2023 IEEE International Conference on Robotics and Automation (ICRA)
29 May-2 June 2023, London, UK

A Reachability-Based Spatio-Temporal Sampling Strategy for Kinodynamic Motion Planning

Yongxing Tang [1,2], Zhanxia Zhu[1,2], Hongwen Zhang [3*]

***Abstract*—By limiting the planning domain to "L2 Informed Set", some sampling-based motion planners (SBMP) (e.g. Informed RRT*, BIT*) can solve the geometric motion planning problems efficiently. However, the construction of informed set (IS) will be very challenging, when further differential constraints are considered. For the time-optimal kinodynamic motion planning problem, this paper defines a modified time informed set (MTIS) to limit the planning domain. Due to drawing inspiration from Hamilton-Jacobi-Bellman (HJB) reachability analysis, MTIS, compared with the original TIS, can not only help save the running time of SBMP, but also extend the applicable scope from linear systems to polynomial nonlinear systems with control constrains. On this basis, a spatio-temporal sampling strategy adapted to MTIS is proposed. Firstly, MTIS is used to estimate the optimal cost and the valid tree structure is reused, so that we do not need to provide a solution trajectory in advance. Secondly, this strategy is generic, allowing it to be combined with common SBMP (such as SST, etc.) to accelerate convergence and reduce the memory requirement. Several simulation experiments also demonstrate the effectiveness of proposed method.**

I. Introduction

The prominent characteristic of SBMP is that they avoid the explicit representation of obstacles through collision detection module, and use connections between countable samples to approximately capture the connectivity of the uncountably infinite free state space. The classical algorithms include PRM [1], which constructs the roadmap first and then searches, RRT [2], which constructs the tree and searches simultaneously, and the optimal versions of both, PRM* [3] and RRT* [3]. As the number of samples approaches infinity, although RRT* can ensure asymptotic convergence to the optimal solution, the search domain is set as the whole state space, which wastes significant computing time and makes it difficult to apply to some practical problems, such as planning in unbounded space and high-dimensional space. Restricting the growth domain of the search tree is a means to improve planning efficiency. [4] proposed Anytime RRT* by using the graph pruning to periodically remove those nodes where the sum of current cost and heuristic cost is greater than the cost of the current best path. But this is a passive, backward-looking way of eliminating redundant samplings outside the growth domain. [5] samples the full state space, and rejects nodes that do not to reduce the cost of the path, once a path is found. However, as the proportion of the focused domain in state space decreases, the rejection sampling strategy becomes less likely to succeed. Informed RRT* [6] and BIT* [7] sample directly in the hyper-ellipsoidal "L_2 Informed Set", which contains all the potentially better paths. As the IS shrinks continually, they can find the asymptotic optimal path [8]. Compared with graph pruning [4] and rejection sampling [5], the direct sampling strategy can not only actively avoid redundant samplings, but also can improve the convergence rate and path quality even when the measure of IS decreases.

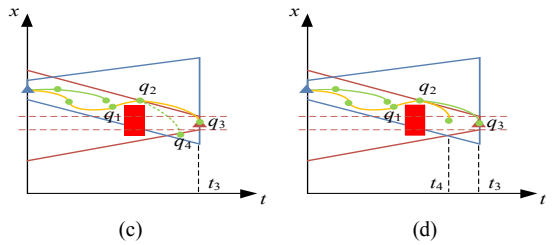


Figure 1. An illustration of the proposed sampling strategy for 1D system. The states are green dots, the start is the blue triangle, the target is the red triangle, the red rectangle is obstacle, valid edges in the tree are solid green lines, invalid edges in the tree are dotted green lines, the current suboptimal trajectory is solid orange lines, the blue trapezoid is FRS, the red trapezoid is BRSII. (a) Regardless of obstacles, the estimated optimal time from the start to the target is determined by BRSII, and the tree is generated in the corresponding MTIS. (b) If no trajectory is found, then MTIS is enlarged. The previous tree is reused, and the search continues. For our spatio-temporal tree, the true nearest node of q_3 should be q_2. If the temporal coordinate is not taken into account, the nearest node of q_3 would be q_1, which results in an invalid extension. (c) When a suboptimal solution is found, MTIS shrinks. Since nodes outside MTIS cannot provide a better solution trajectory, redundant node q_4 is pruned to reduce memory requirement. (d) Continue searching for a better solution trajectory and repeat (c).

Unfortunately, it is not an easy thing to describe the parameterized hyper-ellipsoidal IS when differential constraints are considered. Therefore, although some RRT* class [9, 10, 11] and forward propagation class [12] algorithms have been proposed, they all generate samples in the whole state space. [13] proposed Hierarchical Rejection Sampling technology to alleviate the above redundant sampling problem. [14] reformulated the informed sampling problem as a uniform sampling problem within the sub-level set of an implicit non-convex function, thus enabling Monte Carlo sampling methods to be applied. However, these two methods are only applicable to the systems where a local steering function is available and cannot avoid the inherent drawback of the rejection sampling technique. For systems without a local steering function, [15] used ellipsoidal toolbox (ET) [16] to compute and store the forward reachable set (FRS) of initial state and backward reachable set (BRS) of target set offline, and derived the time-informed set (TIS) using set operations of reachable sets. Once the initial solution is found, subsequent searches are restricted to TIS. Although the regular hyper-ellipsoidal reachable sets allow the algorithm to sample

[1]The authors are with the School of Astronautics, Northwestern Polytechnical University, Xi'an, China. (e-mail: zhuzhanxia@nwpu.edu.cn, dg84@mail.nwpu. edu.cn).
[2] National Key Laboratory of Aerospace Flight Dynamics, Xi'an, China.
[3]The author is with the Zhejiang Lab, Hangzhou, China. (e-mail: zhanghw@zhejianglab.com).
[*]The corresponding author (e-mail: zhanghw@zhejianglab.com).

directly [8], ET can only compute the reachable set corresponding to the linear system at a specified time, which makes the original TIS extremely complicated (the BRS is composed of the union of a large amount of ellipsoids) and cannot work on nonlinear systems.

The main contributions of this paper to the problems mentioned above include: (i) Simplifying the original TIS via HJB reachability analysis and broadening its applicable scope to nonlinear systems. [17] transforms the problem of computing reachable set into the final value problem of HJB partial differential equation (PDE), and it is proved that the subzero level set of viscosity solution of this PDE can be used to represent the reachable set of the linear and nonlinear systems. However, the numerical result of HJB PDE depends on the grids, and the irregularity of the exact reachable set is not the desired characteristic of the IS, which is not conducive to direct sampling. This work substitutes the BRSI in TIS with BRSII (see Ⅱ.B for the definitions) and treats the external ellipsoidal approximation of reachable set as a bilinear optimization problem based on HJB reachability analysis. Furthermore, the Sum-of-Square Programming (SOSP) module [18] of YAMIP [19] is used to solve the problem, so as to obtain a simpler parameterized representation of TIS (i.e. MTIS). (ii) Since MTIS contains both spatio and temporal coordinates, we proposed a reachability-based spatio-temporal sampling strategy, which is illustrated in Fig.1. This strategy doesn't require an initial trajectory in advance as [6] and [15] do, and can be combined with common SBMP (such as SST [12], etc.). In addition, also different from prior works based on reachability [11, 20], our method aims to construct the sampling domain to avoid redundant samplings, rather than to quickly explore the forward reachable set of the system. In other words, the proposed method could be used alongside [11, 20] to achieve faster planning.

II. Problem Formulation and Preliminaries

This section defines the time-optimal kinodynamic motion planning problem and the modified time informed set, introduces the fundamental HJB reachability theory.

A. Time-optimal kinodynamic motion planning problem

Define $X \subset R^n$ ($n \geq 1$) as the state space. Let $X_{obs} \subseteq X$ represent the obstacle region, $X_{free} \subset X$ represent the free state space. Let $U \subset R^m$ ($m \geq 1$) represent the admissible control space. Let $\boldsymbol{x}_s \in X_{free}$ and $X_g \subseteq X_{free}$ represent the initial state and target set, respectively. Let $X_s \subseteq X_{free}$ represent the initial set which contains $\boldsymbol{x}_s$.

Assumption 2.1: X is compact and X_{obs} is closed. To ensure the optimal trajectory exist, X_{free} is defined as the closure, cl ($X \setminus X_{obs}$).

Assumption 2.2: The differential constrains introduced by kinematics and dynamics can be expressed as:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{1}$$

in which $\boldsymbol{x} \in X$ is the state, and $\boldsymbol{u}(t) \in U$ is the admissible control. U is compact. The flow field $\boldsymbol{f}$: $\underline{R^n} \times U \to R^n$ is uniformly continuous, bounded and Lipschitz continuous in $\boldsymbol{x}$ for fixed $\boldsymbol{u}$.

Assumption 2.3: Both the initial set X_s and target set X_g are closed, which can be denoted as the subzero level set of a bounded and Lipschitz continuous function h: $R^n \to R$ and g: $R^n \to R$ respectively.

$$X_s = \left\{ \boldsymbol{x} \in R^n \mid h(\boldsymbol{x}) \leq 0 \right\},\ X_g = \left\{ \boldsymbol{x} \in R^n \mid g(\boldsymbol{x}) \leq 0 \right\} \tag{2}$$

Then the time-optimal kinodynamic motion planning problem is following

$$\begin{aligned} &\min t_f \\ s.t \quad &\dot{\boldsymbol{x}} = \boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{u}(t)\right) \\ &\boldsymbol{x}(0) = \boldsymbol{x}_s,\ \boldsymbol{x}(t_f) \in X_g,\ t_f > 0,\ \textit{and}\ t_f\ \textit{is free} \\ &\boldsymbol{x}(t) \in X_{free},\ \boldsymbol{u}(t) \in U,\ \forall t \in [t_0, t_f]. \end{aligned} \tag{3}$$

Incremental-search based motion planning algorithms solve the above problem by incrementally constructing a tree structure. However, the optimal trajectory is actually only a small fraction of X_{free}. In other words, most of the nodes generated by these algorithms are useless in the search for optimal trajectory. In section IV, we will show how to get the desired trajectory with just fewer samples.

B. Reachable set & modified time informed set

Definition 2.1 (Forward reachable set): The forward reachable set $F(t, 0, \boldsymbol{x}_s)$ at time t from initial position $(0, \boldsymbol{x}_s)$ is the set of all states $\boldsymbol{x}(t)$ reachable at time t by system with $\boldsymbol{x}(0) = \boldsymbol{x}_s$ through all admissible control $\boldsymbol{u}(\tau) \in U$, $0 \leq \tau \leq t$. i.e.

$$\begin{aligned} F(t, 0, x_s) \triangleq \{ &\boldsymbol{x}_t \in X \mid \forall \tau \in [0, t], \exists \boldsymbol{u}(\tau) \in U, \\ &s.t\ \ \boldsymbol{x}(0) = \boldsymbol{x}_s, \boldsymbol{x}(t) = \boldsymbol{x}_t, \dot{\boldsymbol{x}}(\tau) = \boldsymbol{f}(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau)) \}. \end{aligned} \tag{4}$$

Assume that the time cost of current (estimated) best trajectory is $T > 0$ and $t_f \leq T$ holds.

Definition 2.2 (Backward reachable set I): The backward reachable set $B_{\mathrm{I}}(T, t, X_g)$ at time t for the target position (T, X_g) is the set of all states $\boldsymbol{x}(t)$ for which there exists some admissible control $\boldsymbol{u}(\tau) \in U$, $t \leq \tau \leq T$, that steers system to the target set X_g at time T. i.e.

$$\begin{aligned} B_{\mathrm{I}}(T, t, X_g) \triangleq \{ &\boldsymbol{x}_t \in X \mid \forall \tau \in [t, T], \exists \boldsymbol{u}(\tau) \in U, \\ &s.t\ \ \boldsymbol{x}(t) = \boldsymbol{x}_t, \boldsymbol{x}(T) \in X_g, \dot{\boldsymbol{x}}(\tau) = \boldsymbol{f}(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau)) \}. \end{aligned} \tag{5}$$

Definition 2.3 (Backward reachable set II): The backward reachable set $B_{\mathrm{II}}(T, t, X_g)$ at time t for the target position (T, X_g) is the set of all states $\boldsymbol{x}(t)$ for which there exists $\underline{\tau} \in [t, T]$ and some admissible control $\boldsymbol{u}(\tau) \in U$, $t \leq \tau \leq \underline{\tau}$, that steers system to the target set X_g at time $\underline{\tau}$. i.e.

$$\begin{aligned} B_{\mathrm{II}}(T, t, X_g) \triangleq \{ &\boldsymbol{x}_t \in X \mid \exists\, \underline{\tau} \in [t, T],\ \ \forall \tau \in [t, \underline{\tau}], \exists \boldsymbol{u}(\tau) \in U, \\ &s.t\ \ \boldsymbol{x}(t) = \boldsymbol{x}_t, \boldsymbol{x}(\underline{\tau}) \in X_g, \dot{\boldsymbol{x}}(\tau) = \boldsymbol{f}(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau)) \}. \end{aligned} \tag{6}$$

Remark 2.1: According to Definition 2.2 and 2.3, there is the relationship between BRSII and BRSI.

$$B_{\mathrm{II}}(T, t, X_g) = \bigcup_{\tau \in [t, T]} B_{\mathrm{I}}(\tau, t, X_g) \tag{7}$$

Assumption 2.4: Let $F_e(t, 0, \boldsymbol{x}_s)$, $B_{\mathrm{I}e}(\tau, t, X_g)$ and $B_{\mathrm{II}e}(T, t, X_g)$ be the external ellipsoidal approximation of FRS, BRSI and BRSII, respectively. For any $t \in [0, T]$ and $\tau \in [t, T]$, F_e, $B_{\mathrm{I}e}$, and $B_{\mathrm{II}e}$ all belongs to a compact set $X_{appr} \subset X$.

Then the time informed set $\Omega(T)$ in [15] is constructed by $F_e(t, 0, \boldsymbol{x}_s)$ and $B_{\mathrm{I}e}(\tau, t, X_g)$.

$$\Omega(T) \triangleq \bigcup_{t \in [0, T]} \left(F_e(t, 0, \boldsymbol{x}_s) \cap \left(\bigcup_{\tau \in [t, T]} B_{\mathrm{I}e}(\tau, t, X_g) \right) \right) \subset X \tag{8}$$

Now we replace BRSI with BRSII, defining the MTIS as follows:

$$\Omega_M(T) \triangleq \bigcup_{t\in[0,T]} \left(F_e(t,0,\boldsymbol{x}_s) \cap B_{\mathrm{II}e}(T,t,X_g)\right) \subset X. \quad (9)$$

Intuitively according to (7), the theorems about $\Omega(T)$ in [15] also hold for $\Omega_M(T)$, i.e. (i) $\Omega_M(T)$ contains all trajectories whose time costs are less than or equal to T. (ii) $\Omega(T_1)$ contains $\Omega(T_2)$ for any $0<T_2<T_1$. Therefore, every time a (possible) suboptimal solution is found, $X_{free} \cap \Omega_M(T)$ should take place of X_{free} in problem (3). Compared with $\Omega(T)$, $\Omega_M(T)$ has the advantage of reducing the time required to check whether a sample is located in BRSII.

C. Reachable Set Representation Based on HJB Analysis

Define the augmented system as:

$$\bar{\boldsymbol{f}}(\boldsymbol{x},\boldsymbol{u},\underline{u}) \triangleq \underline{u}\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}). \quad (10)$$

where $\underline{u}(t) \in [0, 1]$.

Let $V_B(\boldsymbol{x}, t)$ denote the viscosity solution of the following final value HJB PDE.

$$\begin{aligned} &\frac{\partial V_B(\boldsymbol{x},t)}{\partial t} + \min_{[\underline{u},\boldsymbol{u}]\in[0,1]\times U} \frac{\partial V_B(\boldsymbol{x},t)}{\partial \boldsymbol{x}^T}\bar{\boldsymbol{f}} = 0 \\ &V_B(\boldsymbol{x},T) = g(\boldsymbol{x}) \end{aligned} \quad (11)$$

then the subzero level set of $V_B(\boldsymbol{x}, t)$ describes the BRSII.

$$B_{\mathrm{II}}(T,t,X_g) = \left\{\boldsymbol{x}\in R^n \mid V_B(\boldsymbol{x},t) \le 0\right\} \quad (12)$$

Please refer to [17] for the proof.

Lemma 2.1:

$B_{\mathrm{II}}(T,t_1,X_g) = B_{\mathrm{II}}(T-\tau,t_1-\tau,X_g),\ \forall t_1 \le T,\ \tau \in R.$

Proof : Without loss of generality, we prove

$$B_{\mathrm{II}}(T,t_1,X_g) = B_{\mathrm{II}}(0,t_1-T,X_g).$$

Define $\gamma \in [t_1 - T, 0]$ and

$$\bar{V}_B(\boldsymbol{x},\gamma) = V_B(\boldsymbol{x},\gamma+T). \quad (13)$$

Therefore,

$$\bar{V}_B(\boldsymbol{x},0) = V_B(\boldsymbol{x},T) = g(\boldsymbol{x}). \quad (14)$$

Take the derivative of (13).

$$\frac{\partial \bar{V}_B(\boldsymbol{x},\gamma)}{\partial\gamma} = \frac{\partial V_B(\boldsymbol{x},\gamma+T)}{\partial(\gamma+T)} = \left.\frac{\partial V_B(\boldsymbol{x},t)}{\partial t}\right|_{t=\gamma+T} \quad (15)$$

So, let's substitute (13) and (15) into HJB PDE of $\bar{V}_B$.

$$\begin{aligned} &\frac{\partial \bar{V}_B(\boldsymbol{x},\gamma)}{\partial\gamma} + \min_{[\underline{u},\boldsymbol{u}]\in[0,1]\times U} \frac{\partial \bar{V}_B(\boldsymbol{x},\gamma)}{\partial \boldsymbol{x}^T}\bar{\boldsymbol{f}} \\ &= \left.\frac{\partial V_B(\boldsymbol{x},t)}{\partial t}\right|_{t=\gamma+T} + \min_{[\underline{u},\boldsymbol{u}]\in[0,1]\times U} \left.\frac{\partial V_B(\boldsymbol{x},t)}{\partial \boldsymbol{x}^T}\right|_{t=\gamma+T}\bar{\boldsymbol{f}} \end{aligned} \quad (16)$$

According to (11), the equation (16) is equal to 0. That is, $\bar{V}_B$ satisfies the final value HJB PDE in $[t_1 - T, 0]$. Then:

$$\begin{aligned} B_{\mathrm{II}}(0,t_1-T,X_g) &= \left\{\boldsymbol{x}\in R^n \mid \bar{V}_B(\boldsymbol{x},t_1-T) \le 0\right\} \\ &= \left\{\boldsymbol{x}\in R^n \mid V_B(\boldsymbol{x},t_1) \le 0\right\} = B_{\mathrm{II}}(T,t_1,X_g). \end{aligned} \quad (17)$$

Lemma 2.2: $B_{\mathrm{II}}(T_2, t, X_g) \subset B_{\mathrm{II}}(T_1, t, X_g),\ \forall T_1 > T_2 > t.$

Proof: This lemma is easy to derive from (7).

Theorem 2.1: $B_{\mathrm{II}}(T, t_3, X_g) \subset B_{\mathrm{II}}(T, t_2, X_g),\ \forall t_2 \le t_3 \le T.$

Proof: From Lemma 2.1, it follows that

$$B_{\mathrm{II}}(T,t_2,X_g) = B_{\mathrm{II}}(T+t_3-t_2,t_3,X_g).$$

Recall the Lemma 2.2,

$$B_{\mathrm{II}}(T+t_3-t_2,t_3,X_g) \supset B_{\mathrm{II}}(T,t_3,X_g).$$

Remark 2.2: (i) Theorem 2.1 illustrates the characteristic of BRSII shrinking over time. (ii) Lemma 2.1 implies that for a certain state, we only need to compute once and get much richer reachable sets through time transformation.

The FRS can be computed in a similar way as the BRS. The only difference is that the following initial value HJB PDE needs to be solved instead of final value HJB PDE

$$\begin{aligned} &\frac{\partial V_F(\boldsymbol{x},t)}{\partial t} + \min_{\boldsymbol{u}\in U} \frac{\partial V_F(\boldsymbol{x},t)}{\partial \boldsymbol{x}^T}\boldsymbol{f} = 0 \\ &V_F(\boldsymbol{x},0) = h(\boldsymbol{x}) \end{aligned} \quad (18)$$

where $V_F(\boldsymbol{x}, t)$ is the viscosity solution of (18). So

$$F(t,0,X_s) = \left\{\boldsymbol{x}\in R^n \mid V_F(\boldsymbol{x},t) \le 0\right\}. \quad (19)$$

For the sake of consistency, we can convert the initial value problem to the final value problem by change of variables like proof of Lemma 2.1. Define $\gamma = -t$, and

$$\bar{V}_F(\boldsymbol{x},\gamma) = V_F(\boldsymbol{x},-\gamma).$$

We can obtain

$$\frac{\partial \bar{V}_F(\boldsymbol{x},\gamma)}{\partial\gamma} - \min_{\boldsymbol{u}\in U} \frac{\partial \bar{V}_F(\boldsymbol{x},\gamma)}{\partial \boldsymbol{x}^T}\boldsymbol{f} = 0,\ \bar{V}_F(\boldsymbol{x},0) = h(\boldsymbol{x}). \quad (20)$$

Moreover,

$$\begin{aligned} F(t,0,X_s) &= \left\{\boldsymbol{x}\in R^n \mid V_F(\boldsymbol{x},t) \le 0\right\} \\ &= \left\{\boldsymbol{x}\in R^n \mid \bar{V}_F(\boldsymbol{x},-t) \le 0\right\} = B_{\mathrm{I}}(0,-t,X_s). \end{aligned} \quad (21)$$

Remark 2.3: The keypoint to compute BRSII is the introduction of $\underline{u}$. After that, while the augmented system always moves along the trajectory of the original system, sometimes it moves as fast as the original system, sometimes it stays in place, and sometimes it moves slower.

III. Backward Reachable Set Computation with Relaxed HJB Equation

It is difficult to solve the problems (11) and (20) analytically. Although [17] gives the numerical solution that depends on grids, we prefer the parameterized result sometimes. In this section, the final value problem is replaced by a mathematical optimization problem via relaxing the HJB equation. Further, bilinear formulation also makes the external ellipsoidal approximation of BRS possible.

A. Relax the HJB Equation

Before giving the relaxed optimization problem, let us first unify the notations. Differential constrains $\hat{\boldsymbol{f}}$ and target G are denoted as:

$$(\hat{\boldsymbol{f}},G) \triangleq \begin{cases} (\bar{\boldsymbol{f}},g), & \text{when BRSII is considered,} \\ (-\boldsymbol{f},h), & \text{when FRS is considered.} \end{cases}$$

Let $\hat{\boldsymbol{u}}$ and $\hat{U}$ represent the admissible control and admissible control space, respectively. Then

$$\hat{\boldsymbol{u}} \in \hat{U} \triangleq \begin{cases} [\underline{u},\ \boldsymbol{u}] \in [0,1] \times U, & \text{when BRSII is considered,} \\ \boldsymbol{u} \in U, & \text{when FRS is considered.} \end{cases}$$

So the relaxed optimization problem is defined as

$$\begin{aligned} & \max \iint V(\boldsymbol{x},t) \\ s.t \quad & \frac{\partial V(\boldsymbol{x},t)}{\partial t} + \frac{\partial V(\boldsymbol{x},t)}{\partial \boldsymbol{x}^T} \hat{\boldsymbol{f}} \geq 0, \\ & L(\boldsymbol{x}) G(\boldsymbol{x}) - V(\boldsymbol{x},0) \geq 0,\ L(\boldsymbol{x}) > 0, \\ & \forall \boldsymbol{x} \in R^n,\ \hat{\boldsymbol{u}} \in \hat{U},\ -T \leq t \leq 0. \end{aligned} \tag{22}$$

Theorem 3.1: (i) Any feasible solution of problem (22) implies the external approximation of BRS. (ii) $\bar{V}_B(\boldsymbol{x}, t)$ and $\bar{V}_F(\boldsymbol{x}, t)$ are the global optimal solution to problem (22), if $L(\boldsymbol{x})=1$, $\forall \boldsymbol{x} \in R^n$.

Proof: Let V be a feasible solution of problem (22).

(i) For any $\boldsymbol{x}_1 \in X$, $\hat{\boldsymbol{u}} \in \hat{U}$, there must be a unique trajectory under Assumption 2.2, and let the corresponding state of the trajectory at time 0 be $\boldsymbol{x}_2$. Integrate the time derivative of V for fixed $\hat{\boldsymbol{u}}$.

$$L(\boldsymbol{x}_2) G(\boldsymbol{x}_2) \geq V(\boldsymbol{x}_2,0) = V(\boldsymbol{x}_1,t) + \int_t^0 \frac{dV}{d\tau} d\tau \geq V(\boldsymbol{x}_1,t) \tag{23}$$

By (23),

$$\forall \hat{\boldsymbol{u}} \in \hat{U},\ -T \leq t \leq 0, \quad V(\boldsymbol{x}_1,t) > 0 \Rightarrow G(\boldsymbol{x}_2) > 0. \tag{24}$$

(24) means that

$$\{\boldsymbol{x} \in R^n \mid V(\boldsymbol{x},t) \leq 0\}.$$

is the external approximation of BRS at time t.

(ii) Let the admissible optimal control corresponding to $\bar{V}_B$ or $\bar{V}_F$ be $\hat{\boldsymbol{u}}^*$. For $\boldsymbol{x}_1$ and $\hat{\boldsymbol{u}}^*$, Let the corresponding state of the trajectory at time 0 be $\boldsymbol{x}_3$. Similarly, we have

$$V(\boldsymbol{x}_1,t) \leq V(\boldsymbol{x}_3,0) \leq L(\boldsymbol{x}_3) G(\boldsymbol{x}_3). \tag{25}$$

In addition,

$$\begin{aligned} & G(\boldsymbol{x}_3) = \bar{V}_B(\boldsymbol{x}_3,0) \ or\ \bar{V}_F(\boldsymbol{x}_3,0), \\ & \bar{V}_B(\boldsymbol{x}_3,0) = \bar{V}_B(\boldsymbol{x}_1,t),\ \bar{V}_F(\boldsymbol{x}_3,0) = \bar{V}_F(\boldsymbol{x}_1,t). \end{aligned} \tag{26}$$

(26) means that

$$V(\boldsymbol{x}_1,t) \leq L(\boldsymbol{x}_3) \bar{V}_B(\boldsymbol{x}_1,t) \ or\ L(\boldsymbol{x}_3) \bar{V}_F(\boldsymbol{x}_1,t). \tag{27}$$

Hence, for any feasible solution of problem (22), we have

$$\iint V(\boldsymbol{x}_1,t) \leq \iint L(\boldsymbol{x}_3) \bar{V}_B(\boldsymbol{x}_1,t) \ or\ \iint L(\boldsymbol{x}_3) \bar{V}_F(\boldsymbol{x}_1,t). \tag{28}$$

which implies that $\bar{V}_B(\mathbf{x}, t)$ and $\bar{V}_F(\mathbf{x}, t)$ are the global optimal solution to problem (22), if $L(\mathbf{x})=1$, $\forall \mathbf{x} \in R^n$.

Remark 3.1: (i) However, it is non-trivial to solve (22) because of the infinite dimensional inequality constrains. For ease of numerical implementation, this problem would be converted into a finite dimensional parameterized optimization problem by SOSP. (ii) In addition, the Stone-Weierstrass theorem states that every continuous function defined on a compact Hausdorff space can be uniformly approximated as closely as desired by a polynomial function. That is, the effective approximation for V might not be guaranteed on R^n. Then we need to restrict the approximation domain to X_{appr}.

B. *External ellipsoidal approximation of BRS*

Now, we are ready to propose a SOS optimization problem for computing the external ellipsoidal approximation of BRS based on HJB analysis. Let V be a polynomial function of $\boldsymbol{x}$ and t. Let

$$\begin{aligned} X_{appr}(t) &= \{\boldsymbol{x} \in R^n \mid \forall t \in [-T,0],\ r_x(\boldsymbol{x},t) \leq 0\}, \\ \hat{U} &= \{\hat{\boldsymbol{u}} \in R^m \ or\ [0,1] \times R^m \mid r_{\hat{u}}(\hat{\boldsymbol{u}}) \leq 0\}. \end{aligned}$$

where r_x and $r_{\hat{u}}$ are also polynomial functions. Then the following implications hold

$$\begin{aligned} & \forall r_x(\boldsymbol{x},t) \leq 0,\ r_{\hat{u}}(\hat{\boldsymbol{u}}) \leq 0,\ -T \leq t \leq 0, \\ & \frac{\partial V(\boldsymbol{x},t)}{\partial t} + \frac{\partial V(\boldsymbol{x},t)}{\partial \boldsymbol{x}^T} \hat{\boldsymbol{f}}\ is\ SOS \Rightarrow \frac{\partial V(\boldsymbol{x},t)}{\partial t} + \frac{\partial V(\boldsymbol{x},t)}{\partial \boldsymbol{x}^T} \hat{\boldsymbol{f}} \geq 0. \end{aligned} \tag{29}$$

$$\begin{aligned} & L(\boldsymbol{x}) - \alpha_1,\ L(\boldsymbol{x}) G(\boldsymbol{x}) - V(\boldsymbol{x},0)\ is\ SOS \Rightarrow \\ & \qquad L(\boldsymbol{x}) > 0,\ L(\boldsymbol{x}) G(\boldsymbol{x}) - V(\boldsymbol{x},0) \geq 0. \end{aligned} \tag{30}$$

here α_1 is a small positive constant.

Define discrete external ellipsoids of BRS as

$$\begin{aligned} \varepsilon(t_k) = \{\boldsymbol{x} \in R^n \mid & (\boldsymbol{x}-\boldsymbol{q}_k)^T S_k (\boldsymbol{x}-\boldsymbol{q}_k) \leq 1,\ S_k \geq 0\}, \\ & \forall k = 1,...,N,\ t_k \in [-T,0]\ and\ t_k \leq t_{k+1}. \end{aligned} \tag{31}$$

Then the containment constrain can be expressed as

$$\forall r_x(\boldsymbol{x},t_k) \leq 0,\ V(\boldsymbol{x},t_k) \leq 0 \Rightarrow (\boldsymbol{x}-\boldsymbol{q}_k)^T S_k (\boldsymbol{x}-\boldsymbol{q}_k) - 1 \leq 0. \tag{32}$$

Stronger formulation for (32) is

$$\forall r_x(\boldsymbol{x},t_k) \leq 0,\ V(\boldsymbol{x},t_k) \leq 0 \Rightarrow 1 - (\boldsymbol{x}-\boldsymbol{q}_k)^T S_k (\boldsymbol{x}-\boldsymbol{q}_k)\ is\ SOS. \tag{33}$$

Reformulating (29) and (33) by introducing Lagrange multipliers L_i, we can obtain the following optimization problem:

$$\begin{aligned} & \min \sum_{k=1}^{N} -\ln\det S_k \\ s.t \quad & \frac{\partial V(\boldsymbol{x},t)}{\partial t} + \frac{\partial V(\boldsymbol{x},t)}{\partial \boldsymbol{x}^T} \hat{\boldsymbol{f}} + L_1 r_{\hat{u}}(\hat{\boldsymbol{u}}) + \\ & \qquad L_2 r_x(\boldsymbol{x},t) + L_3 t(t+T)\ is\ SOS, \\ & -V(\boldsymbol{x},0) + L_4 G(\boldsymbol{x})\ is\ SOS,\ S_k \geq 0, \\ & 1 - (\boldsymbol{x}-\boldsymbol{q}_k)^T S_k (\boldsymbol{x}-\boldsymbol{q}_k) + L_{5k} V(\boldsymbol{x},t_k) + L_{6k} r_x(\boldsymbol{x},t_k)\ is\ SOS, \\ & L_1(\hat{\boldsymbol{u}}), L_2(\boldsymbol{x},t), L_3(\boldsymbol{x},t), L_4(\boldsymbol{x}) - \alpha_1, \\ & \qquad L_{5k}(\boldsymbol{x}), L_{6k}(\boldsymbol{x})\ is\ SOS, \quad \forall k = 1,2,...,N. \end{aligned} \tag{34}$$

Remark 3.2: (i) Notice that the states satisfying $V(\boldsymbol{x}, t) \leq 0$ do not necessarily belong to BRS, because they could go out of X_{appr}. But the states belonging to BRS must satisfy $V(\boldsymbol{x}, t) \leq 0$ under Assumption 2.4. (ii)Ideally, the centers $\boldsymbol{q}_k$ of the external ellipsoids should also be treated as decision variables, but their products with the shape matrixes make it challenging to solve (34). In this paper, they are set as corresponding states of a fitting polynomial uncontrolled trajectory $\boldsymbol{q}(t)$ with the center of $G(\boldsymbol{x})$ as the initial state. (iii)The objective function of problem (34) represents the sum of measures of the external ellipsoids. It is equivalent to the one in (22) and is convex. (iv)When V is fixed, the problem (34) is a convex optimization problem with respect to the remaining decision variables, and so is when L_{5k} is fixed. To solve this kind of bilinear problem iteratively [Algorithm 1, 21], we need to provide an initial guess, which will be discussed in III.C.

C. Initial Guess

Consider the hypersphere

$$V_{ini}(\boldsymbol{x},t)=(\boldsymbol{x}-\boldsymbol{q}(t))^T(\boldsymbol{x}-\boldsymbol{q}(t))-p(t). \tag{35}$$

in which $p(t)$ is a positive polynomial function within $[-T, 0]$.

Let $p^*(t)$ be the global optimal solution of the following convex optimization problem:

$$\begin{aligned}&\min \int_{-T}^{0} p(t)\,dt\\ &s.t\quad -\dot{p}+2(\boldsymbol{x}-\boldsymbol{q}(t))^T(\hat{\boldsymbol{f}}-\dot{\boldsymbol{q}})+L_7 r_{\hat{u}}(\hat{\boldsymbol{u}})+\\ &\qquad L_8 r_x(\boldsymbol{x},t)+L_9 t(t+T)\ is\ SOS,\\ &p-\alpha_2+L_{10}t(t+T)\ is\ SOS\ (\alpha_2\ is\ a\ postive\ constant),\\ &p(0)-(\boldsymbol{x}-\boldsymbol{q}(0))^T(\boldsymbol{x}-\boldsymbol{q}(0))+L_{11}G(\boldsymbol{x})\ is\ SOS,\\ &L_7(\hat{\boldsymbol{u}}),L_8(\boldsymbol{x},t),L_9(\boldsymbol{x},t),L_{10}(t),L_{11}(\boldsymbol{x})-\alpha_1\ is\ SOS.\end{aligned} \tag{36}$$

Lemma 3.1: Let M be a given symmetric positive definite matrix, and $\varphi(t)$ be a given positive polynomial in $[-T, 0]$. Then

$$V^*_{ini}(\boldsymbol{x},t)=(\boldsymbol{x}-\boldsymbol{q}(t))^T(\boldsymbol{x}-\boldsymbol{q}(t))-p^*(t).$$

is a feasible solution of (34), when $r_x(\boldsymbol{x},t)$ can be expressed as $(\boldsymbol{x}-\boldsymbol{q}(t))^T M(\boldsymbol{x}-\boldsymbol{q}(t))-\varphi(t)$.

Proof: Obviously, $V^*_{ini}(\boldsymbol{x},t)$ satisfies the first two SOS constrains of (34). Let $\boldsymbol{I}_{n\times n}$ be identity matrix and take

$$L_{5k}=1/2p^*(t_k)\in(0,\ 1/\alpha_2],\ \ L_{6k}=1/2\varphi(t_k)>0,$$
$$\forall\beta_k\ge 0,\ S_k=\frac{M}{2\varphi(t_k)+\beta_k}+\frac{1}{2p^*(t_k)+\beta_k}\boldsymbol{I}_{n\times n}\ge 0.$$

Because of $\boldsymbol{q}(t_k)=\boldsymbol{q}_k$, then

$$\begin{aligned}&1-(\boldsymbol{x}-\boldsymbol{q}_k)^T S_k(\boldsymbol{x}-\boldsymbol{q}_k)+L_{5k}V^*_{ini}(\boldsymbol{x},t_k)+L_{6k}r_x(\boldsymbol{x},t_k)\\ &=\frac{\beta_k}{2\varphi(t_k)(2\varphi(t_k)+\beta_k)}(\boldsymbol{x}-\boldsymbol{q}_k)^T M(\boldsymbol{x}-\boldsymbol{q}_k)+\\ &\frac{\beta_k}{2p^*(t_k)(2p^*(t_k)+\beta_k)}(\boldsymbol{x}-\boldsymbol{q}_k)^T(\boldsymbol{x}-\boldsymbol{q}_k)\ is\ SOS,\ \forall k=1,\ldots,N.\end{aligned} \tag{37}$$

IV. A Reachability-Based Spatio-Temporal Sampling Strategy

A. Sampling Strategy

Lines 1-2 of Algorithm 1 first initialize the search tree. In lines 3-5, a minimum BRSII containing the initial state $\boldsymbol{x}_s$ is found. In other words, the optimal time cost without considering obstacles has been determined. So next we restrict the search domain to the MTIS corresponding to this time cost (Algorithm 2, line 3). If the algorithm finds a better suboptimal trajectory (Algorithm 2, lines 6-8) after n_f iterations, then the current cost is reset, and **indicator** is set to true to call **Prune** to prune the nodes that are outside $\Omega_M(cost)$ (Algorithm 1, line 13). Conversely, if the time cost of the suboptimal trajectory found is greater than the current cost, only the current cost is reset (Algorithm 2, lines 9-10). If no feasible trajectory is found after n_f iterations, it may imply that the optimal time of problem (3) is increased due to the existence of obstacles, and the search domain should also be enlarged (Algorithm 1, lines 10-12). In addition, the search tree ***Tree*** needs to be retained to reuse valid information for the latter two cases when **indicator** is false. **SBMP** in line 4 of Algorithm 2 represents any sample-based kinodynamic motion planner. It takes the current search tree and a new sample as input, grows the tree within the MTIS, and outputs a suboptimal trajectory whose time cost is not greater than the current one, if the suboptimal trajectory exists.

Algorithm 1: Sampling Strategy

```
1  τ = 0; Q = {x_s}; E = {};
2  Initialize Tree(Q, E);
3  while x_s ∉ B_II(0, τ, X_g) do
4      τ = τ − δ_1 (δ_1 is a positive parameter);
5  end while
6  cost_pre = ∞; cost = -τ;
7  while |(cost_pre - cost) / cost_pre| > eps do
8      cost_pre = cost;
9      [Tree, cost, cost_new, indicator] = Find_trajectory(Tree, cost);
10     if cost_new = ∞ do
11         cost = cost_pre + δ_2 (δ_2 is a positive parameter);
12     end if
13     Tree = Prune(Tree, cost, indicator);
14 end while
```

Algorithm 2: Find_trajectory(*Tree*, *cost*)

```
1  indicator = false;
2  for i = 1 : n_f
3      [x_new, t] = Generate_Sample(cost);
4      [Tree, cost_new] = SBMP(Tree, [x_new, t]);
5  end for
6  if cost_new < cost
7      cost = cost_new;
8      indicator = true;
9  else if cost_new ≠ ∞ and cost_new > cost
10     cost = cost_new;
11 end if
```

Algorithm 3 describes a procedure to generate a new sample $[\boldsymbol{x}_{new}, t]$ in Ω_M. Since it is difficult to sample directly in MTIS, we still adopt a similar approach as [15] here. First, uniformly sample a time t in $[0, cost]$ (Algorithm 3, line 1). Then we compare the measures of F_e and B_{IIe}, sample directly and uniformly in the side with the smaller measure to generate $\boldsymbol{x}_{rand}$, and use the other side to determine whether to accept $\boldsymbol{x}_{rand}$ (Algorithm 3, lines 2-14). If the desired sample is not obtained after n_g attempts. Then the algorithm will uniformly sample in X (Algorithm 3, line 15).

Algorithm 3: Generate_Sample(*cost*)

```
1  t = sample([0, cost]);
2  for i = 1 : n_g do
3      if λ(F_e([t], 0, X_s)) < λ(B_IIe([cost - t], 0, X_g))
4          x_rand = sample(F_e([t], 0, X_s));
5          if x_rand ∈ sample(B_IIe([cost - t], 0, X_g))
6              x_new = x_rand ;
7              return [x_new, t];
8      else
9          x_rand = sample(B_IIe([cost - t], 0, X_g));
10         if x_rand ∈ F_e([t], 0, X_s)
11             x_new = x_rand ;
12             return [x_new, t];
13     end if
14 end for
15 x_new = sample(X) ;
16 return [x_new, t] ;
```

Remark 4.1: (i) The FRS corresponding to time t might not be found in the stored discrete reachable sets, e.g. $t\in(-t_{k+1}, -t_k)$, so we replace it with $F_e([t], 0, \boldsymbol{x}_s)$, which represents $F_e(-t_k, 0, \boldsymbol{x}_s)$. As long as N is big enough, this will not cause significant errors. The same treatment is also applied to BRSII, except that lemma 2.2 and theorem 2.1 ensure the rationality. (ii) Since MTIS is a set with time stamp, time information should be token into account when generating samples and using distance metric (see Fig.1 for an illustration). This is

especially useful when dealing with motion planning problems involving dynamic obstacles.

B. Exploit the Invariances in the Differential Constrains

For system (1), lemma 2.1 states that the shape of the reachable set is independent of time, but unfortunately, it is dependent on the target set G. In other words, when G changes, we generally have to recalculate FRS and BRSII. Given the computational complexity, this seems to be a claim that the method cannot be applied in real-time scenarios. However, in fact, many systems have only partial state components in their differential constraints $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})$. For exam- ple, the dynamics of a mobile robot or a quadrotor does not depend on x-y-z position. We denote the components not included in $\boldsymbol{f}$ as $\boldsymbol{x}_c$, and the remaining components as $\boldsymbol{x}_{nc}$. According to the coordinate transformation similar to lemma 2.1, changing $\boldsymbol{x}_c$ does not affect the shape of the reachable set, but merely shifts it along the $\boldsymbol{x}_c$ coordinates, namely the invariance of dynamics [Section 5.2, 21]. Moreover, if $\boldsymbol{x}_{nc}$ belongs to a compact set X_{nc}, e.g. the velocity tends to be set within a certain range, then this compact set can be covered by a finite number of closed subsets X_{nc}^i, which enables us to compute and save the FRS and BRSII of these target sets $\{\boldsymbol{x} \mid \boldsymbol{x}_c = \boldsymbol{0}, \boldsymbol{x}_{nc} \in X_{nc}^i\}$ offline. For a certain state $\boldsymbol{x} = (\boldsymbol{x}_c, \boldsymbol{x}_{nc})$, the trouble of calculating reachable sets in real-time tasks could be solved by finding the reachable set saved offline corresponding to $\boldsymbol{x}_{nc}$ and translating it along the $\boldsymbol{x}_c$ coordinates.

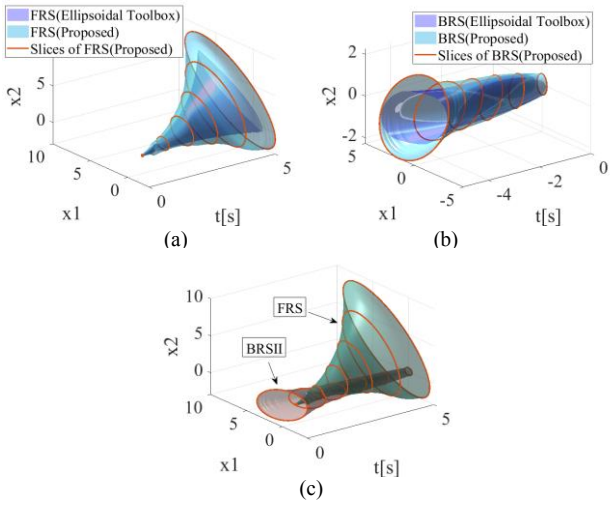


Figure 2. The calculation results of forward reachable set (a), backward reachable set II (b) and modified time informed set (c) for 2D linear system

V. Numerical Simulation

In this section, we take TIE [15] and SST [12] strategies as benchmark to evaluate the performance of the proposed Reachability-based-SST algorithm with spatio-temporal sampling strategy proposed in this paper. All tests were performed in Matlab 2021a using a 64-bit laptop with an AMD-R5800H processor and 16GB RAM.

A. 2D Linear System

Consider the following 2D linear system [15] :

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}0 & 0.5\\ -0.1 & 0.2\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+\begin{bmatrix}0\\ 1\end{bmatrix}u. \tag{38}$$

with $X_s=\varepsilon([-2, 1]^T, 100\boldsymbol{I}_{2\times2})$, $X_g=\varepsilon([0, 0]^T, 4\boldsymbol{I}_{2\times2})$, $u\in[-0.5, 0.5]$. The FRS and BRSII of the system are calculated by using ET and the proposed method respectively, and the results are shown in Fig.2. Since the result calculated by ET is based on 30 random directions, it can be seen that our method does provide a relatively compact external ellipsoidal approximation of the reachable set by comparison. Further, Fig.2 (c) shows the corresponding MTIS when t=5s (i.e. the intersection of FRS and BRSII).

Next, we test these algorithms in an environment containing some rectangular obstacles. Set δ_{BN}=0.2, δ_s=0.1 [15]. For Reachability-based-SST, set δ_1=0.25s, δ_2=0.5s, n_f=500, n_g=10. The statistical results are illustrated in Fig.3, and the experimental data are summarized in Table I. T_{ini} and T_{fin} denote the time when the initial solution trajectory is found and the time when the algorithm terminates, respectively. C_{ini} and C_{fin} represent the cost of the initial solution trajectory and the cost of the final suboptimal solution trajectory respectively. N_{node} is the number of nodes on the tree at the end of the algorithms. Fig.4 shows the suboptimal trajectories found by the three methods.

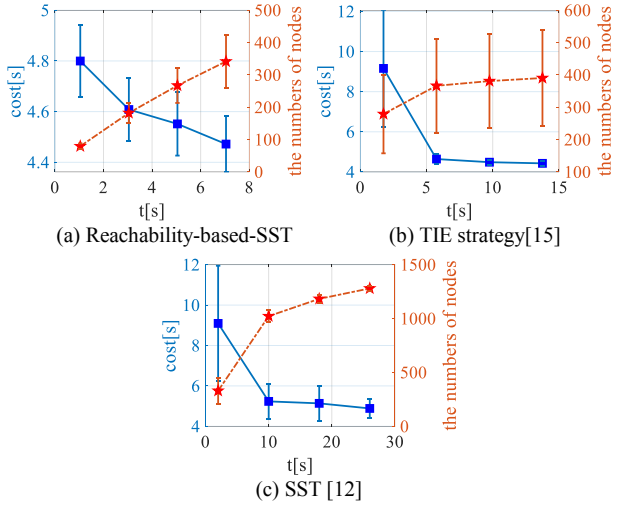


Figure 3. The statistical results of the 100 numerical experiments for 2D linear system. The blue square represents the cost of the solution trajectory, and the red pentagram represents the number of nodes in the tree.

TABLE I. Comparison of the Baseline Methods and the Proposed Method

		T_{ini}[s]	C_{ini} [s]	T_{fin}[s]	C_{fin} [s]	N_{node} ($\times10^2$)
2D-L	Ours	1.04	4.80±0.14	7.04	4.47±0.11	3.42±0.82
	TIE	1.75	9.16±2.9	13.75	4.43±0.13	3.91±1.47
	SST	1.81	9.09±2.85	25.81	4.89±0.49	12.8±0.25
3D-NL	Ours	1.70	4.30±0.24	4.70	4.08±0.09	1.71±0.27
	SST	26.44	6.39±2.09	56.44	5.62±1.41	28.48±7.82
6D-L	Ours	2.85	4.31±0.27	8.85	4.17±0.23	2.84±1.11
	TIE	3.65	14.4±3.56	33.65	4.98±1.02	78.93±34.4
	SST	4.02	14.4±3.56	64.02	13.8±3.43	325.1±13.9

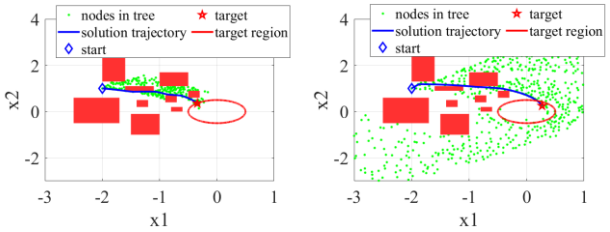
B. 3D Nonlinear System

Consider the following nonlinear system:

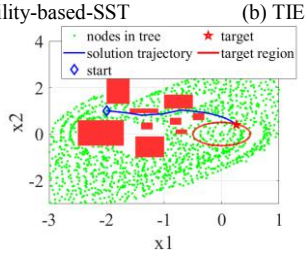
$$\dot{x}_1 = x_3,\ \ \dot{x}_2 = -x_1 + \frac{1}{6}x_1^3 - x_3,\ \ \dot{x}_3 = u \tag{39}$$

with $X_s=\varepsilon([1, 0.5, -0.2]^T, 100\boldsymbol{I}_{2\times2})$, $X_g=\varepsilon([0, 0, 0]^T, 100\boldsymbol{I}_{2\times2})$, $u\in[-0.2, 0.2]$. Since the TIE strategy cannot be used for

nonlinear systems, we only test Reachability-based-SST and SST in an environment containing some rectangular obstacles. For SST, set δ_{BN}=0.1, δ_s=0.05. For Reachability-based-SST, set δ_{BN}=0.2, δ_s=0.1, δ_1=0.25s, δ_2=0.5s, n_f=500, n_g=10. The statistical results are illustrated in Fig.5, and the experimental data are summarized in Table I. Fig.6 shows the suboptimal trajectories found by the two methods.

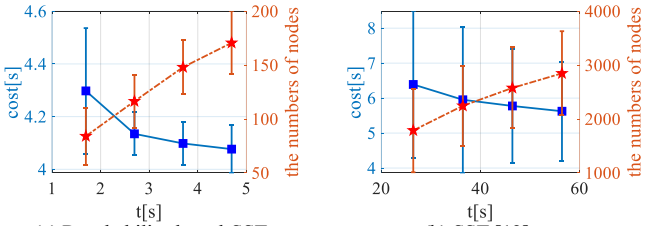


(a) Reachability-based-SST (b) TIE strategy [15]



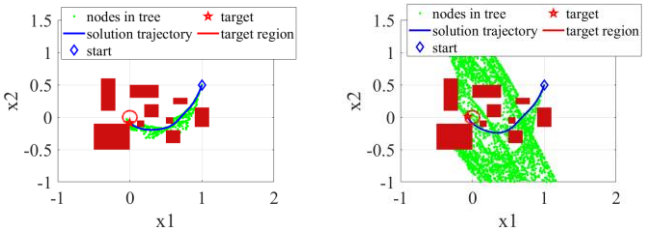
(c) SST [12]

Figure 4. The suboptimal solution trajectory of 2D linear system, when the seed of the random number generator is 1.



(a) Reachability-based-SST (b) SST [12]

Figure 5. The statistical results of the 100 numerical experiments for 3D nonlinear system. The blue square represents the cost of the solution trajectory, and the red pentagram represents the number of nodes in the tree.



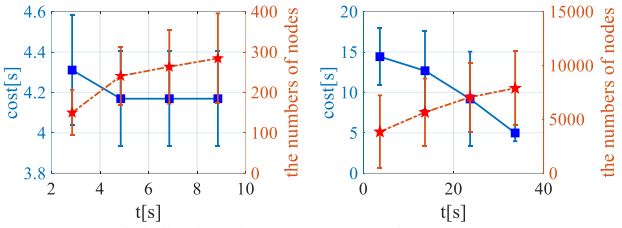
(a) Reachability-based-SST (b) SST [12]

Figure 6. The suboptimal solution trajectory of 3D nonlinear system, when the seed of the random number generator is 1.

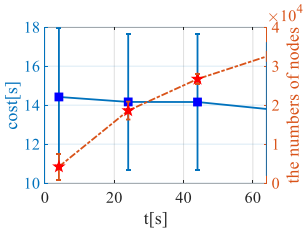
C. 6D Double Integrator System

Consider the common 6D double integrator planning model with X_s=ε([-7, 8, 2, 2, -2, -1]^T, 16$\boldsymbol{I}_{2\times2}$), X_g=ε($\boldsymbol{0}_{6\times1}$, 2/3$\boldsymbol{I}_{2\times2}$), u_1, u_2, u_3 ∈ [-1, 1]. Set δ_{BN}=0.5, δ_s=0.25. For Reachability-based-SST, set δ_1=0.25s, δ_2=0.5s, n_f=1500, n_g=10. The statistical results are illustrated in Fig.7, and the experimental data are summarized in Table I. Fig.8 shows the suboptimal trajectories found by the three methods.

Since the triggering of TIE strategy depends on SST to provide it with an initial solution trajectory, T_{ini} is almost the same for both methods. However, the proposed algorithm uses MTIS to restrict the sampling domain from the beginning, so it can find the initial solution trajectory more quickly. Meanwhile, C_{ini} is reduced by 33% to 70%.

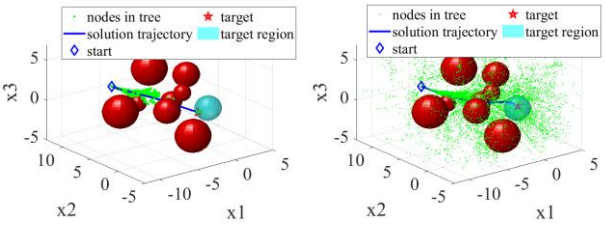


(a) Reachability-based-SST (b) TIE strategy [15]

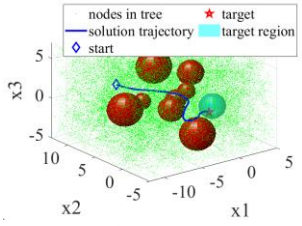


(c) SST [12]

Figure 7. The statistical results of the 100 numerical experiments for 6D linear system. The blue square represents the cost of the solution trajectory, and the red pentagram represents the number of nodes in the tree.



(a) Reachability-based-SST (b) TIE strategy [15]



(c) SST [12]

Figure 8. The suboptimal solution trajectory of 6D linear system, when the seed of the random number generator is 1

In terms of convergence rate, since the proposed method replaces BRSI with BRSII and can find the initial solution trajectory faster, it can generally obtain better suboptimal trajectory with shorter running time. The exception is the 2D linear system, which achieves a 1.4% lower C_{fin} with TIE strategy than with Reachability-based-SST. But it should also be noted that this comes at the cost of nearly doubling the running time.

From the perspective of reducing memory requirement, Reachability-based-SST retains fewer nodes on the search tree than TIE strategy and SST. This is more obvious in the experimental nonlinear system and high-dimensional system. For example, for the 6D double integrator,

Reachability-based-SST reduces the number of nodes by 96.4% and 99.13%, respectively, compared with the other two methods. There are two reasons for this effect. One is that we use the **Prune** module [Algorithm 1, line 13] to prune the redundant nodes outside the MTIS in time, and the other is that MTIS guides the planner to sample only near the optimal trajectory.

Remark 5.1: In this section, we demonstrate the effectiveness of the sampling strategy proposed in Section IV on several linear and nonlinear systems. Although our work and [17] both use HJB analysis to calculate reachable sets, the latter divides the state space into grids, making its efficiency greatly affected by the dimensionality. Instead, this paper chooses to transform the infinite dimensional optimization problem into the finite dimensional parametric optimization problem to alleviate the computational burden caused by the dimensions. However, a subsequent disadvantage is that the selection of X_{appr} has a certain influence on whether the reachable set can be computed successfully and the quality of the obtained reachable set. In the previous experiments, ET [16] and the method in [17] were used to estimate X_{appr} for linear and nonlinear systems, respectively. However, since the method in [17] is difficult to be applied to high-dimensional systems, how to provide reliable X_{appr} for high-dimensional nonlinear systems is still a remaining problem, which will undoubtedly be one of the focuses of future work.

VI. Conclusion

Based on the fact that the optimal trajectory occupies only a small part of the entire state space, MTIS is constructed via HJB reachability analysis and SOSP, thus reducing the sampling domain of the SBMP. Besides, by pruning the redundant nodes outside MTIS and reusing the valid information in the search tree, the reachability-based spatio-temporal sampling strategy requires significantly less computation time and memory than the original [12] and improved [15] algorithm. Nevertheless, we must point out that it is almost impossible to use this method as a global planner for long range outdoor navigation tasks, due to the difficulty of calculating reachable set in a large time horizon. But the authors still believe it will play a positive role in local planners used in conjunction with other global planners (e.g. learning-based planner [22]).

References

[1] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, " Probabilistic roadmaps for path planning in high-dimensional configuration spaces," IEEE Trans. Robot. Automat., vol. 12, no. 4, pp. 566–580, August. 1996.

[2] J. J. Kuffner, S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in Proc. IEEE intl. Conf. Robot. Autom., San Francisco, CA, Apr. 2000, pp. 995-1001.

[3] S. Karaman, E. Frazzoli, "Sampling-based algorithms for optimal motion planning," Int. J. Robot. Res., vol. 30, no. 7, pp. 846-894, Jun. 2011.

[4] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the RRT," in Proc. IEEE intl. Conf. Robot. Autom., Shanghai, China, May 2011, pp. 1478-1483.

[5] B. Akgun and M. Stilman, "Sampling heuristics for optimal motion planning in high dimensions," in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., San Francisco, CA, Sep. 2011, pp. 2640–2645.

[6] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Chicago, IL, Sep. 2014, pp. 2997–3004.

[7] 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 Proc. IEEE Int. Conf. Robot. Automat., Seattle, WA, May 2015, pp. 3067–3074.

[8] J. D. Gammell, T. D. Barfoot, and S. S. Srinivasa, "Informed sampling for asymptotically optimal path planning," IEEE Trans. Robot., vol. 34, no. 4, pp. 966–984, Aug. 2018.

[9] D. J. Webb and J. Van DenBerg, "Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics," in Proc. IEEE Int. Conf. Robot. Automat., Karlsruhe, Germany, 2013, pp. 5054–5061.

[10] A. Perez, R. Platt, G.Konidaris, L. Kaelbling, and T. Lozano-Perez, "LQRRRT*: Optimal sampling-based motion planning with automatically derived extension heuristics," in Proc. IEEE Int. Conf. Robot. Automat., Saint Paul, MN, 2012, pp. 2537–2542.

[11] A. Wu, S. Sadraddini, R. Tedrake. "R3T: Rapidly-exploring random reachable set tree for optimal kinodynamic planning of nonlinear hybrid systems," in Proc. IEEE Int. Conf. Robot. Automat., Paris, France, May 2020, pp. 7063–7070.

[12] Y. Li, Z. Littlefield, and K. E. Bekris. "Asymptotically optimal sampling-based kinodynamic planning," Int. J. Robot. Res., vol. 35, no. 5, pp. 528-564, 2016.

[13] T. Kunz, A. Thomaz, and H. Christensen, "Hierarchical rejection sampling for informed kinodynamic planning in high-dimensional spaces," in Proc. IEEE Int. Conf. Robot. Automat., Stockholm, Sweden, May 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 Proc. IEEE Int. Conf. Robot. Automat., Brisbane, Australia, May 2018, pp. 7063–7070.

[15] S. S. Joshi, S. Hutchinson, P. Tsiotras, "TIE: Time-Informed Exploration for Robot Motion Planning," IEEE Robot. Automat. Lett., vol. 6, no. 2, pp. 3585-3591, Apr. 2021.

[16] A. Kurzhanskiy and P. Varaiya, "Ellipsoidal techniques for reachability analysis," in Proc. Int.Workshop Hybrid Syst.: Comput. Control, Springer, pp. 202–214, 2000.

[17] I. M. Mitchell, A. M. Bayen, C. J. Tomlin, "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," IEEE Trans. Autom. Control, vol. 50, no. 7, pp. 947-957, July. 2005.

[18] P. A. Parrilo, "Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization," Ph.D. dissertation, California Institute of Technology, 2000.

[19] J. Lofberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in Proc. IEEE Int.Conf. Robot. Automat., 2004, pp. 284-289.

[20] A. Shkolnik, M. Walter, and R. Tedrake, "Reachability-guided sampling for planning under differential constraints," in Proc. IEEE Int.Conf. Robot. Automat., Kobe, Japan, May 2009, pp. 2859–2865.

[21] A. Majumdar, R. Tedrake, "Funnel libraries for real-time robust feedback motion planning," Int. J. Robot. Res., vol. 36, no. 8, pp. 947–982, 2017.

[22] B. Ichter, J. Harrison, and M. Pavone, "Learning sampling distributions for robot motion planning," in Proc. IEEE Int. Conf. Robot. Automat., Brisbane, Australia, May 2018, pp. 7087–7094.