

Probabilistic completeness of RRT for geometric and kinodynamic planning with forward propagation

Corrigendum

Michal Kleinbort¹, Kiril Solovey², Zakary Littlefield³, Kostas E. Bekris³, and Dan Halperin¹

Abstract—The Rapidly-exploring Random Tree (RRT) algorithm has been one of the most prevalent and popular motion-planning techniques for two decades now. Surprisingly, in spite of its centrality, there has been an active debate under which conditions RRT is probabilistically complete. We provide two new proofs of probabilistic completeness (PC) of RRT with a reduced set of assumptions. The first one for the purely geometric setting, where we only require that the solution path has a certain clearance from the obstacles. For the kinodynamic case with forward propagation of random controls and duration, we only consider in addition mild Lipschitz-continuity conditions. These proofs fill a gap in the study of RRT itself. They also lay sound foundations for a variety of more recent and alternative sampling-based methods, whose PC property relies on that of RRT.

Our original publication [1] contains an error in the analysis of the case of the kinodynamic RRT. Here, we rectify the problem by modifying the proof of Theorem 2, which, in particular, necessitated a revision of Lemma 3. Briefly, the original (and erroneous) proof of Theorem 2 used a sequence of equal-size balls. The correction uses a sequence of balls of increasing radii. We emphasize that the correction is in Lemma 3 and the proof of Theorem 2 only. The main results remain unchanged.

Index Terms—Motion and Path Planning, Nonholonomic Motion Planning

I. INTRODUCTION

Two decades ago LaValle and Kuffner presented the *Rapidly-exploring Random Tree* (RRT) [2] method for sampling-based motion planning. Even though numerous alternatives for motion planning have been proposed since then, RRT remains one of the most widely used techniques today. This is due to its simplicity and practical efficiency, especially when combined with simple heuristics.

RRT is especially useful in single-query settings, as it focuses on finding a single trajectory moving a robot from an

Work by D.H. and M.K. has been supported in part by the Israel Science Foundation (grant nos. 825/15,1736/19), by NSF/US-Israel-BSF (grant no. 2019754), by the Israel Ministry of Science and Technology (grant no. 103129), by the Blavatnik Computer Science Research Fund, and by the Yandex Machine Learning Initiative for Machine Learning at Tel Aviv University. This paper was prepared while K.S. was a Ph.D. student in Tel Aviv University, where he was supported by the Clore Israel Foundation. Z.L. and K.B. were supported by NSF IIS 1617744 and CCF 1330789.

¹ M.K. and D.H. are with the Blavatnik School of Computer Science, Tel Aviv University, Israel

² K.S. is with the Faculty of Electrical and Computer Engineering, Technion, Israel

³ Z.L. and K.B. are with the Department of Computer Science, Rutgers University, NJ 08854, USA

initial state to a goal state (or region), rather than exploring the full state space of the problem, as roadmap methods do, such as PRM [3]. To achieve this objective, RRT grows a tree, rooted at an initial state, which is periodically extended towards random state samples until the goal is reached.

Notably, RRT is well suited to complex motion planning tasks and, in particular, problems involving kinodynamic constraints. This is due to the fact that RRT can be implemented without a steering function, which is difficult to obtain for many systems with complex dynamics. (This function returns a path between two states in the absence of obstacles. It corresponds to solving a two-point boundary value problem (BVP), which may be a difficult task for many dynamical systems.) Moreover, RRT has low dependence on parameters and is easily extendable to a variety of domains (e.g., graspRRT for integrated motion and grasp planning [4]).

Since its introduction, numerous variations and extensions of RRT have been proposed (see, e.g., [5]–[9]), to allow improved performance. While RRT is not asymptotically optimal (AO) and provably does not converge to the optimal solution [10], [11], it forms the basis of many AO planners, including RRT* and RRG [11]. In particular, the probabilistic completeness (PC) of most of the aforementioned RRT-based algorithms is derived from the PC properties of RRT.

Surprisingly, it is not completely obvious under what conditions RRT is probabilistically complete, especially when using forward propagation of controls for the kinodynamic case. Indeed there has been some debate on this issue in the literature [12], [13]. This paper aims to address this gap.

A. Contribution

We provide two new proofs of PC of RRT. The first one for the purely geometric setting, where we only require that the solution path has a certain clearance from the obstacles. For the kinodynamic case with forward propagation of random controls and duration, we add mild Lipschitz-continuity conditions. This line of work lays sound foundations for arguing the probabilistic completeness of the variety of methods whose PC relies on that of RRT.

Section II describes related work and Section III proceeds with the probabilistic completeness proof for the geometric case. Section IV gives a proof for the kinodynamic setting. A discussion on further research appears in Section V.

II. RELATED WORK

Sampling-based algorithms are among the state-of-the-art alternatives for robot motion planning. Since their introduction in the mid 90's (e.g., PRM, EST [14] and RRT), they have been used in numerous robotic tasks. Sampling-based motion planners are also widely used in various fields other than robotics, such as computational biology and digital animation. There are recent reviews that provide a comprehensive coverage of developments in sampling-based motion planning [15], [16].

Sampling-based planners can potentially provide the following two desirable properties; (i) *probabilistic completeness* (PC) and (ii) *Asymptotic (near)-optimality* (AO). The former implies that the probability that the planner will return a solution (if one exists) approaches one as the number of samples tends to infinity. AO is a stronger property, as it implies that the cost of the solution returned (if one exists) by the planning algorithm (nearly) approaches the cost of the optimal solution as the number of samples tends to infinity.

AO variants of RRT and PRM, i.e., the RRT* and PRM* methods, have been introduced more recently [11]. The same line of work introduced another AO planning algorithm, RRG, which constructs a connected PRM-like roadmap in a single-query setting. Interestingly, the PC property of both RRT* and RRG relies entirely on the PC property of RRT. Since then, many variants of RRT* and RRG have been devised [17]–[22], most of which inherit their PC and AO properties from RRG and RRT*. A different series of planners implicitly maintain a PRM structure to guarantee AO planning [23]–[26]. A recent paper develops precise conditions for PRM-based planners (in terms of the connection radius used) to guarantee AO [27].

Although RRT*, PRM*, and their extensions, were initially developed to deal with geometric planning, they can be extended to kinodynamic planning. This requires proper adjustments to the algorithms and the proofs (see, e.g., [28]–[35]). Nevertheless, these approaches require the use of a steering function, which limits their application to systems for which such a function is readily available. Recent work proposes a different type of approach, called SST, that employs only forward propagation [36] and achieves asymptotic near-optimality. Hauser and Zhou propose a simple yet effective approach termed AO-RRT, which employs a forward-propagating RRT as a black-box component [37], to achieve AO.

A. PC of Kinodynamic RRT

LaValle and Kuffner discuss completeness of RRT in kinodynamic setting in one of the early works on the subject [2]. While this work provides strong evidence for the PC of RRT, it only derives a proof sketch that does not fully addresses many of the complications that arise in analyzing sampling-based planners, be it a geometric [9] or kinodynamic setting. For instance, the proofs in that paper assume the existence of “attraction sequences” and “basin regions”, whose purpose is to lead the growth of the RRT tree toward the goal. It is not clear, however, whether such regions exist at all and for what types of robotic systems. It is also not clear whether the number of such regions is finite, and whether it is possible

to produce samples in such regions with positive probability. Similar concerns were expressed by Caron et al. [13].

Indeed, in 2014, Kunz and Stilman [12] showed that one of the variants of RRT mentioned in the original RRT paper [2] is in fact not PC. In particular, they consider RRT which employs a fixed time step (rather than random propagation time which we use here) and a best-control input strategy, which picks the control input that yields the nearest state to the random sample. For this setting they describe a counterexample consisting of a specific robotic system for which RRT will have a success rate of 0. The reason being that the state space reachable by this type of RRT is a strict subset of the actual reachable space of the robotic system. Completeness of the other variants was left as an open question.

PC proofs of RRT under different steering functions and robot systems were presented in [13] and [38]. Specifically, Caron et al. [13] consider state-based steering, which is different than forward propagation of random controls that we consider here. A setting similar to ours of random forward propagation was considered in [36] and [39]. It should be noted, however, that both papers consider a random-tree planner (and its extensions), which selects the next vertex to expand in a uniform and random manner among all its vertices, unlike RRT which expands the nearest neighbor toward a random sample point. Interestingly, the random tree is AO, in contrast to RRT which is not AO [10], [11]. Nevertheless, the selection process employed by RRT allows it to quickly explore the underlying state space when endowed with an appropriate metric.

III. PROBABILISTIC COMPLETENESS OF RRT: THE GEOMETRIC CASE

We start by defining useful notation in Subsection III-A and then proceed to describe RRT for the geometric case. Then, in Subsection III-B, we provide the PC proof. We call the algorithm in this section GEOM-RRT to distinguish from the kinodynamic version. The geometric case, where a steering function exists and the dimension of the control space is identical to the dimension of the state space, can be considered as a special case of the kinodynamic setting. Thus, this section can be viewed as an introduction to the more involved kinodynamic setting, which is analyzed in the following section.

A. Preliminaries

Let \mathcal{X} be the state space, which is assumed to be $[0, 1]^d$ (a d -dimensional Euclidean hypercube), equipped with the standard Euclidean distance metric, whose norm we denote by $\|\cdot\|$. The free space is denoted by $\mathcal{F} \subseteq \mathcal{X}$. Given a subset $D \subseteq \mathcal{X}$ we denote by $|D|$ its Lebesgue measure. We will use $\mathcal{B}_r(x)$ to denote the ball of radius r centered at $x \in \mathbb{R}^d$. Let $x_{\text{init}} \in \mathcal{F}$ denote the start state, and let $\mathcal{X}_{\text{goal}}$ be an open subset of \mathcal{F} denoting the goal region. For simplicity, we assume that there exist $\delta_{\text{goal}} > 0$, $x_{\text{goal}} \in \mathcal{X}_{\text{goal}}$, such that $\mathcal{X}_{\text{goal}} = \mathcal{B}_{\delta_{\text{goal}}}(x_{\text{goal}})$.

A motion-planning problem is implicitly defined by the triplet $(\mathcal{F}, x_{\text{init}}, \mathcal{X}_{\text{goal}})$. A solution to such a problem is a trajectory that moves the robot from the initial state to the goal

region while avoiding collisions with obstacles. More formally, a valid trajectory is a continuous map $\pi : [0, t_\pi] \rightarrow \mathcal{F}$, such that $\pi(0) = x_{\text{init}}$ and $\pi(t_\pi) \in \mathcal{X}_{\text{goal}}$. The clearance of π is the maximal δ_{clear} , such that $\mathcal{B}_{\delta_{\text{clear}}}(\pi(t)) \subseteq \mathcal{F}$ for all $t \in [0, t_\pi]$. We require that $\delta_{\text{clear}} > 0$.

We describe in Algorithm 1 the (geometric) RRT algorithm, GEOM-RRT, based on [9]. The input for GEOM-RRT consists of an initial configuration x_{init} , goal region $\mathcal{X}_{\text{goal}}$, number of iterations k , and a steering parameter $\eta > 0$ used by the algorithm. GEOM-RRT constructs a tree \mathcal{T} by performing k iterations of the following form. In each iteration, a new random sample x_{rand} is returned from \mathcal{X} uniformly by calling RANDOM_STATE. Then, the vertex $x_{\text{near}} \in \mathcal{T}$ that is nearest (according to $\|\cdot\|$) to x_{rand} is found using NEAREST_NEIGHBOR. A new configuration $x_{\text{new}} \in \mathcal{X}$ is then returned by NEW_STATE, such that x_{new} is on the line segment between x_{near} and x_{rand} and the distance $\|x_{\text{near}} - x_{\text{new}}\|$ is at most η . Finally, COLLISION_FREE($x_{\text{near}}, x_{\text{new}}$) checks whether the path from x_{near} to x_{new} is collision free. If so, x_{new} is added as a vertex to \mathcal{T} and is connected by an edge from x_{near} .

Algorithm 1 GEOM-RRT ($x_{\text{init}}, \mathcal{X}_{\text{goal}}, k, \eta$)

```

1:  $\mathcal{T}.\text{init}(x_{\text{init}})$ 
2: for  $i = 1$  to  $k$  do
3:    $x_{\text{rand}} \leftarrow \text{RANDOM\_STATE}()$ 
4:    $x_{\text{near}} \leftarrow \text{NEAREST\_NEIGHBOR}(x_{\text{rand}}, \mathcal{T})$ 
5:    $x_{\text{new}} \leftarrow \text{NEW\_STATE}(x_{\text{rand}}, x_{\text{near}}, \eta)$ 
6:   if  $\text{COLLISION\_FREE}(x_{\text{near}}, x_{\text{new}})$  then
7:      $\mathcal{T}.\text{add\_vertex}(x_{\text{new}})$ 
8:      $\mathcal{T}.\text{add\_edge}(x_{\text{near}}, x_{\text{new}})$ 
9: return  $\mathcal{T}$ 

```

To retrieve a trajectory for the robot, the single path in \mathcal{T} from the root state x_{init} to the goal is found. It can then be translated to a feasible, collision-free trajectory for the robot by tracing the configurations along this path.

B. Probabilistic completeness proof

Next we devise a PC proof for GEOM-RRT. Throughout this section we will assume that there exists a valid trajectory $\pi : [0, t_\pi] \rightarrow \mathcal{F}$ with clearance $\delta_{\text{clear}} > 0$. Without loss of generality, assume that $\pi(t_\pi) = x_{\text{goal}}$, i.e., the trajectory terminates at the center of the goal region. Denote by L the (Euclidean) length of π . Also, let $\delta := \min\{\delta_{\text{clear}}, \delta_{\text{goal}}\}$.

Let $m = \frac{5L}{\nu}$, where $\nu = \min(\delta, \eta)$, and η is the steering parameter of GEOM-RRT. Then, define a sequence of $m + 1$ points $x_0 = x_{\text{init}}, \dots, x_m = x_{\text{goal}}$ along π , such that the length of the sub-path between every two consecutive points is $\nu/5$. Therefore, $\|x_i - x_{i+1}\| \leq \nu/5$ for every $0 \leq i < m$. Next, we define a set of $m + 1$ balls of radius $\nu/5$, centered at these points, and prove that with high probability GEOM-RRT will generate a path that goes through these balls.

We start by proving Lemma 1, which will be used in the proof of Theorem 1 and specifies a condition for successfully extending the tree to the goal.

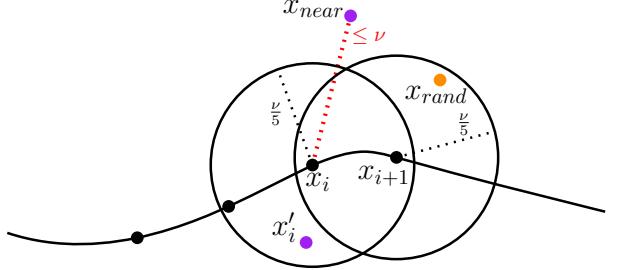


Fig. 1. Illustration of the proof of Lemma 1.

Lemma 1. Suppose that GEOM-RRT has reached $\mathcal{B}_{\nu/5}(x_i)$, that is, \mathcal{T} contains a vertex x'_i such that $x'_i \in \mathcal{B}_{\nu/5}(x_i)$. If a new sample x_{rand} is drawn such that $x_{\text{rand}} \in \mathcal{B}_{\nu/5}(x_{i+1})$, then the straight line segment between x_{rand} and its nearest neighbor x_{near} in \mathcal{T} lies entirely in \mathcal{F} .

Proof. Denote by x_{near} the nearest neighbor of x_{rand} among the RRT vertices. See Figure 1 for an illustration. Then, from the definition of x_{near} , it follows that $\|x_{\text{near}} - x_{\text{rand}}\| \leq \|x'_i - x_{\text{rand}}\|$, where $x'_i \in \mathcal{B}_{\nu/5}(x_i)$.

We show that x_{near} must lie in $\mathcal{B}_\nu(x_i)$, implying that $\overline{x_{\text{near}}x_{\text{rand}}} \subset \mathcal{F}$, as $x_{\text{rand}} \in \mathcal{B}_{\nu/5}(x_{i+1}) \subset \mathcal{B}_\nu(x_i)$. From $\|x_{\text{near}} - x_{\text{rand}}\| \leq \|x'_i - x_{\text{rand}}\|$ and the triangle inequality, we have:

$$\begin{aligned}\|x_{\text{near}} - x_i\| &\leq \|x_{\text{near}} - x_{\text{rand}}\| + \|x_{\text{rand}} - x_i\| \\ &\leq \|x'_i - x_{\text{rand}}\| + \|x_{\text{rand}} - x_i\|.\end{aligned}$$

From the triangle inequality, we have that

$$\|x_{\text{rand}} - x_i\| \leq \|x_{\text{rand}} - x_{i+1}\| + \|x_{i+1} - x_i\|,$$

$$\|x'_i - x_{\text{rand}}\| \leq \|x'_i - x_i\| + \|x_i - x_{i+1}\| + \|x_{i+1} - x_{\text{rand}}\|.$$

Therefore:

$$\|x_{\text{near}} - x_i\| \leq \|x'_i - x_i\| + 2\|x_{i+1} - x_{\text{rand}}\| + 2\|x_{i+1} - x_i\| \leq 5\frac{\nu}{5} = \nu.$$

Hence, $x_{\text{near}} \in \mathcal{B}_\nu(x_i) \subseteq \mathcal{F}$ and thus $\overline{x_{\text{near}} x_{\text{rand}}} \subset \mathcal{F}$.

Note that $\|x_{\text{near}} - x_{\text{rand}}\| \leq \eta$, since: $\|x_{\text{rand}} - x_{\text{near}}\| \leq \|x_{\text{rand}} - x'_i\| \leq \|x'_i - x_i\| + \|x_i - x_{i+1}\| + \|x_{i+1} - x_{\text{rand}}\| \leq 3 \cdot \frac{\nu}{5} < \nu \leq \eta$. The fact that $\|x_{\text{near}} - x_{\text{rand}}\| \leq \eta$, means that $x_{\text{new}} = x_{\text{rand}}$.

We now prove our main theorem.

Theorem 1. *The probability that GEOM-RRT fails to reach x_{goal} from x_{init} after k iterations is at most ae^{-bk} , for some constants $a, b \in \mathbb{R}_{>0}$.*

Proof. Assume that $\mathcal{B}_{\nu/5}(x_i)$ already contains an RRT vertex. Let p be the probability that in the next iteration an RRT vertex will be added to $\mathcal{B}_{\nu/5}(x_{i+1})$. Recall that due to Lemma 1, $x_{\text{rand}} \in \mathcal{B}_{\nu/5}(x_{i+1})$ ensures that RRT will reach $\mathcal{B}_{\nu/5}(x_{i+1})$. Since at each iteration i we draw x_{rand} uniformly at random from $[0, 1]^d$, the probability p that this sample falls inside $\mathcal{B}_{\nu/5}(x_{i+1})$ is equal to $|\mathcal{B}_{\nu/5}| / |[0, 1]^d| = |\mathcal{B}_{\nu/5}|$.

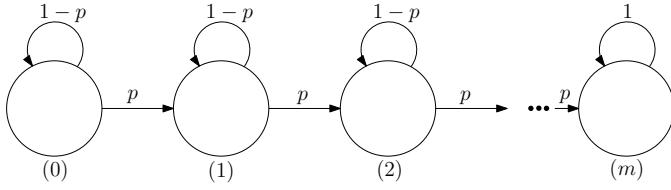


Fig. 2. A Markov chain where the success probability $p = |\mathcal{B}_{\nu/5}|$ is the probability to uniformly sample from a specific ball of radius $\nu/5$. State (m) is a terminal state. m successful outcomes imply that the algorithm finds a path from initial state to goal, where the i th successful outcome switches from state i to state $i + 1$.

In order for GEOM-RRT to reach $\mathcal{X}_{\text{goal}}$ from x_{init} we need to repeat this step m times from x_i to x_{i+1} for $0 \leq i < m$. This stochastic process can be viewed as a Markov chain (see Figure 2). Alternatively, this process can be described as k Bernoulli trials with success probability p . The planning problem can be solved after m successful outcomes (the i th outcome adds an RRT vertex in $\mathcal{B}_{\nu/5}(x_i)$). Note that it is possible that the process ends after less than m successful outcomes, i.e., by defining success to be m successful outcomes we obtain an upper bound on the probability of failure.

Next, we bound the probability of failure, that is, the probability that the process does not reach state (m) , after k steps. Let X_k denote the number of successes in k trials, then

$$\begin{aligned} \Pr[X_k < m] &= \sum_{i=0}^{m-1} \binom{k}{i} p^i (1-p)^{k-i} \\ &\leq \sum_{i=0}^{m-1} \binom{k}{m-1} p^i (1-p)^{k-i} \\ &\leq \binom{k}{m-1} \sum_{i=0}^{m-1} (1-p)^k \\ &\leq \binom{k}{m-1} \sum_{i=0}^{m-1} (e^{-p})^k = \binom{k}{m-1} m e^{-pk} \\ &= \frac{\prod_{i=k-m}^k i}{(k-1)!} m e^{-pk} \leq \frac{m}{(m-1)!} k^m e^{-pk}, \end{aligned}$$

where the transitions rely on (i) $m \ll k$, (ii) $p < \frac{1}{2}$, and (iii) $(1-p) \leq e^{-p}$.

As p, m are fixed and independent of k , the expression $\frac{1}{(m-1)!} k^m m e^{-pk}$ decays to zero exponentially with k . Therefore, GEOM-RRT with uniform samples is probabilistically complete. \square

IV. PROBABILISTIC COMPLETENESS OF RRT UNDER DIFFERENTIAL CONSTRAINTS

We begin by formulating the kinodynamic problem. Our assumptions on the robotic system and the environment as well as the definitions appear in Subsection IV-A and are adapted from Li et al. [36]. Next, we describe the modifications to RRT required for solving the kinodynamic problem. Finally, in Subsection IV-B, we devise a novel PC proof for the kinodynamic RRT.

A. Preliminaries

We adapt the problem attributes introduced in the previous section to accommodate the more involved structure of the kinodynamic case. The state space $\mathcal{X} \subseteq \mathbb{R}^d$ is a smooth d -dimensional manifold. Let $\mathcal{F} \subset \mathcal{X}$ denote the free state space. As before, we assume that there exist $x_{\text{goal}} \in \mathcal{X}, \delta_{\text{goal}} > 0$, such that $\mathcal{X}_{\text{goal}} = \mathcal{B}_{\delta_{\text{goal}}}(x_{\text{goal}})$.

Let $\mathbb{U} \subseteq \mathbb{R}^D$ denote the space of control vectors. The given system has differential constraints of the following form:

$$\dot{x}(t) = f(x(t), u(t)), \quad x(t) \in \mathcal{X}, \quad u(t) \in \mathbb{U}. \quad (1)$$

Trajectories under differential constraints are defined as follows.

Definition 1. A valid trajectory π of duration t_π is a continuous function $\pi : [0, t_\pi] \rightarrow \mathcal{F}$. A trajectory π is generated by starting at a given state $\pi(0)$ and applying a control function $\Upsilon : [0, t_\pi] \rightarrow \mathbb{U}$ by forward integrating Equation 1.

Similar to prior work [36], we consider control functions that are piecewise constant:

Definition 2. 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 $u_i \in \mathbb{U}$, and $1 \leq i \leq k$, for some $k \in \mathbb{N}_{>0}$.

We assume that the system is Lipschitz continuous for both of its arguments. That is, $\exists K_u, K_x > 0$ s.t. $\forall x_0, x_1 \in \mathcal{X}, u_0, u_1 \in \mathbb{U}$:

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

We describe here the (kinodynamic) RRT algorithm, based on [2].

Algorithm 2 RRT ($x_{\text{init}}, \mathcal{X}_{\text{goal}}, k, T_{\text{prop}}, \mathbb{U}$)

```

1:  $\mathcal{T}.\text{init}(x_{\text{init}})$ 
2: for  $i = 1$  to  $k$  do
3:    $x_{\text{rand}} \leftarrow \text{RANDOM\_STATE}()$ 
4:    $x_{\text{near}} \leftarrow \text{NEAREST\_NEIGHBOR}(x_{\text{rand}}, \mathcal{T})$ 
5:    $t \leftarrow \text{SAMPLE\_DURATION}(0, T_{\text{prop}})$ 
6:    $u \leftarrow \text{SAMPLE\_CONTROL\_INPUT}(\mathbb{U})$ 
7:    $x_{\text{new}} \leftarrow \text{PROPAGATE}(x_{\text{near}}, u, t)$ 
8:   if  $\text{COLLISION\_FREE}(x_{\text{near}}, x_{\text{new}})$  then
9:      $\mathcal{T}.\text{add\_vertex}(x_{\text{new}})$ 
10:     $\mathcal{T}.\text{add\_edge}(x_{\text{near}}, x_{\text{new}})$ 
11: return  $\mathcal{T}$ 

```

The RRT algorithm in dynamic settings with no BVP solver has the following inputs: start state x_{init} , goal region $\mathcal{X}_{\text{goal}}$, the number of iterations k , the maximal time duration for propagation T_{prop} , and the set of control inputs \mathbb{U} . Our proof below assumes that T_{prop} is positive and independent of k .

Lines 5–7 in Algorithm 2 replace line 5 in Algorithm 1. Here, a random time duration t is chosen between 0 and T_{prop} as well as a random control input $u \in \mathbb{U}$. The algorithm uses a forward propagation approach (function PROPAGATE) from x_{near} : control input u is applied for time duration t , reaching a

new state x_{new} . Finally, if the trajectory from x_{near} to x_{new} is collision-free, then x_{new} is added to \mathcal{T} together with a connecting edge to x_{near} .

B. Probabilistic completeness proof

We prove that RRT for a system with dynamics satisfying the aforementioned characteristics is PC. To do so, we start by proving three lemmas. The following lemma, which is an extension of Theorem 15 from [36], bounds the distance between the endpoints of two trajectories with similar control inputs and initial positions, for the same duration.

Lemma 2. *Let π, π' be two trajectories, with the corresponding control functions $\Upsilon(t), \Upsilon'(t)$. Suppose that $x_0 = \pi(0), x'_0 = \pi'(0)$. Let $T > 0$ be a time duration such that for all $t \in [0, T]$ it holds that $\Upsilon(t) = u, \Upsilon'(t) = u'$. That is, Υ, Υ' remain fixed throughout $[0, T]$. Then*

$$\|\pi(T) - \pi'(T)\| \leq e^{K_x T} \Delta x + K_u T e^{K_x T} \Delta u,$$

where $\Delta x = \|x_0 - x'_0\|$ and $\Delta u = \|u - u'\|$.

Proof. From the Lipschitz continuity assumption and the triangle inequality, we have that

$$\|f(x_0, u) - f(x'_0, u')\| \leq K_u \Delta u + K_x \Delta x.$$

As in the proof of Theorem 15 in [36], we will use the Euler integration method to approximate the value of the trajectory π at duration T . We divide $[0, T]$ into $\ell \in \mathbb{N}_{>0}$ pieces, each of duration h , i.e., $T = \ell \cdot h$. Let x_i, x'_i denote the resulting approximations of the trajectories π, π' at duration $i \cdot h$. From Euler's method we have that

$$\begin{aligned} x_i &= x_{i-1} + h \cdot f(x_{i-1}, u), \\ x'_i &= x'_{i-1} + h \cdot f(x'_{i-1}, u'). \end{aligned}$$

The proof in [36] shows that

$$\|x_\ell - x'_\ell\| < (1 + K_x h)^\ell \Delta x + K_u T e^{K_x T} \Delta u. \quad (2)$$

Since $(1 + K_x h)^\ell = (1 + K_x T/\ell)^\ell < e^{K_x T}$ we have that

$$\|x_\ell - x'_\ell\| < e^{K_x T} \Delta x + K_u T e^{K_x T} \Delta u.$$

From the Lipschitz continuity assumption we have that the Euler integration method converges to the solution of the *Initial value problem*. That is, $\forall 0 < i \leq \ell$,

$$\begin{aligned} \lim_{\ell \rightarrow \infty, h \rightarrow 0, \ell h = T} \|\pi(i \cdot h) - x_i\| &= 0, \\ \lim_{\ell \rightarrow \infty, h \rightarrow 0, \ell h = T} \|\pi'(i \cdot h) - x'_i\| &= 0. \end{aligned}$$

Therefore,

$$\|\pi(T) - \pi'(T)\| \leq e^{K_x T} \Delta x + K_u T e^{K_x T} \Delta u.$$

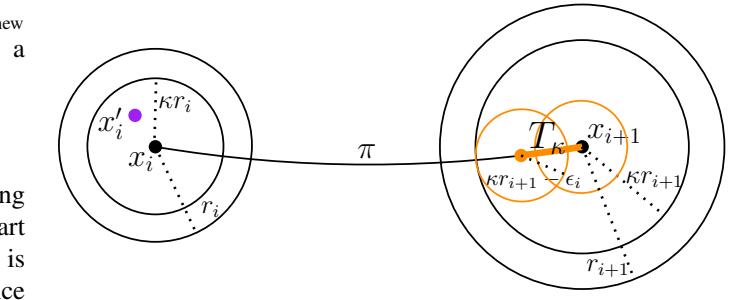


Fig. 3. Illustration of T_κ .

Lemma 3. *Let π be a trajectory with clearance $\delta > 0$, and duration $\tau \leq T_{\text{prop}}$. Suppose that the control function Υ is fixed for all $t \in [0, \tau]$, i.e., $\Upsilon(t) = u \in \mathbb{U}$. Denote by x_i, x_{i+1} the states $\pi(0), \pi(\tau)$, respectively. Let $r_i, r_{i+1} \in \mathbb{R}_{>0}$, such that $r_{i+1} = 4e^{K_x \tau} \cdot r_i$ and $r_{i+1} \leq \delta$.*

Suppose that the propagation step begins at state $x'_i \in \mathcal{B}_{r_i}(x_i)$ and ends in x'_{i+1} . Then for any $\kappa \in (0, 1], \epsilon_i \in (0, \kappa r_{i+1})$, we have that:

$$\rho_i := \Pr[x'_{i+1} \in \mathcal{B}_{\kappa r_{i+1}}(x_{i+1})] \geq p_t \cdot \frac{\zeta_D \cdot \max\left(\frac{(4\kappa-1)e^{K_x \tau} r_i - \epsilon_i}{K_u \tau e^{K_x \tau}}, 0\right)}{|\mathbb{U}|},$$

where ζ_D is the Lebesgue measure of the unit ball in \mathbb{R}^D and $0 < p_t \leq 1$ is some constant.

Proof. Consider a sequence of balls of radius $r' = \kappa r_{i+1} - \epsilon_i$, such that (i) the center c_t of each ball lies on π , that is, $c_t = \pi(t)$ for some duration $t \in [0, \tau]$, and (ii) $\mathcal{B}_{r'}(c_t) \subset \mathcal{B}_{\kappa r_{i+1}}(x_{i+1})$. The centers of all such balls constitute a segment of the trajectory π whose duration is T_κ . See Figure 3 for an illustration.

Fix $t \in [0, \tau]$, such that $\mathcal{B}_{r'}(c_t) \subset \mathcal{B}_{\kappa r_{i+1}}(x_{i+1})$. Additionally denote by u_{rand} the random control generated by RRT, and denote by π_t the trajectory corresponding to the propagation step starting at x'_i , using the control u_{rand} and duration t . By Lemma 2, we have that:

$$\|\pi(t) - \pi_t(t)\| < e^{K_x t} r_i + K_u t e^{K_x t} \Delta u,$$

where $\Delta u = \|u - u_{\text{rand}}\|$. Now, we wish to find the value Δu such that $\|\pi(t) - \pi_t(t)\| < \kappa r_{i+1} - \epsilon_i$, which would imply that $\pi_t(t) = x'_{i+1} \in \mathcal{B}_{\kappa r_{i+1}}(x_{i+1})$. Thus, we require that

$$e^{K_x t} r_i + K_u t e^{K_x t} \Delta u < \kappa r_{i+1} - \epsilon_i.$$

As $r_{i+1} = 4e^{K_x \tau} \cdot r_i$ the above constraint yields the condition

$$e^{K_x t} r_i + K_u t e^{K_x t} \Delta u < \kappa \cdot 4e^{K_x \tau} r_i - \epsilon_i$$

which implies that

$$\Delta u < \frac{(4\kappa e^{K_x \tau} - e^{K_x t}) r_i - \epsilon_i}{K_u t e^{K_x t}}.$$

To ensure that the bound holds for all possible durations t in the relevant range, we should consider $t = \tau$, which is the maximal duration there, as the above expression is decreasing with t . That is, we enforce the following bound

$$\Delta u < \frac{(4\kappa e^{K_x \tau} - e^{K_x \tau}) r_i - \epsilon_i}{K_u \tau e^{K_x \tau}} = \frac{(4\kappa - 1)e^{K_x \tau} r_i - \epsilon_i}{K_u \tau e^{K_x \tau}}.$$

Next, we give a lower bound on the probability of a successful forward propagation step of RRT (Algorithm 2), from a given tree node, using a random control $u \in \mathbb{U}$ and a random duration $t \in T_{\text{prop}}$. We note that our proof uses a construction similar to [36, proof of Theorem 17].

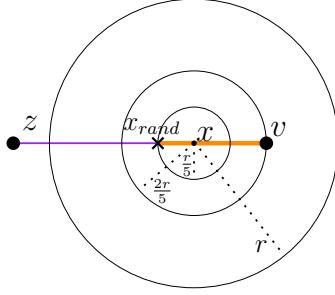


Fig. 4. Illustration of the proof of Lemma 4. z, v are RRT vertices. x_{rand} is the sampled state. Its nearest neighbor will be a vertex in $\mathcal{B}_r(x)$.

To summarize, we have shown that for certain values of t and u_{rand} it is guaranteed to have $x'_{i+1} \in \mathcal{B}_{\kappa r_{i+1}}(x_{i+1})$. It remains to calculate the probability of randomly choosing such values. The probability for successful propagation is at least the (a) probability of choosing a proper t such that $\pi(t)$ is a center c_t of a small ball $\mathcal{B}_{r'}(c_t) \subset \mathcal{B}_{\kappa r_{i+1}}(x_{i+1})$ times the (b) probability for choosing a control input that will cause $\pi_t(t)$ to fall inside $\mathcal{B}_{r'}(c_t) \subset \mathcal{B}_{\kappa r_{i+1}}(x_{i+1})$.

Clearly, the probability to choose a proper duration for propagation is at least $p_t = T_\kappa / T_{\text{prop}} > 0$. The probability¹ to choose a proper control input is at least:

$$p_u = \frac{\zeta_D \cdot \max(\frac{(4\kappa-1)e^{K_x \tau} r_i - \epsilon_i}{K_u \tau e^{K_x \tau}}, 0)}{|\mathcal{U}|}.$$

Therefore, the probability for successfully propagating is at least $\rho_i = p_t \cdot p_u$. \square

Finally, we prove a lower bound on the probability to grow the tree from a vertex in a certain ball.

Lemma 4. *Let $x \in \mathbb{R}^d$ be such that $\mathcal{B}_r(x) \subset \mathcal{F}$. Suppose that there exists an RRT vertex $v \in \mathcal{B}_{2r/5}(x)$. Let x_{near} denote the nearest neighbor of x_{rand} among all RRT vertices (see Algorithm 2). The probability that $x_{\text{near}} \in \mathcal{B}_r(x)$ is at least $|\mathcal{B}_{r/5}|/|\mathcal{X}|$.*

Proof. Suppose that there exists an RRT vertex $z \notin \mathcal{B}_r(x)$, as otherwise it is immediate that $x_{\text{near}} \in \mathcal{B}_r(x)$. We show that if $x_{\text{rand}} \in \mathcal{B}_{r/5}(x)$ then $x_{\text{near}} \in \mathcal{B}_r(x)$. See Figure 4 for an illustration of the proof.

Observe that $\|x_{\text{rand}} - v\| \leq 3r/5$ and $\|x_{\text{rand}} - z\| > 4r/5$. Thus, v is closer to x_{rand} than z is, implying that z will not be reported as the nearest neighbor of x_{rand} . If $x_{\text{near}} \neq v$, then there must be another RRT vertex $y \in \mathcal{B}_{3r/5}(x_{\text{rand}}) \subset \mathcal{B}_r(x)$ such that $\|y - x_{\text{rand}}\|$ is minimal. Finally, the probability to choose $x_{\text{rand}} \in \mathcal{B}_{r/5}(x)$ is $|\mathcal{B}_{r/5}|/|\mathcal{X}|$. \square

Now we are ready to prove our main theorem.

Theorem 2. *Suppose that there exists a valid trajectory π from x_{init} to x_{goal} lying in \mathcal{F} , with clearance $\delta_{\text{clear}} > 0$. Suppose that the trajectory π has a piecewise constant control function. Then the probability that RRT fails to reach x_{goal} from x_{init}*

¹The maxima function guarantees that the probability will be valid, that is, at least 0.

after k iterations is at most $a'e^{-b'k}$, for some constants $a', b' \in \mathbb{R}_{>0}$.

Proof. Let $\tau \leq T_{\text{prop}}$ be a fixed duration for which there exists $\ell \in \mathbb{N}_{>0}$ such that $\ell \cdot \tau = \Delta t$.

We choose a set of times $t_0 = 0, t_1, t_2, \dots, t_m = t_\pi$, such that the difference between every two consecutive ones is τ , where t_π is the duration of π . Let $x_0 = \pi(t_0), x_1 = \pi(t_1), \dots, x_m = \pi(t_m)$ be states along the path π that are obtained after duration t_0, t_1, \dots, t_m , respectively. That is, $x_i = \pi(t_i)$. Obviously, $m = t_\pi/\tau$ is some constant independent of the number of samples.

We now place a set of $m+1$ balls centered at x_0, \dots, x_m such that the radius of the i th ball is $r_i = (4e^{K_x \tau})^i \cdot r_0$ for $0 \leq i \leq m$. Requiring that $r_m = \min\{\delta_{\text{goal}}, \delta_{\text{clear}}\}$, we obtain a value for the smallest radius r_0 . We show that given that an RRT vertex in the i th ball exists, the probability p_i that in the next iteration RRT will generate a new vertex in the $(i+1)$ st ball when propagating from a vertex in the i th ball is bounded from below by a positive constant. More accurately, we show that $p_i \geq p_0$, where p_0 is the probability that RRT will generate a new vertex in $\mathcal{B}_{r_1}(x_1)$ when propagating from $x_0 = x_{\text{init}}$ and it is positive. The rest of the proof is the same as that of Theorem 1.

Recall that Lemma 3 shows a lower bound ρ_i on the probability of a successful propagation between two consecutive balls of radii r_i, r_{i+1} placed in $x_i = \pi(t_i), x_{i+1} = \pi(t_{i+1})$, respectively, such that $t_{i+1} - t_i = \tau$. Assign κ from Lemma 3 the value $2/5$ and fix $\epsilon_i = \kappa r_0 = 2r_0/5$ for all $0 \leq i \leq m$ (note that $\epsilon_i \in (0, \kappa r_i)$, as required). Then $\rho_i > 0$ for a duration τ if

$$\left(4 \cdot \frac{2}{5} - 1\right) e^{K_x \tau} r_i - \epsilon_i = \frac{3}{5} e^{K_x \tau} r_i - \epsilon_i \quad (3)$$

$$= \frac{3}{5} e^{K_x \tau} r_i - \frac{2r_0}{5} > 0. \quad (4)$$

If the above expression is satisfied for $i = 0$ then it also must hold for $1 \leq i \leq m$ as $r_i > r_0$. Since $e^{K_x \tau} \geq 1$ for any $\tau \geq 0$ it must follow that

$$\frac{3}{5} e^{K_x \tau} r_0 - \frac{2r_0}{5} \geq \frac{3}{5} r_0 - \frac{2}{5} r_0 = \frac{r_0}{5} > 0.$$

Moreover, we may set $\tau \leq T_{\text{prop}}$ such that there exists $\ell \in \mathbb{N}_{>0}$ for which $\ell \cdot \tau = \Delta t$ holds.

Suppose that there exists an RRT vertex $v \in \mathcal{B}_{2r_i/5}(x_i) \subset \mathcal{B}_{r_i}(x_i)$. We need to bound the probability p_i that in the next iteration the RRT tree will grow from an RRT vertex in $\mathcal{B}_{r_i}(x_i)$, given that an RRT vertex in $\mathcal{B}_{2r_i/5}(x_i)$ exists, and that the propagation step will add a vertex to $\mathcal{B}_{2r_{i+1}/5}(x_{i+1})$. That is, p_i is the probability that in the next iteration both $x_{\text{near}} \in \mathcal{B}_{r_i}(x_i)$ and $x_{\text{new}} \in \mathcal{B}_{2r_{i+1}/5}(x_{i+1})$.

From Lemma 4, we have that the probability q_i that x_{near} lies in $\mathcal{B}_{r_i}(x_i)$, given that there exists an RRT vertex in $\mathcal{B}_{2r_i/5}(x_i)$, is at least $|\mathcal{B}_{r_i/5}|/|\mathcal{X}|$. Now, since $r_i \geq r_0$ for $0 \leq i \leq m-1$, we have that $q_i \geq q_0 > 0$. From Lemma 3 we have that the probability for $x_{\text{new}} \in \mathcal{B}_{2r_{i+1}/5}(x_{i+1})$ is at least some positive constant $\rho_i > 0$. Moreover, it holds that $\rho_i \geq \rho_0$ for $0 \leq i \leq m-1$. Hence, for all $0 \leq i \leq m-1$ it holds that $p_i \geq p_0$, where $p_0 = q_0 \cdot \rho_0 > 0$. The rest of the proof is the same as that of Theorem 1. \square

V. DISCUSSION

Although our proofs assume uniform samples, they can be easily extended to samples generated using a Poisson point process, which is preferable in certain settings [11], [27]. An immediate extension of this work is to verify whether our proofs hold when other sampling distributions are considered, e.g., Halton sequences (see [40]).

Another possible direction is to further relax some of the assumptions made for kinodynamic systems, such as Lipschitz continuity. Additionally, the work raises the following challenging research question: Is it possible to extend these proofs that have a reduced set of assumptions to other sampling-based planners [14], or informed variants of RRT.

Finally, we mention that the following variants of RRT are not addressed in the current paper, or in the work of Kunz and Stilman [12]: (i) random time + best-control input; (ii) fixed time + random control; (iii) random time larger than a fixed threshold + random or best control. Whether these variants are indeed probabilistically complete remains as a question for future research.

ACKNOWLEDGEMENTS

The authors thank Albert Wu and Thomas Lew for spotting an error in a previous version of the proof of Theorem 2.

REFERENCES

- [1] 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 Robotics and Automation Letters*, vol. 4, no. 2, pp. x–xvi, 2019.
- [2] S. M. LaValle and J. J. Kuffner, “Randomized kinodynamic planning,” *I. J. Robotics Res.*, vol. 20, no. 5, pp. 378–400, 2001.
- [3] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, “Probabilistic roadmaps for path planning in high dimensional configuration spaces,” *IEEE Trans. Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
- [4] N. Vahrenkamp, M. Do, T. Asfour, and R. Dillmann, “Integrated grasp and motion planning,” in *ICRA*, 2010, pp. 2883–2888.
- [5] L. Jaillet, J. Cortés, and T. Siméon, “Sampling-based path planning on configuration-space costmaps,” *IEEE Trans. Robotics*, vol. 26, no. 4, pp. 635–646, 2010.
- [6] A. Yershova, L. Jaillet, T. Siméon, and S. M. LaValle, “Dynamic-domain RRTs: Efficient exploration by controlling the sampling domain,” in *ICRA*, 2005, pp. 3856–3861.
- [7] M. Zucker, J. J. Kuffner, and M. S. Branicky, “Multipartite RRTs for rapid replanning in dynamic environments,” in *ICRA*, 2007, pp. 1603–1609.
- [8] W. Wang, Y. Li, X. Xu, and S. X. Yang, “An adaptive roadmap guided multi-RRTs strategy for single query path planning,” in *ICRA*, 2010, pp. 2871–2876.
- [9] J. J. Kuffner and S. M. LaValle, “RRT-Connect: An efficient approach to single-query path planning,” in *ICRA*, 2000, pp. 995–1001.
- [10] O. Nechushtan, B. Raveh, and D. Halperin, “Sampling-diagram automata: A tool for analyzing path quality in tree planners,” in *WAFA*, 2010, pp. 285–301.
- [11] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *I. J. Robotics Res.*, vol. 30, no. 7, pp. 846–894, 2011.
- [12] T. Kunz and M. Stilman, “Kinodynamic RRTs with fixed time step and best-input extension are not probabilistically complete,” in *WAFA*, 2014, pp. 233–244.
- [13] S. Caron, Q. Pham, and Y. Nakamura, “Completeness of randomized kinodynamic planners with state-based steering,” *Robotics and Autonomous Systems*, vol. 89, pp. 85–94, 2017.
- [14] D. Hsu, J.-C. Latombe, and R. Motwani, “Path planning in expansive configuration spaces,” in *ICRA*, vol. 3, 1997, pp. 2719–2726.
- [15] M. Elbanhawi and M. Simic, “Sampling-based robot motion planning: A review,” *IEEE Access*, vol. 2, pp. 56–77, 2014.
- [16] D. Halperin, L. Kavraki, and K. Solovey, “Robotics,” in *Handbook of Discrete and Computational Geometry*, 3rd ed., J. E. Goodman, J. O’Rourke, and C. D. Tóth, Eds. CRC press, 2018, ch. 51.
- [17] J. D. Gammell, T. D. Barfoot, and S. S. Srinivasa, “Informed sampling for asymptotically optimal path planning,” *IEEE Trans. Robotics*, vol. 34, no. 4, pp. 966–984, 2018.
- [18] O. Salzman and D. Halperin, “Asymptotically near-optimal RRT for fast, high-quality motion planning,” *IEEE Trans. Robotics*, vol. 32, no. 3, pp. 473–483, June 2016.
- [19] O. Arslan and P. Tsiotras, “Use of relaxation methods in sampling-based algorithms for optimal motion planning,” in *ICRA*, 2013, pp. 2421–2428.
- [20] K. Naderi, J. Rajamäki, and P. Hämäläinen, “RT-RRT*: A real-time path planning algorithm based on RRT*,” in *Conference on Motion in Games*, 2015, pp. 113–118.
- [21] M. W. Otte and E. Frazzoli, “RRT^X: Asymptotically optimal single-query sampling-based motion planning with quick replanning,” *I. J. Robotics Res.*, vol. 35, no. 7, pp. 797–822, 2016.
- [22] D. Devaurs, T. Siméon, and J. Cortés, “Optimal path planning in complex cost spaces with sampling-based algorithms,” *IEEE Trans. Automation Science and Engineering*, vol. 13, no. 2, pp. 415–424, 2016.
- [23] 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,” *I. J. Robotics Res.*, vol. 34, no. 7, pp. 883–921, 2015.
- [24] A. Mandalika, O. Salzman, and S. Srinivasa, “Lazy receding horizon A* for efficient path planning in graphs with expensive-to-evaluate edges,” in *ICAPS*, 2018, pp. 476–484.
- [25] K. Solovey and D. Halperin, “Efficient sampling-based bottleneck pathfinding over cost maps,” in *IROS*, 2017, pp. 2003–2009.
- [26] 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.
- [27] K. Solovey and M. Kleinbort, “The critical radius in sampling-based motion planning,” in *RSS*, 2018.
- [28] E. Schmerling, L. Janson, and M. Pavone, “Optimal sampling-based motion planning under differential constraints: The driftless case,” in *ICRA*, 2015, pp. 2368–2375.
- [29] ———, “Optimal sampling-based motion planning under differential constraints: The drift case with linear affine dynamics,” in *CDC*, 2015, pp. 2574–2581.
- [30] S. Karaman and E. Frazzoli, “Optimal kinodynamic motion planning using incremental sampling-based methods,” in *CDC*, 2010, pp. 7681–7687.
- [31] ———, “Sampling-based optimal motion planning for non-holonomic dynamical systems,” in *ICRA*, 2013, pp. 5041–5047.
- [32] 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.
- [33] 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, pp. 2537–2542.
- [34] 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, pp. 4187–4194.
- [35] G. Goretkin, A. Perez, R. Platt, and G. Konidaris, “Optimal sampling-based planning for linear-quadratic kinodynamic systems,” in *ICRA*, 2013, pp. 2429–2436.
- [36] Y. Li, Z. Littlefield, and K. E. Bekris, “Asymptotically optimal sampling-based kinodynamic planning,” *I. J. Robotics Res.*, vol. 35, no. 5, pp. 528–564, 2016.
- [37] 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.
- [38] T. Kunz and M. Stilman, “Probabilistically complete kinodynamic planning for robot manipulators with acceleration limits,” in *IROS*, 2014, pp. 3713–3719.
- [39] 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.
- [40] L. Janson, B. Ichter, and M. Pavone, “Deterministic sampling-based motion planning: Optimality, complexity, and performance,” *I. J. Robotics Res.*, vol. 37, no. 1, pp. 46–61, 2018.