THAMP-3D: Tangent-based Hybrid A* Motion Planning for tethered robots in sloped 3D terrains

Rahul Kumar, Vishnu S. Chipade, Sze Zheng Yong

Abstract— **This paper introduces a novel motion planning algorithm designed for a team of curvature-constrained tethered robots operating on sloped 3D terrains. Our approach addresses the critical issues of tether-terrain interaction, robot stability, and tether entanglement avoidance. The study focuses on a two-robot system, where stability is primarily dependent on tether tension, which is in turn limited by wheel traction. We propose a path-planning method that strategically utilizes terrain features (e.g., rocks) to augment tether tension through additional friction, thereby enhancing overall system stability. Our algorithm employs a modified tangent graph as the underlying structure for a hybrid A* search, incorporating stability constraints throughout the planning process. The proposed method is extensively evaluated through various simulation experiments, demonstrating its effectiveness in planning safe and efficient paths.**

I. Introduction

The quest for scientific knowledge in planetary exploration has increasingly led researchers to focus on extreme terrains that contains crucial geological clues. Areas of high scientific interest, such as steep slopes, deep valleys, and rugged terrains [1], [2] often contain valuable information about planetary geology, climate history, and potential for supporting life. However, these environments pose substantial risks and accessibility issues for traditional rover systems [3]–[5].

Tethered robots have emerged as a promising solution for navigating these challenging terrains. By leveraging the stabilizing force provided by their tethers, these systems can navigate inclines far beyond the capabilities of their untethered counterparts [6]. Tethered robots such as Axel [7], TReX [8], DuAxel [9] have demonstrated the feasibility of tethered exploration in simulated extreme conditions, showcasing their ability to descend steep crater walls and navigate highly challenging terrains. The efficacy of tethered robots in extreme terrain exploration depends on generating adequate tether tension. However, achieving the desired tension is not always straightforward, particularly in unpredictable planetary environments. To overcome this challenge, our research proposes an innovative approach, leveraging the terrain itself to enhance tether functionality. By strategically winding the tether around terrain features such as rock formations or surface irregularities, we can induce the capstan effect, e.g., [10], thereby amplifying the available tension through friction. Although existing research has addressed motion planning for both single [11]–[13] and multiple tethered robots [14], [15], much of this work has concentrated primarily on preventing tether entanglement, with insufficient attention given to optimizing tether tension and its role in navigation.

The authors in [12], [16] discuss using a homotopy-augmented graph to capture information about the homotopy class of the tether. This approach allows planning traversable optimal trajectories in cluttered environments while avoiding entanglement. Authors in [13] presents a visibility graph-based approach for planning optimal paths for a polygonal robot tethered to a fixed base point. They preprocess the workspace to efficiently answer queries about the shortest path between given start and goal configurations. These works does not explicitly consider the tension on the tether or its configuration in terms of stability during motion. Our work is inspired from [17], which introduces a traversability analysis and path planning algorithm for rappelling rovers operating on steep terrains. The method takes into account terrain-tether interaction, focusing on the rover's stability and reachability. It uses a sampling-based planner in 3D space to evaluate paths, considering how tether-terrain interactions affect stability. However, it does not account for specific tether tension limits, assuming that terrain-tether interactions alone can stabilize the rover.

In this paper, we present an innovative approach to motion planning for a team of two tethered robots navigating sloped 3D terrains, such as those encountered in planetary exploration. Our method considers tether-terrain interactions and determines stability based on tether configuration and tension. Our work builds upon and extends two research work—path planning in 2D environments using tangent graphs and hybrid A* search [18] and stability analysis for tethered robots on inclined surfaces [19]. We consider a scenario where Robot 1 operates on a relatively flat terrain at higher elevations, while Robot 2 descends steep slopes. Unlike traditional approaches that rely on fixed anchor points, our method leverages Robot 1's traction and strategic tether-terrain interactions to support Robot 2's weight during descent. Our approach utilizes the tangent graph generated from

R. Kumar and S.Z. Yong are with the Department of Mechanical and Industrial Engineering, Northeastern University, Boston, MA, USA (Email: {kumar.rahul4,s.yong}@northeastern.edu), while V. Chipade is an independent researcher (vishnuc@umich.edu). This work was supported in part by an Early Career Faculty grant 80NSSC21K0071 from NASA's Space Technology Research Grants Program.

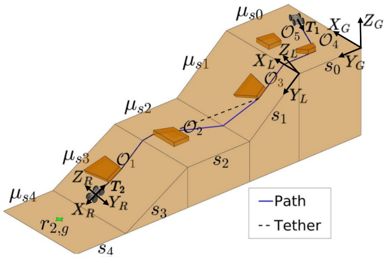


Fig. 1: Illustration of a multi-sloped terrain as well as a robot path and the corresponding tether configuration. (X_G, Y_G, Z_G) form global frame, (X_L, Y_L, Z_L) form surface local frame and (X_R, Y_R, Z_R) form robots' body frame.

[18] to define the potential search space, then applies stability analysis to ensure the identified paths are stable. One novel aspect of our work is the constraint that tether anchoring is only allowed on obstacles that are taller than the tether's height from the ground, ensuring feasible and reliable anchoring points. The key contributions of this research are:

i. Introducing a 3D motion planning approach that leverages 2D tangent graphs, combined with real-time stability analysis involving gravity and tether tension considerations;
ii. Developing a model for tether-terrain interactions that incorporates realistic anchoring constraints;
iii. Designing a comprehensive motion planning algorithm for curvature-constrained robot paths that optimally balances path planning, tangle-free tether configuration, and robot stability, effectively addressing the challenges faced by tethered multi-robot systems in extreme environments.

II. Modeling and Problem Statement

A. Notation

Let $\|\cdot\|$ denote the Euclidean norm, $|\cdot|$ represent the absolute value of a scalar or the cardinality of a set, and $conv(\cdot)$ be the convex hull operator. We define $\mathbf{g}$ as the gravity vector with magnitude $g = \|\mathbf{g}\|$.

B. Environment Model

We consider a structured 3D environment $\mathcal{W}$, modeled as a 2D manifold embedded in $\mathbb{R}^3$, inspired from [6], [20], to simplify the mathematical analysis. The environment consists of (cf. Fig. 1): 1) Flat and inclined rectangular ground surfaces $\mathcal{S}_\ell = conv\left(\{\mathbf{r}^1_{s\ell}, \mathbf{r}^2_{s\ell}, ..., \mathbf{r}^4_{s\ell}\}\right)$, for $\ell \in I_s = 0, 1, 2, ..., N_s$, where adjacent surfaces share one common edge. The coefficient of friction on surface $\mathcal{S}_\ell$ is $\mu_{s\ell}$, and its unit normal vector is $\mathbf{n}_{s\ell}$. The rotation matrix of the surface local coordinate frame relative to the global frame is denoted by $R_{s\ell}$. 2) Obstacles $\mathcal{O}_k$, for $k \in I_o = \{1, 2, ..., N_o\}$, defined as tuples $\mathcal{O}_k = (\mathcal{O}^b_k, s_{ok}, h_{ok}, \mu_{ok})$, where $\mathcal{O}^b_k = conv\left(\{\mathbf{r}^{b,1}_{ok}, \mathbf{r}^{b,2}_{ok}, ..., \mathbf{r}^{b,N_{v,ok}}_{ok}\}\right)$ is the polygonal base, s_{ok} is the surface index on which $\mathcal{O}^b_k$ lies, h_{ok} is the obstacle height, and μ_{ok} is the friction coefficient of obstacle's surface. The vertices of the obstacle base $\mathcal{O}^b_k$ are given by $\mathbf{r}^{b,\dagger}_{ok} = [x^{b,\dagger}_{ok}, y^{b,\dagger}_{ok}, z^{b,\dagger}_{ok}]^T$, for $\dagger \in \{1, 2, ..., N_{bv,ok}\}$, where $N_{bv,ok}$ is the total number of vertices. The normal to the obstacle surface at vertex $\dagger$ is denoted by $\mathbf{n}_{b\dagger,ok}$, and the boundary of obstacle $\mathcal{O}_k$ is represented by $\partial\mathcal{O}_k$.

C. Robot Model

Our system comprises a pair of advanced all-terrain robots, inspired by the TRex design [8], designated as Robot 1 and Robot 2. These robots are interconnected via a tether, attached at a height h_t above the ground surface. Each robot has a body-fixed coordinate system: the x-axis aligns with the robot's forward direction, the z-axis points upward, and the y-axis extends to the robot's left. The robots, with respective masses m_1 and m_2, are positioned in 3D space with coordinates $\mathbf{p}_1 = [x_1, y_1, z_1]^\top$ and $\mathbf{p}_2 = [x_2, y_2, z_2]^\top$ and have heading angles θ_1 and θ_2, which are the angles the x-axis of the robot's body frame makes with the x-axis of the surface local frame of the surface on which the robot is currently located. The robots feature a circular footprint with radius ρ_r.

The motion of the robots is constrained to the 2D manifold $\mathcal{W}$ and is modeled using the kinematics model as in Eq. (1)

$$\dot{\mathbf{r}}_i = \begin{bmatrix}\dot{\mathbf{p}}_i & \dot{\theta}_i\end{bmatrix}^\top = \begin{bmatrix}R(\mathbf{p}_i)\mathbf{v}_i & \omega\end{bmatrix}^\top \tag{1}$$

where for each $i \in \{1, 2\}$, $\mathbf{v}_i = [v\cos(\theta_i), v\sin(\theta_i), 0]^\top$ with v being the constant linear speed, and the rotation matrix $R(\mathbf{p}_i)$ is equal to rotation matrix $R_{s\ell}$ corresponding to the surface $\mathcal{S}_\ell$ on which the position $\mathbf{p}_i$ lies, and ω_i is the angular speed, bounded by $\bar{\omega}$. This constraint imposes a minimum turning radius of $\underline{\rho}_{turn} = \frac{v}{\bar{\omega}}$.

To ensure safe navigation, we employ an obstacle inflation technique inspired by [21]. The boundaries $\partial\mathcal{O}_k$ of obstacles are expanded by $\rho_{\bar{o}}$ (cf. Fig. 2), where $\rho_{\bar{o}} > \max(\rho_r, \underline{\rho}_{turn})$. This inflation accounts for the robots' physical size, safety margins, and turning constraints. The resulting inflated obstacles, denoted as $\bar{\mathcal{O}}_k$, are defined by: $\bar{\mathcal{O}}_k = \mathcal{O}_k \bigoplus \mathcal{B}(\rho_{\bar{o}})$ where $\bigoplus$ represents the Minkowski sum operation, and $\mathcal{B}(\rho_{\bar{o}})$ is a ball with radius $\rho_{\bar{o}}$ centered at the origin.

D. Assumptions

To formulate our problem and develop our solution, we make the following key assumptions:

***Assumption** 1 (Tether Properties):* The tether connecting the robots possesses the following characteristics:

i. It can be both retracted and extended as needed.
ii. It maintains a taut configuration throughout the robots' motion.

***Assumption** 2 (Robot-Tether Interaction):* The robots are physically constrained from crossing over or under the tether at any point during their movement.

***Assumption** 3 (Terrain Interactions):* The tether-terrain interaction and robot motion are subject to the following conditions:

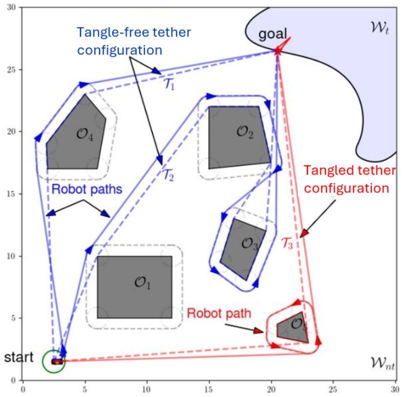


Fig. 2: Visualization of tangled and tangle-free tether configurations and curvature constraint in 2D [18], same idea is extended to 3D in our case.

i. Obstacle surfaces provide sufficient friction to prevent tether slippage once anchored.
ii. The tether can slide laterally on the ground surface at slope transition regions.
iii. Robots move in small, deliberate steps along paths that can be approximated by straight line segments, prioritizing stability[1].

E. Definitions

***Definition** 1 (Path):* A path $\mathcal{P} \in \mathcal{W}$ is defined as a sequence of path segments, $\mathcal{P} = \cup_{i=1}^{N_{ps}} \widetilde{\mathbf{r}_i\mathbf{r}_{i+1}}$, where each segment $\widetilde{\mathbf{r}_i\mathbf{r}_{i+1}} \in \mathcal{S}_\ell$ for some $\ell \in I_s$, represents a segment of straight line or a circular arc connecting consecutive waypoints $\mathbf{r}_i \in \mathcal{S}_\ell$ and $\mathbf{r}_{i+1} \in \mathcal{S}_\ell$. The total number of path segments is denoted by N_{ps}.

***Definition** 2 (Tether Configuration):* A tether configuration, $\mathcal{T}(\mathcal{P}_1, \mathcal{P}_2)$, is the set of tether segments resulting from the robots following their respective paths $\mathcal{P}_1$ and $\mathcal{P}_2$. It is defined as: $\mathcal{T}(\mathcal{P}_1, \mathcal{P}_2) := \cup_{j=1}^{N_a-1} \overline{\mathbf{a}_j\mathbf{a}_{j+1}}$, where each tether segment $\overline{\mathbf{a}_j\mathbf{a}_{j+1}}$ connects consecutive anchor points $\mathbf{a}_j$ along the tether, similar to path segments. The total number of anchor points is denoted by N_a. The anchor points are categorized into two types (cf. Fig. 1): i) Anchor points on the surfaces of obstacles, which are tall enough to allow tether anchoring, i.e., $h_{ok} > h_t$ for all $k \in I_o$, while obstacles with smaller heights are not considered for anchoring (but are still obstacles that the robots must avoid). ii) Anchor points at slope change locations on the ground surface. With Assumption 3, the slope anchor can slide but is important to consider as it changes the direction of tether tension.

***Definition** 3 (Tangle-Free Tether Configuration):* A tether configuration $\mathcal{T}(\mathcal{P})$ is said to be a tangle-free tether configuration if the robot can follow the path $\mathcal{P}$ without crossing the tether or entanglement (cf. Fig. 2).

***Definition** 4 (Winding Angle):* The winding angle ϕ_j is calculated as:

$$\phi_j = \begin{cases} \arccos\left(\frac{(\mathbf{a}_j i - \mathbf{a}_{j-1})\cdot(\mathbf{a}_{j+1} - \mathbf{a}_j)}{||\mathbf{a}_j - \mathbf{a}_{j-1}|| * ||\mathbf{a}_{j+1} - \mathbf{a}_j||}\right), & \text{if } 1 \leq j < N_a - 1, \\ 0, & \text{otherwise,} \end{cases}$$

where N_a is the total number of anchor points and ϕ_j is the winding angle at the j-th anchor point $\mathbf{a}_j = [x_{aj}, y_{aj}]^T$. This formula measures the angle between adjacent tether segments, assuming they are straight lines.

***Definition** 5 (Stable Path):* A path $\mathcal{P}$ is said to be statically stable if, for each point along the path, there exists a feasible solution to the linear program in Eq. 6 of [19], which verifies the balance of forces and moments.

For simplicity, we assume that Robot 1 operates exclusively on the flat surface $\mathcal{S}_0$, similar to [19]. To prevent sliding, the tension T_1 acting on Robot 1 must satisfy the following constraint:

$$T_1 \leq F_0 + \mu_{s0} m_1 g, \tag{2}$$

where F_0 represents the maximum counterforce Robot 1 can exert against the tether tension, μ_{s0} is the coefficient of friction on surface $\mathcal{S}_0$. Considering Robot 2 as the tether-pulling agent, the tension T_2 experienced by Robot 2 must adhere to:

$$T_2 \leq T_{2,max} = (F_0 + \mu_{s0} m_1 g) e^{\left(\sum_{j=1}^{N_a-1} \mu_{aj} |\phi_j|\right)}, \tag{3}$$

where exponential term is the result of tether tension amplification due to the capstan effect [10]. Here, μ_{aj} denotes the friction coefficient at the surface where anchor point $\mathbf{a}_j$ is located, and ϕ_j represents the winding angle at each anchor point.

F. Problem Statement

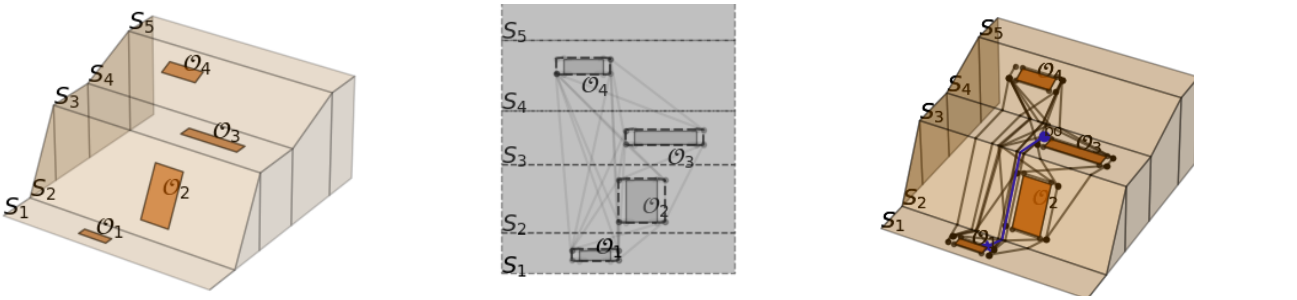
***Problem** 1:* Given the 3D environment $\mathcal{W}$, start positions $\mathbf{r}_{s1} \in \mathcal{S}_0$ and $\mathbf{r}_{s2} \in \mathcal{S}_0$ for Robot 1 and Robot 2 respectively, and initial tether configuration $\mathcal{T}_0$. Consider path $\mathcal{P}$ such that Robot 2 reaches its goal position $\mathbf{r}_{g2} \in \mathcal{S}_\ell$ for some $\ell > 0$, the objective is to design a motion planning algorithm to compute the shortest path $\mathcal{P}$ and the corresponding tether configuration $\mathcal{T}(\mathcal{P})$ such that the following conditions are satisfied:

i. The path $\mathcal{P}$ avoids obstacles and remains within the safe boundaries, i.e., $\mathcal{P} \not\subset \mathcal{O}' \triangleq \{\mathcal{O}_k \bigoplus \rho_{safe} \,|\, k \in I_o\}$, where $\rho_{safe} > 0$ is the safety distance from obstacles.
ii. The path $\mathcal{P}$ satisfies the curvature constraint imposed by the robot's kinematics.
iii. The path $\mathcal{P}$ ensures that the robot remains statically stable, preventing destabilizing movements such as sliding or tipping during traversal.
iv. The resulting tether configuration $\mathcal{T}(\mathcal{P})$ is tangle-free, meaning it does not result in tether entanglement or crossing.

III. Motion Planning Algorithm

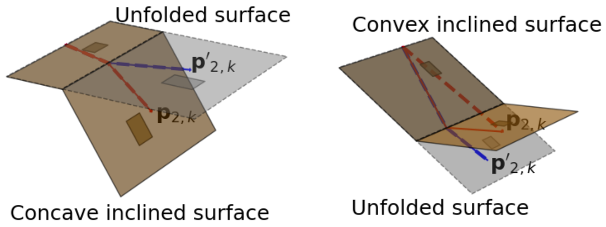
We present a novel approach to motion planning for tethered robots in sloped 3D terrains, as outlined in

[1] Following the approach in [17], we consider slow motions for maximum caution, focusing on quasi-static stability analysis.



(a) Structured 3D Environment $\mathcal{W}$ (b) $\mathcal{C}^1$-Tangent graph in unfolded $\mathcal{W}_{2D}$ (c) $\mathcal{C}^1$-Tangent graph folded back 3D $\mathcal{W}$

Fig. 3: Construction of $\mathcal{C}^1$-Tangent Graph for the structured 3D environment



(a) Unfolding concave surface (b) Unfolding convex surface

Fig. 4: Unfolding the terrain surfaces [19] (Solid red line: (Shortest) robot path on the terrain; Dotted red line: actual tether configuration; Solid blue line: unfolded path; Dotted blue line: unfolded tether configuration)

Problem 1. Our method combines a tangent graph-based search space, inspired by [18], with a stability analysis framework from [19] and extended it to a 3D environment. This integration allows our algorithm to find the shortest stable path while accounting for tether constraints. First, we adapt tangent graphs to define an efficient 3D search space, offering advantages such as reduced computational complexity and obstacle avoidance. Building on this, we introduce a motion planning algorithm that merges hybrid A* search with real-time stability analysis. Key features include explicit consideration of tether constraints, stability guarantees, and computational efficiency achieved through tangent graphs and pruning techniques. We start by detailing the tangent graph construction, followed by a description of the motion planning algorithm that integrates hybrid A* and stability checks to find the optimal path.

A. Pre-Planning: Tangent Graph

Before explaining the construction of a tangent graph, we introduce key concepts to define our graph structure:

***Definition** 6 (Tangent Node):* A tangent node $\mathbf{v}_{t\ell}$ in $\mathcal{G}t$ is either a point on $\partial\bar{\mathcal{O}}_k$ where a tangent from same or another obstacle touches, or an endpoint of a straight line segment on $\partial\bar{\mathcal{O}}_k$. It is associated with vertex † of $\mathcal{O}_k$ if located on the corresponding circular arc of $\partial\bar{\mathcal{O}}_k$.

***Definition** 7 (Graph Node):* A graph node v_ι in the tangent graph is defined by its underlying tangent node $\mathsf{v}_{t\ell}$ and includes the following attributes:

- Tangent node: $\mathsf{v}_\iota.\mathbf{v}_t = \mathbf{v}_{t\ell}$ for some ℓ.
- Parent node: The node from which v_ι is generated, denoted as $\mathsf{v}_\iota.parent$.
- Path from start node: The path from the start node to v_ι is $\mathsf{v}_\iota.\mathcal{P}_s = \mathsf{v}_\iota.parent.\mathcal{P}_s \cup \mathcal{P}(\mathsf{v}_\iota.parent, \mathsf{v}_\iota)$.
- Tether configuration: The tether configuration associated with the path from the start node to v_ι.

***Definition** 8 (Tangent Edge):* A tangent edge $\mathsf{e}^t_{\iota\iota'}$ is the straight line segment connecting two tangent nodes $\mathsf{v}_{t\iota}$ and $\mathsf{v}_{t\iota'}$ in the 2D unfolded world $\mathcal{W}_{2D}$.

***Definition** 9 (Neighboring Tangent Nodes):* For a tangent node $\mathbf{v}_{t\ell}$ on the vertex † of obstacle $\mathcal{O}_k$, its neighbors $\mathcal{N}(\mathbf{v}_{t\ell})$ are all tangent nodes on obstacles on the same or a lower surface (to disallow ascents for efficiency) that are connected to other nodes on the same vertex by straight, tangent line segments.

To efficiently manage our search space and accommodate the robots' kinematic constraints, we construct a $\mathcal{C}^1$-tangent graph as the underlying structure, which is constructed using the following procedure (cf. Fig. 3):

Step 1: Flatten 3D map to 2D – For each surface in the 3D terrain $\mathcal{W}$ (cf. Fig. 3a), rotate the surface along with its corresponding obstacles $\mathcal{O}_k$ to align with the top flat surface (cf. Figs. 3b and 4), producing $\mathcal{W}_{2D}$.

Step 2: Enlarge Obstacles – For each obstacle $\mathcal{O}_k$ in the 2D map $\mathcal{W}_{2D}$, enlarge the obstacle by a margin $\rho_{\bar{o}}$. This accounts for robot's size, safety margin and turning constraint, resulting in an enlarged obstacle $\bar{\mathcal{O}}_k$.

Step 3: Construct 2D Tangent Graph – For each pair of enlarged obstacles $\bar{\mathcal{O}}_k$ and $\bar{\mathcal{O}}_{k'}$, including cases where $k = k'$, connect vertices of $\bar{\mathcal{O}}_k$ and $\bar{\mathcal{O}}_{k'}$ using their common tangents to generate tangent nodes and edges. Store the tangent nodes and edges as graph attributes in the tangent graph $\mathcal{G}_{t,2D}$ (see Definitions 6–8).

Step 4: Rotate 2D tangent graph to 3D – Rotate the constructed 2D tangent graph $\mathcal{G}_{t,2D}$ back to its original 3D orientation to form the final 3D tangent graph $\mathcal{G}_{t,3D}$ (cf. Fig. 3c).

The 3D tangent graph structure forms the foundation of our motion planning algorithm, since paths of the tangent graph naturally satisfy curvature constraints (addressing Problem 1-ii.) [18]. We refer to this 3D tangent graph as $\mathcal{G}_t$ from now on for simplicity. By using this graph as our search space, we significantly reduce computational complexity in complex 3D terrains. Importantly, this graph is created only once for a given map as a pre-planning process, which further enhances efficiency. Once constructed, the motion planning algorithm can use this graph repeatedly to plan paths from any start position on the top flat surface to any end position within the environment. Next, we discuss how this pre-computed tangent graph is integrated into a hybrid A* algorithm.

B. Tangent-based Hybrid A Motion Planning*

In this section, we provide a overview of hybrid A* that uses a $\mathcal{C}^1$-tangent graph $\mathcal{G}_t$ as the underlying graph for search and its integration in our motion planning algorithm.

1) Hybrid A Overview:* The goal is to explore nodes in $\mathcal{G}_t$ to find the shortest path. To achieve this, we define a cost metric that guides the search:

Cost to reach a node: The cost to reach a node v_ι from the start node is the total length of the path from the start to v_ι, expressed as:

$$g[\mathsf{v}_\iota] = l(\mathsf{v}_\iota.\mathcal{P}_s),$$

where $l(\mathsf{v}_\iota.\mathcal{P}_s)$ is the length of the path $\mathcal{P}_s$ from the start node to v_ι.

Heuristic cost to reach the goal: For a node v_ι, the heuristic path to the goal $\mathcal{P}_{heu}(\mathsf{v}_\iota, \mathsf{v}_g)$ is the shortest path from v_ι to the goal, navigating around the obstacle on which v_ι is located but ignoring collisions with other obstacles. This heuristic path is denoted as $\mathsf{v}_\iota.\mathcal{P}_g$. The heuristic cost, $h[\mathsf{v}_\iota]$, is the length of this heuristic path:

$$h[\mathsf{v}_\iota] = l(\mathsf{v}_\iota.\mathcal{P}_g).$$

Total estimated cost: The total estimated cost of a node v_ι, $f[\mathsf{v}_\iota]$, is the sum of the cost to reach v_ι and the heuristic cost to the goal from v_ι:

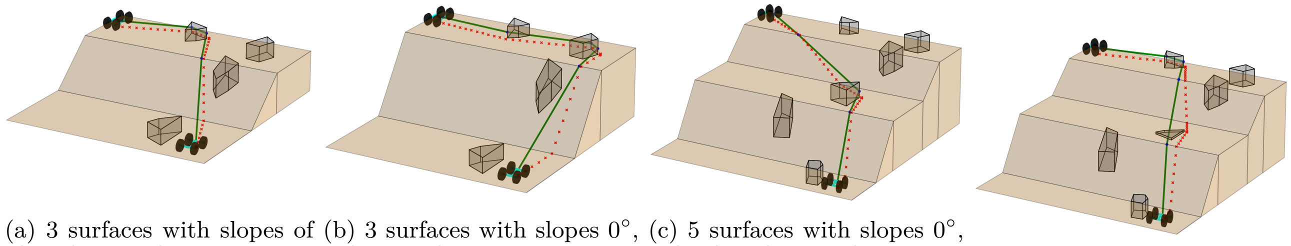
$$f[\mathsf{v}_\iota] = g[\mathsf{v}_\iota] + h[\mathsf{v}_\iota].$$

2) The THAMP-3D Algorithm: The Tangent-based Hybrid A* Motion Planning-3D (THAMP-3D) algorithm, a modified version of the hybrid A* integrated with stability analysis, is outlined in Algorithm 1, inspired from [18]. The steps for developing the algorithm are as follows: 1) Tangent Construction – Construct tangents from the start position to each obstacle and add them to the pre-planned tangent graph. 2) Node Expansion – Expand reachable tangent nodes on the vertices of nearby obstacles in the $\mathcal{C}^1$-tangent graph. 3) Stability Check – Check the stability of the path between the current and neighboring nodes, deciding whether to include or prune those nodes. 4) Cost Calculation – Compare nodes for expansion using path length costs. 5) Goal Exploration – Continue exploring the graph until a stable path to the goal state is found.

Algorithm 1: THAMP-3D: Tangent-based Hybrid A* Motion Planning-3D

```
   Input: $\mathsf{v}_s \leftarrow \mathbf{r}_{s2}$, $\mathsf{v}_g \leftarrow \mathbf{r}_{g2}$, $\mathcal{O}'$, $\mathcal{G}_t$, $\mathcal{W}$
   Output: $\mathcal{P}$
1  Function THAMP3D ($\mathsf{v}_s, \mathsf{v}_g, \mathcal{G}_t$):
2      $\mathcal{V}_{op} \leftarrow \{\mathsf{v}_s\}$;
3      $\mathcal{V}_{cl} \leftarrow \{\}$ ;
4      while $\mathcal{V}_{op}$ is not empty do
5          $\mathsf{v}_b = \arg\min_{v \in \mathcal{V}_{op}} f[v]$;
6          $\mathcal{P}_{dub}(\mathsf{v}_b, \mathsf{v}_g)$ = dubinsPath($\mathsf{v}_b, \mathsf{v}_g$)
7          if $\mathcal{P}_{dub}(\mathsf{v}_b, \mathsf{v}_g) \subset \mathcal{W}_{nt} \setminus \mathcal{O}'$ and
            isStable($\mathsf{v}_b.\mathcal{T}_s, \mathcal{P}_{dub}(\mathsf{v}_b, \mathsf{v}_g)$) then
8              $\mathcal{T}' \leftarrow (\mathsf{v}_b.\mathcal{T}_s, \mathcal{P}_{dub}(\mathsf{v}_b, \mathsf{v}_g))$
9              if isTangleFree($\mathcal{T}'$) then
10                 $\mathcal{P}$ = bestPathAmong($\mathcal{V}_{op}, \mathsf{v}_b$)
11                 return $\mathcal{P}$
12         $\mathcal{V}_{op}.remove(\mathsf{v}_b)$
13         $\mathcal{N}_c(\mathsf{v}_b)$ = childrensOf($\mathsf{v}_b$)
14         foreach $\mathsf{v}_n \in N_c(\mathsf{v}_b)$ do
15             if $\mathsf{v}_n$ in $\mathcal{V}_{cl}$ then
16                 continue;
17             if not isTangleFree ($\mathsf{v}_n.\mathcal{T}_s$) then
18                 $\mathcal{V}_{cl}.append(\mathsf{v}_n)$
19                 continue;
20             if $\mathsf{v}_n$ not in $\mathcal{V}_{op}$ then
21                 $\mathcal{V}_{op}.append(\mathsf{v}_n)$
22     return $\mathcal{P}$
```

Algorithm 1 is based on Algorithm 1 from [18], which works on 2D tangent graphs, but we adapt it to 3D tangent graphs while maintaining similar functionality and integrated stability check. In line 2, Robot 2's start position is added to the open node set, and the node with the smallest estimated cost is expanded in line 5. The `dubinsPath` function (line 6) computes the shortest Dubins path [22] between nodes, satisfying turning constraints (addressing Problem 1-ii.), while line 7 checks if the path $\mathcal{P}_{dub}(\mathsf{v}_b, \mathsf{v}_g)$ is collision-free and stable (addressing Problem 1-i. and 1-iii.). Using the current tether configuration, $\mathsf{v}_b.\mathcal{T}_s$, and the path $\mathcal{P}_{dub}(\mathsf{v}_b, \mathsf{v}_g)$, the function `isStable` continuously updates the tether configuration in the 3D world (cf. Fig. 4) and checks the robot's static stability along the path. Tether updates can be done using existing algorithms from [17], [19], which predict when the tether will anchor or de-anchor as the rover moves between waypoints, while avoiding anchors on rocks of smaller height. Stability checks are performed using a linear program to ensure forces and moments are balanced as the robot progresses in small steps. For the sake of brevity, the readers are referred to [19, Algorithm 1] for the tether update algorithm and [19, Section III-B] for the stability-checking linear program. The `isTangleFree` function (line 9) checks if the tether configuration is free of intersections or entanglements (addressing Problem 1-iv.). In line 10, `bestPathAmong` returns the best valid path through either the current



(a) 3 surfaces with slopes of 0°, 40°, and 0°

(b) 3 surfaces with slopes 0°, 55°, and 0°

(c) 5 surfaces with slopes 0°, 30°, 0°, 45°, and 0°

(d) same as c and obstacle on the middle surface is short

Fig. 5: Shortest stable paths generated by the THAMP-3D algorithm in various environments/scenarios, where the robot travels from the top surface to the bottom surface. The red cross indicates the robot's path, the blue dots represent anchor points on obstacles, and the green line shows the final tether configuration for the shortest stable path. The robot is not shown to scale and is included for visualization purposes only.

node v_b or another node in the open set that is better than current node. The `childrensOf`(v) function identifies valid neighboring nodes that can be reached with forward motion, excluding reverse movements or unstable paths. Finally, lines 14–21 add valid children nodes to the open set if they have not been explored and the tether configuration is free of entanglement. This approach efficiently handles tether constraints, stability checks, and path optimization in 3D terrains, resulting in a feasible and efficient solution for tethered robot navigation.

IV. Results

In this section, we study the effectiveness and performance of the proposed algorithm via various simulation studies, out of which the results of four scenarios are shown in Fig. 5. For the simulations, we consider the mass of both robots to be $m_1 = m_2 = 200kg$, $g = 3.73m/s^2$, and coefficient of friction $\mu = 0.46$ (corresponding to friction angle $\phi = 25°$), where g and μ value are chosen to mimic the Martian surface. In Scenario A, three surfaces $\mathcal{S}_0$, $\mathcal{S}_1$ and $\mathcal{S}_2$ are considered with slopes 0°, 40° and 0°, respectively, from top to bottom, where slope angles are with respect to top surface. Due to the steep slope on $\mathcal{S}_1$, the direct path from start to goal is unstable and the robot has to wind around an obstacle on the top surface to amplify the tether tension utilizing capstan effect in order to reach the goal, as shown in Fig. 5a, which is the shortest stable path in this case. In Scenario B, the slope of $\mathcal{S}_1$ is increased from 40° to 55°. Since the slope is even steeper, the rover has to wind around multiple obstacles to achieve the required tether tension to support the motion (cf. Fig. 5b). Table I shows that the time, the path length and the number of iterations increased for Scenario B compared to Scenario A due to the requirement of extra winding around obstacles. In Scenario C, we considered five surfaces $\mathcal{S}_0$, $\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$ and $\mathcal{S}_4$ with slopes 0°, 30°, 0°, 45° and 0°, respectively. As we can see, the slope in this case is gentle for the surface $\mathcal{S}_1$ as compared to previous scenarios; thus, the rover can directly traverse through $\mathcal{S}_1$ without winding around any obstacles on $\mathcal{S}_0$. But, since $\mathcal{S}_3$ is a steep slope, to traverse it safely, the rover has to wind around an obstacle on $\mathcal{S}_2$, which corresponds to the shortest path in this case, as shown in Fig. 5c. In Scenario D, we keep the surfaces the same as in Scenario C, but reduced the height of the obstacle on $\mathcal{S}_2$ to show the effects of the presence of shorter/smaller obstacles or holes on the final path. Since anchoring is not possible around the short obstacle, the robot has to take another path, as shown in Fig. 5d, which is longer than the path in Fig. 5c to satisfy the stability constraints as shown in Table I. This shows the effects of holes/smaller obstacles on the final path and our approach effectively incorporate this to provide a more realistic solution.

TABLE I: Path data for various scenarios

Scenario	Time [s]	$l(\mathcal{P})$	Number of iterations
A	4.36	183.155	6
B	10.07	233.42	30
C	7.68	217.20	9
D	13.3	243.39	23

Videos of the simulations for all four scenarios are available at https://tinyurl.com/fvj92snj, which demonstrate the efficacy of our approach to solve Problem 1.

V. Conclusions

In this work, we presented the THAMP-3D algorithm, a tangent-based hybrid A* motion planner specifically designed for tethered robots navigating sloped 3D terrains. By combining stability analysis with a tangent graph representation, our method enables safe and efficient traversal through challenging environments. The simulation results demonstrate the algorithm's ability to generate stable and feasible paths while satisfying tether constraints. However, a limitation of the current approach is its reliance on the tangent graphs, which may pose challenges in environments with few or no obstacles. To address this issue, our future work will focus on modifying the algorithm to identify the shortest stable path when obstacles are scarce.

References

[1] S. J. Conway and D. E. Stillman, "The role of liquid water in recent surface processes on Mars," in *Mars Geological Enigmas*. Elsevier, 2021, pp. 207–261.

[2] C. M. Dundas, M. T. Mellon, L. V. Posiolova, K. Miljković, G. S. Collins, L. L. Tornabene, V. G. Rangarajan, M. P. Golombek, N. H. Warner, I. J. Daubar *et al.*, "A large new crater exposes the limits of water ice on Mars," *Geophysical Research Letters*, vol. 50, no. 2, p. e2022GL100747, 2023.

[3] R. Washington, K. Golden, J. Bresina, D. E. Smith, C. Anderson, and T. Smith, "Autonomous rovers for Mars exploration," in *1999 IEEE Aerospace Conference. Proceedings (Cat. No. 99TH8403)*, vol. 1. IEEE, 1999, pp. 237–251.

[4] M. Heverly, J. Matthews, J. Lin, D. Fuller, M. Maimone, J. Biesiadecki, and J. Leichty, "Traverse performance characterization for the Mars Science Laboratory rover," *Journal of Field Robotics*, vol. 30, no. 6, pp. 835–846, 2013.

[5] R. E. Arvidson, "Roving on Mars with opportunity and curiosity: terramechanics and terrain properties," in *Earth and Space 2014*, 2014, pp. 165–173.

[6] P. Abad-Manterola, I. A. Nesnas, and J. W. Burdick, "Motion planning on steep terrain for the tethered axel rover," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 4188–4195.

[7] P. Abad-Manterola, J. Edlund, J. Burdick, A. Wu, T. Oliver, I. A. Nesnas, and J. Cecava, "Axel," *IEEE Robotics & Automation Magazine*, vol. 16, no. 4, pp. 44–52, 2009.

[8] P. McGarey, F. Pomerleau, and T. D. Barfoot, "System design of a tethered robotic explorer (trex) for 3d mapping of steep terrain and harsh environments," in *Field and Service Robotics: Results of the 10th International Conference*. Springer, 2016, pp. 267–281.

[9] I. A. Nesnas, J. B. Matthews, P. Abad-Manterola, J. W. Burdick, J. A. Edlund, J. C. Morrison, R. D. Peters, M. M. Tanner, R. N. Miyake, B. S. Solish, and R. C. Anderson, "Axel and DuAxel rovers for the sustainable exploration of extreme terrains," *Journal of Field Robotics*, vol. 29, no. 4, pp. 663–685, 2012.

[10] S. W. Attaway, "The mechanics of friction in rope rescue," in *International Technical Rescue Symposium*, vol. 7, 1999.

[11] P. G. Xavier, "Shortest path planning for a tethered robot or an anchored cable," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, vol. 2. IEEE, 1999, pp. 1011–1017.

[12] S. Kim, S. Bhattacharya, and V. Kumar, "Path planning for a tethered mobile robot," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1132–1139.

[13] O. Salzman and D. Halperin, "Optimal motion planning for a tethered robot: Efficient preprocessing for fast shortest paths queries," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4161–4166.

[14] X. Zhang and Q.-C. Pham, "Planning coordinated motions for tethered planar mobile robots," *Robotics and Autonomous Systems*, vol. 118, pp. 189–203, 2019.

[15] M. Cao, K. Cao, S. Yuan, T.-M. Nguyen, and L. Xie, "Neptune: Nonentangling trajectory planning for multiple tethered unmanned vehicles," *IEEE Transactions on Robotics*, 2023.

[16] T. Igarashi and M. Stilman, "Homotopic path planning on manifolds for cabled mobile robots," in *Algorithmic Foundations of Robotics IX: Selected Contributions of the Ninth International Workshop on the Algorithmic Foundations of Robotics*. Springer, 2010, pp. 1–18.

[17] M. Paton, M. P. Strub, T. Brown, R. J. Greene, J. Lizewski, V. Patel, J. D. Gammell, and I. A. Nesnas, "Navigation on the line: Traversability analysis and path planning for extreme-terrain rappelling rovers," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 7034–7041.

[18] V. S. Chipade, R. Kumar, and S. Z. Yong, "WiTHy A*: Winding-constrained motion planning for tethered robot using hybrid A*," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 8771–8777.

[19] R. Kumar, V. S. Chipade, and S. Z. Yong, "Stability of tethered ground robots on extreme terrains," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, accepted.

[20] M. M. Tanner, J. W. Burdick, and I. A. Nesnas, "Online motion planning for tethered robots in extreme terrain," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 5557–5564.

[21] R. Hegde and D. Panagou, "Multi-agent motion planning and coordination in polygonal environments using vector fields and model predictive control," in *2016 European Control Conference (ECC)*. IEEE, 2016, pp. 1856–1861.

[22] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of mathematics*, vol. 79, no. 3, pp. 497–516, 1957.