

Efficient Path Planning Algorithm for Mobile Robots Performing Floor Cleaning Like Operations

Vishnu G Nair ^{1*}

¹ Department of Aeronautical and Automobile Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka, India 576104

Email: ¹ vishnu.nair@manipal.edu

*Corresponding Author

Abstract—In this paper, we introduce an efficient path planning algorithm designed for floor cleaning applications, utilizing the concept of Spanning Tree Coverage (STC). We operate under the assumption that the environment, i.e., the floor, is initially unknown to the robot, which also lacks knowledge regarding obstacle positions, except for the workspace boundaries. The robot executes alternating phases of exploration and coverage, leveraging the local map generated during exploration to construct a STC tree, which then guides the subsequent coverage (cleaning) phase. The extent of exploration is determined by the range of the robot's sensors. The path generation algorithms for cleaning fall within the broader category of coverage path planning (CPP) algorithms. A key advantage of this algorithm is that the robot returns to its initial position upon completing the operation, minimizing battery usage since sensors are only active during the exploration phase. We classify the proposed algorithm as an offline-online scheme. To validate the effectiveness and non-repetitive nature of the algorithm, we conducted simulations using VRep/MATLAB environments and implemented real-time experiments using Turtlebot in the ROS-Gazebo environment. The results substantiate the completeness of coverage and underscore the algorithm's significance in applications akin to floor cleaning.

Keywords—Cleaning Path; Mobile Robots; Spanning Tree Coverage; Coverage Path Planning; Exploration; Sensors; Algorithm; Completeness; Non-Overlapping Coverage.

I. INTRODUCTION

Mobile autonomous robots have found diverse applications, including autonomous cleaning [1], lawn mowing [2], mine sweeping [3], among others. These applications entail fundamental challenges such as coverage path planning, exploration, localization, and mapping. Exploration refers to the process of autonomously navigating and uncovering an unknown or partially known environment. In the context of robotics, it involves a robot's ability to move through an area while building a map and gathering information about its surroundings. Mapping pertains to the creation of a representation of the environment in which a robot operates. This representation typically includes details such as obstacles, landmarks, and the robot's own location, allowing the robot to make informed decisions and navigate effectively. Coverage path planning is the task of determining an efficient and systematic path that a robot should follow to ensure it covers an entire area of interest. This is commonly used in applications like environmental monitoring and cleaning, where full coverage of an area is essential. In this paper, we focus on exploration, defined as the process of

gathering information about obstacles to generate a map used for subsequent cleaning operations. The extent of exploration is determined by the robot's sensor range, considering cells outside this range as unknown or unexplored. Our algorithm aims to generate a cleaning path that traverses all points within the workspace while ensuring there are no coverage gaps or overlaps. Thus, it must be both complete and non-repetitive, guaranteeing comprehensive coverage during the cleaning process. The benefit of this approach is that the robot is required to switch on its exploration sensors only during the exploration phase and they remain in off position during the coverage phase, thus drastically increases the battery utilization

Exploration, as indicated in references, is the fundamental process of acquiring information about the environment. The terrain acquisition problem is formulated as a continuous motion planning without imposing constraints on obstacle geometry, presenting two algorithms for acquiring planar terrains with obstacles of arbitrary shape, with performance estimates based on path length upper bounds is given in [4]. In [5] the authors focus on experimental investigations involving a mobile robot's map-building and exploration capabilities by employing a novel metric to assess map quality and tests various exploration strategies in different environments. The most promising results are observed in hybrid strategies that combine reactive navigation with map-based approaches. Yamuchi [6], extends frontier-based exploration to multiple robots, allowing them to share perceptual information while maintaining separate global maps and making independent exploration decisions. The approach enhances exploration efficiency through information sharing and maintains robustness in case of individual robot losses, as demonstrated in real-world office environments. In [7] the authors explore the use of motion planning techniques for acquiring range-images with mobile robots equipped with range sensors, with the goal of creating 3D representations of indoor environments efficiently. It presents two randomized algorithms, one based on set-coverage and the other on an incremental scheme, to minimize sensing operations and address image registration challenges. The study presented in [8] addresses exploration problems aiming to construct complete maps of unknown environments with the shortest possible path. Given the limited range of sensors available to robots, such as LIDAR or vision-based systems, choosing an optimal initial point is crucial to maximize the information gathered during exploration. Typically, exploration leads to the mapping of a



specific region. During this process, robots detect obstacles, a necessity in online coverage path planning. To achieve a comprehensive map of the region, robots must thoroughly cover the workspace, thus integrating the term "coverage" into exploration and mapping problem contexts. In CPP, the primary aim is to systematically cover the entire area of interest, whereas exploration and mapping emphasize the generation of an accurate representation of the environment. In references [9][10], the authors have presented a variety of path generation algorithms tailored for cleaning applications. These encompass optimal coverage algorithms and sensor-based boustrophedon-like Coverage Path Planning (CPP), as discussed by authors such as [13][14] among others. Within the domain of Morse decomposition-based algorithms, critical point detection poses a challenge due to the imperfect characteristics of sensors, particularly in unstructured workspaces [14]. Morse decomposition is a concept used in topological analysis and is particularly relevant to understanding the structure of a dynamical system or a space. It involves partitioning the space into regions called "Morse sets." Each Morse set represents a distinct topological feature or behavior of the system. These sets help in characterizing and understanding the dynamics, including attractors and repellers, within the space. Coverage Path Planning (CPP) algorithms are broadly categorized into offline, where prior information about the workspace is available, and online, where information is gathered during the task. In both cases, localization of individual robots is crucial and can be facilitated by Simultaneous Localization and Mapping (SLAM) algorithms. SLAM refers to the capability of a robot or autonomous system to create a map of an unknown environment while simultaneously determining its own position within that environment. In essence, SLAM allows a robot to explore and navigate through an unknown space while building a map and maintaining awareness of its own location in real-time [82]. This paper emphasizes a CPP methodology that optimizes performance by making use of whatever partial information about the workspace is available (partial map). The map is updated in an online manner by utilizing this partial data, contributing to the strategy of simultaneous exploration and coverage. The paper makes several valuable research contributions:

- The paper introduces an innovative algorithm that seamlessly combines the processes of exploration and coverage path planning in the context of robotics. This novel approach is a significant contribution to the field as it streamlines the robot's operation, making it more efficient and versatile in scenarios requiring both mapping and complete area coverage.
- The paper addresses the critical aspect of long-term robotic operation by emphasizing efficient battery usage. The selective activation of sensors during exploration phases conserves energy, prolonging the robot's operational lifespan. This contribution has implications for sustainability and cost-effectiveness in autonomous robotic systems.

The rest of the paper is organized as follows. Section II presents the motivation of the proposed algorithm followed by recent developments in coverage path planning strategies in section III. The problem statement is discussed in section

IV followed by proposed methodology in section V. Properties of the algorithm in terms of non-repetitiveness, completeness, battery usage and time taken are discussed in sections VI and VII respectively. Section VIII presents the results obtained followed by conclusion in section IX.

II. MOTIVATION

The motivation behind the development of the proposed path planning algorithm for floor cleaning robots in initially unknown environments stems from the growing demand for efficient, autonomous cleaning solutions. Traditional cleaning methods often fall short in terms of coverage, time efficiency, and adaptability to diverse and uncharted spaces. This limitation has become particularly evident in contexts such as commercial cleaning, industrial settings, and even domestic environments, where the need for comprehensive and autonomous cleaning solutions has never been more pressing. There are various shortcomings with the methodologies available at present. Most important one is the optimal usage of resource such as battery. In addition to it the available methods in the literature is either online or offline. In online methodology the robot does not have any a prior information regarding the workspace and the data such as obstacle position etc. are collected on the go. But, in offline strategy the robot already has complete information regarding the workspace. But this paper presents an online-offline methodology in which the online strategy is followed during the exploration phase and once it is finished then the coverage/cleaning is offline. This cyclic procedure from online to offline and back makes the proposed strategy a very efficient one. Also, there is a need to include the motion constraints of the robots in the algorithm. The robot can move easily horizontally and vertically but it will not be the case with diagonal motion. The division of workspace as 2D grids provides a more efficient robot motion within the workspace.

The major motivation for this algorithm is twofold. First and foremost, it addresses the imperative need for efficient path planning in initially unknown environments. Cleaning robots are expected to operate in spaces where prior knowledge of the layout is often lacking. Therefore, the ability to navigate, explore, and clean effectively without prior knowledge is essential. Our algorithm provides a practical and innovative solution to this challenge. Second, the motivation is driven by the urgency to conserve energy and optimize the operation of cleaning robots. Energy efficiency is a critical consideration, especially in the era of sustainability and the increasing demand for eco-friendly technologies. By minimizing battery usage through the offline-online scheme and intelligent sensor management, our algorithm aligns with the broader goal of energy conservation in robotics. Moreover, the versatility of our algorithm makes it relevant to a wide range of applications, not limited to floor cleaning. Its potential use in warehouse management, industrial automation, agricultural robotics as well as various other civil and defense applications. The need for such an algorithm is rooted in the need for innovative and energy-efficient solutions in the realm of robotics, with a specific focus on floor cleaning and the potential for broader applications.

III. RECENT DEVELOPMENTS IN COVERAGE PATH PLANNING STRATEGIES

A 'partition and cover' methodology utilizing Voronoi partition is adopted to facilitate natural passive cooperation among robots, aiming to prevent task duplicity in [18]. Virtual nodes, deployed into CVC, are leveraged to facilitate efficient Voronoi partitioning, and subsequent robot coverage, guided by a gradient-based control law [19]. Another significant contribution involves addressing topologically disconnected cells due to obstacles [20]. Similarly, the problem of multi-robotic coverage is tackled, focusing on covering an area of interest using multiple sensors mounted on autonomous vehicles such as aerial or ground mobile robots [21]. The concern of repetitive coverage, along with complete coverage of a given area, when utilizing multiple robots, is addressed through a Manhattan distance-based Voronoi partitioning scheme [22]. A dynamic workspace re-allocation strategy is studied to optimize the area covered by each robot, particularly in the presence of obstacles, ensuring efficient coverage in [23]. Another noteworthy contribution involves describing a Voronoi-based path generation (VPG) algorithm for energy-constrained mobile robots, specifically unmanned aerial vehicles (UAVs) [24]. Additionally, the researchers delve into precision farming and the application of unmanned aerial vehicles (UAVs) to solve agriculture and animal husbandry issues [25]. In [28], a bipartite cooperative coevolution algorithm is proposed to optimize inter-area and intra-area path planning components, addressing battery limitations and ensuring effective coverage. Furthermore, a novel mechanism to generate non-revisiting uniform coverage paths on arbitrarily shaped object surfaces is presented in [27]. In [30], the limitations of CPP algorithms in handling unexpected changes in the coverage area are addressed through a novel adaptive CPP approach. In [29] the authors proposed Turn-minimizing Multirobot Spanning Tree Coverage Star algorithm, building upon the MSTC* algorithm. TMSTC* partitions the map into minimum branches resembling tree branches, converting the problem into finding a maximum independent set in a bipartite graph. In [29], the authors address coverage path planning for autonomous heterogeneous UAVs over a bounded number of regions. A centralized coverage algorithm for UAV formation in aerial photography is proposed in [31]. A significant contribution is made to equidistant tool path planning on curved freeform surfaces, particularly for robotic machining tasks [32]. In [33] a non-user-oriented path planning scheme using a family of spiral-shaped curves for moving aerial base stations (ABSs), aiming to optimize mechanical energy and achieve uniform coverage across a cell, demonstrating significant power consumption savings compared to existing radial paths is presented. In [36], the authors propose a non-user-oriented path planning scheme based on spiral-shaped curves, aiming to achieve energy efficiency and uniform coverage for all users within a cell. article addresses path planning for multiple UAVs to achieve sweep coverage, especially focusing on forest fire early warning and monitoring. A Predator-Prey reward-based Q-Learning CPP, overcoming local optima challenges is studied in [35] and [36] introduces a visibility-based path planning (VPP) heuristic for optimizing visibility during UAV flights.

The collaborative coverage improved BA* algorithm is proposed in [37]. In [38] the authors studied the utilization of unmanned aerial vehicles in scanning or surveying multiple regions.

The study in [39] introduces a robotic system for automated breast ultrasound scanning that ensures full and uniform coverage by employing 3-D point cloud searching and addressing tissue deformation through a contact force-strain regression model and probe-tissue interaction adjustments, ultimately demonstrating effective maintenance of probe posture and contact force during scanning. In [40] the authors present Cowbot, an autonomous weed mowing robot designed for maintaining cow pastures, addressing the challenge of unknown weed distribution by developing online planning algorithms that optimize path length based on weed detection, validated through field and simulation experiments demonstrating significant path length reduction compared to traditional approaches. In [41] the work presents an integrated approach for coverage path planning with unmanned aerial vehicles (UAVs), optimizing both mobility and camera orientation to achieve comprehensive coverage based on visibility constraints, addressed as a constrained optimal control problem and solved using mixed integer programming optimization. In [42] three Mixed Integer Linear Programming formulations aimed at optimizing the path of a mobile sensor node to cover "coverage holes" in a Wireless Sensor Network, demonstrating superior results compared to existing approaches through simulations and comparisons is presented. Some recent papers address artificial intelligence methods such as genetic algorithms (GA), ant colony optimization (ACO), Voronoi diagrams, and clustering methods. Another research focus pertains to optimizing coverage path planning for robotic single-sided dimensional inspection of free-form surfaces. By introducing a non-random targeted viewpoint sampling strategy, a significant reduction in cycle-time for the inspection task is achieved, enhancing overall solution quality. Additionally, a study is conducted on complete coverage path planning for reconfigurable robots using deep reinforcement learning, offering an optimal set of shapes for reconfigurable robots to maximize area coverage while minimizing energy consumption. A hierarchical approach based on maximum cumulative detection reward is proposed for path planning in the search for static targets by autonomous underwater vehicles (AUVs) in ocean environments, showcasing superior performance through simulation results. Moreover, an innovative approach is presented for UAV coverage path planning with real-time data transfer during flight over 5G networks, significantly reducing data losses and improving flight time. Also, a two-phase method is proposed for optimizing the coverage path planning of multiple UAVs in maritime search and rescue (SAR) operations, effectively minimizing search area extensions and completion time [43-48]. Various studies propose innovative solutions catering to diverse applications. One approach focuses on optimizing trajectory length and turns for inspection operations using a back-and-forth pattern, employing a tree search model based on polygon width and sweep line to navigate complex polygons [49]. Another tackles path planning for underwater vehicle manipulator systems (UVMS) in aquatic organism capture, integrating a novel cost modeling algorithm and

predictive network models for efficient path planning [50]. Addressing the pressing need for disinfecting public areas during the COVID-19 pandemic, a coverage path planner for a spraying drone is introduced, revolutionizing existing planners by considering a sprinkler system and realistic dispersion volumes [51]. Furthermore, agriculture presents its unique challenges in CPP due to varying field shapes and sizes. A generic CPP method is proposed, accompanied by benchmark datasets and quantitative measures to enhance agricultural efficiency [52]. In the context of space exploration and servicing, a novel kinodynamic inspection path planner is presented, addressing proximity operations and achieving efficient inspection trajectory planning for space structures [53]. Similarly, in the maritime domain, an autonomous coverage path planning model based on reinforcement learning significantly improves search efficiency for maritime search and rescue (SAR) operations [54]. When it comes to marine mapping, cooperative surveying of offshore seabed regions is vital, and a sequential algorithm is proposed to facilitate accurate boundary generation and coverage path planning for multiple unmanned surface vehicles (USVs) [55]. Moreover, energy-efficient online coverage path planning is a critical consideration for robotic platforms operating in unknown spaces with limited energy. A novel solution based on contour following shows promising results in optimizing energy consumption and ensuring efficient coverage [58]. Ant Colony Optimization Algorithm for CPP is proposed [59] considering energy consumption and unique characteristics, outperforming existing algorithms by at least 10% in terms of energy efficiency across various scenarios. Another research focuses on unmanned surface vehicles, presenting a coverage path planning algorithm based on improved biological inspired neural network. It enhances coverage efficiency while ensuring 100% coverage, demonstrating superior path length and repetition rate [60]. Additionally, a Complete Coverage Path Planning (CCPP) scheme is introduced for robot operating system-based robots, combining sub-area division, "S" shape path planning, and dynamic tracking to improve coverage ratio significantly [61]. A multi-agent coverage path planning algorithm inspired by social behaviors in the biological world is also proposed, effectively addressing uncertain and random decision-making, showing advantages in make span and coverage repetition rate [62]. Another work introduces a novel Clustering Obstacles (CO)-based path planning algorithm for mobile robots using quintic trigonometric Bézier curves, achieving better collision-free paths compared to existing algorithms [63]. Moreover, a hybrid navigation system for mobile robots is presented, combining autonomous positioning and path planning, showcasing effectiveness in unstructured scenarios [64]. In [65], a complete hybrid navigation system for a class of mobile robots with load tasks and docking tasks is presented. The challenges and potential solutions for path planning in Multiple Robotic Systems (MRS) are surveyed, emphasizing the need for more research and practical applications in dynamic environments is presented in [66]. An algorithm based on neural dynamics is proposed to achieve complete coverage of a grid-based environment while considering collision avoidance and dead-end situations for mobile

robots, demonstrating improvements in collision-free effective path planning in [67]. Another research addresses connectivity and area coverage in Wireless Sensor Networks using mobile robots to optimize sensor positions, improving data transfer to the base station [68]. In [69] the authors comprehensively introduce intelligent path planning technologies in the context of underwater vehicle path planning, emphasizing collaborative and coverage path planning. Another research paper [70] presents a scalable technique that couples and manipulates chaotic systems to improve scanning coverage efficiency in unknown environments, demonstrating a significant performance boost compared to existing planners. Another innovative approach, D-RRT*, addresses the slow convergence issue of RRT* algorithm by focusing on the direction of the goal, resulting in improved performance metrics in cluttered 2D environments [71]. A bio-inspired topological navigation model for aerial mobile robots is proposed, mimicking animal-like navigation behaviors based on topological spatial representation, achieving efficient navigation and higher localization accuracy in challenging real-world visual environments in [72]. A comprehensive study on MRS coordination, categorizing various coordination approaches based on different dimensions and analyzing the factors affecting MRS coordination and efficiency is presented in [73]. For four-wheel independently driven steered mobile robots, a multiple mode-based navigation system is proposed, incorporating kinodynamic interior-exterior cell exploration planning and fuzzy adaptive receding horizon control for efficient motion planning and trajectory following in [74]. In agricultural robotics, a novel full coverage path planning method for harvesting robots is introduced, significantly improving adaptability to the working environment and performance of harvesting robots in [75]. An improved Dynamic Window Approach (DWA) based on Q-learning is proposed in [76] for mobile robot path planning in unknown environments, showcasing higher navigation efficiency and successful rate in complex unknown environments. In border surveillance operations, an adaptive border patrol process is introduced based on a partially observable Markov decision process, leveraging partial information on trespassing agents to dynamically determine patrol paths, resulting in improved trespasser detection rates in [77]. In [78], an efficient coverage path planning algorithm based on Boustrophedon motions and rapid Voronoi diagram is presented, demonstrating superior time efficiency and coverage compared to conventional heuristic methods. In manipulator robot applications, [79] provides an in-depth review of fuzzy logic approaches for collision-free path planning, highlighting their efficiency and applicability in complex and cluttered workspace. For unmanned aerial vehicle surveying missions, an optimal edge-vertex back-and-forth path planner is proposed in [80,81], considering starting and ending points, thus providing an efficient solution for coverage path planning.

The non-ideal behavior of sensors, especially in unstructured workspaces, necessitates further investigation to improve critical point detection. It is imperative to mitigate the impact of imperfect sensor data to extract more accurate critical points. In the context of online CPP algorithms, short-range sensors suffice as obstacle detection is primarily

required in close proximity to the robot. On the other hand, in an offline scenario where the complete coverage path is preplanned, the robot can seamlessly follow it without unnecessary halts. While the coverage problem can conceptually utilize various robot coverage algorithms such as Spanning Tree Coverage (STC) [12], Boustrophedon [11], etc., the specific algorithm employed to solve the problem is contingent on the underlying Coverage Path Planning (CPP) algorithm. In this paper, we adopt the offline STC [11,18] as the underlying CPP algorithm to illustrate the proposed coverage problem. Subsequently, we present an algorithm that effectively addresses and solves this particular coverage problem.

IV. PROBLEM SETTING

The Spanning Tree Coverage (STC) algorithm holds significant promise for implementation in cleaning robots, particularly when the robot's initial and final locations are identical. In this algorithm, the entire region slated for coverage or cleaning is divided into square cells, referred to as major cells, each measuring 4 times the size of the cleaning tool ($2D \times 2D$). These major cells are composed of smaller cleaning tool-sized cells known as minor cells ($D \times D$). During the exploration phase, a local tree is constructed using the $2D \times 2D$ cells as nodes, and the robot's cleaning path is planned through the $D \times D$ cells. However, a notable drawback of the STC algorithm is its tendency to entirely avoid even slightly incomplete major cells, thus neglecting them in the cleaning task. This omission can lead to coverage gaps within the workspace. Consequently, the STC algorithm falls under the category of approximate cellular decomposition-based Coverage Path Planning (CPP) algorithms, aligning with the classification proposed by Choset in his 2001 paper [9]. To achieve complete cleaning, it is essential for the robot to clean all the $D \times D$ cells thoroughly. Redundant cleaning of cells should also be minimized to optimize the utilization of available resources effectively.

The problem setup is depicted in Fig. 1, where the workspace is divided into square cells, each being four times the size of the cleaning tool. Each of these larger cells comprises four cleaning tool-sized cells. During the exploration phase, the entire area is categorized into explored (occupied or free) and unexplored regions, with frontier cells serving as the boundary between the explored and unexplored areas. In the figure, the workspace is partitioned into square cells of size 4 times the cleaning tool size (shown with thick long dashed boundary) and each of these consists of four cleaning tool sized cells (shown with thin dashed line boundary). Obstacle is shown in black. During exploration, the entire area can be classified into explored (occupied or free) and unexplored. Explored free cells are shown in grey and unexplored in white. Frontier cells separates explored from unexplored. In the diagram F is for frontier cell, E for explored, U for unexplored. This paper does not delve into the specific type or characteristics of the sensors utilized for exploration. The proposed algorithm is designed to be sensor-agnostic, capable of functioning effectively with various sensor types. The algorithm primarily operates on the data acquired through exploration sensors. During the cleaning path generation, sensors are not required, and they remain in

an OFF state throughout the cleaning process. Additionally, the paper assumes that the robots are localized, allowing for the potential use of Simultaneous Localization and Mapping (SLAM) or similar localization algorithms.

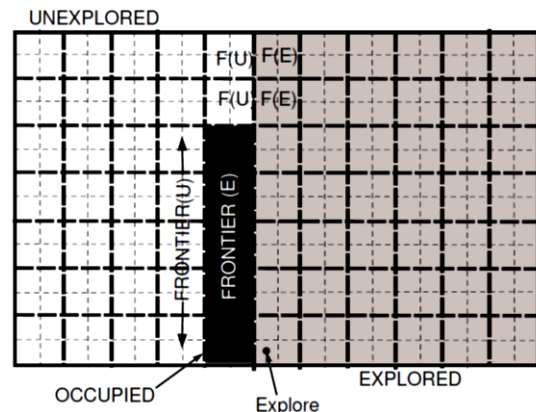


Fig. 1. The workspace is partitioned into square cells of size 4 times the cleaning tool size (shown with thick long dashed boundary) and each of these consists of four cleaning tool sized cells (shown with thin dashed line boundary). Obstacle is shown in black. During exploration, the entire area can be classified into explored (occupied or free) and unexplored. Explored free cells are shown in grey and unexplored in white. Frontier cells separates explored from unexplored. In the diagram F is for frontier cell, E for explored, U for unexplored.

V. PROPOSED ALGORITHM

In the proposed cleaning algorithm, both exploration and mapping procedures are integrated. In this approach, robots explore and obtain a partial map of the workspace, using this information to cover the area. This cycle is repeated until the entire region is covered, resulting in a complete map and a fully covered (cleaned) workspace. During the exploration and mapping process, robots must perform both path planning and mapping tasks using onboard sensors. Path planning involves deciding the future points for exploration to maximize the mapped (explored) area. The robot executes the coverage process based on the path deduced by the coverage strategy but needs to determine where, along this path, exploration needs to be conducted. Following the exploration phase, a local map of the region is generated, utilized to design and execute a coverage (cleaning) path based on the selected coverage algorithm, in this case, STC. During the robot's traversal to specific locations, explained in subsequent sections, the exploration phase is reinitiated. This iterative process continues until the entire workspace is thoroughly cleaned. It's reiterated that precise localization information is assumed to be available, given that this paper primarily focuses on integrating exploration and coverage to generate a comprehensive, non-overlapping coverage (cleaning) path. For real-time applications, localization techniques such as Simultaneous Localization and Mapping (SLAM) can be integrated. Fig. 2 depicts the cleaning problem, where the exploration process generates a map, subsequently used by the CPP algorithm to establish the cleaning path. As the robot follows this path, it combines cleaning and exploration tasks by conducting exploration at specific points along the cleaning route. Flow Chart 1 (Fig. 3) outlines the proposed algorithm employing the offline STC CPP algorithm. The entire process is represented as a pseudocode in Algorithm 1.

Algorithm 1 Exploration and Coverage Pseudocode for Autonomous cleaning Robots

```

Initialize the workspace and robot
Initialize Workspace
Create Empty Map
Initialize Robot
while Workspace is not fully covered do
  Generate a coverage path for the explored region
  CoveragePath ← GenerateCoveragePath(ExploredRegion)
  for each Cell in CoveragePath do
    MoveRobotToCell(Cell)
    ExplorationResults ← ExploreCell(Cell, LIDAR)
    UpdateMap(ExplorationResults)
    FrontierCells ← DetectFrontierCells(ExplorationResults)
    if FrontierCells then
      MST ← CreateMinimumSpanningTree(FrontierCells)
      AppendMSTToCoveragePath(MST)
function GENERATECOVERAGEPATH(ExploredRegion)
  CoveragePath ← []
  while UnvisitedCellsExist(ExploredRegion) do
    NextCell ← FindNearestUnvisitedCell(ExploredRegion)
    PathToCell ← PlanPath(CurrentRobotPosition, NextCell)
    AppendPathToCoveragePath(CoveragePath, PathToCell)
    MoveRobotToCell(NextCell)
    MarkCellsAsVisited(NextCell)
  return CoveragePath
function MOVEROBOTTOCELL(TargetCell)
  PathToTarget ← PlanPath(CurrentRobotPosition, TargetCell)
  for each Cell in PathToTarget do
    MoveRobot(Cell)
  CurrentRobotPosition ← TargetCell
function EXPLORECELL(Cell, LIDAR)
  CellStatus ← LIDARScan(Cell)
  ExplorationResults ← { "CellStatus": CellStatus }
  return ExplorationResults
function UPDATEMAP(ExplorationResults)
  CellStatus ← ExplorationResults["CellStatus"]
  Map.Update(CellStatus)
function DETECTFRONTIERCELLS(ExplorationResults)
  FrontierCells ← []
  CurrentPosition ← ExplorationResults["CurrentPosition"]
  for each NeighboringCell in NeighboringCells(CurrentPosition) do
    if IsUnexplored(NeighboringCell) and NeighboringCell not in FrontierCells then
      FrontierCells.append(NeighboringCell)
  return FrontierCells
function CREATEMINIMUMSPANNINGTREE(FrontierCells)
  MST ← Set()
  SortFrontierCellsByEdgeWeight(FrontierCells)
  DisjointSet ← InitializeDisjointSet(FrontierCells)
  for each Edge in FrontierCells do
    u, v ← Edge
    if FindSet(u, DisjointSet) ≠ FindSet(v, DisjointSet) then
      MST.add(Edge)
      UnionSets(u, v, DisjointSet)
  return MST

```

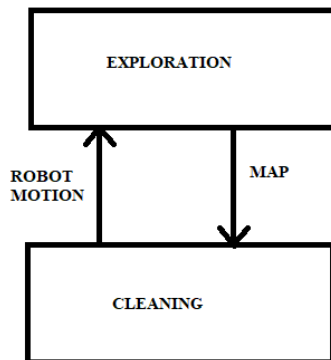


Fig. 2. The proposed methodology integrates both exploration and coverage path planning, where exploration is utilized to generate a map of the environment, and the coverage path is determined for the robot's motion. As the robot traverses the coverage path, it conducts exploration at specific locations to enhance its understanding of the surroundings.

The data acquired from sensors during each exploration phase is utilized to construct a Minimal Spanning Tree (MST) [12] employing the well-established Kruskal algorithm. This

MST is formed over the graph created by the larger cells, each being four times the size of the cleaning tool. The actual cleaning path followed by the robot traverses through the smaller cleaning tool-sized cells, navigating along the edges of the MST. The robot undergoes two primary operative phases: the exploration phase and the cleaning phase. The transition between these phases occurs when the robot reaches a F(E) cell with a neighboring F(U) cell. At the conclusion of the exploration phase, newly explored cells are incorporated into the existing known workspace, and the explored cell list is updated. Simultaneously, the MST is updated to accommodate these newly explored cells. The cleaning process then recommences from the cell where the exploration phase commenced, which is the present current cell. An important feature of this algorithm is that the robot only needs to activate the exploration sensors during a limited number of exploration phases, effectively conserving battery power. During the cleaning operation, the sensors remain in the OFF state.

To enhance comprehension of the proposed algorithm, an illustrative scenario is depicted in Fig. 5. The entire workspace slated for cleaning is divided into major cells, each four times the size of the cleaning tool. In this scenario, there are a total of 36 such cells. Among these, cells 6, 8, 11, 12, 13, 14, 16, and 17 are marked as obstacle-occupied, shown as darkened cells in the figure. The robot's initial position is assumed to be within cell number 18, represented by a dot in the figure. As elucidated earlier, the initial action of the robot is to activate its exploration sensors and conduct exploration starting from its current cell. Once the exploration phase is completed, the sensors are turned OFF. The results of this exploration phase are presented in Fig. 6(a). In this figure, the white region represents the explored free area (unoccupied by obstacles), and the cells that were occupied by obstacles and became known after this exploration phase are indicated in blue (cells 12, 13, 14, 16, and 17). At the initial phase of the process, all cells are assumed to be unknown and unexplored. Also, all the frontier cell lists are empty. Each exploration involves some steps. Steps in an exploration instance is shown as Fig. 4.

After the exploration phase concludes, the robot transitions to the coverage phase. Within the known region, a local spanning tree is constructed (depicted by red lines in the figure). This tree traverses the large cells, represented by red dots. The blue line in Fig. 6(b) illustrates the resulting coverage path generated as the robot moves along this tree. The first frontier known cell (F(E) cell) is major cell 21 since its adjacent cell, 15, is a F(U) cell. Upon reaching cell 21, the coverage action halts, and the next exploration phase commences. During this second exploration phase, additional cells become known (explored) and are depicted in white in the figure. A new spanning tree is constructed through these newly discovered cells and is appended to the initial tree (Fig. 6(c)). This iterative process continues, with subsequent F(E) cells initiating new exploration phases. For instance, the next F(E) cell is cell number 3, adjacent to F(U) cell 2, and the next exploration phase begins upon the robot's arrival at cell 3 (Fig. 6(c)). The algorithm consistently generates new spanning trees based on the cells that become known after each exploration phase, ensuring a thorough cleaning path.

During the coverage phase, as the robot moves from cell 0 to cell 1, it triggers another exploration phase, revealing cell 7. A spanning tree edge is then added to cell 7 since it is unoccupied. The robot eventually returns to the initial cell, cell 18, marking the conclusion of the coverage. The cleaning process terminates when the robot reaches a minor cell, and the next available coverable/cleanable minor cell is the initial cell itself.

Given that this condition is satisfied at cell 18, the cleaning process concludes, ensuring complete exploration and cleaning of the entire region without any gaps or overlaps (Fig. 6(d)).

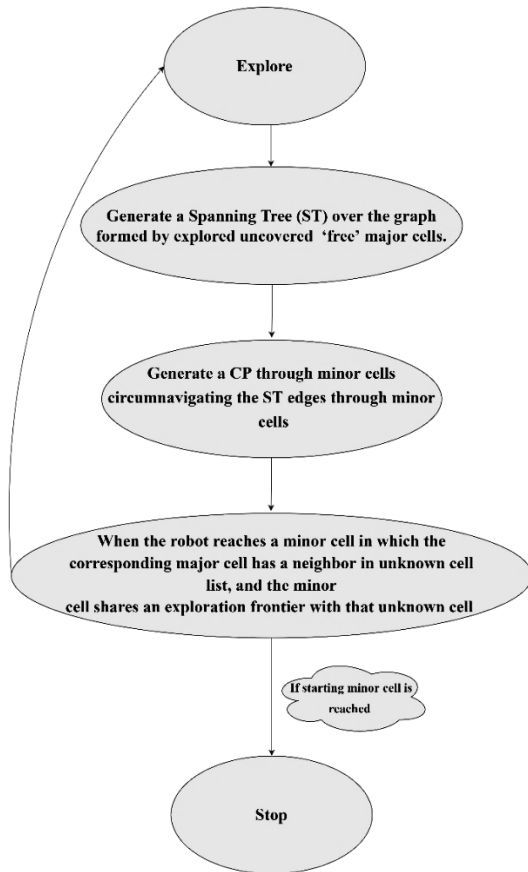


Fig. 3. Flowchart 1: Flowchart of the proposed algorithm using off-line strategy.

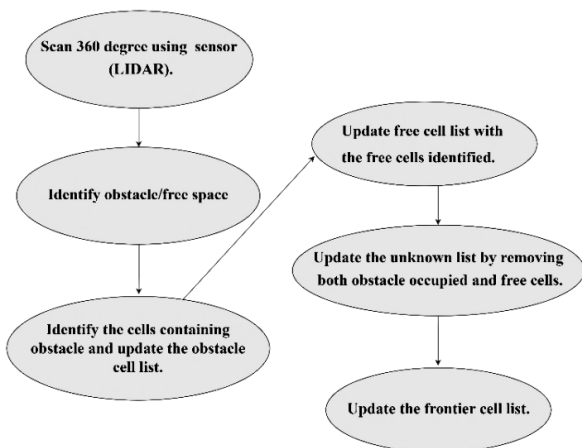


Fig. 4. Flowchart 2: Flowchart depicting the exploration algorithm used in the proposed coverage algorithm shown in flowchart 1.

5	11	17	23	29	35
4		16	22		
3		15			
2	8	14			
1	7	13			
0	6	12	18	24	30

Fig. 5. Illustrative scenario. The entire workspace to be cleaned is divided into cells, 4 times the size of the cleaning tool. There is a total of 36 such cells. The darkened cells with numbers 6; 8; 11; 12; 13; 14; 16, and 17 are obstacle occupied. The initial position of the robot is assumed to be inside cell number 18 (represented by a dot in the figure). [17].

The advantages of this methodology can be summarized as follows:

- a) **Power Efficiency:** One of the primary advantages is its efficient power consumption strategy. By deactivating exploration sensors during the cleaning phase, it significantly reduces power usage, which is crucial for prolonged robotic operations.
- b) **No Prior Knowledge Required:** Unlike some existing approaches that rely on prior knowledge of the workspace, this algorithm can function effectively without any prior information about the environment. This makes it versatile and adaptable to various scenarios and changing environments.
- c) **Comprehensive and Non-Repetitive Coverage:** It ensures thorough and non-repetitive coverage of the workspace, even in the presence of obstacles. This addresses the challenge of ensuring that every area is covered and avoids unnecessary revisiting of already explored regions.
- d) **Optimal Resource Utilization:** The algorithm optimizes the usage of resources, particularly battery power. It activates exploration sensors only during the exploration phase, conserving power during cleaning operations.
- e) **Valuable Map Generation:** At the end of the cleaning process, it generates a detailed map of the workspace as a by-product. This map can be used for various purposes, such as environment analysis, monitoring, or planning future tasks, providing an additional valuable contribution.

In summary, this algorithm offers a comprehensive solution to the challenges of coverage path planning. It efficiently manages power resources, adapts to dynamic environments without prior knowledge, and ensures complete and non-repetitive coverage. Additionally, it contributes to the creation of a valuable workspace map, further enhancing its utility and providing a versatile and efficient approach to coverage path planning in robotics.

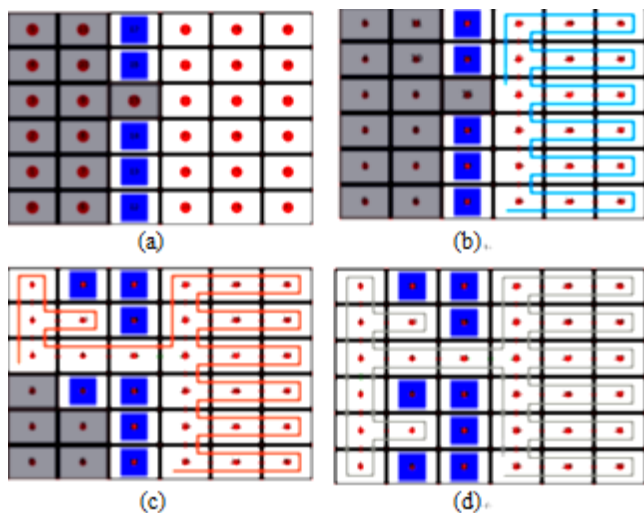


Fig. 6. Various stages of operation of proposed algorithm. a) Exploration phase 1. b) Exploration phase 2 after covering all the coverable explored cells. c) Exploration phase 3 and d) Exploration phase 4 and 100 per coverage/cleaning. The free known (explored) cells are shown in white and unknown cells in grey. Known obstacle occupied cells are in blue. The large cell nodes are represented using red dots whereas the spanning tree edges are shown using red lines. The actual coverage path of the robot is shown as blue lines.

VI. COMPLETENESS AND NON-REPETITIVENESS OF THE PROPOSED ALGORITHM

As demonstrated in the preceding sections, the proposed algorithm ensures thorough cleaning of the workspace, leaving no gaps or overlaps in the cleaning path. The algorithm comprises two sub-algorithms: one for exploration and the other for coverage path planning (offline mode). The overall performance of the algorithm hinges on the specific methodologies chosen for exploration and cleaning. In this context, we have opted for a frontier-based exploration strategy coupled with offline Spanning Tree-based coverage path planning algorithms.

A. Property 1: For Exploration

- The workspace to be cleaned (C) contains a sub-area $C \setminus O$, which is obstacle-free and topologically connected.
- Achieving complete exploration of C is possible if the entire $C \setminus O$ region is covered.
- Frontier cells act as connectors between known (explored) and unknown (unexplored) cells.
- There always exists a pair of adjacent obstacle-free $F(U)$ - $F(E)$ cells, forming an exploration window, as long as the present unexplored region contains a subset of $C \setminus O$.
- The coverage path with the offline spanning tree coverage algorithm encounters an exploration window within the present unknown (unexplored) cell.
- In the absence of exploration windows in the frontier cells, the exploration is considered completed.

B. Property 2: For Coverage

- An initial Spanning Tree (ST) is generated within the initially explored workspace.

- Upon new regions becoming known after the exploration phase, a local ST is generated using this newly explored region. Combining this tree with the old one results in a graph that remains a tree, spanning the new region, which is the new total explored region.
- When the entire workspace, C , is explored, the large tree generated by combining all the small trees forms an ST within the $C \setminus O$ region.
- According to [12], the ST coverage path will always be complete and non-repetitive.

Based on the aforementioned properties, it is evident that the proposed methodology ensures a comprehensive and non-repetitive cleaning of the workspace, $C \setminus O$, within C . Several individual claims within these two properties may seem evident, but they would benefit from a formal proof. However, it's important to note that this paper does not delve into a detailed formal analysis or rigorous mathematical proofs for these claims. The focus here is on presenting the methodology and explaining its workings, assumptions, and properties to establish a conceptual understanding and the potential effectiveness of the proposed algorithm.

VII. BATTERY EXPENDITURE AND TIME TAKEN

Efficiency in mobile robot operations, particularly in terms of time to task completion, number of turns, and sensor usage, is paramount due to the dependency on battery power. The proposed algorithm addresses these critical factors to maximize the robot's efficiency. The total path length taken by the robot, non-repetitiveness in the coverage/cleaning path, and the number of turns required align with high-performance algorithms. Additionally, the proposed algorithm offers an advantage in obstacle detection. In standard online Spanning Tree Coverage algorithms, obstacle detection entails continuous sensor activation at each cell, demanding constant power usage for turning (270°) and sensor operation. This results in substantial power consumption per cell. However, the proposed algorithm optimizes battery usage by activating sensors only during exploration phases. Once a local spanning tree is generated in the explored region, the robot can deactivate its sensors, allowing uninterrupted cleaning motion until the next exploration window is reached. This significantly reduces battery usage and speeds up the cleaning process. As mentioned earlier, the proposed algorithm can seamlessly integrate with any coverage algorithm by replacing the specific spanning tree coverage algorithm used in this paper. The outlined properties hold true regardless of the chosen coverage algorithm, emphasizing the adaptability and effectiveness of the approach.

VIII. RESULTS

This section provides an overview of the simulation experiments conducted to showcase the effectiveness of the proposed algorithm. The simulations were conducted in three phases, utilizing the V-rep/MATLAB environment and involving the use of Turtlebot in the ROS/Gazebo simulator. The simulation environment for this research leverages two key platforms: V-rep/MATLAB and ROS/Gazebo. V-rep (Virtual Robot Experimentation Platform) is a versatile 3D simulation tool used for modeling, simulating, and

controlling robots. It is preferred for its user-friendly interface, comprehensive library of robot models and sensors, and its support for real-time physics simulation. ROS (Robot Operating System) is an open-source framework that provides a robust middleware for building and controlling robot systems. It's chosen for its extensive library of packages, communication tools, and its compatibility with various hardware platforms. Gazebo is a 3D simulation environment integrated with ROS, offering realistic physics-based modeling for accurate robot simulation. This combination of ROS and Gazebo allows for seamless development, testing, and validation of robotic systems before actual deployment. The selection of these platforms was based on their established reputations within the robotics community, their support for a wide range of robot hardware, and their capacity to facilitate comprehensive and realistic simulations. The simulation experiments aimed to provide valuable insights into the algorithm's performance, efficiency, and suitability for practical implementation in a real robotic cleaning application. These phases helped in validating the algorithm's functionality and ensuring its effectiveness in different simulation environments.

The initial phase of simulation involved a graph-level analysis where the robot's actual motion was not taken into account. The simulation focused on generating a spanning tree using the explored free large (major) cells, and the robot's path was established over the graph created using the explored free small (minor) cells. The workspace for this simulation is illustrated in Fig. 7, divided into 16 large cells. The presence of obstacles (depicted in black) in the workspace leads to topological disconnectivity, resulting in permanently unreachable areas for the robot. For instance, starting from a specific cell as shown in the figure, the robot may encounter unreachable cells. Such situations are a consequence of the obstacle layout and cannot be resolved by any algorithm. The cleaning algorithm's objective is to clean all accessible reachable areas without coverage gaps or overlaps and terminate upon completing the cleaning process. In this simulation, the robot initially lacks information regarding the nature of obstacles within the workspace, only being aware of the boundary. The cleaning procedure executed by the robot after each exploration phase is depicted in Fig. 8 and Fig. 9. In Fig. 8(a), after the first exploration phase initiated from the initial cell, a local/partial map of the environment is generated. In Fig. 8(b), the robot enters the second exploration phase when encountering an exploration window (a F(E) cell adjacent to a F(U) cell). In Fig. 8(c), the robot initiates the second exploration phase before completing the cleaning of cells from the first exploration path. Fig. 8(d) shows the spanning tree and cleaning path generated at the end of the fourth exploration phase. Blue shaded cells represent occupied F(E) cells, and spanning tree is depicted using thin lines. A new spanning tree is generated, merged with the initial tree, and the cleaning phase continues. This process of spanning tree generation and cleaning continues until coverage is complete, as shown in Fig. 9(b). The algorithm terminates when the next cleaning path leads to the starting small cell. As depicted in Fig. 9(b), the proposed algorithm consistently provides a complete and non-overlapping cleaning path, along with a basic map of the region as a by-product.

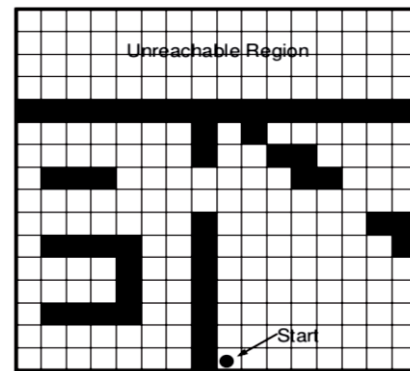


Fig. 7. The entire workspace divided into 16 large cells. The presence of obstacles (depicted in black) in the workspace leads to topological disconnectivity, resulting in permanently unreachable areas for the robot.

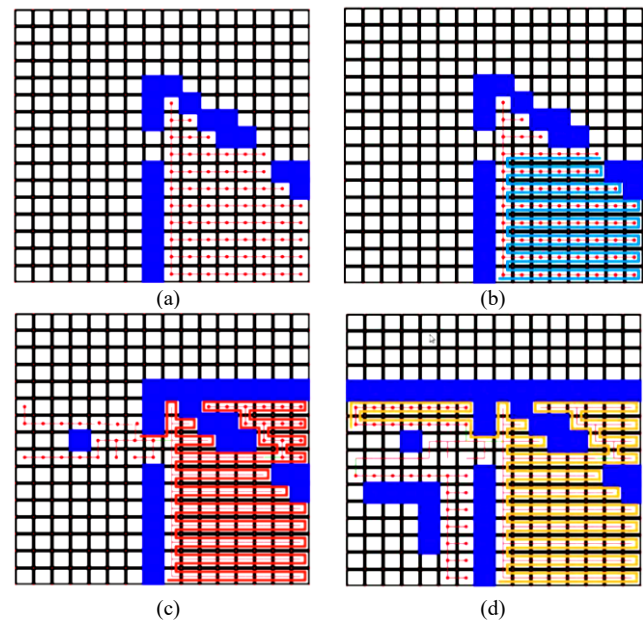


Fig. 8. The cleaning procedure executed by the robot after each exploration phase: a) Spanning tree generated at the end of the initial exploration phase. b) Second exploration initiated at the first exploration window. c) Spanning tree generated at the end of the third exploration phase. d) Spanning tree and cleaning path generated at the end of the fourth exploration phase. Blue shaded cells represent occupied F(E) cells, and spanning tree is depicted using thin lines.

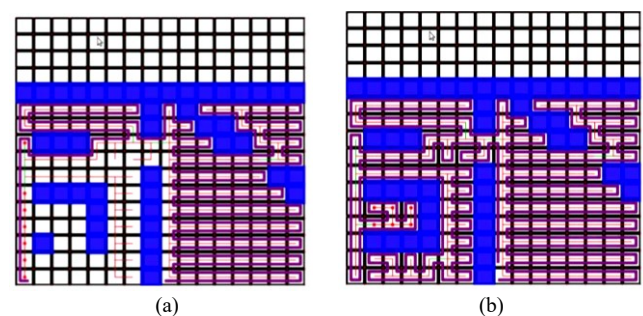


Fig. 9. Continuation of the cleaning procedure executed by the robot after each exploration phase from: a) Snapshot at the end of the fifth exploration phase. b) Final scenario showcasing the spanning tree using thin lines.

The implementation of the proposed algorithm in the V-Rep simulation environment was successfully carried out. The progression of this implementation is depicted in Fig. 10, Fig. 11, and Fig. 12. Fig. 10 presents the initial scenario for implementation, including the exploration sensor range. As

previously mentioned, the cleaning process is executed with intermittent exploration phases. Fig. 12 showcases the final result of the cleaned workspace, demonstrating the successful completion of the cleaning process.

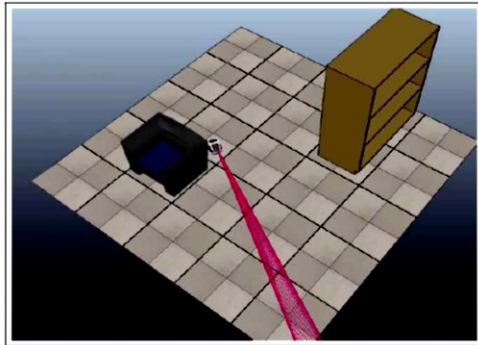


Fig. 10. The initial scenario generated in V-Rep simulator along with exploration sensor range.

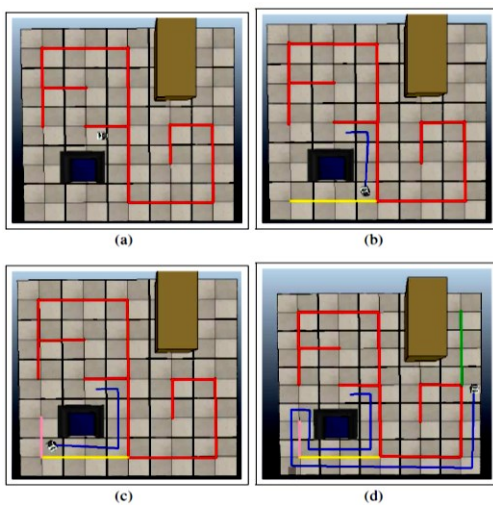


Fig. 11. Various stages of the cleaning operation in the V-Rep simulator, highlighting the intermittent exploration phases: (a) Spanning tree generated at the end of the initial exploration phase. (b) Spanning tree at the end of the second exploration phase (yellow), appended to the initial tree. (c) Spanning tree at the end of the third exploration (shown in light pink colour). (d) Final tree at the conclusion of the exploration phases (green).

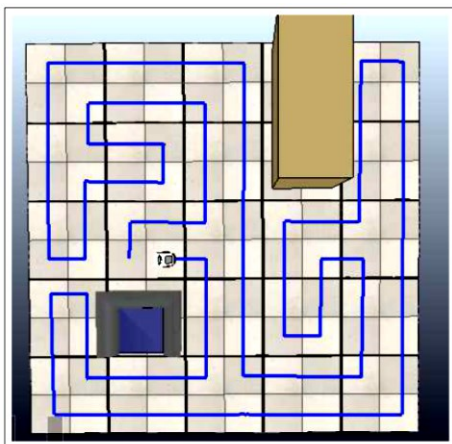


Fig. 12. The final cleaned arena. The blue thick lines show the cleaning path followed by the robot. To improve the clarity, the spanning tree is omitted.

To achieve a more realistic performance, the proposed algorithm was implemented in the ROS-Gazebo environment

using the standard Turtlebot robot with LIDAR sensors for exploration. Fig. 13 illustrates the scenario considered. In the initial exploration process depicted in Fig. 13, the blue sections represent the explored area, and the white sections indicate inaccessible regions due to the presence of obstacles from the robot's current position. Using this knowledge, an occupancy map of the large cells in the known region is generated. Fig. 14 illustrates the cleaning path at various instances during the exploration phases. Three exploration phases were needed to completely explore the given workspace. It's important to note that due to the inherent property of the underlying spanning tree-based coverage used in this paper, partially occupied cells may be left uncovered. However, this can be addressed by selecting an appropriate coverage algorithm, as demonstrated in related literature [11], [15], [16], [17]. The ROS-Gazebo environment provides a realistic simulation platform, and the same program can be implemented on a real Turtlebot robot, yielding comparable results. During real-time implementation, considerations such as using LIDAR or other vision-based sensors and modifying the exploration section of the program to accommodate real-time data from onboard sensors are essential.

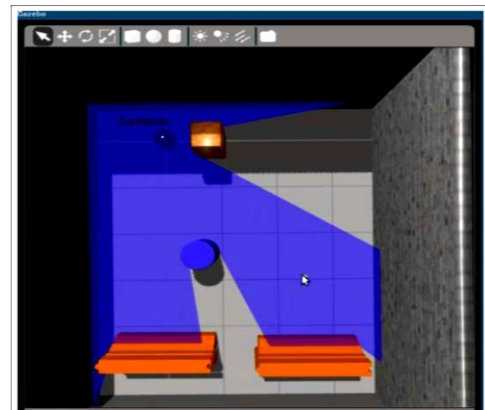


Fig. 13. ROS-Gazebo simulation scenario. The robot is positioned at the top left corner. Explored areas are highlighted in blue, while inaccessible regions due to obstacles from the robot's current position are depicted in white.

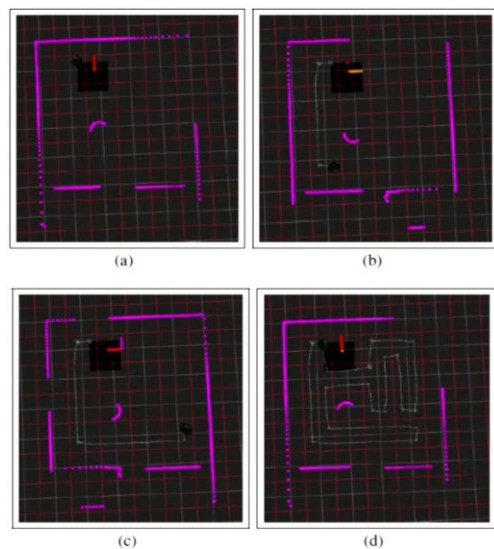


Fig. 14. Various instances of cleaning process. The entire workspace is covered in three exploration phases. The major cells are shown in red and minor ones in white.

In conclusion, the simulation experiments presented in this section served as a crucial step in assessing the effectiveness of the proposed algorithm for robotic cleaning applications. The multi-phased simulations, conducted in V-rep/MATLAB and ROS/Gazebo environments, provided valuable insights into the algorithm's performance, efficiency, and adaptability to real-world scenarios. The algorithm's strengths are notable. It successfully generates complete and non-overlapping cleaning paths, as demonstrated in the simulation results. Furthermore, it consistently produces a basic map of the environment as a by-product, enhancing its utility for various applications. The selective activation of sensors during exploration phases contributes to efficient battery usage, a critical aspect of long-term robotic operation. However, it's essential to acknowledge certain limitations. The inherent property of the underlying spanning tree-based coverage algorithm may result in partially occupied cells being left uncovered. While the paper suggests using alternative coverage algorithms to address this issue, future research might explore further refinements to ensure comprehensive coverage in all scenarios. The simulations conducted in this study provide a strong foundation for the proposed algorithm's practical implementation. The successful progression from simulations to the real-world application, using a Turtlebot in the ROS-Gazebo environment, suggests real-world feasibility. As the algorithm transitions to real-time implementation, considerations such as sensor selection and integration are crucial for seamless operation.

A brief comparison with other existing methodologies can be summarized as follows. When comparing the proposed cleaning algorithm with other similar algorithms, several notable performance characteristics come to the forefront. The most significant distinction is the integration of both exploration and coverage phases within the proposed algorithm. It seamlessly combines these two essential components, where exploration generates a map used by the coverage path planning algorithm. In contrast, several other approaches primarily focus on either exploration or coverage separately, and their explicit integration may not be as robust or evident in those cases [5], [6], [30], [33], [35]. Another standout feature of the proposed algorithm is its commitment to power efficiency. It prioritizes the conservation of battery power by deactivating exploration sensors during the cleaning phase, a strategy not explicitly addressed in many other algorithms, which may lead to higher power consumption [13], [69]. Adaptability to dynamic and unknown environments is a significant advantage of the proposed algorithm. It can function effectively without prior knowledge of the workspace, making it versatile and adaptable to various scenarios and changing environments. In contrast, several other approaches often rely on prior knowledge of the workspace, limiting their adaptability to dynamic settings [40], [13], [58], [66], [71], [77]. Complete and non-repetitive coverage is a key focus of the proposed algorithm, ensuring that all areas are covered without unnecessary revisits to already explored regions. While some other approaches aim for complete coverage, not all explicitly address the issue of non-repetitive coverage, potentially leading to inefficiencies in the cleaning path [16], [17], [45], [61], [64], [67]. Additionally, the proposed algorithm

generates a detailed map of the workspace as a valuable by-product, providing additional value for tasks such as environment analysis, monitoring, or future planning. In contrast, many other approaches do not emphasize map generation as a significant outcome [71], [72], [73], [77]. Frontier cells play a crucial role in the proposed algorithm, acting as connectors between known and unknown cells, efficiently guiding the exploration process. While the use of frontier cells may vary across different approaches, not all explicitly leverage this concept as effectively [4], [9], [24], [50], [53], [55]. One more distinguishing feature of the proposed algorithm is its obstacle detection strategy. Unlike earlier mentioned approaches, it optimizes obstacle detection by activating sensors only during exploration phases, effectively reducing power consumption for obstacle detection. In contrast, most other approaches require continuous sensor activation for obstacle detection, leading to higher power usage per cell [5], [7], [22], [23], [24]. In summary, the proposed cleaning algorithm demonstrates a well-rounded set of performance characteristics, including integration of exploration and coverage, power efficiency, adaptability to dynamic environments, complete and non-repetitive coverage, map generation, efficient use of frontier cells, and an optimized obstacle detection strategy. These characteristics collectively position the proposed algorithm as a robust and versatile solution for coverage path planning in robotics.

IX. CONCLUSION

In this paper, we provided an in-depth exploration of a coverage algorithm designed for a mobile robot engaged in cleaning-like tasks. The algorithm seamlessly integrates exploration, mapping, and coverage, addressing critical aspects of efficient robotic cleaning. We introduced a frontier-based exploration strategy and proposed an offline Spanning Tree Coverage (STC) algorithm to tackle the coverage problem. As already mentioned, offline algorithms have prior information about the workspace. Graph-level simulations were instrumental in demonstrating the efficiency of our proposed algorithm in generating coverage paths free from gaps or redundant coverage. We further validated its effectiveness through simulations conducted in the VRep/MATLAB environment and on a Turtlebot within the ROS-Gazebo environment. The ROS-Gazebo implementation is versatile and can be seamlessly transferred to a physical Turtlebot equipped with appropriate sensors, mirroring the simulation conditions with minimal modifications. The simulation experiments presented in this section served as a crucial step in assessing the effectiveness of the proposed algorithm for robotic cleaning applications. The multi-phased simulations, conducted in V-rep/MATLAB and ROS/Gazebo environments, provided valuable insights into the algorithm's performance, efficiency, and adaptability to real-world scenarios. The algorithm's strengths are notable. It successfully generates complete and non-overlapping cleaning paths, as demonstrated in the simulation results. Furthermore, it consistently produces a basic map of the environment as a by-product, enhancing its utility for various applications. The selective activation of sensors during exploration phases contributes to efficient battery usage, a critical aspect of long-term robotic operation. However, it's

essential to acknowledge certain limitations. The inherent property of the underlying spanning tree-based coverage algorithm may result in partially occupied cells being left uncovered. While the paper suggests using alternative coverage algorithms to address this issue, future research might explore further refinements to ensure comprehensive coverage in all scenarios. As the algorithm transitions to real-time implementation, considerations such as sensor selection and integration are crucial for seamless operation. The adaptability of the algorithm to real-world robot platforms underscores its practical applicability for efficient coverage and cleaning tasks. While our paper does not extensively delve into detailed formal analysis or rigorous mathematical proofs, it's crucial to recognize that there are several potential avenues for future research that could enhance the depth and validation of our approach. One promising direction could involve conducting a formal analysis of the algorithm's coverage capabilities, potentially using mathematical proofs or formal methods to demonstrate its complete coverage guarantees under specific conditions. Additionally, future research could focus on establishing the optimality of the algorithm's path planning through mathematical analysis and comparisons with existing algorithms. Mathematical models could be developed to better understand the algorithm's resource allocation strategies, and formal methods could be employed to validate its adaptability in more complex and dynamic environments. Lastly, addressing sensor noise and uncertainty through mathematical modeling could further enhance the algorithm's robustness in real-world scenarios. These potential areas for formal analysis and mathematical research offer opportunities to bolster the algorithm's theoretical foundations, validating our claims and providing a more comprehensive understanding of its performance and limitations.

ACKNOWLEDGMENT

The authors would like to thank Manipal Institute of Technology, Manipal and Indian Institute of Technology, Kanpur for the facilities and support provided for this research. Also, Mr Minhaz Falaki and Mr Akshar Padman for their contributions in ROS simulations.

REFERENCES

- [1] K. L. Doty, R. R. Harrison, and I. Real, "Sweep strategies for a sensory-driven," in *Behavior-Based Vacuum Cleaning Agent, AAAI 1993 Fall Symposium Series, Instantiating Real-World Agents, Research Triangle Park, Raleigh, NC*, pp. 1-6, 1993.
- [2] M. Weiss-Cohen, I. Sirotnin, and E. Rave, "Lawn Mowing System for Known Areas," *2008 International Conference on Computational Intelligence for Modelling Control & Automation*, pp. 539-544, 2008, doi: 10.1109/CIMCA.2008.145.
- [3] E. U. Acar, H. Choset, Y. Zhang, and M. Schervish, "Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods," *The International Journal of Robotics Research*, vol. 22, no. 7-8, pp. 441-466, 2003.
- [4] V. J. Lumelsky, S. Mukhopadhyay, and K. Sun, "Dynamic path planning in sensor-based terrain acquisition," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 4, pp. 462-472, 1990.
- [5] D. Lee and M. Recce, "Quantitative evaluation of the exploration strategies of a mobile robot," *International Journal of Robotics Research*, vol. 16, no. 4 pp. 413-447, 1997.
- [6] B. Yamauchi, "Frontier-based exploration using multiple robots," *Proceedings of the Second International Conference on Autonomous Agents*, pp. 47-53, 1998.
- [7] B. Gonzalez and J. Latombe, "Planning robot motions for range-image acquisition and automatic 3d model construction," *Proceedings of the AAAI Fall Symposium*, 1998.
- [8] Albers, Kursawe, and Schuijver, "Exploring unknown environments with obstacles," *Algorithmica*, vol. 32, no. 1, pp. 123-143, 2002.
- [9] H. Choset, "Coverage for robotics—a survey of recent results," *Annals of mathematics and artificial intelligence*, vol. 31, no. 1-4, pp. 113-126, 2001.
- [10] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258-1276, 2013.
- [11] H. Choset, "Coverage of known spaces: The boustrophedon cellular decomposition," *Autonomous Robots*, vol. 9, no. 3, pp. 247-253, 2000.
- [12] Y. Gabriely and E. Rimon, "Spanning-tree based coverage of continuous areas by a mobile robot," *Annals of Math and Artificial Intelligence*, vol. 31, pp. 77-98, 2001.
- [13] Shnaps and E. Rimon, "Online coverage of planar environments by a battery powered autonomous mobile robot," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 425-436, 2016.
- [14] E. U. Acar and H. Choset, "Sensor-based coverage of unknown environments: Incremental construction of Morse decompositions," *The International Journal of Robotics Research*, vol. 21, no. 4, pp. 345-366, 2002.
- [15] Y. Gabriely and E. Rimon, "Competitive on-line coverage of grid environments by a mobile robot," *Computational Geometry*, vol. 24, no. 3, pp. 197-224, 2003.
- [16] T. D. Ranjitha and K. R. Guruprasad, "CCSTC: A truly complete competitive spanning tree coverage algorithm for mobile robots," *Proc. of Advances in Robotics, 2nd International Conference of Robotics Society of India*, 2015.
- [17] T. D. Ranjitha and K. R. Guruprasad, "Pseudo spanning tree-based complete and competitive robot coverage using virtual nodes," *IFAC Papers on Line: Proc of 4th IFAC Conference on Advances in Control and Optimization of Dynamical Systems (ACODS)*, 2016.
- [18] V. G. Nair and K. R. Guruprasad, "GM-VPC: An algorithm for multi-robot coverage of known spaces using generalized Voronoi partition," *Robotica*, vol. 38, no. 5, pp. 845-860, 2020.
- [19] V. G. Nair and K. R. Guruprasad, "Centroidal Voronoi Partitioning using Virtual Nodes for Multi Robot Coverage," *International Journal of Engineering and Technology*, vol. 7, no. 2.21, pp. 135-139, 2018.
- [20] V. G. Nair and K. R. Guruprasad, "Geodesic-VPC: Spatial Partitioning for Multi-Robot Coverage Problem," *International Journal of Robotics and Automation*, vol. 33, no. 3, pp. 189-198, 2020.
- [21] V. G. Nair and K. R. Guruprasad, "2D-VPC: An Efficient Coverage Algorithm for Multiple Autonomous Vehicles," *International Journal of Control, Automation, and Systems*, vol. 19, no. 8, pp. 2891-2901, 2021
- [22] V. G. Nair and K. R. Guruprasad, "Manhattan distance based Voronoi partitioning for efficient multi-robot coverage," in *Control Instrumentation Systems: Proceedings of CISCON 2018*, pp. 81-90, 2020.
- [23] V. G. Nair, A. Rag, K. P. Jayalakshmi, M. V. Dileep, and K. R. Guruprasad, "Cooperative Online Workspace Allocation in the Presence of Obstacles for Multi-robot Simultaneous Exploration and Coverage Path Planning Problem," *International Journal of Control, Automation and Systems*, vol. 21, pp- 2338-2349, 2023.
- [24] K. R. Jensen-Nau, T. Hermans, and K. K. Leang, "Near-Optimal Area-Coverage Path Planning of Energy-Constrained Aerial Robots with Application in Autonomous Environmental Monitoring," in *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 1453-1468, July 2021.
- [25] R. I. Mukhamediev *et al.*, "Coverage Path Planning Optimization of Heterogeneous UAVs Group for Precision Agriculture," in *IEEE Access*, vol. 11, pp. 5789-5803, 2023.
- [26] X. -X. Shao, Y. -J. Gong, Z. -H. Zhan, and J. Zhang, "Bipartite Cooperative Coevolution for Energy-Aware Coverage Path Planning of UAVs," in *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 1, pp. 29-42, Feb. 2022.
- [27] T. Yang, J. V. Miro, M. Nguyen, Y. Wang, and R. Xiong, "Template-Free Non-revisiting Uniform Coverage Path Planning on Curved

- Surfaces,” in *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 4, pp. 1853-1861, Aug. 2023.
- [28] M. Hassan and D. Liu, “PCPP: A Predator–Prey-Based Approach to Adaptive Coverage Path Planning,” in *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 284-301, Feb. 2020.
- [29] J. Lu, B. Zeng, J. Tang, T. L. Lam, and J. Wen, “TMSTC*: A Path Planning Algorithm for Minimizing Turns in Multi-Robot Coverage,” in *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 5275-5282, Aug. 2023.
- [30] J. Chen, C. Du, Y. Zhang, P. Han, and W. Wei, “A Clustering-Based Coverage Path Planning Method for Autonomous Heterogeneous UAVs,” in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25546-25556, Dec. 2022.
- [31] Y. Cao, X. Cheng, and J. Mu, “Concentrated Coverage Path Planning Algorithm of UAV Formation for Aerial Photography,” in *IEEE Sensors Journal*, vol. 22, no. 11, pp. 11098-11111, 2022.
- [32] M. Amersdorfer and T. Meurer, “Equidistant Tool Path and Cartesian Trajectory Planning for Robotic Machining of Curved Freeform Surfaces,” in *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 3311-3323, Oct. 2022.
- [33] B. Jafari, H. Saedi, S. Enayati, and H. Pishro-Nik, “Energy-Optimized Path Planning for Moving Aerial Base Stations: A Non User-Oriented Framework,” in *IEEE Communications Letters*, vol. 26, no. 3, pp. 672-676, March 2022.
- [34] J. Li, Y. Xiong, J. She, and M. Wu, “A Path Planning Method for Sweep Coverage With Multiple UAVs,” in *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8967-8978, Sept. 2020.
- [35] M. Zhang, W. Cai and L. Pang, “Predator-Prey Reward Based Q-Learning Coverage Path Planning for Mobile Robot,” in *IEEE Access*, vol. 11, pp. 29673-29683, 2023.
- [36] A. J. Sanchez-Fernandez, L. F. Romero, G. Bandera, and S. Tabik, “VPP: Visibility-Based Path Planning Heuristic for Monitoring Large Regions of Complex Terrain Using a UAV Onboard Camera,” in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 944-955, 2022.
- [37] Y. Ma *et al.*, “C C I B A*: An Improved BA* Based Collaborative Coverage Path Planning Method for Multiple Unmanned Surface Mapping Vehicles,” in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 19578-19588, Oct. 2022.
- [38] J. Xie and J. Chen, “Multiregional Coverage Path Planning for Multiple Energy Constrained UAVs,” in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 17366-17381, Oct. 2022.
- [39] Z. Wang *et al.*, “Full-Coverage Path Planning and Stable Interaction Control for Automated Robotic Breast Ultrasound Scanning,” in *IEEE Transactions on Industrial Electronics*, vol. 70, no. 7, pp. 7051-7061, July 2023.
- [40] P. Maini, B. M. Gonultas, and V. Isler, “Online Coverage Planning for an Autonomous Weed Mowing Robot With Curvature Constraints,” in *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5445-5452, April 2022.
- [41] S. Papaioannou, P. Kolios, T. Theocharides, C. G. Panayiotou, and M. M. Polycarpou, “Integrated Guidance and Gimbal Control for Coverage Planning With Visibility Constraints,” in *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 2, pp. 1276-1291, April 2023.
- [42] C. Zygowski and A. Jaekel, “Optimal path planning strategies for monitoring coverage holes in Wireless Sensor Networks,” *Ad Hoc Networks*, vol. 96, 2020.
- [43] Y. V. Pehlivanoglu and P. Pehlivanoglu, “An enhanced genetic algorithm for path planning of autonomous UAV in target coverage problems,” *Applied Soft Computing*, vol. 112, 2021.
- [44] E. Glorieux, P. Franciosa, and D. Ceglarek, “Coverage path planning with targetted viewpoint sampling for robotic free-form surface inspection,” *Robotics and Computer-Integrated Manufacturing*, vol. 61, 2020.
- [45] A. K. Lakshmanan *et al.*, “Complete coverage path planning using reinforcement learning for Tetromino based cleaning and maintenance robot,” *Automation in Construction*, vol. 112, 2020.
- [46] P. Yao, L. Qiu, J. Qi, and R. Yang, “AUV path planning for coverage search of static target in ocean environment,” *Ocean Engineering*, vol. 241, 2021.
- [47] L. Shi, S. Xu, H. Liu, and Z. Zhan, “QoS-Aware UAV Coverage path planning in 5G mmWave network,” *Computer Networks*, vol. 175, 2020.
- [48] S. W. Cho, H. J. Park, H. Lee, D. H. Shim, and S.-Y. Kim, “Coverage path planning for multiple unmanned aerial vehicles in maritime search and rescue operations,” *Computers & Industrial Engineering*, vol. 161, 2021.
- [49] K. Mayilvaganam, A. Shrivastava, and P. Rajagopal, “An optimal coverage path plan for an autonomous vehicle based on polygon decomposition and ant colony optimization,” *Ocean Engineering*, vol. 252, 2022.
- [50] H. Huang *et al.*, “Robotic organism targets regional coverage capture path planning for marine aquafarm based on value iteration network,” *Ocean Engineering*, vol. 281, 2023.
- [51] E. V. V. Carmona, J. I. V. Gomez, J. C. H. Lozada, and M. A. Cruz, “Coverage path planning for spraying drones,” *Computers & Industrial Engineering*, vol. 168, 2022.
- [52] R. S. Nilsson and K. Zhou, “Method and bench-marking framework for coverage path planning in arable farming,” *Biosystems Engineering*, vol. 198, 2020.
- [53] S. Faghihi, S. Tavana, and A. H. J. de Ruiter, “Kinodynamic on-orbit inspection path planning for full-coverage inspection in close proximity of space structures,” *Acta Astronautica*, vol. 198, 2022.
- [54] B. Ai, M. Jia, H. Xu, J. Xu, Z. Wen, B. Li, and D. Zhang, “Coverage path planning for maritime search and rescue using reinforcement learning,” *Ocean Engineering*, vol. 241, 2021.
- [55] G. Chen, Y. Shen, Y. Zhang, W. Zhang, D. Wang, and B. He, “2D multi-area coverage path planning using L-SHADE in simulated ocean survey,” *Applied Soft Computing*, vol. 112, 2021.
- [56] Y. Ma, Y. Zhao, Z. Li, X. Yan, H. Bi, and G. Królczyk, “A new coverage path planning algorithm for unmanned surface mapping vehicle based on A-star based searching,” *Applied Ocean Research*, vol. 123, 2022.
- [57] S. Yang, J. Huang, X. Xiang, J. Li, and Y. Liu, “Cooperative survey of seabed ROIs using multiple USVs with coverage path planning,” *Ocean Engineering*, vol. 268, 2023.
- [58] S. Dogru and L. Marques, “ECO-CPP: Energy constrained online coverage path planning,” *Robotics and Autonomous Systems*, vol. 157, 2022.
- [59] L. M. S. Bine, A. Boukerche, L. B. Ruiz, and A. A. F. Loureiro, “A Novel Ant Colony-inspired Coverage Path Planning for Internet of Drones,” *Computer Networks*, vol. 235, 2023.
- [60] F. Tang, “Coverage path planning of unmanned surface vehicle based on improved biological inspired neural network,” *Ocean Engineering*, vol. 278, 2023.
- [61] S. Zhao and S. H. Hwang, “Complete coverage path planning scheme for autonomous navigation ROS-based robots,” *ICT Express*, 2023.
- [62] L. Jiao, Z. Peng, L. Xi, S. Ding, and J. Cui, “Multi-Agent Coverage Path Planning via Proximity Interaction and Cooperation,” in *IEEE Sensors Journal*, vol. 22, no. 6, pp. 6196-6207, 2022.
- [63] V. Bulut, “Path planning algorithm for mobile robots based on clustering-obstacles and quintic trigonometric Bézier curve,” *Annals of Mathematics and Artificial Intelligence*, pp. 1-22, 2023.
- [64] C. Li, C. Liu, Y. Song, and Z. Liang, “Parameter value selection strategy for complete coverage path planning based on the Lü system to perform specific types of missions,” *Frontiers of Information Technology & Electronic Engineering*, vol. 24, no. 2, pp. 231-244, 2023.
- [65] S. Shentu, Z. Gong, X. J. Liu, Q. Liu, and F. Xie, “Hybrid Navigation System Based Autonomous Positioning and Path Planning for Mobile Robots,” *Chinese Journal of Mechanical Engineering*, vol. 35, no. 1, pp. 1-13, 2022.
- [66] G. Kyprianou, L. Doitsidis, and S. A. Chatzichristofis, “Towards the Achievement of Path Planning with Multi-robot Systems in Dynamic Environments,” *J. Intell. Robot. Syst.*, vol. 104, no. 15, 2022.
- [67] A. Singha, A. K. Ray, and A. B. Samaddar, “Neural dynamics based complete grid coverage by single and multiple mobile robots,” *SN Applied Sciences*, vol. 3, no. 5, p. 543, 2021.
- [68] P. Tirandazi, A. Rahiminasab, and M. J. Ebadi, “An efficient coverage and connectivity algorithm based on mobile robots for wireless sensor

- networks,” *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-23, 2022.
- [69] B. Xu *et al.*, “High-efficiency inspecting method for mobile robots based on task planning for heat transfer tubes in a steam generator,” *Frontiers of Mechanical Engineering*, vol. 18, no. 2, pp. 1-15, 2023.
- [70] D. An, Y. Mu, Y. Wang, B. Li, and Y. Wei, “Intelligent Path Planning Technologies of Underwater Vehicles: a Review,” *Journal of Intelligent & Robotic Systems*, vol. 107, no. 2, p. 22, 2023.
- [71] K. Sridharan, P. McNamee, Z. N. Ahmadabadi, and J. Hudack, “Online search of unknown terrains using a dynamical system-based path planning approach,” *Journal of Intelligent & Robotic Systems*, vol. 106, no. 1, p. 21, 2022.
- [72] S. Ganesan, S. K. Natarajan, and J. Srinivasan, “A global path planning algorithm for mobile robot in cluttered environments with an improved initial cost solution and convergence rate,” *Arabian Journal for Science and Engineering*, vol. 47, no. 3, pp. 3633-3647, 2022.
- [73] J. Mao, X. Hu, L. Zhang, X. He, and M. Milford, “A bio-inspired goal-directed visual navigation model for aerial mobile robots,” *Journal of Intelligent & Robotic Systems*, vol. 100, pp. 289-310, 2020.
- [74] J. K. Verma and V. Ranga, “Multi-robot coordination analysis, taxonomy, challenges and future scope,” *Journal of intelligent & robotic systems*, vol. 102, pp. 1-36, 2021.
- [75] X. Zhang, Y. Huang, S. Wang, W. Meng, G. Li, and Y. Xie, “Motion planning and tracking control of a four-wheel independently driven steered mobile robot with multiple maneuvering modes,” *Frontiers of Mechanical Engineering*, vol. 16, pp. 504-527, 2021.
- [76] L. Wang, Z. Wang, M. Liu, Z. Ying, N. Xu, and Q. Meng, “Full coverage path planning methods of harvesting robot with multi-objective constraints,” *Journal of Intelligent & Robotic Systems*, vol. 106, no. 1, p. 17, 2022.
- [77] L. Chang, L. Shan, C. Jiang, and Y. Dai, “Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment,” *Autonomous Robots*, vol. 45, pp. 51-76, 2021.
- [78] W. Pawgasame and K. Wipusitwarakun, “Mobile sensor patrol path planning in partially observable border regions,” *Applied Intelligence*, pp. 1-21, 2021.
- [79] K. C. Huang, F. L. Lian, C. T. Chen, C. H. Wu, and C. C. Chen, “A novel solution with rapid Voronoi-based coverage path planning in irregular environment for robotic mowing systems,” *International Journal of Intelligent Robotics and Applications*, vol. 5, no. 4, pp. 558-575, 2021.
- [80] A. Hentout, A. Maoudj, and M. Aouache, “A review of the literature on fuzzy-logic approaches for collision-free path planning of manipulator robots,” *Artificial Intelligence Review*, vol. 56, no. 4, pp. 3369-3444, 2023.
- [81] J. I. Vasquez-Gomez, M. Marciano-Melchor, L. Valentin, and J. C. Herrera-Lozada, “Coverage path planning for 2d convex regions,” *Journal of Intelligent & Robotic Systems*, vol. 97, pp. 81-94, 2020.
- [82] H. Taheri and Z. C. Xia, “SLAM; definition and evolution,” *Engineering Applications of Artificial Intelligence*, vol. 97, p. 104032, 2021.