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

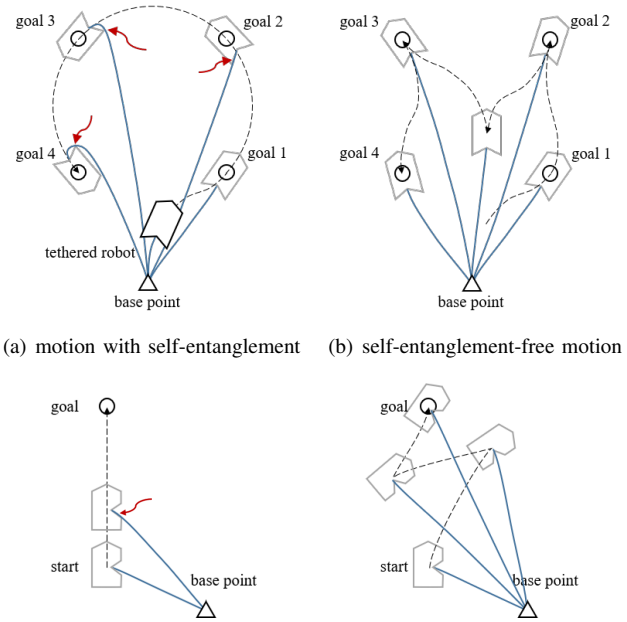
Self-Entanglement-Free Tethered Path Planning for Non-Particle Differential-Driven Robot

Tong Yang, Jiangpin Liu, Yue Wang, and Rong Xiong

***Abstract*— A novel mechanism to derive self-entanglement-free path for tethered differential-driven robots is proposed in this work. The problem is tailored to the applications of tethered robots without an omni-directional tether re-tractor which is often encountered when an omni-directional tether retracting mechanism is incapable of being jointly equipped with other geometrically complex devices (e.g. a manipulator), for instance the disaster recovery, spatial exploration, etc. Without a special consideration on the spatial relation between the pose of the mobile base and the tether, self-entanglement appears when the robot moves, resulting in unsafe motion of the robot and potential damage to the tether. In this paper, the self-entanglement-free constraint is modelled as the admissible orientation of the tether anchoring on the robot with respect to the robot's heading orientation. A searching-based path planning algorithm is then proposed to generate a near optimal path solution with guaranteed null of tether self-entanglement. The effectiveness of the proposed algorithm is compared with the motions without considering self-entanglement-free constraint, illustrated in challenging planning cases, and validated in real-world scenes. An open-source implementation has also been provided for the benefit of the robotics community.**

I. Introduction

The tethered robot, a mobile robot equipped with a tether anchoring to a fixed base point, has natural advantages of maintaining stable communication links and power/material supplies, making them suitable for carrying out energy-consuming tasks and working in wireless communication–denied environments, for instance sewer pipe inspection [1], highway maintenance [2], coverage tasks [3] [4] [5], disaster recovery missions [6], mountain climbing tasks [7] [8], and exploration tasks [9]. Yet in most scenarios, the mobile robots adopted were not intended for tethered applications, hence an omni-directional tether-robot anchoring point does not exist. In this case, a phenomenon referred to as *self-entanglement* appears which may affect the safety of the tethered robot motion: If the mobile robot rotates cyclically, then the tether would entangle with the mobile robot. The self-entanglement problem is fundamentally caused by the inappropriate relative angle between the robot's heading direction and the cable stretching direction. See Fig. 1 for a visual illustration of how the cable orientation may influence the motion of a tethered differential-driven robot with the aim to avoid self-entanglement. Furthermore, due to the gravity, physically the tether will always sag, thus the self-entanglement may further translate to the wheels rolling over the tether, resulting in unexpected tethered robot states (with a self-crossing tether) and incurring damage to the tether structure. Solving this specialised path planning problem for tethered robots, referred to as the *Self-Entanglement-Free Tethered Path Planning* (SEFTPP) task, is the objective of this paper.



(a) motion with self-entanglement (b) self-entanglement-free motion

(c) motion with self-entanglement (d) self-entanglement-free motion

Fig. 1. Examples of different tethered robot motions to visit the goals depicted as small circles. The base point anchoring the tether is depicted as a triangle. For simplicity, no obstacle is presented, and the maximum cable length constraint is ignored. The mobile robot (the polygon) is differential-driven, with the cable (blue curve) being hooked (a)(b) on the back and (c)(d) to the right. (a) The circular motion is valid for an untethered robot but inapplicable for a tethered robot. The tether would contact the rear wheel after the robot visits goal 2. (b) To avoid self-entanglement, after the robot visits goal 2, it performs a backward motion to an intermediate pose which admits the subsequent self-entanglement-free motion. (c) For a robot whose tether extends to the right, even straight forward motion would cause self-entanglement. The admissible tether retracting orientation at the tether-robot anchoring point is depicted by the notch on the robot footprint. (d) A self-entanglement-free resulting path for the case in (c).

Given any self-entanglement-free (SEF) tethered robot configuration, the admissible instant robot motion (primitive) should be conditioned by not only the differential-driven robot kinematic constraints and the maximum tether length constraint but also the tether stretching direction on the robot, which this makes the SEFTPP solution non-intuitive even

This work was supported by the National Key R&D Program of China under Grant 2021ZD0114500. *(Corresponding author: Yue Wang and Rong Xiong.)*

Tong Yang, Jiangpin Liu, Yue Wang, and Rong Xiong are with the State Key Laboratory of Industrial Control and Technology, Zhejiang University, P.R. China.

for simple tasks (Fig. 1(b) and Fig. 1(d)). All existing work (to be recounted in Section II) on tethered robot planning have assumed either particle robots or omni-directional (2D) robots, in which the self-entanglement problem was safely ignored. However, such assumptions were impractical in real-world challenging applications. And one can expect that, restricting the admissible robot heading orientation will significantly reduces the set of admissible motions, which eventually makes the SEFTPP problem non-trivial and much harder than untethered planning problems.

This work advocates for reporting the first solution to the guaranteed self-entanglement-free path for a tethered differential-driven robot. The proposed algorithm departs from all existing tethered robot planners that, the orientation difference between the robot's heading direction and the cable stretching direction is explicitly incorporated in the path searching process which naturally motivates a constrained path planner to solve the SEFTPP problem. The contributions of this paper can be summarised as:

1) The modelling of the SEFTPP problem into a constrained path planning problem.
2) The first SEFTPP solution for tethered differential-driven mobile robots.
3) The open-sourcing [1] of the algorithm.

The remainder of this paper is organised as follows. Section II reviews and contextualises the problem within the existing literature. Section III formally models the SEFTPP problem. Section IV delves into details to describe the proposed solver to generate the SEFTPP solution. Experimental illustrations and comparisons are collected in Section V, with final concluding remarks gathered in Section VI.

II. Related Works

The *tethered path planning* (TPP) task has been intensively investigated in the past decades. Application of tethered robots in various tasks have been presented in [2] [1] [6] [7] [8] [3] [9], etc. Main concentrations have been paid on constructing the shortest tethered robot motion complying with the maximum cable length constraint. It has been observed [10] that the tether states will be non-homotopic if the robot reaches the same goal through paths in different topological routes, and the shortest motion for an untethered robot is in all likelihood untrackable by a tethered robot.

Early work restricted their scope in polygonal environments [11] [12] so that the algorithmic complexity can be calculated [13] [14] as a polynomial of the number of straight segments in the initial tether state and the number of obstacle vertices. In recent years, different tethered robot configurations at the same mobile robot pose have been proven distinguishable by calculating the homotopy classes of the tether state. And the most popular solutions to find the tethered robot path are based on the path-finding within the pre-calculated set of all cable-length-admissible configurations, referred to as the homotopy augmented graph [10].

[1] https://github.com/ZJUTongYang/seftpp

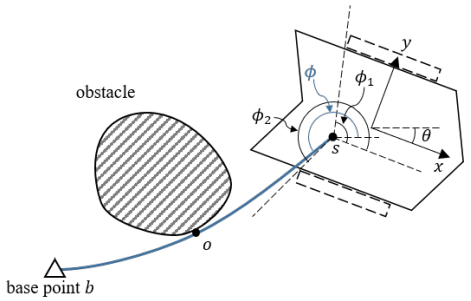


Fig. 2. Illustration of definitions and notations.

Later, with the utilisation of a locally obstacle-free shortest path planner, the cable state is estimable, and pre-calculation is applicable to arbitrary planar environments [15] [16]. Moreover, [11] pre-computed the reachable cell for tethered robots for more efficient querying during planning.

All existing works mentioned above have assumed either a particle robot or an omni-directional robot, and considered only the cable entanglement with the environment obstacle, ignoring the self-entanglement which is the main motivation of this work. Special treatment of the tether self-entanglement will be incorporated into the proposed algorithm in this work, generating guaranteed self-entanglement-free tethered robot motion, thus filling in the gap between simulated path planners and real-world execution.

III. Problem Modelling

In this section, the notations for the environment settings and the tethered differential-driven robot are introduced, and the self-entanglement-free tethered path planning (SEFTPP) problem is modelled.

A. Definitions and Robot Kinematics

Let $M \subset \mathbb{R}^2$ be the environment where the tethered robot moves. The differential-driven robot's footprint is assumed as a polygon. The base point is b, fixed in the environment, and s is the robot-tether anchoring point on the mobile robot [11]. See Fig. 2 for the illustration of notations. The configuration c of a tethered robot consists of the mobile robot's pose and the tether shape:

$$c = \{x, y, \theta, O\} \tag{1}$$

where O is the storage of the location of tether-obstacle contact points in order from the base point to the robot. Let the last element in O be o, then the shape of tether is $[O; s]$ (appending s in O). The retracting direction (tangent) of the tether at s is denoted as ϕ, estimated by the straight line:

$$\phi = \frac{\vec{so}}{\| \vec{so} \|} \tag{2}$$

Given the geometric structure of the tethered robot, without an omni-directional tether-robot anchoring point, the admissible tether retracting orientation will never be $[0, 2\pi)$ but an interval $[\phi_1, \phi_2]$ (in the robot's frame).

Algorithm 1 Self-Entanglement-Free Tethered Path Planner

Input: Map M, Initial configuration c_0, Goal location p_{goal}, Base point b, Maximum tether length L, Resolution $x_{\text{res}}, y_{\text{res}}, \theta_{\text{res}}$

Output: Resulting path R

```
% Initialize Data Structure
{ζ_i} = getRepresentativePointOfObstacles(M)
n_0 = initialiseNode(c_0)
[x_d, y_d, θ_d] = getIndex(n_0, x_res, y_res, θ_res)
V(x_d, y_d, θ_d).push(n_0) % The grid discretisation
Q.push(n_0) % The priority queue
while ~isempty(Q) do
    n_cur = Q.pop()
    if isGoalReached(n_cur, p_goal) then
        R = [tracePath(V, n_cur);p_goal]
        return R
    end if
    N_succ = getValidSuccessors(n_cur)
    for each element n_succ in N_succ do
        [x_ds, y_ds, θ_ds] = getIndex(n_succ, x_res, y_res, θ_res)
        n_homo = findSimilarNode(n_succ, V(x_ds, y_ds, θ_ds))
        if n_homo = ∅ then
            Q.push(n_succ)
            V(x_ds, y_ds, θ_ds).push(n_succ)
        else
            if n_succ.gCost < n_homo.gCost then
                if n_homo is in Q then
                    remove n_homo from Q
                end if
                Q.push(n_succ)
                V(x_ds, y_ds, θ_ds).push(n_succ)
            end if
        end if
    end for
end while
R = ∅
return R
```

Algorithm 2 getValidSuccessors

Input: current node n_{cur}, Map M, Maximum cable length L, representative obstaclel ζ

Output: Child node list N_{succ}

```
P = allPrimitives()
N_succ = ∅
for each element (steer, dir, dis) in P do
    [n_succ, N_mid] = generateMotion(n_cur, (steer, dir, dis))
    n_succ.gCost = movementCost(n_cur)
    for each element n_mid in N_mid do
        n_mid.O = getCableState(n_mid, N_mid)
        if ~isSEF(n_succ) ||
           ~isCollisionFree(n_mid) ||
           CableLength(n_mid)> L then
            continue
        end if
    end for
    n_succ.N_mid = N_mid
    n_succ.H = calculateHsignature(n_succ, ζ)
    N_succ.push(n_nucc)
end for
```

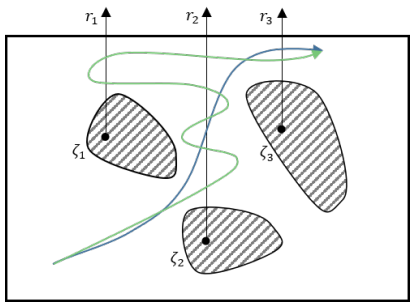


Fig. 3. Illustration of the calculation of H-signature. In this example, the word representation of the green path and the blue path are "$r_2 r_2^{-1} r_2 r_2^{-1} r_1^{-1} r_1 r_2 r_3$" and "$r_2 r_3$", respectively, which are equivalent under the word reduction. Hence the two paths are homotopic. If the tethered robot moves along the two paths, the configurations will be exactly the same.

B. Self-Entanglement-Free Tethered Path Planning

Given the robot's initial configuration c_0 and the goal location $p_{\text{goal}} = (x_p, y_p)$, the solution to the SEFTPP problem is a sequence of mobile robot poses $\{c_0 = (x_1, y_1, \theta_1, O_1), \cdots, (x_n, y_n, \theta_n, O_n)\}$, such that for any intermediate waypoint $(x_i, y_i, \theta_i, O_i)$[2]

1) (Collision-free) The mobile robot staying at (x_i, y_i, θ_i) is collision-free.
2) (Tether length admissible) The length of tether is shorter than the maximum tether length.
3) (Self-entanglement-free) The mobile robot cannot traverse across the tether, and the relative angle between the robot's heading orientation and the tether retracting orientation should satisfy:

$$\phi_1 \leq \text{wrapTo2Pi}(\phi_i - \theta_i) \leq \phi_2 \tag{3}$$

[2]Here given the robot initial configuration $(x_1, y_1, \theta_1, O_1)$ and the motion $(x_1, y_1, \theta_1), \cdots, (x_i, y_i, \theta_i)$, the tether state O_i is well-defined, *induced* by the locally obstacle-free tautening of the concatenation of O_1 and the robot motion.

IV. Algorithm

In this section, the SEFTPP problem is effectively solved.

A. Node Definition

The proposed algorithm constructs a searching tree of valid tethered robot configurations in a similar vein as the constrained searching-based optimal path planner [17] [10]. A *node* during the pathfinding fully parameterises the robot's motion state, which consists of the following elements in our implementation:

$$\begin{aligned} n = \{&\{i, x, y, \theta, O, s, \phi, H\}(\text{configuration related}), \\ &\{gCost, hCost, i_{\text{prev}}\}(\text{searching-tree related}), \\ &\{steer, dir\}(\text{cost related})\} \end{aligned} \tag{4}$$

where i is the index of the node, $\{x, y, \theta, O\}$ is the configuration of the tethered robot. The location of cable-robot

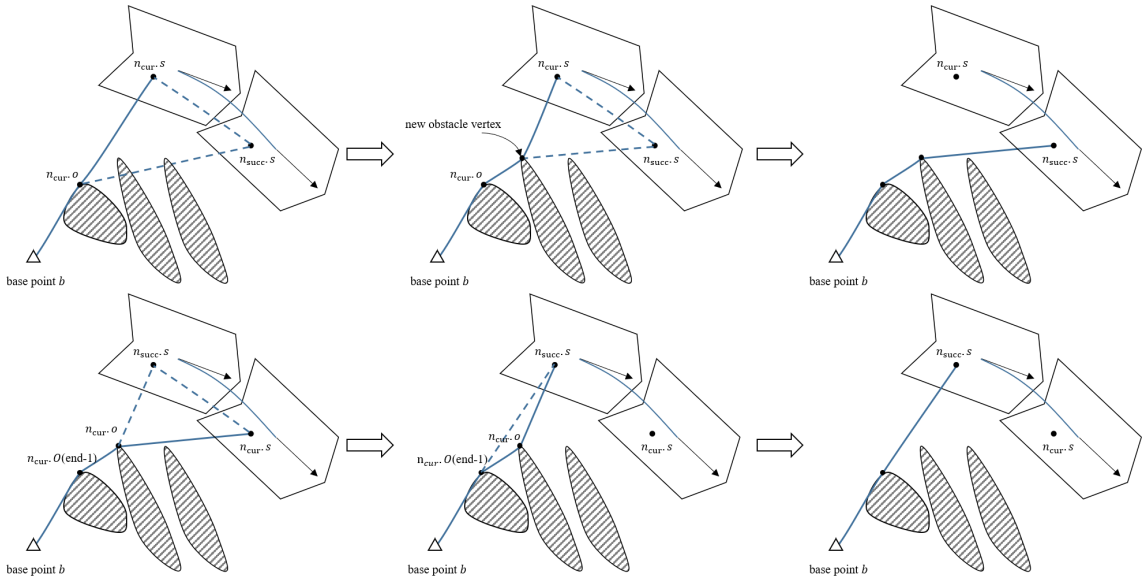


Fig. 4. Illustration of the calculation of the tether state of the child node. Upper row: Iterative steps to add new obstacle vertices that obstruct the tether to O to represent the correct tether state. Lower row: We check whether the tether will depart from any obstacle. If an obstacle is detached, it is removed from the tether-obstacle contact point O list.

anchoring point s and the cable stretching orientation ϕ are derived variables from the robot configuration. H is the H-signature of the robot configuration whose calculation will be detailed later. To execute the path searching process, each node stores the cost-to-move $gCost$, the estimated cost-to-go $hCost$, and the index of its parent node i_{prev}. To consider the penalties in calculating the robot movement cost, the parameters of the motion primitive from n_{cur} to n_{succ}, $steer$ and dir, are stored in n_{succ}.

B. Node Expansion

The pseudocode of the proposed algorithm is shown in **Algorithm 1**. At the beginning of the algorithm, a representative point for each obstacle is firstly distinguished, denoted as $\zeta_1, \cdots, \zeta_n$. Then, parallel non-intersecting rays are constructed denoted as $r_1, \cdots, r_n$. For each path, each crossing of the rays from left to right appends an r_i to the H-signature, whilst each crossing from right to left appends an r_i^{-1}. If r_i and r_i^{-1} are adjacent, then they can be reduced. The H-signature of a path is the reduced word representation. See Fig. 3 for illustration. The author is referred to [15] for a more theoretical explanation. Given the resolution $x_{\text{res}}, y_{\text{res}}$, and θ_{res}, the configuration space of the differential-driven robot is discretised into grids, wherein the grid $V(x_d, y_d, \theta_d)$ stores multiple nodes $\{n_1, \cdots, n_J\}$. The nodes with similar mobile robot pose will be grouped into the same grid, but will not be merged if they have different H-signatures.

Utilising the initial configuration to construct the first node with cost-to-move $n_0.gCost = 0$, it is pushed into a priority queue Q (**Algorithm** 1, line 3∼ 6). In each iteration, the least-cost node n_{cur} will be popped from Q and report all its valid child nodes $N_{\text{succ}} = \{n_{\text{succ}}\}$ (**Algorithm** 1, line 13), detailed in **Algorithm 2**. Various motion primitives may be legit, where in our case since in-situ rotations are rarely SEF, the primitives adopted are car-like circular motions represented by dis (arc length), dir (1 for forward and -1 for backward), and $steer$. For each child node, the robot configuration is calculated as $n_{\text{succ}}.x$, $n_{\text{succ}}.y$, $n_{\text{succ}}.\theta$, $n_{\text{succ}}.O$. The tether state $n_{\text{succ}}.O$ is calculated (**Algorithm** 2, line 8) by checking the following situations iteratively: Whether the tether is obstructed by a new obstacle is observed by checking the existence of obstacle in the triangle $[n_{\text{cur}}.o, n_{\text{cur}}.s, n_{\text{succ}}.s]$. Similarly, whether the tether detaches an obstacle vertex is verified by the null of obstacle in the triangle $[n_{\text{cur}}.O(\text{end}-1), n_{\text{cur}}.o, n_{\text{succ}}.s]$. See Fig. 4 for illustration. After calculating the tether state of the child node, the H-signature is known, $n_{\text{succ}}.H$. Various non-negative cost functions may be applicable to calculate the cost, where in our implementation the cost is calculated as a composite function considering the robot's travelling distance, changes in steering angle, and changes in moving direction, as

$$\begin{aligned} n_{\text{succ}}.gCost = & n_{\text{cur}}.gCost + \alpha dis \\ & + \beta \parallel n_{\text{succ}}.steer - n_{\text{cur}}.steer \parallel \\ & + \gamma \parallel n_{\text{succ}}.dir - n_{\text{cur}}.dir \parallel \\ & \alpha, \beta, \gamma > 0 \end{aligned} \tag{5}$$

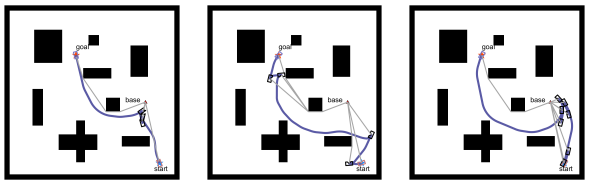
By calculating the discretised index $(x_{\text{ds}}, y_{\text{ds}}, \theta_{\text{dis}})$, n_{succ} is located in a grid (**Algorithm** 1, line 16). If multiple nodes with the same H-signature stay in the same grid, only the least-cost one among them is preserved. Iteratively expanding the least-cost node in the queue until the goal grid is reached, a valid resultant path will be reported by back-tracing the nodes following the child-parent relation.

Note that the proposed algorithm is not necessarily analytically optimal, because among homotopic robot configurations in the same grid the algorithm will only preserve the least-cost one of but not all of them. This has also been

TABLE I

PERFORMANCE OF CASE STUDIES

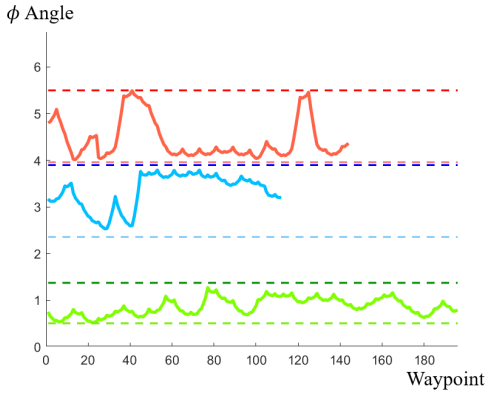
Case	$[\phi_1, \phi_2]$	Base Point	Start Point	Goal Point	Path Length	Run Time	Node Expanded
1 – 1	[2.36, 3.93] (back)	(80.50, 44.50)	(88.50, 9.50, 5.05)	(41.50, 71.50)	100.37	2.15s	134986
1 – 2	[3.93, 5.50] (right)	(80.50, 44.50)	(88.50, 9.50, 2.98)	(41.50, 71.50)	128.69	13.54s	555996
1 – 3	[0.51, 1.11] (left)	(80.50, 44.50)	(88.50,9.50,1.35)	(41.50, 71.50)	176.12	0.87s	14011
2 – 1	[3.93, 5.50] (right)	(24.50, 19.50)	(23.50, 59.50, 0.02)	(87.50, 39.50)	78.90	2.52s	143377
2 – 2	[3.93, 5.70] (right)	(24.50, 19.50)	(23.50, 59.50, 6.21)	(87.50, 39.50)	78.88	3.64s	172109
2 – 3	[3.73, 5.70] (right)	(24.50, 19.50)	(23.50, 59.50, 6.06)	(87.50, 39.50)	68.35	1.26s	50456
3 – 1	[3.93, 5.50] (right)	(75.50, 30.50)	(78.50, 50.50, 5.89)	(40.50, 51.50)	46.64	1.42s	43018
3 – 2	[3.93, 5.70] (right)	(75.50, 30.50)	(78.50, 50.50, 6.04)	(40.50, 51.50)	39.56	0.20s	5697
3 – 3	[3.73, 5.70] (right)	(75.50, 30.50)	(78.50, 50.50, 6.13)	(40.50, 51.50)	39.56	0.25s	6747
4 – 1	[3.93, 5.50] (right)	(23.50, 60.50)	(28.50, 44.50, 3.20)	(85.50, 40.50)	93.88	14.23s	251332
4 – 2	[3.93, 5.70] (right)	(23.50, 60.50)	(28.50, 44.50, 3.10)	(85.50, 40.50)	78.80	12.71s	245908
4 – 3	[3.73, 5.70] (right)	(23.50, 60.50)	(28.50, 44.50, 3.20)	(85.50, 40.50)	72.72	8.79s	163925
5 – 1	[3.93, 5.50] (right)	(77.50, 80.50)	(50.50, 68.50, 2.23)	(75.50, 34.50)	71.67	7.68s	105482
5 – 2	[3.93, 5.70] (right)	(77.50, 80.50)	(50.50, 68.50, 1.89)	(75.50, 34.50)	64.47	6.60s	92374
5 – 3	[3.73, 5.70] (right)	(77.50, 80.50)	(50.50, 68.50, 1.99)	(75.50, 34.50)	60.78	11.14s	100782



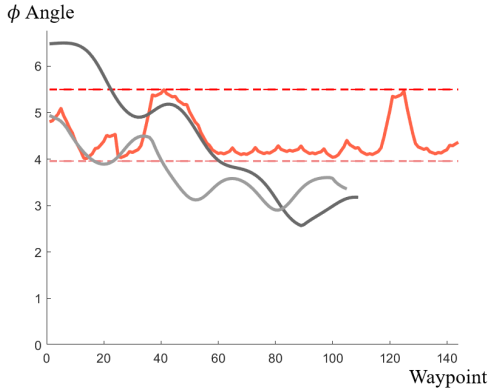
(a) Case 1 – 1 (b) Case 1 – 2 (c) Case 1 – 3

(d) Case 1 – 4 (e) Case 1 – 5

Fig. 5. Illustration of tethered motions from the same start point to the same goal point. (a)(b)(c) Motions satisfying specified SEF constraints. (d)(e) Commonplace (untethered) differential-driven robot motions, where (d) violates the SEF constraint and (e) violates both the SEF constraint and the maximum tether length constraint.



(a) Variation of ϕ in Case 1-1 (blue), 1-2 (red), and 1-3 (yellow).



(b) Variation of ϕ in Case 1-2 (red), 1-4 (dark grey), and 1-5 (light grey).

Fig. 6. Illustration of the angle difference between robot's heading orientation and the tether retracting direction. Admissible upper bounds and lower bounds are depicted.

noted by existing searching-based algorithms [17]. However, with the utilisation of the priority queue, the resulting path is usually near optimal.

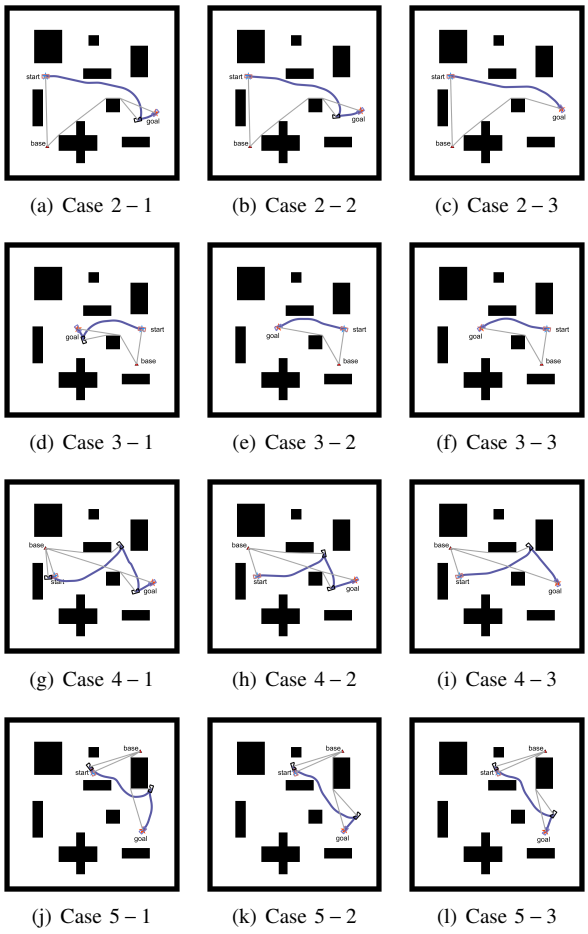
V. EXPERIMENTAL RESULTS

The proposed algorithm constructs the self-entanglement-free motion for a tethered differential-driven robot with arbitrary shape in arbitrary planar environment. To the best of the author's knowledge, there did not exist a self-entanglement-free tethered path planner. So in Section V-A, the proposed algorithm is compared with the usual different-driven robot motions without considering the SEF constraint. Then in Section V-B, challenging planning cases are reported to examine the validity of the proposed algorithm. Finally, a real-world experiment is provided in Section V-C to prove the applicability of the proposed algorithm in real settings. An open-sourcing implementation has been provided here: `https://github.com/ZJUTongYang/seftpp`.

A. Comparisons

See Fig. 5 for illustration. The comparisons are conducted in a 100×100 (grids) planar map. The base point, the initial location, and the goal location are the same in all testings. The maximum tether length constraint is 80.

The base point are illustrated in triangles and are set as obstacles. The tether states are depicted as grey lines. The robot motions are drawn in thick blue curves. In Case 1-1, 1-2, 1-3, various robot kinematics are presented, with the tethers extending to the back, right, and left, respectively. Using the proposed algorithm, the robot motions are guaranteed to satisfy the SEF constraint, shown in Fig. 5(a)~(c).



(a) Case 2 – 1 (b) Case 2 – 2 (c) Case 2 – 3

(d) Case 3 – 1 (e) Case 3 – 2 (f) Case 3 – 3

(g) Case 4 – 1 (h) Case 4 – 2 (i) Case 4 – 3

(j) Case 5 – 1 (k) Case 5 – 2 (l) Case 5 – 3

Fig. 7. Extensive tethered planning cases given different start points, goal points, and base points.

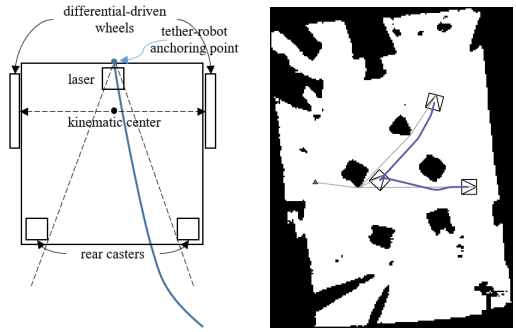
In particular, note that Case 1-3 is the most difficult task as the admissible ϕ interval $[0.51, 1.11]$ is the narrowest. The variation of ϕ angle whilst the robot's moving is visualised in Fig. 6(a). Detailed statistics are collected in Table. I. In contrast, if the paths are designed without explicitly considering the ϕ value, then the resultant motions, such as those visualised in Fig. 5(d)~(e), would almost surely violate the SEF constraint, whose ϕ variation are depicted in Fig. 6(b). The reader is referred to the supplementary video for the animation of the robot motions.

B. Case Studies

More extensive experiments are enumerated in this section. The start points, goal points, and base points are randomly chosen, and different range of ϕ are also adopted. Planning results are shown in Fig. 7. Detailed statistics are collected in Table. I. In most cases, the resultant path can be calculated within 10 seconds, which is believed as reasonable for offline SEFTPP applications.

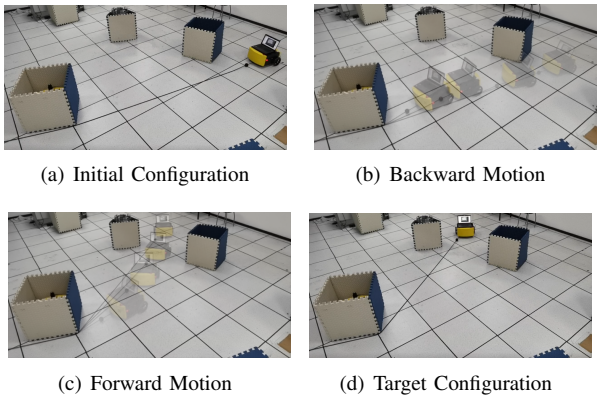
C. Real-world Illustration

Finally, the proposed algorithm is evaluated in a real-world scenario. The structure of the robot is depicted in Fig. 8(a). The front wheels are differential-driven and rear wheels are passive casters. No omni-directional tether retracting mechanism is on the robot. The map is off-line pre-constructed, shown in Fig. 8(b) as well as the resultant path. Although the starting location and the target location are straight-path connectable, because of the limited tether length (6m), the resultant path cannot follow this topology but had to traverse amid obstacles. And at the beginning part of the motion, because of the SEF constraint, the robot cannot rotate to perform a forward motion, but have to move in a backward manner. See Fig. 9 for the real-world robot execution and the supplementary video for animation.



(a) robot structure (b) the environment

Fig. 8. (a) The real-world robot kinematics. The tether is anchoring at the front bottom of the robot chassis, below the laser, and extends to the back, in the middle of two rear casters. (b) The real-world resultant motion.



(a) Initial Configuration (b) Backward Motion

(c) Forward Motion (d) Target Configuration

Fig. 9. Stills of the tethered robot motion to track the path.

VI. Conclusion

A novel mechanism to generate self-entanglement-free (SEF) path for a tethered differential-driven robot has been proposed in this work. The primary motivation of this work is the tethered path planning problem under the absence of an omni-directional tether-robot anchoring point. A searching-based constrained path planner is proposed in this work to generate a near optimal SEF solution for any differential-driven robot in any planar map. Simulated comparisons and illustrations as well as a real-world trial in challenging scenes have proven the validity of the proposed algorithm. These have been supplemented by an open-sourced implementation for the benefit of the community.

REFERENCES

[1] A. A. Nassiraei, Y. Kawamura, A. Ahrary, Y. Mikuriya, and K. Ishii, "Concept and design of a fully autonomous sewer pipe inspection mobile robot "kantaro"," in *Proceedings of the 2007 IEEE international conference on robotics and automation (ICRA)*, pp. 136–143, IEEE, 2007.

[2] D. Hong, S. A. Velinsky, and K. Yamazaki, "Tethered mobile robot for automating highway maintenance operations," *Robotics and Computer-Integrated Manufacturing*, vol. 13, no. 4, pp. 297–307, 1997.

[3] I. Shnaps and E. Rimon, "Online coverage by a tethered autonomous mobile robot in planar unknown environments," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 966–974, 2014.

[4] L. Mechsy, M. Dias, W. Pragithmukar, and A. Kulasekera, "A novel offline coverage path planning algorithm for a tethered robot," in *Proceedings of the 2017 International Conference on Control, Automation and Systems (ICCAS)*, pp. 218–223, IEEE, 2017.

[5] G. Sharma, P. Poudel, A. Dutta, V. Zeinali, T. T. Khoei, and J.-H. Kim, "A 2-approximation algorithm for the online tethered coverage problem.," in *Robotics: Science and Systems*, 2019.

[6] K. S. Pratt, R. R. Murphy, J. L. Burke, J. Craighead, C. Griffin, and S. Stover, "Use of tethered small unmanned aerial system at berkman plaza ii collapse," in *Proceedings of the 2008 IEEE International Workshop on Safety, Security and Rescue Robotics*, pp. 134–139, IEEE, 2008.

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

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

[9] D. Shapovalov and G. A. Pereira, "Exploration of unknown environments with a tethered mobile robot," in *Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6826–6831, IEEE, 2020.

[10] S. Bhattacharya, M. Likhachev, and V. Kumar, "Topological constraints in search-based robot path planning," *Autonomous Robots*, vol. 33, no. 3, pp. 273–290, 2012.

[11] R. H. Teshnizi and D. A. Shell, "Computing cell-based decompositions dynamically for planning motions of tethered robots," in *Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6130–6135, IEEE, 2014.

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

[13] P. G. Xavier, "Shortest path planning for a tethered robot or an anchored cable," in *Proceedings of the 1999 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, pp. 1011–1017, IEEE, 1999.

[14] P. Brass, I. Vigan, and N. Xu, "Shortest path planning for a tethered robot," *Computational Geometry*, vol. 48, no. 9, pp. 732–742, 2015.

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

[16] S. Kim and M. Likhachev, "Path planning for a tethered robot using multi-heuristic a* with topology-based heuristics," in *Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4656–4663, IEEE, 2015.

[17] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 485–501, 2010.