

Online Coverage by a Tethered Autonomous Mobile Robot in Planar Unknown Environments

Iddo Shnaps and Elon Rimon, Dept. of ME, Technion, Israel

Abstract—This paper is concerned with online tethered coverage, in which a mobile robot of size D is attached to a fixed point S by a cable of finite length L . Starting at S , the robot has to cover an unknown planar environment containing obstacles, and return to S with the cable fully retracted. The paper first establishes an optimal off-line tethered coverage methodology, then introduces the TC (Tethered Coverage) algorithm that performs online tethered coverage using position and local obstacle detection sensors. The performance of the TC algorithm is measured by its competitiveness, determined by measuring its total online path length, l , relative to the optimal off-line solution l_{opt} . The paper establishes that the TC algorithm has a competitive performance of $l \leq 2\frac{L}{D}l_{opt}$. Execution example and experiments with a tethered recoiling mechanism illustrate the usefulness of the TC algorithm.

I. Introduction

As mobile robots are deployed in ever more demanding everyday tasks, they must traverse unknown environments containing obstacles while relying on sensors and on-board information storage. Online coverage is perhaps the most common task undertaken by such mobile robots. Much like cleaning a house or mowing a lawn, the objective of this task is to move the robot over every floor tile or patch of the environment, while avoiding collision with obstacles. While such robots are usually battery operated [1, 2], new challenges constantly arise which pose greater energy demands than an on-board battery can provide. For instance, urban street cleaning by autonomous robots requires high power that can only be provided by AC power outlets [7, 17]. Tethered navigation is also necessary in environments where information can only be transmitted via a communication cable, such as underground mines and electrostatic mine-fields [4]. Tethers are also useful as a reliable positioning device in high slippage applications such as lawn mowing [1] and planetary terrain exploration [3].

The paper considers the *online tethered coverage problem*, in which a robot of size D has to cover an unknown environment while connected to a fixed base point S by a cable of finite length L . The robot is initially located at S and must return to this point with the cable fully retracted upon task completion. The robot has no prior knowledge of the environment, but is able to accumulate local information using on-board sensors and data storage. The paper introduces the TC algorithm that performs online tethered coverage using position and local obstacle detection sensors. The algorithm's competitive performance is analyzed and is experimentally demonstrated with a tethered recoiling mechanism.

Due to its many applications, *untethered* online coverage is a highly researched topic in the mobile robot literature

e.g., [5, 6, 8]. However, *tethered* coverage is significantly more complex than untethered coverage, since in addition to avoiding obstacles, the robot must avoid its cable as well as avoid tearing the cable from its base point. The earliest work on *tethered* navigation considered a multi-agent problem, where all robots were connected by separate cables to different base points in the environment and attempted to reach separate targets [9]. A more recent study used a geometric approach to search for the shortest tethered path along the configuration space manifold of a single mobile robot [11]. Tethered navigation with self-crossing of the robot's cable is considered in the computational geometry literature [15, 16]. However, all of these papers are concerned with tethered *navigation* to a given target, while this paper is concerned with tethered *coverage*, where the robot must visit each floor tile in the environment. Moreover, these papers discuss pure *off line* navigation, while this paper emphasizes the *online* problem, where the robot acquires information on the environment during task execution.

The paper's structure and contributions are as follows. The next section describes our setup and discusses complications incurred by the cable. Section 3 establishes an optimal off-line tethered coverage methodology in fully known environments. Based on this methodology, the online TC algorithm is described and illustrated in Section 4. The algorithm's competitive efficiency is analyzed in Section 5. Finally, experimental results with a cable recoiling mechanism are presented in Section 7. The conclusion poses open problems which generalize the tethered coverage problem to a wider set of problems concerned with limited resource mobile robot tasks.

II. Problem Description and Basic Setup

The paper is concerned with the following problem. A mobile robot of size D has to perform online coverage (i.e. visit every point) in a planar unknown environment while connected to a fixed base point, S , by a cable of finite length L . The robot is initially located at S and must return to this point upon task completion with its cable fully retracted. The planar environment is populated by stationary obstacles whose position and shape are unknown to the robot. The robot is equipped with position and local obstacle detection sensors, as well as on-board memory which can be used to accumulate sensor measurements.

The cable is assumed to be a *flexible non-stretchable cord*, that can move freely within the obstacle free portion of the environment. The cable is assumed to be released by a spring loaded recoiling mechanism mounted on the robot, which

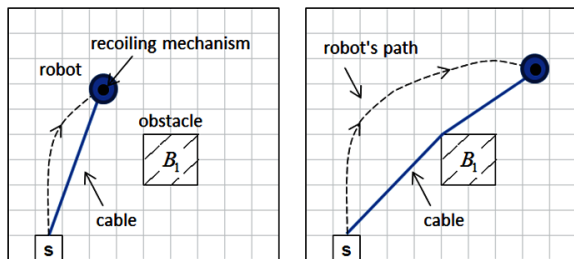


Fig. 1: The cable is kept taut by the on-board recoiling mechanism (Fig. 7 shows a close view of the recoiling mechanism).

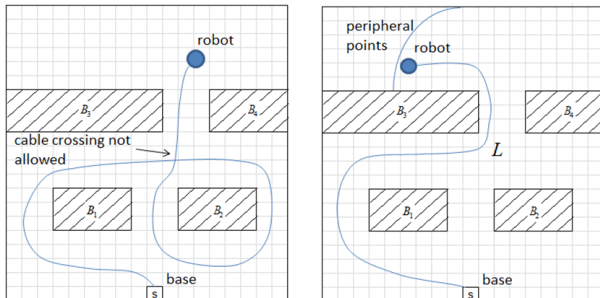


Fig. 2: (a) The cable forms a time varying obstacle. (b) Navigation along inefficient paths incurs repeated coverage.

keeps the cable taut at all times (Figure 1). Note that the cable's recoiling mechanism is mounted on the robot, not at the base point S . This is highly significant in terms of system design, otherwise friction forces at the cable's contacts with obstacle edges would accumulate and generate damping forces on the robot's movements. When the recoiling mechanism is mounted on the robot itself, friction *aids* in keeping the cable taut between successive contact points back to the base point. However, the robot must respect the cable's maximum length, L , and may not travel beyond the cable's reach. Furthermore, the cable forms a single continuous curve from S to the robot's current position, and may not be stepped over by the robot. Last, all obstacles are assumed to be rigid and stationary, meaning they cannot be pushed by the robot or slide due to forces applied by the cable during the robot's motion.

Tethered coverage must satisfy several constraints not encountered in untethered coverage. First, the cable forms a time varying obstacle that must be avoided by the robot. This forces the robot to constantly record the *entire* cable layout in order to avoid crossing it, especially when circumnavigating an internal obstacle (Figure 2(a)). Second, when a tethered mobile robot circumnavigates internal obstacles, it must unwind the cable by retracing its path in order to complete the coverage and eventually return to S . Third, the robot's accessible area is restricted by the cable's finite length. To achieve maximum coverage, the robot may have to reach peripheral points in the environment by way of the shortest path from S (Figure 2(b)). Hence, in addition to its current position, the robot must consider the entire path traveled from S to its current location.

The automatic recoiling mechanism: Careful engineering design can overcome some of the complications associated with tethered coverage. Our recoiling mechanism is mounted on the robot and consists of a loaded torsion spring which keeps the cable taut at all times (Figure 1). The tension causes the cable to form a *piecewise linear* curve along edges of the visibility graph from the robot's current position back to the base point.¹ Assume the cable always lies in the portion of the environment already covered by the robot (this assumption is valid under the TC algorithm). As the robot accumulates data on the environment during coverage, it can compute the cable's piecewise linear shape and thus maintain full knowledge of the cable layout in the environment. Moreover, for a given route between obstacles in the environment, keeping the cable's path along visibility graph edges allows usage of minimal cable length, thus enabling maximum coverage capability. Note that keeping the cable taut does *not* imply that the robot itself must follow the globally shortest path. Note, as well, that the cable always indicates to the robot the shortest path along which it can return to its base point with its cable fully retracted.

III. Optimal Off-Line Tethered Coverage

The online algorithm will be based on the off-line tethered coverage methodology described in this section. The off-line methodology assumes full knowledge of the environments and uses three notions: the shortest path potential function, the internal obstacle saddle lines, and the coverage path split cells. Assume the environment is discretized into $D \times D$ floor tiles, or *cells*. The shortest path potential function is defined as follows.

Definition 1. *The shortest path potential function, $V(p)$, assigns to each cell center point p the minimal traveling distance from S to p along a path that lies in the obstacle free portion of the environment: $V(p) = \text{dst}(p, S)$.*

Remark: The online algorithm will build a *partial* representation of $V(p)$, based on information collected by the robot's sensors and stored in its on-board memory. This partial representation, the *estimated potential function* $\bar{V}(p)$, is defined as the *minimal traveling distance* from S to p , only by way of covered cells. These covered cells represent the portion of the environment currently known to the robot. As opposed to the environment's true potential function, V , the estimated potential, \bar{V} , changes during task execution as the robot discovers new pathways in the environment. Note that only when all cells are covered the robot can conclude that $\bar{V}(p) = V(p)$. \circ

Next consider the saddle lines induced by internal obstacles. By definition, $V(S) = 0$, and V increases monotonically away from the base point S . Around a single internal obstacle, the potential function rises along both sides of the obstacle until its contours meet at the obstacle's far end. The meeting point marks the beginning of a *saddle line* defined as follows.

¹The visibility graph connects all obstacle vertices as well as S with edges that lie in the free space, and it includes all obstacle edges [12].

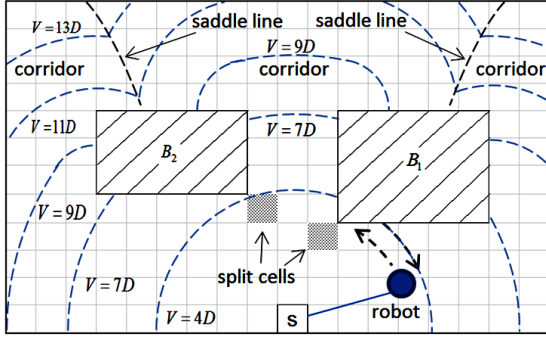


Fig. 3: Equipotential contours of V , with the induced saddle lines and split points.

Definition 2. A *saddle line* is a line segment in the free-space, consisting of cells whose minimal distance to S is achieved via two or more homotopically distinct paths (Fig. 3).

Each internal obstacle induces a *single* saddle line [14][Proposition A.1], and the shortest path from S to any point p lies along a route that has no intersections with any saddle lines. The saddle lines will act as *virtual boundaries* that identify shortest paths from S to points p in the environment. Based on these definitions, the potential function's value at each free cell, $V(p)$, indicates the *minimal cable length* needed in order to reach the point p from S .

Remark: During online coverage, the robot will use the estimated potential function, $\bar{V}(p)$, to determine the saddle lines in the known environment. The location of some saddle lines will therefore *change* at discrete instants when new internal obstacles are discovered during online coverage. \circ

Starting at the base point S , the off-line coverage methodology guides the robot along increasing equipotential contours of V as depicted in Figure 3. When all cells of a certain value $V(p) = c$ are covered, the robot continues along a new contour of cells with potential value $V(p) = c + D$, where D is the robot's size. The robot thus covers D -wide rings that expand outward from S . However, as opposed to a classical breadth first search, when entering a corridor, the robot must follow the equipotential contours *only within this corridor*. Thus, the robot will advance through a series of higher valued contours within one corridor, regardless of the potential value of cells in alternate corridors. When advancing up the equipotential contours, a newly established contact of the current contour with an obstacle may occur. This event divides the equipotential contour and hence the robot's course into two possible corridors. Such events occurs at the following *split cells*.

Definition 3. A *free cell adjacent to an obstacle's boundary* is called a *split cell* if on arrival, the contact with the obstacle boundary splits the equipotential contour of V into two contour segments (Figure 3).

The off-line coverage methodology stores the split cells in a *stack*. Upon reaching a new split cell, the robot chooses

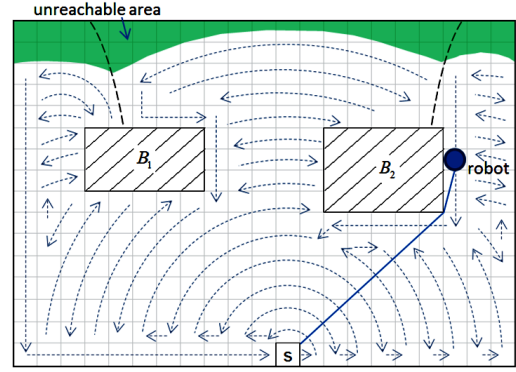


Fig. 4: Execution example of the off-line tethered coverage methodology using cable length $L = 15D$.

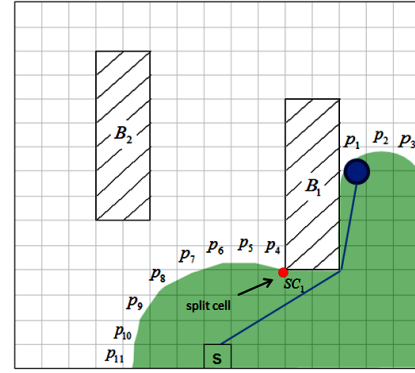


Fig. 5: The environment's open list: $\mathcal{O} = \{p_1, p_2, p_3, SC_1, p_4, \dots, p_{11}\}$.

a single corridor through which it will continue its coverage. When the robot completely covers any particular corridor as described above, it retreats to the previous split cell and proceeds to cover the alternate corridor. Eventually the robot returns to the split cell closest to the base point S , from which it retreats to S with its cable fully retracted.

Example: The off-line coverage methodology is suboptimal only when the saddle lines of V induce D -wide corridors in the environment. In all other cases the off-line coverage methodology is optimal as illustrated in Figure 4. The figure shows a rectangular environment populated by two internal obstacles which induce two saddle lines. Using a cable of length $L = 15D$, the robot follows increasing equipotential contours of V in each corridor while leaving a retreat path. Note that the robot eventually covers the entire accessible environment while visiting every free cell precisely once. \circ

IV. The Online Tethered Coverage Algorithm

This section describes the TC algorithm, an online tethered coverage algorithm for planar unknown environments. The algorithm uses three data structures: the known environment grid M , an *open cell list* \mathcal{O} , and an *unreachable cell list* U . The environment grid, M , holds the covered cells as well as their free neighbors, together with saddle lines information. The open list, \mathcal{O} , consists of the unvisited free cells in M and is

subdivided into sub-lists as discussed below. The unreachable list, U , consists of all free cells known to the robot which hold an estimated potential value $\bar{V} > L$, where L is the cable's length. The free cells in U are located beyond the cable's reach by way of the currently known environment.

The open list \mathcal{O} is sorted by increasing order of the cells' estimated potential function value, $\bar{V}(p)$. However, unlike a classical BFS (i.e. breadth first search), the TC algorithm inserts a stack of the *split cells* into the open list \mathcal{O} , that divides the open cells into sub-lists. As the robot advances up the equipotential contours of \bar{V} along a corridor, it processes only the sub-list of open cells bounded by the split cell that defines the location of the corridor in the environment (see Figure 5). Upon reaching a *new* split cell, the algorithm splits the current sub-list into two sub-lists by inserting the new split cell into \mathcal{O} and dividing the current sub-list accordingly. When all cells within the current sub-list are covered, the algorithm continues to cover cells in the next sub-list of \mathcal{O} , thus reaching every accessible corridor.

The TC algorithm identified a new internal obstacle when the area covered by the robot forms a closed loop. In this event the estimated potential function, \bar{V} , is *updated throughout the known environment grid M* . This on-board computation runs a classical BFS on M . While this takes longer to execute, it is still *much faster* than any physical motion of the robot. During this update, the split cells and the sub-lists of \mathcal{O} are re-defined to match the new estimated potential function. After each update of \bar{V} , the algorithm transfers from the unreachable list U into \mathcal{O} all cells whose new value satisfies $\bar{V} \leq L$. Finally, the robot returns to S only when \mathcal{O} becomes empty. A pseudo-code description of the TC algorithm follows:

TC Algorithm:

Sensors: Position sensor. Obstacle sensor capable of detecting obstacles in the cells adjacent to robot's current position.

Data structures: environment grid M , open list \mathcal{O} , unreachable list U .

Initialize: $\mathcal{O} = \{S\}$, $U = \emptyset$.

While \mathcal{O} is non-empty:

1. Extract $\mathcal{O}[1]$ into p ;
2. If p is not an existing split cell in \mathcal{O} :
 - 2.1 Travel to p along shortest path in M ;
 - 2.2 Add reachable free neighbors of p to \mathcal{O} (sorted by \bar{V});
 - 2.3 Add unreachable free neighbors of p to U ;
 (each free neighbor q of p acquires $\bar{V}(q) = \min\{\bar{V}(q), \bar{V}(p)+D\}$)
 - 2.4 If p is new split cell:
 - Stack p in \mathcal{O} and create sub-lists accordingly;
3. If an internal obstacle is detected:
 - ($\bar{V}(q)$ of a neighboring cell q of p satisfies $\bar{V}(q) < \bar{V}(p) - D$)
 - 3.1 Update \bar{V} by running BFS on M ;
 - 3.2 Update saddle lines in M , update \mathcal{O} and U ;
 - 3.3 Add a new saddle line into M ;
 - 3.4 Retreat along cable's path across new saddle line;
 End of while loop.
 Retreat along cable's path back to S .

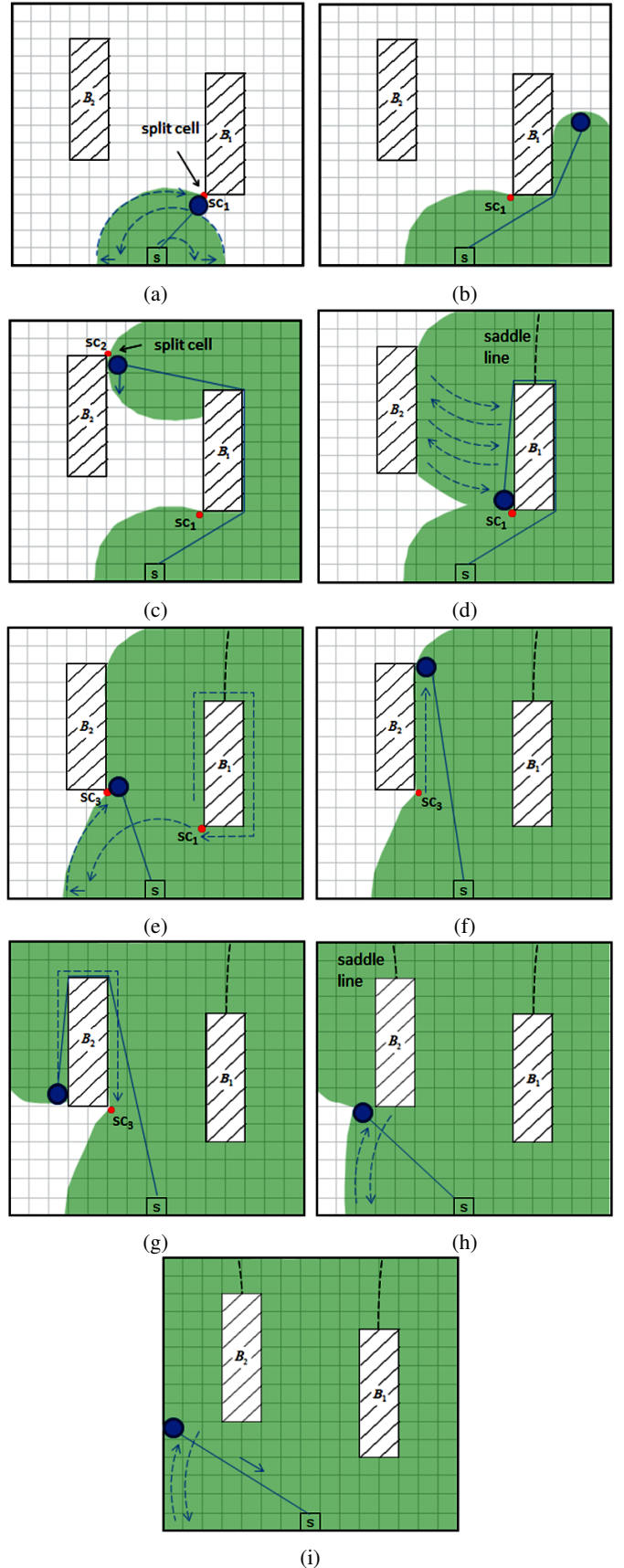


Fig. 6: Execution example of the TC algorithm (see text).

Execution example: Figure 6 depicts an execution example of the TC algorithm in a planar environment containing unknown obstacles B_1 and B_2 . The robot starts a BFS coverage from S until contact with B_1 is established (Figure 6(a)). The contact point is defined as a *split cell*, SC_1 . The robot chooses the corridor on the *right* of this cell, and continues with BFS coverage within this corridor (Figure 6(b)). The robot next encounters the obstacle B_2 and defines the contact point as a second *split cell*, SC_2 . It continues with BFS coverage in the corridor to the *left* of SC_2 (Figure 6(c)). The robot next identifies an internal obstacle by completing a full circumnavigation of B_1 (Figure 6(d)). At this instant the robot updates \bar{V} and defines the *saddle line* of B_1 (note that SC_2 is no longer a split cell). The robot travels back along the cable’s path across the saddle line, then back to SC_1 , from which it proceeds to cover the remaining free cells in the corridor on the *left* of this cell (Figure 6(e)). During this coverage the robot encounters a third *split cell*, SC_3 , and proceeds into the corridor on the *right* of this cell (Figure 6(f)). When the maximal cable length has been reached, the robot returns to SC_3 and proceeds to cover the alternate corridor at this cell (Figure 6(g)). The robot eventually circumnavigates the obstacle B_2 (Figure 6(h)), updates \bar{V} , and defines the *saddle line* of B_2 . When the robot completes the coverage of the current corridor, it returns to SC_3 , then retreats along the cable’s path back to S (Figure 6(i)). \circ

A full analysis of the TC algorithm appears in Ref. [14]. The analysis establishes that the TC algorithm covers all free cells accessible from S without crossing the robot’s cable. A competitive upper bound on the algorithm’s performance is next discussed.

V. Competitive Upper Bound on the TC Algorithm

This section derives an upper bound on the path length generated by the TC algorithm, using the following notion of *competitiveness*.

Definition 4 ([10]). *An online algorithm A solving a navigation problem P is **competitive** with ratio ζ if the cost of solving any instance p of P does not exceed ζ times the cost of solving p optimally using full off-line information, $\forall p \quad l(p) \leq \zeta \cdot l_{opt}(p)$.*

Assuming the robot travels with uniform velocity, we will measure the algorithm’s performance using the robot’s total path length, l , relative to the optimal off-line solution, l_{opt} . Let us first establish a lower bound on l_{opt} in fully known planar environments. Denote by n the total number of $D \times D$ cells reachable by a cable of finite length L . Since the purpose of the task is to cover all accessible cells, the minimal off-line path length is: $l_{opt} \geq nD$, where D represents the robot’s size as well as the distance the robot travels from one cell to another.

The following auxiliary lemma asserts that the robot travels a distance of at most $2L$ between successive cells in the environment.

Lemma V.1. *Under the TC algorithm, the robot travels in worst case a distance of $2L$ between any two uncovered free cells in the environment, where L is the cable’s length.*

Proof: Due to the cable’s finite length, the robot can only reach cells that require a travel distance of at most L from S . Hence, the worst case scenario occurs when the robot must travel via S between two cells located at a minimal distance of L from the base point. Under the TC algorithm, the robot travels through already-covered portion of the environment by way of the shortest path towards the next uncovered cell. Moreover, it is shown in Ref. [14] that the cable always remains within the already-covered portion of the environment. It follows that the worst case scenario shortest path is upper bounded by $2L$. \square

Theorem 1. *The total path length generated by the TC algorithm is upper bounded by $l \leq 2\frac{L}{D}l_{opt}$, where l_{opt} is the optimal off-line tethered coverage path length, L is the cable’s length, and D is the robot’s size.*

Proof: Assume the environment contains n accessible free cells. In each iteration of the TC algorithm, a single cell, $p = \mathcal{O}[1]$, is extracted. If p is not an existing split cell, the robot travels to this cell via the shortest path in the known environment grid M . Based on Lemma V.1, the robot will travel in worst case a distance of $2 \cdot L$ from its current cell to p . Since the total number of accessible cells is n , the TC algorithm’s total path length satisfies $l \leq 2nL$. The optimal off-line solution satisfies $l_{opt} \geq nD$, thus giving: $l \leq 2\frac{L}{D}nD \leq 2\frac{L}{D}l_{opt}$. \square

Note that the upper bound on the TC algorithm is proportional to the cable’s length, L . A longer cable enables a larger coverage area, and accordingly incurs a higher upper bound on l . Also note that the upper bound on l is inversely proportional to the robot’s size D . As D decreases, the area centered at S with radius L can be populated by a larger number of internal obstacles separated by narrow D -wide corridors. This would increase the number of possible paths from S to certain cells in the environment, and consequently increase the upper bound on l . In particular, the upper bound increases to infinity when D shrinks to zero. This represents a point robot, for which the area reachable by the cable’s finite length can be populated by an infinite number of internal obstacles. This is one more aspect of a well known result, recognized by several early papers (e.g. [5, 13]), that a point robot cannot perform competitive online tasks, while a robot of size $D > 0$ can competitively perform these tasks.

The upper bound is almost tight: There exists an adversarial environment of D -wide corridors in which the TC algorithm’s total path length satisfies $l \geq \frac{1}{6}\frac{L}{D}l_{opt}$ [14]. The upper bound specified in Theorem 1 is therefore almost a *tight* upper bound.

VI. Experiments

A series of experiments was conducted in order to verify and confirm the practical implementation of the TC algorithm. The

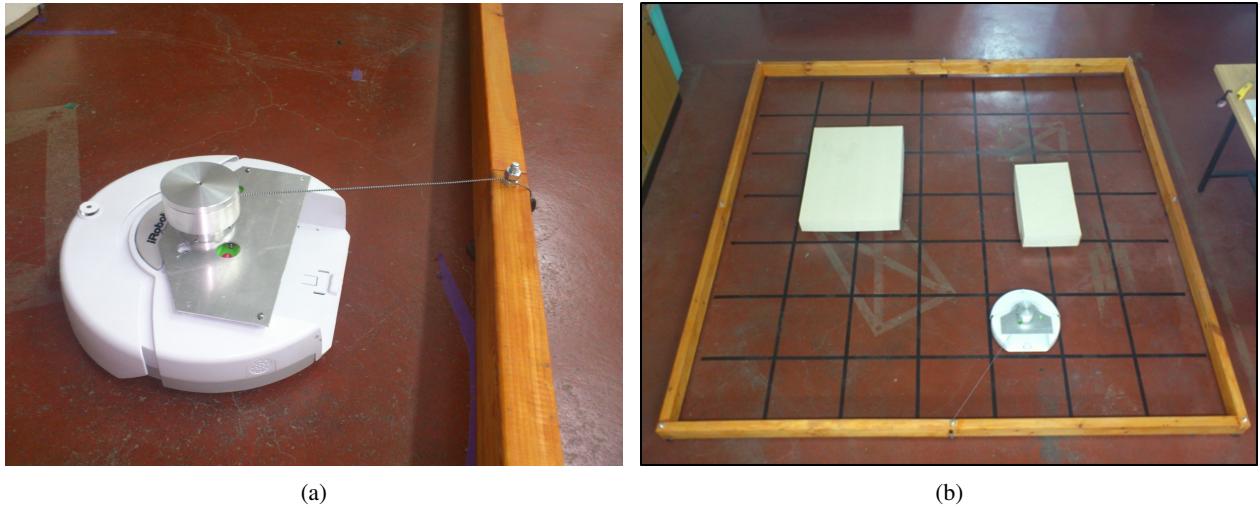


Fig. 7: Experiment platform: (a) Robot platform and cable retraction mechanism. (b) Environment used in final TC algorithm simulations.

experiments were performed using a mobile robotic platform attached by a cable of finite length to a fixed base point, with additional hardware mounted on the robot maintaining cable tension. The objective of the experiments was to verify the algorithm’s practical performance while executing online tethered coverage in an unknown planar environment populated by static obstacles. The experiments test the ability of standard on-board sensors, on-board memory, and on-board cable recoiling mechanism to perform online tethered coverage of unknown planar environments.

The robot platform used in the experiments is an *iRobot Create*, having a diameter of $D = 340$ mm (Figure 7(a)). Among the iRobot’s sensors are a front bumper that detects obstacle collision, and an IR sensor that detects obstacles in the robot’s close vicinity. The Create has no GPS, and its position is computed by odometric sensors mounted on the robot’s wheels. The cable used was a 2 mm diameter steel tether, having a finite length of $L = 5$ meters. The cable used in the experiments did not transfer any data or power to or from the robot. The cable recoiling mechanism consists of a torsion spring coiling within a metal housing. It was mounted on top of the iRobot platform, allowing both the release of the cable and rotation of the whole release mechanism with respect to the robot platform. Finally, the robot was placed within a 2.5×2.5 meter wooden frame that acted as four exterior walls. Additional wooden boxes that acted as internal obstacles were placed inside the wooden frame (Figure 7(a)).

The TC algorithm was then executed within this environment. The different trials tested the algorithm’s online performance in a real physical environment, and the interaction of the robot’s cable and sensors with the internal and external obstacles. Finally, a full coverage execution of the TC algorithm was executed in the environment shown in Figure 7(b). While executing the TC algorithm, the robot achieved coverage of over 95% of the environment’s free space (measured by counting the cells visited by the robot). Moreover, the robot was successful in identifying internal obstacles and redirect the cable’s path once a shorter path had

been detected.

The cable recoiling mechanism mounted on the robot performed as expected, and the cable was kept taut throughout the robot’s motion. Furthermore, the experiments validated the hypothesis stated in this paper concerning friction between the cable and obstacle edges. As presumed, this friction *aided* in keeping the cable taut between successive points of contact back to S . The main flaw was the poor localization system based on travel distance accumulated by the robot’s odometric sensors. As a result, the robot encountered difficulties when returning to previous locations such as *split cells*, as well as accurately detecting already covered areas when circumnavigating an internal obstacle. Last, the robot did not suffer from delay or show any signs of latency due to online computations.

VII. Conclusion

This paper was concerned with online tethered coverage problem. The tethered coverage problem is significantly more complex than the classical *untethered* coverage problem, due to the additional cable. Using the shortest path potential function for the environment, the paper first established an optimal off-line tethered coverage methodology which uses the potential function’s *saddle lines* and *split points*. Based on this methodology, the paper then introduced the TC algorithm that performs online tethered coverage using position and local obstacle detection sensors. The algorithm’s competitiveness was analyzed and shown to satisfy $l \leq 2 \frac{L}{D} l_{opt}$, where L is the cable’s length and D is the robot’s size. This upper bound is proportional to the cable’s length, L , and inversely proportional to the robot’s size D . Finally, experiments with a tethered recoiling mechanism illustrated the practical usefulness of the TC algorithm.

The following are some related open problems. First, the TC algorithm assumes the use of *local* obstacle detection sensors that detect obstacles in the cells adjacent to the robot’s current position. In principle, longer range obstacle detection sensors, such as optic or sonic sensors, will *not* give the robot an advantage in highly complex environments. However, in practical environments such as office buildings, city streets,

or outdoor landscapes, these more sophisticated sensors can provide useful information in order to plan a more efficient coverage paths. Thus, an adaptation of the TC algorithm to longer range obstacle detection sensors is an important practical challenge. Second, the upper bound on the TC algorithm is $l \leq 2\frac{L}{D}l_{opt}$. In order to establish that the *optimality* of the TC algorithm, one must additionally establish a universal (i.e. algorithm independent) lower bound on the online tethered coverage problem. The TC algorithm is optimal up to constants if the universal lower bound satisfies $l > c\frac{L}{D}l_{opt}$, where $c < 2$ is a numerical constant. The challenge of obtaining such a universal lower bound is currently under investigation.

Finally, tethered coverage is a specific example of a much wider family of navigation problems, concerned with *limited resource* navigation. This family of problems deals with various navigation tasks undertaken by mobile robots operating under conditions of limited resources. In this paper, the limited resource is the finite cable length. We are currently working on another limited resource problem, where a mobile robot travels with limited capacity *batteries* in unknown planar environments. We invite the robotics research community to help us identify other central limited resource navigation problems, in the hope of establishing a fundamental understanding of the special hardware, sensors, and algorithms required to solve these problems.

REFERENCES

- [1] *RoboMow by Friendly Robotics*. www.friendlymachines.com.
- [2] *iRobot Corporation*. www.irobot.com, USA.
- [3] P. Abad-Manterola, I. A. D. Nesnas, and J. W. Burdick. Motion planning on steep terrain for the tethered axel rover. In *IEEE Int. Conf. on Robotics and Automation*, pages 4188–4195, 2011.
- [4] E. Acar, H. Choset, Y. Zhang, and M. Schervish. Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods. *The Int. J. of Robotics Research*, 22(7-8):441–466, 2003.
- [5] R. Baeza-Yates, J. Cudrerson, and G. Rawline. Searching in the plane. *J. of Information and Computation*, 106: 234–252, 1993.
- [6] H. Choset. Coverage for robotics—a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31(1):113–126, 2001.
- [7] H. Enders, W. Feiten, and G. Lawitzky. Field test of navigation system: Autonomous cleaning in supermarkets. In *IEEE Int. Conf. on Robotics and Automation*, pages 1779–1781, 1998.
- [8] Y. Gabriely and E. Rimon. Spanning-tree based coverage of continuous areas by a mobile robot. *Annals of Mathematics and Artificial Intelligence*, 31:77–98, 2001.
- [9] S. Hert and V. Lumelsky. The ties that bind: Motion planning for multiple tethered robots. *Robotics and autonomous systems*, 17(3):187–215, 1996.
- [10] C. Icking, T. Kamphans, R. Klein, and E. Langetepe. On the competitive complexity of navigation tasks. In *Sensor Based Intelligent Robots, Lecture Notes in Computer Science*, volume 2238, pages 245–258. Springer Verlag, 2002.
- [11] T. Igarashi and M. Stilman. Homotopic path planning on manifolds for cabled mobile robots. *Algorithmic Foundations of Robotics IX*, pages 1–18, 2011.
- [12] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1990.
- [13] C.H. Papadimitriou and M. Yannakakis. Shortest paths without a map. *Theoretical Computer Science*, 84(1): 127–150, 1991.
- [14] I. Shnaps and E. Rimon. Online coverage by a tethered autonomous mobile robot in planar unknown environments. Tech. report, Dept. of Mechanical Engineering, Technion, <http://robots.technion.ac.il/publications.htm>, May 2013.
- [15] P. G. Xavier. Shortest path planning for a tethered robot or an anchored cable. In *IEEE Int. Conference on Robotics and Automation*, pages 1011–1017, 1999.
- [16] N. Xu, P. Brass, and I. Vigan. An improved algorithm in shortest path planning for a tethered robot. In *22nd Annual Fall Workshop on Computational Geometry*, 2012.
- [17] H. Yaguchi. Robot introduction to cleaning work in the east japan railway co. *Adv. Robotics*, 10 (4):403–414, 1996.