2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Time–Optimal Motion Planning of Autonomous Vehicles in the Presence of Obstacles

Nader Sadegh

Abstract— **This paper presents a novel motion planning approach inspired by the Dynamic Programming (DP) for mobile robots and other autonomous vehicles. The proposed discrete–time algorithm enables a multi–degree of freedom vehicle to reach its destination through an arbitrary obstacle field in a minimum number of time steps. Furthermore, the resulting optimal path is guaranteed to posses the required degree of smoothness and incorporates the motion state constraints such as velocity, acceleration, and jerk limits. The algorithm is proven to terminate in a finite number of steps without its computational complexity increasing with the type or number of obstacles. The effectiveness of the algorithm is demonstrated on a mobile robot modeled as a point–mass in a 2–dimensional space subject to velocity and acceleration limits.**

I. Introduction

Path planning through an obstacle field is a critical task in controlling autonomous vehicles such as mobile robots, Unmanned Air Vehicles (UAV's), underwater vehicles, and exploratory rovers. No autonomous vehicle or robot can successfully operate in a constrained environment without a systematic mechanism for planning its motion path. The basic objective of most path planning algorithms is to generate a feasible path starting from an arbitrary initial location to the final target while avoiding the obstacles. Moreover, the path is often preferred to be optimal with respect to a certain criterion such as time and subject to state constraints such as velocity and acceleration limits as well as the actuation inputs.

The optimization problem associated with most path planning tasks is inherently non–convex [1] and difficult to solve [2]. The non–convexity of the problem is a direct consequence of the obstacle avoidance requirement and is further complicated by the inclusion of the state and actuation constraints. Furthermore, changing and/or uncertain obstacle fields, which arise in most real–world applications, pose additional challenges in achieving the required solution to this problem.

The basic path planning problem has been approached from several different angles. The Potential Field Theory (PFT) is one of the most widely used approaches for the aforementioned path planning problem [3],[4],[5]. Rimon and Koditschek developed a special artificial potential function, called a navigation function, that guarantees collision-free motion and convergence to the destination from almost all initial free configurations. Eichhorn applied the PFT [4] for trajectory planning of an underwater vehicle. Bruijnen et al. [5] proposed an extension of the PFT that can handle a changing obstacle field and generates a sub-optimal smooth path with bounds on the allowed velocity, acceleration and jerk. While the PFT is well suited for a varying obstacle fields, it does not necessarily generate a globally optimal path and in many instances it may render only a local optimum away from the destination. A recent work [6] partially addresses this deficiency by adding an extra control force to that of the PFT but falls short of guaranteeing convergence to the global optimal solution.

The second approach aimed at generating a global solution is based on Linear Programming (LP) and its variations [7],[8]. The main idea behind this approach is to formulate the path planning problem utilizing linear objective function and constraints and solve the resulting LP problem. Even though this approach is well suited for obstacle free situations, the addition of obstacle avoidance constraints makes the optimization problem non–convex and dramatically increases its complexity. The most successful LP based methods include Mixed–Integer Linear Programming (MILP) [7] and disjunctive LP (DLP) [9]. Blackmore et al. [1] uses a probabilistic approach to incorporate environmental uncertainty into an equivalent deterministic DLP problem. Unfortunately, the DLP is an NP–complete optimization and its complexity grows exponentially with the number of obstacles hence intractable for most practical applications.

This paper presents a discrete–time Time–Optimal motion planning algorithm based on the Dynamic Programming (DP) approach [10]. The existing works on the applications of DP to path planning can be found in [2],[11],[12],[13]. In particular, the approach pursued in [12] is the one most closely related to the present work. It incorporates a framed free space approach to reduce the computational complexity of the previous DP approaches. However, none of the aforementioned works guarantees time optimality or the convergence of their algorithm. They also do not consider state or input constraints and are limited to piecewise linear paths.

The motion planning algorithm presented here can be used to navigate a mobile robot through an arbitrary obstacle field to its destination in a minimum number of discrete time steps along a smooth spline path. The degrees of freedom of the robot along with the level of the path smoothness can be chosen arbitrarily. In addition, the motion state (velocity, acceleration, etc.) and input constraints are naturally enforced. The algorithm is proven to terminate in a finite number of steps and its complexity is shown not to increase with the number of obstacles. A two dimensional robot subject to

N Sadegh is with the The George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta Georgia 30332–0405, sadegh@gatech.edu

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

velocity and acceleration limits moving in a field of randomly placed circular obstacles is used to illustrate the effectiveness of this approach.

II. Problem Statement

Consider an n–dimensional open subset $\mathcal{X} \subset \mathbb{R}^n$ representing the geometric constraint associated with all the possible paths. Each path is specified as an m–th order *spline* defined according to

$$\mathcal{S}_m = \left\{ \boldsymbol{x} : \bigcup_{k=0}^{N-1} [kT, (k+1)T] \to \mathcal{X},\ N = 1, 2, \ldots \right\} \quad (1)$$

where T represents the spline step–size (also referred to as the *sampling period*) and $[kT, (k+1)T]$ is the domain of the i–th piece of the spline (an m–th order polynomial). The length or duration of each $\boldsymbol{x} \in \mathcal{S}_m$ is the number of polynomials (N) associated with $\boldsymbol{x}$. The objective of this optimization algorithm is to find $\boldsymbol{x} \in \mathcal{S}_m$ of the shortest duration N that connects an arbitrary initial point $\boldsymbol{x}(0) \in \mathcal{X}$ to a target point $\boldsymbol{\tau} \in \mathcal{X}$ while satisfying following additional constraints:

- The set $\mathcal{X} \subset \mathbb{R}^n$ containing all possible paths is assumed to be open and convex with a compact (closed and bounded) closure $\bar{\mathcal{X}}$.
- $\boldsymbol{x}(t) \notin \mathcal{O} \subset \mathcal{X}$, where $\mathcal{O}$ represents the set of obstacles assumed to be a closed subset of $\mathcal{X}$.
- The time derivatives of $\boldsymbol{x}(t)$, $\boldsymbol{x}^j := \frac{d^j \boldsymbol{x}}{dt^i}$, are bounded: $|x_i^j| < x_{i,\max}^j$, $i = 1, \ldots n$, $j = 1, \ldots, m$.

III. Spline Path Model

Each $\boldsymbol{x} \in \mathcal{S}_m$ spline can be represented by the differential equation $x_i^m(t) = u_i(t)$where each $u_i(t) \in [-x_{i,\max}^m, x_{i,\max}^m]$ is a piecewise constant function such that $u_i(t) = u_{i,k}$ for $t_k \leq t \leq t_{k+1}$. Defining the i–th partial state vector $\boldsymbol{\xi}_i = \left(x_i, x_i^1, \ldots, x_i^{m-1}\right)$, $i = 1, \ldots, n$, and integrating the resulting state equations for $t_k \leq t \leq t_{k+1}$ yields $\boldsymbol{\xi}_i(t) = \mathbf{A}_{t-t_k} \boldsymbol{\xi}_{i,k} + \mathbf{B}_{t-t_k} u_{i,k}$ where $\boldsymbol{\xi}_{i,k} = \boldsymbol{\xi}_i(t_k)$, $\mathbf{A}_\tau$ is an $m \times m$ upper triangular matrix whose ij–th element, $A_{\tau,ij} = \frac{\tau^{(j-i)}}{(j-i)!}$, $j \geq i$, and $\mathbf{B}_\tau$ is an $m \times 1$ vector with its i–th component $\mathbf{B}_{\tau,i} = \frac{\tau^{(m-i+1)}}{(m-i+1)!}$. For $t = t_{k+1}$, we have

$$\boldsymbol{\xi}_{i,k+1} = \mathbf{A}_T \boldsymbol{\xi}_{i,k} + \mathbf{B}_T u_{i,k} \quad (2)$$

Rather than using $\boldsymbol{\xi} = (\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_n)$ as the full state vector, we employ a permutation of it, $\boldsymbol{z} := \left(\boldsymbol{x}, \boldsymbol{x}^1, \ldots, \boldsymbol{x}^{m-1}\right)$, which can incorporate the geometric and dynamic constraints more naturally. It can be seen that the transition from $\boldsymbol{z}_k = \boldsymbol{z}(t_k)$ to $\boldsymbol{z}_{k+1} = \boldsymbol{z}(t_{k+1})$ is given by $\boldsymbol{z}_{k+1} = \boldsymbol{\varphi}_+(\boldsymbol{z}_k, \boldsymbol{u}_k)$where $\boldsymbol{\varphi}_+(\boldsymbol{z}, \boldsymbol{u}) = (\mathbf{A}_T \otimes I_{n\times n})\boldsymbol{z} + (\mathbf{B}_T \otimes I_{n\times n})\boldsymbol{u}$, $\otimes$ is the Kronecker product, and $I_{n\times n}$ the $n \times n$ identity matrix. The backward state transition, $\boldsymbol{\varphi}_- : \mathbb{R}^{mn} \times \mathbb{R}^m \to \mathbb{R}^{mn}$, which maps $(\boldsymbol{z}_{k+1}, \boldsymbol{u}_k)$ to $\boldsymbol{z}_k$, is given by $\boldsymbol{\varphi}_-(\boldsymbol{z}, \boldsymbol{u}) = (\mathbf{A}_{-T} \otimes I_{n\times n})\boldsymbol{z} + (\mathbf{B}_{-T} \otimes I_{n\times n})\boldsymbol{u}$. We shall use the symbol $\mathcal{Z}$ to denote the subset of the state space that meets the geometric ($\boldsymbol{x} \in \mathcal{X}$) and dynamic constraints: $\mathcal{Z} = \mathcal{X} \times \mathcal{X}^1 \times \cdots \times \mathcal{X}^{m-1}$, where $\mathcal{X}^j = \prod_{i=1}^{n} (-x_{i,\max}^j, x_{i,\max}^j)$. The space of all admissible inputs, $\boldsymbol{u}_k$'s, is denoted by $\mathcal{U} = \prod_{i=1}^{n} (-x_{i,\max}^m, x_{i,\max}^m)$.

The path planning algorithms to be presented will guarantee that the spline knots $\boldsymbol{x}_k$, $k = 1, 2, \ldots$, will be outside the obstacle set $\mathcal{O}$. However, it is possible for the intermediary points on the spline to enter the obstacle space. To avoid this situation, the set of obstacles $\mathcal{O}$ may be slightly expanded beyond the actual 'hard' obstacles to ensure that the entire path in addition to the knots will stay outside of the obstacles. Let $\mathcal{H}$ be a closed set representing the hard obstacle set and $B_\varepsilon = \{\boldsymbol{x} : \|\boldsymbol{x}\| \leq \varepsilon\}$, the closed ball of radius ε for some $\varepsilon > 0$. The following proposition shows that if $\mathcal{O} = \mathcal{H} + B_\varepsilon$, then $\boldsymbol{x}(t)$ is guaranteed to stay outside of $\mathcal{H}$ if the discrete positions $\boldsymbol{x}_k \notin \mathcal{O}$ for a sufficiently small sampling period T.

Proposition 1: Let $\mathcal{S}_m$ be the spline space of order m given by (1) and suppose that $\mathcal{O} = \mathcal{H} + B_\varepsilon$ such that $\boldsymbol{x}_k \notin \mathcal{O}$, $\forall k \geq 0$. Then $\boldsymbol{x} \notin \mathcal{H}$, $\forall \boldsymbol{x} \in \mathcal{S}_m$, $\forall m \geq 1$, provided that the sampling period $T < 2\ln(1+\varepsilon)/\sigma$ where $\sigma = \max_{0 \leq j \leq m} \left\| \left(x_{1,\max}^j, \ldots, x_{n,\max}^j \right) \right\|$ and $\|\cdot\|$ is any vector norm.

Proof: Letting $\tau = t - t_k$ then from $x_i^m(t) = u_i(t)$ it can be seen that $\boldsymbol{x}(t) - \boldsymbol{x}_k = \sum_{j=1}^{m} \frac{\tau^j}{j!} \boldsymbol{x}_k^j$, where $\boldsymbol{x}_k^j = \left(d^j \boldsymbol{x}/dt^j\right)_{t=t_k}$. Letting $\sigma = \max_{1 \leq j \leq m} \left\| \left(x_{1,\max}^j, \ldots, x_{n,\max}^j \right) \right\|$, then it follows that $\|\boldsymbol{x}(t) - \boldsymbol{x}_k\| \leq \sum_{j=1}^{m} \frac{T^j \sigma^j}{2^j j!} \leq e^{T\sigma/2} - 1$ for all $t_k - T/2 \leq t \leq t_k + T/2$, $\forall k \geq 1$. Choosing $e^{T\sigma/2} - 1 < \varepsilon$ or equivalently $T < 2\ln(1+\varepsilon)/\sigma$ implies that $\sup_{t \in [t_k - T/2, t_k + T/2]} \|\boldsymbol{x}(t) - \boldsymbol{x}_k\| < \varepsilon$. Thus $\boldsymbol{x} \notin \boldsymbol{x}_k + B_\varepsilon$ and $\boldsymbol{x} \notin \mathcal{H}$. ■

IV. Obstacle Free Motion Planning

In this section we present the main time–optimal optimization algorithm. In terms of the notations introduced in the preceding section, the main objective of this algorithm will be to transfer a given state $\boldsymbol{z} \in \mathcal{Z}$ to the target state $\boldsymbol{z}_\tau = (\boldsymbol{\tau}, 0, \ldots, 0)$ in the minimum number of steps possible while avoiding the obstacles. We begin our development in the absence of any obstacles and then extend the results to incorporate a fairly general set of obstacles. It turns out that the obstacle free case plays a key role in establishing the convergence properties of the general case.

To state our first preliminary result, let $\mathcal{Z}_0$ be the set of equilibrium states in $\mathcal{Z}$: $\mathcal{Z}_0 = \{(\boldsymbol{x}, \mathbf{0}, \ldots, \mathbf{0}) : \boldsymbol{x} \in \mathcal{X}\}$. The following proposition [14], which follows from the convexity of $\mathcal{X}$, shows that any two states in $\mathcal{Z}_0$ can be joined by a spline in a finite number of steps with the resulting path lying entirely in $\mathcal{Z}$.

Proposition 2: There exists an integer K such that any two states $\boldsymbol{z}, \boldsymbol{z}' \in \mathcal{Z}_0$ can be joined by a spline lying entirely in $\mathcal{Z}$ in at most K steps. That is, there exists a sequence of inputs $u_0, u_1, \ldots, u_{K-1} \in \mathcal{U}$ such that if $\boldsymbol{z}_0 = \boldsymbol{z}$ then $\boldsymbol{z}_k = \boldsymbol{\varphi}_+(\boldsymbol{z}_{k-1}, \boldsymbol{u}_{k-1}) \in \mathcal{Z}$ and $\boldsymbol{z}_K = \boldsymbol{z}'$.

The next definition is key in developing the motion planning algorithm.

Definition 1: The *forward* reachable set of $\boldsymbol{z} \in \mathcal{Z}$ is the set of all states in $\mathcal{Z}$ that can be reached from $\boldsymbol{z}$ in a single step:

$$\mathcal{R}_+(\boldsymbol{z}) = \big\{\boldsymbol{y} \in \mathcal{Z} : \boldsymbol{y} = \boldsymbol{\varphi}_+(\boldsymbol{z}, \boldsymbol{u}),\ \boldsymbol{u} \in \mathcal{U}\big\}$$

The *backward* reachable set from $\boldsymbol{z} \in \mathcal{Z}$ is the set of all states in $\mathcal{Z}$ that can reach $\boldsymbol{z}$ in a single step:

$$\begin{aligned}\mathcal{R}_-(\boldsymbol{z}) &= \big\{\boldsymbol{y} \in \mathcal{Z} : \boldsymbol{z} = \boldsymbol{\varphi}_+(\boldsymbol{y}, \boldsymbol{u}),\ \boldsymbol{u} \in \mathcal{U}\big\} \\ &= \big\{\boldsymbol{y} \in \mathcal{Z} : \boldsymbol{y} = \boldsymbol{\varphi}_-(\boldsymbol{z}, \boldsymbol{u}),\ \boldsymbol{u} \in \mathcal{U}\big\}\end{aligned}$$

If $\mathcal{W} \subset \mathcal{Z}$, then $\mathcal{R}_+(\mathcal{W}) = \bigcup\limits_{\boldsymbol{z} \in \mathcal{W}} \mathcal{R}(\boldsymbol{z})$ and $\mathcal{R}_-(\mathcal{W}) = \bigcup\limits_{\boldsymbol{z} \in \mathcal{W}} \mathcal{R}_-(\boldsymbol{z})$. The k–th composition of $\mathcal{R}_\pm$, denoted by $\mathcal{R}_\pm^k$, is defined as follows: $\mathcal{R}_\pm^k(\mathcal{W}) = \mathcal{R}_\pm(\mathcal{R}_\pm^{k-1}(\mathcal{W}))$ with $\mathcal{R}_\pm^0(\mathcal{W}) = \mathcal{W}$. In addition, for each $\boldsymbol{z} \in \mathcal{R}_\pm(\mathcal{W})$, the set of all control inputs $\boldsymbol{u} \in \mathcal{U}$ that connects $\boldsymbol{z}$ to some $\boldsymbol{w} \in \mathcal{W}$ or vice versa is denoted by $\mathcal{C}_\pm(\boldsymbol{z}, \mathcal{W})$:

$$\mathcal{C}_\pm(\boldsymbol{z}, \mathcal{W}) = \big\{\boldsymbol{u} \in \mathcal{U} : \boldsymbol{\varphi}_\pm(\boldsymbol{z}, \boldsymbol{u}) \in \mathcal{W}\big\}$$

The following Proposition [14] summarizes some of the important properties of the forward and backward reachable sets to be utilized in the sequel:

Proposition 3: The forward and backward operation $\mathcal{R}_\pm$ posses the following properties:

i. The sets $\mathcal{R}_\pm^k(\boldsymbol{\tau})$, $\forall k \geq m$, are open sets and nondecreasing: $\mathcal{R}_\pm^k(\boldsymbol{\tau}) \subset \mathcal{R}_\pm^{k+1}(\boldsymbol{\tau})$.

ii. There exists a finite integer $K \geq 1$ such that $\mathcal{Z}_0 \subset \mathcal{R}_\pm^k(\boldsymbol{\tau})$, for all $\boldsymbol{\tau} \in \mathcal{Z}_0$ and $k \geq K$.

iii. $\mathcal{R}_\pm\left(\bigcap_{k\geq 1} \mathcal{W}_k\right) \subset \bigcap_{k\geq 1} \mathcal{R}_\pm(\mathcal{W}_k)$, $\forall \mathcal{W}_k \subset \mathcal{Z}$.

iv. $\mathcal{R}_\pm\left(\bigcup_{k\geq 1} \mathcal{W}_k\right) = \bigcup_{k\geq 1} \mathcal{R}_\pm(\mathcal{W}_k)$, $\forall \mathcal{W}_k \subset \mathcal{Z}$.

Before presenting the general case, we consider the motion planning in the absence of any obstacles. The basic idea used here is inspired by the *Dynamic Programming* [10] approach. Starting from the target we first determine the set of all states that can reach the target (origin) in a single step, or simply $\mathcal{R}_-(\boldsymbol{\tau})$. Having identified $\mathcal{R}_-(\boldsymbol{\tau})$, we will then recursively expand our search to the sets $\mathcal{R}_-^2(\boldsymbol{\tau}), \mathcal{R}_-^3(\boldsymbol{\tau}), \ldots, \mathcal{R}_-^k(\boldsymbol{\tau})$, where as defined earlier, $\mathcal{R}_-^k(\boldsymbol{\tau})$ represents the set of all states within $\mathcal{Z}$ that can reach the target in k steps with all the intermediary states lying in $\mathcal{Z}$. This process continues until $\mathcal{R}_-^k(\boldsymbol{\tau})$ covers the entire equilibrium space $\mathcal{Z}_0$. Theorem 3.*ii* justifies the viability of this approach by furnishing a finite integer K for which the set $\mathcal{R}_-^k(\boldsymbol{\tau})$ (and also $\mathcal{R}_+^k(\boldsymbol{\tau})$), $\forall k \geq K$, for an arbitrary state in $\boldsymbol{\tau} \in \mathcal{Z}_0$ fills the entire $\mathcal{Z}_0$. We shall refer to $\mathcal{D} := \mathcal{R}_-^K(\boldsymbol{\tau}) \supset \mathcal{Z}_0$ as the algorithm's *domain of operation.*

V. The Main Algorithm

In this section we present the main path planning algorithm that incorporates both the obstacle avoidance requirement as well as the dynamic constraints. The obstacle space including the dynamic constraint is a subset of $\mathcal{Z}$ given by $\Theta := \mathcal{O} \times \mathcal{X}^1 \times \cdots \times \mathcal{X}^{m-1}$. As in the obstacle free case, the general algorithm is also based on Dynamic programming. Starting with the target set $\mathcal{W}_0 = \{\boldsymbol{\tau}\}$, we first identify the set of all states except for those within the obstacle set Θ that can reach the target in a single step. We will then recursively remove the obstacle set Θ from each backward reachable space until all the states in $\mathcal{Z}_0$ are covered. We will prove that the algorithm terminates in finite number of steps not depending on the number of obstacles. The only restriction that we place on the set of obstacles is that it is a closed subset of the space $\mathcal{X}$ without additional requirements such as convexity.

Algorithm 1: The main objective of the algorithm is to recursively identify the states than can reach the target set $\{\boldsymbol{\tau}\}$ in k steps for all integer $k = 1, 2, \ldots$ until the termination criterion is met. Given the constrained state space $\mathcal{Z} = \mathcal{X} \times \mathcal{X}^1 \times \cdots \times \mathcal{X}^{m-1}$, equilibrium space $\mathcal{Z}_0 = \{(\boldsymbol{x}, \boldsymbol{0}, \ldots, \boldsymbol{0}) : \boldsymbol{x} \in \mathcal{X}\}$, the obstacle set $\Theta := \mathcal{O} \times \mathcal{X}^1 \times \cdots \times \mathcal{X}^{m-1}$, an open set $\mathcal{O}_o \subset \mathcal{X}$ containing $\mathcal{O}$, and the target set $\{\boldsymbol{\tau}\}$, execute the following steps:

i Initialize the set $\mathcal{W}_0 = \{\boldsymbol{\tau}\}$.

ii For $k = 1, 2, \ldots$update $\mathcal{W}_k$: $\mathcal{W}_k = \mathcal{R}_-(\mathcal{W}_{k-1} - \Theta)$. To each $\boldsymbol{z} \in \mathcal{W}_k - \mathcal{W}_{k-1}$ assign the *next* state $\eta(\boldsymbol{z}) \in \mathcal{R}_+(\boldsymbol{z}) \cap \mathcal{W}_{k-1}$.

iii Continue until $(\mathcal{W}_k \cap \mathcal{Z}_0) - \Theta_o$ reaches a limit: i.e., $(\mathcal{W}_{k+1} \cap \mathcal{Z}_0) - \Theta_o = (\mathcal{W}_k \cap \mathcal{Z}_0) - \Theta_o$ where $\Theta_o = \mathcal{O}_o \times \mathcal{X}^1 \times \cdots \times \mathcal{X}^{m-1}$.

The next lemma is key in proving the convergence of the algorithm.

Lemma 1: Let $\mathcal{W}$ be an open and $\mathcal{B}$ a closed subset of $\mathcal{Z}$. Then $\mathcal{R}_-(\mathcal{W} - \mathcal{B}) = \mathcal{R}_-(\mathcal{W}) - \mathcal{V}$ where $\mathcal{V}$ is a closed subset of $\mathcal{Z}$ given by

$$\begin{aligned}\mathcal{V} &= \{\boldsymbol{z} \in \mathcal{Z} : \mathcal{R}_+(\boldsymbol{z}) \cap \mathcal{W} \subset \mathcal{B}\} \\ &= \big\{\boldsymbol{z} \in \mathcal{Z} : \boldsymbol{\varphi}_+(\boldsymbol{z}, \boldsymbol{u}) \in \mathcal{B},\ \forall \boldsymbol{u} \in \mathcal{C}(\boldsymbol{z}, \mathcal{W})\big\}\end{aligned}$$

Proof: Let $\boldsymbol{z} \in \mathcal{R}_-(\mathcal{W} - \mathcal{B})$. Then $\boldsymbol{\varphi}_+(\boldsymbol{z}, \boldsymbol{u}) \in \mathcal{W} - \mathcal{B}$ for some $\boldsymbol{u} \in \mathcal{U}$. This implies that $\boldsymbol{z} \in \mathcal{R}_-(\mathcal{W})$ and $\boldsymbol{\varphi}_+(\boldsymbol{z}, \boldsymbol{u}) \notin \mathcal{B}$. Thus $\boldsymbol{z} \notin \mathcal{V}$ since otherwise $\boldsymbol{\varphi}_+(\boldsymbol{z}, \boldsymbol{u}) \in \mathcal{W} \cap \mathcal{B}$, $\forall \boldsymbol{u} \in \mathcal{C}(\boldsymbol{z}, \mathcal{W})$. Consequently, $\boldsymbol{z} \in \mathcal{R}_-(\mathcal{W}) - \mathcal{V}$ proving that $\mathcal{R}_-(\mathcal{W} - \mathcal{B}) \subset \mathcal{R}_-(\mathcal{W}) - \mathcal{V}$. To prove the converse, let $\boldsymbol{z} \in \mathcal{R}_-(\mathcal{W}) - \mathcal{V}$. Then $\boldsymbol{z} \in \mathcal{R}_-(\mathcal{W})$ and $\boldsymbol{z} \notin \mathcal{V}$ implying that $\boldsymbol{\varphi}_+(\boldsymbol{z}, \boldsymbol{u}) \notin \mathcal{B}$ for some $\boldsymbol{u} \in \mathcal{C}(\boldsymbol{z}, \mathcal{W})$. Thus $\boldsymbol{\varphi}_+(\boldsymbol{z}, \boldsymbol{u}) \in \mathcal{W} - \mathcal{B}$ and $\boldsymbol{z} \in \mathcal{R}_-(\mathcal{W} - \mathcal{B})$.

To prove that $\mathcal{V}$ is closed, let $\boldsymbol{z}_k$ be a converging sequence in $\mathcal{V}$. We will show that $\boldsymbol{z} = \lim_{k\to\infty} \boldsymbol{z}_k \in \mathcal{V}$. If $\mathcal{R}_+(\boldsymbol{z}) \cap \mathcal{W} = \varnothing$ then $\mathcal{R}_+(\boldsymbol{z}) \cap \mathcal{W} \subset \mathcal{B}$ and there is nothing to prove. Assuming $\mathcal{R}_+(\boldsymbol{z}) \cap \mathcal{W} \neq \varnothing$, let $\boldsymbol{u} \in \mathcal{U}$ be such that $\boldsymbol{\varphi}_+(\boldsymbol{z}, \boldsymbol{u}) \in \mathcal{W}$. We need to show that $\boldsymbol{\varphi}_+(\boldsymbol{z}, \boldsymbol{u}) \in \mathcal{B}$. Since $\mathcal{W}$ is open, there exists an open neighborhood $\mathcal{N} \subset \mathcal{W}$ containing $\boldsymbol{\varphi}_+(\boldsymbol{z}, \boldsymbol{u})$. The continuity of $\boldsymbol{\varphi}_+(., \boldsymbol{u})$ implies that $\lim_{k\to\infty} \boldsymbol{\varphi}_+(\boldsymbol{z}_k, \boldsymbol{u}) = \boldsymbol{\varphi}_+(\boldsymbol{z}, \boldsymbol{u})$. Thus there exists $\bar{k} \geq 1$ such that if $k \geq \bar{k}$ then $\boldsymbol{\varphi}_+(\boldsymbol{z}_k, \boldsymbol{u}) \in \mathcal{N}$. By the definition of $\mathcal{V}$, $\boldsymbol{z}_k \in \mathcal{V}$ and $\boldsymbol{\varphi}_+(\boldsymbol{z}_k, \boldsymbol{u}) \in \mathcal{W}$ imply that $\boldsymbol{\varphi}_+(\boldsymbol{z}_k, \boldsymbol{u}) \in \mathcal{B}$, $\forall k \geq \bar{k}$. The sequence $\boldsymbol{\varphi}_+(\boldsymbol{z}_k, \boldsymbol{u})$ is a converging sequence in the closed set $\mathcal{B}$ hence $\boldsymbol{\varphi}_+(\boldsymbol{z}, \boldsymbol{u}) = \lim_{k\to\infty} \boldsymbol{\varphi}_+(\boldsymbol{z}_k, \boldsymbol{u}) \in \mathcal{B}$ proving that $\boldsymbol{z} \in \mathcal{V}$. Thus the set $\mathcal{V}$ is a closed set. ■

Applying the preceding lemma to $\mathcal{W}_2 = \mathcal{R}_-(\mathcal{W}_1 - \Theta)$ gives $\mathcal{W}_2 = \mathcal{R}_-(\mathcal{W}_1) - \mathcal{V}_1$ where $\mathcal{V}_1$ is a closed set satisfying $\mathcal{R}_+(\mathcal{V}_1) \cap \mathcal{W}_1 \subset \Theta$. Defining $\Theta_1 = \mathcal{V}_1 \cup \Theta$, then it can be seen that $\mathcal{W}_2 - \Theta = \mathcal{R}_-^2(\boldsymbol{\tau}) - \Theta_1$. Applying the lemma once more to $\mathcal{W}_2 - \Theta$ gives $\mathcal{R}_-(\mathcal{W}_2 - \Theta) = \mathcal{R}_-^3(\boldsymbol{\tau}) - \mathcal{V}_2$ where the closed set $\mathcal{V}_2$ satisfies $\mathcal{R}_+(\mathcal{V}_2) \cap \mathcal{R}^2_-(\boldsymbol{\tau}) \subset \Theta_1$. Continuing this procedure recursively using a simple induction, we arrive at the next proposition.

Proposition 4: The sets $\mathcal{W}_k$ in Algorithm 1 satisfy the following recursive relationships $\mathcal{W}_k = \mathcal{R}^k_-(\boldsymbol{\tau}) - \mathcal{V}_{k-1}$ where $\mathcal{V}_k$ is a closed set satisfying $\mathcal{R}_+(\mathcal{V}_k) \cap \mathcal{R}^k_-(\boldsymbol{\tau}) \subset \Theta_{k-1}$, $\Theta_k = \mathcal{V}_k \cup \Theta$ and $\mathcal{V}_0 = \Theta$.

Remark 1: The sets Θ_{k-1}, $k = 1, 2, \ldots$, which is an expansion of the obstacle set in $\Theta \subset \mathcal{Z}$ contains all the states in $\mathcal{R}^k_-(\boldsymbol{\tau})$ that cannot reach the target in k steps or fewer (i.e., *blocked* states). Of course, in general there may be states such as those completely surrounded and blocked by obstacles that can never reach the target. We will show later in the paper that the main algorithm identifies all the blocked equilibrium states in a *finite* number of steps not necessarily depending on the *number* or size of the obstacles.

The next definition formalizes the notion of the blocked states alluded to in the preceding remark.

Definition 2: Let K be the smallest integer such that $\mathcal{D} = \mathcal{R}^K_-(\boldsymbol{\tau}) \supset \mathcal{Z}_0$. The set of states in $\mathcal{D}$ that cannot reach the target in a finite number of steps are called the blocked states and denoted by $\mathcal{V}_\infty = \bigcap_{k \geq K} \mathcal{V}_k$. The subset of $\mathcal{X}$ whose corresponding states $\boldsymbol{z} = (\boldsymbol{x}, 0, \ldots, 0)$, $\boldsymbol{x} \in \mathcal{X}$, are the equilibrium states are in $\mathcal{V}_\infty$ are called the *blocked positions* and are denoted by $\mathcal{B} = \pi(\mathcal{V}_\infty \cap \mathcal{Z}_0)$ where $\pi : \mathbb{R}^{mn} \to \mathbb{R}^n$ is the projection mapping that maps $\boldsymbol{z} = \left(\boldsymbol{x}, \boldsymbol{x}^1, \ldots, \boldsymbol{x}^{m-1}\right)$ to $\pi(\boldsymbol{z}) = \boldsymbol{x}$. The set of all the positions in $\mathcal{X}$ that can reach the target in a finite number of steps is called the admissible set and is denoted by $\mathcal{A} = \mathcal{X} - \mathcal{B}$.

As was shown in Proposition 4 the set of blocked states can be determined indirectly by the algorithm as a limit of $\mathcal{V}_k = \mathcal{R}^k_-(\boldsymbol{\tau}) - \mathcal{W}_k$ as k approaches infinity. The next lemma [14] shows that once the domain of operation $\mathcal{D} = \mathcal{R}^K_-(\boldsymbol{\tau})$, which is independent of the obstacles, is known the set of blocked states $\mathcal{V}_\infty$ can be identified directly without going through an iterative process.

Lemma 2: The set of blocked states $\mathcal{V}_\infty = \bigcap_{k \geq K} \mathcal{V}_k$ introduced in Definition 2 satisfies the relationship $\mathcal{R}_+(\mathcal{V}_\infty) \cap \mathcal{D} \subset \mathcal{V}_\infty \cup \Theta$. Furthermore, for any open set $\mathcal{O}_o \supset \mathcal{O}$ there exists an open set $\mathcal{B}_\varepsilon \subset \mathcal{X}$ containing the blocked positions $\mathcal{B} = \pi(\mathcal{V}_\infty \cap \overline{\mathcal{Z}_0})$ such that $\mathcal{B}_\varepsilon \cup \mathcal{O}_o = \mathcal{B} \cup \mathcal{O}_o$.

The convergence proof of Algorithm 1 is provided in the following theorem.

Theorem 1: Algorithm 1 terminates in a finite number of steps for an arbitrary open set $\mathcal{O}_o \subset \mathcal{X}$ containing $\mathcal{O}$. In particular, for $m = 1$ the algorithm terminates in a finite number of steps with $\mathcal{O}_o = \mathcal{O}$.

Proof: Let $\mathcal{W}^0_k = \mathcal{Z}_0 \cap \mathcal{W}_k$ and $\mathcal{V}^0_k = \bar{\mathcal{Z}}_0 \cap \mathcal{V}_k$, where $\bar{\mathcal{Z}}_0$ is the closure of $\mathcal{Z}_0$. Intersecting $\mathcal{W}_k = \mathcal{R}^k_-(\boldsymbol{\tau}) - \mathcal{V}_{k-1}$ with $\mathcal{Z}_0$ yields

$$\mathcal{W}^0_k = \mathcal{Z}_0 \cap \mathcal{R}^k_-(\boldsymbol{\tau}) - \mathcal{V}^0_{k-1} = \mathcal{Z}_0 - \mathcal{V}^0_{k-1} \qquad (3)$$

for $\forall k \geq K$. The set $\mathcal{W}^0_k$ represents the set of *all* equilibrium sates (i.e., $\boldsymbol{z} = (\boldsymbol{x}, 0, \ldots, 0) \in \mathcal{Z}_0$) that can reach the target in at most k steps. Thus clearly $\mathcal{W}^0_k \subset \mathcal{W}^0_{k+1}$ implying that $\mathcal{V}^0_{k+1} \subset \mathcal{V}^0_k$. The sets $\mathcal{V}^0_k = \overline{\mathcal{Z}_0} \cap \mathcal{V}_k$ are closed subsets of the compact set $\bar{\mathcal{Z}}$. Let $\mathcal{V}^0_\infty = \lim_{k \to \infty} \mathcal{V}^0_k = \bigcap_{k \geq K} \mathcal{V}^0_k = \mathcal{V}_\infty \cap \bar{\mathcal{Z}}_0$. If $\mathcal{V}^0_\infty = \varnothing$, then by the Henie–Borel property [15] of compact sets, there exists a finite $N \geq K$ such that $\bigcap_{k=K}^N \mathcal{V}^0_k = \varnothing$ implying that $\mathcal{W}^0_k = \mathcal{Z}_0$ for all $k > N$. Otherwise, $\mathcal{V}^0_\infty$ is a closed non–empty subset of $\mathcal{Z}_0$. We will now prove that even if $\mathcal{V}^0_\infty \neq \varnothing$, the algorithm terminates in a finite number of steps. Let $\mathcal{B}_\varepsilon \subset \mathcal{X}$ be the open set given in Lemma 2 satisfying $\mathcal{B}_\varepsilon \cup \mathcal{O}_o = \mathcal{B} \cup \mathcal{O}_o$ and consider the *closed* subsets $\tilde{\mathcal{B}}_k = \pi(\mathcal{V}^0_k) - \mathcal{B}_\varepsilon$ of $\bar{\mathcal{Z}}$. It can be easily seen that

$$\bigcap_{k \geq K} \tilde{\mathcal{B}}_k = \pi(\mathcal{V}^0_\infty) - \mathcal{B}_\varepsilon = \mathcal{B} - \mathcal{B}_\varepsilon = \varnothing$$

Once more by the Henie–Borel property of compact sets, there exists a finite $N \geq K$ such that $\bigcap_{p \geq K}^k \tilde{\mathcal{B}}_p = \varnothing$ implying that $\mathcal{B} \subset \pi\left(\mathcal{V}^0_k\right) \subset \mathcal{B}_\varepsilon$, $\forall k \geq N$. Thus $\pi\left(\mathcal{V}^0_{k-1}\right) \cup \mathcal{O}_o = \mathcal{B}_\varepsilon \cup \mathcal{O}_o = \mathcal{B} \cup \mathcal{O}_o$, $\forall k > N$. Using this property in (3) implies that

$$\pi(\mathcal{W}^0_k) - \mathcal{O}_o = \mathcal{X} - \pi\left(\mathcal{V}^0_{k-1}\right) \cup \mathcal{O}_o = \mathcal{X} - \mathcal{B} \cup \mathcal{O}_o$$

which completes the proof. ■

Remark 2: As evidenced from the proof of the theorem, the execution time of algorithm 1 does not necessarily grow with the number of obstacles as many of the existing computational algorithms do. Instead, it is exactly equal to the maximum number of steps required to connect any point in the admissible space $\mathcal{X} - \mathcal{O}_0$ to the target while satisfying all the geometric and dynamic constraints.

VI. Discrete Implementation of the Optimization Algorithm

The trajectory optimization algorithm presented in the preceding section may be implemented by discretizing the input space. A simple way to discretize of the input space would be to divide the interval $[0, u_{i,\max}]$ for some $u_{i,\max} < x^m_{i,\max}$ into M equal subintervals and restrict each u_i to the discrete values $\pm\frac{\ell}{M} u_{i,\max}$, $\ell = 0, 1, \ldots M$. According to Proposition 2 and its discrete counterpart (Lemma 3) setting M to $c_m := \max_{0 \leq j \leq m} \frac{m!}{j!(m-j)!}$ guarantees that any two points in $\mathcal{Z}_0$ can be connected to one another arbitrarily close using the discretized input set for a sufficiently small sampling period T. Let $\widehat{\mathcal{U}} = \left\{(u_1, \ldots, u_n) : u_i = \pm\frac{\ell}{M} u_{i,\max},\ \ell = 0, 1, \ldots c_m\right\}$ denote the set of discretized inputs. Clearly the set of all the states that can reach the target with $\boldsymbol{u} \in \widehat{\mathcal{U}}$ is a discrete subset of the operation domain $\mathcal{D}$. Later in this section, we shall show that the equilibrium states contained in this discrete subset approximate $\mathcal{Z}_0$ to within a prescribed degree of accuracy by choosing the sampling period T sufficiently small. To this end, we first define the discrete counterpart of the forward and backward reachable sets.

Definition 3: The discrete forward and backward reachable sets for any are given by $\hat{\mathcal{R}}_\pm(\boldsymbol{z}) = \left\{\boldsymbol{y} \in \mathcal{Z} : \boldsymbol{y} = \boldsymbol{\varphi}_\pm(\boldsymbol{z}, \boldsymbol{u}),\ \boldsymbol{u} \in \widehat{\mathcal{U}}\right\}$. The k–th composition of $\hat{\mathcal{R}}_\pm$, denoted by $\hat{\mathcal{R}}^k_\pm$, is defined as follows: $\hat{\mathcal{R}}^k_\pm(\mathcal{W}) = \hat{\mathcal{R}}_\pm(\hat{\mathcal{R}}^{k-1}_\pm(\mathcal{W}))$ with $\hat{\mathcal{R}}^0_\pm(\mathcal{W}) = \mathcal{W}$.

The following lemma [14] shows that for any two arbitrary positions in $\mathcal{X}$ can be *approximately* connected by means of a finite length spline to within a prescribed degree of accuracy.

To state the lemma, let $B_\varepsilon = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\| \leq \varepsilon\}$ be the closed ball of radius ε centered at the origin and define $\mathcal{X}_\varepsilon := \{\boldsymbol{x} \in \mathcal{X} : \boldsymbol{x} + B_\varepsilon \subset \mathcal{X}\}$ and $\mathcal{Z}_0^\varepsilon = \{\boldsymbol{z} \in \mathcal{Z}_0 : \pi(\boldsymbol{z}) \in \mathcal{X}_\varepsilon\}$. It can be easily seen that the set $\mathcal{X}_\varepsilon$ is an open and convex subset of $\mathbb{R}^n$ and $\mathcal{X}_\varepsilon + B_\varepsilon \subset \mathcal{X}$.

Lemma 3: For any $\varepsilon > 0$ there exists a sampling period T and an integer $K_\varepsilon \geq 1$ such that given any two equilibrium states $\boldsymbol{z}, \boldsymbol{w} \in \mathcal{Z}_0^\varepsilon$ a third equilibrium state $\hat{\boldsymbol{w}} \in \hat{\mathcal{R}}_\pm^k(\boldsymbol{z}) \cap \mathcal{Z}_0$, $\forall k \geq K_\varepsilon$, can be chosen to approximate $\boldsymbol{w}$ to within ε: $\|\hat{\boldsymbol{w}} - \boldsymbol{w}\| \leq \varepsilon$. In other words, $\hat{\mathcal{R}}_\pm^k(\boldsymbol{z}) + B_0^\varepsilon \supset \mathcal{Z}_0, \forall \boldsymbol{z} \in \mathcal{Z}_0^\varepsilon$, $\forall k \geq \hat{K}$, where $B_0^\varepsilon = B_\varepsilon \times \mathbf{0} \times \cdots \times \mathbf{0}$.

We now present the discrete version of the optimization algorithm. The difference between the two is that the discrete input set $\widehat{\mathcal{U}}$ is used in place of $\mathcal{U}$.

Algorithm 2: The main objective of the algorithm is to recursively identify the states than can reach the target set $\{\boldsymbol{\tau}\}$ in k steps for all integer $k = 1, 2, \ldots$ until the termination criterion is met. Given the constrained state space $\mathcal{Z} = \mathcal{X} \times \mathcal{X}^1 \times \cdots \times \mathcal{X}^{m-1}$, equilibrium space $\mathcal{Z}_0 = \{(\boldsymbol{x}, \mathbf{0}, \ldots, \mathbf{0}) : \boldsymbol{x} \in \mathcal{X}\}$, the obstacle set $\Theta := \mathcal{O} \times \mathcal{X}^1 \times \cdots \times \mathcal{X}^{m-1}$, and the target set $\{\boldsymbol{\tau}\}$, execute the following steps:

i Initialize the set $\widehat{\mathcal{W}}_0 = \{\boldsymbol{\tau}\}$.

ii For $k = 1, 2, \ldots$update $\widehat{\mathcal{W}}_k$: $\widehat{\mathcal{W}}_k = \hat{\mathcal{R}}_-(\widehat{\mathcal{W}}_k - \Theta)$. To each $\boldsymbol{z} \in \widehat{\mathcal{W}}_k - \widehat{\mathcal{W}}_{k-1}$ assign the *next* state $\eta(\boldsymbol{z}) \in \hat{\mathcal{R}}_+(\boldsymbol{z}) \cap \widehat{\mathcal{W}}_{k-1}$.

iii Continue until $\pi\left(\widehat{\mathcal{W}}_k \cap \mathcal{Z}_0\right) - \mathcal{O}$ reaches a limit: i.e., $\pi\left(\widehat{\mathcal{W}}_{k+1} \cap \mathcal{Z}_0\right) - \mathcal{O} = \pi\left(\widehat{\mathcal{W}}_k \cap \mathcal{Z}_0\right) - \mathcal{O}$.

The next proposition follows directly from its continuous counterpart, Proposition 4.

Proposition 5: The sets $\widehat{\mathcal{W}}_k$ in Algorithm 2 satisfy the following recursive relationships $\widehat{\mathcal{W}}_k = \hat{\mathcal{R}}_-^k(\boldsymbol{\tau}) - \widehat{\mathcal{V}}_{k-1}$ where $\widehat{\mathcal{V}}_k$ is a finite discrete (hence closed) set satisfying $\hat{\mathcal{R}}_+(\mathcal{V}_k) \cap \hat{\mathcal{R}}_-^k(\boldsymbol{\tau}) \subset \hat{\Theta}_{k-1}$, where $\hat{\Theta}_k = \widehat{\mathcal{V}}_k \cup \Theta$ and $\widehat{\mathcal{V}}_0 = \Theta$.

Definition 4: Let K_ε, $\varepsilon > 0$ be the smallest integer for which $\hat{\mathcal{R}}_\pm^k(\boldsymbol{z}) + B_0^\varepsilon \supset \mathcal{Z}_0, \forall k \geq K_\varepsilon$ with the sampling period T as specified in Lemma 3. Define the discrete domain of operation to be $\hat{\mathcal{D}}_\varepsilon = \hat{\mathcal{R}}_\pm^{K_\varepsilon}(\boldsymbol{z})$ and denote the set of positions in $\pi(\mathcal{Z}_0 \cap \hat{\mathcal{D}}_\varepsilon)$ that can reach the target $\boldsymbol{\tau}$ in a finite number of steps while remaining in $\mathcal{X} - \mathcal{O}$ by $\widehat{\mathcal{A}}_\varepsilon$ referred to as the set of discrete admissible positions.

A natural question here is how the set of discrete admissible positions $\widehat{\mathcal{A}}_\varepsilon$ is related to its continuous counterpart. The theorem below sheds light on the relationship between the two admissible sets:

Theorem 2: Let $\mathcal{A}$ and $\mathcal{A}_\varepsilon$ be the the admissible positions corresponding to the continuous Algorithm 1 (see Definition 2) with the domain sets $\mathcal{X}$ and $\mathcal{X}_\varepsilon = \{\boldsymbol{x} \in \mathcal{X} : \boldsymbol{x} + B_\varepsilon \in \mathcal{X}\}$ and obstacle sets $\mathcal{O}$ and $\mathcal{O} + B_{3\varepsilon}$, respectively. The discrete admissible positions $\widehat{\mathcal{A}}_\varepsilon \subset \mathcal{A}$ satisfies $\widehat{\mathcal{A}}_\varepsilon + B_\varepsilon \supset \mathcal{A}_\varepsilon$.

Remark 3: The theorem shows that $\widehat{\mathcal{A}}_\varepsilon$ is large enough to approximate the admissible set $\mathcal{A}_\varepsilon$ of the continuous algorithm with domain $\mathcal{X}_\varepsilon$ and obstacle set $\mathcal{O} + B_{3\varepsilon}$ to within ε.

Proof: Clearly we have $\widehat{\mathcal{A}}_\varepsilon \subset \mathcal{A}$. To show that $\mathcal{A}_\varepsilon \subset \widehat{\mathcal{A}}_\varepsilon + B_\varepsilon$, let $\boldsymbol{x} \in \mathcal{A}_\varepsilon$. There exists an integer $K_{\boldsymbol{x}}$ and an input sequence $u_0, u_1, \ldots, u_{K_{\boldsymbol{x}}-1} \in \mathcal{U}$ such that that if $\boldsymbol{z}_0 = (\boldsymbol{x}, 0, \ldots, 0)$ then $\boldsymbol{z}_k = \boldsymbol{\varphi}_+(\boldsymbol{z}_{k-1}, \boldsymbol{u}_{k-1}) \in \mathcal{Z}$, $x_k := \pi(z_k) \notin \mathcal{O} + B_{3\varepsilon}$, and $\boldsymbol{z}_{K_{\boldsymbol{x}}} = (\boldsymbol{\tau}, 0, \ldots, 0)$. By Proposition 1 the resulting spline path $\boldsymbol{x}(t) \in \mathcal{X}_\varepsilon - (\mathcal{O} + B_{2\varepsilon})$, $\forall t \in [0, K_{\boldsymbol{x}}T]$. Since the path $\boldsymbol{x}(t)$ is a continuously differentiable path in $\mathcal{X}_\varepsilon$, there exists an integer $P \geq K_{\boldsymbol{x}}$ and a corresponding sequence of discrete points $\boldsymbol{x}_p \in \{x(t) : 0 \leq t \leq K_x T\}$, $p = 0, 1, 2, \ldots, P$ such that $\boldsymbol{x}_0 = \boldsymbol{x}$, $\boldsymbol{x}_P = \boldsymbol{\tau}$, and the straight line joining $\boldsymbol{x}_p$ to $\boldsymbol{x}_{p+1}$ is not in $\mathcal{O} + B_\varepsilon$. By Lemma 3 it is possible to construct a discrete path $\hat{\boldsymbol{x}}_p$ with the input functions $u_i \in \widehat{\mathcal{U}}$ such that $\hat{\boldsymbol{x}}_P = \boldsymbol{\tau}$ and $\sup_{0 \leq p < P} \|\hat{\boldsymbol{x}}_p - \boldsymbol{x}_p\| < \varepsilon$. Thus $\hat{\boldsymbol{x}}_p \in \mathcal{X} - \mathcal{O}$ implying that $\boldsymbol{x} \in \widehat{\mathcal{A}}_\varepsilon + B_\varepsilon$. ■

The following is the discrete counterpart of the continuous convergence theorem (1):

Theorem 3: Suppose that for $\varepsilon > 0$ the sampling period T is chosen in accordance with Lemma 3. Then, Algorithm 2 terminates in a finite number of steps.

Proof: Let $\hat{\mathcal{Z}}_0 = \mathcal{Z}_0 \cap \widehat{\mathcal{R}}_-^{K_\varepsilon}(\boldsymbol{\tau})$ and define $\widehat{\mathcal{W}}_k^0 = \hat{\mathcal{Z}}_0 \cap \widehat{\mathcal{W}}_k$ and $\widehat{\mathcal{V}}_k^0 = \hat{\mathcal{Z}}_0 \cap \widehat{\mathcal{V}}_k$. Intersecting $\widehat{\mathcal{W}}_k = \widehat{\mathcal{R}}_-^k(\boldsymbol{\tau}) - \widehat{\mathcal{V}}_{k-1}$ with $\hat{\mathcal{Z}}_0$ yields

$$\widehat{\mathcal{W}}_k^0 = \hat{\mathcal{Z}}_0 \cap \widehat{\mathcal{R}}_-^k(\boldsymbol{\tau}) - \widehat{\mathcal{V}}_{k-1}^0 = \hat{\mathcal{Z}}_0 - \widehat{\mathcal{V}}_{k-1}^0 \tag{4}$$

for $\forall k \geq K$. The set $\widehat{\mathcal{W}}_k^0$ represents the set of *all* discrete equilibrium sates (i.e., $\boldsymbol{z} = (\boldsymbol{x}, 0, \ldots, 0) \in \hat{\mathcal{Z}}_0$) that can reach the target in at most k steps. Thus clearly $\widehat{\mathcal{W}}_k^0 \subset \widehat{\mathcal{W}}_{k+1}^0$ implying that $\widehat{\mathcal{V}}_{k+1}^0 \subset \widehat{\mathcal{V}}_k^0$. The sets $\widehat{\mathcal{V}}_k^0 = \hat{\mathcal{Z}}_0 \cap \widehat{\mathcal{V}}_k$ are finite and discrete (hence compact) subsets of the compact set $\bar{\mathcal{Z}}$. Let $\widehat{\mathcal{V}}_\infty^0 = \lim_{k \to \infty} \widehat{\mathcal{V}}_k^0 = \bigcap_{k \geq K} \widehat{\mathcal{V}}_k^0$. We shall show that even if $\widehat{\mathcal{V}}_\infty^0 \neq \varnothing$, the algorithm terminates in a finite number of steps. Consider the *closed* subsets $\widetilde{\mathcal{V}}_k^0 = \widehat{\mathcal{V}}_k^0 - \widehat{\mathcal{V}}_\infty^0$ of $\bar{\mathcal{Z}}$. It can be easily seen that $\bigcap_{k \geq K} \widetilde{\mathcal{V}}_k^0 = \varnothing$. By the Henie–Borel property of compact sets, there exists a finite $N \geq K$ such that $\bigcap_{p \geq K}^k \widehat{\mathcal{V}}_p^0 = \widehat{\mathcal{V}}_k^0 \subset \widehat{\mathcal{V}}_\infty^0$, $\forall k \geq N$. Using that $\widehat{\mathcal{V}}_\infty^0 \subset \widehat{\mathcal{V}}_k^0$, $\forall k \geq K_\varepsilon$, it follows that $\widehat{\mathcal{V}}_k^0 = \widehat{\mathcal{V}}_\infty^0, \forall k \geq K_\varepsilon$. Thus $\widehat{\mathcal{W}}_k^0 = \hat{\mathcal{Z}}_0 - \widehat{\mathcal{V}}_\infty^0$, which proves the theorem. ■

Remark 4: The theorem shows that, similarly to its continuous counterpart, the discrete algorithm terminates in a finite number of steps depending on the size of the domain $\mathcal{X}$ and the complexity of the obstacles but not necessarily the number of obstacles. The computational complexity of the algorithm (number of FLOPS) is directly related to the number of admissible states (i.e., states that can reach the target in a finite number of states). This number clearly is maximum in the absence of any obstacles (i.e., free space motion) and can only decrease as the number of obstacles increases. Therefore the complexity of the algorithm does not grow as the number of obstacles increases. See the illustrative example in the next section for a numerical justification.

VII. Numerical Case Study

The discrete version of the proposed optimal control strategy is applied to the navigation of a mobile robot modeled as a point–mass through a set of arbitrary obstacles. The domain of operation of the robot is a rectangular box $\mathcal{X} = (-x_{\max}, x_{\max}) \times (-x_{\max}, x_{\max})$ with $x_{\max} = 5$ m.

No. of Obstacles	No. of Steps	No. of Admissible States	No. of Blocked States
5	20	76494	13330
10	22	69700	20124
20	33	49039	40785
30	45	38265	51559
40	52	31431	58393
50	57	23260	66564

TABLE I

Table 1: Summary of the Computational Complexity of the Optimization Algorithm

Additionally, the motion of the robot is subject to velocity and acceleration limits ($m = 2$) of $v_{\max} = 3$ m/sec and $a_{\max} = 10$ m/sec^2, respectively, in both the x and y directions. The sampling period is set to $T = 0.1$ sec and for simplicity the origin $\boldsymbol{\tau} = (0, 0)$ is selected as the target position. Each obstacle is a circle with its radius varying between 0.7 m and 1 m. The algorithm is evaluated for randomly placed obstacles ranging in number from a minimum of 5 to a maximum of 50. Figure 1 shows the time optimal trajectories starting from the vicinity of the four corners of $\mathcal{X}$. As can be seen each of the 4 trajectories avoid the obstacles and arrive at the destination along a smooth path. Table 1 shows the number of steps required to terminate the algorithm as well as the the total number of the resulting admissible discrete states (i.e., the states that can reach the target in a finite number of steps) as the number of obstacles increases from 5 to 50. The significance of the latter quantity is that it is directly proportional to the complexity of the algorithm (i.e., number of FLOPS). The table shows that while the number of required steps increases with increasing the number of obstacles the overall algorithm complexity remains fairly constant.

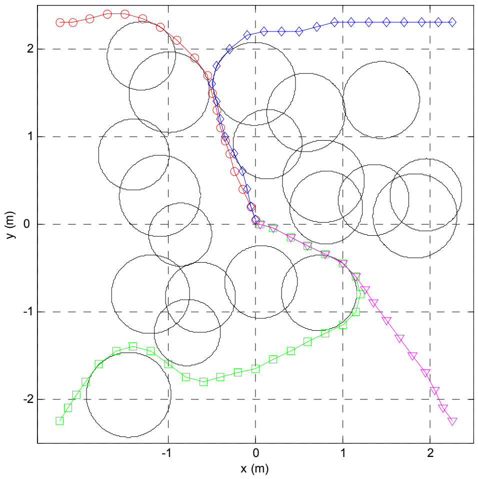

Fig. 1. Time Optimal Trajectories Corresponding to 20 Obstacles

VIII. Conclusions

This paper presented a discrete–time motion planning algorithm for mobile robots and vehicles. It generates a global navigation map, which enables a robot to reach its destination through an arbitrary obstacle field in a minimum number of time–steps. The resulting path is a smooth spline incorporating the motion state and input constraints such as velocity, acceleration, and jerk limits. To implement the proposed algorithm, a discrete version of it was formulated by discretizing the input space into a finite set. Each algorithm was shown to terminate in a finite number of steps with its overall computational complexity being independent of the type and number of obstacles. The effectiveness of the discretized algorithm was demonstrated on a mobile robot modeled as a point–mass in a 2–dimensional space subject to velocity and acceleration constraints.

References

[1] L. Blackmore, H. Li, and B. Williams, "A probabilistic approach to optimal robust path planning with obstacles," in *Proceedings of the 2006 American Control Conference*, June 14–16 2006, pp. 2831–37.

[2] J. Peterson, "Obstacle avoidance using hierarchical dynamic programming," in *Proceedings of The Twenty-Third Southeastern Symposium on System*, March 10–12 1991, pp. 192–6.

[3] E. Rimon and D. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–18, 1992.

[4] M. Eichhorn, "A reactive obstacle avoidance system for an autonomous underwater vehicle," in *Proceedings of the IFAC World Congress*, July 2005, p. CDRom.

[5] D. Bruijnen, J. van Helvoort, and R. van de Molengraft, "Realtime motion path generation using subtargets in a changing enviornment," in *Proceedings of the 2006 American Control Conference*, June 14–16 2006, pp. 4243–48.

[6] J.-y. Zhang, Z.-P. Zhao, and D. Liu, "A path planning method for mobile robot based on artificial potential field," *Journal of the Harbin Institute of Technology*, vol. 38, no. 8, pp. 1306–9, 2006.

[7] B. Cetin, M. Bikdash, and F. Hadaegh, "Hybrid mixed-logical linear programming algorithm for collision–free optimal path planning," *IET Control Theory and Applications*, vol. 1, no. 2, pp. 522–31, 2007.

[8] C. Ma and R. Miller, "Milp optimal path planning for real-time applications," in *Proceedings of the 2006 American Control Conference*, June 14–16 2006, p. 6.

[9] E. Balas, "Disjunctive programming," in *Annals of Discrete Math*, vol. 5, 1979, pp. 3–51.

[10] R. Bellman, *Dynamic Programming*. Dover Publications, Inc., Mineola, NY, 2003.

[11] B. Bakker, Z. Zivkovic, and B. Krose, "Hierarchical dynamic programming for robot path planning," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, August 2–6 2005, pp. 2756–61.

[12] P. Nagabhushan and P. Manohara, "Cognition of free space for planning the shortest path: A framed free space approach," *Pattern Recognition Letters*, vol. 22, no. 9, pp. 971–82, 2001.

[13] A. Meystel, A. Gues, and H. G., "Minimum time path planning for a robot," in *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, April 7–10 1986, pp. 1678–87.

[14] N. Sadegh, "Time–optimal motion planning for mobile robots," Georgia Institute of Technology, G.W. Woodruff School of Mechanical Engineering, Atlanta, Georgia 30332, Technical Report, 2008.

[15] H. Royden, *Real Analysis*. Macmillan Publishing Co., Inc., New York, NY, 1968.