Robotica (2014) volume 32, pp. 743–756. © Cambridge University Press 2013
doi:10.1017/S0263574713000970

Cognitive response navigation algorithm for mobile robots using biological antennas

Jiliang Jiang, Dawei Tu*, Shuo Xu and Qijie Zhao

School of Mechatronic Engineering and Automation, and Shanghai Key Laboratory of Manufacturing Automation and Robotics, Shanghai University, Shanghai 200072, China

(Accepted October 28, 2013. First published online: December 3, 2013)

SUMMARY

We present BioBug, a bionic cognitive response navigation algorithm for mobile robots based on neuroethology principles. It includes a biological antenna model for environment perception and an improved Bug algorithm for motion planning and control. The biological antenna model delineates the interested sensing areas, and thus decreases the computational burden. Then, this obtained environment stimulation is responded to generate the corresponding walking behavior according to BioBug. Simulations and experiments have been carried out in different conditions of obstacle density and boundary shape through algorithm comparisons. Compared with the competitors, BioBug is characterized by not only a smaller path length, but also shorter time for obstacle escape. The results demonstrate the practicality, environment robustness, and obstacle avoidance efficiency of the algorithm.

KEYWORDS: Bug algorithm; Autonomous navigation; Mobile robot; Biological antenna; Cognition; Bionics.

1. Introduction

Autonomous navigation of automated mobile robots (AMRs) and automated guided vehicles (AGVs) in unknown environments basically requires finding a safe path from a starting position to a target position without human intervention. Many navigation algorithms emerge in this research, e.g., artificial potential field (APF) method,[1] vector field histogram (VFH),[2] fuzzy logic control approach,[3] behavior control method,[4] neural network algorithm,[5] genetic algorithm,[6] Bug algorithm, etc. As a famous reactive algorithm, the prototype Bug algorithm was firstly proposed by Lumelsky *et al.* in 1987,[7,8] and later extended to an algorithm cluster in order to improve navigation path in aspects of safety, efficiency, smoothness, and so on. Some representatives include Bug1,[9] Bug2,[9] VisBug,[10] DistBug,[11] TangentBug,[12] Rev,[13] CautiousBug,[14] MRBUG,[15] ABUG,[16] etc. The fundamental characteristic of the Bug algorithm cluster is embodied in its motion manner, in which the whole collision-free path of the robot is combined by two basic motion modes, namely, motion toward the target and obstacle-boundary following if any.[17]

The Bug algorithm cluster is characterized by easy implementation and inherent convergence, at least in theory. However, in majority of actual cases, it is a theoretical algorithm rather than a practical algorithm because the required obstacle rounding capability is usually utopian with respect to real environments and real robots with multi-source noises. It is indicated that the challenging points of designing a practical Bug algorithm are to propose operable mechanisms of obstacle rounding and switching conditions of the two motion modes. This work also tries to make a contribution in this aspect, based on bionic inspirations from neuroethology and cognitive science.

Similar to the cognitive response activity of living beings, robot reactive navigation is also an iterative process of obtaining and understanding environment information (perception), generating corresponding motion strategies and realizing them under control (motion planning and control).[18,19] Based on the practices, in the last decade or so, the principles of neuroethology and cognition have

* Corresponding author. E-mail: tdwshu@staff.shu.edu.cn

been successfully applied in some of these aspects through structural and functional imitation. For example, in ref. [20], a real-time neuronal model and a distributed adaptive control framework are used in map-less autonomous navigation of a mobile robot to perform a foraging task. On its basis, a navigator is designed in ref. [21] by mimicking the neural mechanisms of insect navigation and enables the robot to learn the reliability of landmarks using an expectation reinforcement method. In experiments, this navigator generates analogous navigational behavior with real ants. In ref. [22], a quadruped robot is equipped with an artificial neural system with a central pattern generator (CPG) to accomplish adaptive dynamic walking with medium forward speed on outdoor irregular terrains.

In this work, we propose a robot cognitive response navigation algorithm—BioBug. It is featured as an overall biomimetic design by endowing bionic mechanisms to the complete navigation process rather than only a part of it as usual, e.g., in refs. [20–22]. In Section 2, a biological antenna model is established for environment perception. Based on it, four types of cognitive response walking behaviors (*Moving forward*, *Aimed turning*, *Avoidance turning* and *Arc rounding*) are defined in Section 3.2, and their triggering production rules are discussed in Section 3.3. Further, the overall navigation algorithm is integrated in Section 3.4, implemented by illustrations in Section 3.5, and examined by simulations in Section 4 and by experiments in Section 5. The designed algorithm is competent to overcome the noted defects of previous Bug algorithms in environment robustness and obstacle avoidance efficiency. It not only adopts *Arc rounding* explicitly as the obstacle rounding mechanism, but also extends the basic motion modes of the Bug algorithm cluster by two kinds of spot turning, namely, *Aimed turning* and *Avoidance turning*. The latter is beneficial to take into account the physical dimensions and view field limitations of the robot, which was generally ignored in previous Bug algorithms.

2. Robot Biological Antenna Model

The robot biological antenna model is established to understand sensory information for environment perception in this section. Here, "biological antennas" in different types denote the surrounding areas of the robot in specific shapes. If sensor readings are located in these zones, they are used to recognize the internal and external environment, mainly the distribution of local obstacles and the motion state of the robot itself.[23,24] The acquired information is further used in obstacle avoidance and autonomous navigation.

A laser range finder and motor encoders of two driving wheels are the major sensors relevant to the biological antenna model (See Section 5.1 for the complete list of sensors). The scanning data of the laser rangefinder are recorded as $(d_i,\phi_i)^{\mathrm{T}}$ in the polar coordinate system or $(x_i, y_i)^{\mathrm{T}}$in the Cartesian coordinate system, where i denotes their serial numbers. The minimum distance d_i is denoted by $d_{\min}$. The motor encoders record the equivalent movement distances, denoted by s_l and s_r, of the two driving wheels in the current walking behavior. They are further coupled in Eq. (1) to calculate the linear and angular displacements of the robot, denoted by s and θ, by ignoring the slip of the wheels. Then, they could be accumulated to calculate (joint with other sensors like GPS, etc.) the current position of the robot, denoted by X, if the starting position S is provided as *a priori* knowledge.

$$s = (s_l + s_r)\big/2 \text{ and } \theta = (s_r - s_l)\big/b, \tag{1}$$

where s_l, s_r, s, and θ are always initialized to zero while switching walking behaviors.

Figure 1 shows the established robot biological antenna model, in which the shape of the robot is simplified into its circumcircle with a radius R_r. Some system constants are given as: $R_{\max}$, the maximum detection range of the laser rangefinder; b, the wheel track; d_w, the diameter of the wheels; and c, the width of the wheels. Three types of biological antennas are designed, consisting of (1) two semi-annular antennas Dng_s and Dng_b in front (with radii of R_s and R_b respectively), (2) a wide-area antenna Lng in directly ahead (united by a rectangle in size of $2R_s$-by-a and a semi-circle with a radius of R_s), and (3) two rectangular antennas Wng_l and Wng_r overlapping the planar projections of the driving wheels (in size of d_w-by-c).

For their functions, Dng_s and Dng_b are used to detect the nearest obstacle point within the 180° field of view of the laser rangefinder, in which $R_s > R_r$ is set to avoid the risk of collisions and $R_s < R_b$ is set to improve robustness as a Schmitt trigger design (see Section 3.2 for details). Wng_l and Wng_r are used to record s_l and s_r for localization in real time. Lng is used to detect the nearest

https://doi.org/10.1017/S0263574713000970 Published online by Cambridge University Press

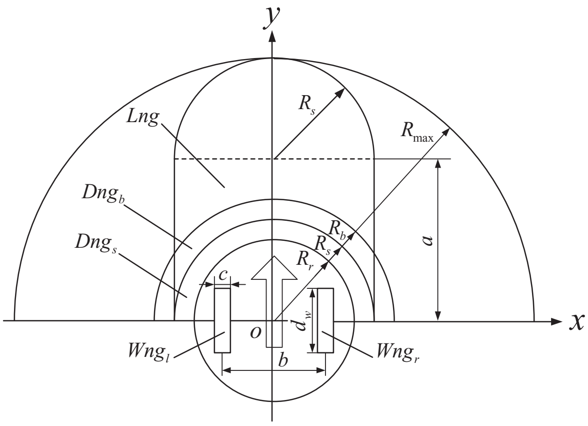


Fig. 1. The established robot biological antenna model.

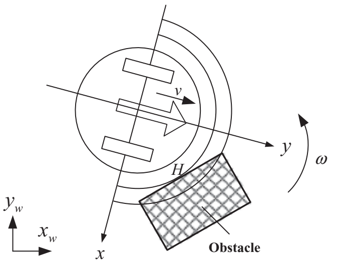


Fig. 2. Robot cognitive response to biological antenna information for obstacle avoidance.

obstacle point within the field of view of the laser rangefinder in directly ahead. It is helpful, from plane geometry, to determine the maximum safe forward movement distance $s_{\max}$, given by

$$s_{\max} = \min\left\{ d_i \sin(\phi_i) - \sqrt{R_s^2 - d_i^2 \cos^2(\phi_i)} \right\} \quad (i = 1, \cdots, n), \tag{2}$$

where n denotes the number of obstacle points detected by the laser rangefinder that are located in the range of Lng; i denotes the serial numbers of these points.

This model is feasible to process a large amount of navigation information, including, S, the starting position of the robot; T, its target position (provided as *a priori* knowledge in this work); X, its current position; $(x_r, y_r, \theta_r)^{\mathrm{T}}$, its current pose; d_{XT}, the distance between X and T; θ_{XT}, the direction angle of the vector $\overrightarrow{XT}$; $d_{\min}$, the minimum obstacle distance within the field of view of the laser rangefinder; s and θ, the linear and angular displacements of the robot in the current walking behavior; and $s_{\max}$, its maximum safe forward movement distance. After sampling at every time, each biological antenna updates its information of the corresponding area, as well as the above navigation information.

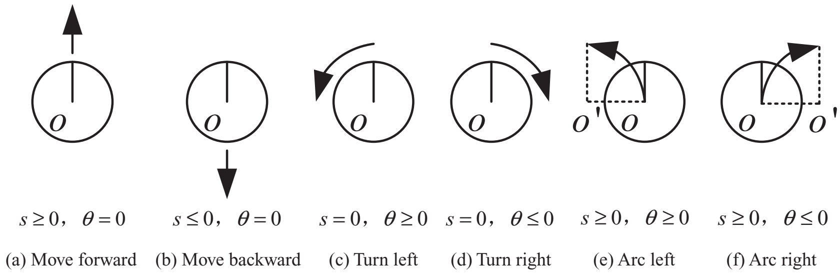
Based on the above design, the established model is characterized by two advantages for environmental perception. One is its rapid processing speed of multi-mode sensor data because only the specific sensor information within the interested antenna areas is dealt with, and the other is its real-time memory and update functions of the antenna information and other navigation information. Therefore, depending on the biological antennas, the robot can obtain some useful environment information for generating the corresponding walking behaviors in navigation.

3. Cognitive Response Navigation Algorithm Based on the Biological Antenna Model

3.1. Philosophy of the cognitive response navigation algorithm

The established cognitive response navigation algorithm belongs to the family of reactive motion planning algorithms. It responds to the stimulation from the biological antennas in an IF-THEN manner. For example, see Fig. 2, if the robot detects an obstacle in front with its biological antennas,

https://doi.org/10.1017/S0263574713000970 Published online by Cambridge University Press



(a) Move forward (b) Move backward (c) Turn left (d) Turn right (e) Arc left (f) Arc right

Fig. 3. Robot basic walking behaviors.

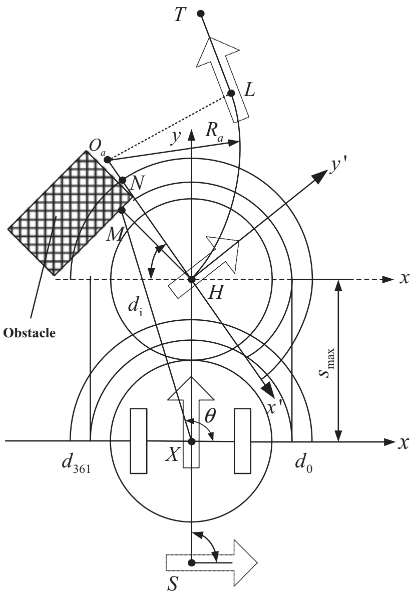


Fig. 4. Four types of cognitive response walking behaviors in navigation.

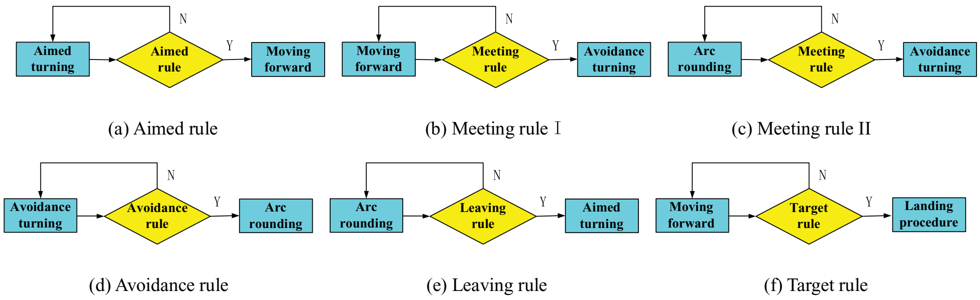
the walking behavior of *Avoidance turning* is triggered immediately for avoidance. Therefore, the navigation process of the robot is actually a process of triggering production rules of different walking behaviors in a series of IF-THEN manners (See Section 3.3 for details). In the last example of Fig. 2, through the boundary-following motion, the robot is able to know the general location and size of surrounding obstacles with the help of biological antennas. Therefore, in the whole process, relevant system parameters and antenna memories can be revised and updated through learning.

3.2. Robot cognitive response walking behaviors

There are six basic walking behaviors of a mobile robot in general applications, as moving forward and backward, spot turning to the left and right, and arc rounding to the left and right, as shown in Fig. 3. Among them, spot turnings can be further distinguished as aimed turnings (to the target) and avoidance turnings (against to obstacles). Besides, moving backward is forbidden due to the blindness of the laser rangefinder. Instead, the arcing mechanism is used for obstacle avoidance. As a result, four types of cognitive response walking behaviors are adopted in this work, as *Moving forward*, *Aimed turning*, *Avoidance turning*, and *Arc rounding*, as shown in Fig. 4. Their uses are explained as below after some notations are defined: meeting points *H*, the locations where the robot meets obstacles; and leaving points *L*, the locations where the robot leaves obstacles after a boundary-following motion.

(1) *Aimed turning*, which is a spot turning motion to aim at the target *T* at the starting point *S* or leaving points *L*. The direction of *Aimed turning* is determined by the minor turning angle principle. As shown in Fig. 4, if the robot has been located at the starting point *S* or the leaving point *L, Aimed turning* is triggered to aim at the target.

https://doi.org/10.1017/S0263574713000970 Published online by Cambridge University Press



(a) Aimed rule (b) Meeting rule Ⅰ (c) Meeting rule II

(d) Avoidance rule (e) Leaving rule (f) Target rule

Fig. 5. (Colour online) The combination of the four types of cognitive response walking behaviors in a series of IF-THEN manners.

(2) *Moving forward*, which is a straight-line motion along $\vec{XT}$. Its starting point could be S or L, and its end point could be H or T (T means the completion of the navigation task). As shown in Fig. 4, if the robot has aimed at the target at the starting point S or leaving points L, *Moving forward* is triggered to approach the target.

(3) *Avoidance turning*, which is a spot turning motion at H as a preparation of following obstacle boundaries in the subsequent *Arc rounding* motion. *Avoidance turning* is triggered by Dng_s detecting H, and then the robot turns oppositely to the obstacle direction until Dng_b is absent of obstacles. As for the design of Dng_s and Dng_b, $R_s < R_b$ is required in order to improve system robustness as a Schmitt trigger design. On the purpose of improving algorithm accuracy and robustness, we do not simply use $\vec{XH}$ to represent the obstacle direction but fuse the information in the area of Dng_b. Suppose m obstacle points are scanned in all by the laser rangefinder in Dng_b, compute the arithmetic mean of their azimuths (with respect to the x-axis direction) in the robot coordinate system, denoted by $\overline{\phi}$:

$$\overline{\phi} = \frac{1}{m}\sum_{i=1}^{m}\phi_i \tag{3}$$

Then define the following integer variable,

$$f_{\text{obt}} = \begin{cases} 0, & if\,\overline{\phi} \le 90^\circ; \\ 1, & if\,90^\circ < \overline{\phi} \le 180^\circ. \end{cases} \tag{4}$$

Where the integer variable $f_{\text{obt}} = 0$ denotes the obstacles being in the right, and $f_{\text{obt}} = 1$ denotes left. The integer variable is initialized to $f_{\text{obt}} = -1$ when the robot is located at the starting point S or the leaving points L. As shown in Fig. 4, if the robot detects an obstacle being in the left by Dng_b at H, *turning right* is triggered for a preparation of avoidance.

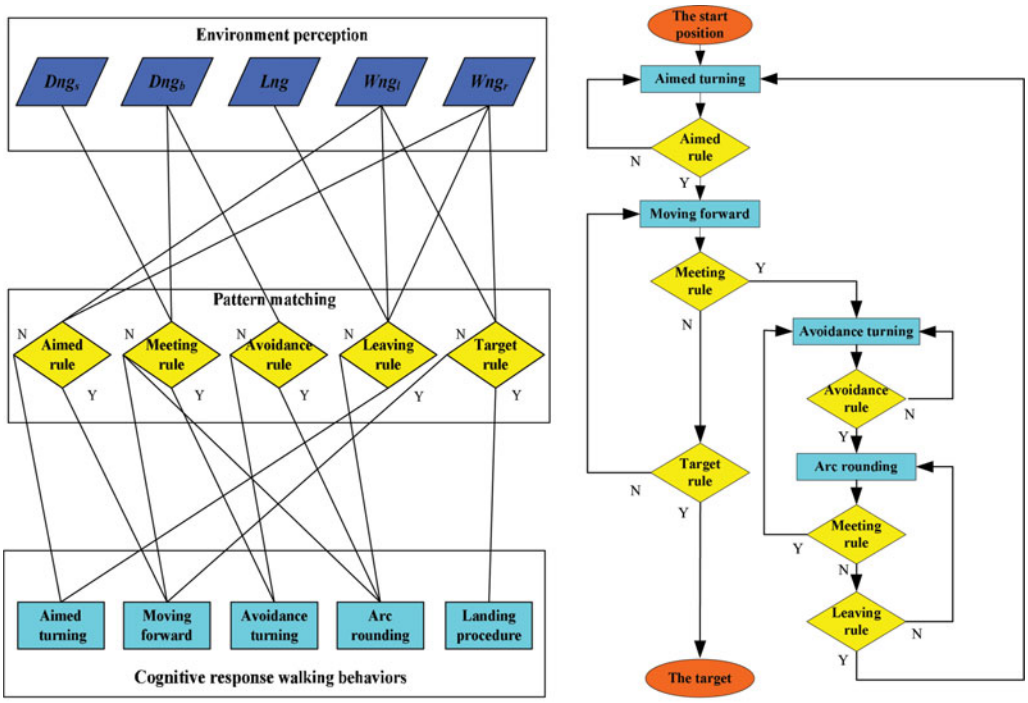
(4) *Arc rounding*, which is an arc rounding motion with a certain radius to follow obstacle boundaries. It is triggered after an *Avoidance turning* motion (but in the opposite direction) until the escaping conditions are met or another meeting point H is discovered. As shown in Fig. 4, since the obstacle is in the left, *turning right* and *arcing left* are triggered alternately until the robot escapes from the obstacle.

3.3. Switching of robot cognitive response walking behaviors

The key role of the cognitive response navigation algorithm is to determine switching rules of the four types of walking behaviors, as shown in Fig. 5. They lay a foundation of realizing complex navigation motions through a reasonable combination of the four types of walking behaviors in a series of IF-THEN manners, as illustrated in Fig. 5 from (a) to (f), and explained as below.

(1) *Aimed rule*, which is used to judge, according to Wng_l and Wng_r, whether the heading of the robot is in the direction of $\vec{XT}$ during *Aimed turning*. It is expressed quantitatively as $\theta_r = \theta_{XT}$, where θ_r denotes the heading angle of the robot in the global coordinate system and θ_{XT} denotes the azimuth of the target with respect to the robot. If true, the current walking behavior should be switched to *Moving forward*, and otherwise the robot maintains the current walking behavior, as

https://doi.org/10.1017/S0263574713000970 Published online by Cambridge University Press



(a) Philosophy of cognitive response navigation

(b) The algorithm flowchart

Fig. 6. (Colour online) The principle of the cognitive response navigation algorithm.

shown in Fig. 5(a). Taking Fig. 4 as an example, if the robot has arrived at the starting position point S or leaving points L, *Aimed turning* is triggered until *Aimed rule* is satisfied.

(2) *Meeting rules* I & II, which is used to judge, according to Dng_s and Dng_b, whether the robot has arrived at H during *Moving forward* or *Arc rounding*. It is expressed quantitatively as $d_{\min} \leq R_s$. If true, the current walking behavior should be switched to *Avoidance turning*, and otherwise the robot maintains the current walking behavior, as shown in Figs. 5(b) and (c). Taking Fig. 4 as an example, in the process of *Moving forward*, if and only if *Meeting rule is true*, *Avoidance turning* is triggered for the preparation of avoidance.

(3) *Avoidance rule*, which is used to judge, according to Dng_b, whether the obstacle has become invisible for the robot during *Avoidance turning*. It is expressed quantitatively as $d_{\min} \geq R_b$, where $R_b > R_s$. If true, the current walking behavior should be switched to *Arc rounding*, and otherwise the robot maintains the current walking behavior, as shown in Fig. 5(d). However, if false in a complete circle of *Avoidance turning*, it means the robot has trapped among obstacles without a hope of escape. Thus, the robot stops and the navigation task fails. Taking Fig. 4 as an example, in the process of *Avoidance turning*, if and only if *Avoidance rule is true*, *Arc rounding* is triggered to follow the obstacle's boundary.

(4) *Leaving rule*, which is used to judge, according to Lng, Wng_l and Wng_r, whether the robot has arrived at L during *Arc rounding* in two sufficient conditions. One is for the final landing in the quantitative expression of ($|\theta_r - \theta_{XT}| \leq \theta_\varepsilon$ & $d_{XT} \leq s_{\max}$), the other is for the safety of obstacle avoidance in the quantitative expression of ($|\theta_r - \theta_{XT}| \leq \theta_\varepsilon$ & $s_{\max} \geq s_{\text{step}}$, where d_{XT} denotes the distance between the robot and the target, θ_ε denotes a predefined tolerance of aimed angle, and s_{step} denotes a predefined threshold of maximum safe forward movement distance. If any one of them is true, the current walking behavior should be switched to *Aimed turning*, and otherwise the robot maintains the current walking behavior, as shown in Fig. 5(e). However, if both of them are false in the whole circle of *Arc rounding*, it means the robot has trapped among obstacles without a hope of escape. Thus, the robot stops and the navigation task fails. Taking Fig. 4 as an example, in the process of *Arc rounding*, if and only if *Leaving rule is true*, *Aimed turning* is triggered to aim at the target.

(5) *Target rule*, which is to judge, according to Wng_l and Wng_r, whether the robot has approached the target T during *Moving forward*. It is expressed quantitatively as $d_{XT} \leq d_\varepsilon$, where d_ε denotes a predefined tolerance of target distance. If true, the robot fires the landing procedure (see in Fig. 6),

https://doi.org/10.1017/S0263574713000970 Published online by Cambridge University Press

and otherwise it maintains the current walking behavior, as shown in Fig. 5(f). Taking Fig. 4 as an example, in the process of *Moving forward*, if and only if *Meeting rule* is always false and finally *Target rule is true*, the landing procedure is fired to land the target.

3.4. The principle of the cognitive response navigation algorithm

This subsection integrates the production rules of walking behavior switching and forms the complete cognitive response navigation algorithm—BioBug. Its principle is depicted in Fig. 6, in which (a) shows the philosophy of cognitive response navigation in a series of IF-THEN manners (see in Fig. 5), and (b) shows the algorithm flowchart. The biological antenna model delineates the interested sensing areas for environment perception, and then this obtained environment stimulation is responded to generate corresponding cognitive response walking behaviors according to their switching rules. The complete steps of BioBug are given with pseudocodes in Algorithm 1.

Evidently, in the whole process of navigation, the four types of walking behaviors keep retentive and coherent. If and only if a certain switching rule is satisfied, the corresponding cognitive response walking behavior is triggered immediately; otherwise, the robot maintains the current walking behavior.

```
Algorithm 1 BioBug: The cognitive response navigation algorithm
Parameters: $f_{obt} \in \{-1, 0, 1\}$
        Step 0:  Initializations
        Step 1:  Motion toward to the target
        Step 2:  Obstacle-boundary following

0) Set $X = S$, $T$ and $f_{\mathrm{obt}} = -1$, and initialize main algorithm parameters
1) if ($X = S$ OR $X = L$)
      Aimed turning is triggered
      if (Aimed rule is true)
         Moving forward is triggered
         if (Meeting rule is true)
            goto Step 2
         else if (Meeting rule is FALSE AND Target rule is true)
            Landing procedure is fired until $T$ is reached, and the algorithm stops
         end
      end
   end
2) if (Meeting rule is true)
      Avoidance turning is triggered
      if (Avoidance rule is true)
         Arc rounding is triggered
         if (Meeting rule is true)
            goto Step 2
         else if (Leaving rule is true)
                 goto Step 1
         end
      end
   end
```

3.5. Implementation of the cognitive response navigation algorithm

The implementation of applications of the cognitive response navigation algorithm is illustrated in two scenarios, in which the obstacle has a linear boundary and an arbitrary shaped boundary respectively. Figure 7 records the navigation track of the robot in the first scenario, $SH_1H_2H_3LT$. The robot starts moving from S and detects an obstacle at the first meeting point H_1. It thus stops to execute *Avoidance turning* until *Avoidance rule* is satisfied. As a result, *Arc rounding* is triggered until the robot arrives at a second meeting point H_2. It stops and switches to *Avoidance turning* again. Then, as *Avoidance turning* and *Arc rounding* are executed alternately, the robot passes another meeting point H_3 and

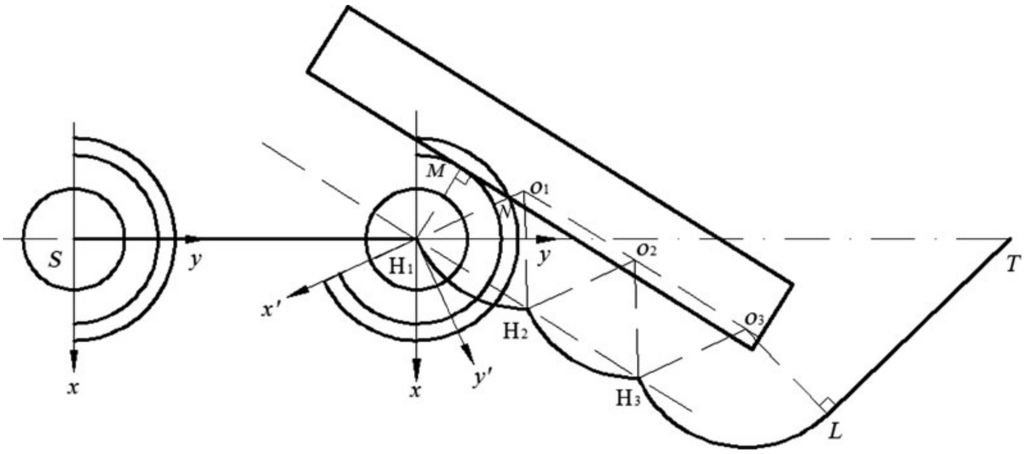


Fig. 7. Robot navigation track in linear obstacle-boundary following.

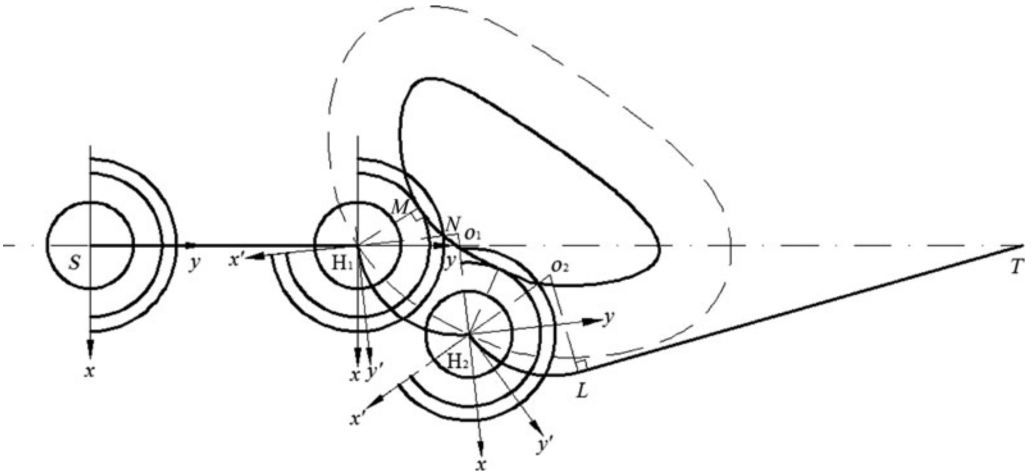


Fig. 8. Robot navigation track in arbitrary shaped obstacle-boundary following.

arrives at the leaving point L. Since no more obstacles are detected, the robot finally lands at T after an *Aimed turning* behavior and a *Moving forward* behavior.

The *Arc rounding* motions are given particular concerns. For their lengths, the linear obstacle boundary causes all arcs except the last one are in the same length, given by $|\widehat{H_1H_2}| = |\widehat{H_2H_3}|$. For their number, it is negatively correlated to the arcing radius R_a. The influence of R_a can be described briefly as, a smaller value will cause *Arc rounding* motions restarted frequently; instead, a larger value is easy to make the robot ignore the gaps between obstacles and thus, trapped in blind areas. Therefore, it should be determined by trade-off according to the size of the robot and the density of obstacles, but at least guaranteeing $R_a > R_b$ to keep algorithm stability. Further, it will be online modified according to the real environment through learning in practice.

Figure 8 records the navigation track of the robot in the second scenario, SH_1H_2LT. It is also made up by walking behaviors of *Aimed turning*, *Moving forward*, *Avoidance turning,* and *Arc rounding*. This example reveals *Arc rounding* is a feasible way for obstacle boundary following with robustness to their shapes (and arrangements, see Section 5.2). The practicality of the algorithm is presented once again.

4. Simulations of the cognitive response navigation algorithm

One of the most challenging fields in motion planning is to find shorter paths in less time. Although the first objective of the motion planner is to find to a free path to the target, that is, avoiding obstacle collision, the second objective will be to decrease the length of the path and the time that

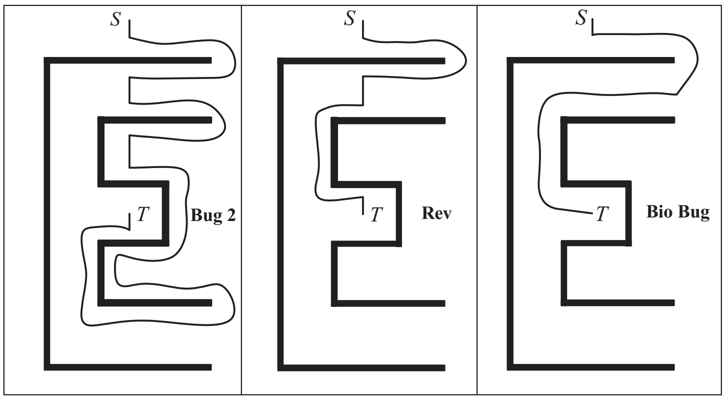


Fig. 9. Paths for BioBug and two types of classic Bug algorithms by simulations.

Table I. Values of main algorithm parameters of BioBug in simulation.

Parameter name	Notation	Value
Radius of robot circumcircle	R_r	200 mm
Radius of antenna *Dng*$_s$	R_s	400 mm
Radius of antenna *Dng*$_b$	R_b	420 mm
Radius of *Arc rounding* motions	R_a	450 mm
Leaving threshold	s_{step}	200 mm
Target distance tolerance	d_ε	100 mm
Aimed angle tolerance	θ_ε	2°
Speed of *Moving forward* motions	$v_{r,f}$	200 mm/s
Speed of *Arc rounding* motions	$v_{r,a}$	100 mm/s
Angular speed of spot turning	$\omega_{r,s}$	0.45 rad/s

the planner finds the path. The performance of the proposed algorithm BioBug was compared with the two types of classic Bug algorithms, Bug2 and Rev, by simulations, as shown in Fig. 9. The simulation environment is constructed in MATLAB 7.1 in Windows 7 using non-optimized code. The hardware consists of an Intel(R) Core(TM) i3 CPU with 2.53 GHz. The main parameters of BioBug in simulation are shown in Table I.

Figure 10 presents path lengths for the above three Bug algorithms intuitively. The path length of Rev is much smaller than that of Bug2 because the shortest viewpoint is being searched in Rev in real time. Compared with the competitors, BioBug, which adopts *Arc rounding* as an obstacle rounding mechanism, harvested not only a smaller path length, but also shorter time for obstacle escape.

5. Robot cognitive response navigation experiments

5.1. Experiment system

The experiment scene is shown in Fig. 11, consisting of a real mobile robot system, a piece of corridor sized 6 m by 4 m and several bins in regular shapes (as obstacles). The real robot system equips an airborne control computer, a pair of differential driving wheels and a variety of sensors, including a SICK LMS200 laser rangefinder, two driving motor encoders, a sonar ring, a digital compass, a GPS, etc. The field of view of the laser rangefinder is 0 $\sim$ 8m in distance and 0 $\sim$ 180° in angle with resolutions of 10 mm and 0.5° respectively. The motor encoders, digital compass and GPS are used for robot self-positioning.

Main algorithm parameters are assigned values in Table II in the following considerations: the smaller s_{step} enables the robot to escape obstacles easily; the moderate speeds, $v_{r,f}$, $v_{r,a}$ and $\omega_{r,s}$, can offset the negative impacts of the limited communication speed and thus, benefit the real-time performance of the algorithm; as well, the radii, R_s, R_a and R_b, are expected to make a balance among the properties of navigation rapidity, obstacle sensitivity, and motion safety.

https://doi.org/10.1017/S0263574713000970 Published online by Cambridge University Press

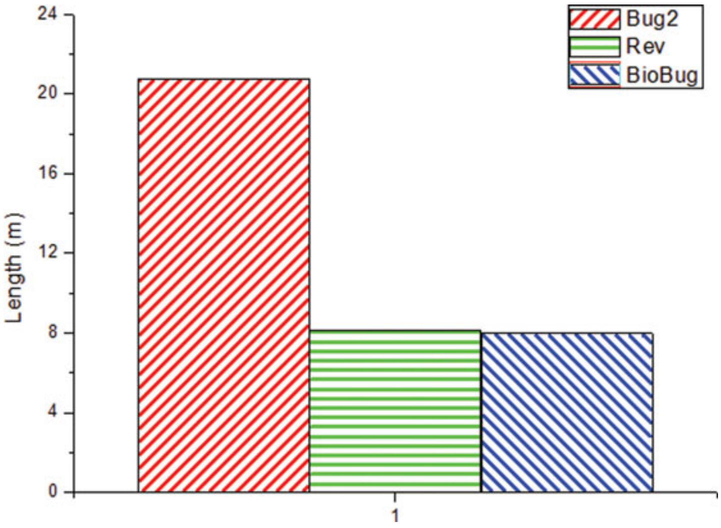


Fig. 10. (Colour online) Path lengths for BioBug and two types of classic Bug algorithms.

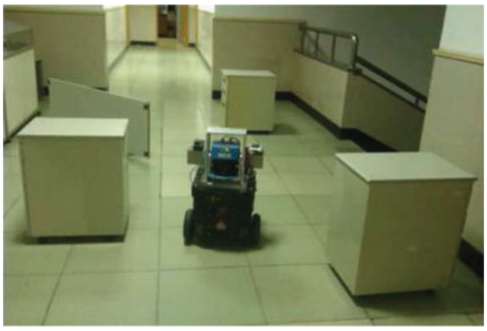


Fig. 11. (Colour online) The experiment scene.

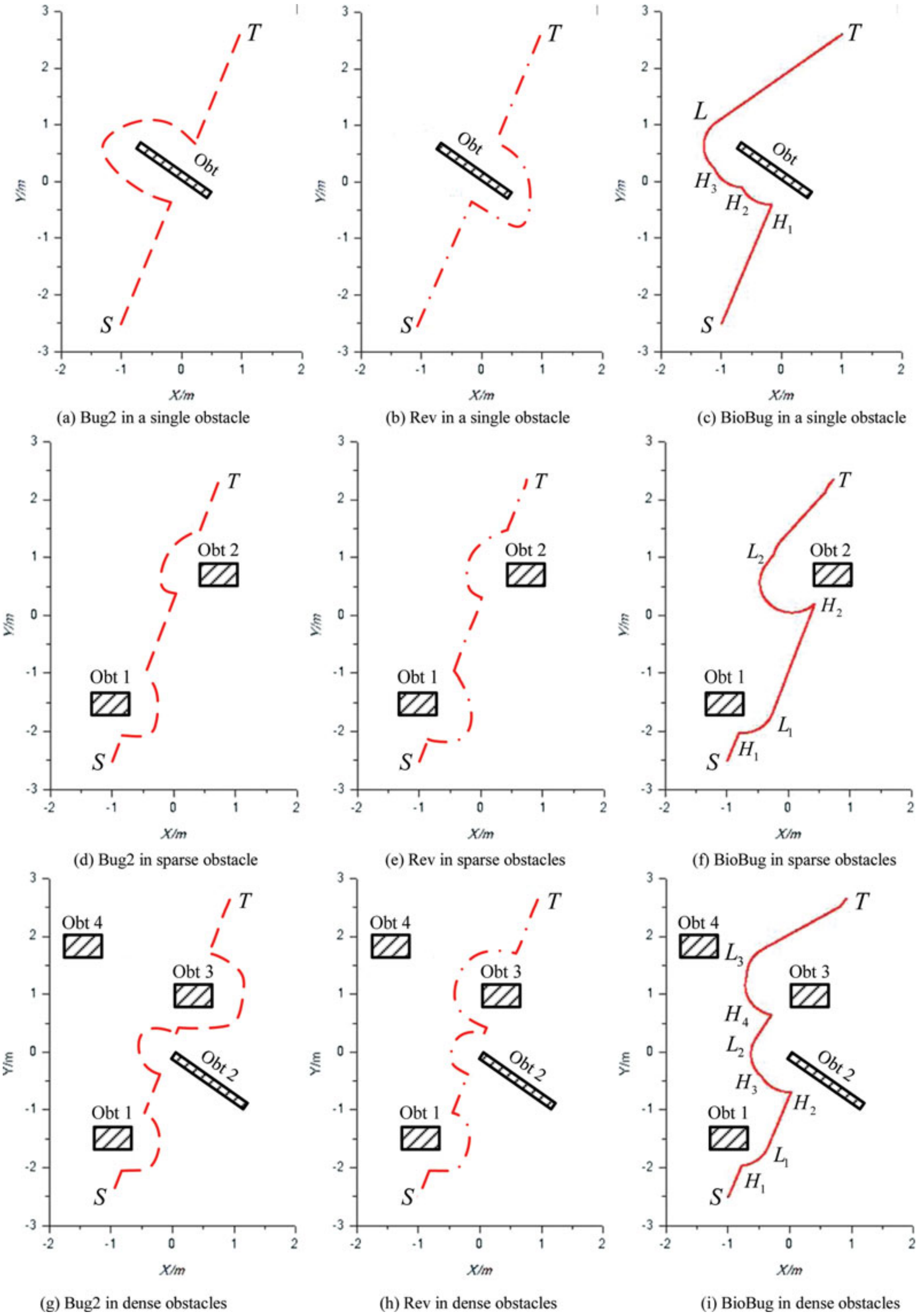
Table II. Values of main algorithm parameters.

Parameter name	Notation	Value
Radius of robot circumcircle	R_r	250 mm
Radius of antenna Dng_s	R_s	480 mm
Radius of antenna Dng_b	R_b	500 mm
Radius of *Arc rounding* motions	R_a	520 mm
Leaving threshold	s_{step}	200 mm
Target distance tolerance	d_ε	180 mm
Aimed angle tolerance	θ_ε	2.5°
Speed of *Moving forward* motions	$v_{r,f}$	100 mm/s
Speed of *Arc rounding* motions	$v_{r,a}$	50 mm/s
Angular speed of spot turning	$\omega_{r,s}$	0.243 rad/s

5.2. *Experiments and discussions*

Three kinds of navigation experiments are conducted corresponding to different densities of obstacles through BioBug and the two types of classic Bug algorithms, Bug2 and Rev. The navigation algorithm codes are written in Visual C++. In each experiment, the position and pose of the robot are recorded in real-time with the navigation process, and finally the path is figured with Origin 8.0 for post processing and result analyzing. The paths of the two types of classic Bug algorithms and BioBug are shown in Fig. 12, in which (a) & (b) & (c) for a single obstacle, (d) & (e) & (f) for sparse obstacles and (g) & (h) & (i) for dense obstacles, and the path lengths for those three Bug algorithms corresponding to different densities of obstacles are show in Fig. 13. For the last experiment adopting

https://doi.org/10.1017/S0263574713000970 Published online by Cambridge University Press



(a) Bug2 in a single obstacle (b) Rev in a single obstacle (c) BioBug in a single obstacle

(d) Bug2 in sparse obstacle (e) Rev in sparse obstacles (f) BioBug in sparse obstacles

(g) Bug2 in dense obstacles (h) Rev in dense obstacles (i) BioBug in dense obstacles

Fig. 12. (Colour online) Paths for Bug algorithms corresponding to different densities of obstacles.

BioBug algorithm, its living information of the navigation process is also supplemented by video screenshots, as shown in Fig. 14.

For the comparison of BioBug algorithm itself, different densities of obstacles are decorated as shown in Figs. 12 (c) & (f) & (i). In the environment of a single obstacle, as shown in Fig. 12(c), the navigation path is similar to the simulation result in Fig. 7. Their minor differences are sourced from system noises in experiment. For example, the lag of updating sensor readings causes a (limited)

https://doi.org/10.1017/S0263574713000970 Published online by Cambridge University Press

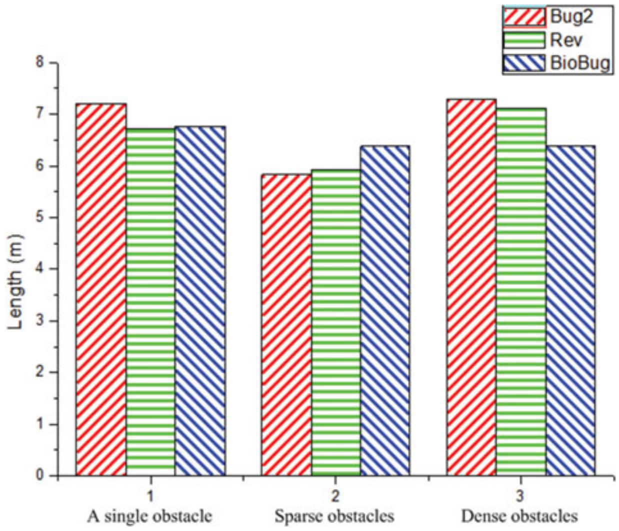


Fig. 13. (Colour online) Path lengths for Bug algorithms corresponding to different densities of obstacles.

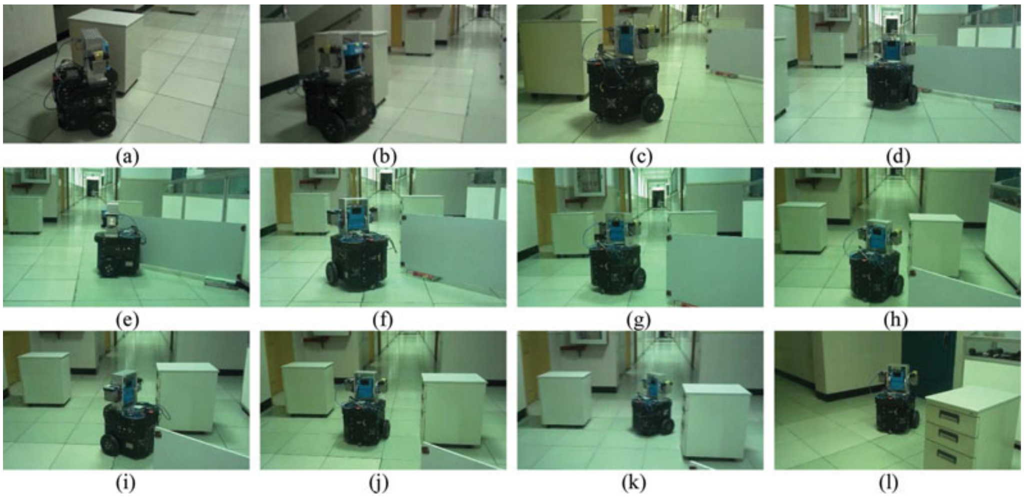


Fig. 14. (Colour online) Video screenshots of the robot navigation process with dense obstacles.

displacement of the leaving point L. This error is compensated by executing *Aimed turning* before *Moving forward*. Further, the robot also completes the navigation tasks with obstacles in increased numbers. Comparing Figs. 12(f) and (i), the robot passes through the gap between Obstacle 1 and Obstacle 2 in the early stage of both experiments. Differently, in (f), after the robot escapes the influence of Obstacle 2, it aims at and moves toward to the target without other obstructions; instead, in (i), it encounters Obstacle 3 and Obstacle 4. Since the gap between Obstacle 3 and Obstacle 4 is wider than that between Obstacle 2 and Obstacle 3, the robot selects the larger one but ignores the smaller one. As noted before, the sensitivity of the robot to narrow gaps is determined by the arcing radius R_a.

For qualitatively evaluating the segmented arcings-based obstacle-boundary following mechanism of BioBug, the paths are smooth and close to obstacle boundaries in all experiments in Fig. 12. This reflects its efficiency for obstacle avoidance and robustness to application environments.

For quantitative evaluation of BioBug, see Table III, turning number and route length are employed as algorithm performance indices. They both increase with the density of obstacles. For the turning numbers, in detail, they are counted at the positions S, H_1, H_2, H_3, and L in Fig. 12(c), at S, H_1, L_1, H_2 and L_2 in Fig. 12(f), and at S, H_1, L_1, H_2, H_3, L_2, H_4, and L_3 in Fig. 12(i). In addition, the numbers of robot cognitive response walking behaviors with respect to different densities of obstacles are counted in Table IV. It must be helpful to understand the navigation process in Fig. 14.

https://doi.org/10.1017/S0263574713000970 Published online by Cambridge University Press

Table III. Algorithm performance indices corresponding to different densities of obstacles.

Performance index	A single obstacle	Sparse obstacles	Dense obstacles
Turning number	5	5	8
Arc rounding length (m)	1.905	2.309	2.585

Table IV. Statistics of robot cognitive response walking behaviors with respect to different densities of obstacles.

	Behaviors			
Types	Aimed turning	Moving forward	Avoidance turing	Arc rounding
A single obstacle	2	2	3	3
Sparse obstacles	3	3	2	2
Dense obstacles	4	4	4	4

In the comparison study with other Bug algorithms, the two types of classic Bug algorithms, Bug2 and Rev, were selected as the representatives for result discussion and performance analysis. In the environment of a single obstacle in Fig. 12 from (a) to (c), the most prominent advantage of BioBug is a shorter time of escaping from the obstacle. As shown in Fig. 12(b), in Rev, the robot selects from the right to avoid the obstacle because the shortest viewpoint is searched in real-time. Therefore, as shown in Fig. 13, the path length for Rev in a single obstacle is a little smaller than that of Bug2 and BioBug. In the environment of sparse obstacles in Figs. 12 from (d) to (f), the robot is able to find a free path to the target through any of the above mentioned algorithms. Due to the specificity of *Arc rounding*, the path length of BioBug in sparse obstacles is a little larger than that of Bug2 and Rev, as shown in Fig. 13. However, in the environment of dense obstacles in Figs. 12 from (h) to (i), owing to shorter time for obstacle escape, the path length of BioBug is much smaller than that of Bug2 and Rev, as shown in Fig. 13. Therefore, through BioBug, the robot can find a free path in less time. The environment robustness and obstacle avoidance efficiency of the algorithm are verified by experiments once again.

6. Conclusions

(1) In this work, a cognitive response navigation algorithm for mobile robots is proposed. It endows bionic mechanisms from neuroethology and cognitive science to the complete navigation process including environment perception as well as motion planning and control.

(2) For environment perception, a biological antennas model is designed to handle sensor readings only in the interested areas of the local environment. As it discards a large amount of irrelevant information, it takes an advantage in processing speed on the basis of feasibility in navigation.

(3) For motion planning and control, an improved Bug algorithm is developed to respond to environment stimulation and generate corresponding walking behaviors. It enhances practicability of the Bug algorithm cluster in two aspects: one is proposing a definite obstacle rounding mechanism, i.e., *Arc rounding*; and the other is extending the basic motion modes by two kinds of spot turning, i.e., *Aimed turning* and *Avoidance turning*, to fit for the practical constraints in robot physical dimensions, field of view, and system noises.

(4) The above design ideas are supported by simulations and experiments conducted in different conditions of obstacle density and boundary shape. An ongoing research is evolving this reactive bionic navigation algorithm to a hybrid deliberative/reactive algorithm to process environment identification and motion planning and control in different levels.

Acknowledgements

This research was supported by the Natural Science Foundation of China (51075252, 61203351, 61101177) and Innovative Foundation of Shanghai University (A.10-0109-11-003).

https://doi.org/10.1017/S0263574713000970 Published online by Cambridge University Press

References

1. S. S. Ge and Y. J. Cui, "Dynamic motion planning for mobile robots using potential field method," *Auton. Robots* **13**, 207–222 (2002).
2. J. Borenstein and Y. Koren, "The vector field histogram: Fast obstacle avoidance for mobile robots," *IEEE Trans. Robot. Autom.* **7**(3), 278–288 (1991).
3. J. Ghommam, H. Mehrjerdi and M. Saad, "Leader-Follower Formation Control of Nonholonomic Robots with Fuzzy Logic Based Approach for Obstacle Avoidance," *Proceedings of the 2011 IEEE International Conference on Intelligent Robots and Systems*, San Francisco, CA, USA (Sep. 25–30, 2011), pp. 2340–2345.
4. Z. Eduardo, G. Jaime and M. Paul and J. Peran, "Adaptive behavior navigation of a mobile robot," *IEEE Trans. Sys.* **32**(1), 160–169 (2002).
5. C. Luo and S. Yang, "A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments," *IEEE Trans. Neural Netw.* **19**(7), 1279–1298 (2008).
6. T. W. Manikas, K. Ashenayi and R. L. Wainwright, "Genetic algorithms for autonomous robot navigation," *IEEE Instrum. Meas. Mag.* **10**(6), 26–31 (2007).
7. V. J. Lumelsky and A. A. Stepanov, "Path-planning strategies for a point mobile automaton moving amidst obstacles of arbitrary shape," *Algorithmica* **2**, 403–430 (1987).
8. V. J. Lumelsky, "Algorithmic and complexity issues of robot motion in an uncertain environment," *J. Complexity* **3**(2), 146–182 (1987).
9. V. J. Lumelsky and A. A. Stepanov, "Dynamic path planning for a mobile automaton with limited information on the environment," *IEEE Trans. Automat. Contr.* **31**, 1058–1063 (1986).
10. V. J. Lumelsky and T. Skewis, "Incorporating range sensing in the robot navigation function," *IEEE Trans.Syst. Man. Cybern.* **20**(5), 1058–1069 (1990).
11. I. Kamon and E. Rivlin, "Sensory-based motion planning with global proofs," *IEEE Trans. Robot. Autom.* **13**(6), 1058–1068 (1997).
12. I. Kamon, E. Rimon and E. Rivlin, "TangentBug: A range-sensor-based navigation algorithm," *Int. J. Robot. Res.* **17**(9), 934–953 (1998).
13. H. Noborio, K. Fujimura and Y. Horiuchi, "A comparative study of sensor-based path-planning algorithms in an unknown maze," *Proc. IEEE/RSI Int. Conf. Intell. Robots Syst.* **2**, 909–916 (2000).
14. E. Magid and E. Rivlin, "Cautiousbug: A Competitive Algorithm for Sensory-Based Robot Navigation," *Proceedings of the 2004 IEEE International Conference on Intelligent Robots and Systems*, Japan (Sep. 28–Oct. 2, 2004), pp. 2757–2762.
15. S. Sarid, A. Shapiro and Y. Gabriely, "MRBUG: A Competitive Multi-Robot Path Finding Algorithm," *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, Roma, Italy (Apr. 10–14, 2007) pp. 877–882.
16. A. Javier, O. Alberto and M. Javier, "ABUG: A Fast Bug-derivative Anytime Path Planner with Provable Suboptimality Bounds," *Proceedings of the 2009 International Conference on Advanced Robotics*, Palma, Spain (Jun. 22–26, 2009) pp. 1–8.
17. H. Choset and K. M. Lynch, *Principles of Robot Motion* (The MIT Press, Cambridge, MA, 2005).
18. D. Lambrinos, R. Möller, T. Labhart, R. Pfeifer and R. Wehner, "A mobile robot employing insect strategies for navigation," *Robot. Auton. Syst.* **30**(1), 39–64 (2000).
19. M. O. Franz and H. A. Mallot, "Biomimetic robot navigation," *Robot. Auton. Syst.* **30**(1), 133–153 (2000).
20. M. Zenon, L. Miguel, and J. M. Blanco Calvo, A. Dhir, A. Duff, S. Bermúdez i Badia and Paul F. M. J. Verschure, "Insect-Like Mapless Navigation Based on Head Direction Cells and Contextual Learning Using Chemo-Visual Sensors," *Proceedings of the 2009 IEEE International Conference on Intelligent Robots and Systems (RSJ)*, St. Louis, MO, USA (Oct. 10–15, 2009) pp. 2243–2250.
21. M. Zenon, Paul F. M. J. Verschure, and B. B. Sergi, "An Insect-Based Method for Learning Landmark Reliability Using Expectation Reinforcement in Dynamic Environments," *Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage*, Anchorage, AK, USA (May 3–7, 2010) pp. 3805–3812.
22. H. Kimura, Y. Fukuoka and A. H. Cohen, "Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts," *Int. J. Robot. Res.* **26**, 1–25 (2007).
23. Y. Kuwana, I. Shimoyama and H. Miura, "Steering control of a mobile robot using insect antennae," *Proceedings of the 1995 IEEE International Conference on Intelligent Robots and Systems (RSJ)*, Tokyo, Japan (Aug. 5–9, 1995), pp. 530–535.
24. N. J. Cowan, E. J. Ma and M. Cutkosky and R. J. Full, "A biologically inspired passive antenna for steering control of a running robot," *Robot. Res. Springer Tracts in Adv. Robot.* **15**, 541–550 (2005).