Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A chaotic coverage path planner for the mobile robot based on the Chebyshev map for special missions*

Cai-hong LI†1, Yong SONG[2], Feng-ying WANG[1], Zhi-qiang WANG[1], Yi-bin LI[2]

([1]*College of Computer Science and Technology, Shandong University of Technology, Zibo 255000, China*)

([2]*School of Mechanical, Electrical & Information Engineering, Shandong University, Weihai 264209, China*)

†E-mail: lich@sdut.edu.cn

Received May 12, 2016; Revision accepted Nov. 10, 2016; Crosschecked Sept. 20, 2017

Abstract: We introduce a novel strategy of designing a chaotic coverage path planner for the mobile robot based on the Chebyshev map for achieving special missions. The designed chaotic path planner consists of a two-dimensional Chebyshev map which is constructed by two one-dimensional Chebyshev maps. The performance of the time sequences which are generated by the planner is improved by arcsine transformation to enhance the chaotic characteristics and uniform distribution. Then the coverage rate and randomness for achieving the special missions of the robot are enhanced. The chaotic Chebyshev system is mapped into the feasible region of the robot workplace by affine transformation. Then a universal algorithm of coverage path planning is designed for environments with obstacles. Simulation results show that the constructed chaotic path planner can avoid detection of the obstacles and the workplace boundaries, and runs safely in the feasible areas. The designed strategy is able to satisfy the requirements of randomness, coverage, and high efficiency for special missions.

Key words: Mobile robot; Chebyshev map; Chaotic; Affine transformation; Coverage path planning
https://doi.org/10.1631/FITEE.1601253 **CLC number:** TP242.6

1 Introduction

Coverage path planning is the task of determining a path that passes over all points of an area or volume of interest while avoiding obstacles (Galceran and Carreras, 2013a). This task has been used in many robotic applications, such as vacuum cleaning robots (Park *et al.*, 2012), lawn mowers (Ousingsawat and Earl, 2007), agricultural field machines (Oksanen and Visala, 2009), and underwater source exploration (Galceran and Carreras, 2013b). In the above circumstances, only the complete coverage task of the entire terrain is needed, while for some special missions, high unpredictability for the robot trajectories and fast scanning of the workplace area are strongly required, especially for military applications, such as surveillance of terrains (Martins-Filho and Macau, 2007), terrain exploration for searching (Sooraksa and Klomkarn, 2010), demining (Prado and Marques, 2013), and patrolling (Hwang *et al.*, 2011).

Chaotic systems, owing to their sensitivity to initial conditions and topological transitivity (Lorenz, 1995), provide the much-needed framework in achieving the above tasks. Recent research works (Fallahi and Leung, 2010; Curiac and Volosencu, 2012; 2015) proposed the imparting of chaotic motion behavior to the mobile robot by methods of a planner of goal position sequences or commanding the velocities of the robot's wheels based on an area-preserving chaotic map. As a consequence, the robot trajectories seem highly opportunistic and unpredictable for external observers, and the trajectories' characteristics ensure quick scanning of the patrolling

* Project supported by the National Natural Science Foundation of China (Nos. 61473179, 61573213, and 61233014), the Natural Science Foundation of Shandong Province, China (Nos. ZR2014FM007 and ZR2015CM016), and the Key Research and Development Project of Shandong Province, China (No. 2016GGX101027)

ORCID: Cai-hong LI, http://orcid.org/0000-0003-0255-9249

space. Though the random signal can also yield unpredictable trajectories, chaotic motion has a very important advantage because it is based on determinism. This means that the behavior of a robot can be predicted in advance by the system's designer.

The integration between the robot motion system and a chaotic system is a very critical step in constructing a chaotic path planner for the robot. This involves searching the dynamic systems which can demonstrate the chaotic characteristics. Researchers have already found some well-known chaotic systems to be used to impart chaotic behavior to a robot. Nakamura and Sekiguchi (2001) studied the chaotic mobile robot by the Arnold equation to achieve complete path planning. Martins-Filho and Macau (2007) proposed an ingenious path-planning mechanism where the sequence of intermediary goal positions was obtained using the well-known Chirikov-Taylor standard map. Curiac and Volosencu (2014) produced a chaotic path based on the chaotic dynamics of the Hénon system, to ensure unpredictable boundary patrol on any shape of chosen closed contour. Volos *et al.* (2012a) used three different nonlinear dynamical systems, the Chua oscillator, the Lorenz system, and a nonlinear circuit with a nonlinear resistor having an *I-V* characteristic based on a saturation function to study the chaotic motion control of a mobile robot. Later, Volos *et al.* (2013) constructed a chaotic random bit generator to provide a time-ordered succession of future robot positions, based their trajectory planning methodology on the use of the logistic map. Volos *et al.* (2012b) used a nonlinear circuit to produce the bit sequence to be converted to a sequence of planned positions. We have done some similar research work. In Li *et al.* (2013), we used arcsine and arccosine transformations to improve coverage evenness. Later, in Li *et al.* (2015), a fusion iteration strategy based on the standard map was proposed to generate a chaotic path planner of the mobile robot for surveillance missions. The proposed strategy can decrease the adjacent distances according to the divided size of the small grids, and is convenient for the robot controller to track.

In general, all of the aforementioned works can achieve fast scanning of the robot's workplace in an unpredictable way due to their chaotic characteristics. However, different chaotic systems have diverse integration methods with the robot. The coverage results produced by differently designed chaotic path planners can demonstrate various statistical characteristics with regard to the coverage rate and evenness. On the basis of our previous work, a relatively simple integration method based on the Chebyshev map of designing a chaotic coverage path planner is proposed in this study. Compared with the logistic map (Li *et al.*, 2013), the improved Chebyshev map by arcsine or arccosine transformation can access better chaotic and statistical characteristics readily. Based on the basic chaotic path planner, a universal algorithm of coverage path planning is designed for environments with obstacles using the affine transformation. The chaotic attracting traits eliminate the boundary detection of obstacles and the workplace.

2 Chebyshev map

2.1 Chaotic characteristics of the Chebyshev map

The Chebyshev map is a simple dynamic system. It is usually used in digital communications and encryption (Liu *et al.*, 2013; Gan *et al.*, 2014). Its difference equation form of degree k is

$$x_{n+1} = f(x_n) = \cos(k \cdot \arccos x_n), \tag{1}$$

where k is the order of the model, n is the iteration index, and x_n is the nth state. The parameter is very important and determines the evolving state of the system. Of course, it also determines when the system is in a state of chaos. Chaos theorists have proposed some tools (Lorenz, 1995), such as the Lyapunov exponent and bifurcation diagram. The chaotic state of the dynamic systems can be judged by them. Fig. 1a represents the bifurcation diagram of the Chebyshev map. The picture shows that with the increase of parameter k, the system gradually becomes chaotic. When the parameter which assumes an integer value is greater than 2, the system is chaotic. The chaotic system is an attractor, and the range is $x_n \in [-1, 1]$. The Lyapunov exponent is another very important index for chaotic systems. Its value indicates the chaotic degree of the system. There are n Lyapunov exponents in an N-dimensional system. If the largest one is greater than 0, the system can be considered to be chaotic. The larger the largest exponent is, the better chaotic performance the system owns. One method

for calculating the exponent can be deduced based on its definition:

$$\lambda=\lim_{N\to\infty}\frac{1}{n}\sum_{n=0}^{N-1}\ln\left|\frac{\mathrm{d}f(x_n)}{\mathrm{d}x_n}\right|. \tag{2}$$

Combining Eq. (2) with Eq. (1), we have

$$\frac{\mathrm{d}f(x_n)}{\mathrm{d}x_n}=\frac{k\cdot\sin(k\cdot\arccos x_n)}{\sqrt{1-x_n^2}}. \tag{3}$$

Fig. 1b illustrates the Lyapunov exponent spectrum of the Chebyshev map computed by Eqs. (2) and (3), through which we can conclude the same result as shown in Fig. 1a; that is, when parameter k takes an integer greater than 2, the system is chaotic. In this study we assume that the Chebyshev map is in a full chaotic state of the range [−1, 1] as k=6. Here, the Lyapunov exponent is equal to 1.7918. Another characteristic of the chaotic system is the trait of sensitive dependence on the initial values. The Chebyshev map is also very sensitive to its initial value. Only little difference of two groups' initial values can produce very different time sequences. Fig. 1c shows the two sets of time sequences of x_n and y_n originated from x_0=0.100 00 and y_0=0.100 01, respectively. After several iterative steps the two trajectories illustrate the significant differences.

2.2 Statistical characteristics of the Chebyshev map

To make the robot quickly traverse the workplace, the chaotic system should have better uniformity. The Chebyshev map is at a full mapping in the range [−1, 1] as k=6. Here, we talk about its uniform distribution qualitatively and quantitatively.

Fig. 2a demonstrates the trajectory distributions of x_n at the initial value x_0=0.1, and the iteration number is n=2000. The picture shows an uneven distribution. Near the two extreme points −1 and 1, the trajectory points are dense, while in the middle area, the points are sparse. We use the histogram (Fig. 2b) to discuss the uniform distributions further. The table shows the same distribution trait in Fig. 2a. It is unfavorable for the robot to cover the workplace quickly. So, we should take some measures to improve the uneven characteristic.

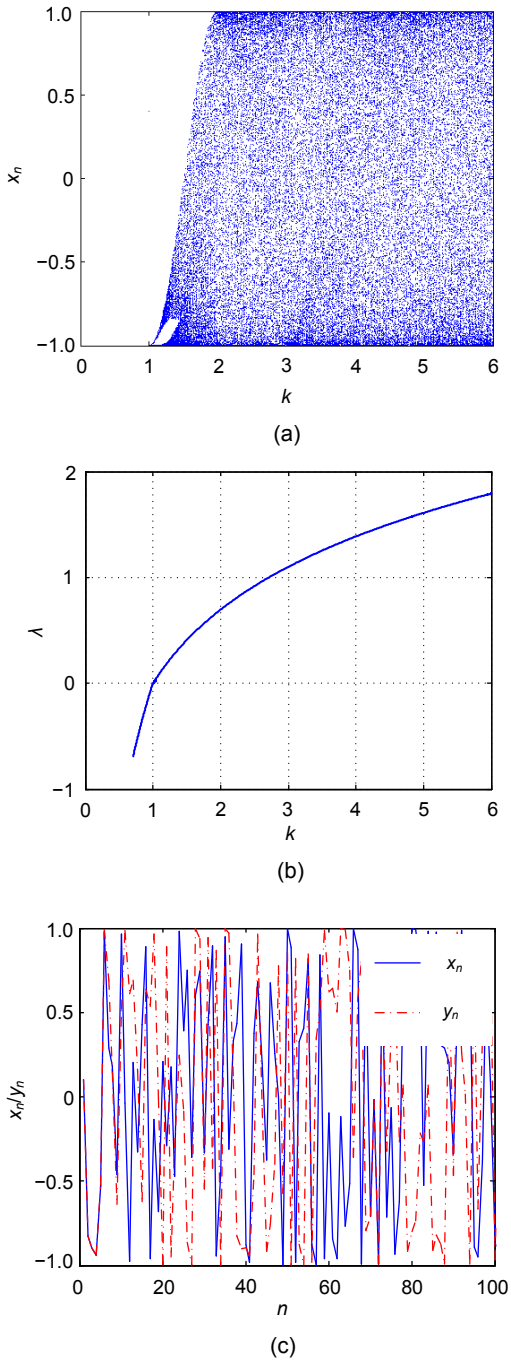


Fig. 1 Analysis of chaotic characteristics of the Chebyshev map: (a) bifurcation diagram; (b) Lyapunov exponent spectrum; (c) sensitive dependence on initial values

2.3 Improvement algorithm and analysis of the Chebyshev map

We should improve the dense distribution trait at both ends for the Chebyshev map. Here, arcsine or

arccosine transformation is used to improve the time sequences x'_n produced by Eq. (1). The arcsine transformation is

$$x_n = \frac{2\arcsin x'_n}{\pi}. \tag{4}$$

The arccosine transformation is

$$x_n = \frac{2\arccos x'_n}{\pi}. \tag{5}$$

Fig. 3 illustrates the time sequence points and histogram of the trajectory distributions of the Chebyshev map improved by the arcsine transformation. Fig. 4 shows the transformation results of the arccosine transformation. The pictures and tables show that both transformations can better the uneven distributions of the map. There is little difference between the two methods. The only distinction is the mapped range. The arcsine transformation has the same variable range as the original Chebyshev map, which is [−1, 1], while the mapped range of the arccosine transformation changes to [0, 2].

We introduce an invariant distribution of the variable difference to research further the randomness of the chaotic time sequences. The variable difference is to solve the absolute means of two groups' sampled time sequences. The smaller the variables' difference is, the better the randomness the chaotic system owns. Suppose one set of sampled time sequences is $\{x_1, x_2, \ldots, x_n\}$, and the other set is $\{y_1, y_2, \ldots, y_n\}$. The initial values are x_1=0.1, y_1=0.2, and the iteration number n is 2000 and 10 000, respectively. Then the variable difference value $v(xy)$ is as given in Eq. (6).

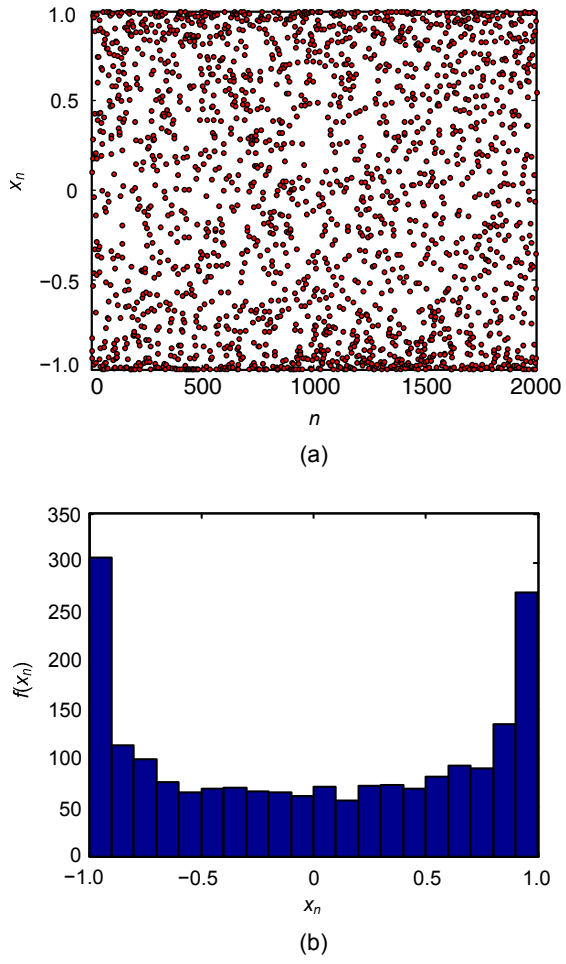


Fig. 2 Trajectory distributions of the Chebyshev map: (a) time sequence points; (b) histogram

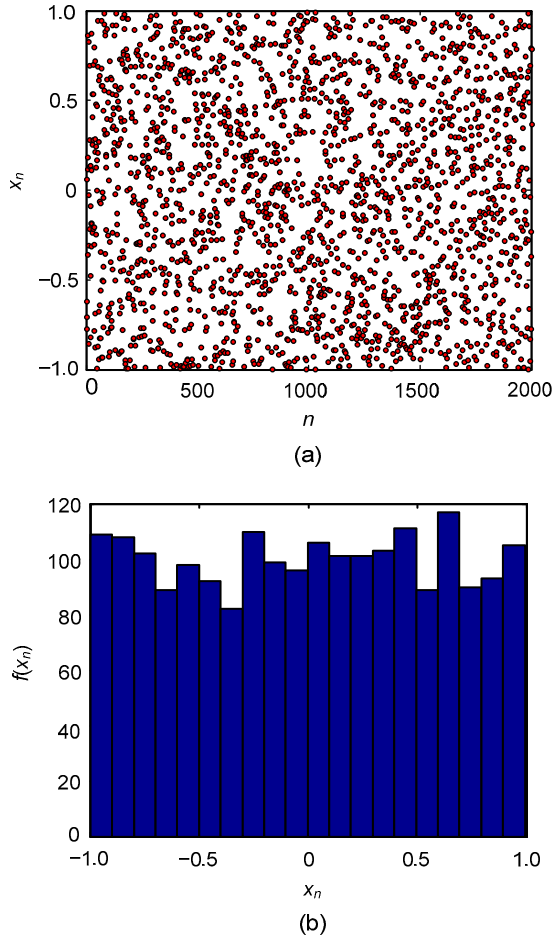
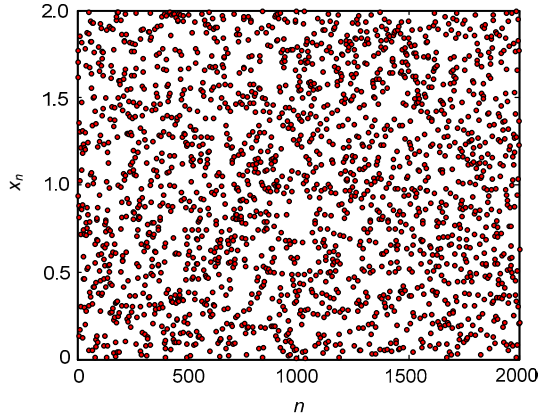
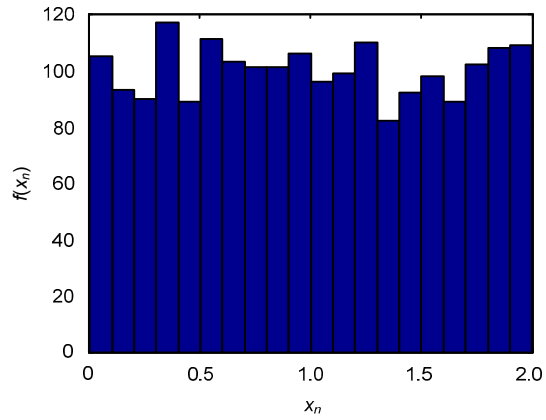


Fig. 3 Trajectory distributions of the Chebyshev map improved by arcsine transformation: (a) time sequence points; (b) histogram

$$v(xy)=\frac{|x_1-y_1|+|x_2-y_2|+...+|x_n-y_n|}{n}=\frac{1}{n}\sum_{i=1}^{n}|x_i-y_i|. \quad (6)$$



(a)



(b)

Fig. 4 Trajectory distributions of the Chebyshev map improved by arccosine transformation: (a) time sequence points; (b) histogram

Table 1 lists the values of $v(xy)$ obtained by different methods.

The data in Table 1 show that the larger the iteration number is, the smaller the $v(xy)$ value is for the same method. This proves that assuming a larger iteration number can obtain better randomness time sequences x_n, while with the same iteration number, arcsine or arccosine transformation can gain the same smaller $v(xy)$ values. This proves that it is effective to achieve better randomness by arcsine or arccosine transformation. Here, we further discuss the chaotic characteristics of time sequences x_n after arcsine transformation by Lyapunov exponent. The differential equation is

$$\frac{\mathrm{d}f(x_n)}{\mathrm{d}x_n}=\frac{2k\cdot\sin(k\cdot\arccos x_n)}{\pi\sqrt{1-[\cos(k\cdot\arccos x_n)]^2}\cdot\sqrt{1-x_n^2}}. \quad (7)$$

Substituting it into Eq. (2), the Lyapunov exponent λ can be calculated. Fig. 5 plots the Lyapunov exponent spectrum after arcsine transformation. The picture shows that the exponent increases obviously compared with Fig. 1b, where λ is equal to 3.2147 at k=6. It is far greater than λ=1.7918 before transformation. So, the arcsine transformation can also improve the chaotic characteristics of the Chebyshev map. The same result can be obtained by arccosine transformation. In conclusion, these two transformations have the same chaotic characteristics, and any of them can be chosen to be the transformation tool. In this study we adopt arcsine transformation to improve the characteristics of the Chebyshev map.

Table 1 Values of $v(xy)$ by different methods

Method	$v(xy)$	
	n=2000	n=10 000
Original Chebyshev map	0.8309	0.8209
By arcsine transformation	0.6830	0.6758
By arccosine transformation	0.6830	0.6758

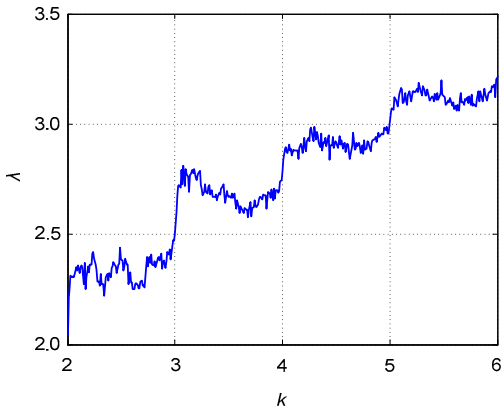


Fig. 5 Lyapunov exponent spectrum after arcsine transformation

3 Chaotic coverage path planner of the mobile robot

We use two one-dimensional (1D) Chebyshev maps to construct a two-dimensional (2D) dynamic

system. It is as follows:

$$\begin{cases} x_{n+1} = \cos(k \cdot \arccos x_n), \\ y_{n+1} = \cos(k \cdot \arccos y_n). \end{cases} \tag{8}$$

The time sequences (x_n, y_n) produced by Eq. (8) can be directly mapped to the sub-goal positions of the robot as the common 2D dynamic systems do. Because of the sensitive dependence characteristic on the initial value of the chaotic system (Fig. 1c), here Eq. (8) can also produce two completely different trajectories if the two values of the starting points x_1 and y_1 are not exactly the same. Fig. 6 plots the robot's sub-goals of the trajectories at x_1=0.100 00 and y_1=0.100 01, at n=1000 and n=2000, respectively.

Fig. 6 shows that the robot trajectories produced by Eq. (8) can traverse the whole workplace after a limited number of iterations. So, the path planner can satisfy the coverage characteristic for special missions. The constructed path planner has the same uneven distribution characteristic as the 1D Chebyshev map. Thus, the arcsine transformation mentioned in Section 2.3 is used to improve the unevenness of the trajectories. The formula is as follows:

$$\begin{cases} x_{n+1} = \dfrac{2\arcsin[\cos(k \cdot \arccos x_n)]}{\pi}, \\ y_{n+1} = \dfrac{2\arcsin[\cos(k \cdot \arccos y_n)]}{\pi}. \end{cases} \tag{9}$$

The planning efficiency is also improved. Fig. 7 shows the improvement results. As the 1D Chebyshev map does, the constructed chaotic path planner can obtain better evenness, randomness, and chaotic characteristics by arcsine transformation. So, it can satisfy the requirements of special missions for the

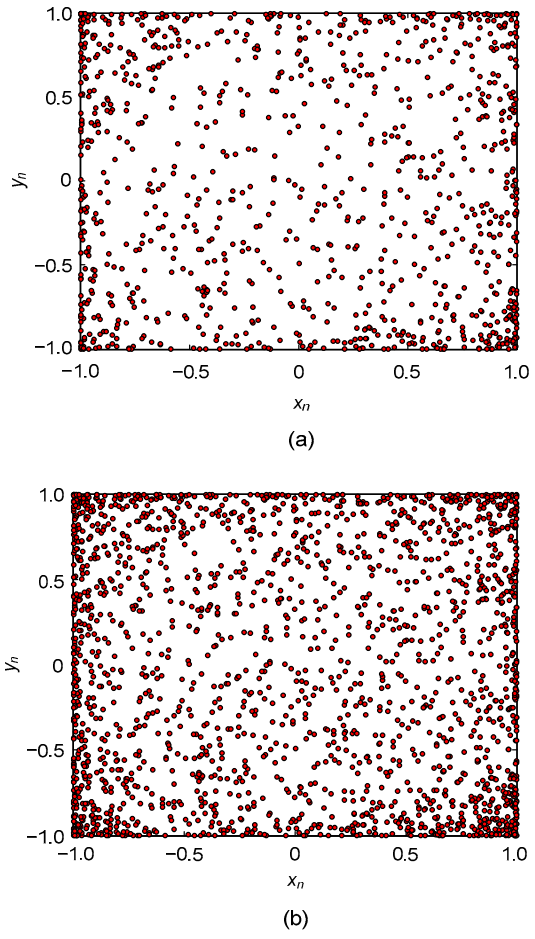


Fig. 6 Sub-goals of the trajectories of the chaotic path planner: (a) n=1000; (b) n=2000

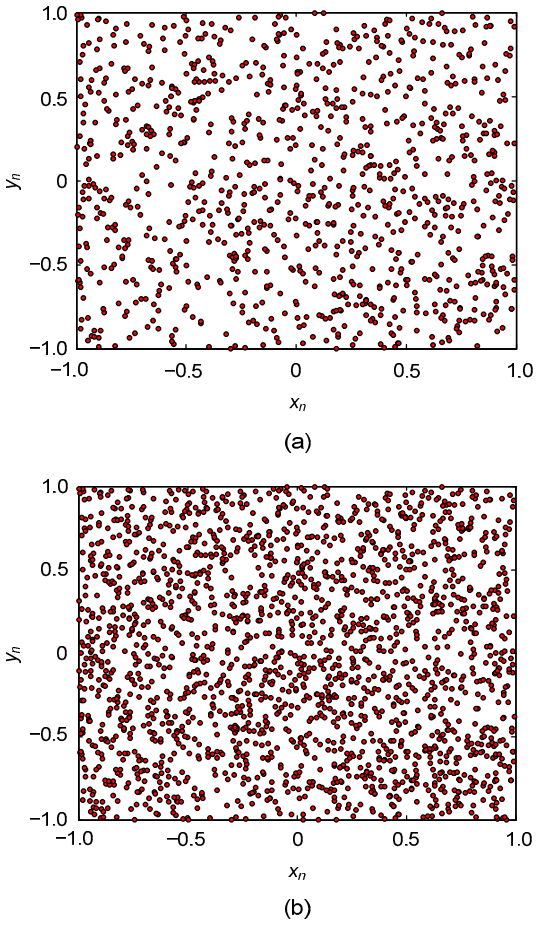


Fig. 7 Trajectories of the chaotic path planner of the robot after arcsine transformation: (a) n=1000; (b) n=2000

mobile robot. The running range of the robot is limited to $-1 \le x \le 1$ and $-1 \le y \le 1$ according to Eq. (9). We can use the translation and scaling affine transformations to change the robot workplace range to any arbitrary size. The mapping formula is

$$\begin{cases} x_n' = a + c \cdot x_n, \\ y_n' = b + d \cdot y_n. \end{cases} \tag{10}$$

Here (a, b) are the translation parameters, and (c, d) are the scaling ones. Supposing the robot workplace range is $0 \le x \le 10$ and $0 \le y \le 10$, then (a, b)=(5, 5), (c, d)=(5, 5), and Eq. (9) is mapped into the above range. Fig. 8a plots the results after mapping. Because the chaotic system is an attractor, any point in the workplace can iterate in the range forever. Any point near the boundaries can be attracted in the range after several iterations. So, as long as the chaotic path planner starts in the workplace, the robot can never run out of the range. The boundaries need no detection as the robot is running. A basic model of the chaotic path planner is constructed by Eqs. (9) and (10) as follows:

$$\begin{cases} x_n = \dfrac{2\arcsin[\cos(6 \cdot \arccos x_{n-1})]}{\pi}, \\ y_n = \dfrac{2\arcsin[\cos(6 \cdot \arccos y_{n-1})]}{\pi}, \\ x_n' = a + c \cdot x_n, \\ y_n' = b + d \cdot y_n. \end{cases} \tag{11}$$

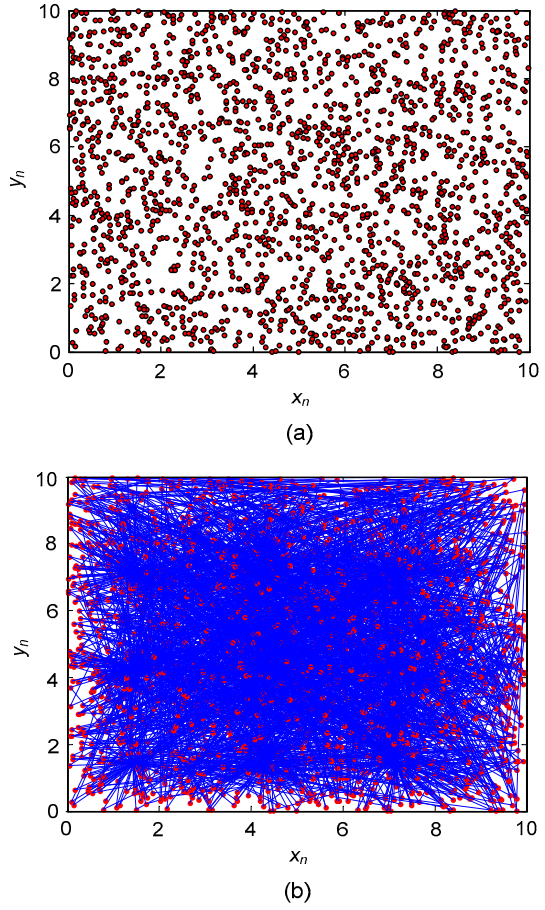


Fig. 8 Trajectories of the robot in the running range $0 \le x \le 10$ and $0 \le y \le 10$, at n=2000: (a) sub-goals of the trajectories; (b) trajectories

4 Coverage path planning in environments without obstacles

When there are no obstacles in the workplace, we can link only the adjacent sub-goals as the running trajectories for robot tracking (Fig. 8b). If the linkage distances of the adjacent sub-goals are sufficiently large, the robot will find it difficult to track the trajectories for the limitation of the robot's each moving step distance. So, the interpolation computations are needed between the adjacent sub-goals. If the robot does not need continuous running, it can stop while arriving at a sub-goal and turns to another one. Now only linear interpolations are needed. Fig. 9a illustrates this circumstance.

Points A, B, and C are supposed to be the planned sub-goals. Then some points need to be inserted between the two sub-goals with equal intervals according to the robot's moving step distance. If the robot needs continuous running, taking into account the mechanical characteristic of the robot, arc interpolation is required at each inflection point, such as point B. Arc EF replaces the straight lines EB and BF (Fig. 9b). Fig. 9c illustrates the complete interpolation result for the continuous running of the robot. Here we discuss mainly the planning algorithm; for simplicity, only linear interpolation is discussed.

When running in real environments, the robot also needs to consider its physical size to avoid collisions with obstacles and the workplace boundaries. Here the robot shape is approximated to be circular. Then the edge expansion method is used to expand both the workplace boundaries and the obstacle edges with the robot radius size to improve the operational

safety of the robot. Correspondingly, the feasible range of the workplace is reduced. Suppose the radius of the robot is 0.1. Then a workplace for $x\in[0, 10]$, $y\in[0, 10]$, is reduced to $x\in[0.1, 9.9]$, $y\in[0.1, 9.9]$. The mapping parameters in Eq. (11) have to be repaired according to the running range variety. Still no detection of the workplace boundaries is needed after edge expansion.

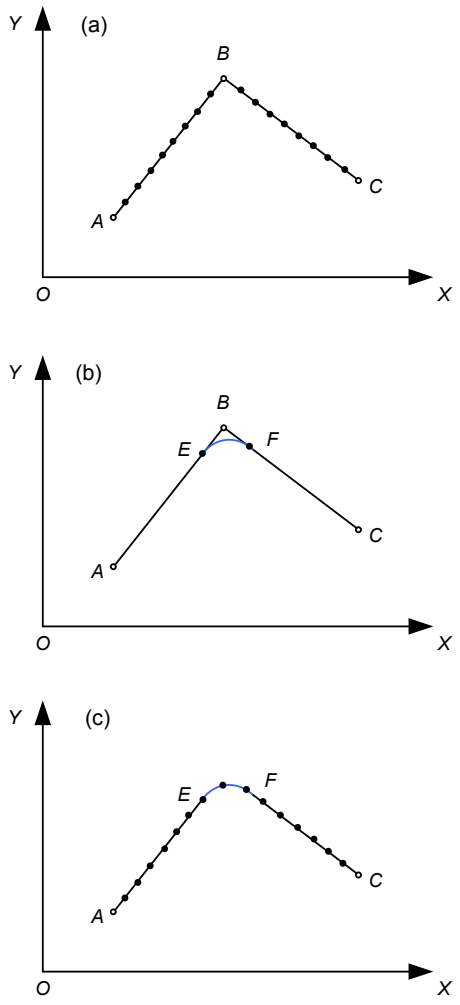


Fig. 9 Interpolation computations between two adjacent sub-goals: (a) linear interpolations; (b) arc interpolation; (c) interpolations for a continuous running

Fig. 10 shows the planned trajectories after edge expansion. To see the simulation results clearly, a relatively small iteration number n is used, and here n=200. The robot radius is 0.1, and the robot step distance is 0.125. Fig. 10a plots only the planned sub-goals. Fig. 10b shows the linkages of the adjacent sub-goals that form the running trajectories of the robot. The true trajectories based on the considerations of interpolations and the robot size are illustrated in Fig. 10c. In Figs. 10a and 10b, the trajectories iterate in the workplace and do not run out of the boundaries and the expansion edges. In Fig. 10c, some trajectories run out of the expansion edges, but not out of the workplace boundaries. This shows that

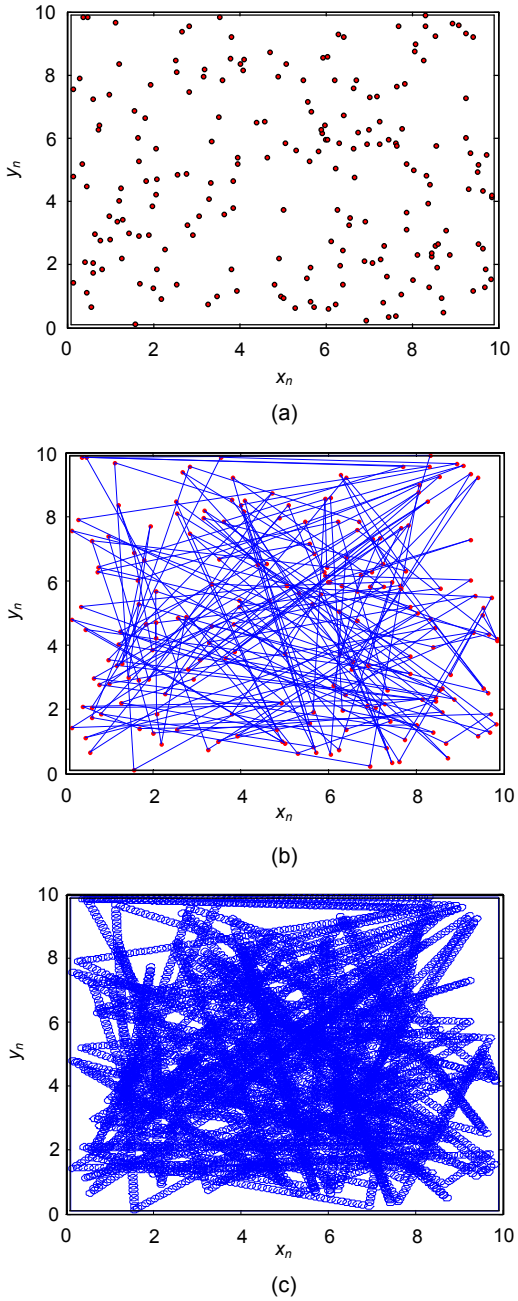


Fig. 10 Robot trajectories after edge expansion: (a) planned sub-goals; (b) planned trajectories; (c) real trajectories

the consideration of edge expansion is effective and provides safety for the running of the robot.

5 Coverage path planning in environments with obstacles

If there exist obstacles in the workplace, of whichever shape the obstacle is, they can be converted and approximated to be several rectangles. Then the feasible area for a robot passing through in the workplace can be divided into some adjacent rectangles. Coverage path planning is achieved in the workplace based on the above-mentioned divisions, the iteration computations, and the affine transformation (Eq. (11)). A novel general algorithm for coverage path planning based on the above ideas is presented in this study. The algorithm includes mainly two parts: one is environment modeling, and the other is the design of the coverage path planning method. The detailed procedures are given as follows.

5.1 Environment modeling

The shapes of all the obstacles in the environment are assumed to be rectangles or converted and approximated to be rectangles. Then the feasible areas in the workplace, here labeled as S_Ω, are divided into some small rectangles to construct the feasible coverage rectangles and the transition rectangles for environment modeling. The transition rectangles are the transiting areas for the robot to run from one feasible coverage rectangle to another. The detailed construction process of environment modeling is as follows:

1. The feasible areas S_Ω in the workplace are divided into some small adjacent rectangles, here labeled as S_z, along the boundaries of obstacles and the workplace.

2. The adjacent rectangles S_z divided above are merged into some relatively large feasible coverage rectangles, here labeled as S_{I_x}, where x expresses the merged rectangles.

3. A transition rectangle is constructed by the overlapping area between two adjacent rectangles S_{I_x}, here labeled as S_{T_y}, where y means the overlapping area.

The details about the division in step 1 are as follows:

The vertical and horizontal segment straight lines extend to both sides along the obstacles' boundaries until they meet the borders of the workplace. The segment lines of one obstacle can cross over other obstacles for simplifying the division procedure. Then the small adjacent rectangles S_z can be obtained by the crossing areas which are produced by the segment lines in the feasible areas. The number of S_z is related to the number of obstacles and their positions in the workplace.

The merged method mentioned in step 2 is realized by the recursive algorithm. First, a special rectangle, called a corner rectangle, is defined to increase the search speed of the algorithm.

Definition 1 (Corner rectangle) Suppose each small adjacent rectangle S_z is expressed by its lower-left coordinates which are assumed to be (x_{S_z}, y_{S_z}). Then the S_z that has the maximum value of x_{S_z} or y_{S_z}, or has the minimum value of x_{S_z} or y_{S_z}, is called a corner rectangle, here labeled as S_{ez}.

Because S_{ez} is at the corner, other rectangles can locate only at its two sides instead of four. Then the search speed of the adjacent rectangles S_{I_z} can be improved.

The construction procedure of S_{I_z} is as follows:

1. Find out all the corner rectangles S_{ez} in the feasible area S_Ω.

2. For each S_{ez}, find out the largest rectangle that includes it. If there exist many, choose one of them randomly. Then it is one of the S_{I_z} that we are looking for.

3. Update the feasible area S_Ω by deleting the new S_{I_z}.

4. Judge whether S_Ω is empty or not. If yes, the construction procedure of S_{I_z} ends; otherwise, jump to step 1.

The robot workplace is then modeled by the feasible coverage rectangles S_{I_x} and the transition rectangles S_{T_y}. The coverage path planning is designed according to the above constructed environment.

5.2 Design of coverage path planning

The design purpose of the algorithm is to fulfill the coverage path planning of each feasible coverage rectangle S_{I_x} by the transition rectangles S_{T_y}, further to achieve the whole feasible area coverage, and to keep the chaotic characteristics of the path planner for the requirements of special missions. A large loop is

achieved when all the feasible coverage rectangles S_{I_x} have fulfilled the coverage task once. The large loop times can be set according to the task requirements. The strategy of coverage path planning is as follows:

1. Solve all the affine transformation coefficients of each S_{I_x} and S_{T_y} according to Eq. (11).

2. Initialize the iteration number N of the large loop and the iteration number n of each S_{I_x} in every large loop. The value of the iteration number n can be set according to the size of each S_{I_x} area.

3. Select an initial starting point arbitrarily in a feasible coverage rectangle S_{I_x}, and start the iteration process according to Eq. (11) with the corresponding affine transformation coefficients computed in step 1.

4. Iterate in the current area S_{I_x} until the corresponding iteration number n completes.

5. Iterate in area S_{I_x} once and judge whether the robot enters the overlapping area of the adjacent transition rectangle S_{T_y} or not. If not, repeat the process until the aforementioned condition attains.

6. Start an iteration process in S_{T_y} and judge whether the robot enters another adjacent area S_{I_x} or not. Execute the above process until the condition has been satisfied.

7. Start a new iteration in the present area S_{I_x}. The next processes are similar to steps 4–6.

8. When all the feasible coverage rectangles S_{I_x} have completed one iteration process, then the whole feasible areas in the workplace have been covered once, or a large loop completes.

9. Repeat the above processes until the iteration number N of the large loop completes.

5.3 Characteristics of environment modeling and the designed algorithm

Coverage path planning is achieved based on the modeled environment. The characteristics of the proposed method are as follows:

1. The division method of the feasible areas in the robot workplace along the boundaries of obstacles and the workplace can let the robot run from one feasible coverage rectangle to another area directly, and will not collide with obstacles and workplace boundaries.

2. By the transition rectangles S_{T_y}, the robot can run from a feasible coverage rectangle S_{I_x} to another one automatically. Then the whole feasible area coverage of the workplace is achieved.

3. By affine transformation, the chaotic path planner of Eq. (11) iterates in each S_{I_x}, and keeps the whole workplace chaotic characteristics relatively unchanged.

4. The iterative steps in the transition rectangles S_{T_y} are limited, and have little influence for the uniformity and chaotic characteristics of the whole workplace, especially when the iteration number is very large in the feasible coverage areas.

5. The initial starting point can be selected arbitrarily in the feasible areas; namely, coverage path planning can start from any S_{I_x}.

5.4 Verification of the proposed algorithm

Here, we take two cases to verify the proposed coverage path planning method. For simplicity, we select the most common obstacles to illustrate the feasibility of the algorithm.

Case 1: As shown in Fig. 11a, the workplace is 10×10. There is a dark gray obstacle in the center of the workplace, whose size is 5×5. So, the requirement is to compute the chaotic coverage path planning trajectories in the blank or the feasible areas.

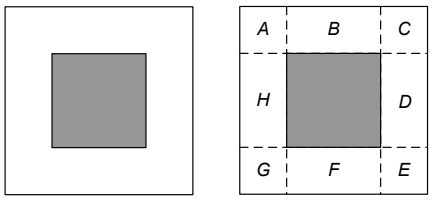


Fig. 11 An environment with one obstacle: (a) original workplace; (b) environment divisions

First, we should model the environment of Fig. 11a based on the method presented in Section 5.1. Then chaotic coverage path planning is designed according to the algorithm proposed in Section 5.2. The procedure of environment modeling is as follows:

1. As shown in Fig. 11b, the feasible areas in the workplace are divided into eight small adjacent rectangles, *A*-*B*-*C*-*D*-*E*-*F*-*G*-*H*, along the boundaries of obstacles and the workplace, labeled as S_A, S_B, S_C, S_D, S_E, S_F, S_G, and S_H, respectively.

2. Merge the eight small rectangles into four feasible coverage rectangles S_{I_x} based on the merged method mentioned in Section 5.1, here marked as

S_{I_ABC}, S_{I_D}, S_{I_EFG}, and S_{I_H}, respectively.

3. Four transition rectangles S_{T_y} are constructed, which are S_{T_CD}, S_{T_DE}, S_{T_GH}, and S_{T_HA}.

Then the coverage path planning algorithm is designed based on the modeled environment. The strategy flowchart for one large loop in the workplace is shown in Fig. 12 based on the above-mentioned method in Section 5.2. To have a better uniformity of the coverage, the iteration numbers for S_{I_D} and S_{I_H} are set to about 1/2 of those for the other two feasible coverage rectangles S_{I_ABC} and S_{I_EFG} according to their sizes of area.

In Fig. 12, the algorithm begins a large loop iteration from the center dark gray part, whose function is to set the iteration number of each of the feasible coverage rectangles S_{I_x}, and to select a starting point in the feasible areas arbitrarily. Then according to the position of the starting point, the robot enters one of the four feasible coverage rectangles (S_{I_ABC}, S_{I_D}, S_{I_EFG}, and S_{I_H}), which are marked as the blue parts in the picture, and starts the iteration procedure. When all the feasible coverage rectangles have fulfilled the coverage task, a large loop ends. Several large loops need only entail executing the iteration procedure several times. Direction of the solid arrows in the figure expresses the execution direction of the algorithm, while the dotted arrows indicate the coverage direction of the four feasible coverage rectangles.

The simulation results for one large loop of the designed algorithm are shown in Fig. 13. The total iteration number in a loop is 200, namely, n=50 in Fig. 12. Fig. 14 shows the simulation results for 10 large loops of Fig. 13.

The selected starting point is (1, 8) locating in S_{I_ABC}. Then the robot runs from area S_{I_ABC} and executes the coverage path planning task according to the initial conditions. Edge expansion of both workplace boundaries and obstacles has been considered in the simulations. The parameter setting of the robot and the processing method of edge expansion are the same as in Section 4.

From Figs. 13 and 14, we can see that the planned trajectories can cover the feasible areas, do not run over the boundaries and the expansion edges, and have no collisions with obstacles. In Figs. 13c and 14c, some trajectories have crossed over the expansion edges, but not the real boundaries of the workplace and obstacles. This shows that the algorithm for

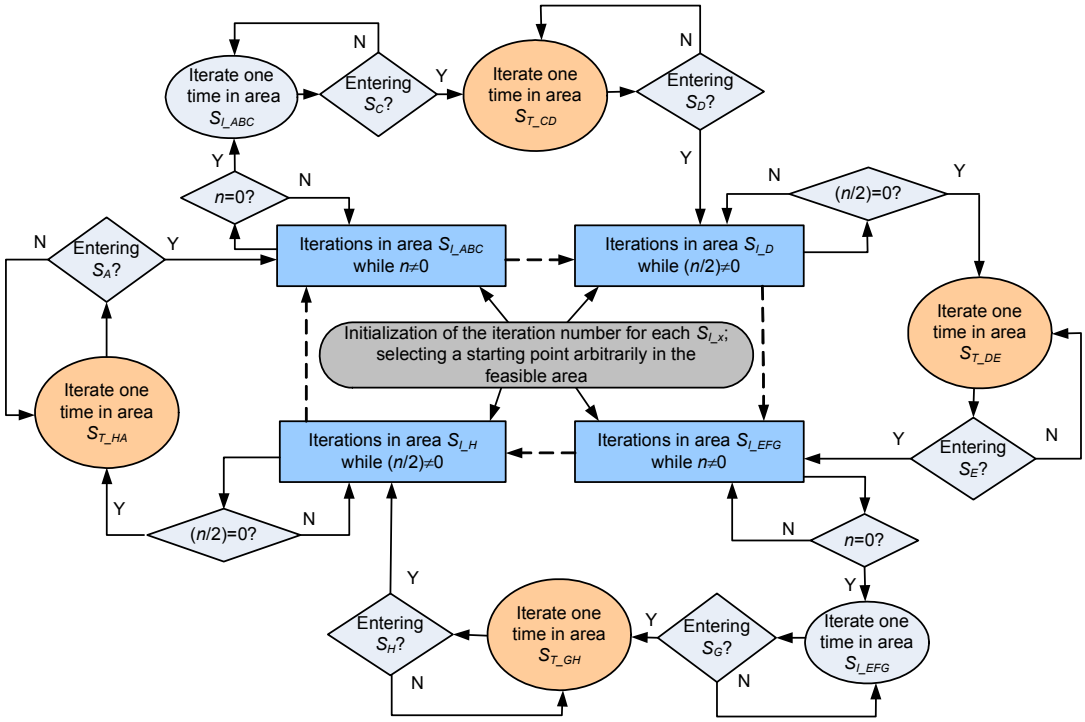


Fig. 12 Flowchart of the coverage path planning algorithm for one large loop iteration for case 1 (References to color refer to the online version of this figure)

coverage path planning is effective and safe for the robot.

Case 2: As shown in Fig. 15a, the size of the workplace is still 10×10. There are two dark gray obstacles, whose sizes are 3×3, in the workplace. The requirement is the same as in case 1.

First, we again model the environment. The details are as follows:

1. As shown in Fig. 15b, the feasible areas in the workplace are divided into seven small adjacent

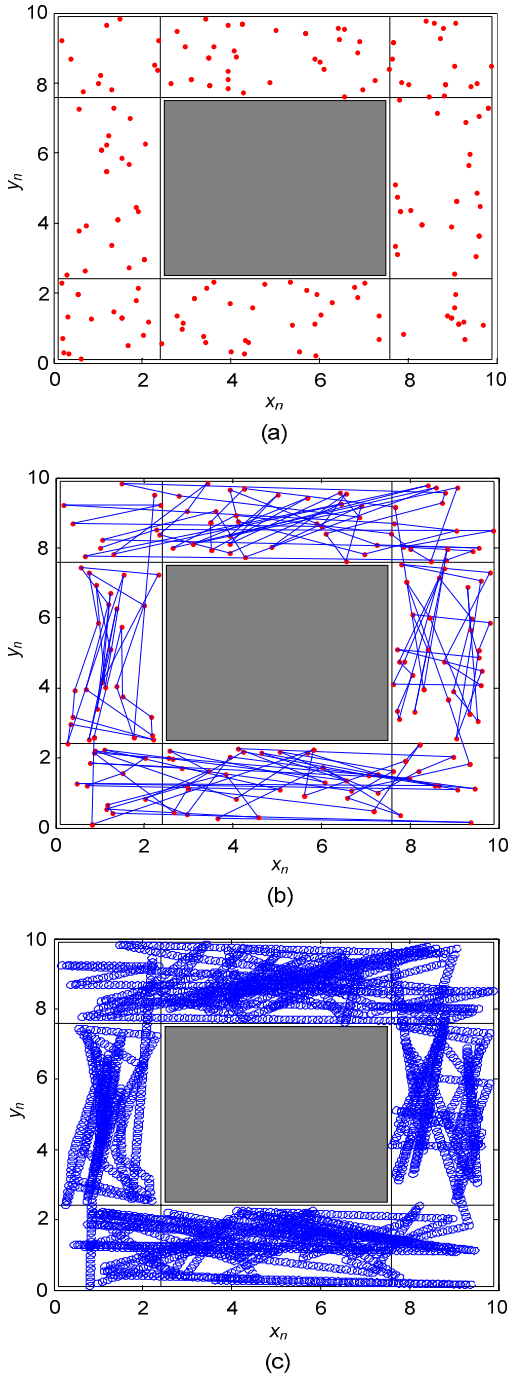


Fig. 13 Simulation results of the designed algorithm for one large loop for case 1: (a) planned sub-goals; (b) planned trajectories; (c) real trajectories

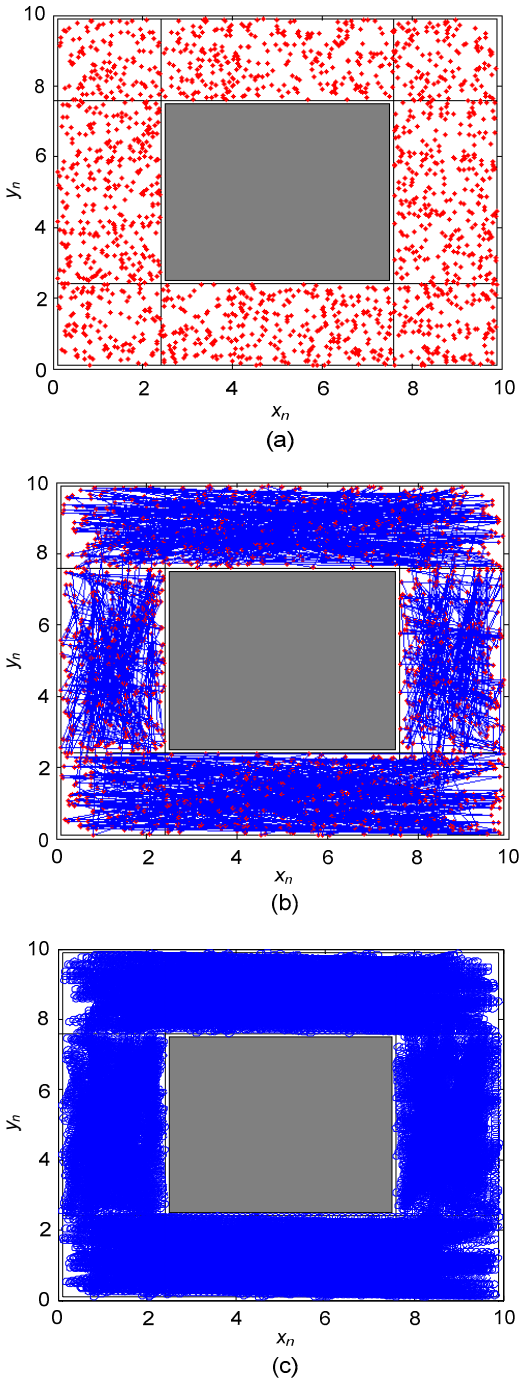


Fig. 14 Simulation results of the designed algorithm for 10 large loops for case 1: (a) planned sub-goals; (b) planned trajectories; (c) real trajectories

rectangles, *A*-*B*-*C*-*D*-*E*-*F*-*G*, along the boundaries of obstacles and the workplace, labeled as S_A, S_B, S_C, S_D, S_E, S_F, and S_G, respectively.

2. Merge the seven small rectangles into three feasible coverage rectangles S_{I_x}, which are S_{I_ABC}, S_{I_D}, and S_{I_EFG}.

3. Three transition rectangles S_{T_y} are constructed, which are S_{T_BD}, S_{T_DF}, and S_{T_BDF}.

Then the coverage path planning algorithm is designed based on the modeled environment. The strategy flowchart for one large loop with two obstacles in the workplace is shown in Fig. 16. The iteration number of S_{I_D} is set at about 1/3 of that of the other two feasible coverage rectangles S_{I_ABC} and S_{I_EFG} according to its area.

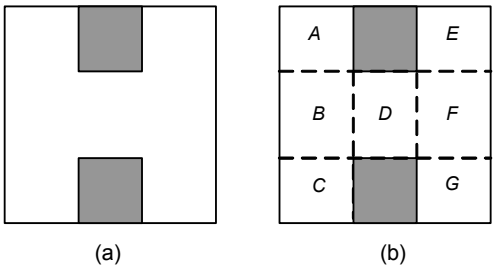


Fig. 15 An environment with two obstacles: (a) original workplace; (b) environment divisions

Fig. 17 provides the simulation results for 10 large loops. The total iteration number in each large loop is 210 (for n=90). The starting point is still (1, 8). From the picture we can obtain the same conclusions as for case 1. This proves that the proposed novel general algorithm for coverage path planning is effective for the workplace with obstacles.

6 Conclusions

In this paper, we have discussed a design strategy of a chaotic coverage path planner for the mobile robot for special missions using a 2D Chebyshev map, which is constructed by two 1D Chebyshev maps. A universal algorithm is constructed based on the above-mentioned strategy for coverage path planning in environments with obstacles and circumstances without obstacles. Characteristics of the method are as follows:

1. The simulation results of the two cases demonstrate that the robot can fulfill the coverage

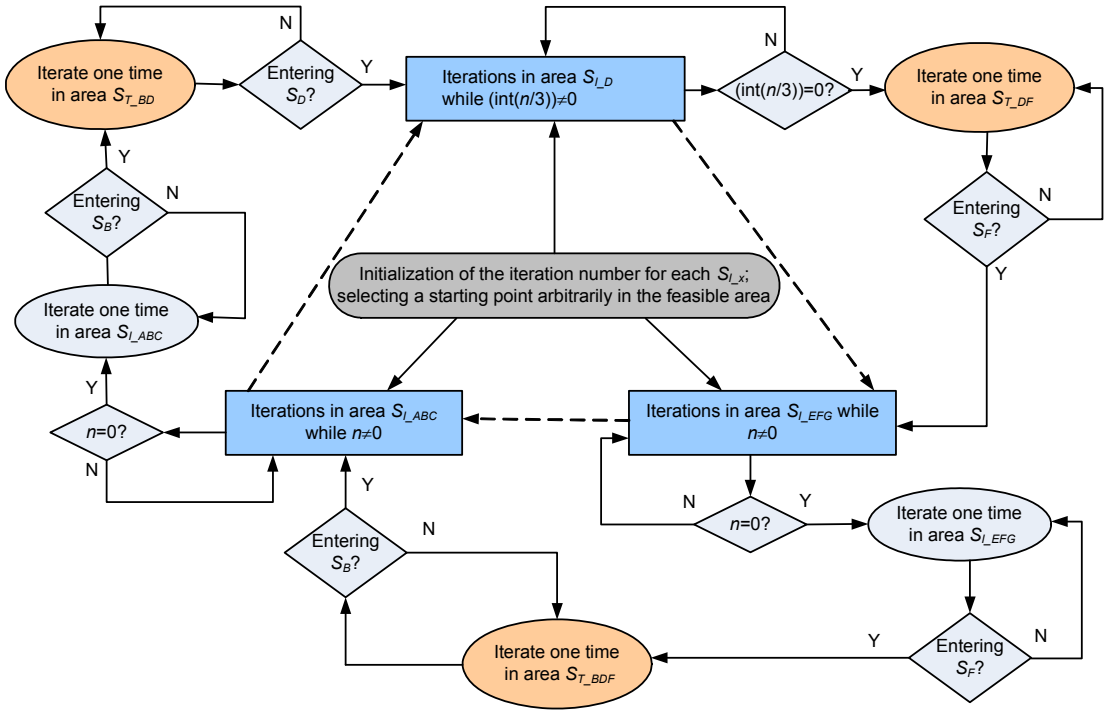


Fig. 16 Flowchart of the coverage path planning algorithm for one large loop iteration for case 2 (References to color refer to the online version of this figure)

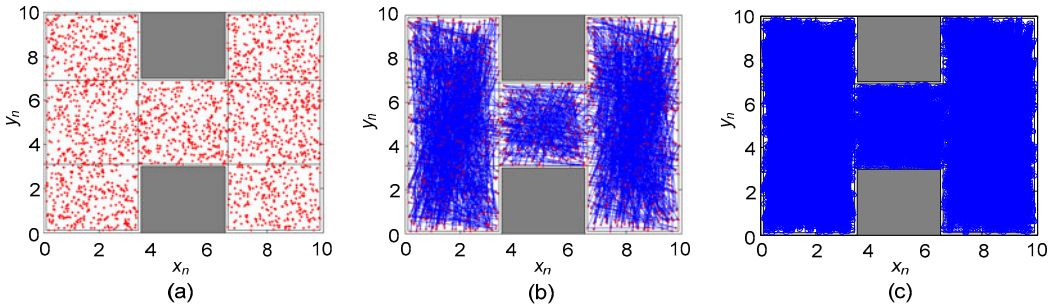


Fig. 17 Simulation results of the designed algorithm for case 2: (a) planned sub-goals; (b) planned trajectories; (c) real trajectories

task safely and uniformly in the environment with obstacles.

2. By affine transformation, the path planner executes chaotic iterations in each feasible coverage area. The design of the algorithm has little influence on the Chebyshev map. So, it can still satisfy the requirements of the special missions.

3. Because the chaotic path planner is an attractor, it iterates only in the feasible coverage areas. Thus, no detection of the boundaries of the workplace and obstacles is needed.

4. The simulation results show that the arcsine or arccosine transformation of the Chebyshev map is effective for improving the evenness of the workplace. So, the coverage rate and efficiency have been improved.

5. By an interpolation and the edge expansion method, the planning trajectories can be passed to the real robot to track in the real environment.

6. This designed strategy of chaotic coverage path planning can also be extended to other 1D or 2D chaotic systems.

It is inevitable to sacrifice some chaotic characteristics of the Chebyshev map for avoiding obstacles in an environment including obstacles. In this study, the whole coverage task has been achieved by covering each feasible area according to a scheduled sequence. Then the coverage and uniformity performance for the special missions can be guaranteed. Because the starting points in each iteration loop are different, the randomness of the covering trajectories can be improved by increasing the transiting frequency between the adjacent feasible areas in the constant iteration numbers, namely decreasing the iteration numbers in one large loop, meanwhile increasing the large loop times.

In the future, there is still much work to do, such as the research on quantitative methods of converting irregularly sized obstacles to be rectangles in the running workplace, and the research methods about chaotic and statistical characteristics for other chaotic systems and a chaotic coverage path planner.

References

Curiac, D.I., Volosencu, C., 2012. Chaotic trajectory design for monitoring an arbitrary number of specified locations using points of interest. *Math. Probl. Eng.*, **2012**: 940276.1-940276.18. https://doi.org/10.1155/2012/940276

Curiac, D.I., Volosencu, C., 2014. A 2D chaotic path planning for mobile robots accomplishing boundary surveillance missions in adversarial conditions. *Commun. Nonl. Sci. Numer. Simul.*, **19**(10):3617-3627. https://doi.org/10.1016/j.cnsns.2014.03.020

Curiac, D.I., Volosencu, C., 2015. Path planning algorithm based on Arnold cat map for surveillance UAVs. *Defen. Sci. J.*, **65**(6):483-488. https://doi.org/10.14429/dsj.65.8483

Fallahi, K., Leung, H., 2010. A cooperative mobile robot task assignment and coverage planning based on chaos synchronization. *Int. J. Bifurc. Chaos*, **20**(1):161-176. https://doi.org/10.1142/S021812741002548X

Galceran, E., Carreras, M., 2013a. Planning coverage paths on bathymetric maps for in-detail inspection of the ocean floor. Proc. IEEE Int. Conf. on Robotics and Automation, p.4159-4164. https://doi.org/10.1109/ICRA.2013.6631164

Galceran, E., Carreras, M., 2013b. A survey on coverage path planning for robotics. *Robot. Auton. Syst.*, **61**(12):1258-1276. https://doi.org/10.1016/j.robot.2013.09.004

Gan, H.P., Li, Z., Li, J., *et al.*, 2014. Compressive sensing using chaotic sequence based on Chebyshev map. *Nonl. Dynam.*, **78**(4):2429-2438. https://doi.org/10.1007/s11071-014-1600-1

Hwang, K.S., Lin, J.L., Huang, H.L., 2011. Dynamic patrol planning in a cooperative multi-robot system. Proc. 14th

FIRA RoboWorld Congress, p.116-123. https://doi.org/10.1007/978-3-642-23147-6_14

Li, C.H., Wang, F.Y., Zhao, L., *et al.*, 2013. An improved chaotic motion path planner for autonomous mobile robots based on logistic map. *Int. J. Adv. Robot. Syst.*, **10**:1-9. https://doi.org/10.5772/56587

Li, C.H., Song, Y., Wang, F.Y., *et al.*, 2015. Chaotic path planner of autonomous mobile robots based on the standard map for surveillance missions. *Math. Probl. Eng.*, **2015**:263964.1-263964.11. https://doi.org/10.1155/2015/263964

Liu, S.C., Song, Y.X., Yu, R.H., 2013. The application of oversampled Chebyshev chaotic sequences in voice communication encryption. *Adv. Mater. Res.*, **655-657**: 1745-1749. https://doi.org/10.4028/www.scientific.net/AMR.655-657.1745

Lorenz, E.N., 1995. The Essence of Chaos. CRC Press.

Martins-Filho, L.S., Macau, E.E.N., 2007. Patrol mobile robots and chaotic trajectories. *Math. Probl. Eng.*, **2007**: 61543.1-61543.13. https://doi.org/10.1155/2007/61543

Nakamura, Y., Sekiguchi, A., 2001. The chaotic mobile robot. *IEEE Trans. Robot. Autom.*, **17**(6):898-904. https://doi.org/10.1109/70.976022

Oksanen, T., Visala, A., 2009. Coverage path planning algorithms for agricultural field machines. *J. Field Robot.*, **26**(8):651-668. https://doi.org/10.1002/rob.20300

Ousingsawat, J., Earl, M.G., 2007. Modified lawn-mower search pattern for areas comprised of weighted regions. Proc. American Control Conf., p.918-923. https://doi.org/10.1109/ACC.2007.4282850

Park, E., Kim, K.J., del Pobil, A.P., 2012. Energy efficient complete coverage path planning for vacuum cleaning robots. *In*: Park, J., Leung, V., Wang, C.L., *et al.* (Eds.), Future Information Technology, Application, and Service, p.23-31. https://doi.org/10.1007/978-94-007-4516-2_3

Prado, J., Marques, L., 2013. Energy efficient area coverage for an autonomous demining robot. Proc. 1st Iberian Robotics Conf., p.459-471. https://doi.org/10.1007/978-3-319-03653-3_34

Sooraksa, P., Klomkarn, K., 2010. "No-CPU" chaotic robots: from classroom to commerce. *IEEE Circ. Syst. Mag.*, **10**(1):46-53. https://doi.org/10.1109/MCAS.2010.935740

Volos, C.K., Bardis, N.G., Kyprianidis, I.M., *et al*., 2012a. Implementation of mobile robot by using double-scroll chaotic attractors. Recent Researches in Applications of Electrical and Computer Engineering, p.119-124. http://www.wseas.us/e-library/conferences/2012/Vouliagmeni/ACA/ACA-18.pdf

Volos, C.K., Kyprianidis, I.M., Stouboulos, I.N., 2012b. A chaotic path planning generator for autonomous mobile robots. *Robot. Auton. Syst.*, **60**(4):651-656. https://doi.org/10.1016/j.robot.2012.01.001

Volos, C.K., Kyprianidis, I.M., Stouboulos, I.N., 2013. Experimental investigation on coverage performance of a chaotic autonomous mobile robot. *Robot. Auton. Syst.*, **61**(12):1314-1322. https://doi.org/10.1016/j.robot.2013.08.004