A Bioinspired Neural Network for Real-Time Concurrent Map Building and Complete Coverage Robot Navigation in Unknown Environments

Chaomin Luo, *Student Member, IEEE*, and Simon X. Yang, *Member, IEEE*

***Abstract*—Complete coverage navigation (CCN) requires a special type of robot path planning, where the robots should pass every part of the workspace. CCN is an essential issue for cleaning robots and many other robotic applications. When robots work in unknown environments, map building is required for the robots to effectively cover the complete workspace. Real-time concurrent map building and complete coverage robot navigation are desirable for efficient performance in many applications. In this paper, a novel neural-dynamics-based approach is proposed for real-time map building and CCN of autoxnomous mobile robots in a completely unknown environment. The proposed model is compared with a triangular-cell-map-based complete coverage path planning method (Oh *et al.*, 2004) that combines distance transform path planning, wall-following algorithm, and template-based technique. The proposed method does not need any templates, even in unknown environments. A local map composed of square or rectangular cells is created through the neural dynamics during the CCN with limited sensory information. From the measured sensory information, a map of the robot's immediate limited surroundings is dynamically built for the robot navigation. In addition, square and rectangular cell map representations are proposed for real-time map building and CCN. Comparison studies of the proposed approach with the triangular-cell-map-based complete coverage path planning approach show that the proposed method is capable of planning more reasonable and shorter collision-free complete coverage paths in unknown environments.**

***Index Terms*—Complete coverage navigation (CNN), map building, mobile robots, neural dynamics, unknown environments.**

I. Introduction

COMPLETE coverage navigation (CCN) needs the robot path to pass through every region of the workspace. In addition to cleaning robots, many other robotic applications also require CCN, e.g., painter robots, autonomous underwater covering vehicles, demining robots, lawn mowers, automated harvesters, crop harvesting equipment, and window cleaners. CCN of mobile robots has been extensively studied. Many previous approaches tackle *stationary* and *known* environments only. For instance, González *et al.*'s approach [1] splits static workspace into elementary regions, using a topological graph representation of the environment for planning coverage and point-to-point paths. In this grid-like map, the cells are grouped together into regions. The regions connected in the graph are recursively defined and covered. Acar *et al.* [2] used exact cell decomposition with critical points for complete coverage exploration, which guides the robot to cover the unstructured environments with back-and-forth patterns. This algorithm does not need any prior knowledge of the environment but can only deal with static environments. Wong and MacDonald [3] built a topological map with natural landmarks, which is capable of constructing a planar graph for coverage algorithms. A static environment can be decomposed into cells and represented by a topological map. Conventional zigzag pattern for complete coverage path is replaced with a spiral path by the backtracking spiral algorithm in static environments [4]. An advantage of this method is that the robot is capable of moving back to unfilled areas, thus complete coverage can be easily achieved.

The methods in [1], [2], [5], and [6] for CNN assume static and known environments, which are not practicable in real-world applications as the environments may be partially or completely unknown in most cases. Therefore, map building is a necessary and important concern in CCN. There have been some studies in map building and its implementation in coverage path planning [7]–[9]. Some approaches use neural networks for robot map building and map learning [10], [11]. To navigate a cleaning robot in a completely unknown environment, both map representation and CCN algorithms are indispensable. The literature addressing concurrent map building and CCN is rather limited [8], [12].

Previous research on CCN may be classified into sensor-based models, cell-decomposition-based models, template-based approaches, neural-network-based models, and others. Sensor-based coverage approaches use incoming sensor data to perform CCN in an initially unknown environment. Choset *et al.* [13] suggested a sensor-based CCN approach for demining robots by combining sensor information with an exact cell decomposition. The approaches enable robots to simultaneously cover a cell and search critical points of the entire unknown environment. Acar *et al.* [14] combined their previous sensor-based work with a generalized Voronoi diagram approach. This sensor-based coverage algorithm guides the

Manuscript received June 6, 2007; revised November 19, 2007; accepted December 30, 2007. First published June 10, 2008; last published July 7, 2008 (projected). This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

C. Luo is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: c2luo@engmail.uwaterloo.ca).

S. X. Yang is with the Advanced Robotics and Intelligent Systems (ARIS) Laboratory, School of Engineering, University of Guelph, Guelph, ON N1G 2W1, Canada (e-mail: syang@uoguelph.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2008.2000394

robot to pass over all points in vast workspaces. Lumelsky *et al.* [15] considered the sensor-based approach proposed for the problem of terrain acquisition, which requires a robot to build a complete map and cover the whole terrain in an unknown environment based on the sensory feedback. Autonomous cleaning robots require map information from robot sensors to navigate in complex and partially unknown environments [15]. Floor map could be obtained by vision-guided range detection sensors [16]. Park and Lee [17] proposed a CCN algorithm in unknown environments that are composed of three components: a sweeping algorithm, a point-to-point moving algorithm, and a corner work algorithm. This model builds an information bitmap and applies these three algorithms to generate coverage paths. The cleaning robots may overlap some areas and miss some corner areas. Butler *et al.* [18] suggested a distributed coverage algorithm, where the robot can detect obstacles by only sensing in a shared, connected rectilinear environment. In this approach, the robot collects incoming sensor data in an initially unknown environment to plan a coverage path.

Much previous effort on CCN has relied on predefined templates for navigation of cleaning robots. For example, Schmidt and Hofner [5] proposed a template-based approach using motion templates and motion mosaic. This model requires a predefined map and memorizes the templates. Thus, it is difficult to handle environmental changes. In addition, it has difficulty identifying where it is covered or uncovered [5], [19]. More recently, a template-based complete coverage strategy proposed by Liu *et al.* [20] integrates multilayer structure, motion template, and task-based methods to implement complete coverage for a cleaning robot.

There have been some neural-network-based approaches to path planning [21]–[24] and CNN [25]. Most neural-network-based models require learning procedures [6], [21], [26], which are computationally expensive and difficult to achieve CCN in real time. Tse *et al.* [6] proposed a neural-network-based CCN and point-to-point navigation method for cleaning robots. A global map is built in the path planning phase and the robot can navigate from the starting position to traverse the entire workspace with the built map relying on a learning procedure. Yasutomi *et al.* [26] suggested a learning-based CCN approach using grid-like map representation of workspace in an unknown environment. As a learning procedure is needed, the robot must depend on prior complex environmental information. Thus, it is suitable for well-structured indoor environments only. Some of previous work combines several approaches together to achieve map building and CCN. Combining template-based model with other approaches is a good solution for CCN in unknown environments [27], [28]. Cell decomposition methods break continuous space into a finite set of cells. Choset [27] combines the advantages of cell-decomposition- and template-based approaches, and minimizes the number of cells used in cell decomposition. It requires the prior knowledge of the obstacle positions. Oh *et al.* [28] developed a triangular cell decomposition approach for unknown environments, where triangular cell map representation enables the cleaning robot navigate along shorter and more flexible paths. The cleaning robot might move to eight potential directions in the rectangular or square cell map model, while the triangular map representation increases the navigational directions to 12. Seven templates are proposed for coverage path planning. This method combines triangular cell decomposition, a template-based approach, and a wall-following navigation algorithm for CCN [28].

Yang and Luo [25] proposed a neural network model for complete coverage path planning in nonstationary environments. However, the current knowledge of the nonstationary environment was assumed to be completely known. In this paper, map-building and coverage algorithms are proposed to extend the work of Yang and Luo [25] to *completely unknown* environments, using onboard robot sensor information only. A novel biologically inspired neural dynamics model and a map-building-based coverage algorithm are proposed for CCN of cleaning robots in completely unknown environments. A local map composed of square or rectangular cells is built through the neural dynamics during complete coverage robot navigation with restricted sensory information. The robot is able to dynamically build a map of its immediate limited surroundings. The onboard robot sensors have only a limited reading range and the obtained sensory information is used for its navigation. It is assumed that the robot can accurately obtain all the environment information within its reading range without any uncertainties through its onboard sensors (multisensory fusion to handle the sensor failure and sensor noise and to obtain its accurate surrounding environment knowledge is beyond the scope of this paper). The state space of the topologically organized neural network is the 2-D Cartesian workspace of the cleaning robot. The dynamics of each neuron is characterized by a shunting equation derived from Hodgkin and Huxley's [29] membrane model for a biological nervous system. There are only local lateral connections among neurons. Thus, the computational complexity depends linearly on the neural network size. The varying environment is represented by the dynamic activity landscape of the neural network. For cost efficiency in terms of shorter path and less turning, the real-time robot path is directly planned from the dynamic neural activity landscape and the previous robot position without any *prior* knowledge of the changing environment, and without any explicit optimizations of any *global* cost functions. Since the neural connection weights are set in the model design and can be chosen in an extensive range, there is no need to find an optimal connection weights among neurons. Thus, there are no learning procedures in the proposed neural network. Simulation results verify that the model does not need any templates in unknown environments during the CCN.

The rest of this paper is organized as follows. In Section II, cleaning robots and map representations of the environments are introduced. The biological inspiration, the model algorithm, and the stability analysis of the proposed neural-dynamics-based approach to real-time map building and collision-free CCN of the mobile robot are presented in Section III. The complete coverage and map-building algorithms are described in Section IV. After that, simulation and comparison studies in various unknown environments are given in Sections V and VI, respectively. In Section VII, parameter sensitivity of the proposed map-building and CCN algorithm is discussed. Finally, several important properties of the proposed approach are concluded in Section VIII.

II. Cleaning Robots, Environments, and Map Representations

Cleaning robots play an increasingly significant role in daily life and industry. To design a cleaning robot, map building and collision-free navigation are two important aspects of cleaning robots. If a cleaning robot equipped with various sensors is required to navigate in an unknown environment, both map representation and environmental information are fundamental requirements for CCN. A qualitative description of their properties is conducted in this section.

A. Cleaning Robots

Cleaning robots have been extensively used to save time and labor cost. Cleaning work falls into the category of "dirty, dull, and dangerous (3d)" for robotics as floor cleaning is a repetitive and monotonous task. Autonomous cleaning robots are particularly useful in hazardous environments, where the area is inaccessible to humans. Cleaning robots may be classified into several categories based on their special purposes such as autonomous vacuum cleaners [30]–[35], robotic floor scrubbers [36], autonomous pool cleaners [37], liquid storage tank cleaning robots [38], robotic road sweepers [39], [40], bathroom cleaning robots [41], facade cleaning robots [42], waste cleanup robots [43], and window cleaners [44], [45]. Recently, a type of domestic service robots were designed to execute tasks for materials delivery and security surveillance, in addition to floor cleaning in the domestic home environments [46]–[48].

Research on cleaning robots has attracted significant attention from both academia and industry [13], [49]–[53]. For example, Schraft *et al.* [45] developed a cleaning robot to clean the outside of windows on a facade. Prassler *et al.* [54] introduced a short development history of cleaning robots. Kamiya *et al.* [55] proposed a hybrid vehicle control system for cleaning robots by unifying the navigation and dead-reckoning control. Branch [36] spent 14 years developing a floor cleaning robot. The path planner adopts a tree graph to form repeated parallel paths for navigation after it has followed the related walls and thus has built a reference map.

One of the most important reasons for developing cleaning robots is because there is a great market demand [56], leading to an extensive application. For instance, they can be used for cleaning of rolling stock, rail cars and stations [57], and cleanup in a construction site [58]. Yaguchi *et al.* [57] introduced four types of cleaning robots for labor savings and safety purpose. In addition, much research effort has being directed toward window cleaning robots, e.g., the cleaning of glass surfaces of high-rise buildings. Ratcliff *et al.* [44] developed an efficient four-axis Cartesian $X - Y - Z$ window cleaning robots. Wall-climbing robots are extended applications of cleaning robots, which are usually intended for cleaning motion on vertical and sloping surfaces in building construction. Wang *et al.* [59] developed a wall-climbing robot for cleaning of high-rise buildings with both ceramic tile and glass surface. Much research work on cleaning robots has been conducted by many researchers [58], [60]–[63].

Research on cleaning robots involves almost all the basic fields in robotics. A cleaning robot consists of body, mind, operator interface, interaction with environment, and navigation system [50]. The term "mind" is referred to environment considerations, commercial and legal considerations, operational requirements, sensor requirements, and reasoning requirements [64], [65]. The cleaning navigation system is the most important component in cleaning robots [39], [66]–[68], which is the focus of this paper.

A cleaning robot is required to move through an entire workspace, i.e., the overall travel must cover the whole region without any overlapping paths or with the smallest overlapping space. Minimal turns of cleaning robots in continuous and sequential paths are desirable. Simple motion trajectories such as straight lines, zigzags, spirals, or circles, should be used for simplicity in motion control. The complete coverage paths are planned based on minimizing the number of rules. The overlapping cells and the number of turns, as important factors for performance evaluation, will be computed in the simulations of this paper.

B. Environments and Discretization

The underlying idea of the proposed model is to develop a neural network architecture whose dynamic neural activity landscape represents the dynamically changing environment. By properly defining the external inputs from the changing environment and internal neural connections, the unclean areas and obstacles are guaranteed to stay at the peak and the valley of the activity landscape of the neural network, respectively. The unclean areas globally attract the robot in the whole workspace through neural activity propagation, while the obstacles have only local effect in a small region to avoid collisions. The collision-free robot motion is planned in real time based on the dynamic activity landscape of the neural network and the previous robot position, such that all areas will be cleaned and the robot will travel along a smooth zigzag path.

The 2-D Cartesian workspace in the proposed approach is discretized into squares (instance) or rectangles as with most other CCN approaches. The size of the 2-D Cartesian workspace discretized by squares or rectangles for the neural system is assumed to be known. However, the environmental knowledge of the dynamically changing environment (e.g., the locations of the static or moving obstacles, the cleaned areas, and the unclean areas) is completely unknown. The diagonal length of each discrete area should be equal to a robot sweeping radius, which is the size of the robot effector or footprint (Fig. 1). Each rectangle has a height of h and a width of l [see (1)]. The robot in this paper is assumed to be round in shape and a rectangle is enclosed in this round as robot's body. A robot sweeping range, which is the size of the robot effector, is proportional to the size of each rectangle. The size of a robot is slightly larger than that of each rectangle. However, the aspect ratio of the robot, which is the ratio of height and width of the robot body, is equal or proportional to that of each rectangle r_a [see (2)]. Therefore, sweeping an area can be achieved by traversing the center of that area represented by a rectangular cell.

It is assumed that a discrete area represented by that rectangular cell is regarded to be covered once a robot visits the discrete cell. If a cleaning robot covers every discrete cell, the robot path is considered as a CCN in the workspace.

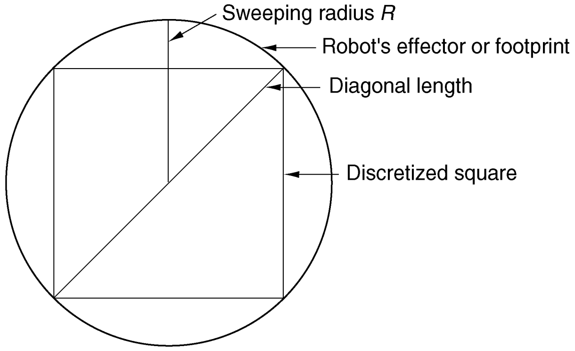


Fig. 1. Discretized square enclosed by sweeping area.

C. Map Representations

There are a variety of map representations depending on different decompositions of workspace. In this paper, square and rectangular cell decomposition are developed as map representations.

1) Square Cell Representations: In our previous research [25], the workspace was broken down into cells which are square. A local map composed of squared cells is constructed through the proposed neural dynamics during CCN with restricted sensory information in unknown environments in this paper. The 2-D Cartesian workspace in the proposed approach is discretized into squares.

2) Rectangular Cell Representations: Oh *et al.* [28] proposed a triangular cell decomposition approach for unknown environments, where triangular cell map representation enables the cleaning robot to move along 12 potential directions. The triangular cell decomposition is shown in Fig. 2, where a workspace may be decomposed into triangular cells or rectangular cells. In Fig. 2, the equilateral triangle (thus the equiangular triangle) ABC is a cell having three sides of equal length of l and the height of h

$$h = \frac{\sqrt{3}}{2} l = 0.866l. \tag{1}$$

Based on the triangular cell decomposition, the workspace may also be decomposed into rectangular cells with the height of h and the width of l. The rectangle is illustrated by the dashed line in Fig. 2. Therefore, the aspect ratio r_a of each rectangular cell denoting the ratio of height and width is given as

$$r_a = \frac{h}{l} = 0.866. \tag{2}$$

The relationship of the triangular cell and rectangular cell map representations is illustrated in Fig. 2.

III. The Model

A computational model for a uniform patch of membrane in a biological neural system using electrical circuit elements was first proposed by Hodgkin and Huxley [29]. In this model, the dynamics of voltage across the membrane V_m is described using a state equation as

$$C_m \frac{dV_m}{dt} = -(V_r + V_m)g_r + (V_e - V_m)g_e - (V_s + V_m)g_s \tag{3}$$

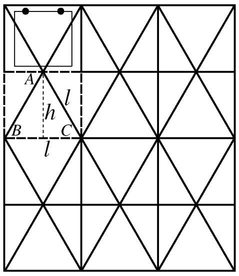


Fig. 2. Triangular cell and rectangular cell representations of workspace.

where C_m is the membrane capacitance; V_s, V_e, and V_r are the resting potentials (saturation potentials) in the membrane; and g_s, g_e, and g_r represent the synaptic and resting conductances. This model lays the foundation for the shunting model.

By setting $C_m = 1$ and substituting $x_i = V_r + V_m$, $A = g_r$, $B = V_e + V_r$, $D = V_s - V_r$, $S_i^e = g_e$, and $S_i^i = g_s$ in (3), a shunting equation is obtained [69] as

$$\frac{dx_i}{dt} = -Ax_i + (B - x_i)S_i^e - (D + x_i)S_i^i \tag{4}$$

where x_i is the neural activity (membrane potential) of the ith neuron in the neural network; A, B, and D are nonnegative constants describing the passive decay rate and the upper and lower bounds of the neural activity x_i, respectively; and S_i^e and S_i^i are the excitatory and inhibitory inputs to the neuron, respectively.

In the proposed model, the collision-free robot motion is planned in real time based on the dynamic activity landscape of the neural network and the previous robot position, to guarantee all areas to be cleaned and the robot to travel a smooth, continuous path with less turning.

The proposed topologically organized model is expressed in a 2-D Cartesian workspace $\mathcal{W}$ of the cleaning robots. The position of the ith neuron in the state space $\mathcal{S}$ of the neural network, denoted by a vector $q_i \in R^2$, *uniquely* represents a position in $\mathcal{W}$. In the proposed model, the excitatory input results from the unclean areas and the lateral neural connections, while the inhibitory input results from the obstacles only. Each neuron has local lateral connections to its neighboring neurons that constitute a subset $\mathcal{R}_i$ in $\mathcal{S}$. The subset $\mathcal{R}_i$ is called the receptive field of the ith neuron in neurophysiology. The neuron responds only to the stimulus within its receptive field. Thus, the dynamics of the ith neuron in the neuron network is characterized by a shunting equation as

$$\frac{dx_i}{dt} = -Ax_i + (B - x_i)\left([I_i]^+ + \sum_{j=1}^{k} w_{ij}[x_j]^+\right) - (D + x_i)[I_i]^- \tag{5}$$

where k is the number of neural connections of the ith neuron to its neighboring neurons within the receptive field $\mathcal{R}_i$. The external input I_i to the ith neuron is defined as $I_i = E$, if it is an unclean area; $I_i = -E$, if it is an obstacle area; $I_i = 0$, if it is a cleaned area, where $E \gg B$ is a very large positive constant, which guarantees that the target (unclean areas) can always attain at the peak and the obstacles can always stay in the valley in the neural activity landscape of the neural network.

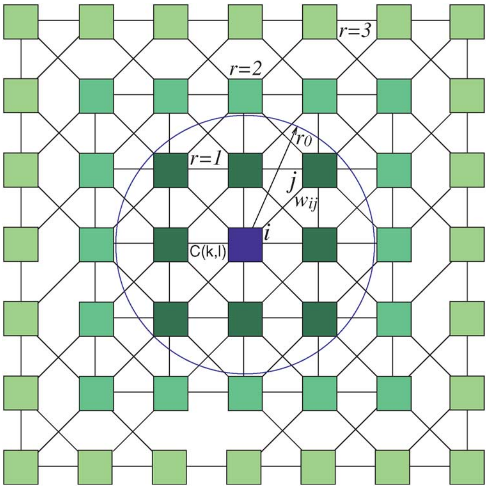


Fig. 3. Architecture of a 2-D neural network with three-layer neighboring neurons with regard to the central neuron C.

The terms $[I_i]^+ + \sum_{j=1}^{n} w_{ij}[x_j]^+$ and $[I_i]^-$ are the excitatory and inhibitory inputs S_i^e and S_i^i in (4), respectively. Function $[a]^+$ is a linear-above-threshold function defined as $[a]^+ = \max\{a, 0\}$, and function $[a]^-$ is defined as $[a]^- = \max\{-a, 0\}$. The connection weight w_{ij} from the ith neuron to the jth neuron is given by $w_{ij} = f(|q_i - q_j|)$, where $|qi - q_j|$ represents the Euclidean distance between vectors q_i and q_j in the state space, and $f(a)$ is a monotonically decreasing function, e.g., a function defined as

$$f(a) = \begin{cases} \mu/a, & \text{if } 0 \le a < r_0 \\ 0, & \text{if } a \ge r_0 \end{cases} \tag{6}$$

where μ and r_0 are positive constants. Therefore, each neuron has only local lateral connections in a small region $[0, r_0]$. It is obvious that the weight w_{ij} is symmetric, i.e., $w_{ij} = w_{ji}$. A schematic diagram of the neural network in 2-D is shown in Fig. 3, where r_0 is chosen as $r_0 = 2$. The receptive field of the ith neuron is represented by a circle with a radius of r_0. The ith neuron has only eight lateral connections to its neighboring neurons that are within its receptive field.

The proposed network characterized by (5) guarantees that the positive neural activity can propagate to all the state space, but the negative activity only stays locally. Therefore, the unclean areas globally attract the robot, while the obstacles only locally push the robot away to avoid collisions. The positions of the unclean areas and obstacles may vary with time, e.g., there are moving obstacles; the cleaned areas become unclean again. The activity landscape of the neural network dynamically changes due to the varying external inputs from the unclean areas, obstacles, and the internal activity propagation among neurons.

For energy and time efficiency, the robot should travel the shortest path with the least revisited clean areas and make the least turns of moving directions. In the proposed model, the robot path is generated from the dynamic activity landscape *and* the previous robot position to avoid least navigation direction changes. For a given current robot position in $\mathcal{S}$ (i.e., a position in $\mathcal{W}$), denoted by p_c, the next robot position p_n (also called "command position") is obtained by

$$p_n \quad \Leftarrow \quad x_{p_n} = \max\{x_j + cy_j, j = 1, 2, \cdots, k\} \tag{7}$$

where c is a positive constant and k is the number of *neighboring neurons* of the p_cth neuron, i.e., all the possible next positions of the current position p_c. Variable x_j is the neural activity of the jth neuron that is the same as that in (4); y_j is a monotonically increasing function of the difference between the current to next robot moving directions, which can be defined as a function of the previous position p_p, the current position p_c, and the possible next position p_j, e.g., a function defined as

$$y_j = 1 - \frac{\Delta\theta_j}{\pi} \tag{8}$$

where $\Delta\theta_j \in [0, \pi]$ is the turning angle between the current moving direction and next moving direction, e.g., if the robot goes straight, $\Delta\theta_j = 0$; if it goes backward, $\Delta\theta_j = \pi$. Thus, $\Delta\theta_j$ can be given as $\Delta\theta_j = |\theta_j - \theta_c| = |\text{atan2}(y_{p_j} - y_{p_c}, x_{p_j} - x_{p_c}) - \text{atan2}(y_{p_c} - y_{p_p}, x_{p_c} - x_{p_p})|$. After the current position reaches its next position, the next position becomes a new current position (if the found next position is the same as the current position, the robot stays there without any movement). The current robot position *adaptively* changes according to the varying environment.

The proposed neural network is a stable system. The neural activity x_i is bounded in the finite interval $[-D, B]$ [70]. The stability and convergence of the present shunting neural network model can also be rigorously proved using a Lyapunov stability theory. By introducing new variables and performing variable substitutions, (4) or (5) can be written as

$$\frac{dz_i}{dt} = a_i(z_i)\left(b_i(z_i) - \sum_{j=1}^{N} c_{ij} d_j(z_j)\right) \tag{9}$$

which is Grossberg's general form [70]. It can be proved that (4) or (5) satisfies all the three stability conditions required by the Grossberg's general form [70]. The rigorous proof of the stability and convergence of (9) can be found in [71]. The dynamics of the neural network is guaranteed to converge to an equilibrium state of the system. Equation (5) combined with previous robot position ensures to generate complete coverage path. At the beginning, when $t = 1$, the neural activity of all neurons is set to zero. The state of the workspace varies in terms of the dynamics of the neural network described by (5) due to the influence of external inputs. The planned motion ends when the network reaches a steady state.

IV. The Coverage and Map-Building Algorithms

This section presents algorithms for map-building and complete coverage tasks. The mission of the cleaning robots is to cover unclean areas and dynamically build as well as update (build-and-update) the map as in real time. The algorithm begins from an initialization phase and then concurrent map-building and coverage phase. In these algorithms, it is necessary to define a flag, denoted by $f(k,l)$, for a neuron at position (k,l) to indicate its status as cleaned, unknown (unvisited), obstacle, or deadlock

$$f(k,l) = \begin{cases} 0, & \text{if it is unknown} \\ 1, & \text{if it is cleaned} \\ 2, & \text{if it is obstacle} \\ 3, & \text{if it is deadlock.} \end{cases} \tag{10}$$

Initially, the cleaning robot has no *prior* knowledge of any obstacles in the workspace, except the whole workspace size. The model algorithm consists of three phases.

1) *The initialization phase:* The initialization algorithm aims to initialize the starting position of the robot, to set all the neural activities as zeros, etc., which is given in Algorithm 1.
2) *The map-building phase:* This phase mainly relies on the robot's onboard sensors with limited reading distance. The environment in the proposed model is assumed to be completely unknown. The external inputs I_i supplied by the robot's onboard sensors provide the knowledge of both the robot itself and its surroundings. The robot thus has only a limited range knowledge of the workspace. A map of the surrounding environment can be dynamically built in an unknown environment. The robot moves toward the nearest unclean areas and dynamically builds and updates the map as it moves, until all *accessible* unclean regions are cleaned.
3) *The CCN phase:* The unclean areas globally attract the robot in the whole state space through neural activity propagation, while the obstacles have only local effect in a small region to avoid collisions. The coverage algorithm for the CCN is given in Algorithm 2. In this algorithm, for a cleaning robot, once it moves from the current position to its next position, the next position becomes a new current position, and the previous position is marked as cleaned (see Algorithm 2). The definition of "next position" is in the sense that the next position is selected based on the neural activity *and* the turning angle [see (7) and Algorithm 2].

In this section, the following notations are used. Note that N_x and N_y are the discretized size of the Cartesian workspace.

$\mathcal{N}(k,l)$	The set of the discretized workspace, $\{(k,l), 1 \le k \le N_x, 1 \le l \le N_y\}$.
$N(k,l)$	The unknown position (k,l), $1 \le k \le N_x$, $1 \le l \le N_y$.
$\bar{N}(k,l)$	The cleaned Position (k,l), $1 \le k \le N_x$, $1 \le l \le N_y$.
$x(k,l)$	Neural activity at unknown position (k,l).
(k_c,l_c).	The current position (k_c,l_c).
(k_n,l_n)	The next position (k_n,l_n) where activity is maximal.
$x_m(k_n,l_n)$	The maximal neural activity at position (k_n,l_n).
$I(k,l)$	External input to neuron $N(k,l)$.

There are following definitions that facilitate the algorithms addressed in this section. For simplicity, the neuron at position $N(k,l)$ is also called neuron $N(k,l)$.

Definition 1 (Complete Coverage): Given the 2-D Cartesian workspace $\mathcal{W}$ of the cleaning robots, and position $N(k,l)$ of robot represented by a neuron $N(k,l)$ at time t, **complete coverage** at a total time T is in the sense that every nonobstacle neuron is visited and the current location neuron $N(k,l)$ has the following properties:

- the external input to neuron $N(k,l)$ is set to be zero, $I(k,l) := 0$;
- the $f(k,l)$ flag, used to indicate the status of this point, to neuron $N(k,l)$ is set to be one, $f(k,l) := 1$.

Definition 2 (Neighborhood): The **neighborhood** of a central neuron $C(k,l)$ consists of some neighboring neurons that enclose the central neuron $C(k,l)$. The **neighborhood** of a central neuron $C(k,l)$ in a shunting neural network is defined by $\mathcal{N}_r(k,l) = \{N(m,n) | \max\{|m-k|, |n-l|\} \le r, 1 \le m \le N_x, 1 \le n \le N_y\}$, where r is the number of circles enclosing the central neuron (see Fig. 3). The position of a neighboring neuron $N(m,n)$ near to the central neuron $C(k,l)$ has the following property: $m \in \{k-1, k, k+1\}$ and $n \in \{l-1, l, l+1\}$ illustrated in Fig. 3.

In Fig. 3, the central neuron $C(k,l)$ illustrated by dark-shaded square has eight neighboring neurons if $r = 1$; it has 16 neighboring neurons if $r = 2$ (the central neuron is surrounded by two layers of neighboring neurons). The different layers of the neighboring neurons are depicted by different color gradient as illustrated in Fig. 3. In this model, the central neuron locally connects with closest neurons, i.e., $r = 1$.

Definition 3 (Deadlock): Given the 2-D Cartesian workspace of robots with the neural networks, and the position of robot represented by a neuron at position (k,l), a **deadlock** situation of a robot is the fact that the neighboring neuron of the central neuron is either visited, obstacle, or with smaller neural activity. This robot occupying central neuron $N(k_c,l_c)$, associated with its neighboring neurons, causes deadlock circumstances. In this case, the following properties hold: if $\exists(m,n), \forall m \in \{k-1, k, k+1\}$ and $n \in \{l-1, l, l+1\}, m \ne k_c, n \ne l_c, 1 \le m \le N_x, 1 \le n \le N_y$, s.t.

- $f(m,n) = 1$, **or**
- $f(m,n) = 2$, **or**
- $x(m,n) < x(k_c,l_c)$

then $f(m,n) := 3$.

Algorithm 1 Initialization Algorithm

```
1) set $k_c := k_0$; set $l_c := l_0$, where $(k_0,l_0)$ is a starting point
   // Set starting point to a current neuron
2) set $f(k,l) := 0$; set $I(k,l) := 0, \forall 1 \le k \le N_x$,
   $1 \le l \le N_y$
   // Set all areas as unclean
3) set $x(k,l) := 0, \forall 1 \le k \le N_x, 1 \le l \le N_y$
   // Set all neural activities as zero
```

Algorithm 2 Coverage Algorithm

```
1) Compute neural activity by (5)
2) $\mathcal{N}_r(k,l) = \{N(m,n) | m \in (k-1, k, k+1)$ and
   $n \in (l-1, l, l+1)\}$
```

```
    // scan unknown neighboring neurons $(k_n, l_n) = \arg\max_{m,n}(x(m,n) + cy(m,n)) \in \{\mathcal{N}_r | m \in (k-1, k, k+1)$ and $n \in (l-1, l, l+1)\}$
    // find the next neighboring neuron with the maximal neural activity
3)  set $k_c := k_n$; set $l_c := l_n$
    // Set current neuron to neighboring neuron
4)  if $\exists (k,l) \in \mathcal{N}_r(m,n)$, s.t. $x(k,l) \leq x(k_c, l_c)$
    // if neighboring neural activity $\leq$ current neural activity then
      • set $I(k,l) := 0$
      • set $\bar{N}(k,l) := N(k,l)$
      • flag $f(k,l) := 1$
        // Mark it as cleaned and external input as zero
    end if
5)  if $\exists (g,h) \in \mathcal{N}_r(m,n), \forall g \in \{k-1, k, k+1\}$ and $h \in \{l-1, l, l+1\}$, s.t. $f(g,h) = 1$
    // if neighboring neurons are all visited then
      • set $I(g,h) := 0$
      • set $\bar{N}(g,h) := N(k,l)$
        // Mark them as cleaned and external input as zero
    end if
6)  go to 1).
```

The computational complexity depends linearly on the state–space size of the neural network, which is proportional to the workspace size. The number of neurons required is equal to $M = N_x \times N_y$. If the workspace is an $N \times N$ square in shape, there are N^2 neurons and each neuron has at most eight local neural connections. Therefore, the total neural connections are $8 \times N^2$. The computational complexity of the proposed algorithm is $O(N^2)$, i.e., the complexity is squarely proportional to the degree of discretization (resolution). Workspace size and rules to be used will principally affect the computational complexity.

In complete coverage issues arisen in the real world, the total workspace that the robot works with is known but the detailed environmental knowledge inside the workspace is unknown. Even in the case without knowledge of overall workspace size, N_x and N_y are not essential parameters needed for the proposed algorithm. Therefore, usually, like all the grid-based methods, N_x and N_y are empirically determined based on the workspace size. The workspace is discretized with dimension of width N_x and height N_y.

V. Simulation Studies

The proposed neural network approach is capable of planning a complete coverage path for cleaning robots autonomously without any human intervention. In this section, this model with map building of *square* cell representation is first applied to a known indoor room environment, then to a completely unknown environment case. This model is also verified in a dynamically changing environment with moving obstacles. The simulations are implemented in C++. The bound wall of the workspace is assumed to be known in the simulation.

A. CCN in a Known Indoor Room Environment

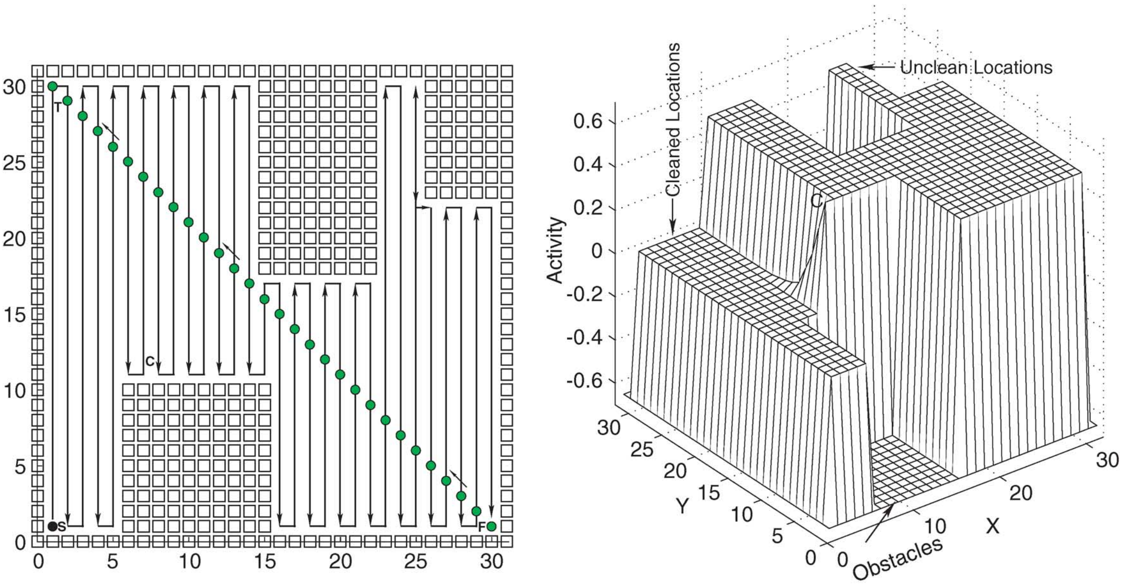
The proposed model is first applied to a known indoor room environment case, where there were some obstacles in the known workspace. The workspace is shown in Fig. 4(a), where S $(1, 1)$ indicates the starting point and the squares represent the obstacles. The neural network includes 32×32 discretely and topologically organized neurons, where all the neural activities are initialized to zero. The model parameters are set as: $A = 50$, $B = 1$, and $D = 1$ for the shunting equation; $\mu = 0.7$ and $r_0 = 2$ for the lateral connections; and $E = 100$ for the external inputs. There are three set of obstacles with different size in the workspace. The mobile robot starts from S $(1, 1)$. Initially, the external inputs I_i of neurons for representing obstacles are set as $-E$, while those of other neurons are E. After the robot covers a point, the external input I_i is reset to zero, standing for a cleaned point described in Algorithm 2. The generated robot path is shown in Fig. 4(a), where the robot can autonomously travel along an ordered zigzag path, from bottom side to top side, then from top side to bottom side, to cover and clean the whole workspace with obstacle avoidance. When the robot reaches position F $(30, 1)$ that is a deadlock point, the robot finishes the cleaning work to the left of point F. Once the robot arrives at the end of the deadlock, even though all of the areas to the left of point F are signed as cleaned or covered, the proposed neural network model and coverage algorithms are readily capable of obtaining the solution. The robot is able to escape the deadlock (according to Definition 3, $f(30, 1) = 3$) by using a dynamic neural neighborhood analysis method and deadlock detection algorithm [25], move out from deadlock, and move to designated point T $(1, 30)$. The robot stuck in the deadlock will wait until neural activity propagates to its position and then it can get out of the deadlock and continue to move to its target.

The simulation result is shown in Fig. 4(a), where the robot is illustrated using a solid circle and a short and smooth point-to-point path to achieve the exit at T $(1, 30)$ using light circles. The dynamic activity landscape of the neural network when the robot arrives at position C $(8, 12)$ after starting from S $(1, 1)$ is shown in Fig. 4(b). The identical position C $(8, 12)$ is intentionally chosen to compare with an unknown environment case in the following section. The neural activities of the unclean areas have very large value represented by peak, and the neural activities of obstacles are represented using valley because they have negative value.

B. Map Building and CCN in an Unknown Indoor Room Environment

For comparison, this model is then applied to a completely unknown indoor room environment. In this case, the entire environment is assumed to be completely *unknown* except that the entire workspace is initialized as unclean areas. The robot can only sense a limited range with a radius of $R_s = 6$ by its onboard robot sensors. The neural network has 32×32 topologically organized neurons with zero initial neural activities and the model holds the same parameters as in the case above for comparison purpose.

When the robot reaches C $(8, 12)$ that is the same point as in the case above, the robot is only capable of sensing a limited range in a circle with radius of $R_s = 6$ [see Fig. 5(a)]. Within the circle, there are some cleaned, unclean, and sensed obstacle points shown as Fig. 5(a). By means of the sensory information,



(a)

(b)

Fig. 4. CCN in a completely known environment. (a) The generated robot path. (b) The neural activity landscape when the robot reaches $C\ (8,12)$.

the robot can smoothly plan a zigzag path to cover every point with obstacle avoidance shown in Fig. 5(b). The dynamic activity landscape of the neural network when the robot reaches position $C\ (8,12)$ during CCN is illustrated in Fig. 5(c). Compared with the case above in a known environment shown in Fig. 4(b), when the robot arrives at $C\ (8,12)$, the robot does not yet know the remaining environmental information shown in Fig. 5(c), whereas the robot does know the rest of the environment shown in Fig. 4(b). A map is simultaneously built during the navigation of the cleaning robot. The robot has already built a global map of the entire environment once it completely explores the whole workspace when reaching $F\ (30,1)$. Therefore, in the point-to-point path planning from $F\ (30,1)$ to $T\ (1,30)$ in Fig. 5(b), the robot can avoid this wall-like obstacle to reach the exit point $T\ (1,30)$. It is assumed that the robot can recognize a closed obstacle. In Fig. 5(b), for instance, the workspace contains four sets of closed obstacles and the robot cannot reach the internal obstacles. Once the robot explores the contour of a closed obstacle whose internal areas are found unreachable, the whole set of obstacles is regarded as a closed obstacle. The dynamic activity landscape of the neural network when the robot approaches point $E\ (30,5)$ is illustrated in Fig. 5(d), where four valley areas are found in this figure, which represent the neural activities of the obstacles.

C. Map Building and CCN in a Dynamically Changing Environment

The proposed neural network approach is capable of building map and navigating for a cleaning robot in a dynamically changing environment. In this section, the simulation is performed to verify that the robot can explore the unknown and dynamically changing environment when it encounters moving obstacles. The neural network adopts the same model parameters as above. The sensing radius of the robot is $R_s = 6$. The robot starts to sweep from position $S\ (1,1)$ shown as Fig. 6(a) by planning a zigzag path. The robot can autonomously avoid two sets of static obstacles.

In this case study, a set of L-shaped obstacles will suddenly appear in front of the robot to test whether the robot can avoid the obstacles on its movement. When the robot reaches position $A\ (22,10)$, the moving obstacles are not yet presented. The robot can plan a path from top side $(22,30)$ to bottom side $(22,1)$, since the obstacles illustrated by light squares have not appeared. The activity landscape of the neural network then is illustrated in Fig. 6(b).

The robot can build up an environmental map and navigate in this unknown environment while it explores the workspace with static and moving obstacles. Once the robot moves to position $C\ (23,17)$, the L-shaped obstacles is suddenly present in front of the robot [see Fig. 6(a)]. The activity landscape of the neural network right after the obstacles are placed is shown in Fig. 6(c). It shows that the neural activities at the positions with sudden obstacles immediately become very large negative values. The robot cannot move forward any longer due to the suddenly appeared obstacles. The robot has to move back through several cleaned areas, pass around the obstacles, reach the nearest unclean area on its right-hand side, and finally clean all the rest of the unclean workspace with obstacle avoidance. The dynamic activity landscape of the neural network when the robot arrives at position $E\ (28,18)$ is shown in Fig. 6(d). Finally, the robot can build up a map of the whole workspace when it reaches the position $F\ (30,1)$.

VI. Comparison Studies

In this section, the proposed map-building and CCN model with coverage algorithm is compared with the model in [28] by simulations implemented in C++. The workspace is decomposed into rectangular cells with the height of h and the width of l (see Fig. 2). Also, this model is compared with Gabriely and Rimon's approach [72] using square cells.

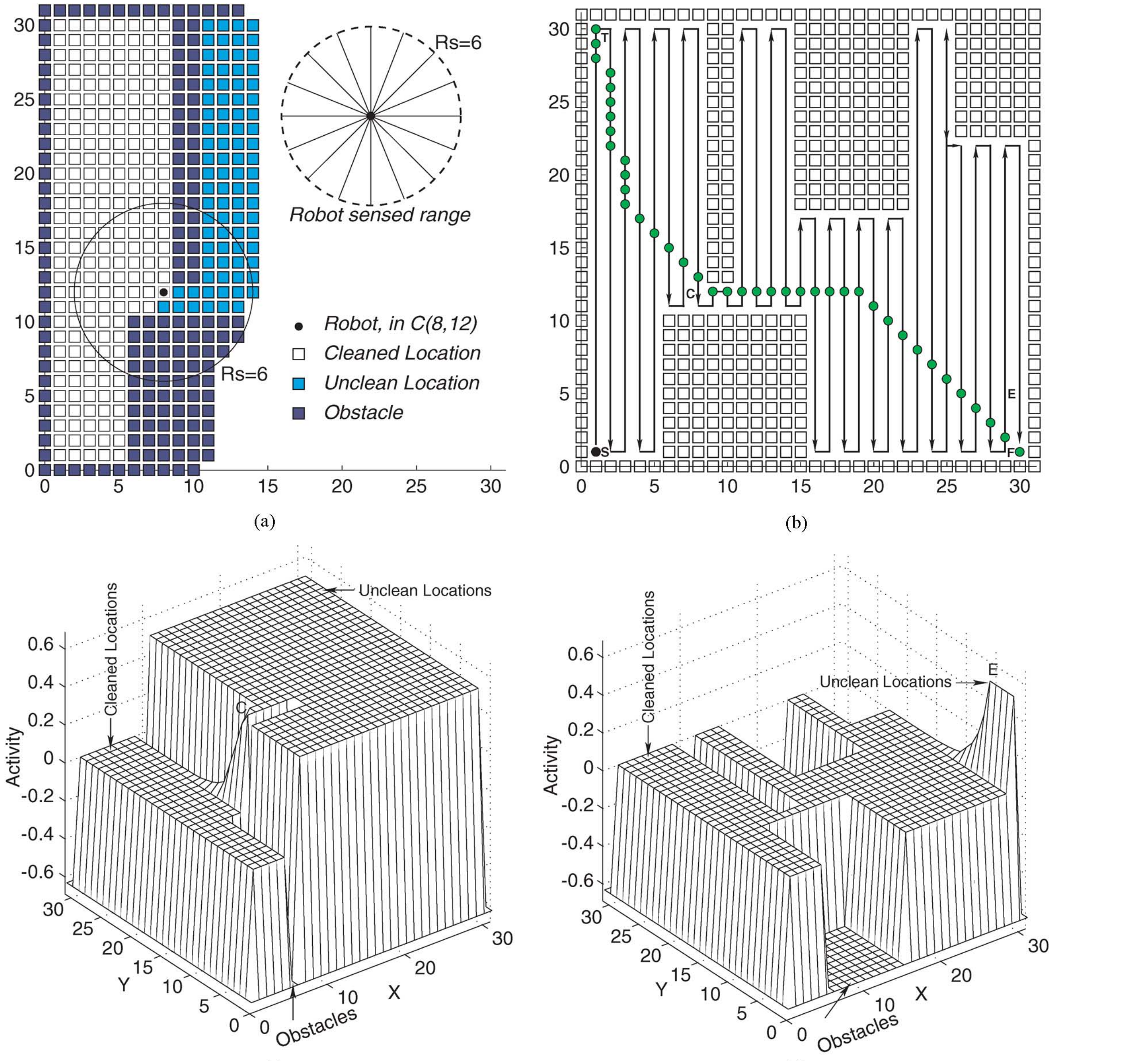


Fig. 5. CCN and map building in a completely unknown environment. (a) The built map when the robot reaches $C\ (8,12)$ and the sensed range of the environment by the onboard sensors. (b) The planned path. (c) The neural activity landscape when the robot reaches $C\ (8,12)$. (d) The neural activity landscape when the robot approaches $E\ (30,5)$.

A. Map Building and CCN in an Unknown Environment

According to [28, Fig. 7], if a robot moves from the starting point S and covers the triangular grids, there is no overlap in A and B regions [see Fig. 7(a)]. However, there are overlapping areas in C region while the robot moves in C region in a back-and-forth fashion, which is not reasonable, obviously [see Fig. 7(a)]. In order to accurately and fairly compare with their model, the workspace is slightly modified by extending the right boundary shown as Fig. 7(b). There is no overlapping area once modifying the workspace slightly that will not affect the comparison itself. In other words, it is sensible to compare the proposed model with Oh *et al.*'s model [28] using the same workspace, which is more reasonable and comparable.

The proposed model now is applied to this case, where there were some obstacles in the unknown workspace. The neural network consists of 13×10 discretely and topologically organized neurons, where all the neural activities are initialized to be zeros. The model parameters are set as: $A = 80$, $B = 1$, and $D = 1$ for the shunting equation; $\mu = 0.7$ and $r_0 = 2$ for the lateral connections; and $E = 100$ for the external inputs. There is one set of obstacles constituted by eight rectangles in the workspace. The mobile robot starts from $S\ (1,1)$. Initially, the external inputs I_i of neurons representing obstacles are set as $-E$, while those of other neurons are E. After the robot covers a point, the external input I_i is reset to zero, to stand for a cleaned point. The generated robot path is shown in Fig. 8(a), where the robot can autonomously travel along an ordered zigzag path, from bottom side to top side, then from top side to bottom side, to cover and clean the whole workspace with obstacle avoidance. When the robot reaches the point $T\ (11,8)$, the robot reaches the end of the

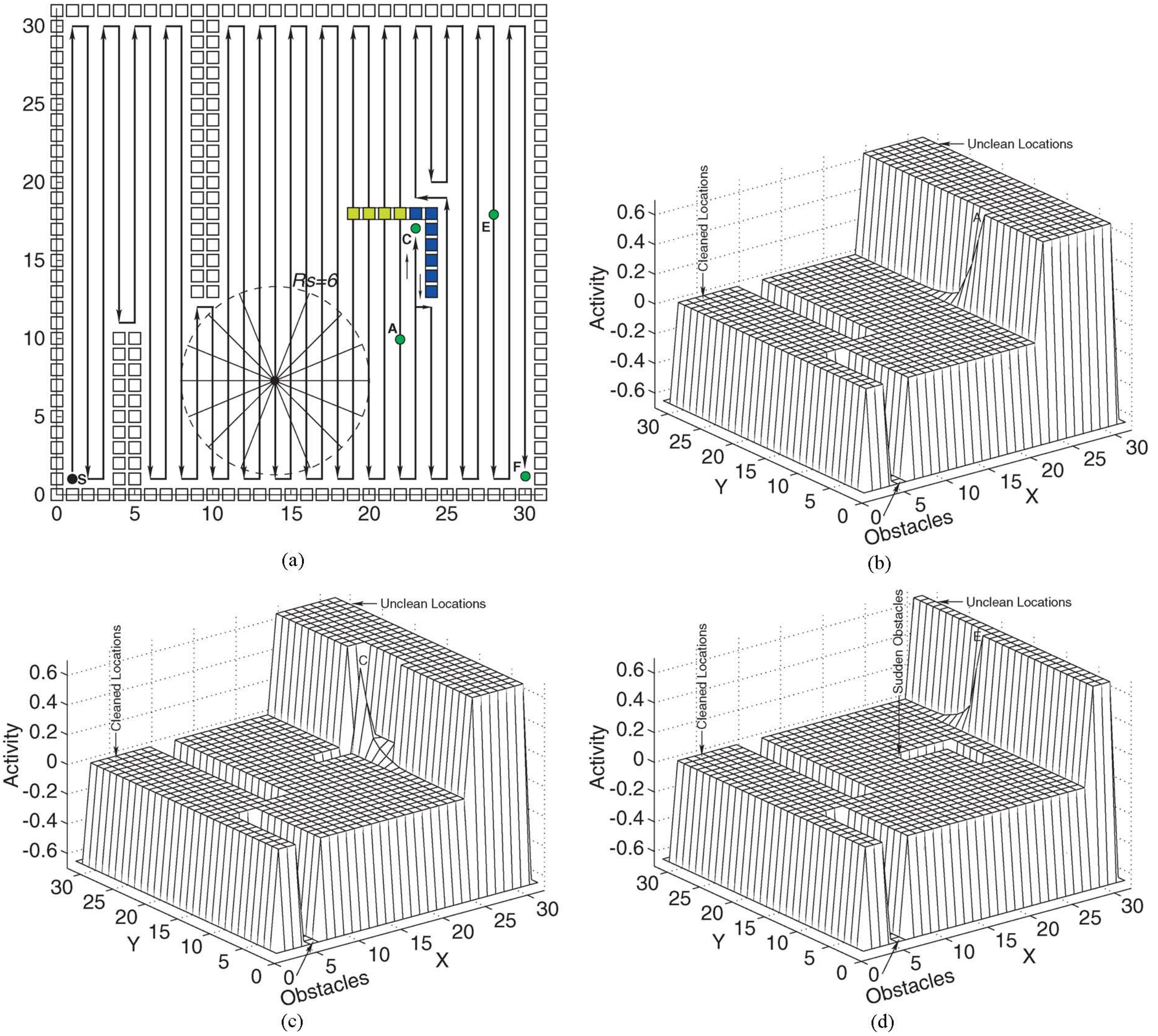


Fig. 6. CCN and map building in a completely unknown and dynamically changing environment. (a) The built map and planned path. (b) The neural activity landscape when the robot reaches $A\ (22, 10)$. (c) The neural activity landscape when the robot reaches $C\ (23, 17)$. (d) The neural activity landscape when the robot approaches $E\ (28, 18)$.

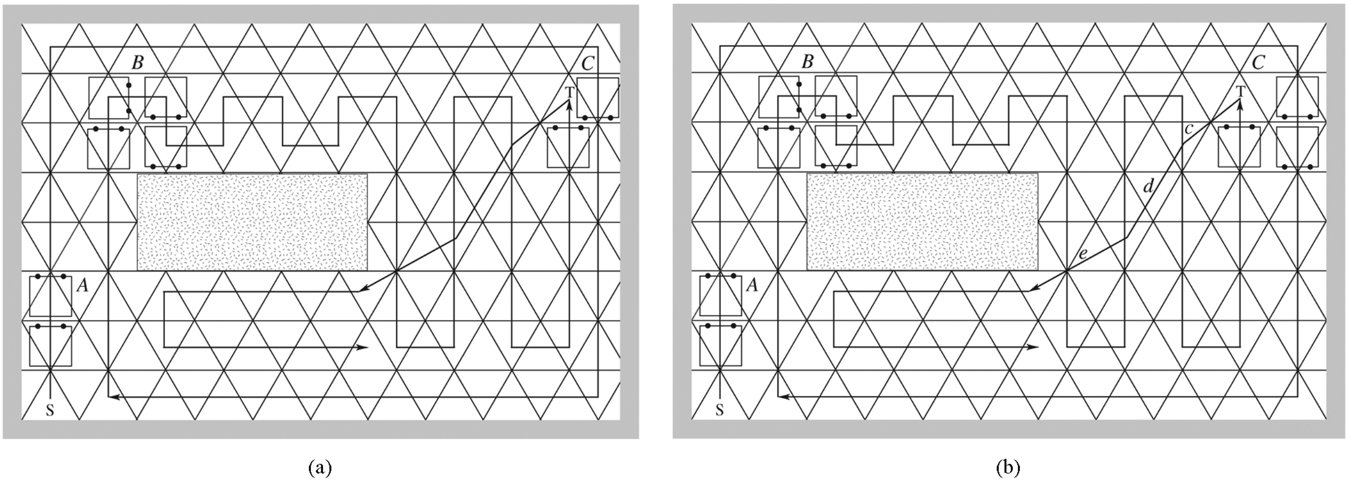
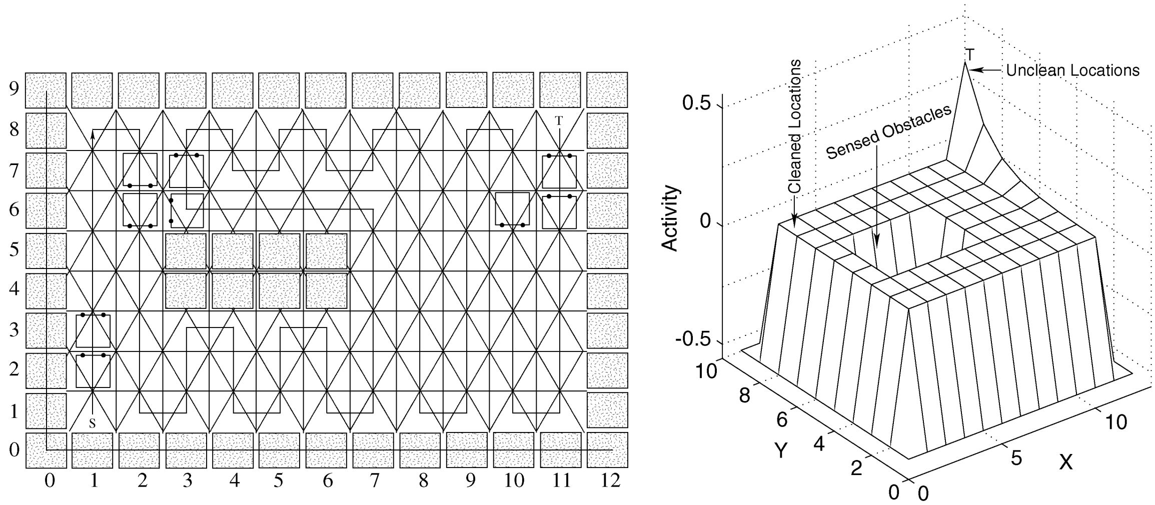


Fig. 7. Example of CCN robot path in a completely unknown environment. (a) The original workspace [28]. (b) A more reasonable workspace.



(a)

(b)

Fig. 8. Completely unknown environment of CCN. (a) The generated robot path. (b) The neural activity landscape when the robot reaches T $(11, 8)$.

workspace. The corresponding activity landscape of the neural network is illustrated in Fig. 8(b).

In Fig. 8(a), the length the robot lans using the proposed model is calculated by

$$61h + 18l. \tag{11}$$

Recalling (1), $h = (\sqrt{3/2})l = 0.866l$, the length the robot plans by the proposed model is $70.83l$. In Fig. 7(b) (modified one), the robot length planned by Oh *et al.*'s model is given by

$$a + b + c + d + e \tag{12}$$

where a is the length of the whole vertical path in Fig. 7(b); b is the length of the horizontal path in total; and c, d, and e are the line segments labeled in Fig. 7(b)

$$a = 45h \tag{13}$$
$$b = 34l \tag{14}$$
$$c = \sqrt{h^2 + l^2} \tag{15}$$
$$d = \sqrt{(2h)^2 + l^2} \tag{16}$$
$$e = \sqrt{h^2 + (1.5l)^2}. \tag{17}$$

Consequently, the length of the robot path generated by Oh *et al.*'s model is $78.02l$, while the proposed model produces a shorter robot path at $70.83l$.

The complete coverage trajectory length and robot's body turning number using the proposed model and Oh *et al.*'s model were calculated (Table I). It shows that the trajectory length by our proposed model is shorter than their model. However, our turning number is only four turns more than Oh *et al.*'s model. Additionally, there is no overlap by the proposed model, but there are five overlapped cells when robot covers the paths by their model shown in Table I. In this case, the length of the trajectory by the proposed model is 10.15% shorter than theirs.

TABLE I
COMPARISON OF TRAJECTORY LENGTH AND NUMBERS OF TURNING AND OVERLAP USING OH *ET. AL.*'S MODEL AND THE PROPOSED MODEL IN THE UNKNOWN ENVIRONMENT SHOWN IN FIGS. 7 AND 8

Model	Length	Turns	Overlap
Oh *et al*'s model	78.02l	26	5
Proposed model	70.83l	30	**0**

B. Map Building and CCN in a Template Case

The proposed model is applied to a template case in an *unknown* environment. In template-based approaches such as [19] and [28], several templates are predefined and their solutions are predeveloped. The cleaning robot has to match various situations to those limited templates, but may not be able to find a suitable solution when the actual situation does not highly match any predefined templates. In Oh *et al.*'s model, the template information including three direction cells (front, right, and left) and the direction at the present position of a robot need to be predefined. In the proposed neural network model, *no* template is needed. The proposed model is applied to a case called back trace (BT) template in [28]. The BT template is shown in Fig. 9(a). The robot can only sense a limited range with a radius of $R_s = 2$, for instance, by its onboard robot sensors. In the simulation, the neural network contains 10×10 topologically and discretely organized neurons. The model parameters are set as: $A = 80$, $B = 1$, and $D = 1$ for the shunting equation; $\mu = 0.7$ and $r_0 = 2$ for the lateral connections; and $E = 100$ for the external inputs.

In Fig. 9(a), the robot starts to move from S $(1, 1)$. When the robot attains the point A $(1, 7)$, it is closely approaching to the deadlock point B $(1, 8)$. The built map when the robot reaches A $(1, 7)$ is illustrated in Fig. 9(b), where there are several cleaned, unclean, and sensed obstacle points in a circle with radius of $R_s = 2$, restrainedly sensed by the robot. The neural activity landscape of the neural networks when the robot reaches A $(1, 7)$ is illustrated in Fig. 10(a).

Once the robot reaches B $(1, 8)$ point, this is usually called a *deadlock* situation (according to Definition 3, $f(1, 8) = 3$),

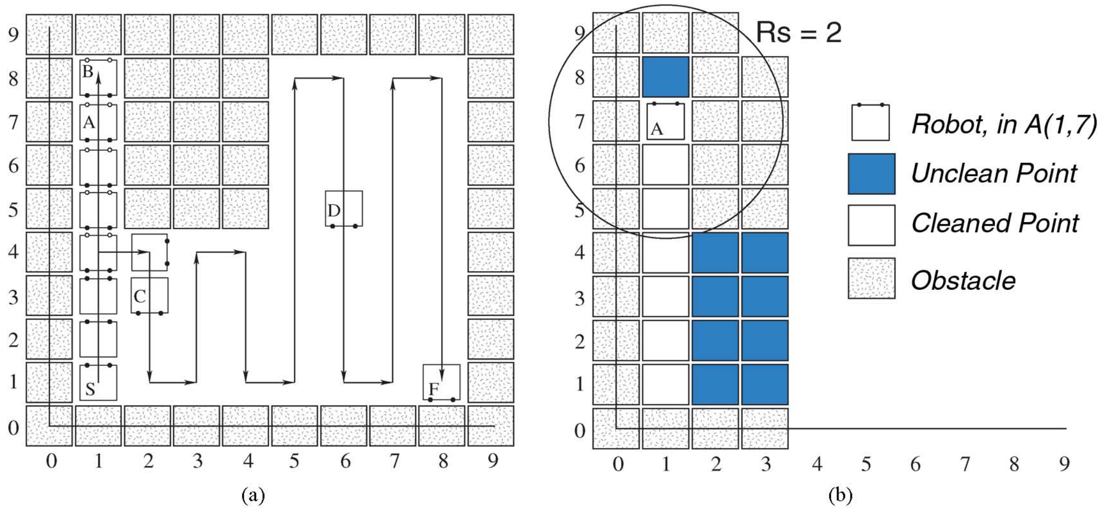


Fig. 9. BT template case in an unknown environment. (a) The deadlock situation and generated entire path. (b) The built map when the robot reaches A $(1,7)$ and the sensed range of the environment by the onboard sensors.

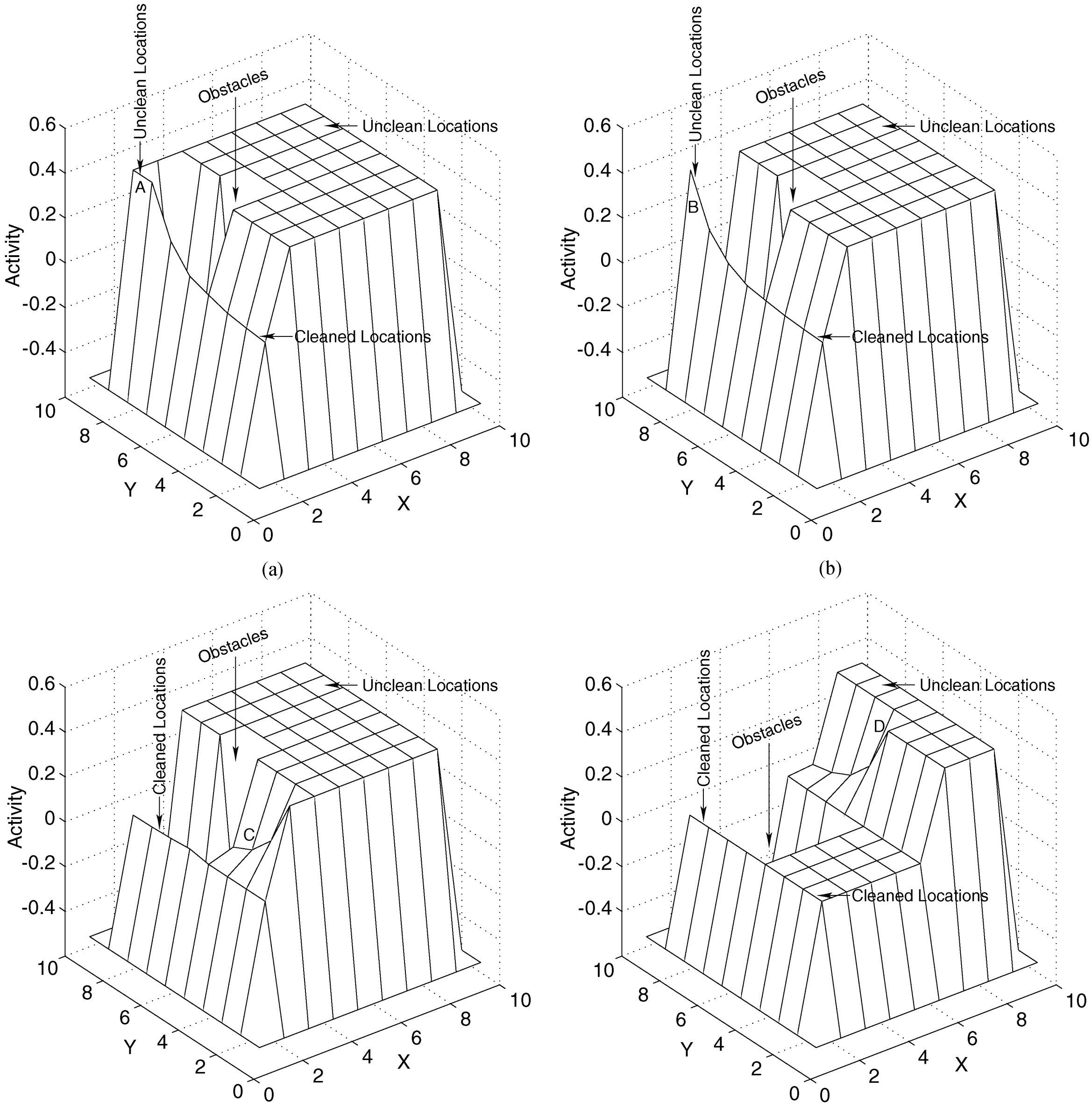
where all the neighboring areas of B are either cleaned areas or obstacles: at point B, the behind areas are cleaned or obstacles, and the front and right areas are obstacles. Thus the neural activities at its neighboring areas are *not* larger than the neural activity at B [see Fig. 10(b)]. To resolve such a deadlock problem $(f(1,8)=3)$, the robot should be able to move back passing some *cleaned* areas in order to reach the unclean areas on the right-hand side. This model will not be trapped in deadlock situations, since the rest of unclean areas can globally attract the robot in the whole workspace through neural activity propagation. When the next position is not immediately available, e.g., in a deadlock situation, the robot has to wait until the next position toward the targets is available through dynamic neural activity propagation [25]. When the robot arrives in a deadlock situation, all the neural activities of its neighboring positions are not larger than the activity at the current position, because its neighboring positions receive either negative external input (obstacles) or no external input (cleaned positions), and all the cleaned neighboring positions passed a longer decay time as they were cleaned earlier than the current position [see Fig. 10(b)]. The neural activity at the deadlock position B will quickly decay to zero due to the passive decay term $-Ax_i$ in shunting (5). Furthermore, due to the neural activity propagation among neurons, the activities from the unclean areas are propagating to all directions. This can propagate toward the current robot position B. The unclean areas have very large external inputs and keep the largest neural activity. Therefore, over a very short time, the activity at point B will become smaller than that at the position right below B, and the robot will start to move down toward the unclean positions. The robot continues moving toward the unclean position that has the largest neural activity. After the robot arrives at point C, the neural activity is shown in Fig. 10(c), where neural activity at C starts to decrease. Thus, in the proposed model, because of the neural activity propagation among neurons, the unclean areas are able to attract the robot, which is similar as the classical point-to-point path planning [25]. In Fig. 10(c), it is shown that the unclean positions have the largest neural activity due to their very large external excitatory inputs and the obstacle positions have the smallest neural activities because of their very large inhibitory inputs.

All the neural activities are initialized to zero. By means of the sensory knowledge, the robot can smoothly plan a zigzag path to cover every point with obstacle avoidance shown in Fig. 9(a). The dynamic activity landscape of the neural network when the robot arrives at position D $(6,5)$ is shown in Fig. 10(d). The neural activities of the unclean areas have very large value represented by peak, and the neural activities of obstacles are represented by valley because they have negative value.

C. Map Building and CCN in an Unknown Indoor Room Environment

Recall that Oh *et al.* proposed a triangular cell decomposition approach for unknown environments. However, the main drawback of their method is that the combination of three techniques, distance-transform-based navigation, template-based approach, and wall-following navigation algorithm, causes longer paths. The property of the template-based model increases many inevitable path repetitions. Therefore, in their simulation, the robot traverses a path with a greater length than ours to cover a given workspace.

In this paper, Fig. 11(a) is redrawn from the simulation result of Oh *et al.* (originally, [28, Fig. 12]). Because of their algorithms and template model, the robot must follow A, B, and C paths moving back to the previous areas for coverage task that will result in the overlaps [see Fig. 11(a)] and increase the length of paths. The proposed approach is applied to simulate the same case for comparison. The neural network consists of 37×29 topologically organized neurons, where all the neural activities are initialized to zero. The model parameters are exactly the same as the last case study. The simulated robot path by the proposed model is shown in Fig. 11(b) by solid lines with arrows representing the moving directions. It shows that the cleaning robot is able to autonomously sweep the whole workspace from left to right with obstacle avoidance with starting point of S $(1,1)$ on the left and final point of F $(35,27)$



(a)

(b)

(c)

(d)

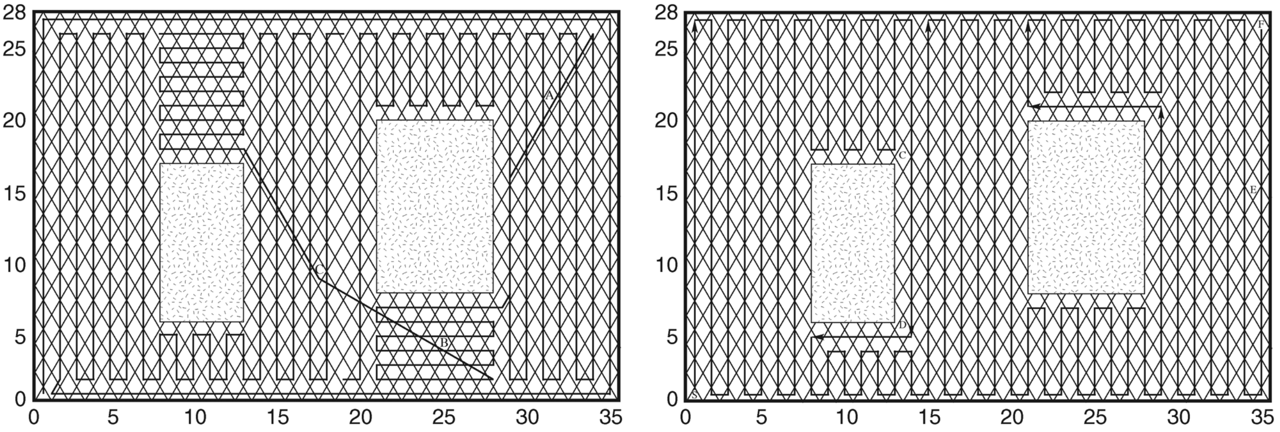
Fig. 10. Neural activity landscape of the neural networks for the BT deadlock case when the robot reaches (a) position A $(1, 7)$; (b) position B $(1, 8)$; (c) position C $(2, 3)$; and (d) position D $(6, 5)$.

on the right in the workspace. The resulting paths are more reasonable, optimal, and complete than Oh *et al.*'s model.

The complete coverage trajectory length and robot's body turning number for the environment shown in Fig. 11 are calculated by (1). The results using the proposed model and Oh *et al.*'s model are shown in Table II, which indicates that our trajectory length and turning number are competitive with their model. In this case, the length of the trajectory by the proposed model is 7.08% shorter than that of Oh *et al.*'s model. Furthermore, there is no overlap in the generated paths by the proposed model, but there are 39 overlapped cells when the robot covers the paths by their model. Additionally, the turning number of the proposed model is two turns less than their model.

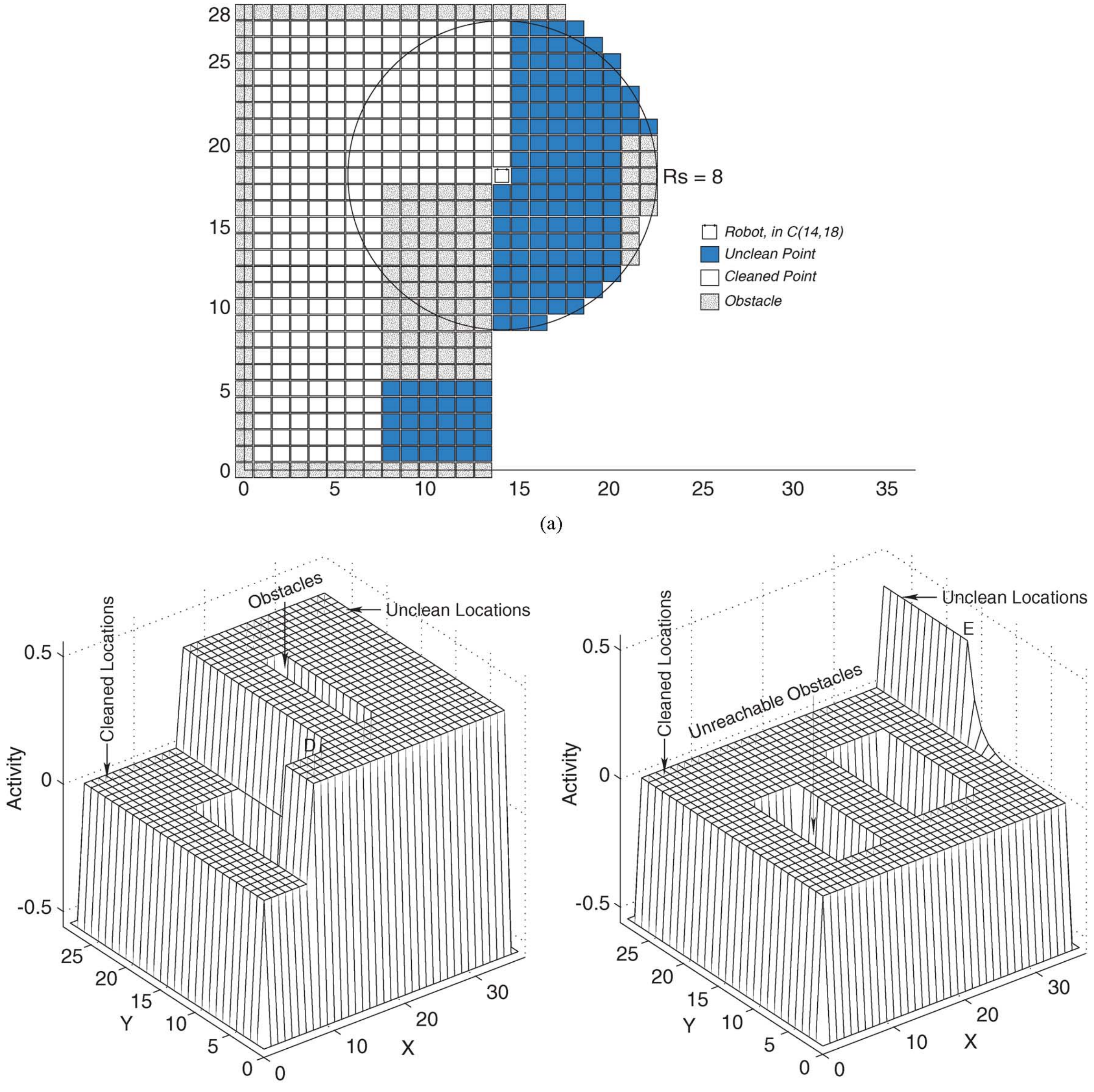
When the robot reaches C $(14, 18)$, the robot is only capable of sensing a limited range in a circle with radius of $R_s = 8$. Within the circle, there are some cleaned, unclean, and sensed obstacle areas shown as Fig. 12(a) that is a built map when the robot reaches C $(14, 18)$.

The dynamic activity landscape of the neural network when the robot arrives at position D $(14, 6)$ is shown in Fig. 12(b). The neural activities of the unclean areas have very large value illustrated by peak and the neural activities of obstacles are depicted by valley. Based on the assumption in this algorithm that the robot can recognize a closed and unreachable obstacle by exploring its contour, the rectangular obstacle with vertices of $(8, 6), (8, 17), (13, 17), (13, 6)$, and $(8, 6)$ is regarded as a closed and unreachable obstacle by the cleaning robot. Consequently, the neural activities of the rectangular obstacle sensed by the robot are staying at the valley of the activity landscape of the neural network in Fig. 12(b). The dynamic activity landscape of the neural network when the robot arrives at position



(a) (b)

Fig. 11. CCN case in a completely unknown large environment. (a) The generated CNN path using Oh *et al.*'s model [28]. (b) The generated CNN path using the proposed model.



(a)

(b) (c)

Fig. 12. CCN case in a completely unknown large environment. (a) The built map when the robot reaches $C\ (14,18)$ and the sensed range of the environment by the onboard sensors. (b) The neural activity landscape when the robot reaches $D\ (14,6)$. (c) The neural activity landscape when the robot reaches $E\ (35,15)$.

$E\ (35,15)$ is shown in Fig. 12(c). Similarly, the neural activities of the two rectangular obstacles sensed by the robot are represented by the valley of the activity landscape of the neural network when the robot reaches position $E\ (35,15)$.

TABLE II
COMPARISON OF ROBOT PATH LENGTH AND NUMBERS OF TURNING AND OVERLAP USING OH *ET AL*'S MODEL AND THE PROPOSED MODEL IN A LARGE UNKNOWN ENVIRONMENT SHOWN IN FIG. 11

Model	Length	Turns	Overlap
Oh *et al*'s model	721.04l	102	39
Proposed model	673.40l	100	**0**

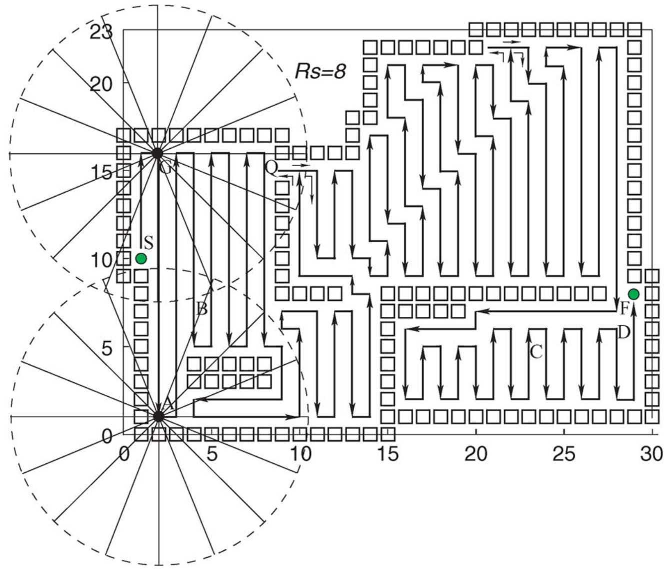


Fig. 13. Built map and planned path in an unstructured, completely unknown environment.

D. Map Building and CCN in an Unstructured Case

Gabriely and Rimon [72] proposed a spanning tree approach to cover environments. A spanning tree coverage approach decomposes the workspace into discrete cells and constructs spanning tree of graph induced by the cells. The robot is capable of covering every point, and traveling a reasonable path in a grid-like representation in the workspace to achieve complete area coverage.

In this section, in order to demonstrate that the proposed model can also be applied to unstructured environments, it is compared to Gabriely and Rimon's model [72] using the same workspace shown in Fig. 13.

The neural network comprises 31 × 24 topologically organized neurons, where all the neural activities are initialized to zero. The bound wall of the unstructured environment and its outside region are all assumed to be known as obstacles (see Fig. 13). The model parameters are exactly the same as the above case study. The built map and simulated robot path are shown in Fig. 13 by solid lines with arrows representing the moving directions. The robot can only sense a limited range with a radius of $R_s = 8$ by its onboard robot sensors. The robot whose paths are represented by solid lines starts to move from the lower left corner $S\ (1, 1)$. It travels along smooth zigzag paths by performing back-and-forth motions. The robot eventually attains the target point $F\ (29, 8)$ thus the complete coverage of the unknown environment is achieved.

When the robot passes the point $G\ (2, 16)$, it can sense the obstacle of point $Q\ (10, 16)$ by means of its onboard robot sensors. As a result, its activity landscape of the neural network when it reaches the point $A\ (2, 1)$ is illustrated in Fig. 14(a). The robot can autonomously travel along an ordered zigzag path, from bottom side to top side, then from top side to bottom side, to cover and clean the whole workspace with obstacle avoidance. The corresponding activity landscapes of the neural network when it reaches the point $B\ (4, 7)$ and $C\ (23, 5)$ are illustrated in Fig. 14(b) and (c), respectively. Finally, before the robot fulfills the entire coverage mission, it arrives at the point $D\ (26, 9)$, in which its activity landscape of the neural network is illustrated in Fig. 14(d). In fact, the whole procedure that the robot cleans the unstructured workspace can be illustrated by the neural activity landscape of the neural network in Fig. 14.

In order to quantitatively compare the length using the proposed model and Gabriely and Rimon's model, a room-like environment in the lower right portion in Fig. 13 is sampled as Fig. 15. The workspace is decomposed into grids in the proposed model and Gabriely and Rimon's model. Comparing the grid below S' points in Fig. 15(a) and (b), however, Gabriely and Rimon break a grid down into four subgrids. Fig. 15(a) shows their room-like environment. In other word, one grid is further decomposed into four subgrids that is illustrated by the grid in the entrance S' points [see Fig. 15(a)]. A spanning-tree graph illustrated by bold lines is constructed in Fig. 15(a). The robot moves into the room-like workspace by this entrance S' point. In Gabriely and Rimon's algorithm, the robot has to spiral inward to position A' and then spiral outward from A' along the edges of the spanning tree and finally it attains at F'. The generated paths are illustrated in Fig. 15(a). The planned paths by the proposed model are shown in Fig. 15(b).

The comparison to Gabriely and Rimon's model with regard to trajectory length, numbers of turning, and overlap is reported in Table III. The proposed model can plan two times shorter paths than that of Gabriely and Rimon's model. This is because their spanning tree model drives the robot to move in the workspace and move out from the workspace following the constructed spanning tree to perform the complete coverage. The turning numbers of the proposed model is 12.5% less than that of Gabriely and Rimon's model for coverage task. Both models have no overlapping paths. Therefore, besides indoor room, dynamically changing, template case environments described previously, the proposed model is also suitable for unstructured environments. The performance in various unknown environments by the proposed model is competitive to that of Gabriely and Rimon's model.

VII. MODEL PARAMETER SENSITIVITY

The sensitivity of a system to parameter variations is a factor of principal importance to be considered when proposing or evaluating a model. An acceptable model should not be very sensitive to variations in its parameter value. There are few parameters in the proposed model: parameters A, B, and D for the shunting equation; parameters μ and r_0 for the lateral connections; and parameter E for the external inputs. Therefore, it

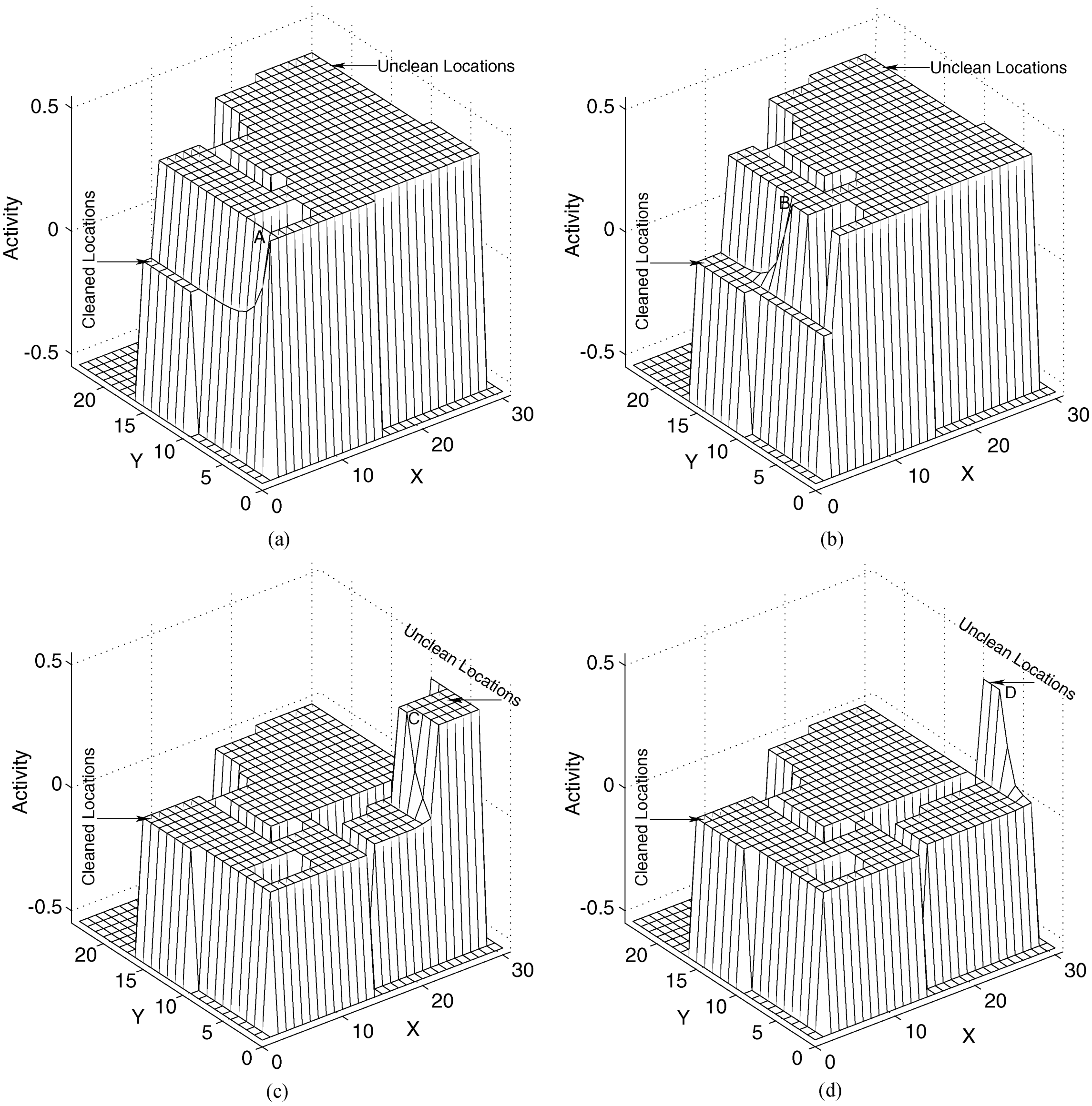


Fig. 14. Neural activity landscape of the neural networks for the unstructured case when the robot reaches (a) position A $(2,1)$; (b) position B $(4,7)$; (c) position C $(23,5)$; (d) position D $(29,6)$.

is necessary to study the parameter sensitivity for the proposed model applied to the concurrent map building and CCN.

In the shunting (4) or (5), parameters B and D are the upper and lower bounds of the neural activity, respectively. The transient response to an input signal does not depend on B and D. At steady state, the neural activity depends linearly on B and D. Because in the proposed model only the relative value of the neural activity is concerned, B and D are not important factors. Thus, B and D can be chosen as $B = 1$ and $D = 1$ for all cases.

Parameter A in (4) or (5) represents the passive decay rate, which determines the transient response to an input signal. Additionally, the steady-state neural activity nonlinearly results from the value of A. Thus, A plays the most important role in the model dynamics, which is essential when the targets (unclean areas) and obstacles are varying. To illustrate the influence of A value to the neural network, two simulations are performed for comparison with map building and CCN in a unknown environment situation, where the simulation result in Fig. (8b) is used for reference. In that case, the model parameters are set as: $A = 80$, $B = 1$, and $D = 1$ for the shunting equation; $\mu = 0.7$ and $r_0 = 2$ for the lateral connections; and $E = 100$ for the external inputs. Now, A value is investigated, and therefore it is first changed to a smaller value, $A = 20$, while other parameters are retained.

When the robot reaches point T, the same point as in Fig. 8(b), the neural activity landscape is illustrated in Fig. 16(a). It shows that the neural activity reduces to zero more slowly and the curve of activity landscape has steeper change than that in Fig. 8(b). Compared to the case of $A = 80$, a larger A value leads to a larger passive decay rate and thus a faster passive decaying of neural activity. The value A plays a principal role in neural activity propagation.

If A value is changed to a much smaller value $A = 5$, while other parameters are retained, the neural activity landscape is

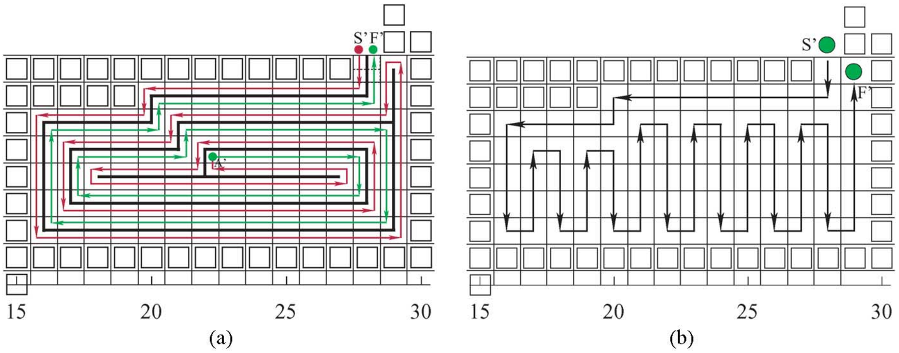


Fig. 15. Comparison of the proposed model with the spanning-tree-based coverage algorithm. (a) The planned paths by Gabriely and Rimon's model (redrawn from [72]). (b) The generated paths by the proposed model.

TABLE III
COMPARISON OF TRAJECTORY LENGTH AND NUMBER OF TURNING USING GABRIELY AND RIMON'S MODEL AND THE PROPOSED MODEL IN THE UNSTRUCTURED UNKNOWN ENVIRONMENT SHOWN IN FIG. 15

Model	Length	Turns
Gabriely and Rimon's model	165	36
Proposed model	82	30

shown in Fig. 16(b). This simulation is carried out when the robot arrives at point T, the same as the case in Figs. 8(b) and 16(a). The simulation demonstrates that the neural activity becomes zero much more slowly and the curve of activity landscape has much steeper change than those in Figs. 8(b) and 16(a). A much smaller A value causes a smaller passive decay rate and thus a much slower passive decaying of neural activity. From shunting (4) or (5), it can be found that a small passive decay rate relatively increases the lateral excitatory connections. Thus, the neural activity will swiftly increase. Those will cause a quick saturation of the neural activity. Consequently, if the A value is too small, the neural activity propagation will be affected because of the quick saturation of the neural activity.

Large enough A value ensures swift enough passive decaying of the neural activity and proper propagation of the neural activity. In other words, too small A value is improper (e.g., $A = 5$). It is necessary to choose a sufficiently large A value. In this paper, the parameter A is chosen as 20, 50, or 80, which all work very well for this map-building and CNN model. Therefore, the A value can be chosen within a broad range in this model.

Although each neuron has only local connections in a small region and the targets are the only positive external stimuli, the positive neural activity can propagate to the whole state space of the neural network. Therefore, the lateral connections among neurons are essential in forming the dynamic neural activity landscape. The present model is not sensitive to the connection weight function $f(a)$ in (6), which can be chosen as any monotonically decreasing function. The connection weight is determined by parameter μ. Therefore, μ is an important factor in the present model.

Parameter r_0 determines the size of the receptive field of the neuron, which is not an important factor in the present model. A large value r_0 will increase the propagation of the neural activity. However, when applying the model to cleaning robot covering problems, a small value is necessary, e.g., $r_0 = 2$, since it is required that the activity cannot pass through any obstacles (wall). Thus, $r_0 = 2$ is used for all cases. Note that, in this map-building and CCN model, the radius r_0 of the circle denotes the receptive field of the ith neuron, while the radius R_s of the circle implies the limited range the robot can sense to some extent. Therefore, they are substantially different.

Parameter E determines the amplitude of the external inputs from the target (unclean areas) and obstacles. To keep the target (unclean areas) and obstacles staying at the peak and the valley, respectively, the value E should be chosen as a very large value over the total input from the lateral connections. To illustrate the influence of E value to the neural network, two simulations are performed for comparison via the same case in a unknown environment [see Fig. 8(b)]. In that case, recall that the model parameters are set as: $A = 80$, $B = 1$, and $D = 1$ for the shunting equation; $\mu = 0.7$ and $r_0 = 2$ for the lateral connections; and $E = 100$ for the external inputs. In this case, a much larger value $E = 200$ is first selected to replace $E = 100$, while keeping other parameters including A value identical.

Once the robot reaches point T, the same point as in Fig. 8(b), the neural activity landscape is shown in Fig. 16(c). It displays that the neural activity landscape is not obviously different from that in Fig. 8(b). A much larger E value does not affect passive decay rate of neural activity.

If E value is changed to a much smaller value $E = 50$, while other parameters are retained, the neural activity landscape is shown in Fig. 16(d). This simulation is carried out when the robot arrives at point T, the same as the case in Figs. 8(b) and 16(c). It exhibits that the neural activity landscape is not significantly different from those in Figs. 8(b) and 16(c). A much smaller E value does not change passive decay rate of neural activity, except there are different amplitudes of neural activity.

Since the neural activity is bounded at the interval $[-D, B]$, by choosing $B = 1$, $D = 1$, and $r_0 = 2$, the maximum total input from lateral connections is eight, then choosing any value $E > 40$ is sufficiently good. Therefore, parameter E is not an important factor in the present model. In this paper, when E is chosen as 50, 100, or 200, the model works well. Thus, the E value may be chosen within a large range in this model.

In summary, only two parameters A and μ are important in the proposed model. The model dynamics is determined by the value A. Parameter μ determines the activity propagation among neurons. The parameter values in the above simulations

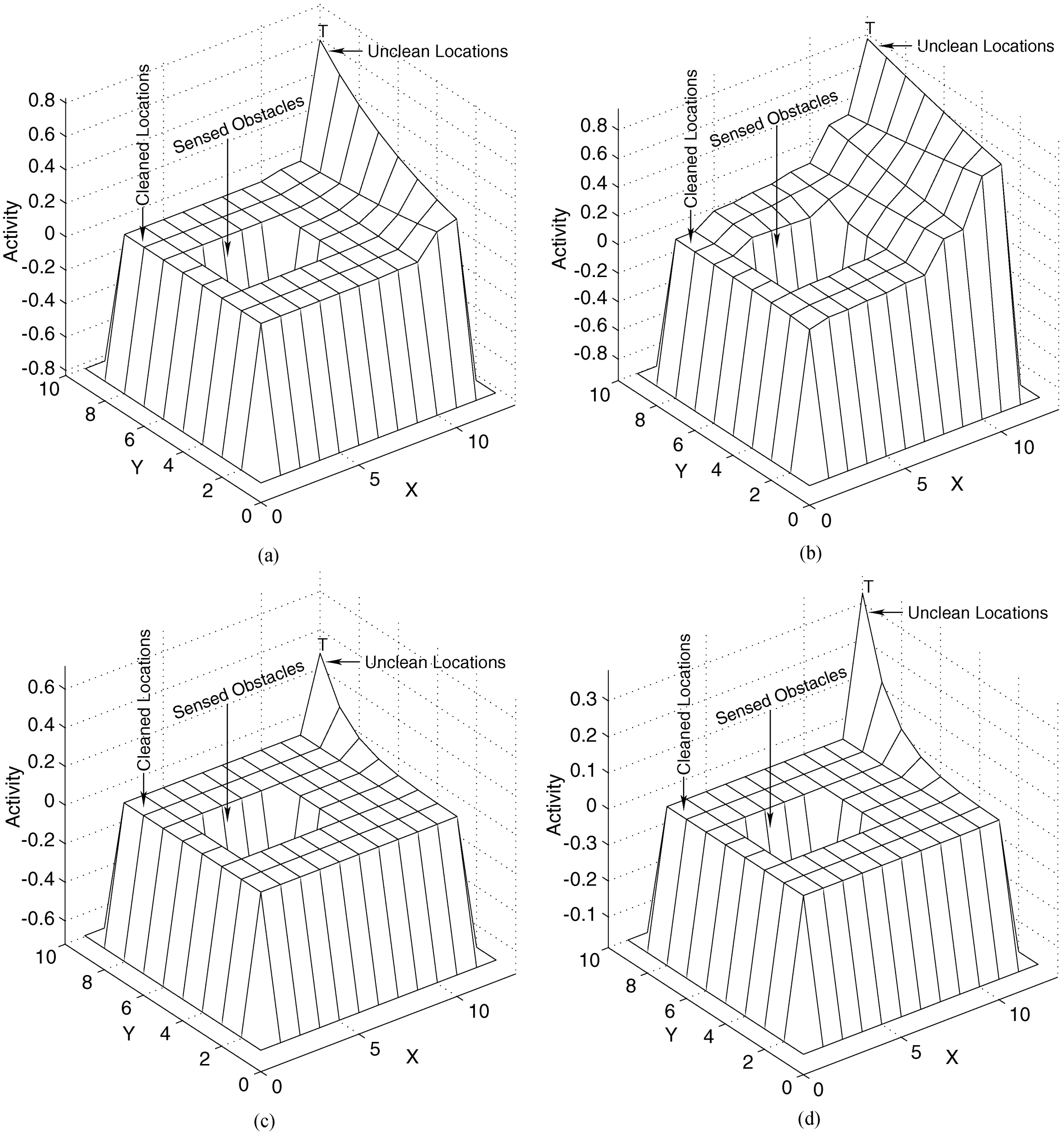


Fig. 16. Activity landscape of the neural network for the case in Fig. 8(b) by different parameters A and E. (a) $A = 20$ instead of $A = 80$. (b) $A = 5$ other than $A = 80$. (c) $E = 200$ instead of $E = 100$. (d) $E = 50$ other than $E = 100$.

are chosen in a very wide range. It is evident that the proposed model is not very sensitive to the model parameters.

VIII. Conclusion

A neural-dynamics-based algorithm for real-time concurrent map building and CCN in unknown environments is proposed. The developed complete coverage algorithms and map-building approach are capable of autonomously planning real-time collision-free paths for cleaning robots in completely unknown environments. The model algorithm is computationally efficient. The model parameter values can be chosen in a wide range. The map is built during the robot navigation. The complete coverage robot path is planned through the dynamic neural activity landscape and coverage algorithms with limited sensory information without any *templates*, without explicitly optimizing any *global* cost functions, without any *prior* knowledge of the dynamic environment, and without any learning procedures. The feasibility and efficiency of the proposed algorithms are discussed and illustrated through simulation studies and further comparison studies in various unknown environments. The proposed model is applicable in various unknown environments such as indoor room, dynamically changing, template case, and unstructured environments.

References

[1] E. González, A. Suarez, C. Moreno, and F. Artigue, "Complementary regions: A surface filling algorithm," in *Proc. IEEE Intl. Conf. Robot. Autom.*, Minneapolis, MN, 1996, pp. 909–914.

[2] 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," *Int. J. Robot. Res.*, vol. 22, pp. 441–466, 2003.

[3] S. C. Wong and B. A. MacDonald, "A topological coverage algorithm for mobile robots," in *Proc. IEEE/RSJ Intl. Conf. Intell. Robots Syst.*, Las Vegas, NV, 2003, pp. 1685–1690.

[4] E. González, O. Álvarez, Y. Díaz, C. Parra, and C. Bustacara, "BSA: A complete coverage algorithm," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, 2005, pp. 2040–2044.

[5] G. Schmidt and C. Hofner, "An advanced planning and navigation approach for autonomous cleaning robot operations," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Victoria, BC, Canada, 1998, pp. 1230–1235.

[6] P. W. Tse, S. Lang, K. C. Leung, and H. C. Sze, "Design of a navigation system for a household mobile robot using neural networks," in *Proc. Int. Conf. Neural Netw.*, Anchorage, AK, 1998, pp. 2151–2156.

[7] J. Palacín, T. Palleja, I. Valgañón, R. Pernia, and J. Roca, "Measuring coverage performances of a floor cleaning mobile robot using a vision system," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, 2005, pp. 4236–4241.

[8] J. Palacín, J. A. Salse, I. Valgañón, and X. Clua, "Building a mobile robot for a floor-cleaning operation in domestic environments," *IEEE Trans. Instrum. Meas.*, vol. 53, no. 5, pp. 1418–1424, Oct. 2004.

[9] F. J. Toledo, J. D. Luis, L. M. Tomas, M. A. Zamora, and H. Martinez, "Map building with ultrasonic sensors of indoor environments using neural networks," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, Nashville, TN, 2000, pp. 920–925.

[10] R. Araujo, "Prune-able fuzzy ART neural architecture for robot map learning and navigation in dynamic environments," *IEEE Trans. Neural Netw.*, vol. 17, no. 5, pp. 1235–1249, Sep. 2006.

[11] B. Banerjee, "String tightening as a self-organizing phenomenon," *IEEE Trans. Neural Netw.*, vol. 18, no. 5, pp. 1463–1471, Sep. 2007.

[12] C.-H. Chen and K.-T. Song, "Complete coverage motion control of a cleaning robot using infrared sensors," in *Proc. IEEE Int. Conf. Mechatron.*, Taipei, Taiwan, 2005, pp. 543–548.

[13] H. M. Choset, E. U. Acar, A. A. Rizzi, and J. E. Luntz, "Sensor-based planning: Exact cellular decompositions in terms of critical points," in *Proc. SPIE—Int. Soc. Opt. Eng.*, Boston, MA, 2001, pp. 204–215.

[14] E. U. Acar, H. Choset, and P. N. Atkar, "Complete sensor-based coverage with extended-range detectors: A hierarchical decomposition in terms of critical points and voronoi diagrams," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Maui, HI, 2001, pp. 1305–1311.

[15] V. J. Lumelsky, S. Mukhopadhyay, and K. Sun, "Dynamic path planning in sensor-based terrain acquisition," *IEEE Trans. Robot. Autom.*, vol. 6, no. 4, pp. 462–472, Aug. 1990.

[16] P. Jasiobedzki, "Detecting drivable floor regions," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Pittsburgh, PA, 1995, pp. 264–270.

[17] J. Y. Park and K. D. Lee, "A study on the cleaning algorithm for autonomous mobile robot under the unknown environment," in *Proc. 6th IEEE Int. Workshop Robot Human Commun.*, Sendai, Japan, 1997, pp. 70–75.

[18] Z. J. Butler, A. A. Rizzi, and R. L. Hollis, "Contact sensor-based coverage of rectilinear environments," in *Proc. IEEE Int. Symp. Intell. Control Intell. Syst.*, Cambridge, MA, 1999, pp. 266–271.

[19] C. Hofner and G. Schmidt, "Path planning and guidance techniques for an autonomous mobile robot," *Robot. Autonom. Syst.*, vol. 14, no. 203, pp. 199–212, 1995.

[20] Y. Liu, S. Zhu, B. Jin, and H. G. S. Feng, "Sensory navigation of autonomous cleaning robots," in *Proc. 5th World Congr. Intell. Control Autom.*, Hangzhou, China, 2004, pp. 4793–4796.

[21] A. Arleo, F. Smeraldi, and W. Gerstner, "Cognitive navigation based on nonuniform Gabor space sampling, unsupervised growing networks, and reinforcement learning," *IEEE Trans. Neural Netw.*, vol. 15, no. 3, pp. 639–652, May 2004.

[22] S. X. Yang and M. Q.-H. Meng, "Real-time collision-free motion planning of mobile robots using neural dynamics based approaches," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1541–1552, Nov. 2003.

[23] S. X. Yang and M. Meng, "Neural network approaches to dynamic collision-free robot trajectory generation," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 31, no. 3, pp. 302–318, Jun. 2001.

[24] A. Zhu and S. X. Yang, "A neural network approach to task assignment of multi-robots," *IEEE Trans. Neural Netw.*, vol. 17, no. 5, pp. 1278–1287, Sep. 2006.

[25] S. X. Yang and C. Luo, "A neural network approach to complete coverage path planning," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 34, no. 1, pp. 718–725, Feb. 2004.

[26] F. Yasutomi, D. Takaoka, M. Yamada, and K. Tsukamoto, "Cleaning robot control," in *Proc. IEEE Int. Conf. Robot. Autom.*, Philadelphia, PA, 1988, pp. 1839–1841.

[27] H. Choset, "Coverage of known spaces: The boustrophedon cellular decomposition," *Autonom. Robots*, vol. 9, no. 3, pp. 247–253, 2000.

[28] J. S. Oh, Y. H. Choi, J. B. Park, and Y. F. Zheng, "Complete coverage navigation of cleaning robots using triangular-cell-based map," *IEEE Trans. Ind. Electron.*, vol. 51, no. 3, pp. 718–726, Jun. 2004.

[29] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *J. Physiol. (London)*, vol. 117, pp. 500–544, 1952.

[30] H. Furuhashi, S. Minami, H. Takemoto, and K. Endo, "Development of an autonomous cleaning robot with an external power cable," in *Proc. 1st Int. Conf. Motion Vib. Control*, Yokohama, Japan, 1992, pp. 499–504.

[31] S. Lang and P. Tse, "Smart vacuum cleaner," in *Proc. 2nd Int. Conf. Mechatron. Mach. Vis. Practice*, Hong Kong, 1995, pp. 264–269.

[32] D. C. MacKenzie and T. R. Balch, "Making a clean sweep: Behavior based vacuuming," in *Proc. AAAI Fall Symp. Instantiating Real-World Agents*, Raleigh, NC, 1993, pp. 93–98.

[33] M. Shimamura, H. Okano, M. Kimura, and K. Kanasaki, "Development of vacuum cleaning device in BWR vessels," in *Proc. Amer. Nuclear Soc. 7th Topical Meeting Robot. Remote Syst.*, Augusta, GA, 1997, pp. 794–801.

[34] I. Ulrich, F. Mondada, and J.-D. Nicoud, "Autonomous vacuum cleaner," *Robot. Autonom. Syst.*, vol. 19, pp. 233–245, 1997.

[35] M. Yamamoto, "'Sozzy': A hormone-driven autonomous vacuum cleaner," in *Proc. SPIE—Int. Soc. Opt. Eng.*, Boston, MA, 1994, vol. 2058, Mobile Robots VIII, pp. 211–223.

[36] A. C. Branch, "Case study of a floor cleaning robot," in *Proc. SPIE—Int. Soc. Opt. Eng.*, Pittsburgh, PA, 1998, Mobile Robots XII, pp. 119–129.

[37] M. Simoncelli, G. Zunino, H. I. Christensen, and K. Lange, "Autonomous pool cleaning: Self localization and autonomous navigation for cleaning," *Autonom. Robots*, vol. 9, no. 3, pp. 261–270, 2000.

[38] A. Mital, M. Kulkami, R. Huston, and S. Anand, "Robot cleaning of underground liquid storage tanks: Feasibility and design considerations," *Robot. Autonom. Syst.*, vol. 20, no. 1, pp. 49–60, 1997.

[39] H. Endres, W. Feiten, and G. Lawitzky, "Field test of a navigation system: Autonomous cleaning in supermarkets," in *Proc. IEEE Int. Conf. Robot. Autom.*, Leuven, Belgium, 1998, pp. 1779–1781.

[40] E. Prassler, D. Schwammkrug, B. Rohrmoser, and G. Schmidl, "A robotic road sweeper," in *Proc. IEEE Int. Conf. Robot. Autom.*, San Francisco, CA, 2000, pp. 2364–2369.

[41] Y. Takahashi and O. Tsubouchi, "Tension control of wire suspended mechanism and application to bathroom cleaning robot," in *Proc. 39th Annu. Conf. Soc. Instrum. Control Eng.*, Iizuka, Japan, 2000, pp. 143–147.

[42] T. Bohme, U. Schmucker, N. Elkmann, and M. Sack, "Service robots for facade cleaning," in *Proc. 24th Annu. Conf. IEEE Ind. Electron. Soc.*, Aachen, Germany, 1998, pp. 1204–1207.

[43] E. Prassler, E. Stroulia, and M. Strobel, "Office waste clean up: An application for service robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Albuquerque, NM, 1997, pp. 1863–1868.

[44] K. Ratcliff, M. Farsi, and K. Z. Karam, "Enhancement of the cleaning effectiveness on window surfaces," in *Proc. 12th Int. Conf. Syst. Sci.*, Wroclaw, Poland, 1995, pp. 513–517.

[45] R. D. Schraft, U. Brauning, T. Orlowski, and M. Hornemann, "Automated cleaning of windows on standard facades," *Autom. Construction*, vol. 9, no. 5–6, pp. 489–501, 2000.

[46] K. S. S. Lian, "Domestic service robot," in *Proc. 29th Int. Symp. Robot. Autom. Control Instrum.*, Birmingham, U.K., 1998, pp. 463–466.

[47] F. Marrone and H. H. Strobel, "Cleaning assistant—A service robot designed for cleaning tasks," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatron.*, Como, Italy, 2001, pp. 1041–1046.

[48] C. Wattanasin, Y. Aiyama, D. Kurabayashi, J. Ota, and T. Arai, "A hybrid technique to supply indoor service robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Leuven, Belgium, 1998, pp. 89–94.

[49] P. N. Atkar, H. Choset, A. A. Rizzi, and E. U. Acar, "Exact cellular decomposition of closed orientable surfaces embedded in R^3," in *Proc. IEEE Int. Conf. Robot. Autom.*, Seoul, Korea, 2001, pp. 699–704.

[50] P. Fiorini and E. Prassler, "Cleaning and household robots: A technology survey," *Autonom. Robots*, vol. 9, no. 3, pp. 227–235, 2000.

[51] R. Kurazume and S. Hirose, "Development of a cleaning robot system with cooperative positioning system," *Autonom. Robots*, vol. 9, no. 3, pp. 237–246, 2000.

[52] M. Schofield, "Cleaning robots engineering dream or commercial vision?," in *Proc. 29th Int. Symp. Robot.*, Birmingham, U.K., 1998, pp. 50–51.

[53] M. Schofield and H. Grunke, "Cleaning robots from concept to product—The users point view," in *Proc. 25th Int. Symp. Ind. Robot.*, Hannover, Germany, 1994, pp. 233–239.

[54] E. Prassler, A. Pitter, C. Schaeffer, and P. Fiorini, "A short history of cleaning robots," *Autonom. Robots*, vol. 9, no. 3, pp. 211–226, 2000.

[55] K. Kamiya, S. Iida, T. Koyama, T. Kominami, N. Matsumoto, and A. Toyoda, "Vehicle control for clean room mobile robot," in *Proc. Int. Robots Vis. Conf.*, Detroit, MI, 1997, pp. 1–15.

[56] C. Pellerin, "The service robot market," *Int. J. Service Robot*, vol. 1, no. 3, pp. 17–20, 1995.

[57] H. Yaguchi, "Robot introduction to cleaning work in the east japan railway company," *Adv. Robot.*, vol. 10, no. 4, pp. 403–414, 1996.

[58] Y. Zhou and M. Skibniewski, "Construction robot force control in cleaning operations," *J. Aerosp. Eng.*, vol. 7, no. 1, pp. 33–49, 1994.

[59] Y. Wang, S. Liu, D. Xu, Y. Zhao, H. Shao, and X. Gao, "Development and application of wall-climbing robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Detroit, MI, 1999, pp. 1207–1212.

[60] F.-W. Bach, M. Rachkov, J. Seevers, and M. Hahn, "High tractive power wall-climbing robot," *Autom. Construction*, vol. 4, no. 3, pp. 213–224, 1995.

[61] M. Farsi, K. Ratcliff, P. J. Johnson, C. R. Allen, K. Z. Karam, and R. Pawson, "Robot control system for window cleaning," in *Proc. 11th Int. Symp. Autom. Robot. Construction*, Brighton, U.K., 1994, pp. 617–623.

[62] S. V. Ulyanov, T. Fukuda, and V. G. Gradetsky, "Intelligent control system for wall climbing robots," in *Proc. Symp. Robot. Cybern. Comput. Eng. Syst. Appl.*, Lille, France, 1997, pp. 794–799.

[63] Y. Zhao, H. Shao, and Y. Wan, "Wall-climbing robot with negative pressure sucker used for cleaning work," *High Technol. Lett. (English Language Edition)*, vol. 5, no. 2, pp. 85–88, 1999.

[64] C. Chang, "Using sensor habituation in mobile robots to reduce oscillatory movements in narrow corridors," *IEEE Trans. Neural Netw.*, vol. 16, no. 6, pp. 1582–1589, Nov. 2004.

[65] R. Sharaf and A. Noureldin, "Sensor integration for satellite-based vehicular navigation using neural networks," *IEEE Trans. Neural Netw.*, vol. 18, no. 2, pp. 589–594, Mar. 2007.

[66] G. Lawitzky, "Sinas—A navigation system for service robots," *Intl. J. Ind. Robot*, vol. 26, no. 6, pp. 451–455, 1999.

[67] G. Lawitzky, "A navigation system for cleaning robots," *Autonom. Robots*, vol. 9, pp. 255–260, 2000.

[68] A. L. Rankin, C. D. Crane, and D. G. Armstrong, "Autonomous path planning navigation system used for site characterization," in *Proc. SPIE—Int. Soc. Opt. Eng.*, Orlando, FL, 1998, pp. 176–186.

[69] H. Öğmen and S. Gagné, "Neural network architectures for motion perception and elementary motion detection in the fly visual system," *Neural Netw.*, vol. 3, no. 5, pp. 487–505, 1990.

[70] S. Grossberg, "Nonlinear neural networks: Principles, mechanisms, and architecture," *Neural Netw.*, vol. 1, pp. 17–61, 1988.

[71] S. Grossberg, "Absolute stability of global pattern formation and parallel memory storage by competitive neural networks," *IEEE Trans. Syst. Man Cybern.*, vol. 13, no. 5, pp. 815–926, Sep. 1983.

[72] Y. Gabriely and E. Rimon, "Spanning-tree based coverage of continuous areas by a mobile robot," *Ann. Math. Artif. Intell.*, vol. 31, no. 1–4, pp. 77–98, 2001.

Chaomin Luo (S'01) received the B.Eng. degree in electrical engineering from the Southeast University, Nanjing, China, in 1988 and the M.Sc. degree in engineering system and computing from the University of Guelph, Guelph, ON, Canada, in 2002. He is currently working towards the Ph.D. degree in the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada.

His research interests lie in two areas. The first is in robotics and automation, intelligent system, control systems, soft computing, instrumentation, measurement, and mechatronics. The second is in very large scale integration (VLSI) physical design automation, VLSI floorplanning and placement, and optimization in electronic design automation.

Mr. Luo has been awarded several Ontario Graduate Scholarships (OGS) and a Postgraduate Scholarship (PGS) from the Natural Sciences and Engineering Research Council (NSERC) of Canada. He received the Best Student Paper Presentation Award at the 2007 Canadian Operational Research Society Conference (SWORD). He is a recipient of the 2003–2005 Graduate Incentive Award and the 2005–2006 President's Graduate Scholarship at the University of Waterloo.

Simon X. Yang (S'97–M'99) received the B.Sc. degree in engineering physics from Beijing University, Beijing, China, in 1987, the M.Sc. degree in biophysics from Chinese Academy of Sciences, Beijing, China, in 1990, the M.Sc. degree in electrical engineering from the University of Houston, TX, in 1996, and the Ph.D. degree in electrical and computer engineering from the University of Alberta, Edmonton, AB, Canada, in 1999.

He joined the School of Engineering, University of Guelph, Guelph, ON, Canada, in 1999, where he is currently Professor and Head of the Advanced Robotics and Intelligent Systems (ARIS) Laboratory. His research interests include intelligent systems, robotics, sensors, wireless sensor networks, multisensor fusion, control systems, soft computing, and computational neuroscience.

Prof. Yang serves as an Associate Editor of the IEEE TRANSACTIONS OF NEURAL NETWORKS, the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, the *International Journal of Robotics and Automation*, and several other international journals. He was involved in the organization committees of many conferences. He was the General Chair of the 2006 International Conference on Sensing, Computing, and Automation. Among many of his awards, he is a recipient of the Distinguished Professor Award for 2004–2006 from the University of Guelph.