

TIE: Time-Informed Exploration for Robot Motion Planning

Sagar Suhas Joshi¹ Seth Hutchinson² Panagiotis Tsiotras³

Abstract—Anytime sampling-based methods are an attractive technique for solving kino-dynamic motion planning problems. These algorithms scale well to higher dimensions and can efficiently handle state and control constraints. However, an intelligent exploration strategy is required to accelerate their convergence and avoid redundant computations. Using ideas from reachability analysis, this work defines a “Time-Informed Set (TIS)”, that focuses the search for time-optimal kino-dynamic planning after an initial solution is found. Such a Time-Informed Set includes all trajectories that can potentially improve the current best solution and hence exploration outside this set is redundant. Benchmarking experiments show that an exploration strategy based on the TIS can accelerate the convergence of sampling-based kino-dynamic motion planners.

Index Terms—Motion and Path Planning, Autonomous Agents, Collision Avoidance.

I. INTRODUCTION

SAMPLING-based motion planners incrementally build a connectivity graph by generating random samples in the search-space. Popular algorithms such as RRT [1] can solve challenging problems in higher-dimensional spaces, but can only ensure probabilistic completeness. The RRT* algorithm [2] combines the exploration procedure in RRT with a “local rewiring” module to guarantee asymptotic optimality. Algorithms such as RRT[#] [3], FMT* [4] and BIT* [5] use heuristics along with dynamic programming ideas to achieve faster convergence than RRT*.

The “geometric” versions of the above sampling-based algorithms ignore kino-dynamic constraints of the robot and connect any two points in a Euclidean search space with a straight line. However, a general kino-dynamic problem requires the solution of a two-point boundary value problem (TPBVP), also called the “local steering” problem, for optimally connecting any two states. Karaman and Frazzoli extended the RRT* algorithm for kino-dynamic planning by incorporating such steering functions in [6]. Perez et al [7] linearized the system dynamics and solved the infinite-horizon linear quadratic regulator (LQR) problem to obtain a locally optimal steering procedure. The kino-dynamic RRT* algorithm [8]

Manuscript received: October, 15, 2020; Revised January, 18, 2021; Accepted February, 20, 2021.

This paper was recommended for publication by Editor Nancy Amato upon evaluation of the Associate Editor and Reviewers’ comments.

This work has been supported by NSF awards IIS-1617630 and IIS-2008686. The authors thank Dipankar Maity and Kelsey Hawkins for insightful discussions on this topic.

^{1,2,3} Authors are with Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, USA. Email: {sagarsjoshi94, seth, tsiotras}@gatech.edu

Digital Object Identifier (DOI): see top of this page.

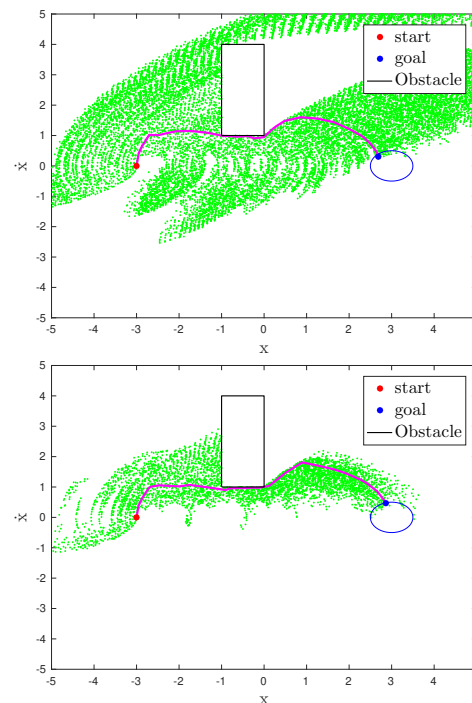


Figure 1: Time-optimal planning for a 2D system using the SST algorithm with uniform exploration (top) and the proposed strategy (bottom). The tree vertices generated are represented in green. Using the proposed strategy leads to a focused search.

penalizes the control effort and the trajectory duration while connecting any two states. The authors of [8] solve a fixed final state, free final time, optimal control problem for linear time invariant (LTI) systems to derive a steering function. A kino-dynamic version of FMT* is presented in [9]. Note that these algorithms rely on the availability of a local steering module to ensure asymptotic optimality. However, developing such computationally efficient TPBVP solvers may not be possible for many cases. The GR-FMT algorithm [10] proposes a local steering method based on polynomial basis functions and segmentation for controllable linear systems. The recently introduced Stable Sparse RRT (SST) and SST* [11] algorithms guarantee asymptotic optimality, while having access only to a forward propagation model of the system’s dynamics. This eliminates the need for TPBVP solvers. The SST procedure promotes the propagation of states with good path costs and performs a selective pruning operation to keep the number of stored nodes small.

While significant progress has been made in the area of

sampling-based kino-dynamic planners, developing intelligent exploration strategies to complement them still remains a challenging problem. Uniform random sampling results in a rapid exploration of the search-space and is effective for finding a first solution. However, after an initial solution is found, exploration can be focused on a subset of the search-space that can potentially further improve the current solution. For the case of geometric, length-optimal planning, Gammell et al [12] introduced the “ L_2 -Informed Set” that contains all the points that can *potentially* improve the current solution. This set is a prolate hyper-spheroid with focii at the start and the goal states and its transverse diameter is equal to the current best solution cost. The direct Informed Sampling (IS) technique proposed in [12] provides a scalable approach to focus search, and shows dramatic convergence improvements in higher dimensions compared to the other state-of-the-art heuristic methods.

However, as discussed in [13], [14] deriving a parameterized representation or direct sampling of such Informed Sets for systems with differential constraints is a challenging problem. In this work, we propose an analogue to the Informed Set for the case of *time-optimal* kino-dynamic planning using ideas from reachability analysis [15], [16].

Given a feasible (but perhaps sub-optimal) solution trajectory with time cost $T > 0$, we define a Time-Informed Set (TIS) as the set that contains all the trajectories with time cost less than or equal to T . The planner can thus avoid redundant exploration outside the TIS. The proposed exploration algorithm can be applied to a variety of systems, even if a tractable TPBVP solver may not be available.

II. RELATED WORK

Prior work on intelligent exploration, such as [17], [12], [18], [19] utilized heuristics and ideas from deep learning to improve the performance of sampling-based planners. The Informed SST (iSST) algorithm [20] also leverages heuristics to guide search for kino-dynamic planning. DIRT [21] uses dominance informed regions along with heuristics to balance exploration and exploitation. However, iSST and DIRT may be ineffective in focusing the search for the cases where a good heuristic function is unavailable.

Concepts from reachability analysis have also been used for guiding exploration in sampling-based kino-dynamic planning. Shkolnik et al [22] used reachable sets in their RG-RRT algorithm to shape the Voronoi bias so as to find a feasible solution quickly. A discretized representation of the reachable space is proposed in [23] to be used for sampling and nearest neighbor search. Chiang et al [24] trained an obstacle-aware time-to-reach (TTR) reachability estimator network to guide the RRT search process. However, the above techniques do not focus search on a subset of the search space based on current solution cost, which can lead to redundant exploration.

The algorithms proposed in [13] and [14] are most relevant to the current work, as they address the problem of Informed Sampling for kino-dynamic motion planning. Kunz et al [13] proposed a hierarchical rejection sampling (HRS) method to generate informed samples for higher-dimensional systems.

HRS essentially is a “bottom up” procedure that generates samples along the individual dimensions and combines them. An accept/reject decision is taken for each partial sample until a complete sample in the informed set is generated. Yi et al [14] proposed a Hit-and-Run Markov Chain Monte-Carlo (HNR-MCMC) algorithm to improve the sampling efficiency compared to HRS. Given a previous sample in the Informed Set, the HNR-MCMC first samples a random direction and then uses rejection sampling to find the largest step-size so that the new sample lies inside the Informed Set. However, both HRS and HNR-MCMC assume availability of a local steering function, that gives the optimal cost (or a good under-estimate) connecting any two states. For minimum time problems, the above two methods can only be applied to specific systems, such as the double integrator. In this work, we address this issue by using ideas from reachability analysis to define the TIS. The proposed algorithm can thus be applied to a wide variety of systems.

In the following sections, the time-optimal kino-dynamic motion planning problem is first defined, followed by the definition of the TIS and some theoretical results. The proposed exploration algorithm is then delineated along with some results from a series of numerical experiments.

III. PROBLEM DEFINITION

Let $\mathcal{X} \subset \mathbb{R}^n$, $n \geq 2$ and $\mathcal{U} \subset \mathbb{R}^m$, $m \geq 1$ be compact sets representing the state and admissible control spaces respectively. Let $\mathcal{X}_{\text{obs}} \subset \mathcal{X}$ denote the obstacle space and $\mathcal{X}_{\text{free}} = \text{cl}(\mathcal{X} \setminus \mathcal{X}_{\text{obs}})$ denote the free space. Here, $\text{cl}(S)$ represents the closure of the set $S \subset \mathbb{R}^n$. Let $\lambda(S)$ denote the Lebesgue measure of the set $S \subset \mathbb{R}^n$. Let $\mathbf{x}_s \in \mathcal{X}_{\text{free}}$ denote the initial state and let $\mathcal{X}_g \subset \mathcal{X}_{\text{free}}$ represent the goal set. The time-optimal motion planning problem can be defined as follows:

$$T^* = \min_{\mathbf{u}} T \quad (1a)$$

$$\text{subject to: } \dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)), \quad (1b)$$

$$\mathbf{x}(0) = \mathbf{x}_s, \mathbf{x}(T) \in \mathcal{X}_g, \quad (1c)$$

$$\mathbf{x}(t) \in \mathcal{X}_{\text{free}}, \mathbf{u}(t) \in \mathcal{U} \quad \text{for all } t \in [0, T]. \quad (1d)$$

Sampling-based algorithms solve the above problem by incrementally building a tree $\mathcal{T} = (V, E)$ that encodes the connectivity between a finite set of vertices $V \subset \mathcal{X}_{\text{free}}$ with edges $E \subseteq V \times V$. The trajectory and the cost representing an edge are calculated either by using a steering function or by forward propagation of the system model using random controls.

IV. TIME-INFORMED SET

Consider the set of points that can be reached at time t , starting from $\mathbf{x}_s$ at time $t_0 < t$, using admissible controls,

$$\mathcal{X}_f[t_0, t] = \{\mathbf{z} \in \mathcal{X} \mid \exists \mathbf{u} : [t_0, t] \rightarrow \mathcal{U}, \mathbf{x} : [t_0, t] \rightarrow \mathcal{X}, \text{ s.t } \mathbf{x}(t_0) = \mathbf{x}_s, \mathbf{x}(t) = \mathbf{z}, \dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t))\}. \quad (2)$$

Let $\mathcal{F}[t_0, t]$ be an over-approximation of $\mathcal{X}_f[t_0, t]$, i.e., $\mathcal{X}_f[t_0, t] \subseteq \mathcal{F}[t_0, t]$. Similarly, the set of points starting at time

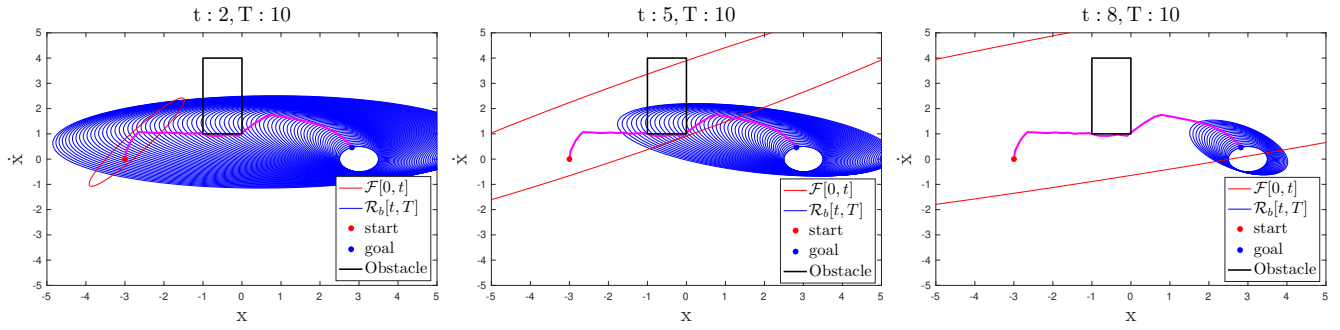


Figure 2: Evolution of the forward reachable set $\mathcal{F}[0, t]$ and the backward reachable tube $\mathcal{R}_b[t, T]$ for the 2D Toy system at time $t = 2, 5, 8$. Note that $\Omega(T)$ comprises of the intersections $\mathcal{F}[0, t] \cap \mathcal{R}_b[t, T]$.

t that can reach $\mathcal{X}_g$ at time $t_f > t$ using admissible controls can be defined as,

$$\mathcal{X}_b[t, t_f] = \{\mathbf{z} \in \mathcal{X} \mid \exists \mathbf{u} : [t, t_f] \rightarrow \mathcal{U}, \mathbf{x} : [t, t_f] \rightarrow \mathcal{X}, \text{ s.t. } \mathbf{x}(t) = \mathbf{z}, \mathbf{x}(t_f) \in \mathcal{X}_g, \dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t))\}. \quad (3)$$

Let $\mathcal{B}[t, t_f]$ be an over-approximation of $\mathcal{X}_b[t, t_f]$, i.e., $\mathcal{X}_b[t, t_f] \subseteq \mathcal{B}[t, t_f]$. Note that state constraints ensuring collision-free trajectories are not imposed while defining the above sets. The (over-approximated) backward reachability tube over the interval $[t, t_f]$ includes the set of all points starting at time t , that can reach $\mathcal{X}_g$ at any time $\tau \in [t, t_f]$

$$\mathcal{R}_b[t, t_f] = \bigcup_{t \leq \tau \leq t_f} \mathcal{B}[t, \tau]. \quad (4)$$

Assume that a feasible (perhaps sub-optimal) solution to problem (1) with time cost $T > 0$ is available. Consider the following definition of the Time-Informed Set (TIS)

$$\Omega(T) = \bigcup_{0 \leq t \leq T} \mathcal{F}[0, t] \cap \mathcal{R}_b[t, T]. \quad (5)$$

Intuitively, $\Omega(T)$ contains all the points $\mathbf{x} \in \mathcal{X}$ that can be reached from $\mathbf{x}_s$ at a time t , where $0 \leq t \leq T$, i.e., $\mathbf{x} \in \mathcal{F}[0, t]$ and then can reach the goal at time τ , $t \leq \tau \leq T$, i.e., $\mathbf{x} \in \mathcal{R}_b[t, T]$. Please see Fig. 2 and the attached video¹ for a visualization of $\Omega(T)$.

The following theoretical arguments formally prove that given a sub-optimal solution with time cost T , the set $\Omega(T)$ contains all the trajectories with time cost T or less.

Lemma 1. *Given a feasible solution with cost $T > 0$, $\mathcal{F}[0, t] \cap \mathcal{B}[t, T] \neq \emptyset$ for all $t \in [0, T]$.*

Proof. Consider the solution trajectory with time cost T , $\zeta : [0, T] \rightarrow \mathcal{X}$, where $\zeta(0) = \mathbf{x}_s$ and $\zeta(T) = \mathbf{x}_g$. For any point $\mathbf{x}$ on this trajectory, there exists $t \in [0, T]$ such that $\mathbf{x} = \zeta(t)$. Thus, $\mathbf{x} \in \mathcal{F}[0, t]$ and $\mathbf{x} \in \mathcal{B}[t, T]$. It follows that, $\mathbf{x} \in \mathcal{F}[0, t] \cap \mathcal{B}[t, T]$. Therefore, $\mathcal{F}[0, t] \cap \mathcal{B}[t, T] \neq \emptyset$. $\square$

Lemma 2. $\mathcal{R}_b[t, T_1] \subset \mathcal{R}_b[t, T_2]$ for any $T_2 > T_1 > t > 0$.

Proof. Note from the definition (4),

$$\begin{aligned} \mathcal{R}_b[t, T_2] &= \bigcup_{t \leq \tau \leq T_2} \mathcal{B}[t, \tau] \\ &= \left(\bigcup_{t \leq \tau \leq T_1} \mathcal{B}[t, \tau] \right) \cup \left(\bigcup_{T_1 \leq \tau \leq T_2} \mathcal{B}[t, \tau] \right). \end{aligned}$$

Since $\mathcal{R}_b[t, T_1] = \bigcup_{t \leq \tau \leq T_1} \mathcal{B}[t, \tau]$ it follows that $\mathcal{R}_b[t, T_1] \subset \mathcal{R}_b[t, T_2]$. $\square$

Theorem 3. *The set $\Omega(T)$ contains all trajectories with time cost exactly T .*

Proof. Consider any solution trajectory $\zeta : [0, T] \rightarrow \mathcal{X}$ with time cost $T > 0$, where $\zeta(0) = \mathbf{x}_s$, $\zeta(T) = \mathbf{x}_g$. For any point $\mathbf{x}$ on this trajectory, there exists $t \in [0, T]$ such that $\mathbf{x} = \zeta(t)$. Then, $\mathbf{x} \in \mathcal{F}[0, t]$ and $\mathbf{x} \in \mathcal{B}[t, T]$. This implies that $\mathbf{x} \in \mathcal{F}[0, t] \cap \mathcal{B}[t, T]$ and hence $\mathbf{x} \in \mathcal{F}[0, t] \cap \mathcal{R}_b[t, T]$. Thus, $\mathbf{x} \in \Omega(T)$. Since t is arbitrary, it follows that $\zeta(t) \in \Omega(T)$ for all $t \in [0, T]$. $\square$

Theorem 4. $\Omega(T_1) \subset \Omega(T_2)$ for any $T_2 > T_1 > 0$.

Proof. Recall that the set $\Omega(T_2)$ is defined by

$$\Omega(T_2) = \bigcup_{0 \leq t \leq T_2} \mathcal{F}[0, t] \cap \mathcal{R}_b[t, T_2], \quad (6)$$

which can be re-written as

$$\Omega(T_2) = \left(\bigcup_{0 \leq t \leq T_1} \mathcal{F}[0, t] \cap \mathcal{R}_b[t, T_2] \right) \cup \left(\bigcup_{T_1 \leq t \leq T_2} \mathcal{F}[0, t] \cap \mathcal{R}_b[t, T_2] \right).$$

From Lemma 2, it follows that $\mathcal{R}_b[t, T_1] \subset \mathcal{R}_b[t, T_2]$. Hence, $\Omega(T_1) = \bigcup_{0 \leq t \leq T_1} \mathcal{F}[0, t] \cap \mathcal{R}_b[t, T_1] \subset \bigcup_{0 \leq t \leq T_1} \mathcal{F}[0, t] \cap \mathcal{R}_b[t, T_2]$. Thus, $\Omega(T_1) \subset \Omega(T_2)$. $\square$

Corollary 5. *Given a solution to (1) with time cost T , the set $\Omega(T)$ defined in (5) contains all the trajectories with cost less than or equal to T . Conversely, any trajectory that is not contained inside $\Omega(T)$ has time cost $T' > T$*

Proof. From Theorem 3, it follows that $\Omega(T)$ contains all trajectories with time cost exactly T . Theorem 4 implies that $\Omega(T)$ is a superset of all the sets containing trajectories

¹<https://www.youtube.com/watch?v=dnMHb7uFEGw>

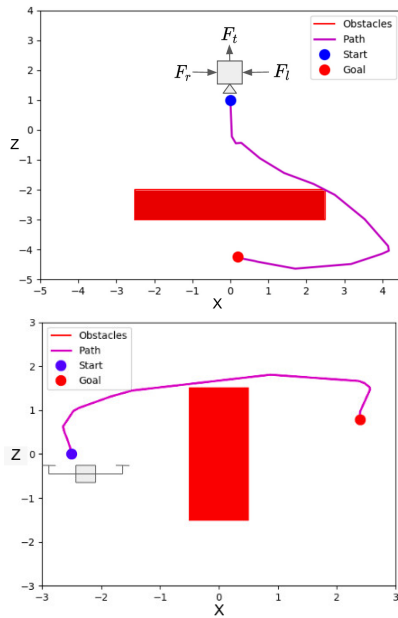


Figure 3: A schematic for the moon-lander robot (top) and quadrotor (bottom) simulation cases with sample solution paths found by the proposed algorithm after 40 sec of planning time.

with time cost less than T . Thus, $\Omega(T)$ also contains all the trajectories with cost less than or equal to T . $\square$

After a, perhaps sub-optimal, solution with cost T is found, any state that lies on an improved solution path necessarily lies inside the TIS. The search can thus be focused onto the TIS. This can avoid redundant computations and accelerate convergence, especially for higher dimensional problems.

V. TIME-INFORMED VS L_2 -INFORMED SET

This section examines the relationship between the TIS defined in (5) and the L_2 -Informed Set from [12] for a special case of a linear single integrator system. The purpose of this investigation is to show that the TIS is a generalization of the L_2 -Informed Set approach in [12]. Consider the case of single-integrator dynamics $\dot{\mathbf{x}}(t) = \mathbf{u}(t)$, for which

$$\begin{aligned} \mathcal{F}[t_0, t] &= \{\mathbf{x} \in \mathcal{X} \mid \|\mathbf{x} - \mathbf{x}_s\|_2 \leq u_{\max}(t - t_0)\} \\ \mathcal{B}[t, t_f] &= \{\mathbf{x} \in \mathcal{X} \mid \|\mathbf{x} - \mathbf{x}_g\|_2 \leq u_{\max}(t_f - t)\}. \end{aligned} \quad (7)$$

Here, $\|\cdot\|_2$ represents the L_2 -norm. As the set $\mathcal{U}$ is compact, there exists a $u_{\max} > 0$, so that $\|\mathbf{u}(t)\|_2 \leq u_{\max}$ for all t . Note that, for this special case, the forward and backward reachable sets defined in (7) are concentric circles. Then, for a given $t < t_f$, we have $\mathcal{B}[t, t_f] = \mathcal{R}_b[t, t_f]$ and hence $\Omega(T) = \bigcup_{0 \leq t \leq T} \mathcal{F}[0, t] \cap \mathcal{B}[t, T]$. Thus, for any $\mathbf{x} \in \Omega(T)$, we have $\|\mathbf{x} - \mathbf{x}_s\|_2 \leq u_{\max}t$ and $\|\mathbf{x} - \mathbf{x}_g\|_2 \leq u_{\max}(T - t)$. Adding the two inequalities we get,

$$\Omega(T) = \{\mathbf{x} \in \mathcal{X} \mid \|\mathbf{x} - \mathbf{x}_s\|_2 + \|\mathbf{x} - \mathbf{x}_g\|_2 \leq u_{\max}T\} \quad (8)$$

The TIS in (8) in this case has the same prolate hyper-spheroid form as the L_2 -Informed Set [12]. Thus, the TIS can be seen as a generalization of the L_2 -Informed Set.

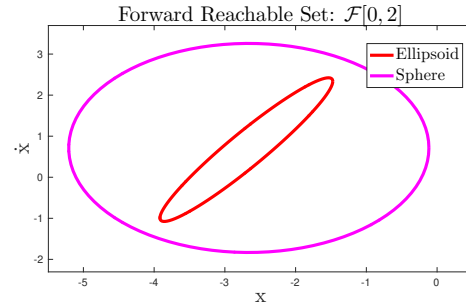


Figure 4: Comparing the forward reachable set $\mathcal{F}[0, t]$ for the 2D system at $t = 2$ using the hyper-sphere and ellipsoidal approximation.

VI. TIME-INFORMED EXPLORATION

Although obtaining the exact reachable sets defined in (2), (3) may not be computationally tractable, various techniques have been proposed to obtain tight over-approximations of these sets. These include application of polytopes and zonotopes [25], ellipsoidal calculus [16] and formulating reachability problem as a Hamilton-Jacobi-Bellman (HJB) PDE [15]. In this work, we use the ellipsoidal technique which provides a scalable framework for reachability analysis of robots with linear-affine dynamics. However, as discussed later on, the HJB reachability formulation can be used to extend the algorithms proposed in this work for general cost-functions and non-linear systems.

Consider the special case of linear kino-dynamic systems. Concretely, the constraint (1b) is $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$, with $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times m}$. Then, $\mathcal{X}_f[t_0, t]$ and $\mathcal{X}_b[t, t_f]$ can be defined as

$$\begin{aligned} \mathcal{X}_f[t_0, t] &= \{\mathbf{x} \in \mathcal{X} \mid \exists \mathbf{u} : [t_0, t] \rightarrow \mathcal{U}, \text{ s.t.} \\ &\quad \mathbf{x} = e^{\mathbf{A}(t-t_0)}\mathbf{x}_s + \int_{t_0}^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau) d\tau\}, \\ \mathcal{X}_b[t, t_f] &= \{\mathbf{x} \in \mathcal{X} \mid \exists \mathbf{u} : [t, t_f] \rightarrow \mathcal{U}, \text{ s.t.} \\ &\quad \mathbf{x} = e^{-\mathbf{A}(t_f-t)}\mathbf{x}_g - \int_t^{t_f} e^{-\mathbf{A}(t_f-\tau)}\mathbf{B}\mathbf{u}(\tau) d\tau\}. \end{aligned} \quad (9)$$

Here, $\mathbf{x}_g \in \mathcal{X}_{\text{goal}}$. A hyper-sphere over-approximation to the above sets can be constructed as follows [25],

$$\begin{aligned} \mathcal{F}[t_0, t] &= \{\mathbf{x} \in \mathcal{X} \mid \|\mathbf{x} - e^{\mathbf{A}(t-t_0)}\mathbf{x}_s\|_2 \leq r(t_0, t, u_{\max})\}, \\ \mathcal{B}[t, t_f] &= \{\mathbf{x} \in \mathcal{X} \mid \|\mathbf{x} - e^{-\mathbf{A}(t_f-t)}\mathbf{x}_g\|_2 \leq r(t, t_f, u_{\max})\}, \\ r(t_1, t_2, u_{\max}) &= (e^{\|\mathbf{A}\|_2(t_2-t_1)} - 1)\|\mathbf{B}\|u_{\max}/\|\mathbf{A}\|_2. \end{aligned} \quad (10)$$

Here, $\|M\|_2$ represents the induced two norm (maximum singular value) for a matrix M . However, the above over-approximation might be too conservative for the current application. See Fig. 4. If the reachable sets are overtly conservative and $\lambda(\Omega(T)) \approx \lambda(\mathcal{X})$, then TIE may result in little or no focus of the search. In contrast, the ellipsoidal technique [16] approximates the reachable sets as ellipsoids,

$$\mathcal{E}(\mathbf{x}_c, Q) = \{\mathbf{x} \in \mathbb{R}^n \mid \langle \mathbf{x} - \mathbf{x}_c, Q^{-1}(\mathbf{x} - \mathbf{x}_c) \rangle \leq 1\}. \quad (11)$$

Here, $\mathbf{x}_c$ is the center and Q is the positive definite shape matrix of the ellipsoid. Forward and backward reachable sets, $\mathcal{F}[0, t]$, $\mathcal{B}[t, T]$ can be obtained by solving an ordinary

Algorithm 1: Sampling Algorithm

```

1 generateSample ( $T$ ):
2    $t \sim p_{[0,T]}(t)$ ;
3   for  $i = 1 : n_s$  do
4     if  $\lambda(\mathcal{F}[0, t]) < \lambda(\mathcal{B}[t, T])$  then
5        $\mathbf{x}_{\text{cand}} \leftarrow \text{sampleUniform}(\mathcal{F}[0, t])$ ;
6       if  $\mathbf{x}_{\text{cand}} \in \mathcal{B}[t, T]$  then
7          $\mathbf{x}_{\text{rand}} \leftarrow \mathbf{x}_{\text{cand}}$ ;
8         return  $\mathbf{x}_{\text{rand}}$ ;
9     else
10       $\mathbf{x}_{\text{cand}} \leftarrow \text{sampleUniform}(\mathcal{B}[t, T])$ ;
11      if  $\mathbf{x}_{\text{cand}} \in \mathcal{F}[0, t]$  then
12         $\mathbf{x}_{\text{rand}} \leftarrow \mathbf{x}_{\text{cand}}$ ;
13        return  $\mathbf{x}_{\text{rand}}$ ;
14   $\mathbf{x}_{\text{rand}} \leftarrow \text{sampleUniform}(\mathcal{X})$ ;
15  return  $\mathbf{x}_{\text{rand}}$ ;

```

Algorithm 2: Vertex Inclusion Algorithm

```

1 includeVertex ( $\mathbf{v}, t, T$ ):
2   if  $t > T$  then
3     return false;
4   foreach  $\tau \in \{t + \delta, t + 2\delta, \dots, T\}$  do
5     if  $\mathbf{v} \in \mathcal{B}[t, \tau]$  then
6       return true;
7   return false;

```

differential equation (ODE) for the center and shape matrix. Please see the Ellipsoidal Toolbox² documentation for a brief overview. Note that the boundary conditions for the forward and backward reachable set ODE are the start and goal ellipsoids respectively. From the problem definition in (1), the start ellipsoid is encoded as a hyper-sphere with negligible radius around the center $\mathbf{x}_s$. The goal set $\mathcal{X}_g$ is represented also as a hyper-sphere with a set radius around a center $\mathbf{x}_g \in \mathcal{X}_g$. The ODE for the shape matrix can be solved and stored off-line. An analytical solution for the ODE describing the center's trajectory can also be constructed. Thus, a “library” of reachable sets $\mathcal{F}[0, t], \mathcal{B}[t, T]$ can be created off-line to be used in the sampling and vertex inclusion algorithm described below. This library stores the value of center vector $\mathbf{x}_c$ and matrices Q, L of the forward and backward reachable sets. Here L is obtained using the Cholesky decomposition of Q , $Q = LL^T$ and is used for generating samples inside $\mathcal{E}(\mathbf{x}_c, Q)$ [12]. Please see Fig. 2 for a visualization of $\mathcal{F}[0, t]$ and $\mathcal{B}[t, T]$ constructed using the ellipsoid technique.

A. Sampling Algorithm

Algorithm 1 describes a procedure to generate a new sample $\mathbf{x}_{\text{rand}}$ in $\Omega(T)$. Notice from (5) that $\Omega(T)$ consists of a union over the intersections of sets. Devising a direct sampling

technique to generate uniform random samples in $\Omega(T)$ (as done for the L_2 Informed Set in [12]) is hence a challenging task. The proposed algorithm proceeds by first sampling a time t in the interval $(0, T)$ according to a probability distribution $p_{[0,T]}(t)$ (line 2). Ideally, to generate uniform random samples in $\Omega(T)$ with respect to the Lebesgue measure, this distribution needs to be $p_{[0,T]}(t) = \lambda(\mathcal{F}[0, t] \cap \mathcal{R}_b[t, T]) / \lambda(\Omega(T))$. However, calculating and sampling from this distribution may not be tractable for general higher dimensional systems. Hence, for the sake of simplicity, we choose $p_{[0,T]}(t)$ to be uniform over the interval $[0, T]$. Given t , the sets $\mathcal{F}[0, t], \mathcal{B}[t, T]$ can then be obtained from the library of stored reachable sets as discussed in the previous section. We leverage the fact that $\mathcal{F}[0, t] \cap \mathcal{B}[t, T] \neq \emptyset$ from Lemma 1 to generate a $\mathbf{x}_{\text{rand}} \in \mathcal{F}[0, t] \cap \mathcal{B}[t, T]$. If the Lebesgue measure of $\mathcal{F}[0, t]$ is less than $\mathcal{B}[t, T]$, a uniform sample is generated in $\mathcal{F}[0, t]$ and checked if it belongs to $\mathcal{B}[t, T]$, otherwise, $\mathcal{B}[t, T]$ is sampled and checked if it belongs to $\mathcal{F}[0, t]$ (lines 4-13). Notice from Fig. 2 that $\lambda(\mathcal{F}[0, t])$ increases and $\lambda(\mathcal{B}[t, T])$ decreases as t varies from 0 to T . An efficient algorithm for generating uniform samples inside a hyper-ellipsoid is discussed in [12]. If no $\mathbf{x}_{\text{rand}} \in \mathcal{F}[0, t] \cap \mathcal{B}[t, T]$ can be generated in n_s attempts, the algorithm returns a uniform random sample from the search-space $\mathcal{X}$ (line 14-15).

B. Vertex Inclusion Algorithm

The vertex inclusion procedure, described in Algorithm 2, accepts a candidate vertex if it lies in $\Omega(T)$. Consider a candidate vertex $\mathbf{v}$ with cost-to-come t , i.e., the cost of trajectory from $\mathbf{x}_s$ to $\mathbf{v}$ is t . Since the cost-to-come is t , we have $\mathbf{v} \in \mathcal{F}[0, t]$. Thus, if $\mathbf{v} \in \mathcal{R}_b[t, T]$, then $\mathbf{v} \in \Omega(T)$. The proposed algorithm discretizes the interval $[t, T]$ with a step-size δ . A vertex is accepted if it lies in any $\mathcal{B}[t, \tau]$, for $\tau \in \{t + \delta, t + 2\delta, \dots, T\}$ (line 4-6). The sets $\mathcal{B}[t, \tau]$ are again obtained from the stored library of reachable sets.

In order to maintain the theoretical guarantees of TIE, an over-estimate of the solution cost T is required. This over-estimate can be obtained (and updated) after the planner discovers (and then improves) an initial, sub-optimal solution. Also, learning-based methods similar to [24] can be used to obtain an estimate of the solution cost given a planning environment. In this work, the above algorithms are called only after an initial solution is discovered.

VII. NUMERICAL EXPERIMENTS

Benchmarking experiments were performed by pairing different exploration strategies with the SST planner [11]. All algorithms were implemented in C++ using the OMPL framework [26], and the tests were run using OMPL's standardized benchmarking tools [27]. Please see <https://github.com/DCSLgitech/tie>. The data was recorded over 100 trials for all the cases on a 64-bit laptop PC with 16 GB RAM and an Intel i7 Processor, running Ubuntu 16.04 OS. The performance of the proposed exploration strategy was benchmarked against uniform sampling (Uni) and uniform sampling combined with Informed propagation (IP). Informed propagation essentially rejects expansion vertices with cost-to-come $t > T$, if there

²http://systemanalysisdpt-cmc-msu.github.io/ellipsoids/doc/main_manual.html

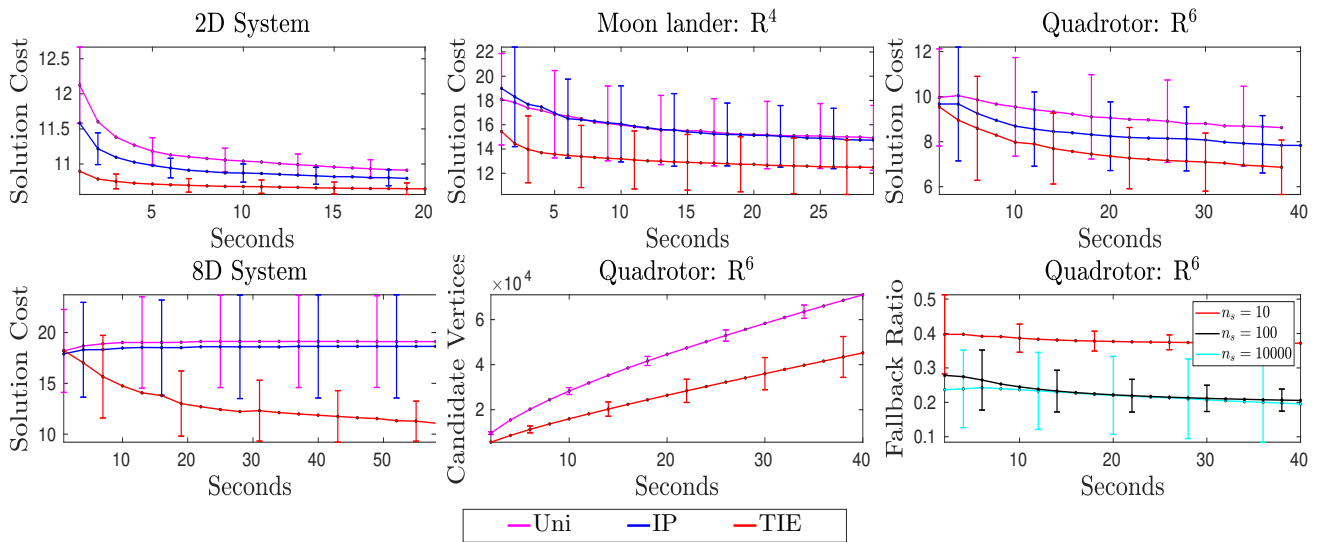


Figure 5: Convergence plots for the numerical experiments. Using the proposed TIE leads to a faster convergence in all cases (red plot). The bottom middle figure illustrates number of candidate vertices generated using uniform and TIE exploration method. The bottom right figure plots the fallback ratio for different values of n_s . Solid lines indicate the value averaged over 100 trials and the error bars represent the standard deviation.

exists a sub-optimal solution with cost T . If $t < T$, then forward propagation from the vertex is done for at most $T - t$ duration. Thus, Informed Propagation (IP) prohibits exploration outside the set $\bigcup_{0 \leq \tau \leq T} \mathcal{F}[0, \tau]$. The proposed Time-Informed exploration (TIE) algorithm uses the sampling and vertex inclusion procedures described in Algorithms 1 and 2 with $n_s = 10$ and $\delta = 0.1$. The SST planner parameters, namely, the selection and pruning radius were set to standard OMPL values of 0.2 and 0.1 respectively. The L_2 -norm was used as the distance function. A description of different case-studies is given below.

2D System: Consider a 2D kino-dynamic system $\dot{\mathbf{x}} = A_{2 \times 2} \mathbf{x} + B_{2 \times 1} u$ with, $\mathbf{x} = [x, \dot{x}]^T$ and

$$A_{2 \times 2} = \begin{bmatrix} 0.0 & 0.5 \\ -0.1 & 0.2 \end{bmatrix}, \quad B_{2 \times 1} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (12)$$

The set-up of the planning problem is illustrated in Fig. 1, with $\mathbf{x}_s = [-3 \ 0]^T$, $\mathbf{x}_g = [3 \ 0]^T$, $\mathcal{X}_g = \mathcal{E}(\mathbf{x}_g, 0.25 \ I_2)$, $u \in [-0.5 \ 0.5]$. Here, I_2 represents the 2×2 identity matrix.

8D System: The 2D system described above is extended to a 8D system $\dot{\mathbf{x}} = A_{8 \times 8} \mathbf{x} + B_{8 \times 4} \mathbf{u}$, with $\mathbf{x} = [x_1 \ \dot{x}_1 \ x_2 \ \dot{x}_2 \ x_3 \ \dot{x}_3 \ x_4 \ \dot{x}_4]^T$, $\mathbf{u} = [u_1 \ u_2 \ u_3 \ u_4]^T$ and

$$A_{8 \times 8} = \text{blkdiag}[A_{2 \times 2}, A_{2 \times 2}, A_{2 \times 2}, A_{2 \times 2}], \quad (13)$$

$$B_{8 \times 4} = \text{blkdiag}[B_{2 \times 1}, B_{2 \times 1}, B_{2 \times 1}, B_{2 \times 1}].$$

The single obstacle in 2D case was extended to 8D by adding a length of 2 units symmetrically in the extra dimensions. Also, $\mathbf{x}_s = [-2 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$, $\mathbf{x}_g = [2 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$, $\mathcal{X}_g = \mathcal{E}(\mathbf{x}_g, I_8)$, $u_i \in [-1 \ 1]$, $i \in \{1, 2, 3, 4\}$.

Moon-lander Robot: A simplified version of a planar “moon-lander robot” is illustrated in Fig. 3. The robot has three thrusters F_l , F_r and F_t acting in the left, right and up direction respectively. In the absence of upwards thrust, the robot falls

under gravity. The dynamics of the robot is assumed to be as follows.

$$\frac{d}{dt} \begin{bmatrix} x \\ z \\ \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ z \\ \dot{x} \\ \dot{z} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_l \\ F_r \\ F_t \end{bmatrix}. \quad (14)$$

The start, goal and admissible control space were set as follows: $\mathbf{x}_s = [0 \ 1 \ 0 \ -2]^T$, $\mathbf{x}_g = [0 \ -4 \ 0 \ 0]^T$, $\mathcal{X}_g = \mathcal{E}(\mathbf{x}_g, 0.25 \ I)$, $F_l \in [0 \ 1]$, $F_r \in [0 \ 1]$, $F_t \in [-2 \ 2]$. The objective is to land the robot in time-optimal fashion.

Planar Quadrotor model: A linearized quadrotor model for longitudinal flight based on [28] can be written as $\dot{\mathbf{x}} = A_{6 \times 6} \mathbf{x} + B_{6 \times 2} \mathbf{u}$, with $\mathbf{x} = [x \ z \ u \ w \ q \ \theta]^T$, $\mathbf{u} = [f_t \ \tau_y]^T$ and

$$A_{6 \times 6} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -g \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad B_{6 \times 2} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1/m & 0 \\ 0 & 1/I_y \\ 0 & 0 \end{bmatrix}. \quad (15)$$

The start, goal and admissible control space were set as follows: $\mathbf{x}_s = [-2.5 \ 0 \ 0 \ 0 \ 0 \ 0]^T$, $\mathbf{x}_g = [2.5 \ 0 \ 0 \ 0 \ 0 \ 0]^T$, $\mathcal{X}_g = \mathcal{E}(\mathbf{x}_g, I)$, $f_t \in [-1 \ 1]$, $\tau_y \in [-1 \ 1]$. The set-up for time-optimal planning problem is shown in Fig. 3.

The results of the numerical simulations are illustrated in Fig. 5. It can be seen that Informed Propagation (blue) performs better than the naïve uniform exploration (magenta). However, using the proposed TIE strategy, a combination of Algorithm 1 and 2, outperforms the other methods in all cases. Note that for a planner such as SST, the sampling procedure influences the vertex to be selected for forward propagation. Generating random samples $\mathbf{x}_{\text{cand}} \in \Omega(T)$ biases the selection of vertices in the TIS for expansion. After a vertex is selected,

expansion is performed by forward propagating the system dynamics to generate a new candidate vertex $\mathbf{v}$. The vertex inclusion algorithm then ensures that the candidate vertex $\mathbf{v} \in \Omega(T)$. Thus, the combination of the proposed sampling and inclusion algorithm avoids redundant exploration focuses search, and leads to a faster convergence in all cases. In order to study the computational cost incurred by the TIE procedure, the quadrotor simulation was run without obstacles. Compared to uniform sampling, TIE generates a lower number of feasible, candidate vertices for inclusion in the planner tree, as illustrated in Fig. 5. Future work will explore leveraging GPUs and operating on batch of samples and reachable sets in parallel. Parameter n_s controls the maximum number of attempts made to generate a new sample $\mathbf{x}_{\text{cand}} \in \Omega(T)$ before a uniform random sample is returned. In order to analyze the effect of n_s , the “fallback ratio” is defined as,

$$\text{Fallback Ratio} = \frac{\text{Number of Fallbacks to Uniform Sampling}}{\text{Number of Calls to the TIE Sampler}}$$

The fallback ratio was found to be negligible for the lower dimensional 2D system. It is relatively higher for the 6D quadrotor simulation run in a no-obstacle environment (see Fig. 5). This ratio can be decreased by increasing n_s . However, a large value of n_s corresponds to a larger amount of computations invested in the sampling procedure, which can adversely impact the convergence of solution cost. Note that while the sampling algorithm may return $\mathbf{x}_{\text{cand}}$ outside the TIS if n_s attempts are exhausted, the vertex inclusion procedure ensures that a candidate vertex $\mathbf{v}$ is incorporated in the planner tree only if it lies in the TIS.

VIII. CONCLUSION

In this work, we use ideas from reachability analysis to define a “Time-Informed Set”, to focus exploration after an initial solution is found. We prove that exploring the TIS is a necessary condition to improve the current solution. The proposed method can be applied to a variety of systems for which an efficient local steering module may not be available, but (over-)approximations of the reachable sets can be constructed.

It should be noted that the L_2 -Informed set is *sharp* [12], i.e., it uses a heuristic estimate which gives the exact cost-to-come and cost-to-go for any point in the absence of obstacles. The TIS is not so, as it is constructed using *over-approximations* of the reachable sets. Hence, finding tight approximations of the reachable sets is critical for the efficacy of the proposed approach.

In order to apply TIE for sampling-based planning, the reachability library needs to be constructed offline. Creating, storing and accessing this library should be computationally efficient for higher dimensional systems to be of use in practice. The ellipsoidal reachable sets used in this work satisfy these criteria. The HJB reachability toolboxes [15] can be potentially used to create this library for a general non-linear systems. These frameworks solve the value function PDE by discretizing the state space. However, the computational cost of these methods scale exponentially with the dimension. In order to address this curse of dimensionality and apply TIE

for non-linear systems and general cost functions, application of machine-learning frameworks for reachability, such as [29], [30], can be explored. Recent works such as DeepReach [30] avoid gridding the state space and use deep neural networks (DNN) to learn a parameterized approximation of the value function. These DNNs can be stored and used to classify or generate new samples in the TIS.

REFERENCES

- [1] S. M. LaValle and J. J. Kuffner Jr, “Randomized kinodynamic planning,” *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.
- [2] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 6 2011.
- [3] O. Arslan and P. Tsotras, “Use of relaxation methods in sampling-based algorithms for optimal motion planning,” in *IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, May 6–10 2013, pp. 2421–2428.
- [4] L. Janson, E. Schmerling, A. Clark, and M. Pavone, “Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions,” *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 883–921, May 2015.
- [5] 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 *IEEE International Conference on Robotics and Automation*, Seattle, WA, May, 25–30 2015, pp. 3067–3074.
- [6] S. Karaman and E. Frazzoli, “Optimal kinodynamic motion planning using incremental sampling-based methods,” in *49th IEEE Conference on Decision and Control*, Atlanta, GA, Dec 2010, pp. 7681–7687.
- [7] A. Perez, R. Platt, G. Konidaris, L. Kaelbling, and T. Lozano-Perez, “LQR-RRT*: Optimal sampling-based motion planning with automatically derived extension heuristics,” in *IEEE International Conference on Robotics and Automation*, Saint Paul, MN, 2012, pp. 2537–2542.
- [8] D. J. Webb and J. Van Den Berg, “Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics,” in *IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, 2013, pp. 5054–5061.
- [9] E. Schmerling, L. Janson, and M. Pavone, “Optimal sampling-based motion planning under differential constraints: the drift case with linear affine dynamics,” in *IEEE Conference on Decision and Control (CDC)*, Osaka, Japan, Dec 15–18 2015, pp. 2574–2581.
- [10] J. hwan Jeon, S. Karaman, and E. Frazzoli, “Optimal sampling-based feedback motion trees among obstacles for controllable linear systems with linear constraints,” in *IEEE International Conference on Robotics and Automation*, Seattle, Washington, May 26–30 2015, pp. 4195–4201.
- [11] Y. Li, Z. Littlefield, and K. E. Bekris, “Sparse methods for efficient asymptotically optimal kinodynamic planning,” in *Algorithmic Foundations of Robotics XI*. Springer, 2015, pp. 263–282.
- [12] J. D. Gammell, T. D. Barfoot, and S. S. Srinivasa, “Informed sampling for asymptotically optimal path planning,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 966–984, 8 2018.
- [13] T. Kunz, A. Thomaz, and H. Christensen, “Hierarchical rejection sampling for informed kinodynamic planning in high-dimensional spaces,” in *IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, May 16–21 2016, pp. 89–96.
- [14] D. Yi, R. Thakker, C. Gulino, O. Salzman, and S. Srinivasa, “Generalizing informed sampling for asymptotically-optimal sampling-based kinodynamic planning via Markov Chain Monte Carlo,” in *IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia, May 21–25 2018, pp. 7063–7070.
- [15] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, “Hamilton-Jacobi reachability: A brief overview and recent advances,” in *IEEE 56th Annual Conference on Decision and Control (CDC)*, Brisbane, Australia, Dec. 12–15 2017, pp. 2242–2253.
- [16] A. Kurzhanskiy, P. Varaiya, and W. Levine, “Computation of reach sets for dynamical systems,” *The Control Systems Handbook*, 2010.
- [17] B. Akgun and M. Stilman, “Sampling heuristics for optimal motion planning in high dimensions,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA, Sept. 25–30 2011, pp. 2640–2645.

- [18] O. Arslan and P. Tsiotras, "Machine learning guided exploration for sampling-based motion planning algorithms," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Hamburg, Germany, Sept. 28–Oct. 2, 2015, pp. 2646–2652.
- [19] B. Ichter, J. Harrison, and M. Pavone, "Learning sampling distributions for robot motion planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia, May 21–25 2018, pp. 7087–7094.
- [20] Z. Littlefield and K. E. Bekris, "Informed asymptotically near-optimal planning for field robots with dynamics," in *Field and Service Robotics*. Springer, 2018, pp. 449–463.
- [21] —, "Efficient and asymptotically optimal kinodynamic motion planning via dominance-informed regions," in *IEEE International Conference on Intelligent Robots and Systems*, Madrid, Spain, Oct. 1–5 2018, pp. 1–9.
- [22] A. Shkolnik, M. Walter, and R. Tedrake, "Reachability-guided sampling for planning under differential constraints," in *IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 12–17 2009, pp. 2859–2865.
- [23] S. D. Pendleton, W. Liu, H. Andersen, Y. H. Eng, E. Frazzoli, D. Rus, and M. H. Ang, "Numerical approach to reachability-guided sampling-based motion planning under differential constraints," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1232–1239, July 2017.
- [24] H.-T. L. Chiang, J. Hsu, M. Fiser, L. Tapia, and A. Faust, "RL-RRT: Kinodynamic motion planning via learning reachability estimators from rl policies," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4298–4305, 2019.
- [25] A. Girard, C. Le Guernic, and O. Maler, "Efficient computation of reachable sets of linear time-invariant systems with inputs," in *International Workshop on Hybrid Systems: Computation and Control*. Springer, 2006, pp. 257–271.
- [26] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, Dec. 2012.
- [27] M. Moll, I. A. Sucan, and L. E. Kavraki, "Benchmarking motion planning algorithms: An extensible infrastructure for analysis and visualization," *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, pp. 96–102, 2015.
- [28] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The GRASP multiple micro-UAV testbed," *IEEE Robotics & Automation Magazine*, vol. 17, no. 3, pp. 56–65, 2010.
- [29] R. E. Allen, A. A. Clark, J. A. Starek, and M. Pavone, "A machine learning approach for real-time reachability analysis," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, IL, Sept 14–18 2014, pp. 2202–2208.
- [30] S. Bansal and C. Tomlin, "Deepreach: A deep learning approach to high-dimensional reachability," *arXiv preprint arXiv:2011.02082*, 2020.