Proceedings of the 2006 IEEE International Conference on Robotics and Automation
Orlando, Florida - May 2006

Complete Coverage Control for Nonholonomic Mobile Robots in Dynamic Environments

Yi Guo and Mohanakrishnan Balakrishnan
Department of Electrical and Computer Engineering
Stevens Institute of Technology
Hoboken, NJ 07030
Email: yguo1@stevens.edu

***Abstract*— We study the problem of generating continuous steering control for robots to completely cover a bounded region over a finite time. First, we pack the area by disks of minimum number, and then a neural network based path planning method is adopted to generate complete coverage paths avoiding collisions with stationary and moving objects. Smooth trajectory is developed using parametric polynomial approximation, and continuous steering control is designed to track the trajectory exactly. The algorithm works for car-like robots which have nonholonomic motion constraints, and the control inputs are represented by analytic functions of trajectory parameters. The design is finally extended to cooperative sweeping of multiple robots. Satisfactory performances are observed in simulations.**

***Index Terms*— Complete coverage, path planning, motion planning, nonholonomic systems.**

I. Introduction

Complete coverage path planning has received increasing research attention during the past decade. It requires the robot's end-effector or sensors to cover a complete area. Applications include cleaning robots ([6]), painter robots, lawn mowers, de-mining, land mine detection, search and rescue, security patrolling ([9], [11]), and sensor networks ([1], [5]). Comparing to the large amount of work in conventional path planning between a start and a goal, there are few mature methods for dynamic environments, and the performances of the methods remain hard to measure. Particularly, there's little work reported that studies feasible trajectory generation for a car-like robot. We consider the problem of generating feasible trajectories for mobile robots to completely cover a bounded region, and design steering control input to drive the robot on the paths.

There are various methods used for complete coverage path planning, to name a few, distance transform ([16]), cellular decomposition ([4]), spanning tree ([8]), template based model ([13]), and neural networks ([15]). Distance transform method uses a distance wave front to flow around obstacles and eventually through all free space in the environment. To achieve the complete coverage path planning, the robot follows the path of steepest ascent instead of conventional steepest descent to reach a goal. In [4], the author proposed boustrophedon cellular decomposition approach by breaking down the workspace, and allowing the robot to cover each cell like the ox. While many results work in environments with stationary obstacles, the authors of [15] presented a novel neural network approach for nonstationary environments. Complete coverage paths are generated from a dynamic activity landscape of the neural network and the previous robot location. The dynamics of each neuron in the topologically organized neural network is characterized by a shunting equation derived from Hodgkin and Huxley's membrance equation ([12]).

There is little or none consideration of physical robot's kinematic motion constraints in complete coverage path planning of existing literature. Since the complete coverage paths have frequent turns, from the performance point of view, it's necessary to design continuous steering control to replace the zig-zag motion. In this paper, we plan complete coverage paths based on neural network method, and generate smooth trajectories using parametric polynomials. Then using the concept of differential flatness, we design continuous steering control input for a mobile robot to exactly track the trajectory generated. We also extend the method to cooperative coverage of multiple robots. Simulation results demonstrate satisfactory performances of four robots's cooperative sweeping in a bounded region with both stationary and moving objects.

II. Problem Statement

Consider a car-like mobile robot shown in Fig. 1. The front wheels of the mobile robot are steering wheels and the rear wheels are driving wheels with a fixed forward orientation. The kinematic model of the mobile robot can be written as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ \tan\phi/l \\ 0 \end{bmatrix} \rho u_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_2 \tag{1}$$

where $q = [x, y, \theta, \phi]^T$ is the system state, (x, y) represents the Cartesian coordinates of the middle point of the rear wheel axle, θ is the orientation of the robot body with respect to the X-axis, ϕ is the steering angle; l is the distance between the front and rear wheel-axle centers, ρ is the radius of rear driving wheel; u_1 is the angular velocity of the driving wheels, and u_2 is the steering velocity of the front wheels. $\phi \in (-\pi/2, \pi/2)$ due to the structure constraint of the robot.

For the robot described as above, the complete coverage control problem is defined as:

Given a connected and bounded region with stationary or moving objects, find continuous steering control under which

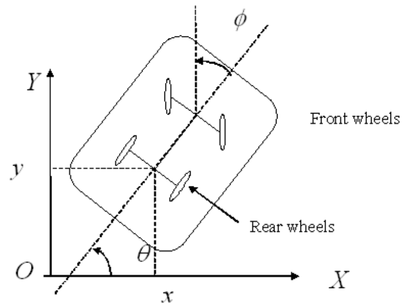

Fig. 1. A car-like robot

the robot moves collision-free and completely covers the region over a finite time.

We make the following assumptions:

Assumption 1: *The robot's coverage range (by its end-effector or detection sensors) is described by a circle of radius* R_c.

Assumption 2: *The positions of the stationary obstacles are known, which are represented by coordinates in a pre-existing map; and the positions and the speeds of moving obstacles are detected by the robot's onboard sensors so that an updated map with the positions of moving obstacles available real time.*

III. Complete Coverage Path Planning

In [10], the authors proposed a path planning algorithm for complete region coverage. It first packs the bounded region with disks of radius R_c. It was shown that the disk placement pattern in Figure 2 has a minimum number of disks to cover a rectangle. Since the radius of the disk is the same as the coverage range of robot's sensors, complete coverage with minimum repeated coverage can be achieved by visiting every center of the disks. Complete coverage path planning is then to find the sequence to visit the centers. A path planning algorithm was proposed to find a complete coverage path in [10]. However, the algorithms works only in environments without obstacles. In [15], a neural network approach was developed for complete coverage path planning in a nonstationary environment. We modify the algorithm corresponding to our data structure and generate collision-free complete coverage paths.

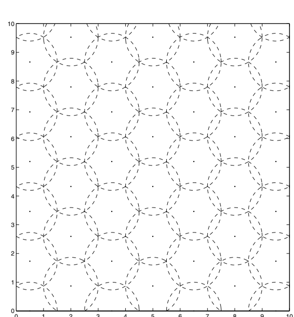

Fig. 2. Covering a rectangle using a minimum number of disks

The basic idea of the neural network approach is to generate a dynamic landscape for the neural activities, such that through neural activity propagation, the uncleaned areas globally attract the robot in the entire state space, and the obstacles locally repel the robot to avoid collisions. The dynamics of each neuron in the topologically organized neural network is characterized by a shunting equation derived from Hodgkin and Huxley's membrane equation ([12]). The robot path is autonomously generated from the activity landscape of the neural network and the previous location. The neural network model is expressed topologically in a discretized workspace. The location of the neuron in the state space of the neural network uniquely represents an area. 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. The dynamics of the neuron in the neural network is characterized by the following equation:

$$\begin{aligned}\dot{x}_i &= -Ax_i + (B - x_i)\left([I_i]^+ + \sum_{j=1}^{k} w_{ij}[x_j]^+\right) \\ &\quad -(D + x_i)[I_i]^- \end{aligned} \tag{2}$$

where k is the number of neural connections of the ith neuron to its neighboring neurons within the receptive field. Six neighbors of a point (neuron) is shown in Fig. 3.

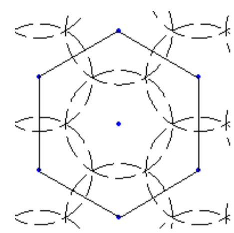

Fig. 3. Six neighbors of a neuron

The external input I_i to the ith neuron is defined as

$$I_i = \begin{cases} E & \text{if it is an unclean area} \\ -E & \text{if it is an obstacle area} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $E >> B$ is a very large positive constant. The terms $[I_i]^+ + \sum_{j=1}^{k} w_{ij}[x_j]^+$ and $[I_i]^-$ are the excitatory and inhibitory inputs respectively. Function $[a]^+$ is a linear threshold function defined as $[a]^+ = \max\{a, 0\}$, and $[a]^- = \max\{-a, 0\}$. The connection weight w_{ij} between the ith and the jth neurons is 1 if they're neighbors or 0 if they are not neighbors.

To make the path having less navigation turns, for a current robot location p_c, we select the next point p_n within the uncleaned neighbors according to:

$$x_n = \max\{x_j + (1 - \frac{\Delta\theta_j}{\pi}),\ j = 1, 2, \ldots, k\} \tag{4}$$

where k is the number of neighboring neurons, x_j is the neuron activity of the jth neuron, $\Delta\theta_j$ is the absolute angle change between the current and next moving directions, i.e., $\Delta\theta_j = 0$ if going straight, and $\Delta\theta_j = \pi$ if going backward.

After the robot reaches its next position, the next position becomes a new current position. Because of the excitatory neural connections in (2), the neural activity propagates to the entire state space so that the complete coverage is achieved.

A complete coverage path in an environment with stationary obstacles is shown in Figure 4. It is shown that the path completely covers the bounded region without covering a point twice. Note that in a trap situation, that is, there is no uncleaned neighbors, the neighbor's neighbors become the neighbors of the neuron, so that uncleaned area can be continuously searched. This can be seen from the right bottom and top middle parts of the figure. The algorithm works for moving obstacles if the speed of the obstacles are known, since the landscape activities of the environment are updated dynamically in the algorithm.

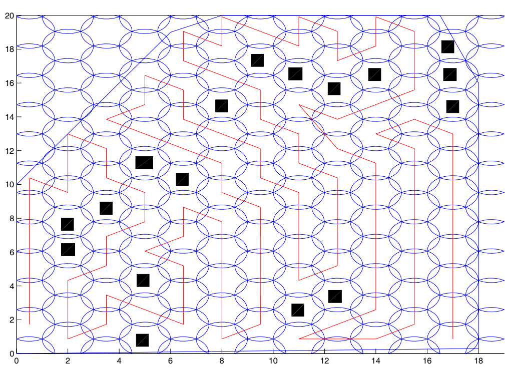

Fig. 4. Complete coverage paths, the dark rectangles are stationary obstacles

IV. Complete Coverage Control Design

A. Smooth Trajectory Design

After the complete coverage paths are generated, we need to develop feasible trajectories so that continuous steering control can be designed. The method we use is inspired by the method presented in [14] for generating smooth irregular curves through an arbitrary set of points in a plane.

We consider a local subset of six points to define sequentially a local polynomial approximation to the curve between the third and fourth points in the local subset. After each step, a new point is added and the first point is deleted. It iteratively generates polynomial segments between adjacent points and there is no slope discontinuity. We add two pseudo-points at the beginning of the path so that the first point in the path becomes the third point in the process. Define a cumulative polygon approximation to arc length for each point in the data subset as

$$\begin{aligned} s_1 &= 0 \\ s_k &= s_{k-1} + [(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2]^{1/2}, \\ & \quad k = 1, 2, \ldots, 6 \end{aligned} \tag{5}$$

So s is single-valued and monotonically increasing. Use cubic polynomials to generate three curves: the left, which uses the first 4 points in the data subset, the middle, which uses the middle 4 points, and the right, which uses the last 4 points in the 6-point data subset. Since boundary conditions are well-defined as:

$$\begin{aligned} (s,x,y)_L &= \{0, x_1, y_1; s_2, x_2, y_2; s_3, x_3, y_3; s_4, x_4, y_4\}, \\ (s,x,y)_M &= \{s_2, x_2, y_2; s_3, x_3, y_3; s_4, x_4, y_4; s_5, x_5, y_5\}, \\ (s,x,y)_R &= \{s_3, x_3, y_3; s_4, x_4, y_4; s_5, x_5, y_5; s_6, x_6, y_6\}, \end{aligned}$$

the cubic polynomials are generated as

$$\begin{aligned} x &= a_1 + a_2 s + a_3 s^2 + a_4 s^3, \\ y &= b_1 + b_2 s + b_3 s^2 + b_4 s^3, \end{aligned} \tag{6}$$

where the constant coefficient matrices can be calculated as

$$\begin{aligned} A &= S^{-1} X \\ B &= S^{-1} Y \end{aligned} \tag{7}$$

where S and X are appropriate matrices defined from the boundary conditions. That is, for each of the *Left, Middle, Right* polynomials, we have

$$\begin{aligned} A_L &= [a_{1L}\ a_{2L}\ a_{3L}\ a_{4L}]^T \\ B_L &= [b_{1L}\ b_{2L}\ b_{3L}\ b_{4L}]^T \\ \text{Replacing} &\quad L \text{ by } M, R \text{ to get } A_M, B_M, A_R, B_R \\ X_L &= [x_1\ x_2\ x_3\ x_4]^T \\ X_M &= [x_2\ x_3\ x_4\ x_5]^T \\ X_R &= [x_3\ x_4\ x_5\ x_6]^T \\ \text{Replacing} &\quad X \text{ by } Y \text{ to get } Y_L, Y_M, Y_R \\ S_L &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & s_2 & s_2^2 & s_2^3 \\ 1 & s_3 & s_3^2 & s_3^3 \\ 1 & s_4 & s_4^2 & s_4^3 \end{bmatrix} \\ S_M &= \begin{bmatrix} 1 & s_2 & s_2^2 & s_2^3 \\ 1 & s_3 & s_3^2 & s_3^3 \\ 1 & s_4 & s_4^2 & s_4^3 \\ 1 & s_5 & s_5^2 & s_5^3 \end{bmatrix} \\ S_R &= \begin{bmatrix} 1 & s_3 & s_3^2 & s_3^3 \\ 1 & s_4 & s_4^2 & s_4^3 \\ 1 & s_5 & s_5^2 & s_5^3 \\ 1 & s_6 & s_6^2 & s_6^3 \end{bmatrix} \end{aligned} \tag{8}$$

The approximation for the unknown curve between the third and fourth points of the subset makes use of the preliminary cubic polynomials to obtain slopes and the rate change of slope. That is, for the third point,

$$\left.\frac{dx}{ds}\right|_{s_3} = \frac{1}{2}\left[\left.\frac{dx_L}{ds}\right|_{s_3} + \left.\frac{dx_M}{ds}\right|_{s_3}\right], \tag{9}$$

$$\left.\frac{dy}{ds}\right|_{s_3} = \frac{1}{2}\left[\left.\frac{dy_L}{ds}\right|_{s_3} + \left.\frac{dy_M}{ds}\right|_{s_3}\right], \tag{10}$$

$$\left.\frac{d^2x}{ds^2}\right|_{s_3} = \frac{1}{2}\left[\left.\frac{d^2x_L}{ds}\right|_{s_3} + \left.\frac{d^2x_M}{ds^2}\right|_{s_3}\right], \tag{11}$$

$$\left.\frac{d^2y}{ds^2}\right|_{s_3} = \frac{1}{2}\left[\left.\frac{d^2y_L}{ds^2}\right|_{s_3} + \left.\frac{d^2y_M}{ds^2}\right|_{s_3}\right]. \tag{12}$$

Replacing the left and middle polynomials with the middle and right ones respectively, we obtain similar equations for the fourth point. The slope and rate change of slope at the third and fourth points are used as boundary conditions for the curve generated between these two points. Therefore, we obtain a fifth order polynomial between the third and fourth points:

$$\begin{aligned} x &= c_1 + c_2 s + c_3 s^2 + c_4 s^3 + c_5 s^4 + c_6 s^5, \\ y &= d_1 + d_2 s + d_3 s^2 + d_4 s^3 + d_5 s^4 + d_6 s^5. \end{aligned} \quad (13)$$

The coefficients are obtained by boundary conditions, *i.e.*,

$$\begin{aligned} C &= S^{-1} X \\ D &= S^{-1} Y \end{aligned} \quad (14)$$

where

$$\begin{aligned} C &= [c_1\ c_2\ c_3\ c_4\ c_5\ c_6]^T \\ D &= [d_1\ d_2\ d_3\ d_4\ d_5\ d_6]^T \\ X &= \left[x_3\ x_4\ \left.\frac{dx}{ds}\right|_{s_3}\ \left.\frac{dx}{ds}\right|_{s_4}\ \left.\frac{d^2x}{ds^2}\right|_{s_3}\ \left.\frac{d^2x}{ds^2}\right|_{s_4} \right]^T \\ Y &= \left[y_3\ y_4\ \left.\frac{dy}{ds}\right|_{s_3}\ \left.\frac{dy}{ds}\right|_{s_4}\ \left.\frac{d^2y}{ds^2}\right|_{s_3}\ \left.\frac{d^2y}{ds^2}\right|_{s_4} \right]^T \\ S &= \begin{bmatrix} 1 & s_3 & s_3^2 & s_3^3 & s_3^4 & s_3^5 \\ 1 & s_4 & s_4^2 & s_4^3 & s_4^4 & s_4^5 \\ 0 & 1 & 2s_3 & 3s_3^2 & 4s_3^3 & 5s_3^4 \\ 0 & 1 & 2s_4 & 3s_4^2 & 4s_4^3 & 5s_4^4 \\ 0 & 0 & 2 & 6s_3 & 12s_3^2 & 20s_3^3 \\ 0 & 0 & 2 & 6s_4 & 12s_4^2 & 20s_4^3 \end{bmatrix} \end{aligned} \quad (15)$$

Figure 5 shows the process, that is, by generating the left, middle, right curves of 6-point data set, a smooth curve between the third and fourth points is generated.

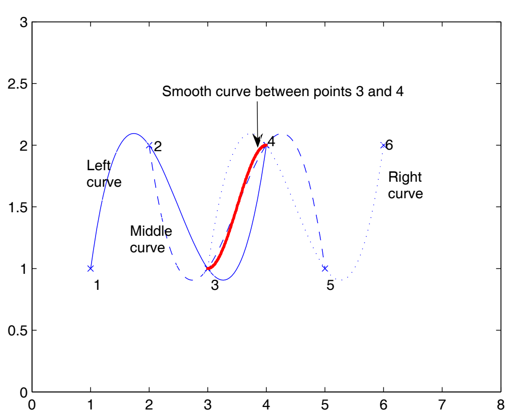

Fig. 5. Generating polynomial approximation of 6-point data set

Repeating the above process by adding one point at the end and deleting the first point in the subset, the middle and right polynomials becomes the left and middle ones respectively for the next subset, so the resulting polynomials between the fourth and fifth points join smoothly with the previously determined curve between the third and fourth points. In another words, the second-order derivatives at any points of the trajectory are continuous.

The generated smooth curve on the complete coverage path is shown in Figure 6. Note that this step of smooth trajectory design id necessary for steering control design of the mobile robot.

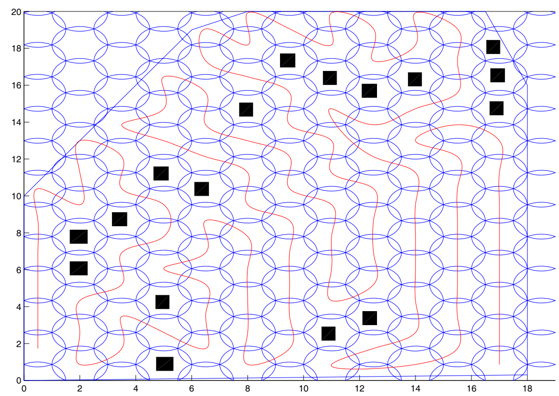

Fig. 6. Smooth trajectory

B. Steering Control Design

It is well known that car-like robots have nonholonomic constraints in their kinematics and movement from one configuration to the other is done through steering ([2], [3]). We use differential flatness for the design of steering control. Recall that a system is defined differential flatness if there exists a set of outputs, such that all states and inputs can be expressed in terms of the outputs and their finite-order derivatives, see [7]. From (1), we obtain:

$$\begin{aligned} \frac{dy}{dx} &= \tan\theta \\ \frac{d^2y}{dx^2} &= \frac{\tan\phi}{l\cos^3\theta} \end{aligned} \quad (16)$$

It is clear that θ, ϕ can be represented by the derivatives of the flat outputs, defined as x, y. Since x, y are parametric polynomials in (13), assuming $s = t$, we obtain:

$$\begin{aligned} \theta &= \arctan \frac{d_2 + 2d_3 s + 3d_4 s^2 + 4d_5 s^3 + 5d_6 s^4}{c_2 + 2c_3 s + 3c_4 s^2 + 4c_5 s^3 + 5c_6 s^4}, \\ \phi &= \arctan l\cos^3\theta \frac{2d_3 + 6d_4 s + 12d_5 s^2 + 20d_6 s^3}{2c_3 + 6c_4 s + 12c_5 s^2 + 20c_6 s^3}, \\ u_1 &= \frac{c_2 + 2c_3 s + 3c_4 s^2 + 4c_5 s^3 + 5c_6 s^4}{\cos\theta}, \end{aligned} \quad (17)$$

$$u_2 = \dot{\phi} = \frac{1}{1+z^2}\frac{dz}{ds} \quad (18)$$

where

$$\begin{aligned} z &= l\cos^3\theta \frac{2d_3 + 6d_4 s + 12d_5 s^2 + 20d_6 s^3}{2c_3 + 6c_4 s + 12c_5 s^2 + 20c_6 s^3}, \\ \cos\theta &= \frac{c_2 + 2c_3 s + 3c_4 s^2 + 4c_5 s^3 + 5c_6 s^4}{\sqrt{\Lambda}}, \\ \Lambda &= (c_2 + 2c_3 s + 3c_4 s^2 + 4c_5 s^3 + 5c_6 s^4)^2 \\ &\quad + (d_2 + 2d_3 s + 3d_4 s^2 + 4d_5 s^3 + 5d_6 s^4)^2. \end{aligned}$$

Therefore, the steering control (17) and (18) can be calculated analytically without taking differentiation. Using them, the planned trajectory will be tracked exactly by the robot.

V. Cooperative Coverage of Multiple Robots

In the case of multiple robots to conduct a cooperative coverage, we first partition the region into M approximately equal sub-regions, where M is the number of the robots. Inspired by the fact that a collection of smaller polygon pieces can be arranged to form different polygons, we use the disk as the basic unit to form bigger sub-regions. If n circles are used to cover a bounded region, we divide it by M and round it up to a whole number k, and the resulting connected k circles forms a sub-region. The algorithm starts from the center of a circle near the boundary, and we keep a list which expands itself by adding its neighboring centers until the total number of points in the list equal to k; then the next neighbor point becomes the starting points for the next sub-region. Repeat the process for the $M-1$ times, and what's left is the M sub-region.

VI. Simulation Results

We have run extensive simulations of the proposed complete coverage control algorithms using Matlab. Cooperative coverage trajectories for four robots are shown in Figure 7. We have an animation to show that a region with stationary and moving obstacles is completed covered by four robots' cooperative sweeping over time. Figure 8 shows the result. We observed from the simulations that minimum repeated coverage is achieved and minimum numbers of backups are needed in a random shaped area.

VII. Conclusions

We have considered complete coverage control for nonholonomic mobile robots. First, we pack a rectangle with disks of minimum number, and then plan a complete coverage path going through all centers of the the disks using the neural network method. A smooth trajectory is then generated using parametric polynomials of fifth-order, which ensure continuity of second-order derivatives at any point of the trajectory. Finally, open-loop steering control is designed based on differential flatness of the system. Since the smooth trajectory is an analytical function of parameters such as the time, the steering control can be calculated analytically, which is computationally less expensive than numerical methods, and works for online planning and control. Future work includes experimental tests of the algorithm on a team of Amigobots.

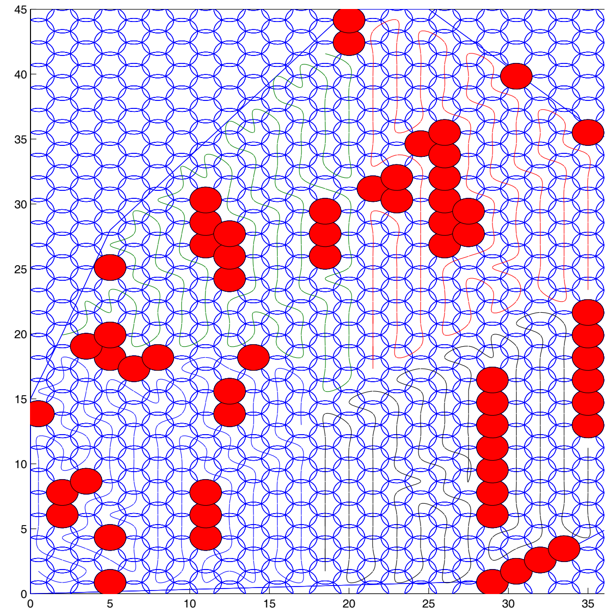

Fig. 7. Cooperative coverage trajectories: each colored curve represents one robot's trajectory, and the red solid circles denote areas occupied by stationary obstacles.

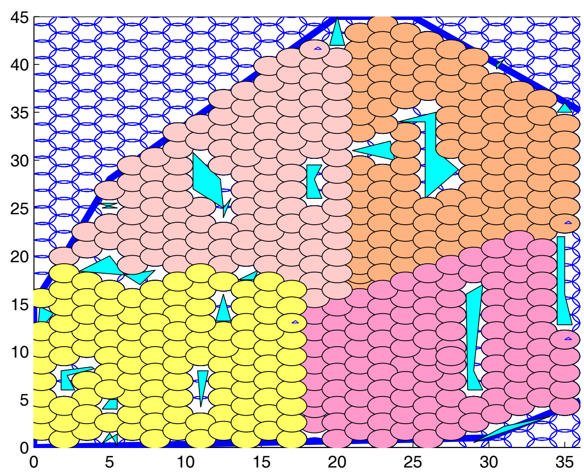

Fig. 8. Complete coverage by four robots: solid circles represent covered area over time, each color stands for one robot; the blue irregular shapes represent stationary obstacles.

References

[1] M. Batalin and G. S. Sukhatme. Coverage, exploration and deployment by a mobile robot and communication network. In *Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks*, pages 376–391, Palo Alto, CA, 2003.

[2] A. Bloch, M.Reyhanoglu, and N.H. McClamroch. Control and stabilization of nonholonomic dynamic systems. *IEEE Trans. on Automatic Control*, 37(11):1746–1757, 1992.

[3] R.W. Brocket and L. Dai. *Nonholonomic kinematics and the role of elliptic functions in constructive controllability*, pages 1–22. 1992.

[4] H. Choset. Coverage of known spaces: the boustrophedon cellular decomposition. *Autonomous Robots*, 9(3):247–253, 2000.

[5] J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1327–1332, 2002.

[6] R. N. de Carvalho, H. A. Vidal, P. Vieira, and M. I. Ribeiro. Complete coverage path planning and guidance for cleaning robots. In *Proceedings of IEEE Int. Symposium on Industrial Electronics*, 1997.

[7] M. Fliess, J. Levine, Ph. Martin, and P. Rouchon. Flatness and defect of nonlinear systems: Introductory theoy and examples. *Int. J. of Control*, 61(6):1327–1361, 1995.

[8] Y. Gabriely and E. Rimon. Spanning -tree based coverage of continuous areas by a mobile robot. *Ann. Math. Artificial Intell.*, 31(1-4):77–98,

2001.
[9] Y. Guo, L. E. Parker, and R. Madhavan. Towards collaborative robots for infrastructure security applications. In *Proceedings of The 2004 International Symposium on Collaborative Technologies and Systems*, pages 235–240, San Diego, CA, Jan. 2004.
[10] Y. Guo and Z. Qu. Coverage control for a mobile robot patrolling a dynamic and uncertain environment. In *Proceedings of World Congress on Intelligent Control and Automation*, pages 4899–4903, China, June 2004.
[11] T. Heath-Pastore, H.R. Everett, and K. Bonner. Mobile Robots for Outdoor Security Applications. In *Proceedings of the American Nuclear Society 8th International Topical Meeting on Robotics and Remote Systems*, April 1999.
[12] 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.*, 117:500–544, 1952.
[13] C. Hofner and G. Schmidt. Path planning and guidance techniques for an autonomous mobile robot. *Robotics Auton. Syst.*, 14(203):199–212, 1995.
[14] J. L. Junkins and J. R. Jancaitis. Smooth irregular curves. *Photogrammetric Engineering*, 38(6):565–573, 1972.
[15] S. X. Yang and C. Luo. A neural network approach to complete coverage path planning. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, 34(1):718–725, 2004.
[16] A. Zelinsky, R. A. Jarvis, J. C. Byrne, and S. Yuta. Planning paths of complete coverage of an unstructured environment by a mobile robot. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2612–2619, 1993.