[6] W. Chen, J. K. Mills, J. Chu, and D. Sun, "Uniform ultimate boundedness of a fuzzy logic controlled industrial robot," *J. Robot. Syst.*, vol. 18, no. 9, pp. 553–561, 2001.

[7] Y. Ch. Hsu, G. Chen, and E. Sanchez, "A fuzzy PD controller for multi-link robot control: stability analysis," in *Proc. IEEE Int. Conf. Robotics and Automation*, Albuquerque, NM, 1997, pp. 1412–1417.

[8] T. Fukuda and T. Shibata, "Hierarchical intelligent control for robotic motion by using fuzzy, artificial intelligence, and neural networks," in *Proc. Int. J. Conf. Neural Networks*, vol. 1, Baltimore, MD, 1992, pp. 269–274.

[9] S. Commuri and F. L. Lewis, "Adaptive-fuzzy logic control of robots manipulators," in *Proc. IEEE Int. Conf. Robotics and Automation*, Mineapolis, MN, 1996, pp. 2604–2609.

[10] B. K. Yoo and W. C. Ham, "Adaptive control of robot manipulators using fuzzy compensator," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 2, pp. 186–199, 2000.

[11] J. M. Meslin, J. Zhou, and P. Coiffet, "Fuzzy dynamic control of robot manipulators: A scheduling approach," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Le Touquet, France, 1993, pp. 69–73.

[12] G. Calcev, "Some remarks on the stability of Mamdani fuzzy control systems," *IEEE Trans. Fuzzy Syst.*, vol. 6, pp. 436–442, Aug. 1998.

[13] G. Calcev, R. Gorez, and M. de Neyer, "Passivity approach to fuzzy control systems," *Automatica*, vol. 34, no. 3, pp. 339–344, 1998.

[14] M. Spong and M. Vidyasagar, *Robot Dynamics and Control*. New York: Wiley, 1989.

[15] M. Vidyasagar, *Nonlinear Systems Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[16] L. X. Wang, *A Course in Fuzzy Systems and Control*. Upper Saddle River, NJ: Prentice-Hall, 1997.

[17] F. Reyes and R. Kelly, "Experimental evaluation of identification schemes on a direct-drive robot," *Robotica*, vol. 15, pp. 563–571, 1997.

[18] ——, "Experimental evaluation of model-based controllers on a direct-drive robot arm," *Mechatronics*, vol. 11, pp. 267–282, 2001.

[19] D. M. Dawson, J. J. Carroll, and M. Schneider, "Integrator backstepping control of a brush DC motor turning a robotic load," *IEEE Trans. Contr. Syst. Technol.*, vol. 2, pp. 233–244, Sept. 1994.

[20] M. S. De Queiroz, D. Dawson, and T. Burg, "Reexamination of the DCAL controller for rigid link robots," *Robotica*, vol. 14, pp. 41–49, 1996.

A Neural Network Approach to Complete Coverage Path Planning

Simon X. Yang and Chaomin Luo

***Abstract*—Complete coverage path planning requires the robot path to cover every part of the workspace, which is an essential issue in cleaning robots and many other robotic applications such as vacuum robots, painter robots, land mine detectors, lawn mowers, automated harvesters, and window cleaners. In this paper, a novel neural network approach is proposed for complete coverage path planning with obstacle avoidance of cleaning robots in nonstationary environments. The dynamics of each neuron in the topologically organized neural network is characterized by a shunting equation derived from Hodgkin and Huxley's (1952) membrane equation. There are only local lateral connections among neurons. The robot path is autonomously generated from the dynamic activity landscape of the neural network and the previous robot location. The proposed model algorithm is computationally simple. Simulation results show that the proposed model is capable of planning collision-free complete coverage robot paths.**

***Index Terms*—Cleaning robots, complete coverage path planning, neural dynamics, neural network, obstacle avoidance.**

I. Introduction

Path planning is a fundamentally important issue in robotics. Complete coverage path planning (CCPP, which is also called region filling or area covering) of cleaning robots is a special type of path planning in a two-dimensional (2-D) environment, which requires the robot path to pass through every area in the workspace. In addition to cleaning robots, many other robotic applications also require complete coverage path planning, e.g., vacuum robots [1], painter robots [2], autonomous underwater covering vehicles [3], de-mining robots [4], land mine detectors [5], lawn mowers [6], automated harvesters [7], agricultural crop harvesting equipment [8], and window cleaners [9]. Autonomous cleaning robots are particularly useful in hazardous environments. There have been many studies on CCPP using various approaches, e.g., artificial potential field, approximate cellular decomposition, exact cellular decomposition, template-based model, neural networks, and fuzzy logic.

Many approaches to CCPP use artificial potential fields. Pirzadeh and Snyder [10] proposed an indirect control strategy to deal with the coverage and search problem, where the complete coverage is accomplished using an artificial potential field. The underlying strategy of the algorithm is to discretize the workspace and the robot motion. The robot movement is designated with four orthogonal directions—up, down, left, and right—without considering any diagonal neighbors. Moravec and Elfes [11] first proposed an approximate cellular decomposition model, where the workspace is decomposed into cells with the same size and shape. Zelinsky [12] developed another approximate cellular decomposition approach using a grid based complete coverage model. A distance transform algorithm is used to assign a specific number to each grid element, which is a function of the distance to the goal. The complete coverage is then achieved by a gradient descent rule. The generated path may include some unexpected turns. Recently, Gabriely and Rimon [13] proposed a spanning tree covering approach

Manuscript received August 22, 2001; revised August 27, 2002. This work was supported by Natural Sciences and Engineering Research Council (NSERC) and Materials and Manufacturing Ontario (MMO) of Canada. The review of this paper was recommended by Associate Editor C. T. Lin.

The authors are with Advanced Robotics and Intelligent Systems (ARIS) Laboratory, School of Engineering, University of Guelph, Guelph, ON, N1G 2W1, Canada (e-mail: syang@uoguelph.ca; cluo@uoguelph.ca).

Digital Object Identifier 10.1109/TSMCB.2003.811769

by subdividing the workspace into discrete cells and following a spanning tree of a graph induced by the cells. The robot is able to cover every point precisely once, and travel an optimal path in a grid-like representation in the workspace.

There are many exact cellular decomposition based approaches to CCPP as well. The fundamental concept is to decompose the workspace into a collection of nonoverlapping cells, and then, the robot searches the connectivity graph that represents the adjacency relation among cells. Thus the complete coverage can be achieved by back and forth robot motions. Choset [14] proposed a novel boustrophedon cellular decomposition approach by breaking down the workspace, which allows the robot to cover each cell like "the way of the ox." This solution 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 locations and the critical points. Acar and Choset [15] later improved the boustrophedon cellular decomposition algorithm to ensure the covering task does not require any prior knowledge. Recently, Atkar *et al.* [2] applied the exact cellular decomposition to complete coverage of a closed orientable surface in three dimensions. Cao *et al.* [6] proposed an approach to fill the whole region with obstacle avoidance for a lawn mower robot using a raster scanning strategy that is related to the sensor based boustrophedon decomposition. This region-filling algorithm requires the boundary of obstacles and walls in advance. Butler *et al.* [16] proposed a DC_R (distributed coverage of rectilinear environments) algorithm, where the robot can detect obstacles by contact sensing in a shared, connected rectilinear environment. Hert *et al.* [3] presented an online terrain-covering algorithm for a robot moving in an unknown environment, where the robot path can cover the entire workspace without repeating the cleaned area multiple times. The free space is partially discretized into cells with fixed width but with varying top and bottom. When the robot arrives at a corner, it has to remember the environment information.

Another common method for CCPP is template based approaches. Hofner and Schmidt [17] proposed a template based approach using motion templates and motion mosaic, where the obstacles are modified by subgoals. This model requires a predefined map and memorizes the templates. Thus, it is difficult to handle environmental changes. In addition, it has difficulty identifying what area is uncovered. Thus, there exist uncovered areas near obstacles even after more templates such as side-shift ("SS") template are supplemented. De Carvalho *et al.* [18] proposed another template based CCPP model based on the prior knowledge of the 2-D map of a cleaning environment.

There are some neural networks and fuzzy logic based approaches to CCPP. Tse *et al.* [19] developed a backpropagation neural network based model, which can generate a robot path through self-learning. During the cleaning process, the robot should record the previously generated path, the number of grid points traveled and the number of turns. If the robot encounters a new environment, the memory map has to be updated. Yasutomi *et al.* [1] also presented a learning based CCPP approach, which can effectively avoid obstacles and walls in an unknown environment. Due to the computational complexity for the learning, this model has difficulty dealing with unstructured environments. Thus it is suitable for well-structured indoor environments. Fu and Lang [20] proposed a fuzzy logic based method for CCPP, which is able to effectively correct and reduce the robot motion direction errors. When the robot sees obstacles with irregular shape, however, it ignores some regions in the vicinity of obstacles as it can follow straight lines only to skirt the obstacles. Lang and Chee [21] proposed a behavior model based on fuzzy logic, which has the ability to guide a robot to clean an unstructured room environment from any starting location. Due to the difficulty in defining suitable fuzzy rules, the generated paths are not smooth enough at turning and traversing.

Various other approaches were also proposed for CCPP. Arkin and Hassin [22] proposed a novel approach based on covering salesman problem (CSP), which is a variant of the famous traveling salesman problem (TSP). CSP requires the agent to visit all points in the object space instead of just passing over each city with a minimal travel distance. Russell [23] proposed a CCPP approach by marking the robot path with a heat trail; then, the robot can avoid repeating previously marked areas. Chang and Shyu [24] proposed a CCPP model by dividing the workspace into subregions with wires that are buried under the ground, where the mobile robot cleans each subregion one by one. The robot visits each regular subregion along the wires. If the environment changes, the workspace has to be redivided. Park and Lee [25] proposed a CCPP model that consists of three components: a sweeping algorithm, a point-to-point moving algorithm and a corner work algorithm. However, the cleaning robots may overlap some areas and miss some corner areas.

There are several studies on coverage path planning of multirobot systems. Kurabayashi *et al.* [26] proposed an offline path planning algorithm for multiple cleaning robots, which is suitable for covering unstructured environments. The cleaning performance depends on the step length of the robot movement and the shape of the cleaned area. Thus the robot is unable to plan in real time. They also proposed a floor cleaning path planning algorithm for cooperative sweeping with movable obstacles [27], which is able to calculate an appropriate path distribution so that it can generate a complete coverage path. These approaches require the number, size and location of all the movable obstacles [26], [27]. Rekleitis *et al.* [28] employed a graph-like decomposition of space to deal with the cooperative sweeping problem. The robots together explore the environmental information and carry out the cooperative task, where each robot is regarded as a beacon for the others. Wagner and Bruckstein [29] presented an approximate cellular decomposition approach for multirobot sweeping. The dirt grid on the floor is employed for communication among robots. Each robot communicates with the others by leaving traces.

In this paper, a novel neural network approach is proposed for complete coverage path planning of a single robot and multiple cleaning robots, which is based on a neural network model for *conventional* real-time path planning for a mobile or manipulation robot in a nonstationary and unstructured environment [30]–[33]. The state space of the topologically organized neural network is the 2-D Cartesian workspace of a cleaning robot. The dynamics of each neuron is characterized by a shunting equation derived from Hodgkin and Huxley's [34] membrane model for a biological neural system. There are only local lateral connections among neurons. The varying environment is represented by the dynamic activity landscape of the neural network. For cost efficiency in term of a shorter path and fewer turns, the real-time robot motion is directly generated from the dynamic neural activity landscape and the previous robot location, which is distinct from the models for conventional robot path planning in [30]–[33]. The robot path is autonomously planned without any prior knowledge of the time-varying environment, without explicitly searching globally over the free space, without explicitly optimizing any global cost functions, and without any learning procedures. Therefore the model algorithm is computationally simple. The proposed model is capable of planning real-time complete coverage paths with obstacle avoidance in an unstructured indoor environment. The term "real-time" is in the sense that the coverage path planer responds immediately to the dynamic environment including the robot, targets (unclean areas) and obstacles. To the best of our knowledge, it is the first time that a nonlearning based neural network approach is developed for real-time complete coverage path planning.

The paper is organized as follows. In Section II, the proposed neural network approach to real-time CCPP is presented. The simulation studies of the proposed approach are presented in Section III, where a cleaning robot is in a typical template situation, in a dynamic environment, in an unstructured environment, and in cooperative sweeping with other robots. Finally, some concluding remarks are summarized in Section IV.

II. Proposed Model

In this section, the originality of the proposed neural network approach to CCPP is briefly introduced. Then, the model algorithm of the proposed approach is presented.

A. Originality

In 1952, Hodgkin and Huxley [34] proposed a computational model for a patch of membrane in a biological neural system using electrical circuit elements. In this model, the dynamics of the voltage across the membrane V_m is described using a state equation technique as

$$C_m \frac{dV_m}{dt} = -(E_p + V_m)g_p + (E_{Na} - V_m)g_{Na} - (E_K + V_m)g_K \quad (1)$$

where C_m is the membrane capacitance, and E_K, E_{Na}, and E_p are the Nernst potentials (saturation potentials) for potassium ions, sodium ions, and passive leak current in the membrane, respectively. Parameters g_K, g_{Na}, and g_p represent the conductances of potassium, sodium, and passive channels, respectively. This model provided the foundation of the shunting model and led to many model variations and applications [31], [33], [35].

By setting $C_m = 1$ and substituting $x_i = E_p + V_m$, $A = g_p$, $B = E_{Na} + E_p$, $D = E_k - E_p$, $S_i^e = g_{Na}$ and $S_i^i = g_K$ in (1), a shunting equation is obtained as [35]

$$\frac{dx_i}{dt} = -Ax_i + (B - x_i)S_i^e(t) - (D + x_i)S_i^i(t) \quad (2)$$

where x_i is the neural activity (membrane potential) of the ith neuron. Parameters A, B, and D are non-negative constants representing the passive decay rate and the upper and lower bounds of the neural activity, respectively. Variables S_i^e and S_i^i are the excitatory and inhibitory inputs to the neuron. This shunting model was first proposed by Grossberg to understand the real-time adaptive behavior of individuals to complex and dynamic environmental contingencies and has many applications such as visual perception and sensory motor control [31], [33], [35].

B. Model Algorithm

The fundamental concept of the proposed model is to develop a neural network architecture, whose dynamic neural activity landscape represents the dynamically varying environment. By properly defining the external inputs from the varying environment and the internal neural connections, the neural activities of 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 entire state space through neural activity propagation, whereas the obstacles have only local effect to avoid collisions. The real-time collision-free robot motion is planned based on the dynamic activity landscape of the neural network and the previous robot location, such that all areas will be cleaned and the robot will travel a smooth path with fewer turns.

The 2-D Cartesian workspace in the proposed approach is discretized into squares as in most other CCPP models. The diagonal length of each discrete area is equal to the robot sweeping radius that is the size of the robot effector or footprint. Thus sweeping an area can be achieved by traversing the center of that area represented by a discrete point. Once the robot visits a discrete point, it is assumed that the robot has covered (and cleaned) the discrete area of that point. If a cleaning robot covers every discrete point in a workspace, the robot path is then considered as a complete coverage path in the workspace.

The proposed neural network model is expressed topologically in a discretized workspace $\mathcal{W}$ of a cleaning robot. The location of the ith neuron in the state space $\mathcal{S}$ of the neural network, which is denoted by a vector $q_i \in R^2$, *uniquely* represents an area (a robot location) in $\mathcal{W}$. In the proposed model, the excitatory input results from the unclean areas and the lateral neural connections, whereas 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 neural network can be characterized by a shunting equation as

$$\frac{dx_i}{dt} = -Ax_i + (B - x_i)\left([I_i]^+ + \sum_{j=1}^{k} w_{ij}[x_j]^+\right) - (D + x_i)[I_i]^- \quad (3)$$

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

$$I_i = \begin{cases} E, & \text{if it is an unclean area} \\ -E, & \text{if it is an obstacle area} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where $E \gg B$ is a very large positive constant. 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 (2), respectively. Function $[a]^+$ is a linear-above-threshold function defined as $[a]^+ = \max\{a, 0\}$, and the nonlinear function $[a]^-$ is defined as $[a]^- = \max\{-a, 0\}$. The connection weight w_{ij} between the ith and jth neurons can be defined as

$$w_{ij} - f(|q_i - q_j|) \quad (5)$$

where $|q_i - q_j|$ represents the Euclidean distance between vectors q_i and q_j in the state space, and $f(a)$ can be any monotonically decreasing function, such as a function defined as

$$f(a) = \begin{cases} \frac{\mu}{a}, & \text{if } 0 < a < r_0 \\ 0, & \text{if } a \geq r_0 \end{cases} \quad (6)$$

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., $w_{ij} = w_{ji}$. Note that the neural connection weights that satisfy the fundamental concept of the proposed approach are predefined in (5) and (6) at the neural network design stage. Thus, neither online nor offline learning procedures are needed to obtain a proper connection weights among neurons. A schematic diagram of the neural network is shown in Fig. 1, 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. Thus, it has lateral connections only to its 8 neighboring neurons within its receptive field.

Because there are only excitatory neural connections in (3), the proposed neural network characterized by (3) guarantees that the positive neural activity can propagate to the *entire* state space, but the negative activity stays *locally* only. Therefore, the unclean areas *globally* attract the robot, whereas the obstacle areas just *locally* push the robot away to avoid collisions. The locations of the unclean areas, cleaned areas and obstacle areas may vary with time, e.g., there are moving obstacles; unclean areas become cleaned areas; or the cleaned areas become unclean again. The activity landscape of the neural network dynamically changes due to the varying external inputs from the changing environment and the internal activity propagation among neurons. For energy and time efficiency, the robot should travel a shorter path (with fewer

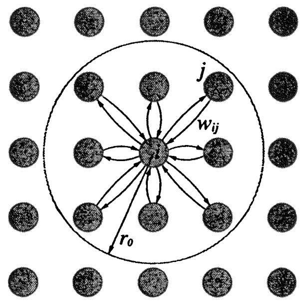

Fig. 1. Schematic diagram of the proposed neural network for complete coverage path planning.

areas visited more than once) and make fewer turns of moving direction. Distinct from the models for conventional robot path planning in [30]–[33], where the robot path is generated from the neural activity landscape *alone*, in the proposed CCPP model, the robot path is generated from both the dynamic activity landscape *and* the previous robot location to achieve fewer changes in navigation direction. For a given current robot location in the state space $\mathcal{S}$ (i.e., an area in workspace $\mathcal{W}$) denoted by p_c, the next robot location p_n (also called "command location") 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 locations of the current location p_c. Variable x_j is the neural activity of the jth neuron, and y_j is a monotonically increasing function of the difference between the current and next robot moving directions. Variable y_j can be defined as a function of the previous location p_p, the current location p_c, and the possible next location 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 absolute angle change between the current and next moving directions, 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 robot reaches its next location, the next location becomes a new current location. The current robot location *adaptively* changes according to the varying environment.

The dynamic activity landscape of the topologically organized neural network is used to determine the next robot location. The robot movement is also determined by the robot speed when the next location is available from the current activity landscape. However, when the next location is not immediately available, e.g., in a deadlock situation, the robot has to wait until the next location toward the targets is available from the neural activity landscape. Whenever the neural activity at the current robot location is smaller than the largest neural activity of its neighboring locations, the robot starts to move to its next location. Thus, the robot movement is determined by both the robot speed and the neural activity landscape. In a dynamic environment, the neural activity landscape will never reach a steady state as in a static environment. The moving speed of a cleaning robot can be assumed to be slow because of its cleaning task. In a fast changing environment, e.g., where obstacles suddenly appear in front of the robot, the neural activities at those locations will immediately reduce to a very large negative value due to their very large inhibitory input. Thus, the robot should be able to avoid those suddenly appearing obstacles. In the proposed model, due to the very large external input constant E, the unclean areas and the obstacle areas keep staying at the peak and the valley of the activity landscape of the neural network,

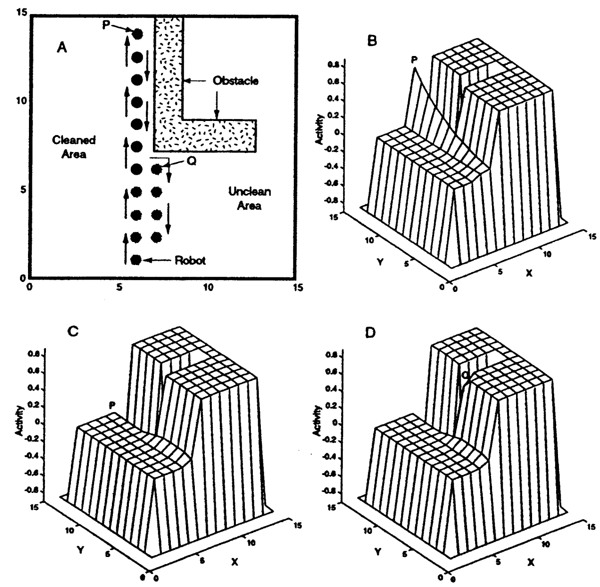

Fig. 2. Complete coverage path planning in a side-shift template situation. (a) Robot path. (b) Activity landscape when the robot arrives at Location P. (c) Activity landscape when the robot starts to escape from P. (d) Activity landscape after the robot arrives at Q.

respectively. The robot keeps moving toward an unclean area with obstacle avoidance until the designated objective is achieved or it is forced to stop by a human operator.

The proposed neural network model shares some common ideas with the standard artificial potential field and standard distance transform path planning techniques: a topologically organized discrete map is used to represent the workspace; each location uses a number to represent its environmental information; the target locations have the largest value; and the robot moves from a location with a smaller value to that with a larger value. However, there are important differences between the proposed model and the artificial potential field or distance transform based model. In a potential based approach, the potential at each location is calculated based on *all* the locations of the targets (unclean areas) and the obstacle locations throughout the *entire* workspace [36]. Thus, the locations in a potential based approach are *globally* connected, and the potential at each location is an explicit function of *all* the target and obstacle locations in the entire workspace. Once there is any change in the locations of targets and/or obstacles, all the potentials have to be recalculated to reflect the change. Thus it is more suitable for static environments and has difficulty dealing with fast changing environments. In addition, the standard potential based approaches may be trapped in deadlock situations, e.g., in the situation of Fig. 2, where the potential at Location P is larger than those at its neighboring locations. In the proposed model, each neuron has only *local* connections to its neighboring neurons and its input is only from its neighboring neurons, instead of all the target and obstacle locations in the entire workspace. Thus, it has potential VLSI implementation for parallel computation. Any changes in the environment directly result in the changes in the external inputs at those locations. The activity landscape of the neural network is *automatically* changing due to the neural activity propagation. Thus, it can deal with arbitrarily changing environments, and will not be trapped in any deadlock situations if a solution exists.

When the robot arrives in a deadlock situation, i.e., all the neighboring locations are either obstacles or cleaned locations [e.g., Location P in Fig. 2(a)], all the neural activities of its neighboring locations are *not* larger than the activity at the current location P [see Fig. 2(b)], because its neighboring locations receive either negative external input (obstacles) or no external input (cleaned locations), and all the cleaned neighboring locations passed a longer decay time as they were cleaned earlier than Location P. Thus, for a pure artificial potential field-based approach, the robot is unable to move away from such a deadlock situation. In the proposed model, the neural activity at the deadlock location P will quickly decay to zero due to the passive decay term $-Ax_i$ in (3). Meanwhile, due to the lateral excitatory connections among neurons, the positive neural activity from the unclean locations in the workspace will propagate toward the current robot location through neural activity propagation [see Fig. 2(c)]. Thus, the robot is able to find a smooth path from the current deadlock location directly to an unclean location, just in the same way as the conventional path planning from a start point to a target point [30], [33]. The robot continues its cleaning task until all the areas in the workspace become cleaned. Thus, the proposed model is capable of achieving complete coverage path planning.

The proposed neural network approach can be easily applied to a multirobot system. In some cleaning applications, the space needs to be cleaned in a limited short period. Thus, it requires multiple robots to share the cleaning task without any collisions. Thus, in cooperative sweeping of multiple cleaning robots, each robot must treat all the other robots as moving obstacles, which can be easily handled by the proposed approach.

The proposed neural network is a stable system. The neural activity x_i is bounded in the finite interval $[-D, B]$. In addition, the stability and convergence of the proposed shunting neural network model can also be rigorously proved using a Lyapunov stability theory (for details, see [33]).

III. Simulation Studies

The proposed neural network approach is capable of planning complete coverage paths for cleaning robots, autonomously without any human intervention. In this section, the proposed approach is first applied to a typical side-shift template situation that is commonly used in most template based models. Then, CCPP in a dynamic environment with sudden changes is studied. After that, this model is applied to a cleaning robot in an unstructured environment. Finally, CCPP of cooperative sweeping by a multirobot system is presented.

A. CCPP in a Side-Shift Template Situation

In template-based approaches such as [17], 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 match any predefined templates. In the proposed neural network model, no templates are needed. The proposed model is applied to a famous template situation, side-shift template shown in Fig. 2(a), where the robot has cleaned all the left areas and arrives at Location P. This is normally called a *deadlock* situation, where all the neighboring areas of Location P are either cleaned areas or obstacles; at Location P, the left and behind areas have been cleaned, and the front and right areas are obstacles. Thus the neural activities at its neighboring areas are *not* larger than the neural activity at P [see Fig. 2(b)]. That is why pure artificial potential based approaches cannot deal with deadlock situations. To resolve such a deadlock problem, the robot should be able to move back passing some *cleaned* areas in order to reach the unclean areas on the right side. In the proposed model, because of the neural activity propagation among neurons, the unclean areas are able to attract the robot, which is in the same way as the conventional path planning [30]–[33].

In the simulation, the neural network has 14×14 topologically organized neurons, where all the neural activities are initialized to zero. The model parameters are set as $A = 20$, $B = 1$, and $D = 1$ for the shunting equation; $\mu = 0.7$ and $r_0 = 2$ for the lateral connections; and $E = 50$ for the external inputs. The robot starts to sweep from the left side [see Fig. 2(a)]. When the robot arrives at Location P, the activity landscape of the neural network is shown in Fig. 2(b). It shows that the unclean areas have the largest neural activity due to their very large external excitatory inputs; the obstacle areas have the smallest neural activity due to their very large inhibitory inputs; the neural activities at the cleaned areas are decreasing to zero due to the passive decay term in the shunting equation in (3). It is obvious that when the robot arrives at P, the neural activity at P is larger than all the activities at its neighboring locations. Thus, the robot cannot move away from P at this time. Because of the passive decay term in (3), the neural activity at P decays to zero exponentially. Meanwhile, because of the neural activity propagation among neurons, the positive activities from the unclean areas are propagating to all directions, including toward the current robot location P. The unclean areas have very large external inputs and keep the largest neural activity. Therefore, in a very short time, the activity at P will become smaller than that at the location right below P, and the robot will start to move down toward the unclean locations. Fig. 2(c) shows the activity landscape when the activity of a neighboring location is larger than that at P and the robot starts to escape from the deadlock location P. The robot keeps moving down toward the unclean locations that have the largest activity. After the robot arrives at Location Q, the neural activity landscape is shown in Fig. 2(d), where activity at Q starts to decrease. The robot path is shown in Fig. 2(a), where the black circles represent the areas that are passed twice by the robot. It shows that the robot is able to move backward from P, pass through the cleaned areas, and reach the nearest unclean location Q, without any collisions.

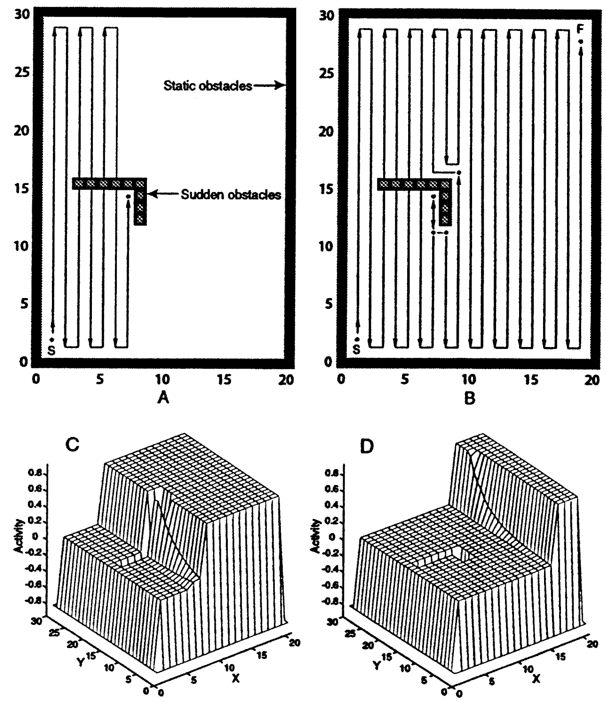

Fig. 3. Complete coverage path planning in a dynamic environment with obstacles suddenly appeared in front of the robot. (a) Robot path right after the L-shaped obstacles are placed. (b) Whole collision-free robot path. (c) Activity landscape right after the L-shaped obstacles appear. (d) Activity landscape when the robot arrives at Location (16,26).

B. CCPP in a Dynamic Environment With Sudden Changes

The proposed neural network approach is capable of generating complete coverage paths for cleaning robots in a dynamic environment, even with obstacles suddenly placed in front of the robot. The neural network has 20×30 neurons, with initial neural activities at zero. The model parameters are chosen as the same as in the previous case. In an empty room shown in Fig. 3(a), the robot starts to sweep from Location

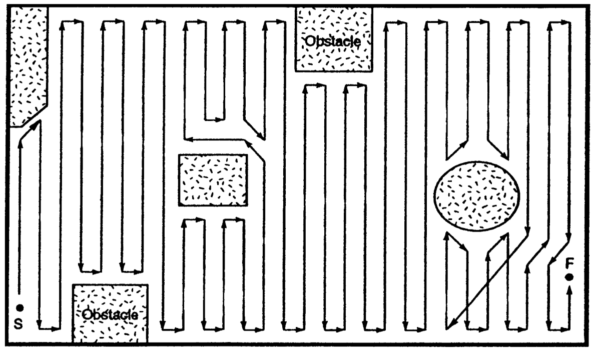

Fig. 4. Complete coverage path planning in an unstructured environment.

(1,1). When the robot arrives at (7,14), however, a set of L-shaped obstacles suddenly appear in front of the robot [see Fig. 3(a)]. The activity landscape of the neural network right after the obstacles are placed is shown in Fig. 3(c). It shows that the neural activities at the areas with the suddenly placed obstacles immediately become very large negative values. The cleaning robot cannot move forward due to the suddenly added obstacles. Such a situation is similar to the side-shift template in the previous case. The robot has to move back through several cleaned areas, then passes around the obstacles, reaches an unclean area on its right side, and finally cleans all the unclean workspace with obstacle avoidance [see Fig. 3(b)]. The neural activity landscape when the robot arrives at (16,26) is shown in Fig. 3(d).

C. CCPP in an Unstructured Environment

The proposed model is then applied to a complicated CCPP case, where there are unstructured obstacles with arbitrary shapes in the workspace (see Fig. 4). In the discretization of workspace, an area partially occupied with an obstacle is assumed as an obstacle area. The neural network has 28×25 neurons, with zero initial neural activities and the same model parameters as in the previous cases. In the workspace, there are five sets of obstacles with different shapes and sizes. The planned robot path is shown in Fig. 4 by solid lines with arrows representing the moving directions. It shows that the cleaning robot is able to autonomously sweep the entire workspace from left to right with obstacle avoidance.

D. Cooperative Sweeping of Multi-Robots

The proposed neural network approach is capable of planning complete coverage paths for multiple cleaning robots, autonomously without any human intervention. All the robots share the same dynamic environmental information, whereas each robot treats the other robots as moving obstacles. The neural activity landscape of each robot is able to make the robot follow a reasonable path and to cooperate with the other robots.

To illustrate the cooperative sweeping of a multi-robot system, the proposed model is first applied to two cleaning robots in a simple indoor environment (see Fig. 5). In the simulation, there are two neural networks for the two robots, respectively. Each neural network has 30×30 neurons with zero initial neural activities. The model parameters for both neural networks are chosen as the same as in the previous cases. In the case shown in Fig. 5, Robot 1 represented by a filled circle starts to sweep from the lower-left corner S1 at Location (1,1), and Robot 2 represented by an empty circle sweeps from the upper-right corner S2 at (29, 29). The real-time robot cleaning paths are shown in Fig. 5, where the solid line represents the path of Robot 1, whereas the dashed line shows that of Robot 2. After Robot 1 sweeps four columns, it encounters a wall and starts to clean in a narrow area like an aisle. However, Robot 2 meets a wall after it has swept six columns. Obviously, Robot

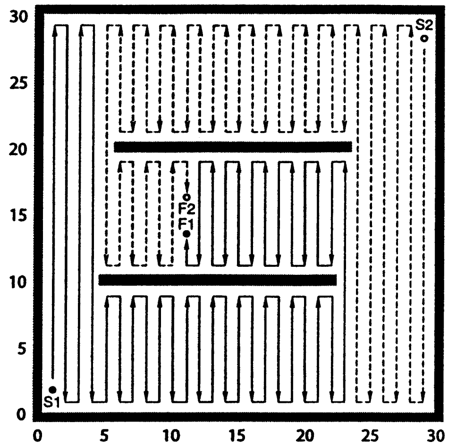

Fig. 5. Complete coverage path planning of two robots in an indoor environment.

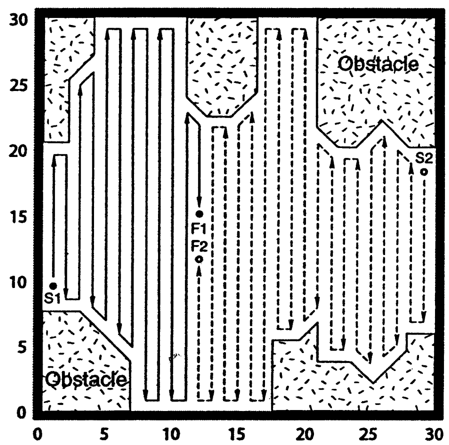

Fig. 6. Complete coverage path planning of two robots in an unstructured environment.

1 reaches the central corridor earlier than Robot 2. When Robot 1 arrives at F1 (11, 12), it meets with Robot 2 at F2 (11,13); thus, the whole space is cleaned. It shows that Robot 1 does more work than Robot 2 in the middle areas. These two robots first sweep their own areas respectively, and then, they clean the public aisle areas cooperatively.

The proposed model is then applied to a complicated case of cooperative sweeping in an unstructured environment, where there exist five sets of unstructured obstacles with different shape and size in the workspace shown in Fig. 6. The neural networks are the same as in the previous case with the same model parameters. In the simulation, Robot 1 starts to move from the lower-left corner S1 at (1,8), whereas Robot 2 sweeps from the upper-right corner S2 at (29, 20). The planned real-time collision-free robot paths are shown in Fig. 6. It shows that these two cleaning robots are able to autonomously sweep the whole workspace along the curves of the irregularly shaped obstacles with obstacle avoidance.

IV. CONCLUSION

In this paper, a biologically inspired neural network approach to real-time complete coverage path planning of cleaning robots is proposed. The developed approach is capable of autonomously planning collision-free paths for cleaning robots in a nonstationary environment. Some points are worth mention about the proposed neural network approach.

- The proposed model is not very sensitive to model parameters. There are only few model parameters, which can be chosen in a wide range (for a detailed discussion, see [33]).
- There are only local connections among neurons. If the workspace is a rectangle, the number of neurons required is equal to $M = N_x \times N_y$, where N_x and N_y are the discretized size of the Cartesian workspace. Each neuron has at most eight local connections. Thus, the total neural connections are not more than $8M$. If the workspace is an $N \times N$ square, N^2 neurons are needed, and there are at most $8N^2$ neural connections. Thus, the computational complexity of the proposed algorithm is $O(N^2)$, i.e., the complexity is squarely proportional to the degree of discretization (resolution).
- The model algorithm is computationally simple. The robot path is generated through dynamic neural activity landscape and the previous robot location without explicitly optimizing any *global* cost functions (there is *local* search for next robot location among its eight neighboring neurons), without any *prior* knowledge of the dynamic environment (this model requires the current knowledge of the environment, which is assumed to be *completely known* through sensor measurements. The multisensor fusion and localization of a robot are beyond the scope of this paper and are assumed to be perfect) and *without* any learning procedures.
- This model will not be trapped in deadlock situations, which are defined as areas whose neighboring areas are either cleaned or obstacle areas. The remain unclean areas can globally attract the robot in the whole workspace through neural activity propagation. Thus the cleaning robot is able to escape from deadlock situations.
- This model can deal with changing environments, even with sudden environmental changes, such as suddenly adding or removing obstacles.
- This model is biologically plausible. The neural activity is a continuous analog signal and has both upper and lower bounds. In addition, the continuous activity prevents the possible oscillations related to parallel dynamics of discrete neurons [37]. Thus, it has potential VLSI implementation for parallel computation.

Acknowledgment

The authors would like to thank the editor-in-chief, the associate editor, and the anonymous reviewers for their valuable comments.

References

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

[2] 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. Robotics Automation*, Seoul, Korea, 2001, pp. 699–704.

[3] S. Hert, S. Tiwari, and V. Lumelsky, "A terrain-covering algorithm for an auv," *Autonomous Robots*, vol. 3, no. 2–3, pp. 91–119, 1996.

[4] D. W. Gage, "Randomized search strategies with imperfect sensors," in *Proc. SPIE, Mobile Robots VIII—Int. Soc. Optical Engineering*, Boston, MA, 1994, pp. 270–279.

[5] H. Najjaran and N. Kircanski, "Path planning for a terrain scanner robot," in *Proc. 31st Int. Symp. Robotics*, Montreal, QC, Canada, 2000, pp. 132–137.

[6] Z. L. Cao, Y. Y. Huang, and E. L. Hall, "Region filling operations with random obstacle avoidance for mobile robots," *J. Robotic Syst.*, vol. 5, no. 2, pp. 87–102, 1988.

[7] M. Ollis and A. Stentz, "Vision-based perception for an automated harvester," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robot Syst.*, Grenoble, France, 1997, pp. 1838–1844.

[8] ——, "First results in vision-based crop line tracking," in *Proc. IEEE Int. Conf. Robotics Automation*, Minneapolis, MN, 1996, pp. 951–956.

[9] 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. Automation Robotics Construction*, Brighton, U.K., 1994, pp. 617–623.

[10] A. Pirzadeh and W. Snyder, "A unified solution to coverage and search in explored and unexplored terrains using indirect control," in *Proc. IEEE Int. Conf. Robotics Automation*, Raleigh, NC, 1990, pp. 2113–2119.

[11] H. P. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proc. IEEE Int. Conf. Robotics Automation*, St. Louis, MO, 1985, pp. 116–121.

[12] A. Zelinsky, R. A. Jarvis, J. C. Byarne, and S. Yuta, "Planning paths of complete coverage of an unstructured environment by a mobile robot," in *Proc. IEEE Int. Conf. Robotics Automation*, Tokyo, Japan, 1993, pp. 533–538.

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

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

[15] E. U. Acar and H. Choset, "Critical point sensing in unknown environments," in *Proc. IEEE Int. Conf. Robotics Automation*, San Francisco, CA, 2000, pp. 3803–3810.

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

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

[18] R. N. De Carvalho, H. A. Vidal, P. Vieira, and M. I. Ribeiro, "Complete coverage path planning and guidance for cleaning robots," in *Proc. IEEE Int. Symp. Ind. Electron.*, Guimaraes, Portugal, 1997, pp. 677–682.

[19] 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 Networks*, Anchorage, AK, 1998, pp. 2151–2156.

[20] Y. Fu and S. Y. T. Lang, "Fuzzy logic based mobile robot area filling with vision system for indoor environments," in *Proc. IEEE Int. Symp. Computational Intelligence Robotics Automation*, Monterey, CA, 1999, pp. 326–331.

[21] S. Y. T. Lang and B. Y. Chee, "Coordination of behaviors for mobile robot floor cleaning," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Syst.*, Victoria, BC, Canada, 1998, pp. 1236–1241.

[22] E. M. Arkin and R. Hassin, "Approximation algorithms for the geometric covering salesman problem," *Discrete Appl. Math.*, vol. 55, pp. 197–218, 1994.

[23] R. A. Russell, "Mobile robot guidance using a short-lived heat trail," *Robotica*, vol. 11, pp. 427–431, 1993.

[24] C. L. Chang and J. M. Stentz, "A region-filling mobile robot," in *Proc. 8th Int. Conf. CAD/CAM*, Metz, France, 1992, pp. 1128–1141.

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

[26] D. Kurabayashi, J. Ota, T. Arai, and E. Yoshida, "Cooperative sweeping by multiple mobile robots," in *Proc. IEEE Int. Conf. Robotics Automation*, Minneapolis, MN, 1996, pp. 1744–1749.

[27] D. Kurabayashi, J. Ota, T. Arai, S. Ichikawa, S. Koga, H. Asama, and I. Endo, "Cooperative sweeping by multiple mobile robots with relocating portable obstacles," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Syst.*, Osaka, Japan, 1996, pp. 1472–1477.

[28] I. M. Rekleitis, D. Dudek, and E. E. Milios, "Multi-robot exploration of an unknown environment, efficiently reducing the odometry error," in *Proc. 15th Int. Joint Conf. Artificial Intelligence*, Nagoya, Japan, 2000, pp. 1340–1345.

[29] I. A. Wagner and A. M. Bruckstein, "Distributed covering by ant-robots using evaporating traces," *IEEE Trans. Robotics Automat.*, vol. 15, pp. 918–933, Oct. 1999.

[30] S. X. Yang and M. Meng, "An efficient neural network approach to dynamic robot motion planning," *Neural Networks*, vol. 13, no. 2, pp. 143–148, 2000.

[31] ——, "An efficient neural network method for real-time motion planning with safety consideration," *Robotics Auton. Syst.*, vol. 32, no. 2–3, pp. 115–128, 2000.

[32] ——, "Real-time collision-free path planning of robot manipulators using neural network approaches," *Auton. Robots*, vol. 9, no. 1, pp. 27–39, 2000.

[33] ——, "Neural network approaches to dynamic collision-free robot trajectory generation," *IEEE Trans. Syst., Man, Cybern.*, vol. 31, pp. 302–318, June 2001.

[34] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *J. Physiol. Lond.*, vol. 117, pp. 500–544, 1952.
[35] S. Grossberg, "Nonlinear neural networks: principles, mechanisms, and architecture," *Neural Networks*, vol. 1, pp. 17–61, 1988.
[36] A. Zelinsky, "Using path transforms to guide the search for findpath in 2D," *Int. J. Robotics Res.*, vol. 13, no. 4, pp. 315–325, 1994.
[37] C. M. Marcus, F. R. Waugh, and R. M. Westervelt, "Associative memory in an analog iterated-map neural network," *Phys. Rev. A*, vol. 41, no. 6, pp. 3355–3364, 1990.

An Empirical Risk Functional to Improve Learning in a Neuro-Fuzzy Classifier

Giovanna Castellano, Anna M. Fanelli, *Member, IEEE*, and Corrado Mencar

***Abstract*—The paper proposes a new Empirical Risk Functional as cost function for training neuro-fuzzy classifiers. This cost function, called Approximate Differentiable Empirical Risk Functional (ADERF), provides a differentiable approximation of the misclassification rate so that the Empirical Risk Minimization Principle formulated in Vapnik's Statistical Learning Theory can be applied. Also, based on the proposed ADERF, a learning algorithm is formulated. Experimental results on a number of benchmark classification tasks are provided and comparison to alternative approaches given.**

***Index Terms*—Classification error, empirical risk functional, gradient-based learning, misclassification rate, neuro-fuzzy classifier.**

I. Introduction

Fuzzy set theory is widely considered an appropriate framework for pattern classification because of the inherent fuzziness involved in the definition of a class [1], [2]. Unlike conventional crisp classification techniques, which assume that a pattern belongs to only one class, fuzzy classifiers assign a pattern with a distributed membership value to each class, yielding soft class partitions (labels). This information is useful not only to evaluate the quality of the classification results, but also to treat separately patterns that are classified poorly. However, most classification systems require hard labels for objects being classified, thus fuzzy classifier design almost always means arriving at a hard classifier. In this case the fuzzy class labels are "defuzzified" and the fuzzy classifier becomes a hard classifier but uses the idea of fuzziness in the model.

There are numerous studies discussing the design of fuzzy classifiers with learning capability, among which are neuro-fuzzy models [3]–[7] and fuzzy systems constructed using genetic algorithms [8]–[10]. Learning in neuro-fuzzy systems is usually based on the minimization of a cost function. Commonly, the Mean Squared Error (MSE) is used, since it can be shown to approximate the posterior class probabilities in a neural classifier, but different choices of cost functions can be made corresponding to different assumptions about the statistical properties of data [11].

Manuscript received March 26, 2002; revised October 3, 2002. This paper was recommended by Associate Editor H. Ishibuchi.

The authors are with the Department of Informatics, University of Bari, Bari, Italy (e-mail: castellano@di.uniba.it; fanelli@di.uniba.it; mencar@di.uniba.it).

Digital Object Identifier 10.1109/TSMCB.2003.811291

An attractive approach, however, is to train neuro-fuzzy classifiers according to the Empirical Risk Minimization (ERM) principle [12], [13]. This states that, given a Risk Functional (RF), which is a theoretical measure of the performance of the classifier (classification error), minimization of RF can be achieved by minimization of the Empirical Risk Functional, defined as the sample mean of RF computed on the training data (misclassification rate). The problem associated with the use of misclassification rate as ERF is that it can not be minimized via gradient descent learning, since it is not differentiable.

In this paper, we formulate an Empirical Risk Functional as cost function to be minimized during the training procedure of a family of neuro fuzzy classifiers. This cost function, which we call Approximate Differentiable Empirical Risk Functional (ADERF), provides a differentiable approximation of the misclassification rate so that it can be directly minimized through a steepest-descent procedure.

The paper is organized as follows. In Section II a family of neuro-fuzzy classifiers is introduced. Section III formulates the proposed ADERF and the related learning algorithm. Simulation results are given in Section IV. Finally, Section V concludes the paper.

II. Family of Neuro-Fuzzy Classifiers

Consider an n-dimensional classification problem for which N patterns $\mathbf{x} \in X \subseteq \Re^n$, $\mathbf{x} = (x_1, \ldots, x_n)$ are given from m disjoint classes $C_1, \ldots, C_m$, i.e., a *training set* of N examples

$$T = \left\{ \left(\mathbf{x}^{(p)}, \mathbf{d}^{(p)}\right) \middle| \mathbf{x}^{(p)} \in X, \right.$$
$$\left. \mathbf{d}^{(p)} = \text{class}\left(\mathbf{x}^{(p)}\right), p = 1 \ldots N \right\}$$

is available. The task of a pattern classifier is to assign a given pattern $\mathbf{x}$ to one of the m classes based on its feature values. Thus, a classification task can be represented as a mapping

$$\text{class} : X \to E$$

where

$$E = \left\{ \mathbf{e}_j \,|\, \mathbf{e}_j = \left(\underbrace{0, 0, \ldots 0}_{j-1}, 1, \underbrace{0, \ldots, 0}_{m-j} \right) \right\} \subset \{0, 1\}^m$$

so that: $\mathbf{x}$ belongs to class $C_l \Leftrightarrow \text{class}(\mathbf{x}) = \mathbf{e}_l$.

To approximate the mapping $\text{class}(\cdot)$, a fuzzy inference model can be used that represents the pattern feature values by fuzzy sets and performs the classification task by a set of linguistic rules. The type of fuzzy model adopted in this work is based on rules of the following form:

$$\begin{aligned} &\text{IF}\,(x_1 \text{ IS } A_1^r) \text{ AND} \cdots \text{AND}\,(x_n \text{ IS } A_n^r) \text{ THEN} \\ &(\mathbf{x} \in C_1 \text{ with degree } v_{r1}) \text{ AND } \ldots \\ &\text{AND } (\mathbf{x} \in C_m \text{ with degree } v_{rm}) \end{aligned} \tag{1}$$

where r is the rule index, A_i^r, $i = 1 \ldots n$ are fuzzy sets defined on the input variables and v_{rj}, $j = 1 \ldots m$ are fuzzy singletons representing the degree to which a pattern $\mathbf{x}$ belongs to class C_j, $j = 1 \ldots m$. Gaussian membership functions are employed for input fuzzy sets A_i^r

$$\mu_{A_i^r}(x_i) := \exp\left(-\frac{(x_i - w_{ri})^2}{2\sigma_{ri}^2}\right) \tag{2}$$

where w_{ri} and σ_{ri} are the center and the width of the function.