The Knowledge Engineering Review, Vol. 32, e5, 1–17. © Cambridge University Press, 2017
doi:10.1017/S0269888916000151

A migrant-inspired path planning algorithm for obstacle run using particle swarm optimization, potential field navigation, and fuzzy logic controller

PING-HUAN KUO, TZUU-HSENG S. LI, GUAN-YU CHEN, YA-FANG HO and CHIH-JUI LIN

aiRobots Lab, Department of Electrical and Electronic Engineering, National Cheng Kung University, Tainan 70101, Taiwan, ROC; e-mail: coll22000@hotmail.com, thsli@mail.ncku.edu.tw, hank2579@hotmail.com, makinosakuya@hotmail.com, lirere0225@hotmail.com

Abstract

Obstacle avoidance is an important issue in robotics. In this paper, the particle swarm optimization (PSO) algorithm, which is inspired by the collective behaviors of birds, has been designed for solving the obstacle avoidance problem. Some animals that travel to the different places at a specific time of the year are called migrants. The migrants also represent the particles of PSO for defining the walking paths in this work. Migrants consider not only the collective behaviors, but also geomagnetic fields during their migration in nature. Therefore, in order to improve the performance and the convergence speed of the PSO algorithm, concepts from the migrant navigation method have been adopted for use in the proposed hybrid particle swarm optimization (H-PSO) algorithm. Moreover, the potential field navigation method and the designed fuzzy logic controller have been combined in H-PSO, which provided a good performance in the simulation and the experimental results. Finally, the Federation of International Robot-soccer Association (FIRA) HuroCup Obstacle Run Event has been chosen for validating the feasibility and the practicability of the proposed method in real time. The designed adult-sized humanoid robot also performed well in the 2015 FIRA HuroCup Obstacle Run Event through utilizing the proposed H-PSO.

1 Introduction

With the development of robot technology, there are many corresponding applications that have also been developed in the world. In the robotic field, obstacle avoidance is an important issue. Wherever robots go, they must avoid obstacles and dangers during their movements. So in order for them to avoid obstacles or dangers, there are many path planning algorithms that have been designed to solve this problem.

The potential field navigation (PFN) (Koren & Borenstein, 1991; Shimoda *et al*., 2005; Tu & Baltes, 2006; Kuo & Li, 2011) method is one of the traditional methods used in the path planning issue. This method provides a dynamic vector field to guide the robot, and it can also be executed in real time. However, the generated path may not always be the shortest one. Another method is the deterministic improved Q-learning method (Konar *et al*., 2013), which gives good results for the path planning of a mobile robot. Nevertheless, because of the increment of the calculated amount, this method is not useful in high-resolution image data. Another approach is multi-robot sensor-based coverage path planning (Yazici *et al*., 2014), which considers the remaining energy capacities in calculating the optimal path for the robot in narrow environments. However, the environmental map must be completely defined before the calculating process. In other words, this method may not be performed in an unknown environment. Then there is O(n log n) shortest-path algorithm (Jan *et al*., 2014), which gives good results and has a high

https://doi.org/10.1017/S0269888916000151 Published online by Cambridge University Press

calculating speed, but the obtained path may not be a safer path. Lastly, the integrated sensor path planning method (Lu *et al*., 2014) performs well in an unknown environment, but some additional sensor information is necessary to be adopted in the algorithm. In this paper, the path planning algorithm is mainly established on a vision-based system.

In nature, magnetic information will affect the migration direction of birds (Weindler *et al*., 1996), so this concept is adopted for the designed particle swarm optimization (PSO) (Kennedy & Eberhart, 1992; Campos *et al*., 2014; Zhigang *et al*., 2014) path planning algorithm in this paper. PSO is an intelligent algorithm which is inspired by the collective behaviors of birds, and robots can obtain the best path by utilizing the PSO path planning algorithm. However, the vector field of the PFN method can be treated as a geomagnetic field in the environment. Therefore, the proposed hybrid particle swarm optimization (H-PSO) algorithm also considers the effect of the PFN method. Moreover, a fuzzy logic controller (FLC) is designed to determine the dominant ratio of the PSO algorithm and the PFN method. The simulation and experimental results demonstrate the good performance of the proposed H-PSO algorithm. Comparisons with the genetic algorithm (Holland, 1992; Chen *et al*., 2013; Sun *et al*., 2013; Yoon & Kim, 2013) and the gravitational search algorithm (GSA) (Rashedi *et al*., 2009; Duman *et al*., 2012; Shaw *et al*., 2012) are also given in this paper. Finally, in order to verify the feasibility and practicality of the proposed method, the FIRA HuroCup Obstacle Run Event (FIRA Homepage), which is performed in a random environment, has been chosen for the validation.

The major contributions of this paper are as follows: (1) the proposal of a migrant-inspired path planning algorithm for the obstacle run; (2) the design of an FLC to adjust the dominant ratio of the PSO algorithm and the PFN method for improving the convergence speed; (3) the demonstration of the performance of H-PSO in comparison with other methods in the simulation results; (4) the verification of the feasibility and practicality of the simulation, experimental, and FIRA HuroCup competition results.

This paper is organized as follows. In Section 2, the PFN method is introduced. The attractive forces generated by the finish line are also clearly defined. The basic knowledge of PSO, the fitness function, and the details of the proposed H-PSO algorithm are addressed in Section 3. Section 4 gives the simulation, experimental, and competition results. Comparisons with other methods are also completely denoted in this section. Section 5 concludes the paper.

2 The potential field navigation method

In order to speed up the execution time of the obstacle avoidance task, the PFN method is applied in the proposed algorithm. In the potential field method, there are two types of electric charges: positive charge and negative charge. The electric charge is attracted to a charge with a different type of charge but is repulsive to a charge with the same type of charge. The attractive force makes the two different types of charges move toward each other, while the repulsive force makes charges with the same type of charge move away from each other.

In order to generate a repulsive force, the robot and obstacles are charged by the same type, while the finish line is charged by a different type. In so doing, the robot will tend to approach the finish line and avoid the obstacles. In this paper, the obstacles and robots are set by a positive electric charge, while the finish line is set by a negative electric charge.

2.1 The repulsive force

In this paper, the robot is charged by positive electric charge q_R, and the obstacles are charged by positive charge q_O. As shown in Figure 1, a repulsive force $\overrightarrow{F_{OR}}$ is generated by the charge of the obstacle and the robot. The arrow denotes the direction of the repulsive force and the length of the arrow denotes the magnitude of the repulsive force.

In the Obstacle Run Event, there are several obstacles on the field. Therefore, each obstacle is assigned a unique index I. The direction of the repulsive force can be defined from the following equations:

$$\overrightarrow{R_{\mathrm{ID}}} = (x_R - x_I, y_R - y_I) \tag{1}$$

https://doi.org/10.1017/S0269888916000151 Published online by Cambridge University Press

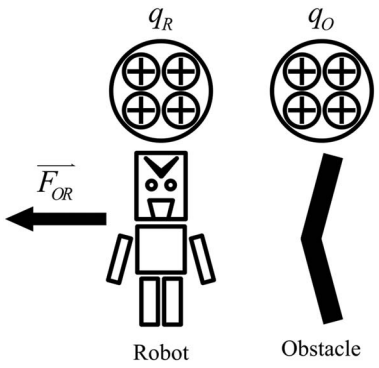


Figure 1 Repulsive force generated by the obstacle

$$\left\|\overrightarrow{R_{\mathrm{ID}}}\right\| = \sqrt{(x_R - x_I)^2 + (y_R - y_I)^2} \tag{2}$$

$$\overrightarrow{R_{\mathrm{ID,Unit}}} = \frac{\overrightarrow{R_{\mathrm{ID}}}}{\left\|\overrightarrow{R_{\mathrm{ID}}}\right\|} \tag{3}$$

where (x_I, y_I) is the center coordinates of the obstacle with index I and (x_R, y_R) is the center coordinates of the robot. $\overrightarrow{R_{\mathrm{ID}}}$ represents the vector from the obstacle to the robot. $\left\|\overrightarrow{R_{\mathrm{ID}}}\right\|$ is the distance between the robot and the obstacle. Finally, $\overrightarrow{R_{\mathrm{ID,Unit}}}$ is a unit vector of $\overrightarrow{R_{\mathrm{ID}}}$. And the repulsive force is defined by the following equations:

$$q_I = \frac{A_I}{K_{\mathrm{Region}}} \tag{4}$$

$$\overrightarrow{R_I} = \begin{cases} q_R \times q_I \times \overrightarrow{R_{\mathrm{ID,Unit}}}, & \left\|\overrightarrow{R_{\mathrm{ID}}}\right\| \le \tau \\ q_R \times q_I \times \overrightarrow{R_{\mathrm{ID,Unit}}} \times e^{-K_{\mathrm{Decay}} \times \frac{\left\|\overrightarrow{R_{\mathrm{ID}}}\right\| - \tau}{\tau}}, & \left\|\overrightarrow{R_{\mathrm{ID}}}\right\| > \tau \end{cases} \tag{5}$$

where A_I is the area of the obstacle. q_I is the quantity of the electric charge on the obstacle. K_{Region} is a constant and denotes the unit area. Within the same distance, the robot can receive a more powerful repulsive force from the obstacle that has a large quantity of electric charge.

The safety distance τ controls the influence region and is related to the robot's size. In addition, the decay parameter K_{Decay} controls the decrease rate. When the robot is far from the obstacle, the repulsive force can be neglected to reduce the computation cost. The difference between a strong charge and a weak charge is illustrated in Figure 2. Because the quantities of electric charge on the obstacles are not the same, the magnitudes at the center of the obstacle are also different. Figure 3 shows the repulsive vector. The influence regions from each obstacle are marked with colored rectangles. The influence region is related to the area of the obstacle region; the bigger obstacle produces a wider influence region.

2.2 The attractive force

The robot is charged by a positive electric charge q_R, and the finish line is charged by a negative charge q_L. As shown in Figure 4, attractive force $\overrightarrow{F_{RL}}$ is generated by the finish line. The arrow denotes the direction of the attractive force.

In this paper, the successive negative charges are charged on the finish line. The idea of the parallel vector field is derived by an infinitely long line charge. Figure 5 shows a positive electric charge and a negative infinite line charge. The attractive force on the positive electric charge can be divided into two parts: a right part and a left part. The x component of the attractive forces (F_{xL} and F_{xR}) are in opposite directions at any midway point and they cancel each other out. After the cancellation, the y component of

https://doi.org/10.1017/S0269888916000151 Published online by Cambridge University Press

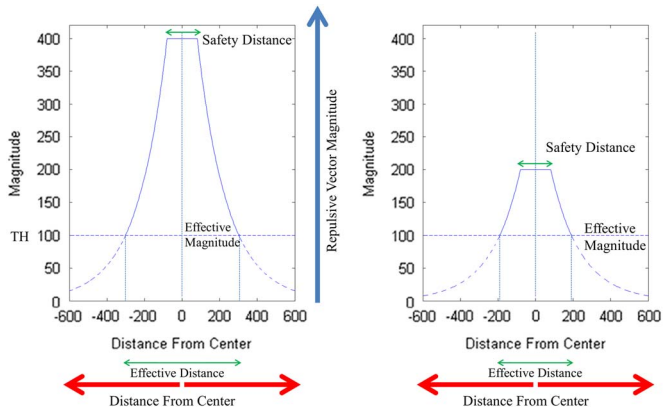


Figure 2 Comparison of the effective distance between a strong charge and a weak charge

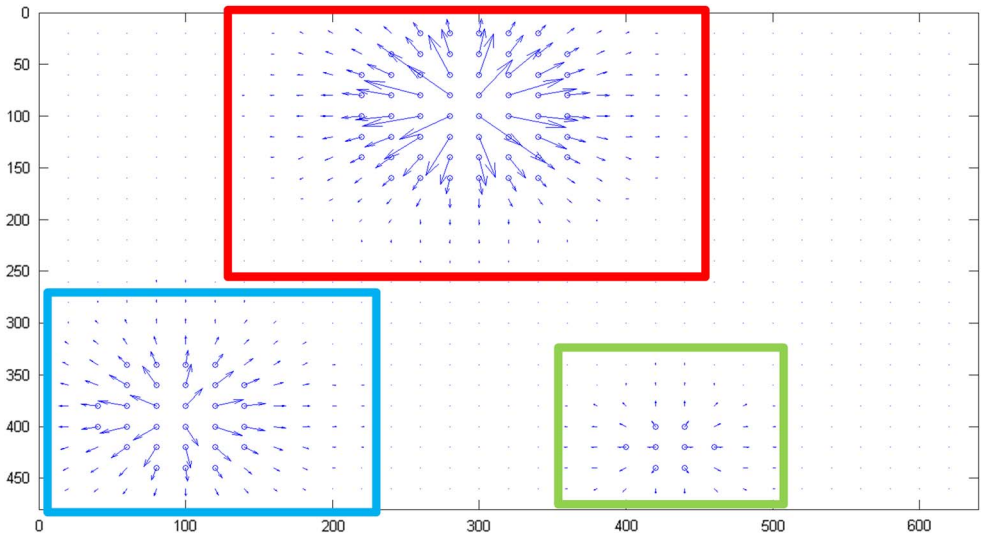


Figure 3 The repulsive vector field

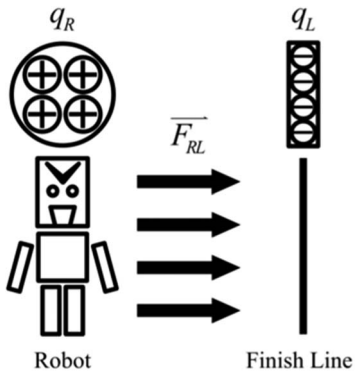


Figure 4 Attractive force generated by the finish line

the attractive forces (F_{yL} and F_{yR}) still remain. However, the direction of the remaining force is orthogonal to the line. For simplicity, although the actual finish line is not an infinity line, the direction of the parallel vector can be set to be orthogonal to the finish line.

Figure 6 shows an illustration of the attractive direction of the finish line. Line T depicts the finish line as well as the lines. N_1 and N_2 are the orthogonal lines which intersect with line T at the

https://doi.org/10.1017/S0269888916000151 Published online by Cambridge University Press

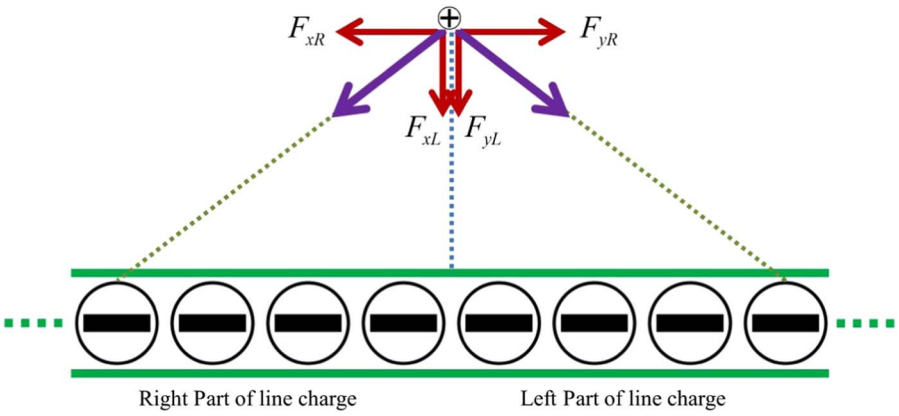


Figure 5 The positive electric charge in the parallel vector field

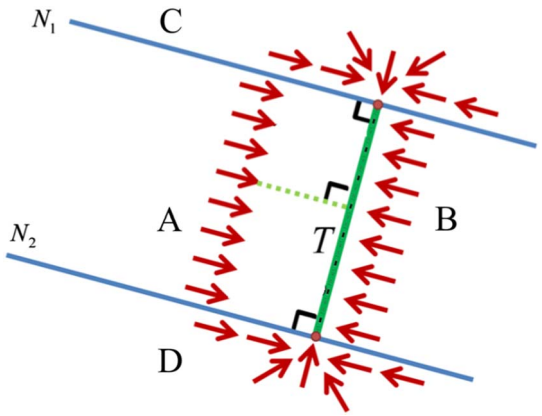


Figure 6 Illustration of the attractive direction of the finish line

endpoints of the finish line. Lines T, N_1, and N_2 divide the entire attractive vector field into four regions. Region A and region B are the parallel vector field, and region C and region D are the radial vector field.

2.3 The net force

Figure 7 shows the combined forces by attractive force $\overrightarrow{F_A}$ as well as the repulsive force $\overrightarrow{F_R}$ from the obstacle. The net force $\overrightarrow{F_{\text{net}}}$ consists of all the attractive forces and the repulsive forces, as shown in the following equation:

$$\overrightarrow{F_{\text{net}}} = \frac{\overrightarrow{F_A} + \sum_{I=1}^{O} \overrightarrow{F_{RI}}}{\left\| \overrightarrow{F_A} + \sum_{I=1}^{O} \overrightarrow{F_{RI}} \right\|} \tag{6}$$

where $\overrightarrow{F_{\text{net}}}$ is a vector with a unity magnitude and I is the index of the obstacles. O is the number of obstacles. The robot can move from its present position to the next position along the vector field.

The main advantage of the PFN method is its low computation cost. When the map of the playing field is obtained by the vision system, the vector field can be calculated immediately. However, the estimated path generated by the vector field may not always be the shortest path. Therefore, the PSO algorithm for the obstacle avoidance issue is proposed in the following section.

https://doi.org/10.1017/S0269888916000151 Published online by Cambridge University Press

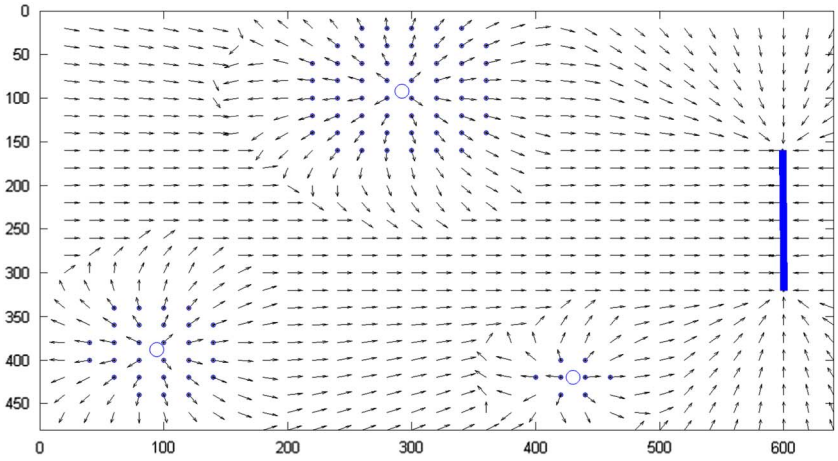


Figure 7 The vector field built by the obstacles and the finish line

3 Proposed hybrid particle swarm optimization algorithm

PSO is a heuristic optimization algorithm which is inspired by animal swarm behaviors. The update rule of each particle can be defined as the following equations:

$$\overrightarrow{V_i}(t+1) = w \times \overrightarrow{V_i}(t) + c_1 \times \varphi_1 \times \left(X_{i,L\text{Best}} - X_i(t)\right) + c_2 \times \varphi_2 \times \left(X_{G\text{Best}} - X_i(t)\right) \tag{7}$$

$$X_i(t+1) = X_i(t) + \overrightarrow{V_i}(t+1) \tag{8}$$

where $\overrightarrow{V_i}(t)$ is the velocity which controls the moving direction of ith particle at time t. $X_{i,L\text{Best}}$ is the local best location for each particle and $X_{G\text{Best}}$ is the global best location among all particles. w is the inertia weight. c_1 is the cognition parameter and c_2 the social parameter. The random parameters φ_1 and φ_2 are the random numbers within [0,1]. In this paper, the number of turning points N is set to be 8, the population size M is set to be 10, and the iteration number E is set to be 300.

The shortest path can be obtained by the PSO algorithm. However, in the PSO algorithm, the computation cost is higher than that of the potential field method. In the FIRA HuroCup Obstacle Run competition, the robot must walk to the finish line (without touching any obstacle) within a specific time. Therefore, the PSO algorithm and the PFN method are combined for this work.

3.1 The particle swarm optimization path definition

There are two goals for this optimization: one is to obtain the shortest path and the other is to achieve obstacle avoidance. Figure 8 shows the PSO path definition. First, the start point and the end point must be defined. Neither the start point nor the end point is unique. In this work, the start point is set on the start line or on the bottom edge of the robot vision. However, the end point needs to be set on the finish line or on the upper edge where the robot needs to approach. The number of paths is set to be M and the path is denoted as P. One path P can be separated into N line segments L. Besides, there are also some turning points included in the path. The PSO algorithm can adjust the location of these turning points Q and generate the path by connecting all the turning points in order. The number N of turning points influences the performance of the algorithms. More turning points may generate smoother paths. However, the increment of N of turning points may also increase the computation cost. Moreover, the safety width is defined by the width of the robot. The safety region is used to ensure that the robot can walk toward the finish line without touching obstacles. For example, in Figure 8, path P_2 is safer than path P_1 because there are some overlapping regions between the obstacle region and the safety margin region of path P_1 (the blue rectangle region).

https://doi.org/10.1017/S0269888916000151 Published online by Cambridge University Press

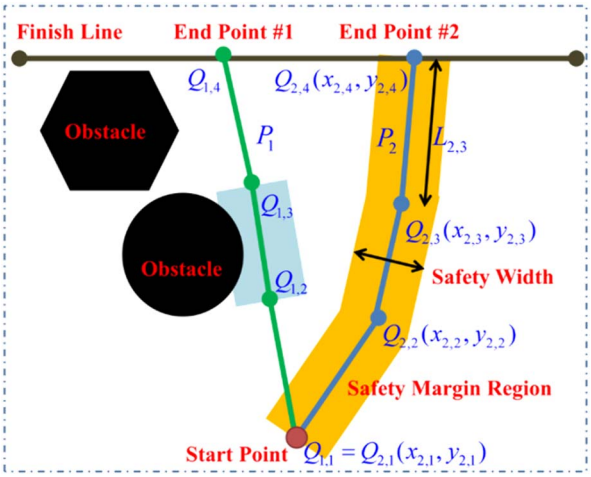


Figure 8 The particle swarm optimization path definition

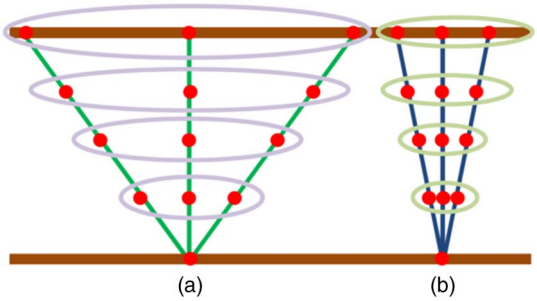


Figure 9 The divergence for path distribution. (a) The more scattered divergence. (b) The closer divergence

3.2 Behavior identification for the proposed algorithm

In order to identify the behavior mode of exploration and exploitation, the divergences of all the particles are considered. Figure 9 explains the divergences of all the particles. Initially, the distribution is like Figure 9(a) for searching all the given space with exploration behavior. After a period of time, the particles start to converge with a distribution as in Figure 9(b). In this work, the summation of standard deviation of the particle's turning points is used to judge the divergence of the particle distributions.

Besides, if the current standard deviation is larger than the previous one, this means that the particles become more decentralized. On the contrary, when the current standard deviation is smaller than the previous one, this means that the particles become closer. The variation of the standard deviation at time t is defined by $\dot{\sigma}(t)$, as shown in the following equation:

$$\dot{\sigma}(t) = \sigma(t) - \sigma(t-1) \tag{9}$$

3.3 Fuzzy logic controller for the proposed algorithm

In order to apply the FLC in the proposed algorithm, the membership functions are defined in Figures 10 and 11. The input variables are all decomposed into three fuzzy partition terms. S (small), M (medium), and B (big) are defined for the standard deviation σ; N (negative), Z (zero), and P (positive) are defined for the variation of the standard deviation $\dot{\sigma}$.

The output of the FLC is defined as W_{field}, and it controls the weighting value between the potential field method and the PSO method. If W_{field} is >0.5, the dominant method is the potential field method; if W_{field} is <0.5, the dominant method is PSO. The membership function of the output variable is also decomposed into three fuzzy partition terms: B (big), M (medium), and S (small). Figure 12 shows the output membership function μ_{output} for the output variable W_{field}, and the fuzzy rule table is shown in Table 1.

https://doi.org/10.1017/S0269888916000151 Published online by Cambridge University Press

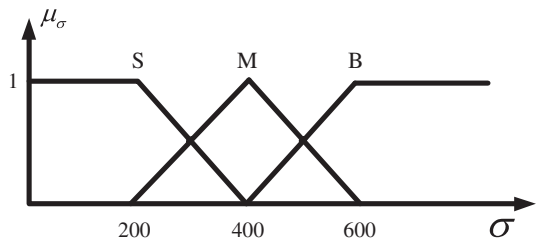


Figure 10 The membership function of σ

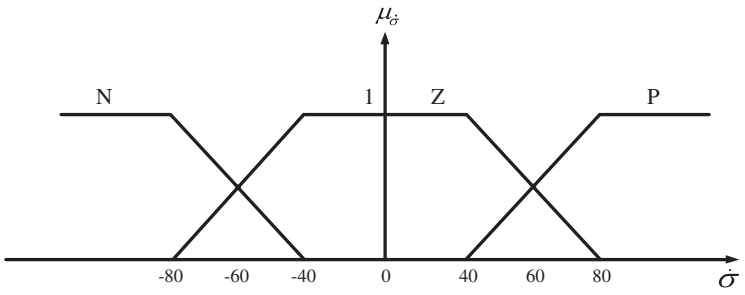


Figure 11 The membership function of $\dot{\sigma}$

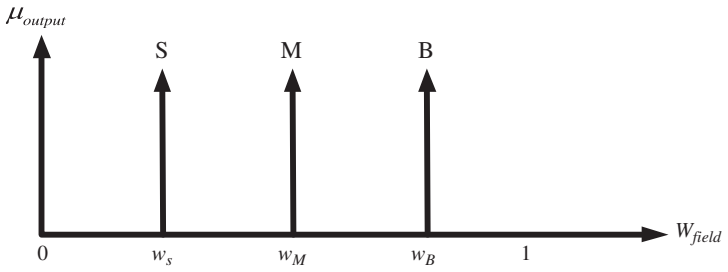


Figure 12 The membership of the output W_{field}

Table 1 The rule table of the fuzzy logic controller

σ \ $\dot{\sigma}$	N	Z	P
S	S	S	M
M	S	M	B
B	M	B	B

N = negative; Z = zero; P = positive; S = small; M = medium; B = big.

Moreover, the effect of the potential field vector on the turning point is shown in Figure 13, where $Q_{i,j}$ is the jth turning point of the ith particle. $L_{i,j}$ is the line segment of the path connected by $Q_{i,j}$ and $Q_{i,j+1}$; the direction of vector $\overrightarrow{L}$ is from $Q_{i,j}$ to $Q_{i,j+1}$. The vector $\overrightarrow{s_{i,j}}$ depicted in red is the potential field unit vector on turning point $Q_{i,j}$. The vector $\overrightarrow{s_{i,j}}$ is multiplied with the length of line segment $L_{i,j}$ to generate the vector $\overrightarrow{S}$. One end point of vector $\overrightarrow{S}$ is $Q_{i,j}$, the other one is denoted as $Q'_{i,j+1}$. Therefore, the vector $\overrightarrow{U}$ can be obtained by $\overrightarrow{U} = \overrightarrow{S} - \overrightarrow{L}$. The unit vector of $\overrightarrow{U}$, $\overrightarrow{u_{i,j}}$ is an important component of tuning the turning

https://doi.org/10.1017/S0269888916000151 Published online by Cambridge University Press

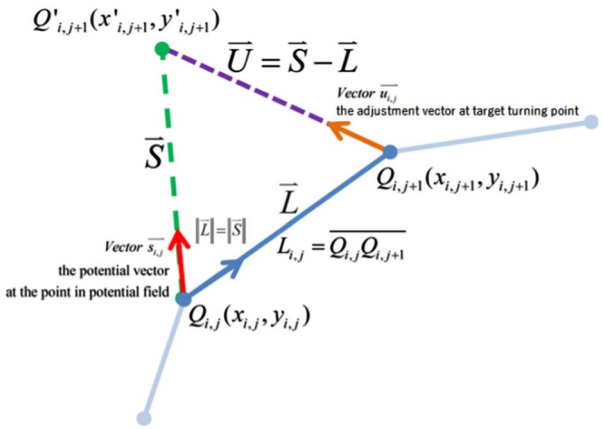


Figure 13 The effect of the potential field vector on the turning point

points and is called the potential field adjustment vector. Then, the PSO update rule can be rewritten as the following equations:

$$W_{\text{pso}} = (1 - W_{\text{field}}) \tag{10}$$

$$\overrightarrow{\alpha_{\text{pso}}} = w\,\overrightarrow{V_i}(t) + c_1\varphi_1\left(X_{i,L\text{Best}} - X_i(t)\right) + c_2\varphi_2\left(X_{G\text{Best}} - X_i(t)\right) \tag{11}$$

$$\overrightarrow{u_{i,j}} = \frac{\left\|\overrightarrow{L_{i,j}}\right\| \overrightarrow{s_{i,j}} - \overrightarrow{L_{i,j}}}{\left\|\overrightarrow{L_{i,j}}\right\|} \tag{12}$$

$$\overrightarrow{\alpha_{\text{pot}}} = \left\|\overrightarrow{L_{i,j}}\right\| \overrightarrow{u_{i,j}} \tag{13}$$

$$\overrightarrow{V_i}(t+1) = W_{\text{field}}\,\overrightarrow{\alpha_{\text{pot}}} + W_{\text{pso}}\,\overrightarrow{\alpha_{\text{pso}}} \tag{14}$$

where α_{pso} and α_{pot} are the tuning components of the PSO algorithm and potential field, respectively. W_{field} controls the behavior of the proposed hybrid algorithm. Figure 14 shows the flowchart of the proposed H-PSO algorithm. This flowchart can be divided into five partitions: vision system, PFN method, PSO algorithm, FLC, and motion control system. These partitions will be executed in order during the process of the proposed algorithm. However, if there is no acceptable path for the robot, this algorithm will be restarted from the PFN method.

3.4 The fitness function

Finally, the fitness function must be defined to judge the quality of the path. The fitness function of the proposed method is defined as the following equations:

$$T_i = \sum_{(x,y) \in \text{ safety region of } P_i} t_{(x,y)} \tag{15}$$

$$t_{(x,y)} = \begin{cases} 1000, & (x, y) \text{ is a turn point, and it is in the obstacle region} \\ 1, & (x, y) \text{ is not a turn point, and it is in the obstacle region} \\ 0, & \text{Otherwise} \end{cases} \tag{16}$$

https://doi.org/10.1017/S0269888916000151 Published online by Cambridge University Press

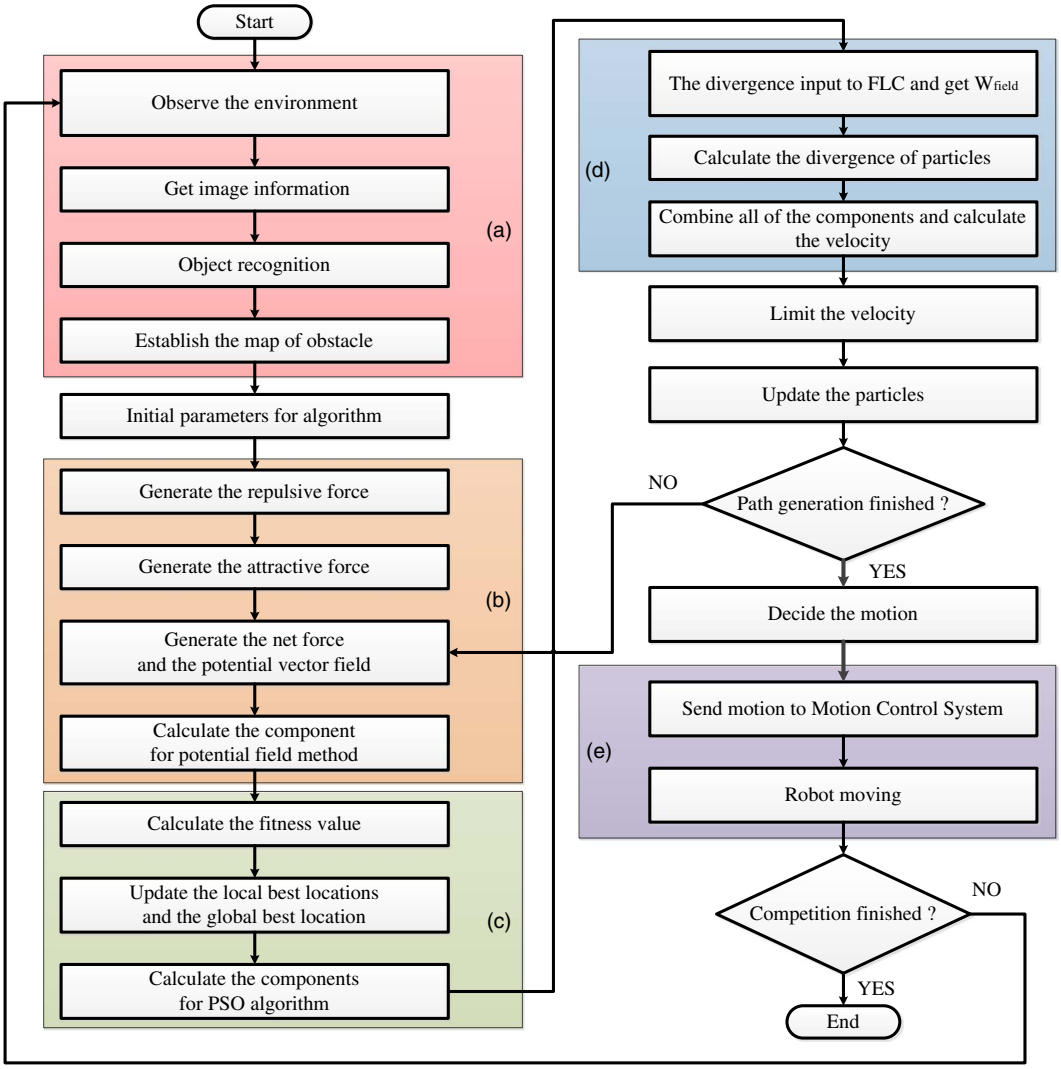


Figure 14 Flowchart for the proposed hybrid particle swarm optimization algorithm. (a) Vision system, (b) potential field navigation method, (c) PSO algorithm, (d) fuzzy logic controller (FLC) system, and (e) motion control system

$$fitness_i = \begin{cases} K_{\text{Touch}} \times e^{\frac{-T_i^2}{2\sigma^2}}, & T_i > 0 \\ K_{\text{Bias}} + K_{\text{Clear}} \times e^{\frac{D_i^2}{2\sigma^2}}, & T_i = 0 \end{cases} \tag{17}$$

where T represents the path's degree of danger. All the points in the safety region of the path will be checked. If the point at (x, y) is a turning point and it is in the obstacle region, $t_{(x,\,y)}$ will be set to 1000. On the other hand, if the point at (x, y) is not a turning point and it is in the obstacle region, $t_{(x,\,y)}$ will be set to 1. Otherwise, $t_{(x,\,y)}$ will be set to 0. Besides, D is the distance of the entire path. In this work, σ is set to be $800 \times (N-1)/3$ by the heuristic method. K_{Bias}, K_{Touch}, and K_{Clear} are the parameters for adjusting the boundary values of the fitness function. The fitness function is depicted in Figure 15.

4 Simulation and experimental results

4.1 Simulation results

In order to validate the performance of the proposed algorithm, the proposed method will be compared with other methods in this subsection. In this simulation, the obstacle distribution maps are input by the

https://doi.org/10.1017/S0269888916000151 Published online by Cambridge University Press

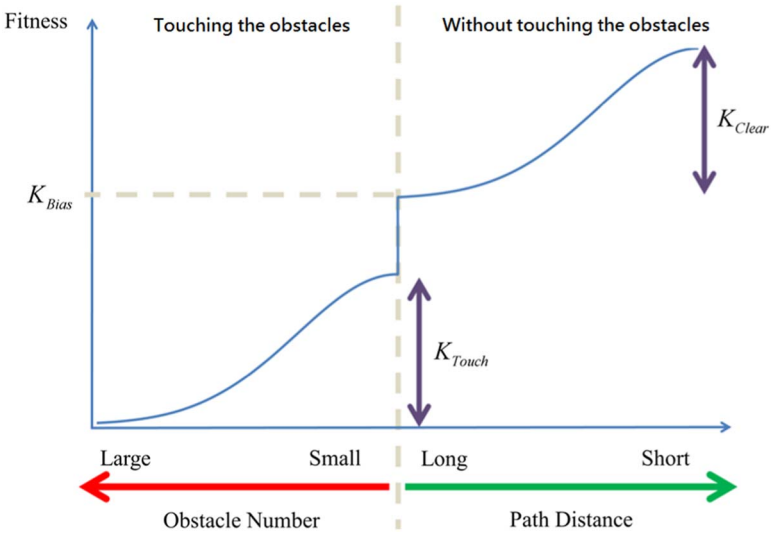


Figure 15 The fitness function

Table 2 Basic settings for each standard algorithm

GA	GSA	PSO
• Mutation rate: 0.1 • Crossover rate: 0.3	• Gravitational constant G: 400 • Attenuation coefficient t: 8	• Cognition parameter $c_1 = 2$ • Social parameter $c_2 = 2$ • Inertial weight w: 0.4

GA = genetic algorithm; GSA = gravitational search algorithm; PSO = particle swarm optimization.

Table 3 Methods for comparison

	GA	GSA	PSO
The standard algorithm	GA-Native	GSA-Native	PSO-Native
Applies the PFN method		GSA-PFN	PSO-PFN
Applies the fuzzy logic controller		H-GSA	H-PSO (the proposed algorithm)

GA = genetic algorithm; GSA = gravitational search algorithm; PSO = particle swarm optimization; PFN = potential field navigation; H-GSA = gravitational search algorithm; H-PSO = hybrid particle swarm optimization.

bitmap files for all the algorithms. The start point and the finish line are also defined before the process of the simulation. Table 2 shows the basic parameter setting for the computer simulations. Besides, the PFN method and the FLC are applied into GSA, PSO method, as illustrated in Table 3. Table 4 shows the parameter settings for all the methods.

Figure 16 shows the convergence process of the proposed algorithm. Figure 16(a) shows the initial state of the particles. In this situation, all the paths are randomly generated. In Figure 16(b), the turning points are moved downward by the forces of the potential vector field. The red line denotes the best path. After that, the proposed algorithm continuously makes the paths shorter, as shown in Figure 16(c). Finally, in Figure 16(d), all the candidate paths converge.

The simulation results of the path planning are represented in Figure 17. There are 100 testing times for each algorithm with one map. The average values of the learning curves are also depicted in Figure 18, and

https://doi.org/10.1017/S0269888916000151 Published online by Cambridge University Press

Table 4 Settings for the potential navigation method and the fuzzy logic controller (FLC)

	GA	GSA	PSO	GSA -PFN	PSO - PFN	Hybrid GSA	Hybrid PSO
Particles (Paths)	M=10						
Turning Points	N=8						
Fitness Value Parameter	K_{Touch}=K_{Clear}=100 K_{Bias}=400						
Potential Field				K_{Region}=2500 pixels τ=35 pixels			
The parameters for FLC output variable W_{field}				No FLC W_{field}=0.5		w_B={0.6,0.7,0.8,0.9,1.0} w_M=0.5 w_S=1-w_B	
Velocity Limit	[-40,40]						

GA = genetic algorithm; GSA = gravitational search algorithm; PSO = particle swarm optimization; PFN = potential field navigation.

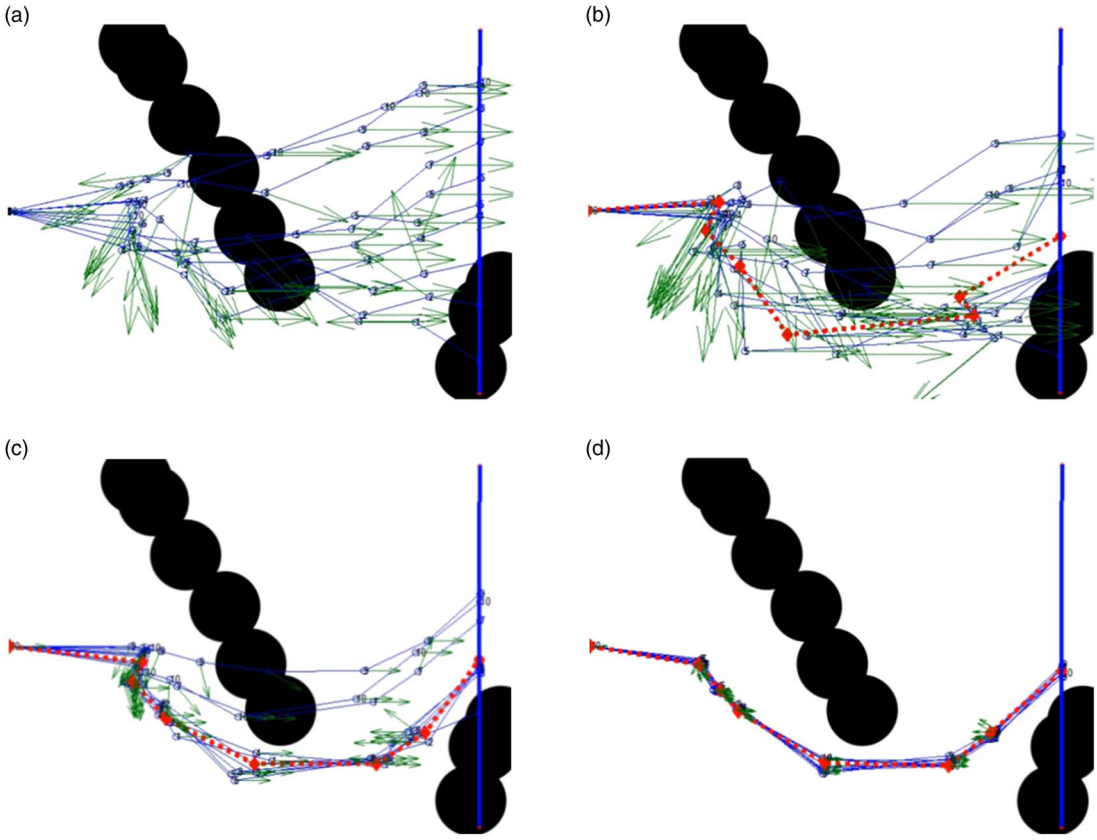


Figure 16 Convergence process of the proposed algorithm. The convergence progress is listed sequentially in (a)~(d)

the analysis results are illustrated in Table 5. The success ratio of 100 test runs is the most important index. The final averaged fitness value is also affected by the success ratio. Besides, the convergence rate is one of the important items in the performance. When the fitness value reaches 95% of the final averaged fitness value, the iteration number will be recorded for comparison. 'Run time' denotes the total time of 100 tests. In this work, an optimal path can be obtained within about 0.21 seconds by utilizing the proposed method. It is fast enough for a real-time robot application.

https://doi.org/10.1017/S0269888916000151 Published online by Cambridge University Press

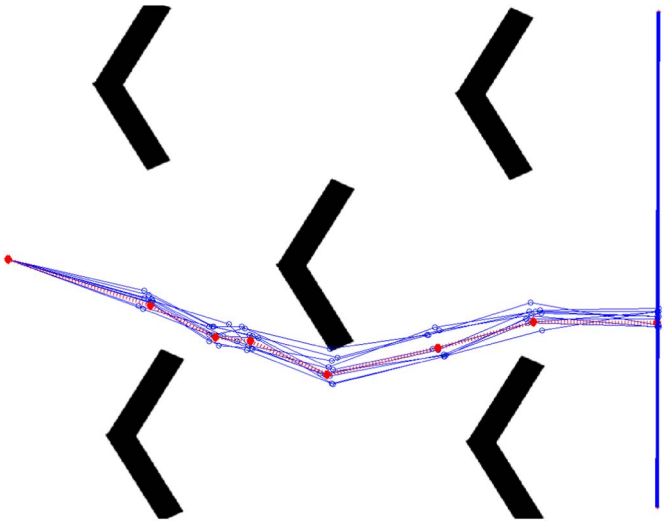


Figure 17 Simulation result of the path planning. PSO = particle swarm optimization; H-PSO = hybrid particle swarm optimization; PFN = potential field navigation; GSA = gravitational search algorithm; GA = genetic algorithm

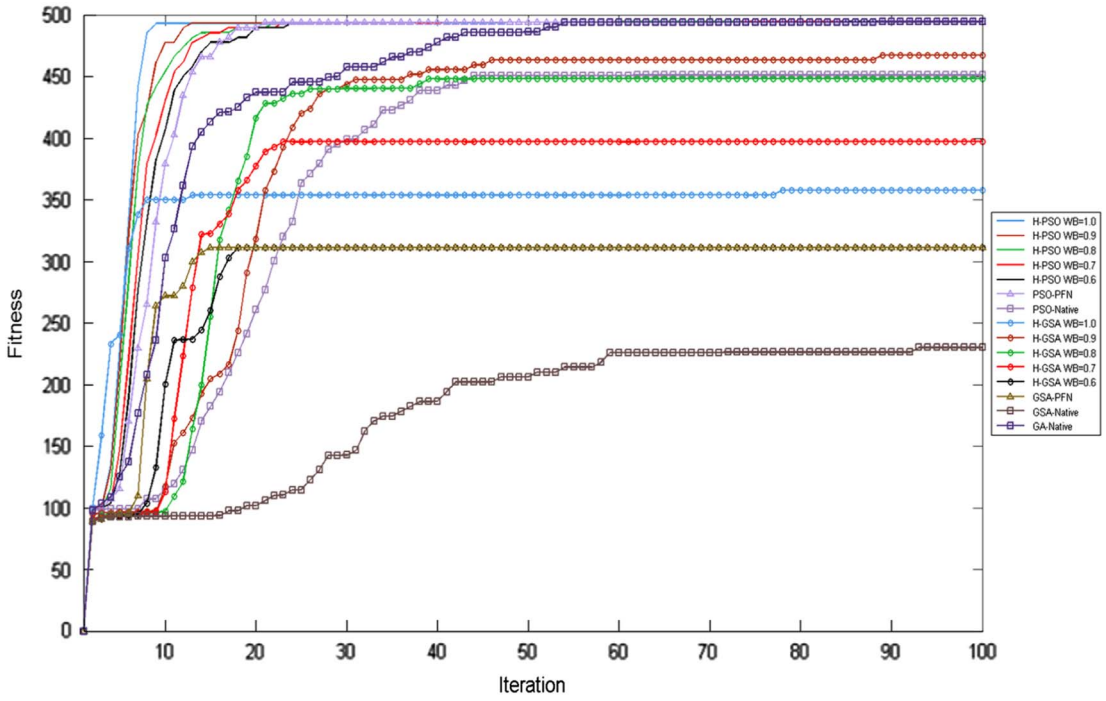


Figure 18 Learning curve for each algorithm

The simulation results show that the advantages of the proposed H-PSO algorithm are a higher success ratio, higher averaged fitness value, faster convergence rate, and acceptable computation time. The computation cost of H-PSO is indeed higher than PSO-Native. However, the convergence speed and the success rate of H-PSO are higher than PSO-Native. Furthermore, the computation time of these two methods are still almost the same, as shown in Table 5. If the fitness value is larger than K_{Bias}, the path is free from obstacles. Therefore, Table 5 also contains the iteration number when the fitness value is over K_{Bias}. Although the computation cost slightly increases when the PFN method is added, the success ratio and the performance are significantly improved. The proposed FLC also provides a good performance. In this work, w_B is suggested to be set to 1.0 because the convergence speed of H-PSO with $w_B = 1.0$ is the

https://doi.org/10.1017/S0269888916000151 Published online by Cambridge University Press

Table 5 Analysis of the simulation result

	w_B	Success rate (%)	Final averaged fitness value	Convergence iteration number	Fitness > K_{Bias} iteration number	Run time (100 tests) (seconds)
Hybrid PSO	**1.0**	**100**	**494.468**	**8**	**7**	**21.701**
Hybrid PSO	0.9	100	494.497	10	7	21.568
Hybrid PSO	0.8	100	494.471	12	8	21.407
Hybrid PSO	0.7	100	494.369	13	9	21.711
Hybrid PSO	0.6	100	494.297	14	10	21.574
PSO-PFN		100	494.177	16	11	21.762
PSO-Native		89	451.097	37	32	20.231
Hybrid GSA	1.0	66	357.755	8		32.706
Hybrid GSA	0.9	94	467.451	31	24	32.872
Hybrid GSA	0.8	89	447.948	21	20	32.494
Hybrid GSA	0.7	76	396.968	20		32.641
Hybrid GSA	0.6	54	310.965	17		32.884
GSA-PFN		54	311.182	13		31.930
GSA-Native		34	230.226	59		30.231
GA-Native		100	494.299	37	14	26.567

PSO = particle swarm optimization; PFN = potential field navigation; GSA = gravitational search algorithm; GA = genetic algorithm.
The bold values represent the performance of Hybrid PSO with $w_B = 1.0$. As shown in figure 18, the convergence speed of Hybrid PSO with $w_B = 1.0$ is the highest one. In this work, w_B is suggested to be set to 1.0.

highest one. The proposed algorithm, the H-PSO algorithm, not only generates the highest quality path but also provides a fast convergence speed. These two advantages are what the robot needs in the obstacle run completion.

4.2 Experimental results

Figure 19 shows the experimental results for the humanoid robot. The robot uses the proposed H-PSO algorithm to plan a path for the obstacle avoidance task. In Figure 19, the first column is the vision system of the robot, the second column shows the generated path by the proposed algorithm, and the third column denotes the real environment. When the path is generated, the decision-making system will decide on the appropriate motion and send the command to the motion control system. By utilizing the proposed method, the robot can walk toward the finish line without touching an obstacle.

4.3 Competition results

Figure 20 shows the competition results from FIRA HuroCup 2015. According to the rules, all the obstacles are randomly placed on the field. By using the proposed method, the robot accomplished the obstacle run event in real time. Moreover, the robot also won the championship at the FIRA HuroCup Obstacle Run Event 2015 in South Korea.

5 Conclusions

In this paper, the migrant-inspired path planning algorithm (H-PSO) is designed and implemented for the humanoid robot to accomplish the FIRA HuroCup Obstacle Run Event. The PFN method, the PSO, and the FLC are designed and combined in the proposed algorithm. By adopting the generated forces of the PFN method, the convergence speed is increased for this issue. The feasibility and the performance of the proposed method are also validated by the simulation, experimental, and competition results. Compared with other methods, the proposed algorithm provides fastest convergence speed and obtains the highest fitness values in the simulation results. Eventually, the proposed method produced good results in the 2015 FIRA HuroCup Obstacle Run Event.

https://doi.org/10.1017/S0269888916000151 Published online by Cambridge University Press

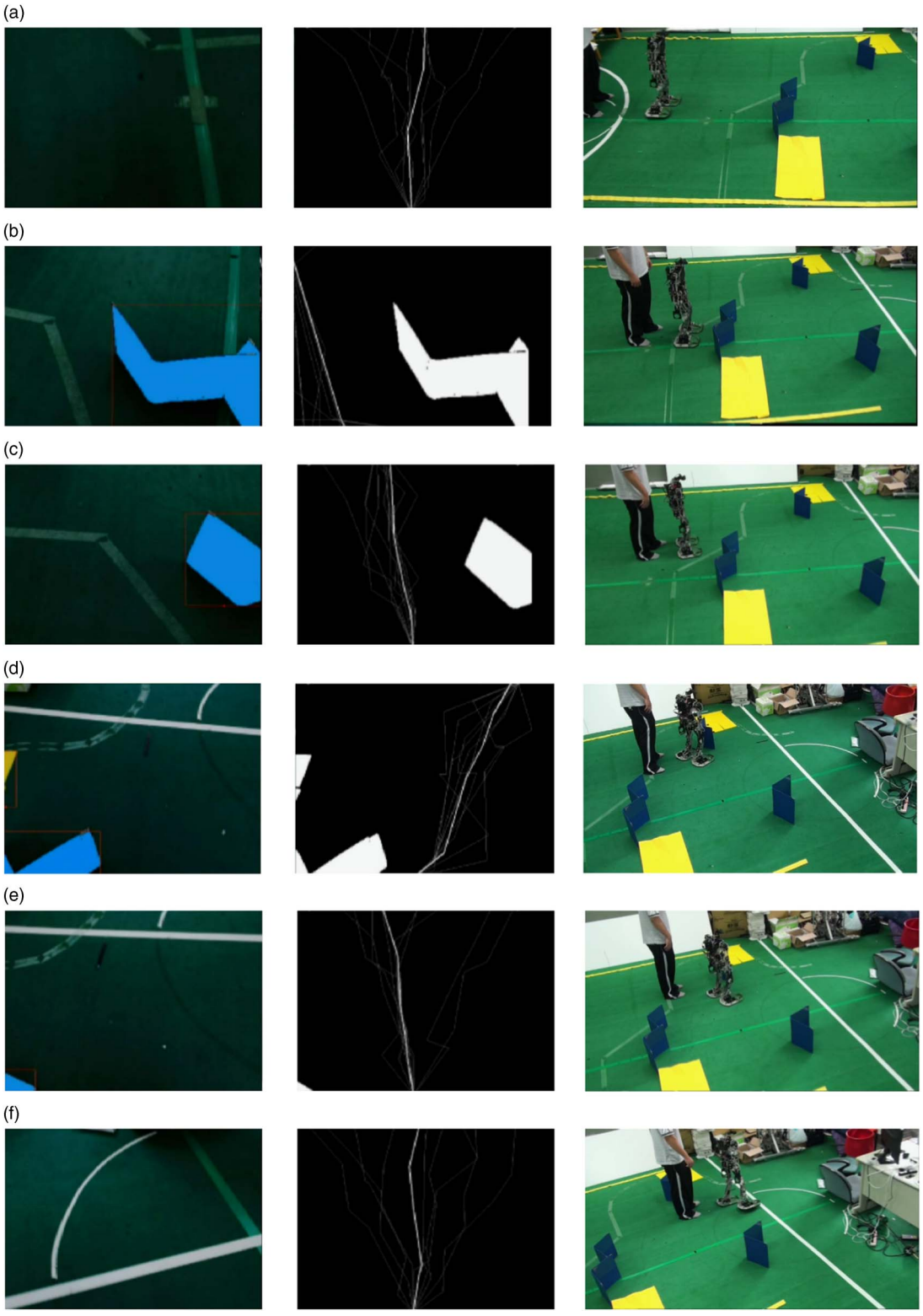


Figure 19 Experimental results. The obstacle run progress is listed sequentially in (a)~(f)

Acknowledgments

This work was supported in part by the Ministry of Science and Technology, Taiwan, ROC, under grants MOST 103-2221-E-006-252 and MOST 104-2221-E-006-228-MY2, and in part by the Ministry of

https://doi.org/10.1017/S0269888916000151 Published online by Cambridge University Press

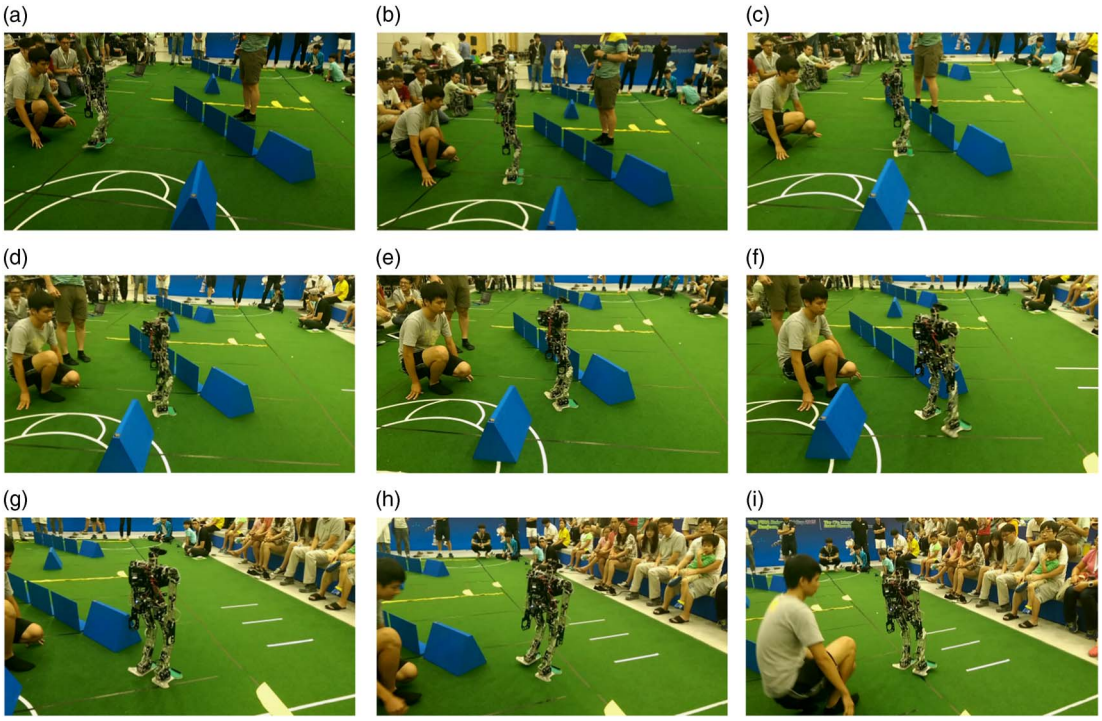


Figure 20 Competition results at FIRA HuroCup 2015. The competition progress is listed sequentially in (a)~(i)

Education, Taiwan, within the Aim for the Top University Project through National Cheng Kung University, Tainan, Taiwan.

References

Campos, M., Krohling, R. A. & Enriquez, I. 2014. Bare bones particle swarm optimization with scale matrix adaptation. *IEEE Transactions on Cybernetics* **44**(9), 1567–1578.

Chen, C.-H., Liu, T.-K. & Chou, J.-H. 2013. Integrated short-haul airline crew scheduling using multiobjective optimization genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **43**(5), 1077–1090.

Duman, S., Güvenç, U., Sönmez, Y. & Yörükeren, N. 2012. Optimal power flow using gravitational search algorithm. *Energy Conversion and Management* **59**, 86–95.

FIRA Homepage 2014. FIRA HuroCup Rules. `http://www.fira.net/contents/sub03/sub03_1.asp`.

Holland, J. H. 1992. Genetic algorithms. *Scientific American* **267**(1), 66–72.

Jan, G. E., Sun, C. C., Tsai, W. C. & Lin, T. H. 2014. An O (n log n) shortest path algorithm based on Delaunay triangulation. *IEEE/ASME Transactions on Mechatronics* **19**(2), 660–666.

Kennedy, J. & Eberhart, R. 1992. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, 66–72.

Konar, A., Chakraborty, I. G., Singh, S. J., Jain, L. C. & Nagar, A. K. 2013. A deterministic improved Q-learning for path planning of a mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **43**(5), 1141–1153.

Koren, Y. & Borenstein, J. 1991. Potential field methods and their inherent limitations for mobile robot navigation. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, 1398–1404.

Kuo, P.-H. & Li, T.-H. S. 2011. Development of simulator for AndroSot in FIRA. In *Proceedings of the FIRA 2011, CCIS* **212**, 233–240.

Lu, W., Zhang, G. & Ferrari, S. 2014. An information potential approach to integrated sensor path planning and control. *IEEE Transactions on Robotics* **30**(4), 919–934.

Rashedi, E., Nezamabadi-Pour, H. & Saryazdi, S. 2009. GSA: a gravitational search algorithm. *Information Sciences* **179**(13), 2232–2248.

Shaw, B., Mukherjee, V. & Ghoshal, S. P. 2012. A novel opposition-based gravitational search algorithm for combined economic and emission dispatch problems of power systems. *International Journal of Electrical Power & Energy Systems* **35**(1), 21–33.

https://doi.org/10.1017/S0269888916000151 Published online by Cambridge University Press

Shimoda, S., Kuroda, Y. & Iagnemma, K. 2005. Potential field navigation of high speed unmanned ground vehicles on uneven terrain. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2839–2844. http://ieeexplore.ieee.org/document/1570542/.

Sun, X., Gong, D., Jin, Y. & Chen, S. 2013. A new surrogate-assisted interactive genetic algorithm with weighted semisupervised learning. *IEEE Transactions on Cybernetics* **43**(2), 685–698.

Tu, K.-Y. & Baltes, J. 2006. Fuzzy potential energy for a map approach to robot navigation. *Robotics and Autonomous Systems* **54**(7), 574–589.

Weindler, P., Wiltschko, R. & Wiltschko, W. 1996. Magnetic information affects the stellar orientation of young bird migrants. *Nature* **383**, 158–160.

Yazici, A., Kirlik, G., Parlaktuna, O. & Sipahioglu, A. 2014. A dynamic path planning approach for multirobot sensor-based coverage considering energy constraints. *IEEE Transactions on Cybernetics* **44**(3), 305–314.

Yoon, Y. & Kim, Y.-H. 2013. An efficient genetic algorithm for maximum coverage deployment in wireless sensor networks. *IEEE Transactions on Cybernetics* **43**(5), 1473–1483.

Zhigang, R., Aimin, Z., Changyun, W. & Zuren, F. 2014. A scatter learning particle swarm optimization algorithm for multimodal problems. *IEEE Transactions on Cybernetics* **44**(7), 1127–1140.

https://doi.org/10.1017/S0269888916000151 Published online by Cambridge University Press