions with precision in time and velocity. Beyond AIM, precision in time and velocity often plays a pivotal role for a team of robots to collaborate effectively. In the future, we will apply the equations in Fig. 5 to other motion planning problems for autonomous vehicles and mobile robots.

ACKNOWLEDGMENTS

This work has been taken place in the ART Lab at Ulsan National Institute of Science & Technology. ART research is supported by NRF (2.190315.01 and 2.180557.01).

REFERENCES

[1] J. H. Reif, "Complexity of the movers problem and generalizations," in *IEEE Symposium on Foundations of Computer Science*, 1979.

[2] J. E. Hopcroft, J. T. Schwartz, and M. Sharir, "On the complexity of motion planning for multiple independent objects: Pspace-hardness of the "warehouseman's problem"," *International Journal of Robotics Research*, vol. 3, no. 4, pp. 76–88, 1984.

[3] L. E. Kavaki, P. Švestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, 1996.

[4] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Computer Science Dept, Iowa State University, Tech. Rep. TR 98-11, 1998.

[5] T.-C. Au and P. Stone, "Motion planning algorithms for autonomous intersection management," in *AAAI 2010 Workshop on Bridging The Gap Between Task And Motion Planning (BTAMP)*, 2010.

[6] M. Quinlan, T.-C. Au, J. Zhu, N. Stiurca, and P. Stone, "Bringing simulation to life: A mixed reality autonomous intersection," in *IROS*, 2010.

[7] T.-C. Au, M. Quinlan, and P. Stone, "Setpoint scheduling for autonomous vehicle controllers," in *ICRA*, 2012, pp. 2055–2060.

[8] J. Johnson and K. Hauser, "Optimal acceleration-bounded trajectory planning in dynamic environments along a specified path," in *ICRA*, 2012, pp. 2035–2041.

[9] T. McMahon, S. Thomas, and N. M. Amato, "Sampling based motion planning with reachable volumes: Application to manipulators and closed chain systems," in *IROS*, 2014, pp. 3705–3712.

[10] N. Vahrenkamp, D. Berenson, T. Asfour, and J. K. R. Dillmann, "Humanoid motion planning for dual-arm manipulation and re-grasping tasks," in *IROS*, 2009, pp. 2464–2470.

[11] J. Johnson and K. Hauser, "Optimal longitudinal control planning with moving obstacles," in *IEEE Intelligent Vehicles Symposium*, 2013, pp. 605–611.

[12] P. Švestka and J. Vleugels, "Exact motion planning for tractor-trailer robots," in *ICRA*, 1995, pp. 2445–2450.

[13] D. Halperin, "Robust geometric computing in motion," *International Journal of Robotics Research*, vol. 21, no. 3, pp. 219–232, 2002.

[14] G. Varadhan, S. Krishnan, T. V. N. Sriram, and D. Manocha, "A simple algorithm for complete motion planning of translating polyhedral robots," *International Journal of Robotics Research*, vol. 24, no. 11, pp. 983–995, 2005.

[15] S. M. LaValle and J. James J. Kuffner, "Rapidly-exploring random trees: progress and prospects," in *Algorithmic and Computational Robotics: New Directions*, 2000, pp. 293–308.

[16] S. Hirsch and D. Halperin, "Hybrid motion planning: Coordinating two discs moving among polygonal obstacles in the plane," in *Algorithmic Foundations of Robotics V*. Springer, 2004, pp. 239–255.

[17] D. Godbole and J. Lygeros, "Longitudinal control of the lead car of a platoon," *IEEE Transactions on Vehicular Technology*, vol. 43, no. 4, pp. 1125–1135, 1994.

[18] S. Sheikholeslam and C. A. Desoer, "Longitudinal control of a platoon of vehicles," in *American Control Conference*, 1990, pp. 291–296.

[19] F. Broqua, "Impact of automatic and semi-automatic vehicle longitudinal control on motorway traffic," in *Proceedings of the Intelligent Vehicles '92 Symposium*, 1992, pp. 144–147.

[20] E. Frazzoli, "Robust hybrid control for autonomous vehicle motion planning," Ph.D. dissertation, MIT, 2001.

[21] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *International Journal of Robotics Research*, vol. 4, no. 3, pp. 3–17, 1985.

[22] T. Kunz and M. Stilman, "Time-optimal trajectory generation for path following with bounded acceleration and velocity," in *RSS*, 2012.

[23] T. Kröger, A. Tomiczek, and F. M. Wahl, "Towards on-line trajectory computation," in *IROS*, 2006, pp. 736–741.

[24] K. Hauser and V. Ng-Thow-Hing, "Fast smoothing of manipulator trajectories using optimal bounded-acceleration shortcuts," in *ICRA*, 2010, pp. 2493–2498.