



Refined Analysis of Asymptotically-Optimal Kinodynamic Planning in the State-Cost Space

Michal Kleinbort, Edgar Granados, Kiril Solovey, Riccardo Bonalli, Kostas E.
Bekris, Dan Halperin

► To cite this version:

Michal Kleinbort, Edgar Granados, Kiril Solovey, Riccardo Bonalli, Kostas E. Bekris, et al.. Refined Analysis of Asymptotically-Optimal Kinodynamic Planning in the State-Cost Space. 2020 IEEE International Conference on Robotics and Automation (ICRA), May 2020, Paris, France. pp.6344-6350, 10.1109/ICRA40945.2020.9197236 . hal-03467674

HAL Id: hal-03467674

<https://hal.science/hal-03467674>

Submitted on 6 Dec 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Refined Analysis of Asymptotically-Optimal Kinodynamic Planning in the State-Cost Space

Michal Kleinbort¹, Edgar Granados², Kiril Solovey³, Riccardo Bonalli³, Kostas E. Bekris², and Dan Halperin¹

Abstract—We present a novel analysis of AO-RRT: a tree-based planner for motion planning with kinodynamic constraints, originally described by Hauser and Zhou (AO-X, 2016). AO-RRT explores the state-cost space and has been shown to efficiently obtain high-quality solutions in practice without relying on the availability of a computationally-intensive two-point boundary-value solver. Our main contribution is an optimality proof for the single-tree version of the algorithm—a variant that was not analyzed before. Our proof only requires a mild and easily-verifiable set of assumptions on the problem and system: Lipschitz-continuity of the cost function and the dynamics. In particular, we prove that for any system satisfying these assumptions, any trajectory having a piecewise-constant control function and positive clearance from the obstacles can be approximated arbitrarily well by a trajectory found by AO-RRT. We also discuss practical aspects of AO-RRT and present experimental comparisons of variants of the algorithm.

I. INTRODUCTION

Motion planning is a fundamental problem in robotics, concerned with allowing autonomous robots to navigate in complex environments while avoiding collisions with obstacles. The problem is already challenging in the simplified geometric setting, and even more so when considering the kinodynamic constraints that the robot has to satisfy. This work is concerned with the latter setting, and consider the case where the robot’s system is specified by differential constraints of the form

$$\dot{x} = f(x, u), \quad \text{for } x \in \mathcal{X}, u \in \mathcal{U}, \quad (1)$$

where $\mathcal{X} \subseteq \mathbb{R}^d$ is the robot’s state space, and $\mathcal{U} \subseteq \mathbb{R}^D$ is the control space, for some $d, D \geq 2$. The objective of motion planning is thus to find a control function $\Upsilon : [0, T] \rightarrow \mathcal{U}$, which induces a *valid* trajectory $\pi : [0, T] \rightarrow \mathcal{X}$, such that (i) Equation (1) is satisfied, (ii) π is contained in the free space $\mathcal{F} \subseteq \mathcal{X}$, and (iii) the motion takes the robot from its initial state x_{init} to the goal region $\mathcal{X}_{\text{goal}} \subseteq \mathcal{X}$.

In *optimal motion planning*, the objective is to find a control function Υ and a trajectory π satisfying the constraints (i), (ii), (iii), which also minimize the trajectory cost, specified by

$$\text{COST}(\pi) = \int_0^T g(\pi(t), \Upsilon(t)) dt, \quad (2)$$

Work by D.H. and M.K. has been supported in part by the Israel Science Foundation (grant nos. 825/15, 1736/19), by the Blavatnik Computer Science Research Fund, and by grants from Yandex and from Facebook. K.B. was supported by NSF awards IIS-1734492, IIS-1723869, CCF-1934924.

¹Blavatnik School of Computer Science, Tel-Aviv University, Israel.

²Computer Science Department, Rutgers University, NJ 08854, USA.

³Aeronautics and Astronautics Department, Stanford University, CA 94305, USA.

where $g : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}_+$ is a cost derivative. Depending on the precise formulation of g , $\text{COST}(\pi)$ may represent the distance traversed by the robot, the energy required to execute the motion, or other metrics.

Almost thirty years of research on motion planning have led to a variety of approaches to tackle the problem, ranging from computational-geometric algorithms, potential fields, optimization-based methods, and search-based solutions [1], [2]. To the best of our knowledge, the only approach that is capable of satisfying global optimality guarantees, while still being computationally practical, is sampling-based planning. Sampling-based algorithms capture the connectivity of the free space of the problem via random sampling of states (and sometimes controls) and connecting nearby states, to yield a graph structure.

The celebrated work of Karaman and Frazzoli [3] laid the foundations for optimality in sampling-based motion planning. They introduced several new algorithms and proved mathematically that they converge to the optimal solution as the number of samples generated by the algorithms tends to infinity. This property is termed *asymptotic optimality* (AO). Many researchers have followed their footsteps, and designed new algorithms, which can be used in various applications [4], [5], [6], [7].

Unfortunately, the applicability of most of the aforementioned results to optimal planning with kinodynamic constraints remains limited. In particular, the majority of results only apply to the geometric (holonomic) setting of the problem. While a small subset of results do consider the kinodynamic case, they assume the existence of a two point *boundary value problem (BVP) solver*, which given two states $x, x' \in \mathcal{X}$ returns the lowest-cost trajectory connecting them (see, [8], [9], [10], [11], [12], [13], [14]). In practice BVP solvers are usually only available for simple robotic systems, and in many cases they are prohibitively costly to use, which limits their applicability.

Recently, there have been sampling-based approaches that do not rely on the existence of a BVP solver [15]. These methods employ forward propagation instead. Li et al. [16], [17] provided an analysis of tree sampling-based planners that perform random propagation from first principles and proposed the SST algorithm. SST is in practice computationally efficient and achieves asymptotic near-optimality, which is the property of converging toward a path with bounded suboptimality. True AO properties can be achieved by SST*, which sacrifices computational efficiency by progressively shrinking a pruning radius parameter. The approach proposed here aims for AO properties and computational efficiency,

while avoiding the critical dependence on parameters, such as pruning radii, that are difficult to tune for a variety of motion planning problems.

Most recently, Hauser and Zhou [18] proposed a meta algorithm AO- \mathbf{x} , which allows to adapt any *well-behaved* non-optimal kinodynamic sampling-based planner, denoted by \mathbf{x} , into an AO algorithm. This is achieved by substituting the d -dimensional state space \mathcal{X} on which the former is run with the $(d+1)$ -dimensional space $\mathcal{Y} = \mathcal{X} \times \mathbb{R}$, where the last coordinate encodes the solution cost. Then, \mathbf{x} is iteratively applied to shrinking subsets \mathcal{Y}_i of \mathcal{Y} for $i \in \mathbb{N}_+$, where the maximal value of the last coordinate (representing the cost) is gradually decreased with i , and hence the cost of the returned solution. The authors combined their framework with the forward-propagating versions of RRT [19] and EST [20], to yield AO-RRT and AO-EST, both of which demonstrated favorable performance over competitors. The observation that the cost induced by a system can be analyzed by augmenting the state space in the above manner was first considered by Pontryagin (see, [21]).

We follow up on Hauser and Zhou’s approach. We augment their work by addressing aspects of the analysis that we believe require more attention, namely what are the precise conditions under which using the augmented-space approach will lead to provably AO solutions. The main issue that we address is the assumption [18] that \mathbf{x} is well-behaved, without proving this property for neither RRT nor for EST. Well behavedness consists of two requirements: (i) \mathbf{x} must find a feasible solution eventually within each \mathcal{Y}_i —a property corresponding to probabilistic completeness (PC) [22]—and (ii) the cost of the solution found in \mathcal{Y}_i is smaller (with non-negligible probability) than the maximal cost value over \mathcal{Y}_i . Note that requirement (ii) is a particularly strong assumption, essentially requiring \mathbf{x} to be “nearly” AO, i.e., gradually reducing the cost of the solution when applied to the bounded subspaces \mathcal{Y}_i for $i \in \mathbb{N}_+$.

In this context, it should be noted that some variants of RRT are not even PC [23] (and thus not well behaved). Furthermore, it is not specified for what types of robotic systems [18], with respect to \mathcal{X}, f, g , or problem instances $\mathcal{F}, x_{\text{init}}, \mathcal{X}_{\text{goal}}$ this property holds. Another logical gap that has not been adequately addressed is that the proof focuses on a version of AO-RRT which grows multiple trees, and does not seem to directly extend to the single-tree version of AO-RRT used in the experiments of that paper.

A. Contribution

We present a novel analysis of AO-RRT: a tree-based planner for motion planning with kinodynamic constraints, originally described by Hauser and Zhou [18]. We focus on a variant that constructs a single tree, rather than multiple trees, embedded in the augmented state space \mathcal{Y} , and which was not analyzed before. We note that this variant was used in the experiments in [18]. The approach does not require a BVP solver and can be viewed as an AO generalization of the non-AO RRT planner [19].

Our main contribution is a rigorous optimality proof for the single-tree AO-RRT. Our proof only requires an easily-verifiable set of assumptions on the problem and system: we require Lipschitz-continuity of the cost function and the dynamics. In particular, we prove that for any system satisfying these assumptions, any trajectory having a piecewise-constant control function and positive clearance from obstacles can be approximated arbitrarily well by a trajectory found by AO-RRT. (We also discuss extensions to trajectories whose control function is not necessarily piecewise constant.) Furthermore, we develop explicit bounds on the convergence rate of the algorithm. Our AO proof relies on the theory that we have recently developed for the probabilistic completeness of RRT [24].

We also discuss practical aspects of AO-RRT, namely node pruning and a hybrid approach that combines the algorithm with other planners, while still maintaining AO. Then we present an experimental comparison of AO-RRT variants with the vanilla RRT, and SST for both geometric and kinodynamic scenarios.

The paper is organized as follows. The AO-RRT algorithm is described in Section II. Section III proceeds with the theoretical properties of AO-RRT and gives the asymptotic optimality proof. Practical aspects of the algorithm are discussed in Section IV and experiments are presented in Section V. Finally, in Section VI we discuss further research.

II. THE SINGLE-TREE AO-RRT ALGORITHM

We describe the single-tree AO-RRT approach. Henceforth we will refer to this algorithm simply as AO-RRT. Recall that $\mathcal{X}, \mathcal{F}, \mathcal{U}$ denote the state, free, and control spaces, respectively. We assume that \mathcal{X} is compact, and \mathcal{F} is open. The AO-RRT algorithm is very similar to the (kinodynamic) RRT algorithm, based on [19]. Whereas RRT grows a tree embedded in \mathcal{X} , AO-RRT (see Algorithm 1) does so in the state-cost space. In particular, we define the augmented (state) space $\mathcal{Y} := \mathcal{X} \times \mathbb{R}_+$, which is $(d+1)$ -dimensional, where the additional coordinate represents the cost of the (non-augmented) state. That is, a point $y \in \mathcal{Y}$ can be viewed as a pair $y = (x, c)$, where $x \in \mathcal{X}$ and $c \geq 0$ represents the cost of the trajectory from x_{init} to x over the tree $\mathcal{T}(\mathcal{Y})$. Given a point $y \in \mathcal{Y}$ we use the notation $x(y), c(y)$ to represent its component of \mathcal{X} and cost, respectively.

The AO-RRT algorithm has the following inputs: In addition to an initial start state x_{init} , goal region $\mathcal{X}_{\text{goal}}$, number of iterations k , maximal total duration for propagation T_{prop} , and control space \mathcal{U} , which RRT accepts, AO-RRT also accepts a maximal cost c_{max} . See Section IV for more information on how to choose c_{max} .

AO-RRT constructs a tree $\mathcal{T}(\mathcal{Y})$, embedded in \mathcal{Y} and rooted in $y_{\text{init}} = (x_{\text{init}}, 0)$, by performing k iterations of the following form. In each iteration, it generates a random sample y_{rand} in \mathcal{Y} , by randomly sampling \mathcal{X} and the cost space $[0, c_{\text{max}}]$ (lines 3-4). In addition a random control u_{rand} and duration t_{rand} are generated by calling the routine SAMPLE (lines 5-6). For a given set S , the procedure SAMPLE(S) produces a sample uniformly and randomly from S .

Next, the nearest neighbor y_{near} of y_{rand} in $\mathcal{T}(\mathcal{Y})$ is retrieved (line 7). We emphasize that this operation is performed in the $(d + 1)$ -dimensional space \mathcal{Y} using a suitable distance metric such as the Euclidean metric in the augmented space (see Section IV). Then, in line 8, the algorithm uses a forward propagation approach (using PROPAGATE) from y_{near} to generate a new state y_{new} : the random control input u_{rand} is applied for time duration t_{rand} from $x(y_{\text{near}})$ reaching a new state $x_{\text{new}} \in \mathcal{X}$ through a trajectory π_{new} . The state $x(y_{\text{near}})$ is then coupled with the cost of executing π_{new} together with $c(y_{\text{new}})$ (line 9). Mathematically, for $x \in \mathcal{X}, u \in \mathcal{U}, t > 0$, we have that

$$\text{PROPAGATE}(x, u, t) := \int_0^t f(x(t), u) dt.$$

Finally, COLLISION-FREE(π_{new}) checks whether the trajectory reaching y_{new} from y_{near} using the control u_{rand} and duration t_{rand} is collision free. This operation is known as *local planning*, and is typically achieved by densely sampling the trajectory and applying a dedicated collision detection mechanism [25]. If indeed the trajectory is collision free, y_{new} is added as a vertex to the tree and is connected by an edge from y_{near} (lines 10-12). The trajectory π_{new} is also added to the edge. If y_{new} is in the goal region and its cost is the smallest encountered so far, then y_{min} is substituted with this point (lines 13,14). Finally, a lowest-cost trajectory (if exists) is returned in line 15. Note that the algorithm maintains the lowest-cost trajectory discovered so far by keeping track of the last vertex y_{min} on such a trajectory.

III. THEORETICAL PROPERTIES OF AO-RRT

We spell out the assumptions that we make with respect to the system and the cost function, and state our main theorem. Then, in Section III-A, we describe the problem in the augmented space \mathcal{Y} , define the augmented system F , and study its properties. We then leverage this in the proof of the main theorem in Section III-B. In Section III-C we discuss the extension of the theorem to trajectories not necessarily having piecewise-constant control functions.

Throughout this section we use the following notations. For simplicity, in our proofs we use the standard Euclidean norm, denoted by $\|\cdot\|$. We note, however, that all proofs can be generalized to work with the weighted Euclidean norm. Given a set $S \subseteq \mathbb{R}^{d'}$, for some $d' > 0$, we denote by $|S|$ its Lebesgue measure. For a given point $y \in \mathbb{R}^{d'}$, and a radius $r > 0$, we use $\mathcal{B}_r^{d'}(y)$ to denote the d' -dimensional Euclidean ball of radius r centered at y .

We make the following assumption concerning f (Eq. (1)):

Assumption 1 Lipschitz continuity of the system. *The system f is Lipschitz continuous for both of its arguments. That is, $\exists K_u^f, K_x^f > 0$ s.t. $\forall x_0, x_1 \in \mathcal{X}, \forall u_0, u_1 \in \mathcal{U}$:*

$$\begin{aligned} \|f(x_0, u_0) - f(x_0, u_1)\| &\leq K_u^f \|u_0 - u_1\|, \\ \|f(x_0, u_0) - f(x_1, u_0)\| &\leq K_x^f \|x_0 - x_1\|. \end{aligned}$$

We make the following assumption concerning g (Eq. (2)):

Assumption 2 Lipschitz continuity of the cost. *The cost derivative g is Lipschitz continuous for both of its arguments. That is, $\exists K_u^g, K_x^g > 0$ s.t. $\forall x_0, x_1 \in \mathcal{X}, \forall u_0, u_1 \in \mathcal{U}$:*

$$\begin{aligned} \|g(x_0, u_0) - g(x_0, u_1)\| &\leq K_u^g \|u_0 - u_1\|, \\ \|g(x_0, u_0) - g(x_1, u_0)\| &\leq K_x^g \|x_0 - x_1\|. \end{aligned}$$

Definition 1. A piecewise constant control function $\bar{\Upsilon}$ with resolution Δt is the concatenation of constant control functions $\bar{\Upsilon}_i : [0, \Delta t] \rightarrow u_i$, where $u_i \in \mathcal{U}$, and $1 \leq i \leq k$, for some $k \in \mathbb{N}_{>0}$.

From this point on, when we say a *valid trajectory* we mean a valid trajectory as described in Section I, with the extra proviso that the control function is piecewise constant.

Definition 2. Let π be a valid trajectory, and let T be its duration. We define the clearance of π to be the maximal value $\delta > 0$ such that

$$\bigcup_{t \in [0, T]} \mathcal{B}_\delta^d(\pi(t)) \subset \mathcal{F} \text{ and } \mathcal{B}_\delta^d(\pi(T)) \subset \mathcal{X}_{\text{goal}}.$$

We say that a trajectory is *robust* if its clearance is positive.

We arrive to our main contribution that establishes the rate of convergence of AO-RRT.

Theorem 1. *Assume that Assumptions 1, 2 hold and fix $\varepsilon \in (0, 1)$. Denote by π_k the solution obtained by AO-RRT after k iterations. For every robust trajectory π having a piecewise-constant control function there exist a finite $k_0 \in \mathbb{N}, a > 0, b > 0$, such that for every $k > k_0$ it holds that*

$$\Pr[\text{COST}(\pi_k) > (1 + \varepsilon)\text{COST}(\pi)] \leq ae^{-bk}.$$

A. Properties of the augmented system

It would be convenient to view the problem of optimal planning with respect to f, g , as a feasible motion planning for an augmented system F , which is defined as follows. The *augmented system* F encompasses both types of transitions

Algorithm 1 AO-RRT($x_{\text{init}}, \mathcal{X}_{\text{goal}}, k, T_{\text{prop}}, \mathcal{U}, c_{\text{max}}$)

```

1:  $y_{\text{init}} \leftarrow (x_{\text{init}}, 0); \mathcal{T}(\mathcal{Y}).\text{init}(y_{\text{init}}); y_{\text{min}} = (\text{NULL}, \infty)$ 
2: for  $i = 1$  to  $k$  do
3:    $x_{\text{rand}} \leftarrow \text{SAMPLE}(\mathcal{X})$  ▷ sample state
4:    $c_{\text{rand}} \leftarrow \text{SAMPLE}([0, c_{\text{max}}])$  ▷ sample cost
5:    $t_{\text{rand}} \leftarrow \text{SAMPLE}([0, T_{\text{prop}}])$  ▷ sample duration
6:    $u_{\text{rand}} \leftarrow \text{SAMPLE}(\mathcal{U})$  ▷ sample control
7:    $y_{\text{near}} \leftarrow \text{NEAREST}(y_{\text{rand}} = (x_{\text{rand}}, c_{\text{rand}}), \mathcal{T}(\mathcal{Y}))$ 
8:    $(x_{\text{new}}, \pi_{\text{new}}) \leftarrow \text{PROPAGATE}(x(y_{\text{near}}), u_{\text{rand}}, t_{\text{rand}})$ 
9:    $c_{\text{new}} \leftarrow c(y_{\text{near}}) + \text{COST}(\pi_{\text{new}})$ 
10:  if COLLISION-FREE( $\pi_{\text{new}}$ ) then
11:     $\mathcal{T}(\mathcal{Y}).\text{add\_vertex}(y_{\text{new}} = (x_{\text{new}}, c_{\text{new}}))$ 
12:     $\mathcal{T}(\mathcal{Y}).\text{add\_edge}(y_{\text{near}}, y_{\text{new}}, \pi_{\text{new}})$ 
13:    if  $x(y_{\text{new}}) \in \mathcal{X}_{\text{goal}}$  and  $c(y_{\text{new}}) < c(y_{\text{min}})$  then
14:       $y_{\text{min}} \leftarrow y_{\text{new}}$ 
15: return TRACE-PATH( $\mathcal{T}(\mathcal{Y}), y_{\text{min}}$ )
```

in f and g , respectively. The control space for this system is simply \mathcal{U} , and its state space is $\mathcal{Y} = \mathcal{X} \times \mathbb{R}_+$. Formally,

$$\dot{y} = (\dot{x}, \dot{c}) = F(y, u) = (f(x, u), g(x, u)), \quad (3)$$

for $y = (x, c)$, where $x \in \mathcal{X}, c \in \mathbb{R}_+, u \in \mathcal{U}$.

We have the following claim with respect to F :

Claim 1. *Under Assumptions 1,2, the augmented system F is Lipschitz continuous for both of its arguments. That is, $\exists K_u, K_x > 0$ s.t. $\forall y_0, y_1 \in \mathcal{Y}, u_0, u_1 \in \mathcal{U}$:*

$$\begin{aligned} \|F(y_0, u_0) - F(y_0, u_1)\| &\leq K_u \|u_0 - u_1\|, \\ \|F(y_0, u_0) - F(y_1, u_0)\| &\leq K_x \|y_0 - y_1\|. \end{aligned}$$

Proof. It follows that

$$\begin{aligned} \|F(y_0, u_0) - F(y_0, u_1)\| &= \sqrt{\|f(x_0, u_0) - f(x_0, u_1)\|^2 + \|g(x_0, u_0) - g(x_0, u_1)\|^2} \\ &\leq \sqrt{(K_u^f \|u_0 - u_1\|)^2 + (K_u^g \|u_0 - u_1\|)^2} \\ &= \sqrt{(K_u^f)^2 + (K_u^g)^2} \cdot \|u_0 - u_1\| = K_u \cdot \|u_0 - u_1\|, \end{aligned}$$

for $K_u := \sqrt{(K_u^f)^2 + (K_u^g)^2}$.

The second inequality requires an additional transition since $\|F(y_0, u_0) - F(y_1, u_0)\| \leq K_x \cdot \|x_0 - x_1\|$. It remains to use the fact that $\|x_0 - x_1\| \leq \|y_0 - y_1\|$. \square

We can think of AO-RRT planning for the system f with cost g , state space \mathcal{X} and control space \mathcal{U} , as the standard RRT operating over the system F , state space \mathcal{Y} , and control space \mathcal{U} . Lines 8-9 in Algorithm 1 are identical to propagating with F . This equivalence allows to exploit useful properties of RRT recently developed in [24].

B. Proof of Theorem 1

We first provide an outline of the proof. Fix $\varepsilon \in (0, 1)$ and let π_δ be a robust trajectory whose clearance is $\delta > 0$. The clearance is with respect to both distance from the obstacles, and from the boundary of the goal region. Let $c_\delta := \text{COST}(\pi_\delta)$. We draw π_δ in the $(d+1)$ -dimensional space \mathcal{Y} , such that the new trajectory $\pi_\delta^\mathcal{Y}$ begins in $y_{\text{init}} = (x_{\text{init}}, 0)$ and ends in $y_{\text{goal}} = (x_{\text{goal}}, c_\delta)$, where $x_{\text{goal}} \in \mathcal{X}_{\text{goal}}$. Next, similarly to [24], we place a constant number of balls of radius $r = \min(\varepsilon c_\delta, \delta)$ along the trajectory $\pi_\delta^\mathcal{Y}$. The balls are constructed in a manner that guarantees that each such transition is collision free. Then we show that with high probability AO-RRT will visit all such balls as the number of samples k tends to infinity, by transitioning from one ball to the next incrementally. Reaching the last ball, centered at y_{goal} , implies that AO-RRT will find a solution whose cost is at most $c_\delta + r \leq c_\delta + \varepsilon c_\delta = (1 + \varepsilon)c_\delta$, since by definition any trajectory in \mathcal{Y} that terminates in $\mathcal{B}_r^{d+1}(y_{\text{goal}})$ must have a cost (which is its $(d+1)$ th coordinate) of at most $c_\delta + r$.

To achieve this, we first adapt with minor changes the following two lemmata from [24] to the setting of AO-RRT. Lemma 1 shows that there exists a constant $\tau \leq T_{\text{prop}}$ such that if we place the centers of the balls along $\pi_\delta^\mathcal{Y}$ where the duration between two consecutive centers is τ then the probability for successfully propagating from one ball to the

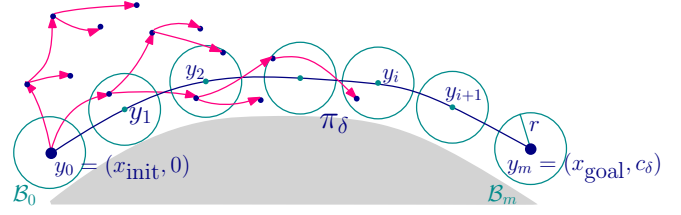


Fig. 1. Illustration of the proof of Theorem 1.

next is positive (assuming that the propagation duration and the control input are chosen uniformly at random). We note that it follows from [24] that τ can be chosen such that $\pi_\delta^\mathcal{Y}$ can be divided into sub-trajectories of duration τ , where the control function is fixed during each sub-trajectory.

Lemma 1. *There exists $\tau \leq T_{\text{prop}}$ for which the following holds: Let π be a trajectory for F with clearance $\delta > 0$ and a control function that is fixed during the interval $[0, \tau]$. Let $r \leq \delta$. Let t_{rand} be a random duration sampled uniformly from $[0, T_{\text{prop}}]$, and u_{rand} a uniformly sampled control input from \mathcal{U} . Suppose that the propagation step of AO-RRT begins at state $y_{\text{near}} \in \mathcal{B}_r^{d+1}(\pi(0))$ and ends in y_{new} (lines 8,9 in Algorithm 1). Then*

$$p_{\text{prop}} := \Pr \left[y_{\text{new}} \in \mathcal{B}_{2r/5}^{d+1}(\pi(\tau)) \right] > 0.$$

Lemma 2 shows that the probability that the nearest neighbor of a random sample y_{rand} lies in a specific ball is positive, when y_{rand} is sampled uniformly at random from \mathcal{Y} .

Lemma 2. *Let $y \in \mathcal{Y}$ be such that $\mathcal{B}_r^{d+1}(y) \subset \mathcal{F}_\mathcal{Y} := \mathcal{F} \times \mathbb{R}_+$. Suppose that there exists an AO-RRT vertex $v \in \mathcal{B}_{2r/5}^{d+1}(y)$. Let y_{near} denote the nearest neighbor of y_{rand} among all AO-RRT vertices. Then*

$$p_{\text{near}} := \Pr \left[y_{\text{near}} \in \mathcal{B}_r^{d+1}(y) \right] \geq |\mathcal{B}_r^{d+1}| / |\mathcal{Y}|.$$

Note that both probabilities $p_{\text{prop}}, p_{\text{near}}$ are independent of the number k of iterations of the algorithm. Next, we will place $m+1$ balls of radius $r = \min(\varepsilon c_\delta, \delta)$ centered at states $y_0 = y_{\text{init}}, y_1, \dots, y_m = y_{\text{goal}}$ along the trajectory $\pi_\delta^\mathcal{Y}$. See Figure 1 for an illustration.

Denote by t_δ the duration of π_δ . We determine the sequence of points y_0, \dots, y_m in the following manner: Choose a set of durations $t_0 = 0, t_1, t_2, \dots, t_m = t_\delta$, such that the difference between every two consecutive ones is τ (see Lemma 1). That is, let $y_0 = \pi_\delta^\mathcal{Y}(t_0), y_1 = \pi_\delta^\mathcal{Y}(t_1), \dots, y_m = \pi_\delta^\mathcal{Y}(t_m)$ be states along the path $\pi_\delta^\mathcal{Y}$ that are obtained after duration t_0, t_1, \dots, t_m , respectively. Obviously, $m = t_\delta / \tau$ is some constant independent of the number of samples.

Suppose that there exists an AO-RRT vertex $v \in \mathcal{B}_{2r/5}^{d+1}(y_i) \subset \mathcal{B}_r^{d+1}(y_i)$. We shall bound the probability p that in the next iteration the AO-RRT tree will extend from a vertex in $\mathcal{B}_r^{d+1}(y_i)$, given that a vertex in $\mathcal{B}_{2r/5}^{d+1}(y_i)$ exists, and that the propagation step will add a vertex to $\mathcal{B}_{2r/5}^{d+1}(y_{i+1})$. That is, p is the probability that in the next iteration both $y_{\text{near}} \in \mathcal{B}_r^{d+1}(y_i)$ and $y_{\text{new}} \in \mathcal{B}_{2r/5}^{d+1}(y_{i+1})$. From Lemma 2 we have that the probability that y_{near} lies in $\mathcal{B}_r^{d+1}(y_i)$, given that there exists an RRT vertex in $\mathcal{B}_{2r/5}^{d+1}(y_i)$,

is at least p_{near} . Next, we wish to sample duration t_{rand} and control u_{rand} such that a random propagation from y_{near} will yield $y_{\text{new}} \in \mathcal{B}_{2r/5}^{d+1}(y_{i+1})$. According to Lemma 1, the probability for this to occur is at least p_{prop} . Thus, jointly the probability that y_{near} falls in $\mathcal{B}_r^{d+1}(y_i)$ and of sampling the correct propagation duration and control is at least $p = p_{\text{near}} \cdot p_{\text{prop}} > 0$. As we mentioned earlier, this value is also independent of the number of iterations.

It remains to bound the probability of having m successful such steps. This process can be described as k Bernoulli trials with success probability p . The planning problem can be solved after m successful outcomes, where the i th outcome adds an AO-RRT vertex in $\mathcal{B}_{2r/5}^{d+1}(y_i)$. Let X_k denote the number of successes in k trials. As in [24], we have that

$$\Pr[X_k < m] = \sum_{i=0}^{m-1} \binom{k}{i} p^i (1-p)^{k-i} \leq a e^{-bk},$$

where a and b are positive constants. This concludes the proof of Theorem 1.

C. Beyond piecewise-constant control

Theorem 1 argues that for any robust trajectory π having a piecewise-constant control function, with high probability AO-RRT will find a trajectory whose control function is piecewise constant and whose cost is at most $(1+\varepsilon)\text{COST}(\pi)$, where $\varepsilon > 0$ is a constant.

Next, we show that this theorem is not limited to trajectories with piecewise-constant control functions. Let π^* be the optimal trajectory with respect to COST , with control function u^* and duration T^* . Notice that π^* is not necessarily robust, u^* is not necessarily piecewise constant. Nevertheless we show that such π^* can be approximated arbitrarily well, with respect to $\alpha > 0$, using a trajectory π_α , which has piecewise-constant control. This implies that to achieve a cost close to $\text{COST}(\pi^*)$ it suffices to apply Theorem 1 and get close to π_α .

The following proposition states that for any optimal solution (T^*, π^*, u^*) , which satisfies certain assumptions, (1) for any $\alpha > 0$, there exists a robust solution $(T_{u_\alpha}, \pi_{u_\alpha}, u_\alpha)$, where the control function u_α is in L^∞ , i.e., bounded, but not necessarily piecewise constant, and the cost of π_{u_α} is at most $1 + \alpha$ times the cost of π^* . This statement is then used to prove part (2) of the proposition, which asserts that a similar result holds even when u_α is piecewise constant. In the following, it would be convenient to represent the free space as $\mathcal{F} = \{x \in \mathbb{R}^d | h(x) \leq 0\}$, where $h(x)$ can be interpreted as the negative value of the clearance of x .

Proposition 1. Assume that $\mathcal{F} = \{x \in \mathbb{R}^d | h(x) \leq 0\}$, where $h : \mathbb{R}^d \rightarrow \mathbb{R}$ is of class C^1 , and that $\mathcal{U} = \mathbb{R}^D$. Assume also that the dynamics f and the cost COST are C^1 functions, and that there exists an optimal strategy (T^*, π^*, u^*) that has a unique extremal which is moreover normal. Then, the following holds:

- 1) For every $\alpha > 0$, there exist $\delta_\alpha > 0$ and a control $u_\alpha \in L^\infty([0, T_\alpha], \mathcal{U})$ such that the related trajectory π_{u_α} is defined in $[0, T_\alpha]$ and satisfies

$$|\text{COST}(\pi^*) - \text{COST}(\pi_{u_\alpha})| < \alpha, \quad \pi_{u_\alpha}(T_\alpha) = \pi^*(T^*),$$

and $h(\pi_{u_\alpha}(t)) = -\delta_\alpha < 0$, for $t \in [0, T_\alpha]$.

- 2) For every $\alpha > 0$, there exist $\delta_\alpha > 0$ and a piecewise-constant control u_α defined in $[0, T_\alpha]$ such that the related trajectory π_{u_α} is defined in $[0, T_\alpha]$ and satisfies

$$|\text{COST}(\pi^*) - \text{COST}(\pi_{u_\alpha})| < \alpha, \\ \|\pi_{u_\alpha}(T_\alpha) - \pi^*(T^*)\| < \alpha,$$

and $h(\pi_{u_\alpha}(t)) = -\delta_\alpha < 0$, for $t \in [0, T_\alpha]$.

For this, we recall that, given a feasible strategy (T, π, u) for the motion planning problem, a related *extremal* (T, π, u, p, μ, p_0) , where $p^0 \leq 0$ is constant, $p : [0, T] \rightarrow \mathbb{R}$ is an absolutely continuous function, and μ is a non decreasing function of bounded variation, is by definition a quantity satisfying the *Pontryagin Maximum Principle* [21], [26], i.e., the Pontryagin adjoint equations, maximality and transmission conditions (see [26] for precise definitions). The Pontryagin Maximum Principle is a necessary condition for optimality, therefore, to any optimal solution (T^*, π^*, u^*) it is associated a non-trivial extremal $(T^*, \pi^*, u^*, p^*, \mu^*, p_0^*)$. An important class of extremals are the so-called normal extremals, that by definition satisfy $p^0 \neq 0$.

For what concerns Proposition 1, the assumption on the existence of a unique normal extremal requires some (informal) comments. Normal extremals naturally exist for optimal control problems and are often unique (see, e.g., [27], [28]). Their uniqueness is related to the regularity of solutions to the Hamilton-Jacobi-Bellman equation: a smooth solution provide an (at least locally) unique normal extremal (see, e.g., [28]). Since the regularity of the solutions to the Hamilton-Jacobi-Bellman equation are related to the regularity of the data, enough regular dynamics, cost and scenario (i.e., at least C^1) provide the existence of unique normal extremals.

The proof of Proposition 1 makes use of the surjective form of the *Implicit Function Theorem* in infinite dimensional Banach spaces (see, e.g., [29]). The assumption on the existence of a unique normal extremal will be crucial to apply the theorem to our framework. Below, we provide a sketch-of-proof considering fixed final time T (for free final time T , the proof goes similarly with slight modifications, see also [30, pp. 310–314]).

Sketch-of-proof of Proposition 1: Consider fixed final time T (therefore, with the notation in Proposition 1, $T = T^* = T_\alpha$) and let us introduce the following new family of constraints:

$$h_\delta(x) := h(x) + \delta \tag{4}$$

where $\delta \in \mathbb{R}$. Since π^* is defined in $[0, T]$, it is easy to prove that, by multiplying the dynamics f by *smooth cut-off functions* (see, e.g., [31]) around π^* , for every control $u \in L^\infty([0, T], \mathcal{U})$, the related trajectory π_u is defined in the whole interval $[0, T]$ (see, e.g., [28]). Therefore, the following infinite-dimensional, parameter-dependent *End-Point*

Mapping is correctly defined

$$E : [0, 1] \times L^\infty([0, T], \mathcal{U}) \rightarrow \mathbb{R}^d \times C^0([0, T], \mathbb{R})$$

$$(\delta, u) \mapsto \left(\pi_u(T) - \pi^*(T), h_\delta(\pi_u(\cdot)) - h(\pi^*(\cdot)) \right).$$

Moreover, by the differentiability of π_u with respect to u (see, e.g., [28]), the mapping E is C^1 . Remark that to obtain such differentiability properties we need to ask that f , h and COST are C^1 , which is among our first assumptions (the C^1 regularity of COST is required for the existence of any Pontryagin extremal in the smooth case, see, e.g., [26]).

At this step, we make use of the surjective form of the Implicit Function Theorem in infinite dimensional Banach spaces applied to the End-Point Mapping E above. The theorem can be applied because we assume the existence of a unique and moreover normal extremal related to (T, π^*, u^*) , which implies that the differential with respect to u of E at $(0, u^*)$ is surjective. From this, by adapting the framework considered in [32], [33] (that is, replacing control constraints with pure state constraints), one proves that there exist $r > 0$ and a continuous mapping $\varphi : [0, r] \rightarrow L^\infty([0, T], \mathbb{R}^D)$ (with respect to the topology of L^∞ , see, e.g., [34]) such that $\varphi(0) = u^*$ and $E(\delta, \varphi(\delta)) = 0$ for every $\delta \in [0, r]^k$. In other words:

$$\forall \delta \in [0, r] : \pi_{\varphi(\delta)}(T) = \pi^*(T),$$

$$h_\delta(\pi_{\varphi(\delta)}(t)) = h(\pi^*(t)) \leq 0, \quad t \in [0, T]. \quad (5)$$

Now, by denoting $u_\delta := \varphi(\delta) \in L^\infty([0, T], \mathbb{R}^D)$, the continuity of φ (with respect to δ), of π (with respect to u) and of COST (with respect to π) under appropriate topologies gives that, for $\alpha > 0$ there exists $\delta_\alpha \in (0, r)^k$ such that

$$|\text{COST}(\pi^*) - \text{COST}(\pi_{u_{\delta_\alpha}})| < \alpha,$$

which together with (4) and (5) provides the first claim.

To obtain the second claim, we just need to approximate controls u_δ above with piecewise constant controls. Similarly to above, since COST is continuous with respect to the topology of L^∞ , if we fix $\alpha > 0$, there exists $s_\alpha > 0$ such that $|\text{COST}(\pi^*) - \text{COST}(\pi_u)| < \alpha$ for every control $u \in L^\infty([0, T], \mathbb{R}^D)$ for which $\|u^* - u\|_{L^\infty} < s_\alpha$. Now, thanks to the continuity in L^∞ of the mapping φ and the fact that $\varphi(0) = u^*$, there exists $\delta_\alpha \in (0, r)^k$ such that $\|u^* - u_{\delta_\alpha}\| < s_\alpha/4$. Now, recall that the set of piecewise constant functions is dense in $L^\infty([0, T], \mathbb{R}^D)$. This means that there exists a piecewise-constant control u_α such that $\|u_{\delta_\alpha} - u_\alpha\|_{L^\infty} < s_\alpha/4$. Importantly, up to reducing the value of $s_\alpha > 0$, the continuity of trajectories π with respect to u (in the topology of L^∞) gives that π_{u_α} is defined in the whole interval $[0, T]$ (use smooth cut-off functions as above) and that the following holds by (5) and the continuity of h :

$$\|\pi_{u_\alpha}(T) - \pi^*(T)\| < \alpha, \quad h(\pi_{u_\alpha}(t)) = -\bar{\delta}_\alpha < 0, \quad t \in [0, T]$$

for a given $\bar{\delta}_\alpha > 0$. Since $\|u^* - u_\alpha\|_{L^\infty} \leq \|u^* - u_{\delta_\alpha}\|_{L^\infty} + \|u_{\delta_\alpha} - u_\alpha\|_{L^\infty} < s_\alpha/4 + s_\alpha/4 = s_\alpha/2 < s_\alpha$, from above $|\text{COST}(\pi^*) - \text{COST}(\pi_{u_\alpha})| < \alpha$ and the conclusion follows.

IV. PRACTICAL ASPECTS OF AO-RRT

We discuss several approaches to potentially speed up the performance of AO-RRT in practice, while retaining its AO property.

Cost sampling. AO-RRT samples $(d+1)$ -dimensional points from the augmented space \mathcal{Y} by randomly sampling \mathcal{X} and the cost space $[0, c_{\max}]$, where c_{\max} provides an upper bound on the maximal cost of the solution. Setting c_{\max} to be much larger than the cost of existing tree vertices may bias the NEAREST procedure towards selecting vertices with high cost, which may affect the time to find an initial solution. Thus, we propose to set c_{\max} to be the maximal cost among the tree nodes, until an initial solution is found. Then, we can fix c_{\max} to be the cost of the solution.

Augmented-space metric. In some applications the coordinates of the \mathcal{X} -component and the cost component in \mathcal{Y} may be on different scales, which can bias NEAREST procedure towards either the cost or the \mathcal{X} component. This in turn may affect the behaviour of the algorithm and its convergence rate. Thus, we propose to use a Weighted Euclidean metric for NEAREST, defined as

$$\text{DIST}(y_a, y_b) := \sqrt{w_x \|x_a - x_b\|^2 + w_c |c_a - c_b|^2}, \quad (6)$$

where $y_a = (x_a, c_a), y_b = (x_b, c_b) \in \mathcal{Y}$. To avoid biasing, w_x, w_c should be chosen such that the maximal possible squared distance between the \mathcal{X} components and the maximal possible squared distance between costs would be of the same order. Note that this weighted version can be viewed as using an unweighted version on an augmented space \mathcal{Y}' in which the cost coordinate has been rescaled. Thus, the theoretical analysis presented in the previous section holds for the weighted version as-is.

Node pruning. After a solution of some cost $c > 0$ is found, existing tree nodes whose cost-to-come is greater than c cannot participate in the returned solution, or in a solution of better cost. Such vertices can therefore be removed from the tree. We emphasize that the proof of the previous section still applies to this setting, as after pruning, the probability to grow the tree from a certain node whose cost-to-come value is at most c only increases.

Hybrid planning. As the performance of sampling-based planners varies from one scenario to another, we propose a hybrid approach HybAO-RRT, combining AO-RRT with other planners. This approach may perform better in scenarios where AO-RRT struggles to find a solution, and effectively guide the planning and expedite the convergence towards the optimum. HybAO-RRT combines AO-RRT with an additional tree planner, termed PLN, while operating in the augmented space \mathcal{Y} . It extends the constructed tree by alternating between AO-RRT and PLN. Each node added to the tree is assigned with a cost value, as in AO-RRT.

HybAO-RRT is AO since by applying AO-RRT every other iteration we still have a positive probability $p' = p/2$ for a successful transition from \mathcal{B}_i^{d+1} to \mathcal{B}_{i+1}^{d+1} , for every i . Moreover, the addition of tree nodes due to the steps

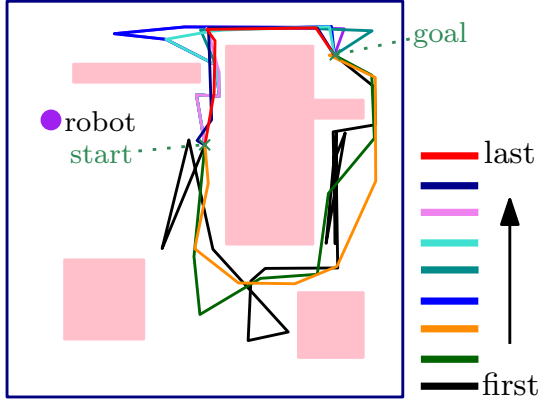


Fig. 2. Planning for a disc robot in a 2D geometric setting using AO-RRT for 120 seconds. Obstacles are depicted in light magenta, while the disc robot is depicted in purple. Start and goal positions are marked with a green cross. The Paths found by AO-RRT during the fixed time budget are drawn. The paths converge to the optimum as the number of samples increases.

of the other planner `PLN` does not affect the transition probability p' . We note that `PLN` is not required to be AO, nor is it assumed to be PC. This is in the spirit of Multi-Heuristic A* [35], where multiple inadmissible heuristic functions are used simultaneously with a single consistent heuristic to preserve guarantees on completeness of the search.

V. EXPERIMENTAL RESULTS

We present an experimental evaluation of the performance of AO-RRT on both geometric and kinodynamic scenarios. Our experiments were conducted on an Intel(R) Xeon(R) CPU E5-1660 v3 3.00GHz with 32GB of memory. We set the optimization objective to be the duration of the trajectory. Unless otherwise stated, throughout the experiments we assign $w_x = w_c$ in the distance metric (Eq. (6)) for all AO-RRT variants.

We first visualize the behaviour of AO-RRT (Alg. 1) in a simple geometric setting. We run AO-RRT for 120 seconds in a simple 2D environment consisting of a disc robot moving among rectangular obstacles (see Figure 2). The constants in Eq. (6) were set to $w_x = 1, w_c = 0.2$. We depict the paths found during the run. As the number of samples increases the paths improve, gradually converging to the optimal path.

Next, we compare the algorithms within the AO-RRT framework with RRT [19] and SST [17], which have weaker guarantees. The AO-RRT variants that we used are: (i) Multi-tree AO-RRT—the algorithm analyzed in [18], (ii) AO-RRT—an implementation of Algorithm 1, (iii) AO-RRT Pruning, which performs node pruning, and (iv) HybAO-RRT-STRIDE. The latter is a hybrid planner, as described in Section IV, combining AO-RRT and STRIDE [36]. We mention that STRIDE was originally defined for geometric settings and was not shown to be AO. STRIDE uses a data structure that enables it to produce density estimates in the full state space. More precisely, it samples a configuration s , biased towards relatively unex-

plored areas of the state space. The tree is then grown from s , if possible. HybAO-RRT-STRIDE maintains, as STRIDE does, a data structure for states in the augmented space \mathcal{Y} and alternates between the two methods for choosing the node y_{near} to grow the tree from. Once the node is chosen, the algorithm proceeds as AO-RRT does.

For each planner we report on both the success rate and the minimum solution cost averaged over all successful runs, displaying values within one standard deviation of the mean. Each result is averaged over 50 runs. Note that since we plot the average cost over all successful runs, we may observe for a certain planner an increase in the average cost. This is only possible if the success rate increases as well.

We begin with a simple geometric scenario involving a point robot translating in a 2D environment consisting of a single rectangular obstacle (see Figure 3, left). Figure 4 depicts the results. Indeed, all AO-RRT variants improve their solution as a function of time and are comparable in terms of performance. In fact, these variants were able to find the best solutions among all tested planners. The hybrid planner obtains better solutions quicker, possibly due to the STRIDE component included in it that enhances the exploration in geometric settings. RRT was inferior in terms of cost and success rate when compared to the other planners.

Then we consider two kinodynamic scenarios. The first involves a fixed-wing 2nd-order airplane moving through a building with tight stairwells to reach the top floor (see Figure 3, middle). The state space is nine-dimensional. The task space is the x, y, z location of the fixed-wing airplane. We present the results in Figure 5.

An additional scenario (Figure 3, right) involves a rally car (green) moving through a parking lot trying to reach a parking space (yellow) while avoiding other static cars and obstacles. The state space is eight-dimensional, while the task space consists of the 2D pose (x, y, θ) of the car. We present the results in Figure 6.

These two experiments demonstrate that, as expected, all AO-RRT variants improve their solution as a function of time. However, single-tree variants within the AO-RRT framework perform better than the Multi-tree AO-RRT approach. This further justifies the dedicated analysis for the single-tree AO-RRT. Additionally, we observe that AO-RRT and AO-RRT Pruning, differing in the addition of a pruning step, find solutions of similar quality, while the former obtains a slightly better success rate. The variance of the solutions found by the hybrid planner is higher than that of the other approaches for lower success rates.

Moreover, when compared to RRT, which is not AO, all AO-RRT variants were able to find solutions of better quality. The comparison against SST, which is near-AO, yielded different results; for the fixed-wing scenario all AO-RRT variants had better success rates and obtained better costs. For the rally car, all the single-tree AO-RRT variants and SST found comparable solutions, with a slight advantage to SST.

Finally, we present experiments examining the effect of the weighting scheme used in the state-cost space distance

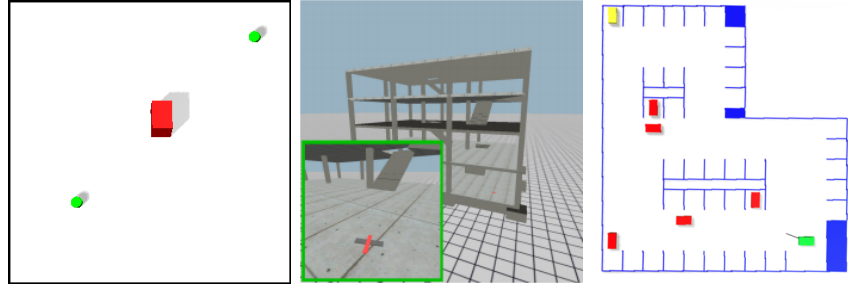


Fig. 3. Scenarios: Geometric 2D point robot (left), fixed-wing airplane in a building (middle), and rally car in a parking lot (right).

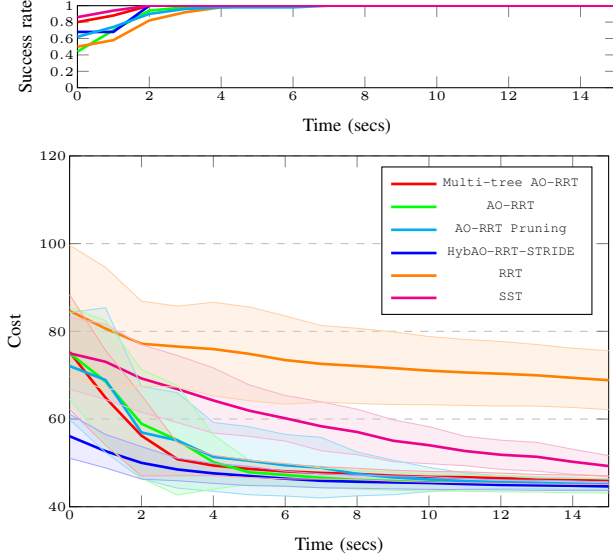


Fig. 4. Plots for a geometric scenario: a point robot operating in a 2D environment (see Figure 3, left).

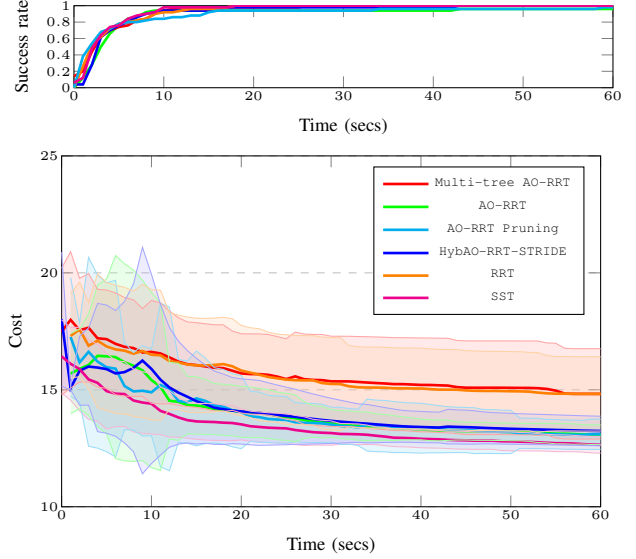


Fig. 6. Plots for a rally car (see Figure 3, right).

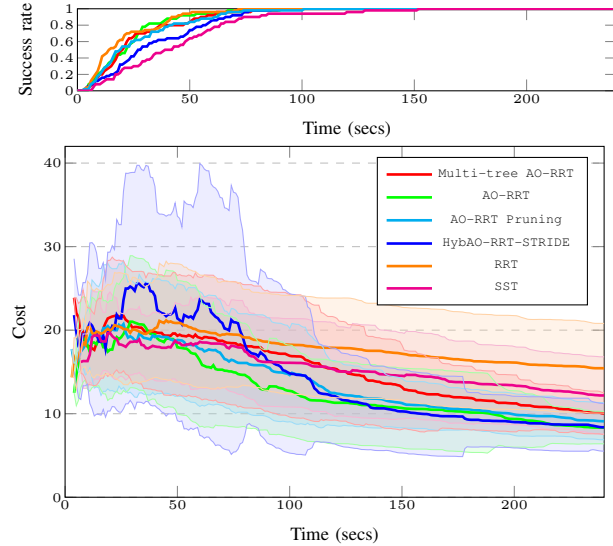


Fig. 5. Plots for a fixed-wing airplane (see Figure 3, middle).

metric (Eq. (6)). We ran AO-RRT Pruning with different weighting schemes on the fixed-wing airplane scenario (Figure 3, middle). We plot the success rate and the cost averaged over all successful runs in Figure 7. The plots demonstrate

that the choice of weights may affect the convergence rate of the algorithm. Note that with weights $w_x = 1.0, w_c = 0.0$ the algorithm acts like vanilla RRT, showing almost no improvement in cost. However, all runs with $w_c > 0$ were able to converge to the optimum.

VI. DISCUSSION

This work analyzed the desirable theoretical properties of AO-RRT, which is a method for kinodynamic sampling-based motion planning. In particular, relaxed sufficient conditions have been identified for which AO-RRT is asymptotically optimal.

Future work will extend the framework to manifold-type constraints. In this case, one has to consider the notion of Riemannian distance instead of the more classic Euclidean distance that we use here.

We noticed that AO-RRT's performance strongly depends on the choice of weights w_x, w_c used by the distance function in Eq. (6). It would be desirable to come up with an automatic scheme to choose these weights, or even modify them on-the-fly in order to obtain favorable results.

Another possible research direction involves a deeper examination of the properties of the hybrid approach. This could shed light on the settings in which the hybrid planner has an advantage over AO-RRT.

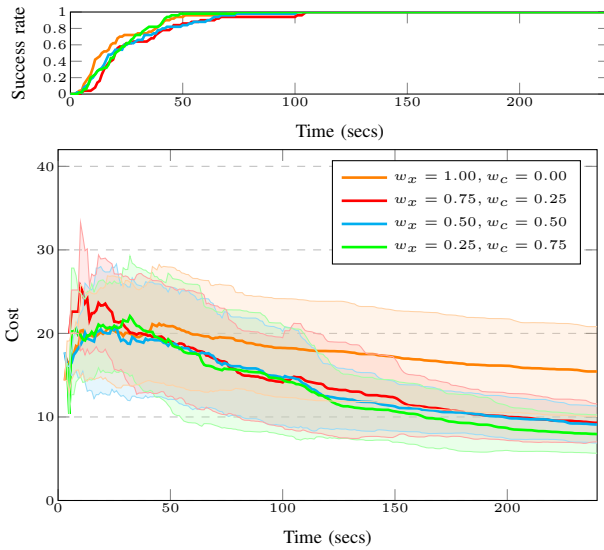


Fig. 7. The effect of the weights w_x, w_c for a fixed-wing airplane (see Figure 3, middle).

Finally, the following question, concerning the sampling scheme used by the algorithm, arises from this work: Is it possible to replace the uniform sampling of durations, or states with a different sampling method to converge more quickly to the optimum in the state-cost space?

REFERENCES

- [1] L. E. Kavraki and S. M. LaValle, "Motion planning," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds., 2008.
- [2] S. M. LaValle, *Planning Algorithms*. Cambridge Univ. Press, 2006.
- [3] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *IJRR*, vol. 30, no. 7, pp. 846–894, 2011.
- [4] L. Janson, E. Schmerling, A. A. Clark, and M. Pavone, "Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions," *IJRR*, vol. 34, no. 7, 2015.
- [5] O. Salzman and D. Halperin, "Asymptotically-optimal motion planning using lower bounds on cost," in *ICRA*, 2015, pp. 4167–4172.
- [6] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Batch informed trees (BIT*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs," in *ICRA*, 2015, pp. 3067–3074.
- [7] K. Solovey and D. Halperin, "Sampling-based bottleneck pathfinding with applications to Fréchet matching," in *ESA*, 2016, pp. 76:1–76:16.
- [8] E. Schmerling, L. Janson, and M. Pavone, "Optimal sampling-based motion planning under differential constraints: The drift case with linear affine dynamics," in *CDC*, 2015, pp. 2574–2581.
- [9] —, "Optimal sampling-based motion planning under differential constraints: The driftless case," in *ICRA*, 2015, pp. 2368–2375.
- [10] S. Karaman and E. Frazzoli, "Optimal kinodynamic motion planning using incremental sampling-based methods," in *CDC*, 2010.
- [11] —, "Sampling-based optimal motion planning for non-holonomic dynamical systems," in *ICRA*, 2013, pp. 5041–5047.
- [12] A. Perez, R. Platt, G. Konidaris, L. Kaelbling, and T. Lozano-Pérez, "LQR-RRT*: Optimal sampling-based motion planning with automatically derived extension heuristics," in *ICRA*, 2012.
- [13] D. J. Webb and J. P. van den Berg, "Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics," in *ICRA*, 2013, pp. 5054–5061.
- [14] C. Xie, J. P. van den Berg, S. Patil, and P. Abbeel, "Toward asymptotically optimal motion planning for kinodynamic systems using a two-point boundary value problem solver," in *ICRA*, 2015.
- [15] G. Papadopoulos, H. Kurniawati, and N. M. Patrikalakis, "Analysis of asymptotically optimal sampling-based motion planning algorithms for lipschitz continuous dynamical systems," *CoRR*, vol. abs/1405.2872, 2014. [Online]. Available: <http://arxiv.org/abs/1405.2872>
- [16] Y. Li, Z. Littlefield, and K. E. Bekris, "Sparse methods for efficient asymptotically optimal kinodynamic planning," in *WAFR*, 2014.
- [17] —, "Asymptotically optimal sampling-based kinodynamic planning," *IJRR*, vol. 35, no. 5, pp. 528–564, 2016.
- [18] K. Hauser and Y. Zhou, "Asymptotically optimal planning by feasible kinodynamic planning in a state-cost space," *IEEE Trans. Robotics*, vol. 32, no. 6, pp. 1431–1443, 2016.
- [19] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *IJRR*, vol. 20, no. 5, pp. 378–400, 2001.
- [20] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *IJRR*, vol. 21, no. 3, pp. 233–255, 2002.
- [21] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko, *The Mathematical Theory of Optimal Processes*. Wiley, New York, 1962.
- [22] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementation*. MIT Press, June 2005.
- [23] T. Kunz and M. Stilman, "Kinodynamic RRTs with fixed time step and best-input extension are not probabilistically complete," in *WAFR*, 2014, pp. 233–244.
- [24] M. Kleinbort, K. Solovey, Z. Littlefield, K. E. Bekris, and D. Halperin, "Probabilistic completeness of RRT for geometric and kinodynamic planning with forward propagation," *IEEE RA-L*, pp. 1–1, 2018.
- [25] D. M. Ming C. Lin and Y. J. Kim, "Collision and proximity queries," in *Handbook of Discrete and Computational Geometry*, 3rd ed. CRC press, 2018, ch. 51.
- [26] A. Dmitruk, "On the development of Pontryagin's Maximum Principle in the works of A. Ya. Dubovitskii and AA Milyutin," *Control and Cybernetics*, vol. 38, no. 4A, pp. 923–957, 2009.
- [27] Y. Chitour, F. Jean, and E. Trélat, "Singular trajectories of control-affine systems," *SICON*, vol. 47, no. 2, pp. 1078–1095, 2008.
- [28] E. Trélat, "Some properties of the value function and its level sets for affine control systems with quadratic cost," *Journal of Dynamical and Control Systems*, vol. 6, no. 4, pp. 511–541, 2000.
- [29] P. Antoine and H. Zouaki, "Etude locale de l'ensemble des points critiques d'un problème d'optimisation paramétré," *C. R. Acad. Sci. Paris Sér. I Math.*, vol. 310, pp. 587–590, 1990.
- [30] E. B. Lee and L. Markus, "Foundations of optimal control theory," Minnesota Univ Center For Control Sciences, Tech. Rep., 1967.
- [31] J. M. Lee, *Introduction to smooth manifolds*. Springer, 2001.
- [32] T. Haberkorn and E. Trélat, "Convergence results for smooth regularizations of hybrid nonlinear optimal control problems," *SICON*, vol. 49, no. 4, pp. 1498–1522, 2011.
- [33] R. Bonalli, B. Hérissé, and E. Trélat, "Continuity of Pontryagin extremals with respect to delays in nonlinear optimal control," *SICON*, vol. 57, no. 2, pp. 1440–1466, 2019.
- [34] H. Brezis, *Functional analysis, Sobolev spaces and partial differential equations*. Springer Science & Business Media, 2010.
- [35] S. Aine, S. Swaminathan, V. Narayanan, V. Hwang, and M. Likhachev, "Multi-heuristic A*," *IJRR*, vol. 35, no. 1-3, pp. 224–243, 2016.
- [36] B. Gipson, M. Moll, and L. E. Kavraki, "Resolution independent density estimation for motion planning in high-dimensional spaces," in *ICRA*, 2013, pp. 2437–2443.