Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 263964, 11 pages
http://dx.doi.org/10.1155/2015/263964

Research Article

Chaotic Path Planner of Autonomous Mobile Robots Based on the Standard Map for Surveillance Missions

Caihong Li,[1] Yong Song,[2] Fengying Wang,[1] Zhenying Liang,[1] and Baoyan Zhu[1]

[1]*School of Computer Science and Technology, Shandong University of Technology, Shandong 255049, China*
[2]*School of Mechanical, Electrical & Information Engineering, Shandong University, Shandong 264209, China*

Correspondence should be addressed to Caihong Li; lich@sdut.edu.cn

Received 30 April 2015; Revised 27 June 2015; Accepted 30 July 2015

Academic Editor: Jonathan N. Blakely

This paper proposes a fusion iterations strategy based on the Standard map to generate a chaotic path planner of the mobile robot for surveillance missions. The distances of the chaotic trajectories between the adjacent iteration points which are produced by the Standard map are too large for the robot to track. So a fusion iterations strategy combined with the large region iterations and the small grids region iterations is designed to resolve the problem. The small region iterations perform the iterations of the Standard map in the divided small grids, respectively. It can reduce the adjacent distances by dividing the whole surveillance workspace into small grids. The large region iterations combine all the small grids region iterations into a whole, switch automatically among the small grids, and maintain the chaotic characteristics of the robot to guarantee the surveillance missions. Compared to simply using the Standard map in the whole workspace, the proposed strategy can decrease the adjacent distances according to the divided size of the small grids and is convenient for the robot to track.

1. Introduction

An important issue on surveillance missions for autonomous mobile robots is the complete coverage path planning, with the specific purpose of generating a trajectory, which will guarantee the surveillance of the entire terrain. In the case of patrolling for intrusion, the path of the robot must be as much difficult as possible to be predicted by the intruder [1]. It should satisfy three major conditions: the unpredictability of the trajectory, the scan of the entire terrain, and the fast scanning of the robot's workplace. These are the basic requirements for selecting the most suitable autonomous mobile robots for the specific kind of missions [2].

Chaotic systems provide the much-needed framework in achieving the surveillance missions. The main characteristics of the chaotic systems are the topological transitivity and the sensitive dependence on initial conditions [3]. Due to topological transitivity, the chaotic autonomous mobile robot is guaranteed to patrol the whole surveillance region completely. The sensitive dependence on initial conditions is a desirable characteristic for patrol robots, since the trajectory of the chaotic autonomous mobile robot is very unpredictable [4–7].

The aim of using chaotic systems in autonomous robots is achieved by designing path planers, which ensure chaotic motion. Signals, which are produced by chaotic systems or circuits, are used to guide autonomous robots for exploration of a terrain for vigilance. In the literature, very known chaotic systems, such as Arnold dynamical system, Standard or Taylor-Chirikov map, Lorenz system, Van der Pol map, and Chua circuit, have been used [7–11]. In [12], we also present a chaotic motion path planner, which is improved by arcsine and arccosine transformations, based on a Logistic Map for an autonomous mobile robot to cover an unknown terrain randomly, unpredictably, and evenly.

The key problem of this research is how to impart the chaotic motion behavior to an autonomous mobile robot. The combination means of the robot kinematics equation with the chaotic map are different due to their various dimensions and control parameters. Since there are many chaotic systems, which pattern is a good candidate to be chosen for use as

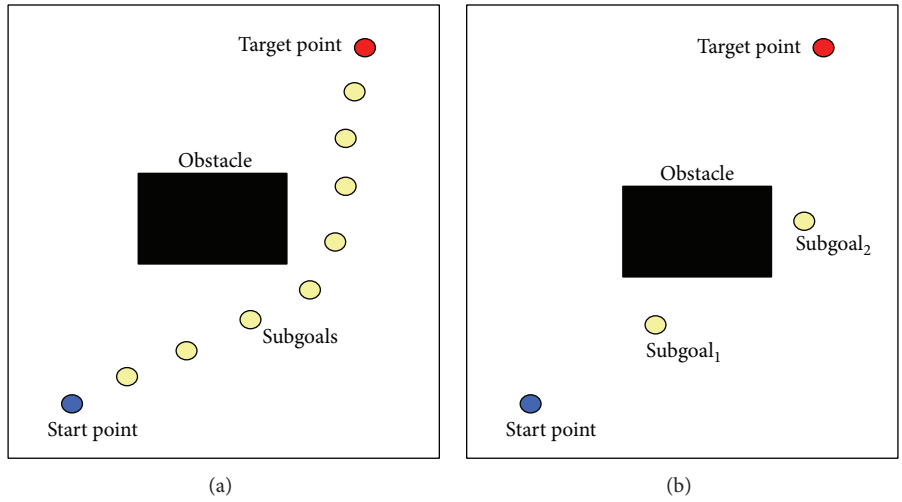


Figure 1: Sequence subgoals generated by the path planner.

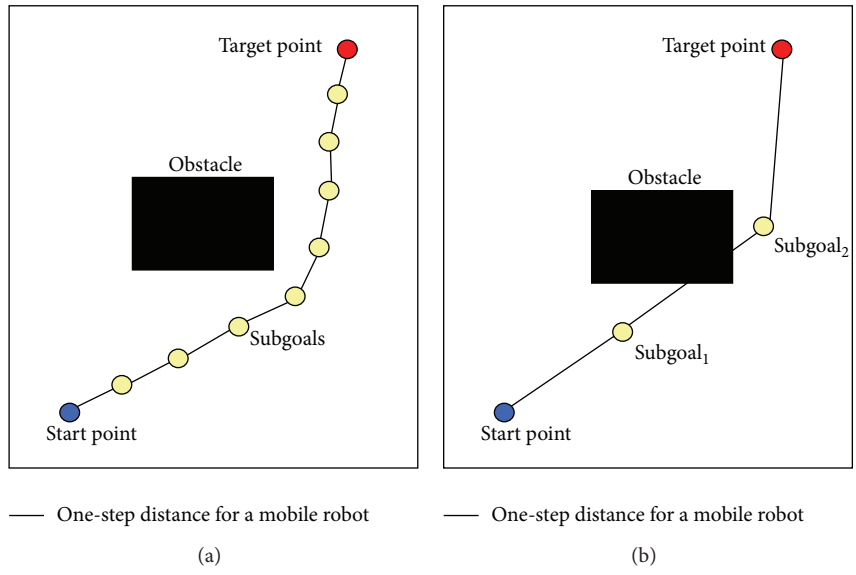


Figure 2: Line all the points subsequently.

a chaotic path planning generator? In general, the criterion in many works is the largest possible coverage area through computer simulation. The Taylor-Chirikov, or Standard map, is a well-known two-dimensional mapping [13]. It is easy to be controlled for having single parameter K. Furthermore it has the better coverage area of the whole region. So it arises in many applications including the surveillance missions of the robot. In [8], using the well-known Standard map, L. S. Martins-Filho and E. E. N. Macau proposed an ingenious path-planning mechanism where the sequence of intermediary goal positions is obtained.

However, most of the above researches have ignored the process of the planned subgoals, or the intermediary iteration points, such as [8]. When the untreated points are served as the subgoals of the path planner and sent to the robot motion controller to track, it is likely to fail fulfilling the whole patrolling missions due to the large distances of the trajectories between adjacent subgoals. Like the circumstance in Figure 1, the robot is required to run from the start point to the target point. The path planner generates some sequences points as the subgoals for the robot according to the task requirements. Then the subgoals are passed to the robot controller to track. In Figure 1(a), there are eight subgoals, while in Figure 1(b) there are only two. In Figure 2, we line all the points successfully to show a rough running trajectory for the robot. But it is not a real running path. If the lines lengths are too large compared to one-step distance of a robot, such as Figure 2(b), the robot controller cannot know how to get to subgoal$_2$ from subgoal$_1$ safely. So the robot needs to plan the path again, such as Figure 3. Then the last planning task failed for Figure 1(b) circumstance.

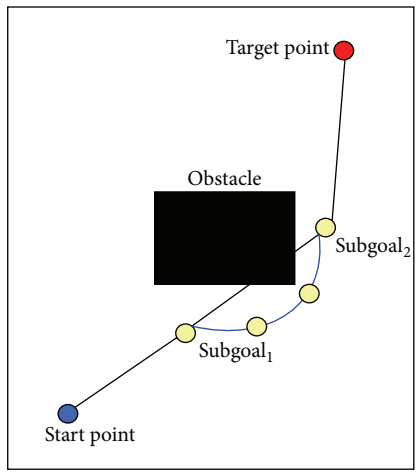


Figure 3: Replaned path between subgoal$_1$ and subgoal$_2$.

In this paper, we mainly talk about the path planner. The straight lines are rough path. By proposing a fusion strategy based on the Standard map by affine transformations, our task is to produce the shorter distance of the adjacent subgoals for the robot to track conveniently and safely, such as Figure 1(a). When the robot controller tracks the subgoals, its trajectory is the real path. Figure 4 illustrates the procedure of the path planning and tracking.

The paper is organized as follows. Section 2 discusses the dynamic characteristics of the Standard map with the single parameter K changing. When it is in its chaotic state, we further study the intermediate iteration points distribution in the phase space and the adjacent trajectories. Affine transformations of the Standard map are introduced in Section 3. And the proposed fusion iterations strategy is designed based on the transformation. Section 4 provides the simulation results of the designed strategy and the comparison results of the original Standard map. Conclusions are drawn in Section 5.

2. Dynamic Characteristic of the Standard Map

The Standard map, or the Taylor-Chirikov, is the two-dimensional area preserving dynamical system [14, 15]. Area-preserving mapping gives rise to incredibly rich dynamics and mathematics [16]. It describes the dynamics of magnetic field lines on the kicked rotor [17].

The continuous nonlinear mapping is as follows [13]:

(mod1)

$$\begin{aligned} x' &= x + y \\ y' &= y + \frac{K}{2\pi} \sin(2\pi x). \end{aligned} \tag{1}$$

Here x is a periodic configuration variable (angular position) and y is the momentum variable (angular speed); both computed mod (2π). The map has a single parameter, K, that represents the strength of the nonlinear kick. Randomly selected, an initial point (x_0, y_0) and its next N iterates are

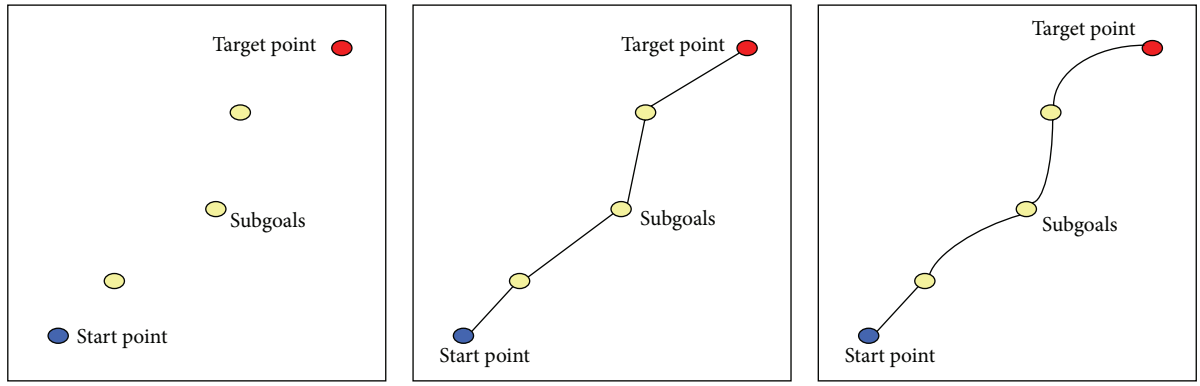
$$(x_t, y_t) = f(x_{t-1}, y_{t-1}) \quad 1 \le t \le N. \tag{2}$$

So we commonly use the discrete mathematical model of the Standard map to compute the sequence iteration values step by step:

$$\begin{aligned} x_{n+1} &= x_n + K \sin y_n, \\ y_{n+1} &= y_n + x_{n+1}, \\ x_{n+1} &= \mod(x_{n+1}, 2\pi), \\ y_{n+1} &= \mod(y_{n+1}, 2\pi). \end{aligned} \tag{3}$$

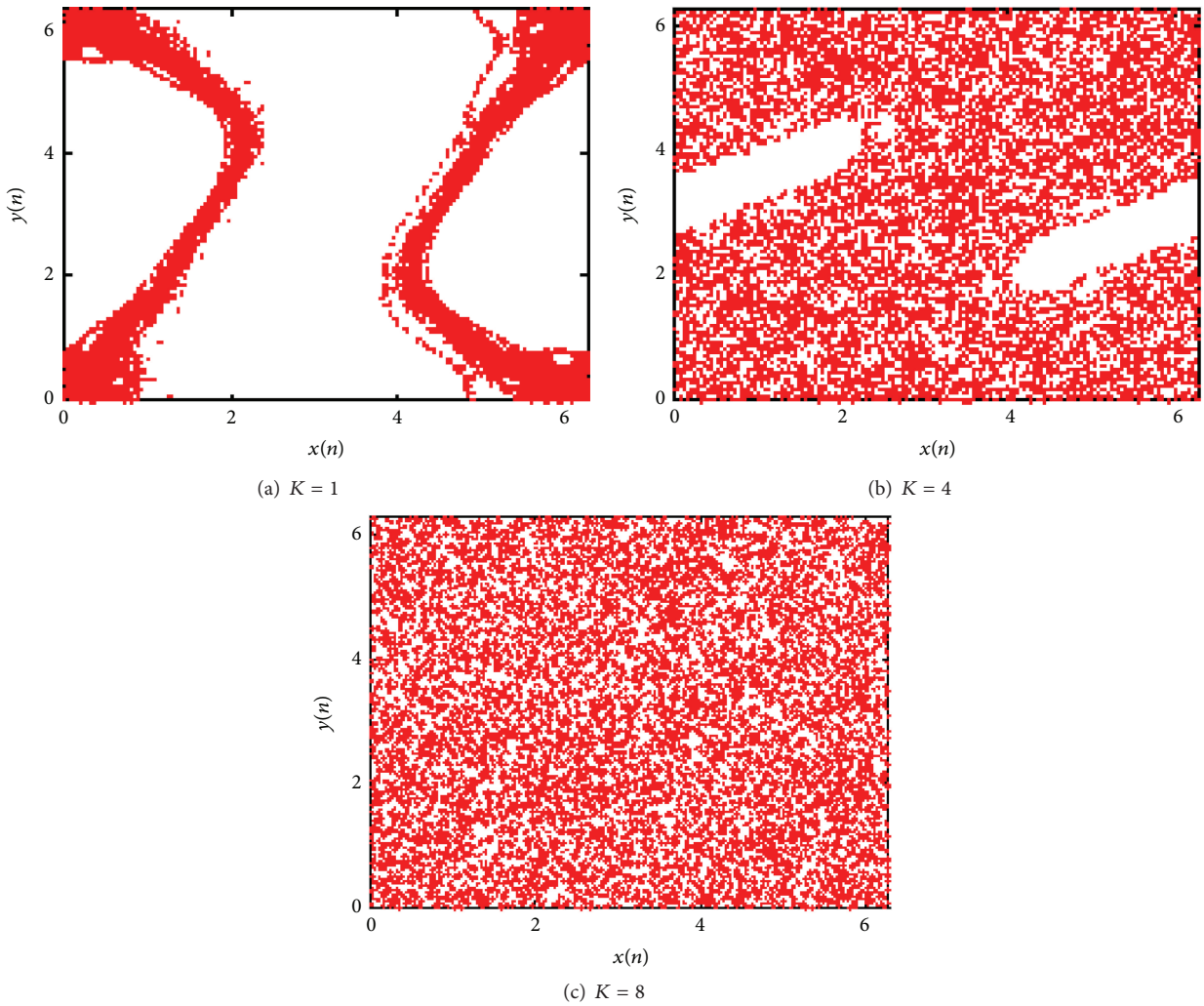
The single parameter K controls the dynamic properties of the map. If K is small, the map demonstrates periodic orbit, such as Figure 5(a), which is simulated by the computer according to (3). With the parameter value increasing, the trajectory becomes chaotic and fills a large region of phase space, as Figure 5(b). As K increases further, when $K > 6$, for $K = 8$, one has the impression that chaos is complete, which is shown in Figure 5(c).

When the Standard map is completely chaotic, from Figure 5(c), we can see it demonstrates better uniform distribution and iterates in a limited region. According to (3), values x_n and y_n are defined in $(0, 2\pi)$, respectively. So the iteration values (x_n, y_n) are bounded and can be used as the intermediary points, or subgoals, to generate a chaotic path planner to guide the mobile robot to complete coverage of the whole surveillance region. But when the robot tracks the subgoals, namely, runs from one subgoal (x_n, y_n) to the next point (x_{n+1}, y_{n+1}), the distance of the trajectory between the adjacent points, $|(x_n, y_n), (x_{n+1}, y_{n+1})|$, should be considered. Taking into account the robot's each moving step and the whole surveillance region, if the adjacent subgoals distance is too large, the guiding navigation significance of the subgoals loses to the robot. Then the robot planner has to supplement other new intermediate points between the subgoals.



(a) (b) (c)

FIGURE 4: Illustration of the path planning and tracking: (a) subgoals generated by the path planner; (b) rough path; (c) path tracking.



(a) $K = 1$ (b) $K = 4$

(c) $K = 8$

FIGURE 5: Phase space of the Standard map with various parameter K (for $x(0) = 0.5$, $y(0) = 0.5$, and $n = 10000$).

Figure 6 analyzes iteration points distribution of the Standard map in the phase space and the adjacent trajectories qualitatively and quantitatively. Randomly select an initial point, (0.5,0.5), in the phase space, for $k = 8$; Figure 6(a) demonstrates the 500 iterations' distribution of intermediate coordinate point (x_n, y_n). We select a smaller iteration number, 500, in order to make the simulation pictures clearly. Figure 6(a) shows that the points distribute uniformly and

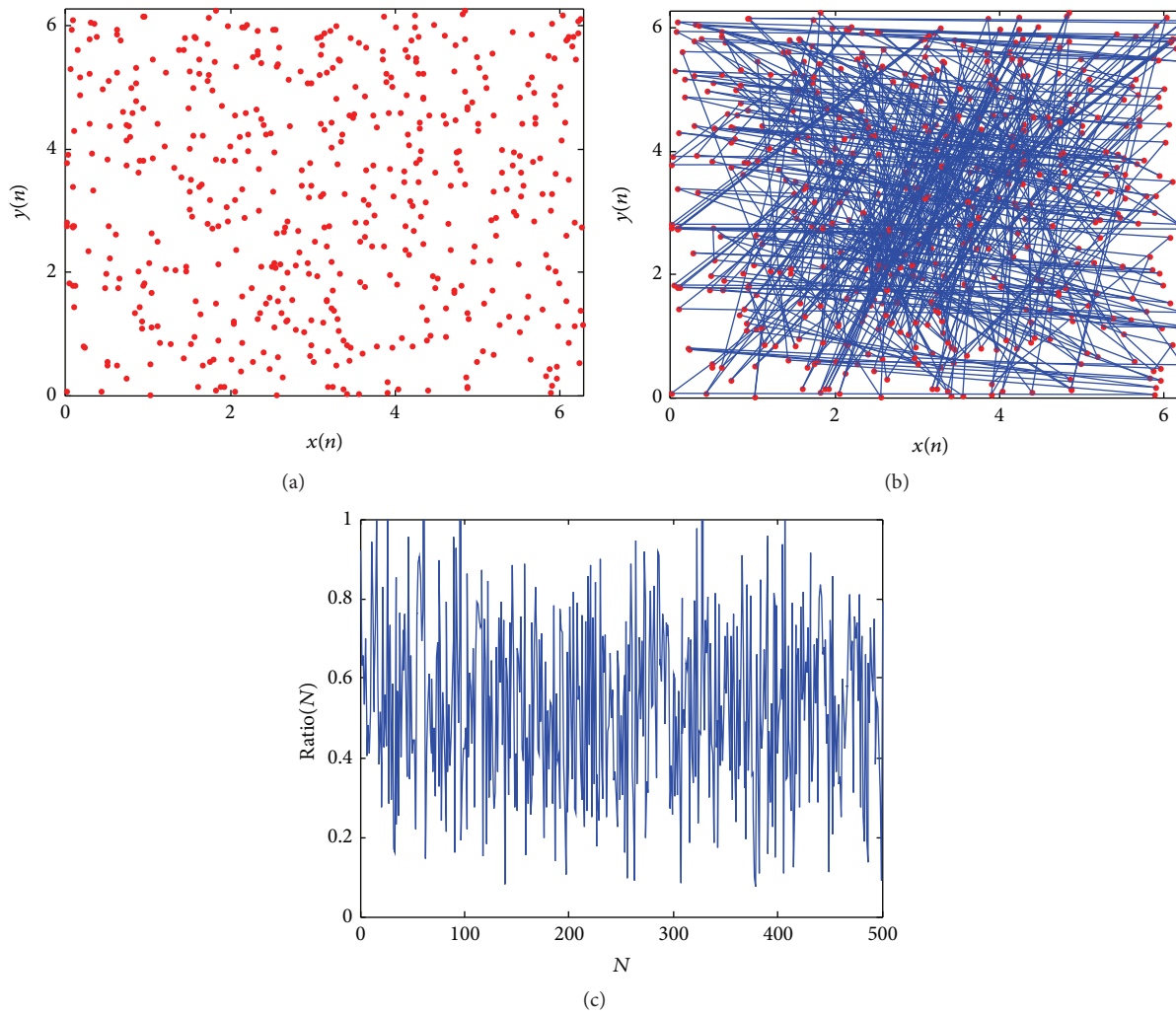


Figure 6: Intermediate iteration points distribution in the phase space and their trajectories (for $k = 8$; $x(0) = 0.5$; $y(0) = 0.5$; $n = 500$): (a) iteration points; (b) iteration trajectories; (c) Ratio(N) values to its defined region.

completely. Figure 6(b) draws all the iterations trajectories between the adjacent points. Compared to the defined region of the phase space, $2\pi \times 2\pi$, the interval distances are too large. In order to qualitatively analyze the distances value, we design an evaluation index, Ratio(N):

$$\text{Ratio}(N) = \frac{\left|(x_n, y_n), (x_{n+1}, y_{n+1})\right|}{2\pi} \quad (N = 1, \ldots, 500), \tag{4}$$

where 2π is the Standard map bounded value. Figure 6(c) shows that most of the Ratio(N) values are concentrated at the vicinity of 0.5. The average value of the 500 times' Ratio(N) is about 0.5323. It means that the bigger the workspace boundary values, the bigger the adjacent distances. Furthermore it is about half of the bounded values, while, for a defined region, values of Ratio(N) are constant. So we have to change the size of the defined region to decrease the adjacent distances to make the subgoals of the path planner meaningful to the robot navigation.

3. Fusion Strategy of the Standard Map

In order to fulfill the track task for the mobile robot, the adjacent planned subgoals distance produced by the path planner should not be too large. Equation (3) produces the sequence values step by step based on a random selected initial value. The values are only associated with the iteration number, so the Ratio(N) values are constant to a given defined Standard map. Then we cannot directly decrease the Ratio(N) values by (3). Furthermore the whole chaotic characteristics of the Standard map should be maintained to meet the satisfaction of the surveillance missions for the robot. Based on the above mentioned principle, this paper proposes a fusion strategy based on the Standard map which is combined with large region iterations and small grids region iterations. Both iterative processes are still Standard map based on simple affine transformations. They are linked by the intermediate iteration points and can transform from one state to the other by controlling the iterative cycles. Function of the small grids region iterations is to decrease the adjacent planned subgoals distance by

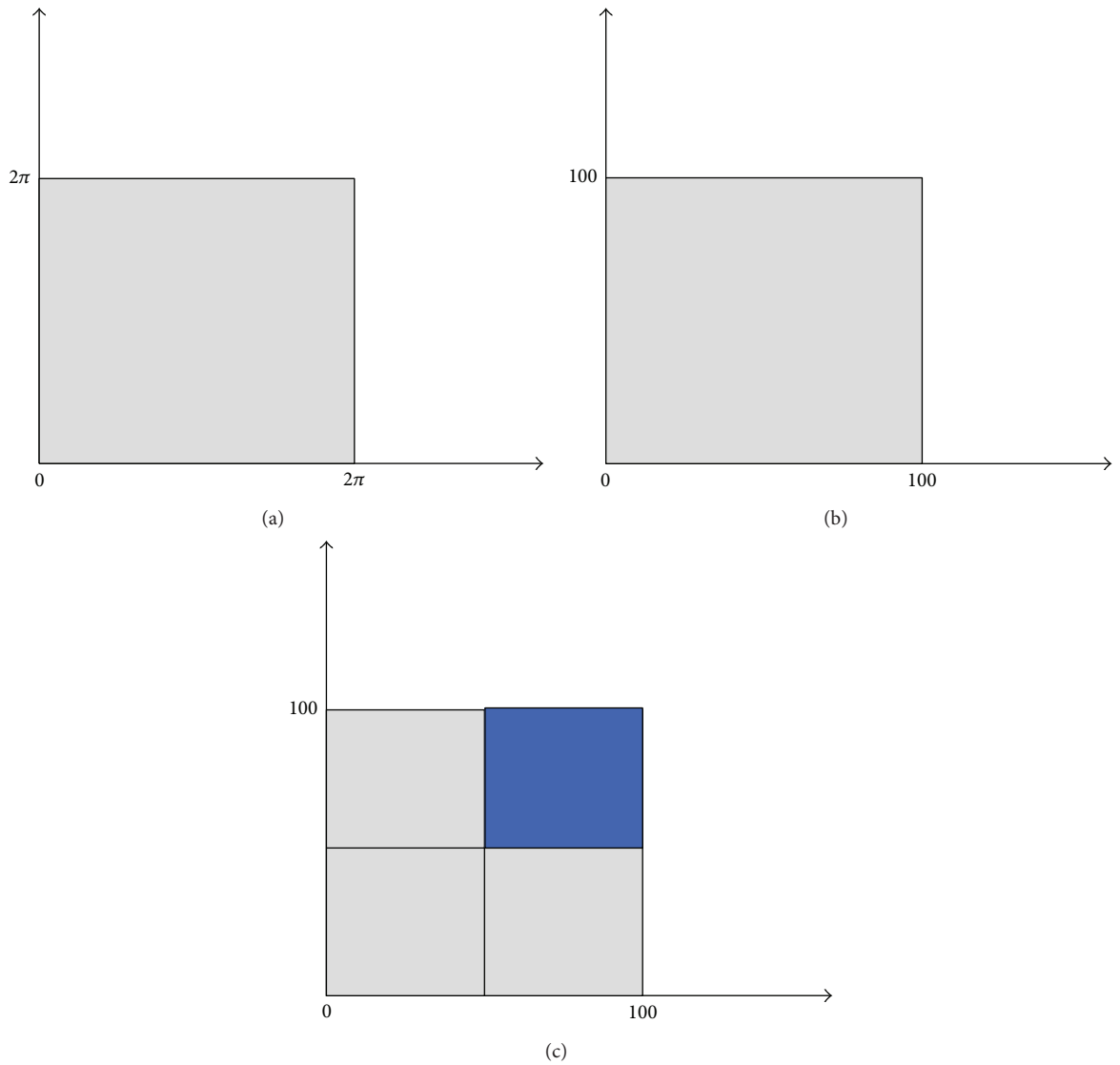


Figure 7: Mapping procedure of the original Standard map: (a) defined region of the original Standard map; (b) surveillance workspace of the mobile robot; (c) divided small grids.

reducing their workspace bounded value and perform the small grids region coverage, while the large one realizes the automatic conversion of small grids region to combine all the small iterations together and maintain the whole chaotic characteristics of the surveillance workspace.

3.1. Affine Transformations of the Standard Map. Figures 5 and 6 show that the Standard map bounded region is defined in a square, about $2\pi \times 2\pi$. This defined region can be mapped into the robot surveillance workspace, such as 100×100. Then the whole region is divided into some small square grids; here suppose the number is 2×2. Size of the number is determined by the requirement of the adjacent subgoals distances for the surveillance mission. Because the Ratio(N) values are constant in each iteration cycle, the bigger the divided size of the number, the smaller the distance values. Figure 7 depicts the mapping procedure of the original Standard map. The previously mentioned large region iterations work on the 100×100 map. And the small iterations run on the 4-grid map, respectively, of Figure 7(c). For example, the blue grid is one of them in the figure.

The mapping procedure can be accomplished using simple affine transformations like scalings and translations that are applied to the original Standard map. We commonly use Cartesian coordinate system on engineer problems. Suppose transform one coordinate point A to a new coordinate point B:

$$B = AM, \tag{5}$$

where $A = [x, y, 1]$, $B = [x', y', 1]$, and then M is a 3×3 affine matrix:

$$M = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ t_x & t_y & 1 \end{bmatrix}. \tag{6}$$

So (5) can be written in

$$[x' \quad y' \quad 1] = [x \quad y \quad 1] \times \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ t_x & t_y & 1 \end{bmatrix}. \tag{7}$$

Expand formula (7):

$$\begin{aligned} x' &= ax + cy + t_x, \\ y' &= bx + dy + t_y. \end{aligned} \tag{8}$$

From (8) we can derive the scalings and translations matrix.

Translations transform is to move arbitrary point t_x distance away along X coordinate axis and t_y distance away along Y coordinate axis. The transform matrix T is

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}. \tag{9}$$

After expansion,

$$\begin{aligned} x' &= x + t_x, \\ y' &= y + t_y. \end{aligned} \tag{10}$$

Scalings transform is to enlarge or reduce the s_x times of the horizontal coordinate value of an arbitrary point, while the longitudinal direction is s_y times. The transform matrix S is

$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{11}$$

After expansion,

$$\begin{aligned} x' &= x \cdot s_x, \\ y' &= y \cdot s_y. \end{aligned} \tag{12}$$

According to scalings transform, the parameters values of mapping Figure 7(a) to Figure 7(b) are $s_x = s_y = 100/2\pi$. The 4 small grids in Figure 7(c) use both scalings transform and translations transform. Like the blue one, the scalings transform parameters are $s_x = s_y = 50/2\pi$, and the translations transform parameters are $t_x = 50$ and $t_y = 50$, namely, its coordinate values of the lower left corner. All translations are performed based on (3).

In the fusion strategy in this paper, the whole workspace is supposed to be as in Figure 7(c). The large iteration cycles run on the 100×100 region and the small iteration cycles work on the 4 small grids, respectively. Because the iteration values are bounded in the standard map, the values only iterate in their own defined region and cannot run out of its region to other grids. If the robot only runs in the small grids and cannot automatically switch to other ones, the surveillance mission of the whole workspace fails. So we integrate the large iteration cycles into the small ones to realize various grids transformation.

3.2. Realization of the Fusion Iterations Strategy. All iteration cycles are based on the Standard map and (3). The detailed designed process of the fusion strategy is as follows.

(1) The whole workspace is divided into $k \times k$ small grids, according to the requirement value of the adjacent subgoals distances for the surveillance mission. Then each grid region is determined.

(2) Initialization of the small grids region iterations parameters: the total iterative steps are M and each iterative step in one cycle is m, $M = m \times k^2$.

(3) Initialization of the large region iterations parameters: the total iterative steps are N, $N \geq n \times k^2$, and each iterative step in one cycle is n.

(4) Randomly select an initial point in the whole workspace; start up the large iterations cycle until the iterations step n is complete. Lastly, the iteration break-point $(X(n), Y(n))$ is recorded.

(5) Judge which small grid the coordinates $(X(n), Y(n))$ point falls into.

(6) Judge whether the small grid has been totally covered. If it is yes, then continue the large iterations cycle from the last break-point $(X(n), Y(n))$ and restart the next n iterative steps.

(7) If it is no, judge whether the small grid has been already covered or not. If it is no, use the point $(X(n), Y(n))$ as the initial point, and start up the small iteration cycle until the iteration steps m are complete. At last, record the break-point $(X(m), Y(m))$ and use it as the initial point in the next iteration cycle. If it is yes, the robot moves to the last break-point $(X(m), Y(m))$ and starts up the small iteration cycle based on it.

(8) All the iterations cycles are transformed according to the iterative steps and the above regulations until all the small grids are totally covered or the large iterations cycles are complete.

The pseudocodes of the above strategy procedure are in Pseudocode 1.

Figure 8 illustrates part operations of an example for the designed fusion strategy according to the above process. In order to demonstrate the simulation picture clearly, all the numbers are selected relatively small values. Each iterative step in the small iterations cycles is $n = 10$, and the large one is $m = 10$ too. The divided grids number is 4. Each small grid is encoded clockwise, which is labeled Grids 1, 2, 3, and 4, respectively. Blue lines in the pictures are the small iterations trajectories. And the black ones are the large iterations trajectories.

From Figure 8, we can conclude that the robot can scan the whole workspace completely by the fusion of the large region iterations and small grids iterations. In the small grids iterations, the adjacent subgoals distances decrease due to their small defined region. If the size of the divided small grids is big enough, the adjacent subgoals distances in the large region iterations can be ignored by selecting the appropriate

```
// initialization of the parameters
   suppose a whole workspace G_Ω;
   size of the divided small grids ← k * k;
   each small grid region G_0, G_1, ..., G_X, ..., G_{k*k} of G_Ω;
   the total small iterative steps ← M;
   each small iterative steps in one cycle ← m;
   the total large iterative steps ← N;
   each large iterative steps in one cycle ← n;

// the whole iteration procedure
   while (N! = 0);
      (X(0), Y(0)) ← rand(G_Ω); // randomly select an initial point in G_Ω
      large_iteration_cycle;
      N = N − n;
      (X(0), Y(0)) ← (X(n), Y(n));
      (X(n), Y(n)) ∈ G_X; // compute small grid number
      if Sk = M; // Sk is the small grid iteration times in the current G_X;
         large_iteration_cycle;
         elseif Sk = 0;
            small_iteration_cycle;
            M = M − m;
            (X(0), Y(0)) ← (X(m), Y(m));
            if M = 0;
               breaks;
            end
         else small_iteration_cycle;
      end
   end

   function small_iteration_cycle
      mk = 0;
      while (mk! = m);
         Equation (3);
         mk = mk + 1;
      end
   end

   function large_iteration_cycle
      Lk = 0;
      while (Lk! = n);
         Equation (3);
         Lk = Lk + 1;
      end
   end

 function Equation (3)
   K = 8;
   X(n + 1) = X(n) + K * sin(Y(n));
   Y(n + 1) = Y(n) + X(n + 1);
   X(n + 1) = mod(X(n + 1), 2 * pi);
   Y(n + 1) = mod(Y(n + 1), 2 * pi);
 end
```

Pseudocode 1: Pseudocodes of the iterations fusion strategy.

large and small iterations times. Then the average adjacent subgoals distances are reduced to the $1/K$ times of the conventional methods, where K is the divided small grids number.

4. Simulation of the Fusion Strategy

Figure 9 gives complete simulation results of the designed fusion strategy for the 16 divided grids of the whole workspace. The total iterations times of each small grid are $N = 100$, and iterations times of each cycle are $n = 10$. The large iterations times of each cycle are $m = 10$, and the total iterative number is about 400. When all the small grids are totally covered, the iterations end. In Figure 9(b), the cyan circle points are produced by the large iterations cycles, and the red dots are given by the small iterations cycles. For comparison, Figure 10 gives the original Standard map simulation results on the whole workspace. Or it can be said

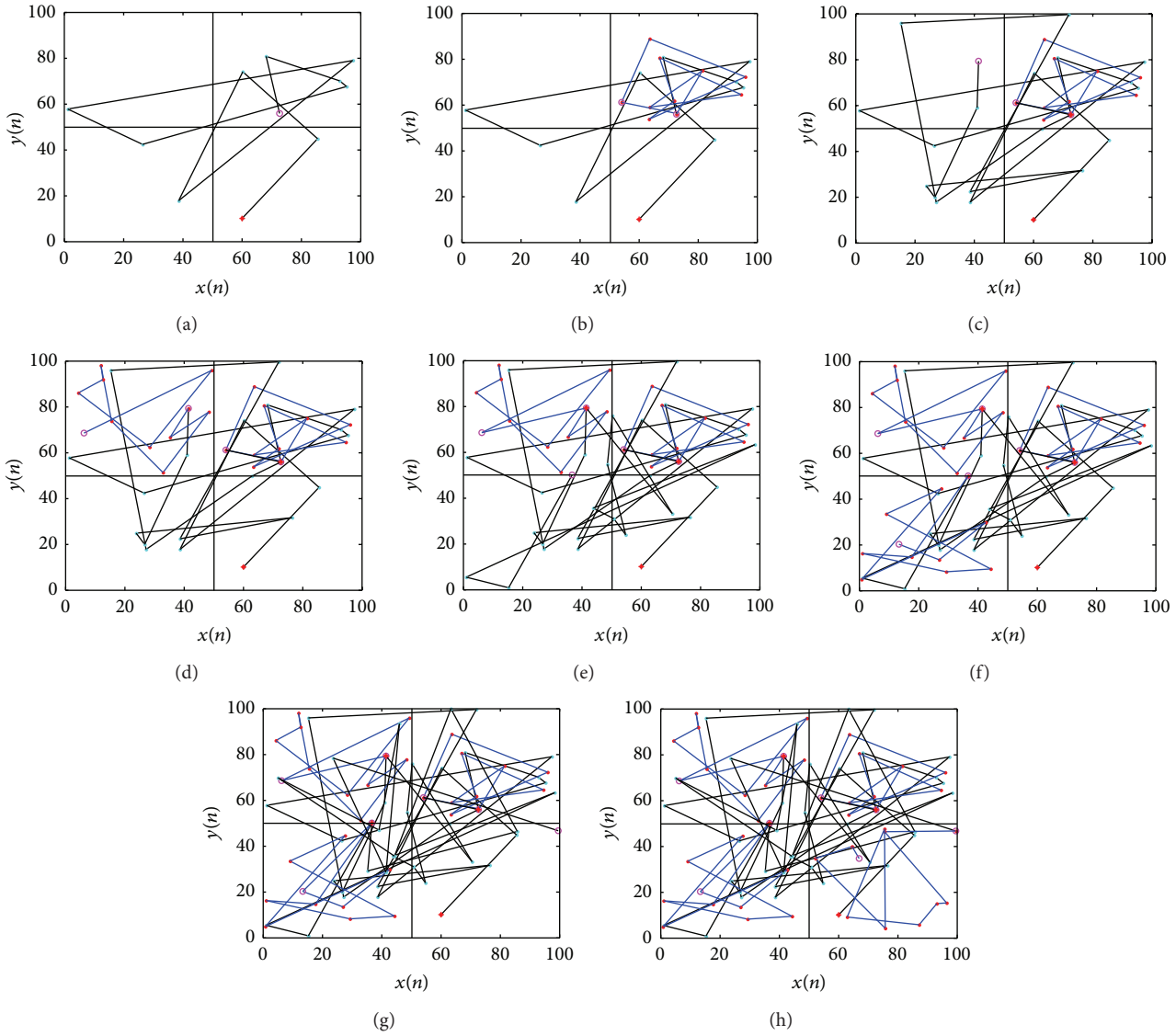


Figure 8: An example of the fusion strategy: (a) in Grid 4, a red "∗" is selected as the initial point. Start up a large iterations cycle from the point. When the iterations stop, the final point, the purple "o," falls into Grid 3; (b) begin a small iterations cycle in Grid 3 from the starting point purple "o." The iterations cycle ends at the left purple "o" position; (c) the robot moves from the current final position to the last break-point, namely, the right purple "o" position, and starts up the next large iterations cycle. The final point falls into Grid 2, the purple "o" position; (d) similar to (b), begin a small iterations cycle in Grid 2; (e) similar to (c), start up a large iterations cycle and end in Grid 1; (f) similar to (b) and (d), begin a small iterations cycle in Grid 1; (g) similar to (c) and (e), start up a large iterations cycle and end in Grid 2; (h) similar to (b), (d), and (f), begin a small iterations cycle in Grid 2.

that we use only large iterations in the 100×100 region. The iterations times are 2000, which is the total iterative times of Figure 9 fusion iterations times.

In Figure 9(a), the blue lines occupy most of the area. Compare to Figure 10(a), the adjacent planned subgoals distances in Figure 9(a) indeed reduce. Figures 9(b) and 10(b) both give the subgoals distribution in the whole workspace. They show almost the same uniform distribution characteristics. So the fusion strategy has little influence on the Standard map chaotic performance. This planned intermediate point can be used as the robot path planner subgoals and be sent to the robot controller to track.

5. Conclusion

Using the original Standard map of the chaotic state as the robot path planner generator to perform the surveillance mission is a good choice except for its large distance of the chaotic trajectories between the iteration adjacent intermediate points. After being treated properly, the coordinates (x_n, y_n) produced by the map can be used as the path planner intermediate subgoals directly. The proposed fusion strategy in this paper accomplishes the treatment of the coordinates points by dividing the whole surveillance workspace into small grids region to reduce the iteration adjacent distance.

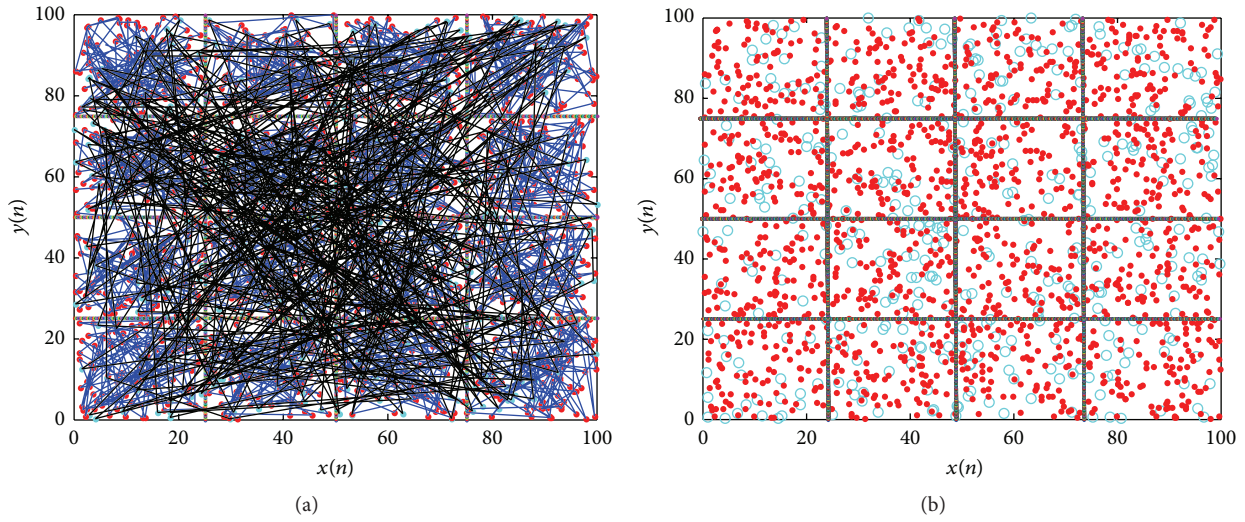


Figure 9: Fusion strategy: (a) iterations trajectory of the planned subgoals; (b) planned subgoals.

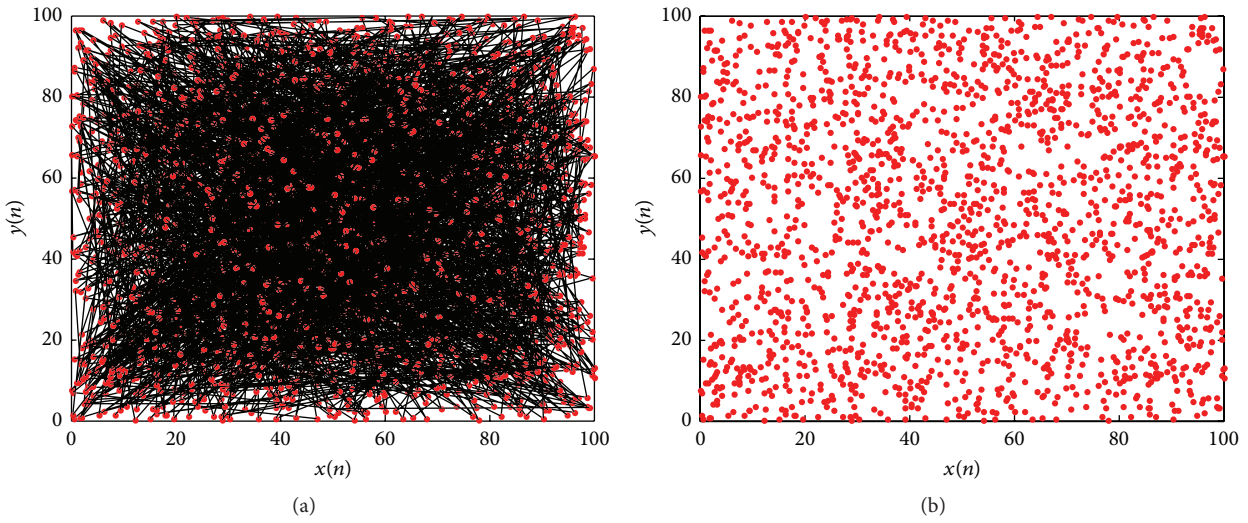


Figure 10: Original Standard map: (a) iterations trajectory of the planned subgoals; (b) planned subgoals.

The strategy has been simulated by MATLAB. It has the following features:

(1) Implementation of the strategy is simple. It has been realized by simple affine transformation based on the Standard map iterations and the partition of the whole surveillance workspace.

(2) Both the large region iterations and the small grids iterations continue the iterations cycles from their last break-points, respectively. Then the whole iterations process of the Standard map in the strategy has not been destroyed. So the whole chaotic characteristics of the fusion strategy remain the same as the original Standard map.

(3) Because each small grid owns the same chaotic characteristics as the original Standard map, the whole surveillance workspace combined with them has almost the same features too, especially for the characteristics of the topological transitivity.

(4) The switch frequency between the large region iterations and the small grids iterations is as follows: the bigger, the better. Or each iteration time in the small grids cannot be too big. Otherwise the unpredictable feature of the planned trajectory can be influenced.

In the fusion strategy, some large region iterations are needed to combine all the small grids region. The adjacent distances in the large region are still big. Here because they occupy less proportion, we can ignore their influences. In the future work, we want to study the method to further reduce the proportion in the total iterations, such as assumption of the adaptive or varied iterations times in the large region iterations.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by National Natural Science Foundation (NNSF) of China under Grant 61473179; Shandong Province Natural Science Fund of China under Grants ZR2014FM007 and ZR2013FM012; Shandong Province Science and Technology Development Projects under Grant 2013GGX10116; A Project of Shandong Province Higher Educational Science and Technology Program under Grant J13LN27.

References

[1] L. S. Martins-Filho and E. E. N. Macau, "Patrol mobile robots and chaotic trajectories," *Mathematical Problems in Engineering*, vol. 2007, Article ID 61543, 13 pages, 2007.

[2] C. K. Volos, N. G. Bardis, I. M. Kyprianidis, and I. N. Stouboulos, "Implementation of mobile robot by using double-scroll chaotic attractors," in *Recent Researches in Applications of Electrical and Computer Engineering*, pp. 119–124, WSEAS, 2012.

[3] B. Hasselblatt and A. Katok, *A First Course in Dynamics: With a Panorama of Recent Developments*, Cambridge University Press, Cambridge, UK, 2003.

[4] C. K. Volos, I. M. Kyprianidis, and I. N. Stouboulos, "A chaotic path planning generator for autonomous mobile robots," *Robotics and Autonomous Systems*, vol. 60, no. 4, pp. 651–656, 2012.

[5] A. I. Zhu and H. Leung, "Cooperation random mobile robots based on chaos synchronization," in *Proceedings of the 4th IEEE International Conference on Mechatronics (ICM '07)*, pp. 1–5, May 2007.

[6] K. Fallahi and H. Leung, "A cooperative mobile robot task assignment and coverage planning based on chaos synchronization," *International Journal of Bifurcation and Chaos*, vol. 20, no. 1, pp. 161–176, 2010.

[7] Y. Nakamura and A. Sekiguchi, "The chaotic mobile robot," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 898–904, 2001.

[8] L. S. Martins-Filho and E. E. N. Macau, "Trajectory planning for surveillance missions of mobile robots," *Studies in Computational Intelligence*, vol. 76, pp. 109–117, 2007.

[9] P. Sooraska and K. Klomkarn, "'No-CPU' chaotic robots: from classroom to commerce," *IEEE Circuits and Systems Magazine*, vol. 10, no. 1, pp. 46–53, 2010.

[10] A. Jansri, K. Klomkarn, and P. Sooraksa, "On comparison of attractors for chaotic mobile robots," in *Proceedings of the 30th Annual Conference of IEEE Industrial Electronics Society (IECON '04)*, vol. 3, pp. 2536–2541, Busan, Korea, November 2004.

[11] D. I. Curiac and C. Volosencu, "Developing 2D trajectories for monitoring an area with two points of interest," in *Proceedings of the 10th WSEAS International Conference on Automation and Information*, pp. 366–369, June 2009.

[12] L. Caihong, W. Fengying, Z. Lei, Y. Li, and Y. Song, "An improved chaotic motion path planner for autonomous mobile robots based on logistic map," *International Journal of Advanced Robotic Systems*, vol. 10, pp. 1–9, 2013.

[13] G. Contopoulos, "Order and chaos in dynamical systems," *Milan Journal of Mathematics*, vol. 77, no. 1, pp. 101–126, 2009.

[14] M. Shamis and T. Spencer, "Bounds on the lyapunov exponent via crude estimates on the density of states," *Communications in Mathematical Physics*, 2015.

[15] J. D. Simoi, "Fermi acceleration in anti-integrable limits," *Communications in Mathematical Physics*, vol. 321, pp. 703–745, 2013.

[16] J. D. Meiss, "Visual explorations of dynamics: the standard map," *Pramana*, vol. 70, no. 6, pp. 965–988, 2008.

[17] A. J. Lichtenberg and M. A. Lieberman, *Regular and Stochastic Motion*, Springer, Berlin, Germany, 1983.