Time-Informed Exploration For Robot Motion Planning

A PREPRINT

Sagar Suhas Joshi * Seth Hutchinson † Panagiotis Tsiotras ‡

April 14, 2020

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. This work defines a “Time Informed Set”, using ideas from reachability analysis, 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. Exploration outside this set is hence redundant. Benchmarking experiments show that an exploration strategy based on the TIS can accelerate the convergence of sampling-based kino-dynamic motion planners.

1 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 and dynamic programming 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, for a general kino-dynamic system, solving a two-point boundary value problem (TPBVP), also called the “local steering” problem, is necessary 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] 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

*Robotics PhD student at Georgia Institute of Technology, USA. sagarsjoshi94@gmail.com
†Professor and KUKA Chair for Robotics in the School of Interactive Computing at Georgia Institute of Technology, USA. seth@gatech.edu
‡Professor and David and Andrew Lewis Chair in the Guggenheim School of Aerospace Engineering at Georgia Institute of Technology, USA. tsiotras@gatech.edu
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] defined 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 foci at the start and the goal states and its transverse diameter is equal to the current best solution cost. The direct Informed Sampling 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 of the Informed Set for the case of time-optimal kino-dynamic planning using ideas from reachability analysis. Given a feasible (but sub-optimal) solution trajectory with time cost $T > 0$, we define a Time Informed Set (TIS), that contains all the trajectories with time cost less than $T$. 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.

## 2 RELATED WORK

Prior work on intelligent exploration, such as [15], [12], [16], [17] utilized heuristics and ideas from deep learning to improve the performance of sampling-based planners. The Informed SST (iSST) algorithm [18] also leverages heuristics to guide search for kino-dynamic planning. DIRT [19] uses dominance informed regions along with heuristics to balance exploration and exploitation. However, in many cases iSST and DIRT cannot be applied directly for time optimal planning due to the unavailability of a heuristic function.

Concepts from reachability analysis have also been used for guiding exploration in sampling-based kino-dynamic planning. Shkolnik et al. [20] 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 [21] to be used for sampling and nearest neighbor search. Chiang et al. [22] train an obstacle-aware time-to-reach (TTR) reachability estimator network to guide the RRT search process. However, the above techniques do not focus search to a sub-set 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 magnitude to travel 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
In this work, we address this issue by using ideas from reachability analysis to define the TIS. The proposed exploration 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 numerical experiments and their results.

3 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 $x_0 \in \mathcal{X}_{\text{free}}$ 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:

$$
T^* = \min_u T
$$

subject to:

$$
\begin{align*}
\dot{x}(t) &= f(x(t), u(t)), \\
x(0) &= x_0, \ x(T) &\in \mathcal{X}_g, \\
x(t) &\in \mathcal{X}_{\text{free}}, \ u(t) \in \mathcal{U} \quad \text{for all } t \in [0, T].
\end{align*}
$$

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

4 TIME-INFORMED SET

Consider the set of points that can be reached at time $t$, starting from $x_0$ at time $t_0 < t$, using admissible controls,

$$
\mathcal{X}_f[t_0, t] = \{ x \in \mathcal{X} \mid \exists u : [t_0, t] \rightarrow \mathcal{U}, \ x : [t_0, t] \rightarrow \mathcal{X}, \ s.t. x(t_0) = x_0, x(t) \in \mathcal{X}_g, \ \dot{x}(t) = f(x(t), u(t)) \}. 
$$

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

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

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 $[t, t_f]$ can be defined as the set of all points starting at time $t$, that can reach $\mathcal{X}_g$ at any time $\tau \in [t, t_f]$

$$
\mathcal{R}_b[t, t_f] = \bigcup_{\tau \leq \tau \leq t_f} \mathcal{B}[t, \tau].
$$
Assume that a feasible (sub-optimal) solution to problem (1) with time cost $T > 0$ is available. Consider the following definition of the Time Informed Set (TIS)

$$ \Omega(T) = \bigcup_{0 \leq s \leq T} F[0, t] \cap R_b[t, T]. $$ (5)

Intuitively, $\Omega(T)$ contains all the points $x \in X$ that can be reached from $x_a$ at a time $t$, $0 \leq t \leq T$, i.e., $x \in F[0, t]$ and then can reach the goal at time $\tau$, $t \leq \tau \leq T$, i.e., $x \in R_b[t, T]$. Please see Fig. 2 and the attached video for a visualization of $\Omega(T)$.

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

**Lemma 1.** Given a sub-optimal solution with cost $T > 0$, $F[0, t] \cap B[t, T] \neq \emptyset$ for all $t \in [0, T]$.

**Proof.** Consider the solution trajectory with time cost $T$, $\zeta : [0, T] \to X$, where $\zeta(0) = x_a$ and $\zeta(T) = x_g$. For any point $x$ on this trajectory, there exists $t \in [0, T]$ such that $x = \zeta(t)$. Thus, $x \in F[0, t]$ and $x \in B[t, T]$. It follows that, $x \in F[0, t] \cap B[t, T]$. Therefore, $F[0, t] \cap B[t, T] \neq \emptyset$. \qed

**Lemma 2.** $R_b[t, T_1] \subset R_b[t, T_2]$ for any $T_2 > T_1 > t > 0$.

**Proof.** Note from the definition (4),

$$ R_b[t, T_2] = \bigcup_{t \leq \tau \leq T_2} B[t, \tau] = \left( \bigcup_{t \leq \tau \leq T_1} B[t, \tau] \right) \bigcup \left( \bigcup_{T_1 \leq \tau \leq T_2} B[t, \tau] \right). $$

Now note that, $R_b[t, T_1] = \bigcup_{t \leq \tau \leq T_1} B[t, \tau]$. Hence, $R_b[t, T_1] \subset R_b[t, T_2]$. \qed

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

**Proof.** Consider any solution trajectory $\zeta : [0, T] \to X$ with time cost $T > 0$, where $\zeta(0) = x_a$, $\zeta(T) = x_g$. For any point $x$ on this trajectory, there exists $t \in [0, T]$ such that $x = \zeta(t)$. Then, $x \in F[0, t]$ and $x \in B[t, T]$. This implies $x \in F[0, t] \cap B[t, T]$ and hence $x \in F[0, t] \cap R_b[t, T]$. Thus, $x \in \Omega(T)$. \qed

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

**Proof.** Consider,

$$ \Omega(T_2) = \bigcup_{0 \leq s \leq T_2} F[0, t] \cap R_b[t, T_2] = \left( \bigcup_{0 \leq s \leq T_1} F[0, t] \cap R_b[t, T_2] \right) \bigcup \left( \bigcup_{T_1 \leq s \leq T_2} F[0, t] \cap R_b[t, T_2] \right) $$

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

**Theorem 5.** Given a sub-optimal solution to (7) 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 $T$. Theorem 4 implies that $\Omega(T)$ is a superset of all the sets containing trajectories with time cost less than $T$. Thus, $\Omega(T)$ contains all the trajectories with cost less than or equal to $T$. \qed

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

---

https://www.youtube.com/watch?v=x0W8stLCR50
Algorithm 1: Sampling Algorithm

generateSample \((T)\):

\begin{algorithmic}[1]
  \STATE \(t \sim p_{0,T}(t)\);
  \FOR {\(i = 1 : n_s\)}
    \IF {\(\lambda(F[0,t]) < \lambda(B[t,T])\)}
      \STATE \(x_{cand} \leftarrow \text{sampleUniform}(F[0,t])\);
      \IF {\(x_{cand} \in B[t,T]\)}
        \STATE \(x_{rand} \leftarrow x_{cand}\);
        \STATE \text{return} \(x_{rand}\);
      \ENDIF
    \ELSE
      \STATE \(x_{cand} \leftarrow \text{sampleUniform}(B[t,T])\);
      \IF {\(x_{cand} \in F[0,t]\)}
        \STATE \(x_{rand} \leftarrow x_{cand}\);
        \STATE \text{return} \(x_{rand}\);
      \ENDIF
    \ENDIF
  \ENDFOR
  \STATE \(x_{rand} \leftarrow \text{sampleUniform}(\mathcal{X})\);
  \STATE \text{return} \(x_{rand}\);
\end{algorithmic}

Algorithm 2: Vertex Inclusion Algorithm

includeVertex \((v, t, T)\):

\begin{algorithmic}[1]
  \IF {\(t > T\)}
    \STATE \text{return} false;
  \ENDIF
  \FORALL {\(\tau \in \{t + \delta, t + 2\delta, \ldots T\}\)}
    \IF {\(v \in B[t, \tau]\)}
      \STATE \text{return} true;
    \ENDIF
  \ENDFOR
  \STATE \text{return} false;
\end{algorithmic}

5 EXPLORATION ALGORITHM

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 [23], ellipsoidal calculus [24] and formulating reachability problem as a Hamilton-Jacobi-Bellman (HJB) PDE [25]. 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 now the special case of linear kina-dynamic systems. Concretely, the constraint (1b) now is \(\dot{x}(t) = Ax(t) + Bu(t)\), with \(A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m}\). Then, \(\mathcal{X}_f[t_0, t] \) and \(\mathcal{X}_b[t, t_f] \) can be defined as

\[
\mathcal{X}_f[t_0, t] = \{x \in \mathcal{X} : \exists u : [t_0, t] \rightarrow \mathcal{U}, \text{s.t.} x = e^{A(t-t_0)}x_0 + \int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau) \, d\tau \},
\]

\[
\mathcal{X}_b[t, t_f] = \{x \in \mathcal{X} : \exists u : [t, t_f] \rightarrow \mathcal{U}, \text{s.t.} x = e^{-A(t_f-t)}x_g - \int_{t}^{t_f} e^{-A(t-\tau)}Bu(\tau) \, d\tau \}. \tag{6}
\]

Here, \(x_g \in \mathcal{X}_{\text{goal}} \). A hyper-sphere over-approximation to the above sets can be constructed as follows [23].

\[
\mathcal{F}[t_0, t] = \{x \in \mathcal{X} : \|x - e^{A(t-t_0)}x_0\|_2 \leq r(t_0, t, u_{\text{max}})\}
\]

\[
\mathcal{B}[t, t_f] = \{x \in \mathcal{X} : \|x - e^{-A(t_f-t)}x_g\|_2 \leq r(t, t_f, u_{\text{max}})\} \tag{7}
\]

\[
r(t_1, t_2, u_{\text{max}}) = (e^{\|A\|_2(t_2-t_1)} - 1) \frac{\|B\|u_{\text{max}}}{\|A\|_2}
\]
In contrast, the ellipsoidal technique approximates the reachable sets as ellipsoids. The forward reachable sets $\mathcal{F}[0,t]$ at $t = 2$ using the hyper-sphere and ellipsoidal approximation.

Figure 3: Comparison the forward reachable set $\mathcal{F}[0,t]$ at $t = 2$ using the hyper-sphere and ellipsoidal approximation.

Here, $\|\cdot\|_2$ represents the $L_2$-norm and $\|\cdot\|_2$ represents the induced two norm (maximum singular value) for a matrix $M \in \mathbb{R}^{n \times k}$. As the set $\mathcal{U}$ is compact, there exists a $u_{\text{max}}$, so that $\|u(t)\|_2 \leq u_{\text{max}}$ for all $t$. However, the above over-approximation might be too conservative for the current application. Please see Fig. 2.

In contrast, the ellipsoidal technique approximates the reachable sets as ellipsoids $\mathcal{E}(x_c, Q)$,

$$\mathcal{E}(x_c, Q) = \{x \in \mathbb{R}^n | \langle x - x_c, Q^{-1}(x - x_c) \rangle \leq 1 \}. \quad (8)$$

Here, $x_c$ 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 a ordinary differential equation (ODE) for the center and shape matrix. Please see the ellipsoidal toolbox 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 $x_c$. The goal set $x_g$ is represented also as a hyper-sphere with a set radius around a center $x_g \in x_g$. The ODE for the shape matrix can be solved and stored off-line. An analytical solution for the ODE describing center’s trajectory can also be constructed. Thus, a “library” of reachable sets $\mathcal{F}[0,t], \mathcal{B}[t,T]$ can be created to be used in the sampling and vertex inclusion algorithm described below. Please see Fig. 2 for a visualization of $\mathcal{F}[0,t]$ and $\mathcal{B}[t,T]$ constructed using the ellipsoid technique.

### 5.1 Sampling Algorithm

Algorithm 1 describes a procedure to generate a new sample $x_{\text{rand}}$ 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 $t$ in the interval $[0,T]$ according to a probability distribution $p_{[0,T]}(t)$ (line 2). Ideally, to generate uniform random samples in $\Omega(T)$ w.r.t to the Lebesgue measure, this distribution needs to be $p_{[0,T]}(t) = \lambda(\mathcal{F}[0,t] \cap \mathcal{R}_u[t,T]) / (\lambda(\Omega(T)))$. However, calculating and sampling this distribution may not be tractable for a 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 $x_{\text{rand}} \in \mathcal{F}[0,t] \cap \mathcal{B}[t,T]$. 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 $\mathcal{B}[t,T]$, else, $\mathcal{B}[t,T]$ 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 $x_{\text{rand}} \in \mathcal{F}[0,t] \cap \mathcal{B}[t,T]$ can be generated in $n_s$ attempts, the algorithm returns a uniform random sample from the search-space $\mathcal{X}$ (line 14-15).

### 5.2 Vertex Inclusion Algorithm

The vertex inclusion procedure, described in Algorithm 2 accepts a candidate vertex $v$ with cost-to-come $t$, i.e., the cost of trajectory from $x_s$ to $v$ is $t$. As the cost-to-come is $t$, we have $v \in \mathcal{F}[0,t]$. Thus, if $v \in \mathcal{R}_u[t,T]$, then $v \in \Omega(T)$. The proposed algorithm discretizes the interval $[t,T]$ with a

---

[1] http://systemanalysisdpt-cmc-msu.github.io/ellipsoids/doc/main_manual.html
Figure 4: A schematic for the moon-lander robot (top) and quadrotor botton) simulation cases with sample solution paths found by the proposed algorithm after 40 sec of planning time.

6 NUMERICAL EXPERIMENTS

Benchmarking experiments were performed by pairing different exploration strategies with the SST planner. All algorithms were implemented in C++ using the popular OMPL framework [26], and the tests were run using OMPL’s standardized benchmarking tools [27]. 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 with Informed propagation, i.e., given a sub-optimal solution with cost $T$ and a vertex $v$ with a cost-to-come $t$, forward propagation from $v$ was done for almost $T - t$ duration. A description of different case-studies is given below.

6.1 2D Toy system

Consider a 2D kino-dynamic system described below,

$$\frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \\ z \end{bmatrix} = \begin{bmatrix} 0, 0.5 \\ -0.1, 0, 2 \\ 0, 0, 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} u. \quad (9)$$

The set-up of the planning problem is illustrated in Fig. 1 with $x_s = [-3, 0], x_g = [3, 0], x_g = E(x_g, 0.25 I), u \in [-0.5, 0.5].$ Here, $I$ represents the identity matrix of appropriate dimensions.

6.2 Moon-lander Robot

A simplified version of a planar “moon-lander robot” is illustrated in Fig. 4. 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.

$$\frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \\ z \end{bmatrix} = \begin{bmatrix} 0, 0, 1, 0 \\ 0, 0, 1 \\ 0, 0, 1, 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ z \end{bmatrix} + \begin{bmatrix} 0, 0, 0 \\ 0, 0, 0 \\ -2, 1, 0 \\ 0, 0, 1 \end{bmatrix} \begin{bmatrix} F_l \\ F_r \\ F_t \end{bmatrix}. \quad (10)$$

The start, goal and admissible control space were set as follows: $x_s = [0, 1, 0, -2], x_g = [0, -4, 0, 0], x_g = E(x_g, 0.25 I), F_l \in [0, 1], F_r \in [0, 1], F_t \in [-2, 2].$ The objective is to land the robot in time-optimal fashion.
6.3 Planar Quadrotor model

A linearized quadrotor model for longitudinal flight based on [28] is given below.

\[
\frac{dx}{dt} = \begin{bmatrix}
0 & 0, 1, 0, 0 & 0, 0, 1, 0, 0 & 0, 0, 0, 0, 0 & 0, 0, 0, 0, 0 \\
0, 0, 0, 0, 0, & -g & 0, 0, 0, 0, 0, & 0, 0, 0, 0, 0 \\
0, 0, 0, 0, 0, & 0, 0, 0, 0, 0, & 0, 0, 0, 0, 0 \\
0, 0, 0, 0, 0, & 0, 0, 0, 0, 0, & 0, 0, 0, 0, 0 \\
0, 0, 0, 0, 0, & 0, 0, 0, 0, 0, & 0, 0, 0, 0, 0
\end{bmatrix}
\begin{bmatrix}
x \\
z \\
u \\
w \\
v \\
q \\
\theta \\
f_t \\
\tau_y
\end{bmatrix}
\]

(11)

The start, goal and admissible control space were set as follows: \(x_s = [-2.5, 0, 0, 0, 0, 0] \), \(x_g = [2.5, 0, 0, 0, 0, 0] \), \(X_g = E(x_g, l) \), \(f_t \in [-2, 2] \) and \(\tau_y \in [-2, 2] \). The set-up for time-optimal planning problem is shown in Fig. [3].

Results of the numerical simulations are illustrated in Fig. 5. It can be seen that the application of only the vertex inclusion algorithm [2] (blue plot), leads to a better performance than the naive uniform exploration (green plot). However, using the Time Informed Sampling and vertex inclusion algorithm [1 and 2] together outperforms 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 in the TIS ensures that vertices in the TIS are selected for forward propagation. After a new candidate vertex is generated, the inclusion procedure ensures that the vertex lies in the TIS. Thus, the combination of proposed sampling and inclusion algorithm leads to a focused search in the TIS and a faster convergence in all cases.

7 CONCLUSION

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

It should be noted that \(L_2 \) Informed set is sharp [12], i.e., it is uses 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 an over-approximations of the reachable sets. Hence, finding tight approximations of the reachable sets is critical for the efficacy of the proposed approach.

This work presents many opportunities for future research. First, the proposed framework can be extended for general cost-functions and non-linear dynamics using the HJB reachability framework. Second, the system decomposition techniques, proposed in [26] can be explored to obtain reachable sets for sub-systems that can be used in HRS [13] like procedure for Informed exploration. Third, machine learning approaches like [30] can be used to classify if a sample point lies in the TIS. Deep generative models can also be used to devise a more computationally efficient procedure to generate new samples inside the TIS.

Acknowledgements: Authors thank Dipankar Maity and Kelsey Hawkins for insightful discussions on this topic. This work has been supported by NSF award IIS-1617630.
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, 2011.

[3] O. Arslan and P. Tsiotras, “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, May 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, 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] 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.

[16] 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.

[17] 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.

[18] 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.

[19] ———, “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.
[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] A. Kurzhanskiy, P. Varaiya, and W. Levine, “Computation of reach sets for dynamical systems,” The Control Systems Handbook, 2010.

[25] 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.

[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] M. Chen, S. L. Herbert, M. S. Vashishtha, S. Bansal, and C. J. Tomlin, “Decomposition of reachable sets and tubes for a class of nonlinear systems,” IEEE Transactions on Automatic Control, vol. 63, no. 11, pp. 3675–3688, 2018.

[30] 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.

[31] B. Ivanovic, J. Harrison, A. Sharma, M. Chen, and M. Pavone, “BaRC: Backward reachability curriculum for robotic reinforcement learning,” in IEEE International Conference on Robotics and Automation (ICRA), Montreal, Canada, May 20–24 2019, pp. 15–21.