IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier

Reinforcement learning-based Energy-Aware Area Coverage for Reconfigurable hRombo Tiling Robot

ANH VU LE[1,2], RIZUWANA PARWEEN[1], PHONE THIHA KYAW[1,3], RAJESH ELARA MOHAN[1], TRAN HOANG QUANG MINH[2], CHARAN SATYA CHANDRA SAIRAM BORUSU[1]

[1]ROAR lab, Engineering Product Development Pillar, Singapore University of Technology and Design, Singapore 487372, Singapore
[2]Optoelectronics Research Group, Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City 700000, Vietnam
[3]Department of Mechatronic Engineering, Yangon Technological University, Insein, Myanmar

Corresponding author: Tran Hoang Quang Minh (tranhoangquangminh@tdtu.edu.vn)

This research is supported by the National Robotics Programme under its Robotics Enabling Capabilities and Technologies (Funding Agency Project No. 192 25 00051), National Robotics Programme under its Robot Domain Specific (Funding Agency Project No. 192 22 00058) and administered by the Agency for Science, Technology and Research.

ABSTRACT Applying the automation in covering the areas entirely eases manual jobs in various domestic fields such as site investigation, search, rescue, security, cleaning, and maintenance. A self-reconfigurable robot with adjustable dimensions is a viable answer to improve the coverage percentage for predefined map areas. However, the shape-shifting of this robot class also adds to the complexity of locomotion components and the need for an optimal complete coverage strategy for this new type of robot. The typical complete coverage route, including the least times of shape-shifting, the shortest navigation route, and the minimum travel time, is presented in the article. By splitting the map into the sub-areas similar to the self-reconfigurable robot's available shapes, the robot can design the ideal tileset and optimal navigation strategies to cover the workspace. To this end, we propose a Complete Tileset Energy-Aware Coverage Path Planning (CTPP) framework for a tiling self-reconfigurable robot named hRombo with four rhombus-shaped modules. The robot can reconfigure its base structure into seven distinct forms by activating the servo motors to drive the three robot hinges connecting robot modules. The problem of optimal path planning assisting the proposed hRombo robot to clear optimally all predefined tiles within the arbitrary workspace is considered a classic Travel Salesman Problem (TSP), and this TSP is solved by the reinforcement learning (RL) approach. The RL's reward function and action space are based on robot kinematic and the required energies, including transformation, translation, and orientation actions, to move the robot inside the workspace. The CTPP for the hRombo robot is validated with conventional complete coverage methods in simulation and real workspace conditions. The results showed that the CTPP is suitable for producing Pareto plans that enable robots to navigate from source to target in different workspaces with the least consumed energy and time among considered methods.

INDEX TERMS Reconfigurable robot; Tiling robotic; Reinforcement learning, Complete coverage planing; Energy path planning

I. INTRODUCTION

Autonomous systems have been developing for both home and industrial appliances as their consumer demand witnesses a huge increase during recent years. The routine cleaning and maintenance duties consume significant time and effort by manual operators. Tiling technology plays significant role in automation approach in various areas, including cleaning [1], maintenance [2], construction [3], [4] inspection in both indoor and outdoor spaces [5], [6]. The tiling robots are available in different forms in the market, such as oval, square, symmetrical shapes, and asymmetrical shapes, but their fixed morphological form constrains each of them practically. Reconfigured tiling robots [7], [8], [9] can cover more segments of any workspace that contrasts with a fixed morphological robot. This is due to their ability to change their form, which can be achieved staggeringly

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3038905, IEEE Access

by tiling robots. In general, the ability to transform into different shapes allows them to select forms that suit their current inclusion needs. One such robot to clean up the predefined workspace is the polyform based reconfigurable robots, including hTetro [9] hTrihex [10], hTetrakix [11]. Robots can transform into seven diverse tetromino shapes using four squares with differential drive locomotion mechanism. This gives the robot stage the ability to move in tricky situations and around obstacles.

However, the need for thoughtful researches such as path planning for a reconfigurable robot is rapidly arousing. Typically, conventional path planning focus on finding the feasible solution in consideration of the shortest distance to navigate the fixed form robot from source to destination. Besides, the complete coverage path planning methods are also mostly proposed for conventional fixed robots and extend the idea of traditional path planning. The works of [12], [13] use sensor fusion and peception network to enhance the complete are coverage task in the sense of human robot interaction. In the tiling robot cases, the complete coverage while avoiding the obstacle needs to be considered. The fundamental problem of reconfigured tiling robots is to create the optimal set from the available shapes and navigation strategies to cover the entire area. This includes arranging adequate and fully encompassing territory coverage while maintaining a strategic distance from any existing obstacles.

Specifically, the hRombo self-reconfigurable tiling robot proposed in this paper with the shape-shifting to seven shapes provides the possible idea to tile the pre-setup rombo based area. Because of the complexity of shape-shifting robots, smooth locomotion among available configurations is challenging, and complete coverage with multiple configurations is even more challenging and interesting. Sine the reconfigurable robot has a number of degree of freedoms and the additional constraints due to the base footprint size, the conventional complete coverage approaches no longer appropriate to derive the idea solutions. Therefore, robust or revised complete coverage approaches need to be implemented for the proposed reconfigurable robots considering the possible morphologies and the available locomotion.

Conventional complete coverage path planning techniques can be comprehensively aggregated depending on the decomposition techniques used to simplify the workspace [14]. A decomposition technique involves splitting the predefined map into smaller partitions, likewise referring to submap or cellular. The exemplary technique consists of isolating space with basic fixed shapes such as grid=based motiong planning [15] and infinite morphologies [16]. Other techniques can slip the map equally based on each sub-region complexity, such as the isolated method used in Morse [17] work. A number of different methods combine the use of graph theory[18], and high-order observers-based LQ control scheme [19]. The other common and popular methods are the standard grid-based probability assignment proposed by Moravec and Elfes [20] and Choset [21]. This method gives each considered cell the probability scheme to indicate how the obstacle occupies this cell. The higher the probability value, the higher change the exiting of obstacles in the considered cell. Several calculations can be used to segment a situation using matrix technology, combining vitality mindfulness calculations, neural network-based system [22], across trees [23], and energy based optimization [24]. The use of lattice-based attenuation drastically reduces the multifaceted nature required until the computation is decided. However, these map simplifying methods are applied for fixed morphological robots.

The usual technique for tiling robot-based complete coverage inside the grid-based map consists of two phases. Initially, a tile showing the shapes needed to occupy space was created using the polyomino [25] hypothesis with some lemmas. After that, the tiling robot will move to each defined tile location within the selected cell and change its morphology to an appropriate form. This method can ensure that the workspaces can be paved entirely with a tiling robot. A preeminent method demonstrates how to sort this problem using the cells produced as Travelsaleman Problem (TSP). This derives the lowest cost (generally proficient) under the guise of all reference points to ensure the greatest inclusion. Resolving this TSP is an impractical NP-hard problem in a specified time. The conventional method can apply the evolutionary-based optimization such as genetic algorithm [26] and ant colony optimization [27] to derive the solution for this TSP in a reasonable time. As such, these methods depend on the tiling hypothesis, which is firmly bound by destinations and cannot be adjusted to any self-assertion condition. Besides, improving the evolutionary-based TSP arrangement requires many computational cycles to distinguish an ideal solution and can be adversely affected by local minimums during optimization.

RL has been applied in various fields to get the optimal solution automatically. Changxi $et\ al.$ [28] has proposed using learning aids to guide self-sufficiency facilities. Kenzo $et\ al.$ [29] used DDPG-assisted learning calculations to design the motion of bipedal robots in football coordinates. Farad $et\ al.$ [30] has created a way to master proficiency in difficult conditions through the Enterunder Pundit Fortress learning model. A model prepared using Q-Find a way to produce the route from point A to point B in a grid-based partitioned map has been proposed in Aleksandr $et\ al.$ [31], Amit $et\ al.$ [32] and Soong $et\ al.$ [33]. David $et.al.$ extends this method to multiple robot agents [34]. Yuan $et\ al.$ [35] used the RNN GRU system to directly design an optimal path from source to the goal while avoiding obstacles in frame-based conditions. Lakshmanan $et\ al.$ [36] discussed using Q-Learning to arrange modern tilling robotics. In all cases, these works focus on the overage-oriented demarcation of guidelines from the source to destination and do not directly propose a complete coverage situation of reconfigurable tiling robots.

This paper proposes a CTPP deep learning model using an RL technique for the hRombo, which can determine

the optimal energy-aware navigation solution. The proposed complete path planning framework consists of the three phases for considered reconfigurable tiling hRomborobot. Basing on the kinematic design of the proposed robot platform, RL's reward function evaluates the cost of navigation based on individual transformation, translation, and orientation actions of the robot. The outcome of proposed trained RL models creates effective navigation strategies by limiting the number of form changes while amplifying solitary shapes in the considered workspace. This model is also flexible with challenging conditions of obstacle settings. There are threefold as the contributions of this article: (1) We proposed a complete tileset coverage CTPP approach developed for rhombus shape-based reconfigurable tiling robots.(2) We build the RL reward function based on the Travel Salesman Problem, which depends on the platform's real actions within any defined workspace.(3) CTPP is proposed to be tested on a real robot platform and proves energy and travel time effectiveness.

The article is composed as follows. HRombo's design is presented in Part 2. The proposed robot description on the cross-sectional workspace is divided into each item in part 3. In part 4, the CTPP technique is proposed with the hRombo robot representation, according to the tiling theory. The proposed system's optimal CTPP is approved in Section 5. The final section, along with potential future work, is investigated in the Final Section VII

II. THE HROMBO ROBOT DESCRIPTION

The hRombo platform has four rhombus-shaped blocks linked by three hinges, as shown in Fig. 1. The hinge is a planar revolute joint. Upon rotation of the blocks about the hinges, the platform is capable of forming seven forms, as shown in Fig. 2. The sidewalls of the platform are modular and fabricated with 3D printing using PLA material. The base of each block is an acrylic sheet which is manufactured using a laser cutting machine. Each hinge is an active joint, driven by a Herkulex servo motor. The platform follows the four-wheel independent steering drive principle for locomotion. Each block has a separate locomotion unit, as in Fig. 1. The servo motor can change each wheel's heading angle within 0 to 2π rad around the center shaft. Figure 3 describes the electronic block diagram of the hRombo platform. Each locomotion unit of the platform consists of a standard steerable wheel connected to a geared DC motor with a gear ratio of 250:1, voltage rating of 7.4V, operation torque of 1.37 Nm, and an operation speed 60 rpm and an attached Herkulex servo motor also steers each wheel. A 14.4v Lion battery with proper regulators is the main power unit. The platform weighs approximately 2.5 Kg.

For mapping and indoor localization, an Ultra Wide Band UWB infrastructure is used. This system provides real-time two-dimensional data (x,y) about the global position during the platform's navigation. Each wheel is connected to a wheel encoder that provides the platform's position in the

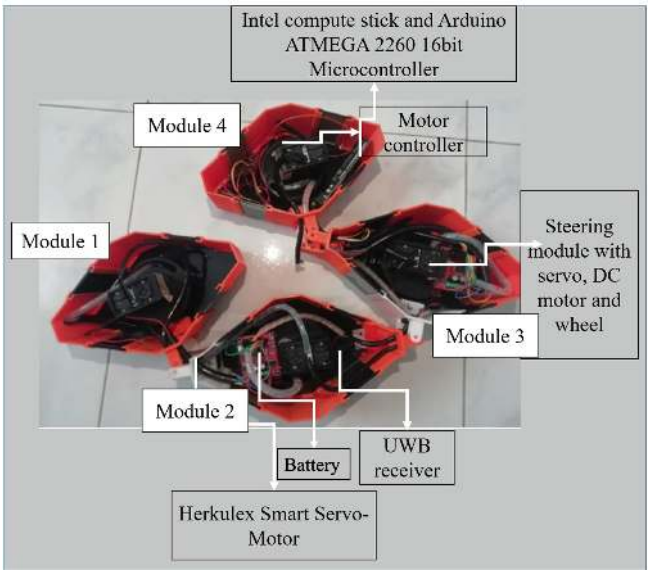

Figure 1: hRombo platform showing the electronic components.

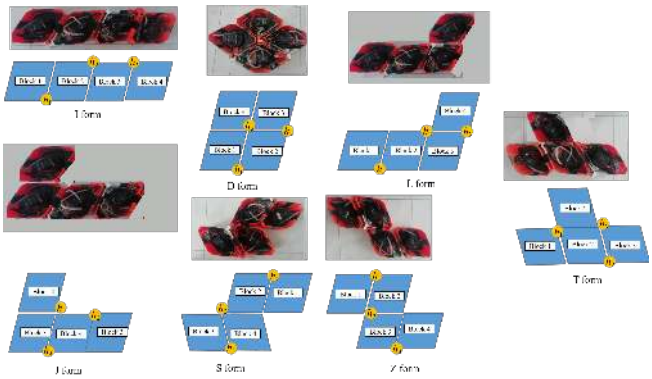

Figure 2: Different forms of the hRombo platform.

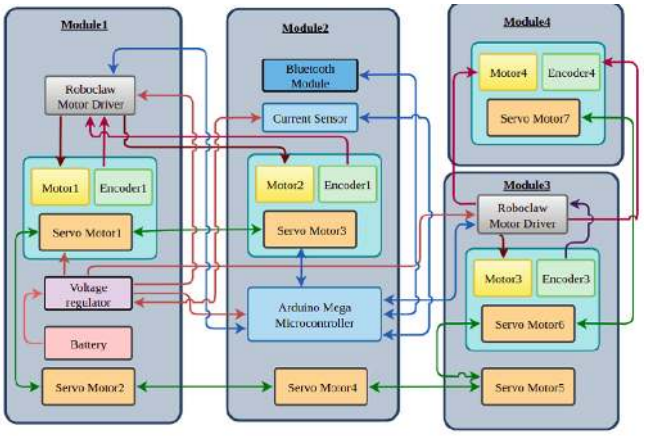

Figure 3: Electronic layout of hRombo.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3038905, IEEE Access

robot frame (local location) after starting navigation. An inertial measuring unit (IMU) is used to monitor the heading angle. The x, y positions from UWB, wheel encoders, and heading data from the IMU are fused by extended Kalman Filter to overcome the noise then get a reliable robot location. The robot operating system (ROS) is the main communication infrastructure of the proposed platform. The main processing unit is ComputeStick V5 from Intel, generating the trajectory and transmitting commands of desired form, desired travel distance, and heading to the wheel and the servomotors. The motion controller monitors the steering angle of the wheel and hinged joint during navigation, depending upon the heading angle and the form. The controller synchronizes all the four motors during locomotion.

III. REPRESENTATION OF HROMBO PLATFORM IN THE RHOMBUS-BASED WORKING ENVIRONMENT

The pre-built working environment is divided into regulated rhombus-based-grids the same as robot block shape. The robot inside this workspace is defined as four-dimensional waypoint $W(x, y, T, \varphi_h)$ consisting of the center of gravity (COG) of each robot x, y, tile name T among robot shapes, and heading of robot φ_h. Modules and COGs for hRombo forms are shown in Figure 4. Considering this figure, the operation of deriving the robot from the I form to the D form then to the L form around the active hinges ID edge of $h_1, h_2, h3$ is accomplished by the required angle rotations of robot blocks. The hRombo location of block b denoted as $\{x_b^w, y_b^w, \varphi_b^w\}$, where b is within four modules of hRombo ($b \in \{B_1, B_2, B_3, B_4\}$) can be derived from robot location and shape inside the workspace. The mass of each module is assigned among m_1, m_2, m_3, m_4. Given robot form within seven available from, the four-block location base on the robot heading within the workspace is shown in Figure 5.

Basing on these descriptions, the corresponding robot actions such as transformation, translation, and orientation can be modelled mathematically to move the robot's shape between any points within the workplace. Specifically, the robot's route direction to visit all reference points is partitioned into different sets of two reference points. For routing all the n waypoint, the route's pair is characterized as $p(W_k^s, W_k^g)$, where k represents the pair order and s is the source reference point and g is a destination point of pair number k. The reference point will have $k = 1$ and the last waypoint will have $k = n - 1$. For a trajacory including n points, the set of $n-1$ linking pairs of two points is formed, and the possible pairs is $\Omega = n(n-1))/2$.

IV. ENERGY-AWARE COMPLETE AREA COVERAGE FRAMEWORK FOR HROMBO ROBOT

A. RHOMBUS TILE-BASED COMPLETE AREA COVERAGE PLANNING

The hRombo platform follows tiling based path planning during floor cleaning. Due to the complex and irregular shape of the platform, as shown in Figure 6, we propose isohedral based tiling theory. In isohedral tiling, a single form of hRombo is fitted to itself repeatedly in a number of same or different orientations. This tiling method is of two types, i.e., (a) Firstly, the tiled workspace consists of only one form in the same orientation connected with translation symmetry. (b) secondly, the tiled workspace consists of a number of different rotated or reflected forms connected together. The hRombo platform tiles the pre-described environment with any of its seven forms. The rhombus-based tilesets with the robot forms are sampled as Figure 6. The isohedral tiling concept of the hRombo platform is described in qualitative forms, as described below.

A tetra rhombo of 'I'and 'D' forms can be arranged without any rotation and tiled to form a closed (no internal void)) and regular workspace with smooth boundary,as shown in Figure 6(a) and (b).

A tetra rhombo of 'Z' and 'S' forms can be arranged without any rotation and tiled to form a closed (no internal void) and regular workspace with a rough boundary, as shown in Figure 6(c) and (d).

A tetra rhombo of 'L' form with in-pivot rotations can be tiled to form various workspace, as shown in Figure 6(e)-(h).

- When the 'L' form is combined with another 'L' form with 180-degree rotation, it can form both closed and open workspace with a smooth boundary, as shown in Figure 6(e)-(f).
- When the 'L' form is combined with another 'L' form with 180-degree rotation, it can form a closed workspace with a rough boundary, as shown Figure 6(g)-(h).

A tetra rhombo of 'J' form can be tiled to another 'J' form with 180-degree rotation to form a closed workspace with a rough boundary, as shown in Figure 6(i).

B. OPTIMAL COMPLETE RHOMBUS-BASED TILESET COVERAGE

The block diagram of the CTPP framework, as in Figure 7, combines three stages: workspace preparation, planning, and platform execution. To find the tileset after defining the shapes and workspace dimension, The rollback calculation [37] is applied. Specifically, an arbitrary shape is placed randomly inside the predefined workspace. If the rollback circles cannot arrange the following cells, the past cell's different possibilities will be tried. The process is circled until the robot shapes completely fill the predefined workspace. In order to complete the navigation inside the workspace, The hRombo platform tiles the workspace by loading the planned tilesets from consecutive predefined reference points, as described in Figure 8. Then hRombo performs three separate required actions, including changing the structure into an ideal target point called waypoint, doing linear moving directly from the COM of the reference source waypoint W^s to the COM of the reference destination point W^d, and doing the heading correction to compensate the heading offset between robot current heading and desired heading at the destination. For a detailed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3038905, IEEE Access

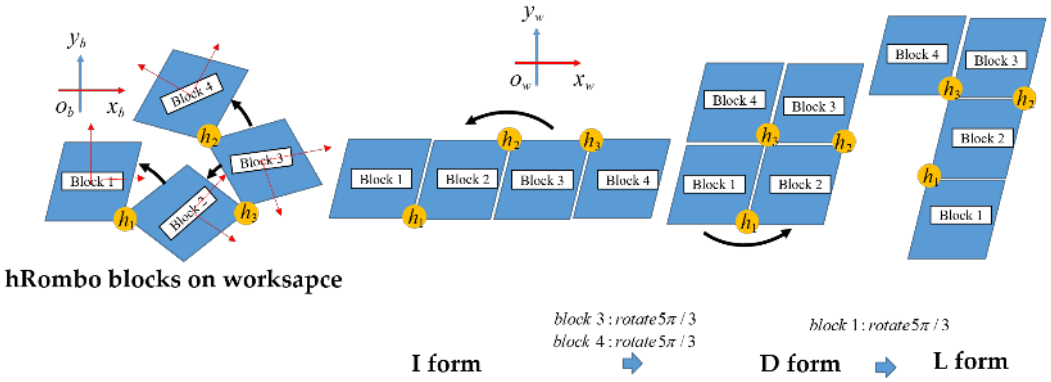

Figure 4: Representation of hRombo with shapeshifting within a workspace.

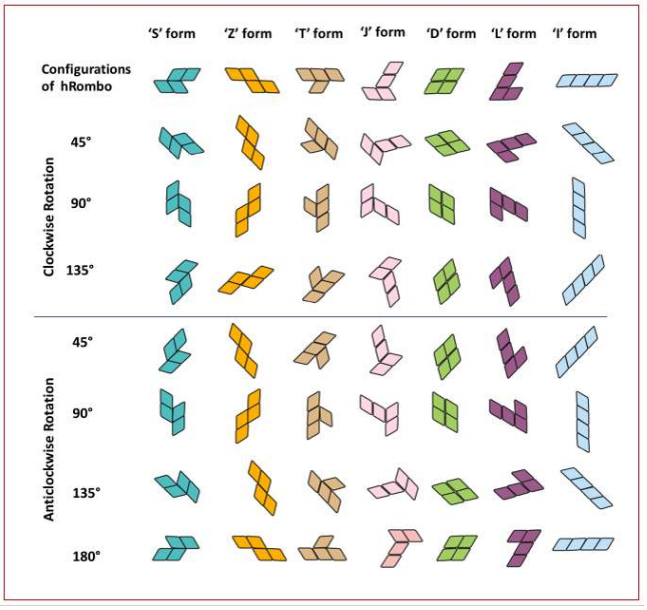

Figure 5: hRombo block location respecting to heading.

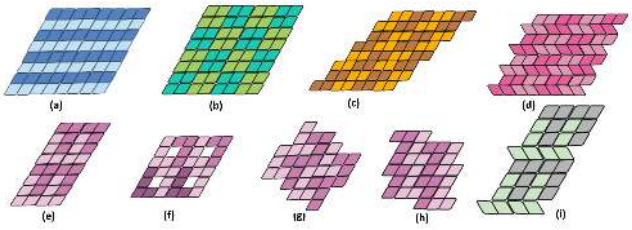

Figure 6: Rhombus based-tilsets to cover the workspace

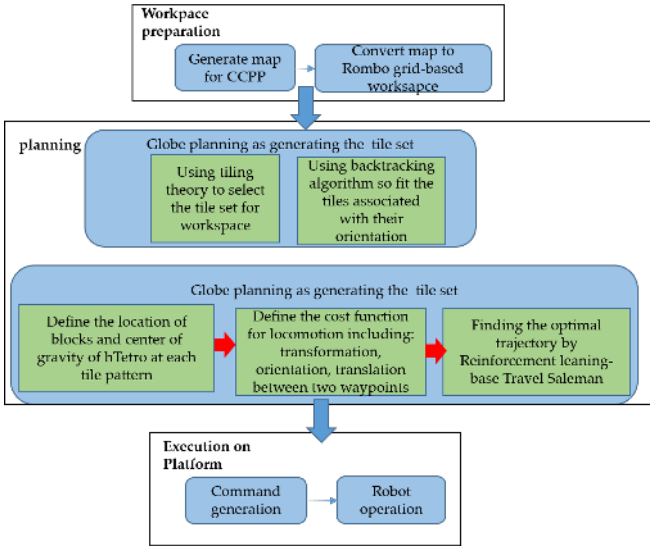

Figure 7: CTPP framework for hRombo robot.

definition of each action, Rotation θ_k of each robot module to change between the seven potential shapes is shown in Table 1. black The required tuning magnitudes in radial of each robot block between source and target shapes can be $l_m = \sum(l_1 + l_2)$ where l_1 equals to the length from hinge to block COM and l_2 equals to the length from hinge to the next block COM. These values are presented in Table 2. The required orientation correction of the robot title is defined by the offset between the robot heading at the goal waypoint φ_h^g and the source waypoint φ_h^s. Given the predefined map, as depicted in Figure 8, the robot stores the required orders into a robot database to perform three actions sequentially to fulfill the CTPP.

V. RL BASED CTPP

A. ENERGY BASED REWARD FUNCTION

The sequence of actions among shape-shifting, linear movement, and heading correction while clearing the waypoint is shown in Figure 8. The required energies to accomplish these actions are calculated by multiplying the actuators rotation distance, including servo motors at the hinges and DC motors at the locomotion units with the corresponding robot module mass. The energies for linear movement, shapeshifting, and heading correction are described in Equations (1), (2), and (3), respectively. The costweight as shown in Equation (4) is the summation of all component energies

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3038905, IEEE Access

Table 1: Rotation angle θ_k of robot blocks when shapeshifting.

W^s \ W^d	**D Shape** $B_1B_2B_3B_4$	**I Shape** $B_1B_2B_3B_4$	**L Shape** $B_1B_2B_3B_4$	**Z Shape** $B_1B_2B_3B_4$	**T Shape** $B_1B_2B_3B_4$	**J Shape** $B_1B_2B_3B_4$	**S Shape** $B_1B_2B_3B_4$
D Shape	0 0 0 0	π π 0 0	(π,$-\pi$) π 0 0	$-\pi$ 0 $-\pi$ ($-\pi$,$-\pi$)	$-\frac{2\pi}{3}$ 0 0 $-\pi$	($\frac{2\pi}{3}$,$-\pi$) $\frac{2\pi}{3}$ 0 π	($\frac{2\pi}{3}$,$-\pi$) $-\frac{2\pi}{3}$ 0 0
I Shape	$-\pi$ $-\pi$ 0 0	0 0 0 0	$-\pi$ 0 0 0	$-\pi$ 0 0 $-\pi$	$-\frac{2\pi}{3}$ 0 π (π,$-\pi$)	($-\frac{2\pi}{3}$,$-\pi$) $-\frac{2\pi}{3}$ 0 $-\pi$	($-\frac{2\pi}{3}$,$-\pi$) $-\frac{2\pi}{3}$ 0 0
L Shape	($-\pi$,π) $-\pi$ 0 0	π 0 0 0	0 0 0 0	0 0 0 $-\pi$	$-\frac{2\pi}{3}$ 0 $-\pi$ (π $-\pi$)	0 0 0 π	$-\frac{2\pi}{3}$ $-\frac{2\pi}{3}$ 0 0
Z Shape	π 0 π (π,π)	π 0 0 π	0 0 0 π	0 0 0 0	$\frac{2\pi}{3}$ 0 π π	$-\frac{2\pi}{3}$ $-\frac{2\pi}{3}$ 0 0	$-\frac{2\pi}{3}$ $-\frac{2\pi}{3}$ 0 π
T Shape	$\frac{2\pi}{3}$ 0 0 π	$\frac{2\pi}{3}$ 0 $-\pi$ ($-\pi$ π)	$\frac{2\pi}{3}$ 0 π π	$-\frac{2\pi}{3}$ 0 $-\pi$ $-\pi$	0 0 0 0	$\frac{2\pi}{3}$ 0 $\frac{2\pi}{3}$ $\frac{2\pi}{3}$	($\frac{2\pi}{3}$, $-\frac{2\pi}{3}$) $\frac{2\pi}{3}$ 0 π
J Shape	($-\frac{2\pi}{3}$,π) $-\frac{2\pi}{3}$ 0 π	($\frac{2\pi}{3}$,π) $\frac{2\pi}{3}$ 0 π	0 0 0 $-\pi$	$-\frac{2\pi}{3}$ $-\frac{2\pi}{3}$ 0 0	$-\frac{2\pi}{3}$ 0 $-\frac{2\pi}{3}$ $-\frac{2\pi}{3}$	0 0 0 0	0 0 0 π
S Shape	($-\frac{2\pi}{3}$,π) $\frac{2\pi}{3}$ 0 0	($\frac{2\pi}{3}$,π) $\frac{2\pi}{3}$ 0 0	$\frac{2\pi}{3}$ $\frac{2\pi}{3}$ 0 0	$-\frac{2\pi}{3}$ $-\frac{2\pi}{3}$ 0 $-\pi$	$-\frac{2\pi}{3}$ 0 $-\frac{2\pi}{3}$ ($-\pi$,$-\frac{2\pi}{3}$)	0 0 0 $-\pi$	0 0 0 0

Table 2: Tuning modul of robot blocks when shapeshfiing.

W^s \ W^d	**D Shape** $B_1B_2B_3B_4$	**I Shape** $B_1B_2B_3B_4$	**L Shape** $B_1B_2B_3B_4$	**Z Shape** $B_1B_2B_3B_4$	**T Shape** $B_1B_2B_3B_4$	**J Shape** $B_1B_2B_3B_4$	**S Shape** $B_1B_2B_3B_4$
O Shape	0 0 0 0	l_2 l_1 0 0	(l_1,l_2) l_1 0 0	l_1 0 l_1 (l_2,l_1)	l_1 0 0 l_1	(l_2,l_1) l_1 0 l_1	(l_2,l_1) l_1 0 0
I Shape	l_2 l_1 0 0	0 0 0 0	0 0 0 l_1	l_1 0 0 l_1	l_1 0 l_1 (l_2,l_1)	(l_2,l_1) l_1 0 l_1	(l_2,l_1) l_1 0 0
L Shape	(l_1,l_2) l_1 0 0	0 0 0 l_1	0 0 0 0	l_1 0 0 0	l_1 0 l_1 l_2	l_1 0 l_1 l_2	l_1 0 l_1 (l_2,l_1)
Z shape	l_1 0 l_1 (l_2,l_1)	l_1 0 0 l_1	0 0 0 l_1	0 0 0 0	l_1 0 l_1 l_2	l_1 l_1 0 0	l_1 l_1 0 l_1
T Shape	l_1 0 0 l_1	l_1 0 l_1 (l_1,l_2)	l_1 0 l_1 l_2	(l_1,l_1) 0 l_1 l_2	0 0 0 0	l_1 0 l_1 l_2	l_1 0 l_1 (l_2,l_1)
J Shape	(l_2,l_1) l_1 0 l_1	(l_2,l_1) l_1 0 l_1	l_1 0 l_1 l_2	l_1 l_1 0 0	l_1 0 l_1 l_2	0 0 0 0	0 0 0 l_1
S Shape	(l_2,l_1) l_1 0 0	(l_2,l_1) l_1 0 0	l_1 0 l_1 (l_2,l_1)	l_1 l_1 0 l_1	l_1 0 l_1 (l_2,l_1)	0 0 0 l_1	0 0 0 0

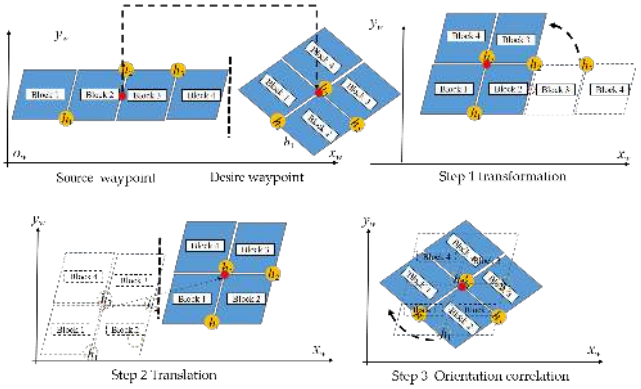

Figure 8: Three opeations of hRombo when navigate from source W_k^s to goal W_k^d

to carry the platform mass within the required distance between pair k of source waypoint $W_k^s(x, y, T, \varphi_h)$ and the goal waypoint $W_k^g(x, y, T, \varphi_h)$.

$$E_{tranl}(W_k^s, W_k^g) = \sum_{b=B1}^{B4} m_b \sqrt{(x_b^g - x_b^s)^2 + (y_b^g - y_b^s)^2} \tag{1}$$

$$E_{tranf}(W_k^s, W_k^g) = \sum_{b=B1}^{B4} m_b \theta_b l_m \tag{2}$$

$$E_{ori}(W_k^s, W_k^g) = \sum_{b=B1}^{B4} m_b |\varphi_h^g - \varphi_h^s| l_m \tag{3}$$

$$\mathbf{E}(W_k^s, W_k^g) = E_{tranl}(W_k^s, W_k^g) + E_{tranf}(W_k^s, W_k^g) + E_{ori}(W_k^s, W_k^g) \tag{4}$$

We derived the cost function based on the robot kinematic design and the operation within the rombo tileset generated by the tiling theory. Note that the cost function of the paper [38] used the 2D Euclidean Distance between two locations inside the workspace. Specifically, given an input tileset as the state space of the RL framework, we find the waypoints permutation, i.e., a trajectory π, that visits each waypoint once (except the starting waypoint) and has the minimum total energy. We propose the cost of a trajectory noted by a permutation π as:

$$L(\pi|S) = \mathbf{E}(W_n^s, W_1^g) + \sum_{k=1}^{n-1} \mathbf{E}(W_k^s, W_k^g), \tag{5}$$

where input state tileset consists of n waypoints $S = \{W_k\}_{k=1}^n$ and each W_k store locations and robot shape in the defined workspace. The energy cost function in Equation 5 is used as our total expected return $R(\pi|S) = L(\pi|S)$ (which we seek to minimize). In the case of TSP, we are dealing with an episodic task, where the termination of episode depends on the number of waypoints in the input state tileset. Discount rate has been set to one to make the

return objective takes the future rewards into account more strongly.

The algorithm depends on waypoint locations provided by the localization system of hRombo to yield an optimal navigation trajectory. The robot clears the workspace with the objective function of minimizing the overall trajectory energy-cost as the non-deterministic polynomial-time hardness problem of TSP. To handle the complexity of TSP with a large number of points, a non-deterministic approach has been proposed to derive the Pareto-optima solution. In this work, we solve the four blocks of rhombus-based tileset sequencing by using the neural networks with RL. A customized recurrent neural network that takes a set of robot locations as the predefined tileset waypoints is utilized to predict a distribution over various waypoint permutations. By defining energy-aware based reward function as Equation 5, the parameters of the recurrent neural network are optimized by the RL approach. The intelligent heuristics (or distribution over waypoint permutations) for the classic TSP can be achieved by training Neural Networks using RL with less engineering and no labeling efforts.

B. NEURAL NETWORK ARCHITECTURE FOR TSP

The RL network following the actor-critic architecture [39] to learn optimal heuristic TSP trajectories (or distribution over waypoint permutations) is shown in Figure 9.

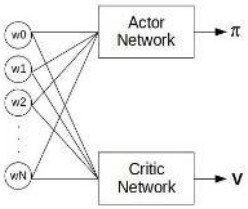

Figure 9: RL complete path planning for hRombo tiling robot

Following [38], our proposed neural network architecture also applies the chain rule technique to factorize the probability of trajectory π in Equation 5 as:

$$p(\pi|S) = \prod_{k=1}^{n} p(\pi(k) \mid \pi(< k), S), \tag{6}$$

Furthermore, each component on the right side of Equation (6) is processed consecutively by the softmax modules. Similar to [38], we use a method called pointer network [40] as our actor policy model, which consists of two recurrent neural network (RNN) modules, encoders, and decoders, each includes Long Short Term Memory (LSTM) cells [41]. The input states with the order of one waypoint at a time is examined by the encoder network. This network converts it into a series of latent memory states $\{enc_k\}_{k=1}^{n}$ where $enc_k \in \mathbb{R}^d$. The input to the encoder network at timestep k is a d-dimensional embedding of 4D waypoints W_k, obtained via a linear transformation of W_k, shared

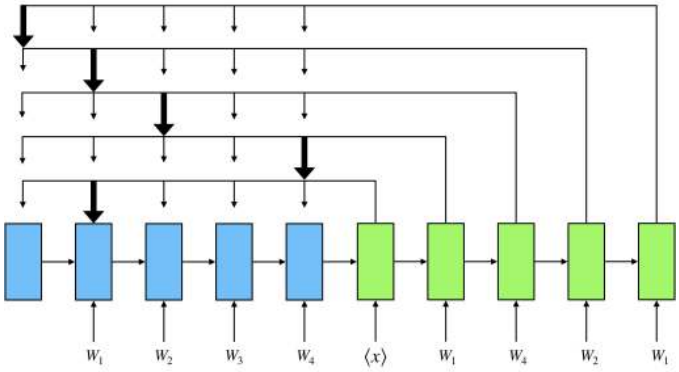

Figure 10: A pointer network architecture introduced by [40]

across all input steps. The decoder network is also in charge of maintaining its Latent memory states $\{dec_k\}_{k=1}^{n}$ where $dec_k \in \mathbb{R}^d$, and utilizes the pointing mechanism to generate a distribution over the upcoming waypoints (or chooses the discrete actions one step at a time) to yield the optimal trajectory length. Once the subsequent waypoint is determined, it is sent as an input to the next decoding step. For the choice of action space, since we are using the pointer network with a softmax output layer, the network predicts a probability distribution, utilizing the discrete set of actions, which points back to the input state sequence. The first decoding step input, (denoted by $\langle x \rangle$ in Figure 10) which is reproduced from [40] is a d-dimensional vector interpreted as a trainable parameter of our neural network.

C. OPTIMIZATION WITH RL

Solving NP-hard problems such as TSP and its variations by supervised learning is undesirable since the model accuracy depends on supervised labels of the dataset, and getting them is the burden works and infeasible. On the contrary, RL offers a proper and feasible paradigm for training neural networks, where an RL agent explores different trajectories and characterize the corresponding rewards. Hence, we propose using the Proximal Policy Optimization (PPO) algorithm [42], a new family of policy gradient methods for RL, to optimize our pointer network parameters. PPO performs comparably in small size TSP or better in larger size TSP than state of the art approaches like TRPO [43], DDPG [44] , while being much simpler to implement and tune. . The algorithm actively builds on Trust Region Policy Optimization (TRPO) and applies the critical concepts of TRPO like importance sampling, which improves the sample efficiency, as well as an alternative and simple method called Clipped Surrogate Objective function for stabilizing updates during the optimization step.

By utilizing the reward function described in Equation 5 as the training objective, i.e., given an tileset S, the expected trajectory length as Equation 7, we optimize the parameters θ of the policy pointer network.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3038905, IEEE Access

$$J(\theta|S) = \mathbb{E}_{\pi\sim p_\theta(.|S)} R(\pi|S) \tag{7}$$

Then we formulate the policy gradient of the objective by utilizing the PPO's clipped surrogate function as Equation 8, which controls stable updates during the optimization step.

$$\nabla_\theta J^{CLIP}(\theta|S) = \hat{\mathbb{E}}_{\pi\sim p_\theta(.|S)} \Big[min \Big(\hat{A}_t \nabla_\theta r_t(\theta), \hat{A}_t \nabla_\theta clip(r_t(\theta), 1-\epsilon, 1+\epsilon) \Big) \Big] \tag{8}$$

where the expectation $\hat{\mathbb{E}}_t[\cdots]$ denotes the empirical average over a finite batch of samples, $r_t(\theta) = \frac{\pi_\theta(A_t|S_t)}{\pi_{\theta_{old}}(A_t|S_t)}$ denotes the probability ratio between current policy π_θ and old policy $\pi_{\theta_{old}}$, $\hat{A}_t = R(\pi|S) - B(S)$ is an estimator of the advantage function at timestep t, where $B(S)$ being the baseline independence on the policy π and estimates the expected trajectory length to reduce the variance of the gradients. Epsilon is a hyperparameter, say, $\epsilon = 0.2$ and the probability ratio $r_t(\theta)$ is clipped between interval $[1-\epsilon, 1+\epsilon]$, by increasing $r_t(\theta)$ at most 20% no matter how good the new policy is.

The proposed baseline $B(S)$, which is the estimated trajectory length value, is obtained from an auxiliary network, called a critic and parameterized by θ_v. The critic network is a many-to-one RNN architecture with LSTMs, where the value estimate or the baseline is predicted based on the final state input. The critic network parameters θ_v are trained in batches B using the stochastic gradient descent on a mean squared error objective between its predictions $B(S)$ and the reward trajectory length $R(\pi|S)$:

$$J(\theta_v) = \frac{1}{B} \sum_{i=1}^{B} \left(B(S_k) - R(\pi_k|S_k) \right)^2 \tag{9}$$

D. AUTONOMOUS CTPP IMPLEMENTATION BY HROMBO

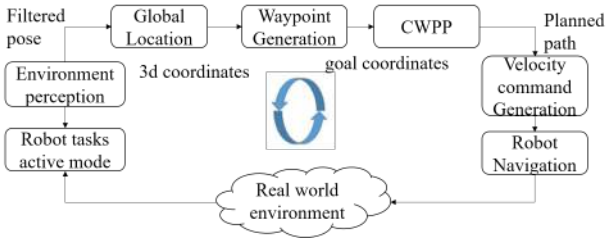

Figure 11: Flowchart of Autonomous Area Coverage by hRombo.

After CTPP got the required reference points and shapes, the autonomous navigation is triggered to let the hRombo began to cover the entire workspace, as shown in Figure 11. The autonomous framework relies on the open-source Robot Operating System [45]. During the programmed development process, the robot will continuously promote its current plan by focusing on reference-based on perception found by Ultra Wide Band sensor localization to acknowledge whether the waypoint has been visited and trigger the following required plan toward the next moves among transformation, orientation, and translation in order to clear all the waypoint sequentially.

If an abnormality between the current hRombo structure is found in the k pair at source reference point W_k^s and the associated structure at target point W_k^g in the direction, it will provide the request for the microcontroller in the robot to fulfill its structural movement request by command the servo motor turning to the predefined point. The current region of the robot x_h^w, y_h^w is continuously being observed to determine if the distinction between the robot region and the wanted area is a lower defined value. As the condition is verified, the robot takes the route to the associated improvement point. A similar procedure is performed for the following reference point until all actions stored in the robot database are cleared.

VI. EXPERIMENTAL RESULTS

In this section, after presenting the result and analysis of RL training, simulated workspaces and real environment setups are used to validate the outperformance of the proposed tiling-based complete coverage path planning framework for the hRombo robot in terms of saving navigation energy travel time.

A. RESULTS AND ANALYSIS OF RL TRAINING

We verified the performance of generated trajectories derived by different CTPP algorithms in simulated workspaces with rhombus-based tileset setups. Simulations of the rhombus grid-based workspace with various layout setups are generated by the Matlab Simulink. The grid cell is the same size as a hRombo block, as shown in Figure 12. Each four rhombus cells corresponding to the robot form are set with different colors to denote the robot shape identically inside the defined workspace. The cells corresponding with obstacle regions are placed randomly and colored as black with the value of -1. The backtracking algorithm loop over the entire workspace to generate the set of random tiles and cover the whole workspace. The optimal trajectories are plotted inside each workspace and denoted as brown arrows linking tiles in order. To demonstrate the novelty of hRombo shape-shifting, the complex workspaces such as Figure 12 (c) and (d) are generated so that only fixed robot form such as D or I shape will fail to cover completely without overlapped cells.

We used Tensorflow RL software (with pointer network for TSP) and changed REINFORCE loss to PPO. The zigzag, spiral, greedy search, genetic algorithm, and ant colony optimization are coded using python3 in the ubuntu version. All experiments run on computing nodes with the following specs: Intel Core i7-9750H processor and 16GB Memory with GPU Nvidia Quadro P620. We have experimented with 1000 graphs of 20,50 and 100-waypoint

instances of TSP. The mini-batch is set to 256 sequences with length n = 10, n = 20 and n = 50. The reward function considers the analysis energy usage during hRombo navigation within simulated workspace as Equation 5 are derived at each iteration step. The coefficient $\alpha = 0.3$ is selected based on the experimental trials. We use Adam optimizer [46] with an initial learning rate of 1e-3 to minimize the cross-entropy loss over each mini-batch.

The conventional TSP approaches that include zigzag, spiral, greedy search, genetic algorithm, and ant colony optimization are used to generate the trajectories cost-weights for each workspace to compare with the corresponding results of RL based proposed method. Figure 12 presents visualization for trajectory outputs of RL based method for different workspaces and tileset setups. The Figure 13 shows the comparison trajectories of all tested methods for workshops with obstacles of Figure 12 and the table 3 is numerical data for costweights trajectory generation time. Note that the cost function of RL as Equation 5 is also used during optimization processes of RL, evolutionary-based optimization Genetic Algorithm (GA) [8] and Ant Colony Optimization (ACO) [4]

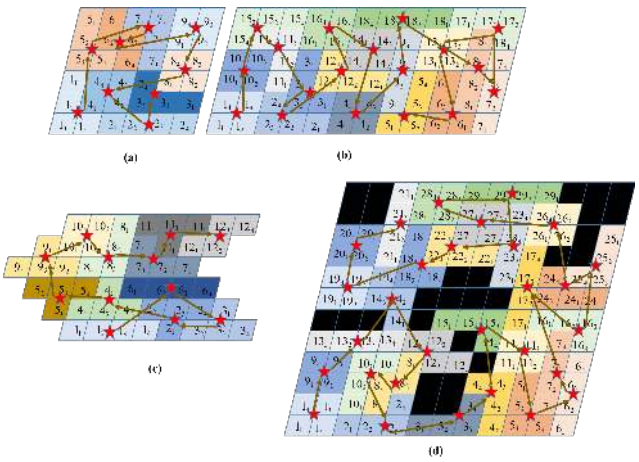

Figure 12: Worksapces with corresponding the trajectories by RL method. **(a)** 6x6; **(b)** 12x12; **(c)** random; **(d)** obstacles.

From the data in Table 3. All the tried-out techniques have comparable Euclidean length. As similar to [38] for small TSP, the solutions of the RL-TSP framework reached to the optimal cost weight for all tested workspaces. The difference between RL-TSP and evolutionary-based optimization GA and ACO also varies very slightly with relatively small numbers of waypoints. Although the fastest time is achieved, the simple zigzag and spiral techniques linking the pair by straight lines and outer-wise order produce weights slightly higher than the Greedy search. The running time and costweight of the greedy search are extremely higher than in GA and ACO strategies. Nevertheless, the RL-based approach archives both outperform in numerical values of costweight and generation time. The costweight of the RL-base method is slightly about 5% less than the second-best method as ACO.

Considering the technique based on RL, two reference points with similar morphology and less directional modification are chosen to pair within the found trajectory as in Figure 13f. Optimization for similar tile heading during path generation, RL frequently offers a higher priority to select the following waypoint with the cell of less directional adjustment. For example, with the same with Z shapes, from the tile 14, CTPP routes to the tile 10 instead of the tile 12 to be the next tile since there is no heading correction in rad is required. Besides, the RL optimization-based CTPP framework chooses the following tile that remains unchanged in shape or with fewer modules among four modules that need to rotate to shift the robot form to the next waypoint. For instance, from the tile 7 of I shape, it selects the tile 17 of the same I shape, even though the tiles $5, 11, 16$ have the shorter Euclidean distance. Moreover, from the tile 17, the proposed RL select the tile 16 of L shape, which requires only on module rotation of pi rad around hinge h_3 with a magnitude of l_1 rather than tile 26 of D shape, which requires two modules do a rotation of same π rad around hinge h_2 and h_3 (making the total 2π required rotation angle) with the magnitude of l_1 and l_2, respectively. As a result of reducing change steps and directions when moving away from the reference points with a predefined workspace, the minimum weight can be found by CTPP.

Table 3: Numerical costweight and trajectory generation time comparisons .

Approach	2D Distance (m)	Total Cost Weight (Nm)	Running Time (second)
Zigzag	21.63	195.93	0.01
Spiral	20.89	194.17	0.05
Greedy search	19.93	185.16	29.15
GA	19.52	156.86	5.25
ACO	19.82	155.69	5.25
RL	19.29	136.39	1.16
Optimal	18.02	136.39	

B. REAL ENVIRONMENT TESTBED

In real environment setup, the energy and travel time spent to complete the generated routes according to the instructions found in the planned database are estimated during robot navigation. Descriptions of the complete area coverage routes for the workspace (Figure 12d) is shown in Figure 14. The robot is set to autonomous mode and navigate sequentially to fit its COM to each defined waypoint, combining its desired location and shape. Their navigation includes the sequence of action among transformation, heading correction, and linear movement planned in an organized manner. Robot navigation works under the communication mechanism of the ROS network. The movement order by the proportional integral derivative(PID) controller [47] is loaded to the motor drivers to provide the proper linear speed for the DC motor and the rotation of the servo motor

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3038905, IEEE Access

Table 4: Numerical comparison for consumed energy and travel time in real testbed workspace

Method -	Costweight (Nm)	Summation Energy(J)	Translation Energy(J)	Transformation Energy(J)	Orientation Energy(J)	Travel Time(second)
Zigzag	195.93	41.63	19.39	12.32	9.92	691
Spiral	194.17	40.96	19.02	11.81	10.13	687
Greedy search	187.93	38.19	18.44	10.86	8.89	674
GA	156.86	35.86	16.69	9.83	9.34	656
ACO	155.69	35.59	16.41	9.61	9.57	648
RL	136.39	32.65	15.26	8.96	8.43	612

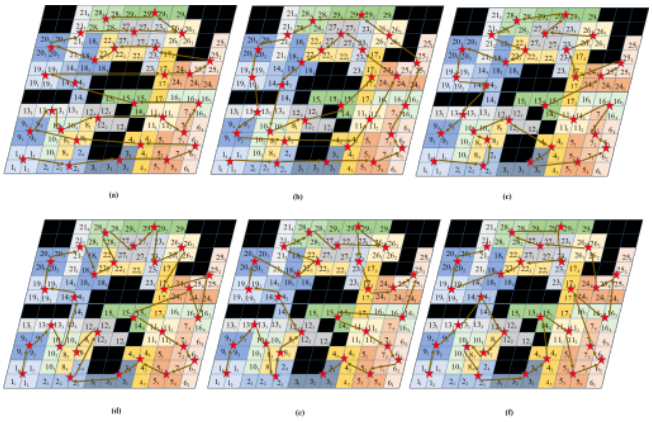

Figure 13: Generated trajectories by different tested methods . (**a**) Zigzag ; (**b**) Spiral; (**c**) Greedy search; (**d**) GA; (**e**) ACO; (**f**) Proposed RL TSP-based method.

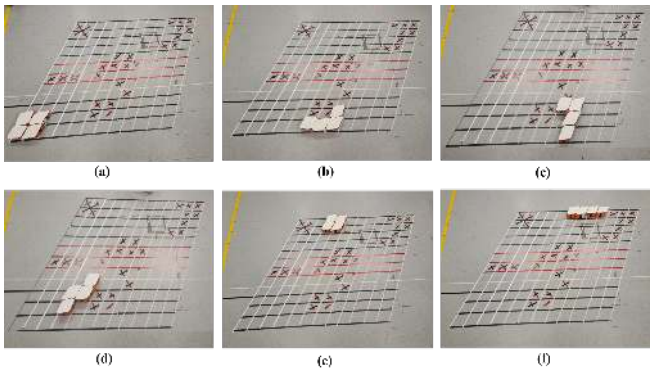

Figure 14: Real workspace setup with 29 waypoints similar as Figure 12d. **(a)** hRombo at waypoint 1; **(b)** hRombo at waypoint 3; **(c)** hRombo at waypoint 12; **(d)** hRombo at waypoint 9, **(e)** hRombo at waypoint 28, **(f)** hRombo at waypoint 29.

at the robot axes to change the robot forms. After the direction has been specified, the servo motor drives the steering units to this direction, then the bearings of the same parts as the DC motor are activated to conduct the linear motion. The real-time localization of the robots is enhanced by the various sensors function of the Kalman EKF approach that incorporates modern UWB frames and wheel encoders, and the IMU ensures robots comprehend the current location even in the case of any sensors struggles the malfunction or environment noise. Robot avoids the obstacles during the navigation. We can see at tile 9 in the workspace as Figure 14d for limited space; robots need to change to the I form to explore the narrow space between obstacles. The energy usage by of hRombo is determined using the current sensors connecting to the robot's main battery power 14.4V, 1000mAh. The current reading is set at 10 kHz. The DC motor is set with a maximum speed of 50 rpm.

Comparative analysis of energy and time spent by all the discussed strategies is presented in Table 4. From the given numerical comparison data, one can realize that if the robot follows the strategy's direction, which comes from the less costweight, the less energy usage can be archived. The best CTPP technique with the best energy and time usage is the proposed RL based method. This method yield is about ten % less than the ACO as the second-best technique. The results demonstrate that the proposed CTPP is a feasible technique that can be achieved energy-aware coverage planning by the hRombo tiling robot.

The energy spent on the single operation among transformation, heading correction, and linear movement to complete the tested directions is also given in Table 4. As per results, the linear motion spends the most battery energy because all three DC motors need to transmit the entire robot block, and all the guided servo motors are controlled to solve the problem, the change that brings more power usage. The transformation is the second; then, the heading correction is the third in energy usages.

VII. CONCLUSIONS AND FUTURE WORKS

The hRombo platform with reconfigurable forms provides an achievable answer to cover the various predefined workspace with saving power and consuming less time than conventional CTPP techniques. The RL based CTPP has proved to be outperformed in terms of deriving the shortest trajectory for proposed TSP than the conventional evolutionary-based methods such as GA and ACO. The proposed CTPP is ready to be applied flexibly to other tiling robot stages. The framework in this paper is the first step to implementing the proposed platform into the cleaning industry where the fixed form cleaning robots have constraints in covering the workspace of the complex environment.

Since the robot is underdeveloped and can operate with the relatively small workspace, testing the proposed method

on the bigger workspace to verify different RL-TSP frameworks is planned for future works. Since policy-based methods offer practical ways of dealing with large action spaces, exploring continuous action spaces in larger workspaces is also planned as future works. Alternatively, for the big workspace we can use cellular decomposition techniques such as hlto simplify the map to small sub-maps. The inspection opens up various potential researches that should be addressed, including optimal control methods. Future exploration works can be devised to follow: (1) a model for estimating vitality in a dynamic and bundled workspace, (2) Considering simultaneously how to generate tileset and trajectory by RL frameworks, (3) multi-objective RL, (4) RL policy-based methods continuous actions spaces with normal distributions (5) Focusing on long-distance independence with robot stage tiling motion. (6) Further studies on the power of devouring electrical parts, robot movements, and friction

References

[1] J. Yin, K. G. S. Apuroop, Y. K. Tamilselvam, R. E. Mohan, B. Ramalingam, and A. V. Le, "Table cleaning task by human support robot using deep learning technique," Sensors, vol. 20, no. 6, p. 1698, 2020.

[2] A. V. Le, A. A. Hayat, M. R. Elara, N. H. K. Nhan, and K. Prathap, "Reconfigurable pavement sweeping robot and pedestrian cohabitant framework by vision techniques," IEEE Access, vol. 7, pp. 159 402–159 414, 2019.

[3] L. Yi, A. V. Le, A. A. Hayat, C. S. C. S. Borusu, R. E. Mohan, N. H. K. Nhan, and P. Kandasamy, "Reconfiguration during locomotion by pavement sweeping robot with feedback control from vision system," IEEE Access, vol. 8, pp. 113 355–113 370, 2020.

[4] A. V. Le, P.-C. Ku, T. Than Tun, N. Huu Khanh Nhan, Y. Shi, and R. E. Mohan, "Realization energy optimization of complete path planning in differential drive based self-reconfigurable floor cleaning robot," Energies, vol. 12, no. 6, p. 1136, 2019, https://doi.org/10.3390/en12061136.

[5] M. Muthugala, A. V. Le, E. Sanchez Cruz, M. Rajesh Elara, P. Veerajagadheswar, and M. Kumar, "A self-organizing fuzzy logic classifier for benchmarking robot-aided blasting of ship hulls," Sensors, vol. 20, no. 11, p. 3215, 2020.

[6] V. Prabakaran, A. V. Le, P. T. Kyaw, R. E. Mohan, P. Kandasamy, T. N. Nguyen, and M. Kannan, "Hornbill: A self-evaluating hydro-blasting reconfigurable robot for ship hull maintenance," IEEE Access, vol. 8, pp. 193 790–193 800, 2020.

[7] B. Ramalingam, A. K. Lakshmanan, M. Ilyas, A. V. Le, and M. R. Elara, "Cascaded machine-learning technique for debris classification in floor-cleaning robot application," Applied Sciences, vol. 8, no. 12, p. 2649, 2018.

[8] A. Le, M. Arunmozhi, P. Veerajagadheswar, P.-C. Ku, T. H. Minh, V. Sivanantham, and R. Mohan, "Complete path planning for a tetris-inspired self-reconfigurable robot by the genetic algorithm of the traveling salesman problem," Electronics, vol. 7, no. 12, p. 344, 2018, https://doi.org/10.3390/electronics7120344.

[9] A. V. Le, V. Prabakaran, V. Sivanantham, and R. E. Mohan, "Modified a-star algorithm for efficient coverage path planning in tetris inspired self-reconfigurable robot with integrated laser sensor," Sensors, vol. 18, no. 8, p. 2585, 2018.

[10] A. V. Le, R. Parween, R. Elara Mohan, N. H. Khanh Nhan, and R. Enjikalayil, "Optimization complete area coverage by reconfigurable htrihex tiling robot," Sensors, vol. 20, no. 11, p. 3170, 2020.

[11] R. Parween, A. V. Le, Y. Shi, and M. R. Elara, "System level modeling and control design of htetrakis–a polyiamond inspired self-reconfigurable floor tiling robot," IEEE Access, vol. 8, pp. 88 177–88 187, 2020.

[12] A. V. Le and J. Choi, "Robust tracking occluded human in group by perception sensors network system," Journal of Intelligent & Robotic Systems, vol. 90, no. 3-4, pp. 349–361, 2018.

[13] A. Farooq, F. Farooq, and A. V. Le, "Human action recognition via depth maps body parts of action." TIIS, vol. 12, no. 5, pp. 2327–2347, 2018.

[14] 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, https://doi.org/10.1016/j.robot.2013.09.004.

[15] P. Veerajagadheswar, K. Ping-Cheng, M. R. Elara, A. V. Le, and M. Iwase, "Motion planner for a tetris-inspired reconfigurable floor cleaning robot," International Journal of Advanced Robotic Systems, vol. 17, no. 2, p. 1729881420914441, 2020.

[16] S. B. P. Samarakoon, M. V. J. Muthugala, A. V. Le, and M. R. Elara, "htetro-infi: A reconfigurable floor cleaning robot with infinite morphologies," IEEE Access, vol. 8, pp. 69 816–69 828, 2020.

[17] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, "Morse decompositions for coverage tasks," The International Journal of Robotics Research, vol. 21, no. 4, pp. 331–344, 2002, https://doi.org/10.1177/027836402320556359.

[18] K. P. Cheng, R. E. Mohan, N. H. K. Nhan, and A. V. Le, "Graph theory-based approach to accomplish complete coverage path planning tasks for reconfigurable robots," IEEE Access, vol. 7, pp. 94 642–94 657, 2019.

[19] A. V. Le and T. D. Do, "High-order observers-based lq control scheme for wind speed and uncertainties estimation in wecss," Optimal Control Applications and Methods, vol. 39, no. 5, pp. 1818–1832, 2018.

[20] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in 1985 IEEE International Conference on Robotics and Automation, vol. 2. IEEE, 1985, pp. 116–121, https://doi.org/10.1109/ROBOT.1985.1087316.

[21] H. Choset, "Coverage for robotics - a survey of recent results," Annals of Mathematics and Artificial Intelligence, vol. 31, no. 1, pp. 113–126, Oct 2001, https://doi.org/10.1023/A:1016639210559.

[22] S. X. Yang and C. Luo, "A neural network approach to complete coverage path planning," IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 34, no. 1, pp. 718–724, 2004, https://doi.org/10.1109/TSMCB.2003.811769.

[23] Y. Gabriely and E. Rimon, "Spiral-stc: An on-line coverage algorithm of grid environments by a mobile robot," in 2002 IEEE International Conference on Robotics and Automation, vol. 1. IEEE, 2002, pp. 954–960, https://doi.org/10.1109/ROBOT.2002.1013479.

[24] A. Manimuthu, A. V. Le, R. E. Mohan, P. Veerajagadeshwar, N. Huu Khanh Nhan, and K. Ping Cheng, "Energy consumption estimation model for complete coverage of a tetromino inspired reconfigurable surface tiling robot," Energies, vol. 12, no. 12, p. 2257, 2019.

[25] J. H. Conway and J. C. Lagarias, "Tiling with polyominoes and combinatorial group theory," Journal of Combinatorial Theory, Series A, vol. 53, no. 2, pp. 183–208, 1990, https://doi.org/10.1016/0097-3165(90)90057-4.

[26] K. P. Cheng, R. E. Mohan, N. H. K. Nhan, and A. V. Le, "Multi-objective genetic algorithm-based autonomous path planning for hinged-tetro reconfigurable tiling robot," IEEE Access, vol. 8, pp. 121 267–121 284, 2020.

[27] A. V. Le, N. H. K. Nhan, and R. E. Mohan, "Evolutionary algorithm-based complete coverage path planning for tetriamond tiling robots," Sensors, vol. 20, no. 2, p. 445, 2020.

[28] C. You, J. Lu, D. Filev, and P. Tsiotras, "Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning," Robotics and Autonomous Systems, vol. 114, pp. 1–18, 2019, https://doi.org/10.1016/j.robot.2019.01.003.

[29] K. Lobos-Tsunekawa, F. Leiva, and J. Ruiz-del Solar, "Visual navigation for biped humanoid robots using deep reinforcement learning," IEEE Robotics and Automation Letters, vol. 3, no. 4, pp. 3247–3254, 2018, https://doi.org/10.1109/LRA.2018.2851148.

[30] F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, "Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments," IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 610–617, 2019, https://doi.org/10.1109/LRA.2019.2891991.

[31] A. I. Panov, K. S. Yakovlev, and R. Suvorov, "Grid path planning with deep reinforcement learning: Preliminary results," Procedia Computer Science, vol. 123, pp. 347–353, 2018, https://doi.org/10.1016/j.procs.2018.01.0545.

[32] A. Konar, I. G. Chakraborty, S. J. Singh, L. C. Jain, and A. K. Nagar, "A deterministic improved q-learning for path planning of a mobile robot," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 43, no. 5, pp. 1141–1153, 2013, https://doi.org/10.1109/TSMCA.2012.2227719.

[33] E. S. Low, P. Ong, and K. C. Cheah, "Solving the optimal path planning of a mobile robot using improved q-learning," Robotics and Autonomous Systems, vol. 115, pp. 143–161, 2019, https://doi.org/10.1016/j.robot.2019.02.013.

[34] D. L. Cruz and W. Yu, "Path planning of multi-agent systems in unknown environment with neural kernel smoothing and rein-

forcement learning," Neurocomputing, vol. 233, pp. 34–42, 2017, https://doi.org/10.1016/j.neucom.2016.08.108.

[35] J. Yuan, H. Wang, C. Lin, D. Liu, and D. Yu, "A novel gru-rnn network model for dynamic path planning of mobile robot," IEEE Access, vol. 7, pp. 15 140–15 151, 2019, https://doi.org/10.1109/ACCESS.2019.2894626.

[36] A. K. Lakshmanan, R. E. Mohan, B. Ramalingam, A. V. Le, P. Veerajagadeshwar, K. Tiwari, and M. Ilyas, "Complete coverage path planning using reinforcement learning for tetromino based cleaning and maintenance robot," Automation in Construction, vol. 112, p. 103078, 2020.

[37] "A polyomino tiling algorithm," https://gfredericks.com/gfrlog/99, 2018, [Online; accessed 15-July-2020].

[38] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, "Neural combinatorial optimization with reinforcement learning," arXiv preprint arXiv:1611.09940, 2016.

[39] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in Advances in neural information processing systems, 2000, pp. 1008–1014.

[40] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in Advances in neural information processing systems, 2015, pp. 2692–2700.

[41] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[42] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347, 2017.

[43] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in Proceedings of the 32nd International Conference on International Conference on Machine Learning, 2015, pp. 1889–1897, https://dl.acm.org/citation.cfm?id=3045319, Accessed date: 2 January 2020.

[44] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," arXiv preprint arXiv:1509.02971, 2015.

[45] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in ICRA workshop on open source software, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[47] Y. Shi, M. R. Elara, A. V. Le, V. Prabakaran, and K. L. Wood, "Path tracking control of self-reconfigurable robot htetro with four differential drive units," IEEE Robotics and Automation Letters, vol. 5, no. 3, pp. 3998–4005, 2020.

ANH VU LE received the B.S. degree in electronics and telecommunications from the Hanoi University of Technology, Vietnam, in 2007, and the Ph.D. degree in electronics and electrical from Dongguk University, South Korea, in 2015. He is currently with the Opto-electronics Research Group, Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam. He is also a Postdoctoral Research Fellow with the ROAR Laboratory, Singapore University of Technology and Design. His current research interests include robotics vision, robot navigation, human detection, action recognition, feature matching, and 3D video processing

RIZUWANA PARWEEN (Member, IEEE) received the bachelor's and master's degrees in mechanical engineering from the National Institute of Technology Rourkela, India, and the Ph.D. degree from the Indian Institute of Science, Bengaluru, India. She has over two years of industrial experience as a Product Development Engineer (KSB Tech Private Ltd., Pune) and a Structural Analyst (CUMMINS, Pune). As a Postdoctoral Research Fellow at the Singapore University of Technology and Design (SUTD), she worked on the design and development of Unloader Knee Brace for Asian Patients, in collaboration with physicians at the Changi General Hospital, Singapore. She is currently a Research Fellow with the Engineering Product Development Pillar, SUTD, where she is involved in the design, development, and modelling of the self-reconfigurable floor cleaning robots.

PHONE THIHA KYAW is currently working as a Visit Fellow in the Robotics and Automation Research Laboratory (ROAR) at the Singapore University of Technology and Design. He is also a final year student in B.E. Mechatronics from Yangon Technological University. His research interests include autonomous robots, sensor fusion systems, control engineering, and computer vision applications. He participated in many different robotic competitions, including First Global Challenge 2017, which was held in Washington DC, and his Team Myanmar achieved rank 6 out of 163 teams.

MOHAN RAJESH ELARA received the B.E. degree from the Bharathiar University, India, in 2003, and the M.Sc. and Ph.D. degrees from Nanyang Technological University in 2005 and 2012, respectively. He is currently an Assistant Professor with the Engineering Product Development Pillar, Singapore University of Technology and Design. He is also a Visiting Faculty Member with the International Design Institute, Zhejiang University, China. He has published over 80 papers in leading journals, books, and conferences. His research interests are in robotics with an emphasis on self-reconfigurable platforms as well as research problems related to robot ergonomics and autonomous systems. He was a recipient of the SG Mark Design Award in 2016 and 2017, the ASEE Best of Design in Engineering Award in 2012, and the Tan Kah Kee Young Inventors' Award in 2010.

TRAN HOANG QUANG MINH received his Ph.D. from Tomsk Polytechnic University, Tomsk City, Russian Federation. His research interests include high-voltage power systems, optoelectronics, wireless communications and network information theory. He serves as Lecturer in the Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam.

CHARAN SATYA CHANDRA SAIRAM BORUSU received his Bachelors of Technology in Electronics and Communication Engineering in Amrita Vishwa Vidyapeetham in 2019. He is currently working as a visiting research fellow in ROAR Lab at Singapore University of Technology and Design, Singapore. He worked in Robert Bosch, Coimbatore as an Associate engineer. He also worked as a Research assistant in Humanitarian Technology (HuT) lab, a robotics research laboratory in Amrita School Of Engineering, Amritapuri. His research interests include Robotics and VLSI.