Path Searching Algorithms of Multiple Robot System Applying in Chinese Chess Game

Jr-Hung Guo[1], Kuo-Lan Su[2] and Sheng-Ven Shiau[1]
[1]Graduate school Engineering Science and technology,
National Yunlin University of Science & Technology
[2]Department of Electrical Engineering,
National Yunlin University of Science & Technology
Taiwan

1. Introduction

Chinese chess game [1] is one of the most popular games. A two-player game with a complexity level is similar to Western chess. In the recent, the Chinese chess game has gradually attracted many researcher's attention. The most researchers of the fields are belong to expert knowledge and artificial intelligent. There are many evolutionary algorithms to be proposed. Darwen and Yao proposed the co-evolutionary algorithm to solve problems where an object measure to guide the searching process is extremely difficult to device [2]. Yong proposed multi-agent systems to share the rewards and penalties of successes and failures [3]. Almost all the chess game can be described by game tree. Game tree presents the possible movements and lists all situations for the Chinese chesses. We want to use the multi-robots system to present the scenario of the chess movement for the Chinese chess game, and play the Chinese chess game according to the real-time image feedback to the supervised computer via wireless image system.

The application of co-evolutionary models is to learn Chinese chess strategies, and uses alpha-beta search algorithm, quiescence searching and move ordering [4]. Wang used adaptive genetic algorithm (AGA) to solve the problems of computer Chinese chess [5]. Lee and Liu take such an approach to develop a software framework for rapidly online chess games [6]. Zhou and Zhang present the iterative sort searching techniques based on percentage evaluation and integrate percentage evaluation and iterative sort into problem of Chinese chess computer game [7]. Su and Shiau developed smart mobile robots to speak real-time status using voice module, and program the motion trajectories for multiple mobile robot system [8].

With the robotic technologies development with each passing day, robot system has been widely employed in many applications. Recently, more and more researchers are interest in the robot which can helps people in our daily life, such as entertaining robots, museum docent robots, educational robots, medical robots, service robots, office robots, security robots, home robots, and so on. In the future, we believe that intelligent robots will play an important role in our daily life. In the past literatures, many experts researched in the mobile robot, and proposed many methods to enhance the functions of the mobile robot [9]. So far, developing a big sized mobile robot to be equipped with many functions to become

complex and huge, and the development period is too long. Thus, recently small-sized mobile robot system has been investigated for a specific task, and program the optimal motion path on the dynamic environment [18].
There is a growing in multi-robot cooperation research in recent year. Compare to single mobile robot, cooperation multiple mobile robots can lead to faster task completion, higher quality solution, as well as increase robustness owing its ability adjust to robot failure [10]. Grabowski and Navarro-serment [11] suggested multiple mobile robots in which each mobile platform has a specific sensor for some purpose and therefore the system's task can be distributed to each mobile platform during surveillance. The feature of this system is that each mobile robot has a common motion platform, but has different sensors. Chung et al. [12] composed of one ship and four small-sized search robots for team work in hazardous environment. Balch et al. [13] and Alami et al. [14] investigated cooperation algorithm of multiple robot system.
Some papers consider the problem of the multiple robot system working together. The multiple mobile robot system has more advantages than one single robot system [15]. The multiple mobile robots have the potential to finish some tasks faster than a single robot using ant colony system [16]. Multiple mobile robots therefore can be expected more fault tolerant than only one robot. Another advantage of multiple mobile robots is due to merging of overlapping information, which can help compensate for sensor uncertainty [17]. We have developed multiple small-size robot system to be applied in Chinese chess [8]. We extend the application field of the multiple mobile robot system, and program the shortest path moving on the chessboard platform using A* searching algorithm. The A^* heuristic function are introduced to improve local searching ability and to estimate the forgotten value [19].
We present the searching algorithms based Chinese chess game, and use multiple mobile robots to present the scenario on the chessboard platform. The mobile robot has the shape of cylinder and its diameter, height and weight is 8cm, 15cm and 1.5kg. The controller of the mobile robot is MCS-51 chip, and acquires the detection signals from sensors through I/O pins, and receives the command from the supervised compute via wireless RF interface. The chesses (mobile robots) can speak Chinese language for real-time status using voice module. We develop the user interface of multiple mobile robots according to the basic rules of Chinese chess game on the supervised computer. The mobile robot receives the command from the supervised computer, and calculates the displacement using the encoder module The supervised computer program the motion trajectories using evaluation algorithm and A* searching algorithm for the mobile robot to play the Chinese chess game. The A^* searching algorithm can solve shortest path problem of mobile robots from the start point to the target point on the chessboard platform.
The simulation results can found the shortest motion path for mobile robots (chesses) moving to target points from start points in a collision-free environment. The two points are selected by two players according to the rules of the Chinese chess game. In the experimental results, we test some functions of the mobile robot to obey the moving rules of the game on the chessboard, and implement the simulation results on the chessboard platform using mobile robots. Users can play the Chinese chess game on the supervised computer using the mouse. Mobile robots can receive the command from the supervised computer, and move the next position according to the attribute of the chess. The chess (mobile robot) moving scenario of the Chinese chess game feedback to the user interface using wireless image system.

2. System architecture

The system architecture of the multiple mobile robots based Chinese chess game system is shown in Fig 1. The system contains a supervised computer, a image system, some wireless RF modules, a remote supervised computer and thirty-two mobile robots. Mobile robots are classified two sides (red side and black side) belong two players. There are sixteen mobile robots in each side. The supervised computer can transmit the command signals to control mobile robots, and receives the status of the mobile robots via wireless RF interface. These signals contain ID code, orientation and displacement of mobile robots from the supervised computer to mobile robots. The feedback signals of mobile robots contain ID code, position (X axis and Y axis) to the supervised computer. Each robot is arranged an ID code to classify the attribute of the chess. Users move the chess on the user interface using the mouse. The supervised computer can controls the mobile robot moving the target position according the command. The mobile robot transmits the ending code to the supervised computer after finishing the task via wireless RF interface.

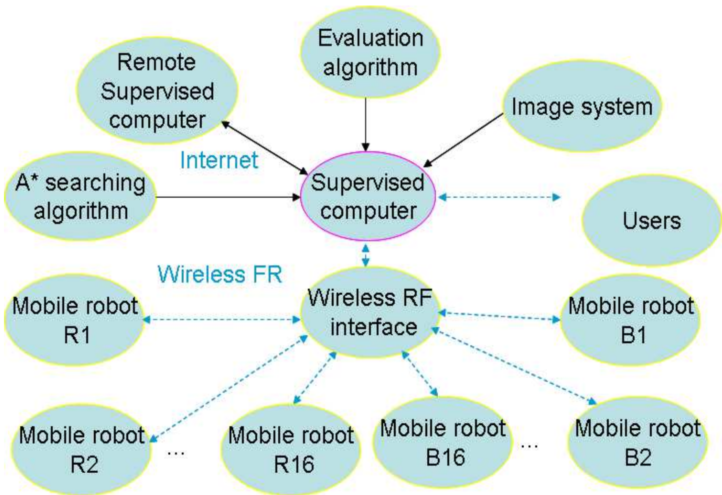


Fig. 1. The architecture of the mobile robots based Chinese chess game system.

The Chinese chess game system can transmit the real-time image to the supervised computer via image system. Users can play the Chinese chess game with others on the supervised computer using the mouse, or play the game on the remote supervised computer via wireless Internet. The player can move the chess piece using the mouse according to the rules of the Chinese chess game. Mobile robots receive the status from the supervised computer via wireless RF interface. There are two sides of the game system. One is the red side; the other is black side. Each side has sixteen chesses. There are one “king” chess, two “advisor” chesses, two “elephant” chesses, two “horse” chesses, two “rook” chesses, two “cannon” chesses and five “pawn” chesses. The definition of the chessboard is shown in Fig. 2. Then we define the initial position all chesses. Such as the position of “red king” is (4,0), and “black king” is (4,9)…etc.

The basic rules of the Chinese chess are found easily on the Internet. Before we want to control the multiple mobile robots based Chinese chess game. A chess engine needs to be designed and tested to ensure that all chess displacement and game rules are strictly adhered. We proposed the engine to be simple programmed basic rules of the games. The chessboard, denoted a 9 by 10 matrix, is the most important information that determine a players next moving position. We use axis position to define all chesses. First, the chessboard, denoted a axis position (x, y) from (0,0) to (8,9).

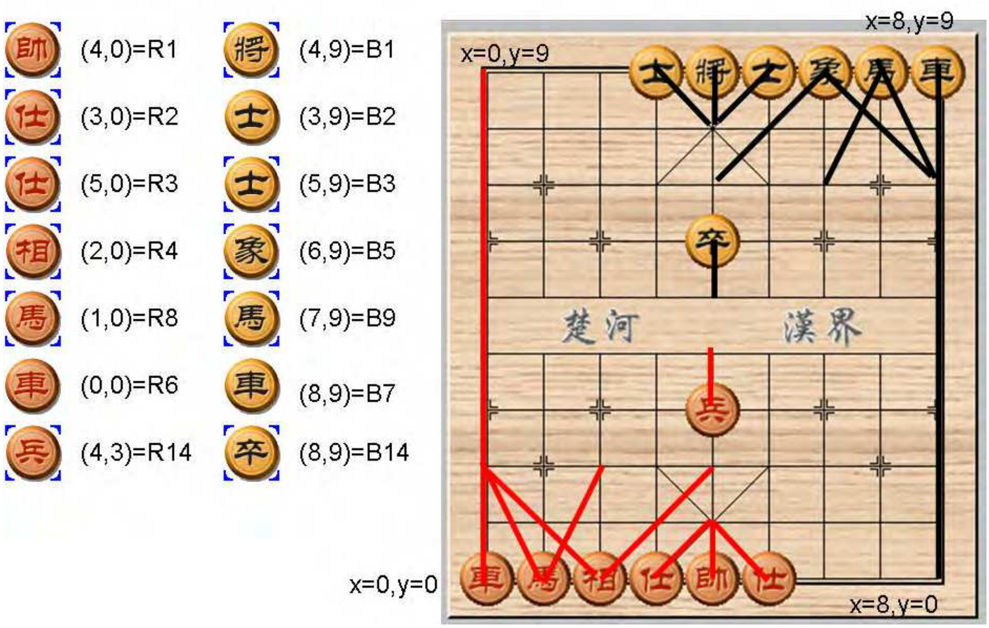


Fig. 2. The positions of the chesses.

We plot the possible motion trajectories using thick lines for the chesses on the chessboard. Then we define the game tree, and move the chess to the target position. Such as the chess "black horse" can moves to the position (8,7), (6,7) or (5,8). But the chess can't moves to the position (5,8) according the rules of the Chinese chess game. The chess "black horse" has an obstacle (black elephant) on the right side, and can't walk over the chessboard. We plot the possible motion trajectory for the chess piece on the chessboard. Then we define the game tree, and move the chess to the assigned position. Such as the chess "red horse" can moves to the position (0,2) or (2,2). But the chess can't moves to the position (3,1) according to the rules of the Chinese chess game.

The communication protocol of the multiple mobile robot system is 10 bytes. It contains one start byte, eight data bytes and one checksum byte. The supervised computer transmits 10 bytes to control the mobile robot, and the communication protocol of the control command is listed in Table 1. The mobile robot receives the command to discriminate the robot ID code to be right, and moves to the target point step by step, and transmits the environment status to the supervised computer on real-time. The communication protocol of the feedback data from the mobile robot is listed in Table 2. The byte 3 and 4 represents the positions of the mobile robot on X axis and Y axis.

Byte	0	1	2	3	4
Definition	Start byte	ID code	Robot ID code	X axis of start position	Y axis of start position
5	6		7	8	9
X axis of target position	Y axis of target position		Orientation	No use	Checksum

Table 1. The communication protocol of the supervised computer.

Byte	0	1	2	3	4
Definition	Start byte	ID code	Robot ID code	X axis of robot	Y axis of robot
5	6		7	8	9
Obstacle status	Orientation of robot		No use	No use	Checksum

Table 2. The communication protocol of the mobile robot.

3. Mobile robot

We develop two module based mobile robot (MBR-I and MBR-II) for the Chinese chess game. The mobile robot (module based robot-I, MBR-I) has the shape of cylinder, and it's equipped with a microchip (MCS-51) as the main controller, two DC motors and driver circuits, a reflect IR sensor circuits, a voice module, an encoder module, three Li batteries and some wireless RF modules. Meanwhile, the mobile robot has four wheels to provide the capability of autonomous mobility.

The mobile robot has some hardware circuits to be shown in Fig. 3. The encoder module uses two reflect IR sensors to calculate the pulse signals from the two wheels of the mobile robot, and calculates the moving distance on the chessboard platform. The power of the mobile robot is three Li batteries, and connects with parallel arrangement. The driver circuits can control two DC motors to execute the displacement command through PWM signal from the controller. The controller of the mobile robot can acquires the detection signals from sensors through I/O pins, and receives the control command via wireless RF interface. The switch input can turns on the power of the mobile robot, and selects power input to be Li batteries or adapter. The voice module can speak the real-time status of the environment for mobile robot moving on the chessboard. In the encoder module, we plot the white line and black line on the wheel of the mobile robot, and use two reflect IR sensors to calculate the pulse signals from the two wheels of the mobile robot. We can set the pulse number for per revolution to be P , and the mobile robot moves pulse number to be B . We can calculate the movement displacement D of the mobile robot using the equation.

$$D = 4.25 \times \pi \frac{B}{P} \tag{1}$$

The diameter of the wheel is 4.25 cm. We design the new chessboard using grid based platform to be shown in Fig. 4. The arrangement of the chess board is 11 grids on the horizontal direction (X axis), and is 12 grids on the vertical direction (Y axis). The distance is 30cm between the center of corridor on the X axis and Y axis of the chessboard. The width of the corridor is 12cm. The mobile robot uses IR sensors to detect obstacles, and decides the cross points of the chess board. The game only uses 9 grids (horizontal) by 10 grids (vertical). We release two grids on each direction (horizontal and vertical) to arrange the leaving chesses of red side and black side. We put the mobile robots that are eaten moving around the platform. We improve the mobile robot (MBR-I) to be implemented in the new chessboard platform, and design new driver device using DC servomotor.

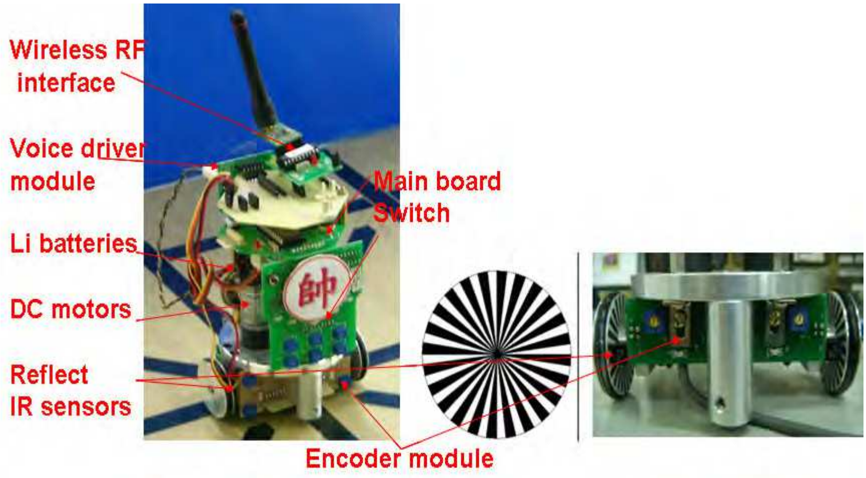


Fig. 3. The structure of the MBR-I.

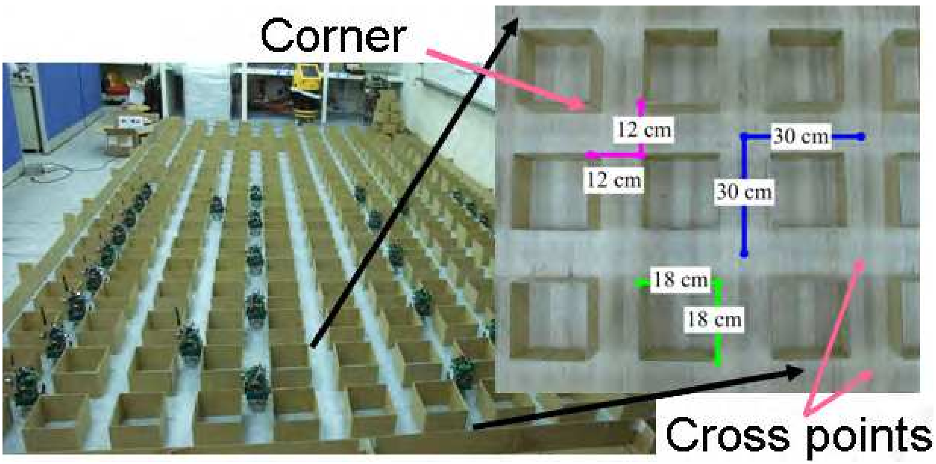


Fig. 4. The chessboard of the Chinese chess game.

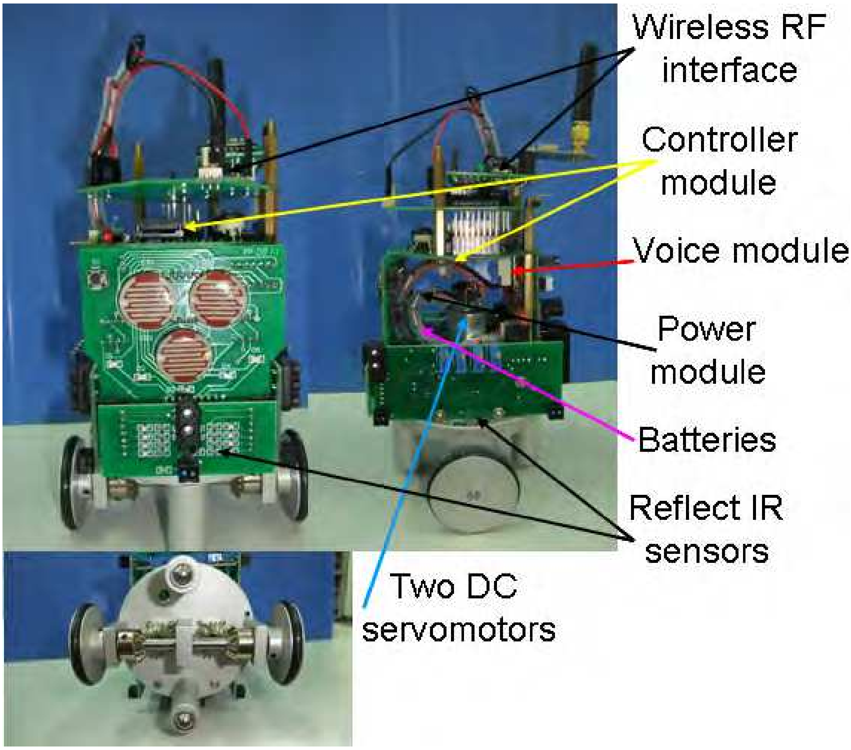


Fig. 5. The structure of the MBR-II.

The mobile robot (module based robot-II, MBR-II) has the shape of cylinder to be equipped with a microchip (MCS-51) as the main controller, and contains two DC servomotors and driver devices, some sensor circuits, a voice module, a compass module, three Li batteries, a wireless RF interface and three reflect IR sensor modules. Meanwhile, the mobile robot has four wheels to provide the capability of autonomous mobility. The structure of the mobile robot is shown in Fig. 5.

The controller calculates the orientation of the mobile robot from the compass module. The compass module has the error range to be 5^0, and transmits the measured values $(x \text{ and } y)$ in X axis and Y axis, and transmits to the controller of the mobile robot. The mobile robot can calculates the orientation angle θ is

$$\theta = Tan^{-1}(-y / x) \tag{2}$$

The distance is 30cm between the center of corridor on the X axis and Y axis, and the width of the corridor is 12cm. The mobile robot uses three IR sensors to detect the signals on the front side, right side and left side of the mobile robot, and decides the position of the obstacles, and detect the cross points. The definition of the orientation is east (X- axis), west (X+ Axis), south (Y+ axis) and north (Y- axis) on the motion platform. We set measurement range on the orientation of the mobile robot. For example, the measurement value of the orientation between 226^0 and 315^0, and define the direction of the mobile robot is west.

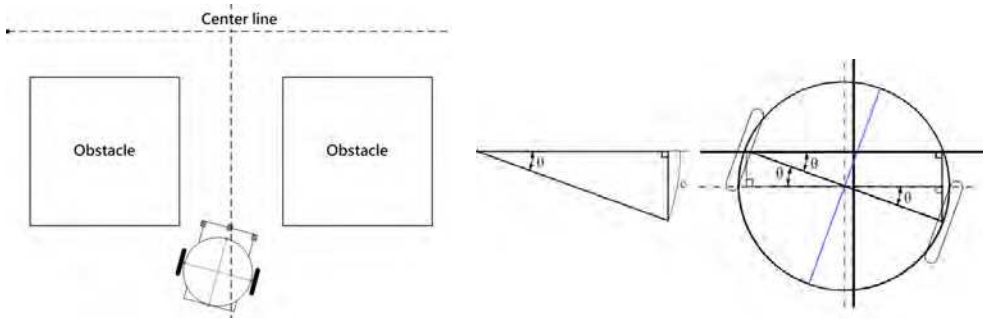


Fig. 6. The mobile robot tuning method.

The mobile robot moves on the corridor of the motion platform, and detects the location to be leaving the center line using the three reflect IR sensors. It tunes the motion path moving to the center line of the corridor. For example, the mobile robot leaves the center line, and moves on the left side of the center line. The mobile robot turns right θ (degree) to be set by users, and is calculated by the encoder module. Then it moves the displacement S according to Eq. (3), and moves to the center line. Finally, the mobile robot turns left θ to face the center line. The overview of the tuning schedule is shown in Fig. 6.

$$S = \frac{R\pi\theta}{180} \tag{3}$$

The parameter R (=4.25cm) is the diameter of the wheel. Furthermore, the mobile robot moves on the right side of the centre line. The tuning processing is the same as the previous method.

4. User interface

The operation interface of the multiple robot based Chinese chess game system is shown in Fig. 7. There are two regions in the operation interface. The chessboard of the Chinese chess is the main monitor of the user interface. Next we explain the chessboard of the Chinese chess game, and describe how to use the interface. It can displays "communication port", "communication protocol" and "axis position" on the bottom side of the operation interface. We set the communication port is 2, and Baud rate is 9600. The start position and target position of each chess displays on the left-bottom side of the operation interface. We make an example to explain the operation interface to be shown in Fig. 7. Players can move "black elephant"to the target position (5,8) from the start position (3,10). The bottom side of the interface can displays the start position (3,10) and the target position (5,8). We develop the operation interface using Visual Basic language for the Chinese chess game system.
We make other example to explain the Chinese chess game interface to be shown in Fig. 8. We can move the chess "red horse" from the start position (2,1) to the next position (3,3) using the mouse. The supervised computer control "red horse" chess moving to the next position (3,3). The scenario of the operation interface is shown in Fig. 8. The bottom side of the operation interface can displays the start position (2,1) and the next position (3,3).

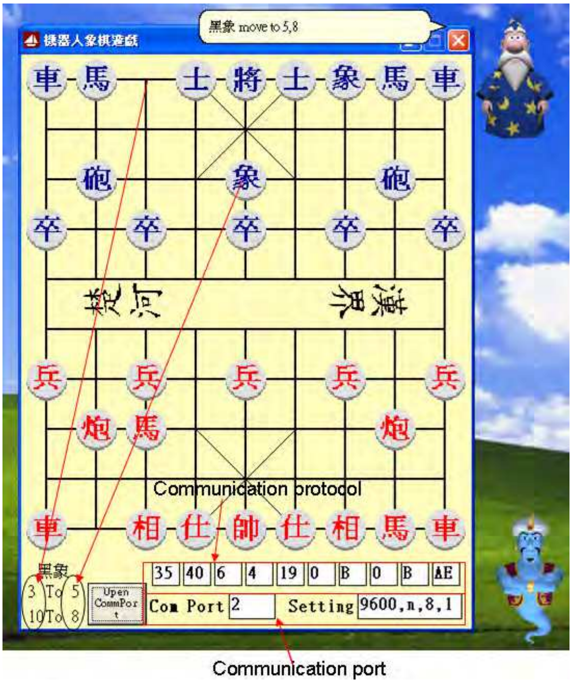


Fig. 7. The operation interface of moving the chess “black elephant“.

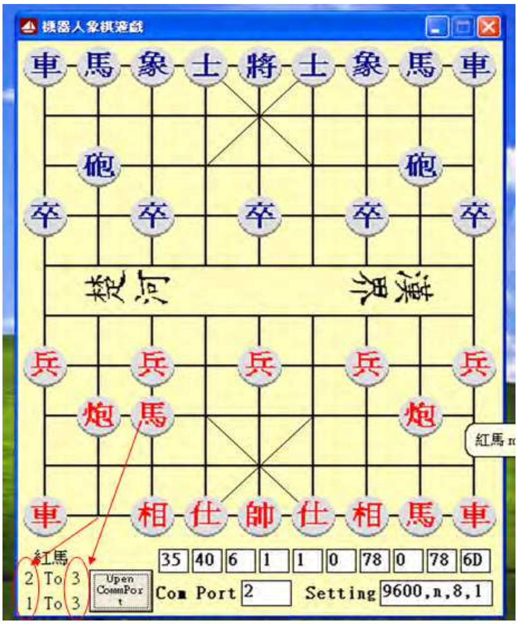


Fig. 8. The operation interface of moving the chess “red horse”.

The user interface of the multiple mobile robot based Chinese chess game system has four parts to be shown in left side of the Fig. 9. Players can start or close the game in the part "I". The part "II" can displays the communication protocol on the bottom of the monitor, and the communication port is 2, and Baud rate setting is 9600. The supervised computer transmits the control command to the mobile robot, and receives the status of the mobile robot, and displays the position of the mobile robot moving on the chessboard platform simultaneously to be shown in the part "III".

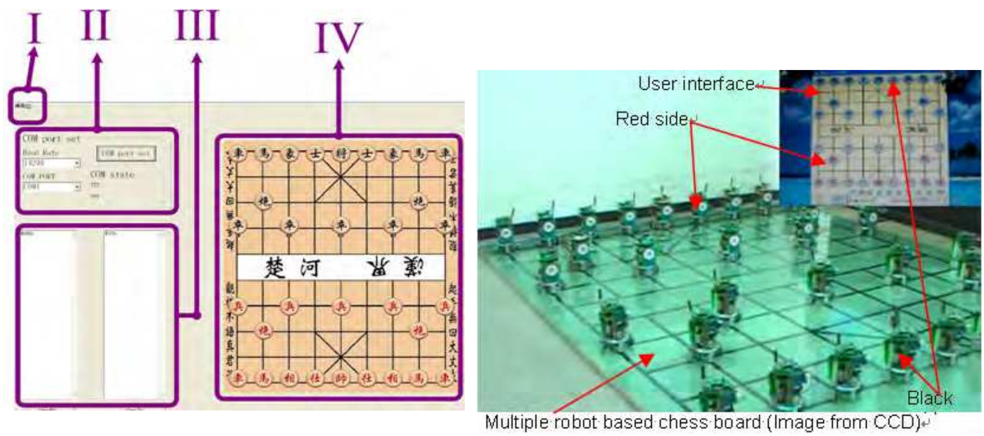


Fig. 9. The user interface of the game.

The part "IV" is chessboard platform. It can displays the motion path that is computed by evaluation algorithm and A* searching algorithm. The operation interface of the multiple robot based Chinese chess game system is shown in right side of the Fig. 9. There are two regions in the user interface. One is real-time image from the image system, and is equipped in the mobile robot based chessboard. The other is the chessboard of the Chinese chess game, and displays on the right side or the monitor. The players can move the chess using the mouse according the rules of the Chinese chess game. Mobile robots receive the status from the supervised computer via wireless RF interface, and move to the next position.

5. Evaluation method

The evaluation algorithm of the Chinese chess game uses the rule based method. Players can move the chess to the target position on the user interface. The position of the chess is not abided by the proposed rules. The user interface can't execute the command, and can't control the mobile robot moving to the target position. Players must renew the operation on the step, and obey the exact rules of the Chinese chess game. We can define the initial position of the chess piece is (x,y), and define the movement rules of all chesses as following. n is movement displacement on the x axis, and m is movement displacement on the y axis. n and m of the target position must be plus integer. The positions of all chesses must obey the rules that are listed in the Table 3 at any time.

Chess	Target position	Rules
Red king	$(x \pm n, y \pm m)$, $0 \le n \le 1$, $0 \le m \le 1$	$3 \le x \le 5$, $3 \le x \pm n \le 5$, $0 \le y \le 2$, $0 \le y \pm m \le 2$
Black king	$(x \pm n, y \pm m)$, $0 \le n \le 1$, $0 \le m \le 1$	$3 \le x \le 5$, $0 \le x \pm n \le 5$, $7 \le y \le 9$, $0 \le y \pm m \le 9$
Red advisor	$(x \pm n, y \pm n)$, $n = 1$	$3 \le x \le 5$, $0 \le x \pm n \le 5$, $0 \le y \le 2$, $0 \le y \pm n \le 2$
Black advisor	$(x \pm n, y \pm n)$, $n = 1$	$3 \le x \le 5$, $0 \le x \pm n \le 5$, $7 \le y \le 9$, $0 \le y \pm n \le 9$
Red elephant	$(x \pm n, y \pm n)$, $n = 2$	$0 \le x \le 8$, $0 \le x \pm n \le 8$, $0 \le y \le 4$, $0 \le y \pm n \le 4$
Black elephant	$(x \pm n, y \pm n)$, $n = 2$	$0 \le x, x \pm n \le 8$, $5 \le y, y \pm n \le 9$
Red rook, Red cannon Black rook, Black cannon	$(x \pm n, y \pm m)$, $n = 0$ or $m = 0$	$0 \le x, x \pm n \le 8$, $5 \le y, y \pm m \le 9$
Red horse, Black horse	$(x \pm n, y \pm m)$, $n + m = 3$	$0 \le x \le 8$, $0 \le x \pm n \le 8$, $0 \le y \le 9$, $0 \le y \pm m \le 9$
Red pawn	$(x, y + 1)$	$0 \le x \le 8$, $3 \le y \le 4$, $3 \le y \pm m \le 4$
	$(x \pm n, y + m)$, $n + m = 1$	$0 \le x \le 8$, $0 \le x \pm n \le 8$, $5 \le y \le 9$, $5 \le y + m \le 9$
Black pawn	$(x, y - 1)$	$0 \le x \le 8$, $5 \le y \le 6$, $5 \le y \pm m \le 6$
	$(x \pm n, y - m)$, $n + m = 1$	$0 \le x \le 8$, $x \pm n \le 8$, $0 \le y \le 4$, $0 \le y - m \le 4$

Table 3. The rules of Chinese chesses.

Mobile robots move on the chessboard to obey the rules of Chinese chess game. The user interface knows the start position and the target position, and programs the shortest path of the chess (mobile robot) moving to the target position from the start position. We use A* searching algorithm to find the shortest path on the Chinese chess game. A^* searching algorithm is proposed by Hart in 1968, and solved the shortest path problem of multiple nodes travel system. The formula of A^* searching algorithm is following

$$f(n) = g(n) + h(n) \tag{4}$$

The core part of an intelligent searching algorithm is the definition of a proper heuristic function $f(n)$. $g(n)$ is the exact cost at sample time n from start point to the target point. $h(n)$ is an estimate of the minimum cost from the start point to the target point. In this study, n is reschedules as n' to generate an approximate minimum cost schedule for the next point. The equation (5) can be rewritten as follows:

$$f(n) = g(n) + h(n') \tag{5}$$

We make an example to explain algorithm. Such as a mobile robot move to the target point "T" from the start point "S". The position of the start point is (2,6), and the target position is (2,9). We set some obstacle on the platform. The white rectangle is unknown obstacle. The black rectangle (obstacle) is detected by the mobile robot using A* searching algorithm. We construct two labels (Open list and Close list) in the right side of Fig. 8. The neighbour points of the start point fill in the "Open list". The "Close list" fills the start point and evaluation points. We construct label on the first searching result to be shown in Fig. 8. We calculate the values of $f(n)$, $g(n)$ and $h(n)$ function according to the pulse numbers by the encoder of the DC servomotor, and use the proposed method to compare the values of the function. We select the minimum value of the function $f(n)$ to be stored in "Close list". We can find the target point on the final searching result to be shown in Fig. 10, and we can decide a shortest path to control the mobile robot moving to the target point.
The total distance of the shortest path C_{st} can be calculated as

$$C_{st} = \sum_{n=1}^{m} G_n(m = t-1) = 2043 \tag{6}$$

In the other condition, we rebuild the positions of the obstacles on the platform to be shown in Fig. 11. The mobile robot can't find the path C_{st} moving to the target position using the proposed method. The total distance C_{st} as

$$C_{st} = \infty \tag{7}$$

Open	Closed	Point	F_{12}	G_{12}	H_{12}	D
(3,6)	S	(3,6)	1380	732	648	→
(1,6)	(2,7)	(1,6)	1380	732	648	↑
(1,7)	(3,7)	(1,7)	1056	570	486	↑
(6,8)	(3,8)	(6,8)	1704	864	810	↑
(6,9)	(4,8)	(6,9)	1338	690	648	↑
(6,10)	(5,8)	(6,10)	1662	852	810	↑
(4,11)	(5,9)	(4,11)	1338	690	648	↑
(3,11)	(5,10)	(3,11)	1014	528	486	↑
(2,11)	(4,10)	(2,11)	648	162	324	↑
	(3,10)					
	(2,10)					
	T					

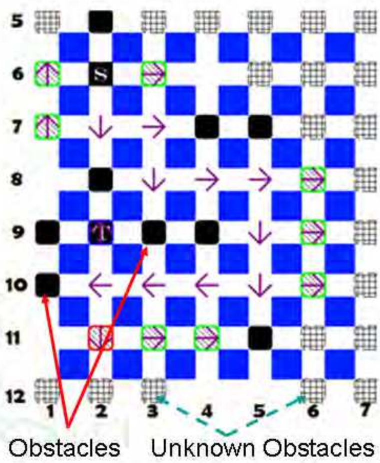


Fig. 10. The searching result for case 1.

Open	Closed	Point	F_1	G_1	H_1	D
	S					
	(2,7)					
	(3,7)					
	(3,8)					
	(4,8)					
	(5,8)					
	(5,9)					
	(5,10)					
	(6,10)					
	(6,9)					
	(6,8)					
	(3,6)					
	(4,6)					
	(1,6)					
	(1,7)					
	(1,8)					

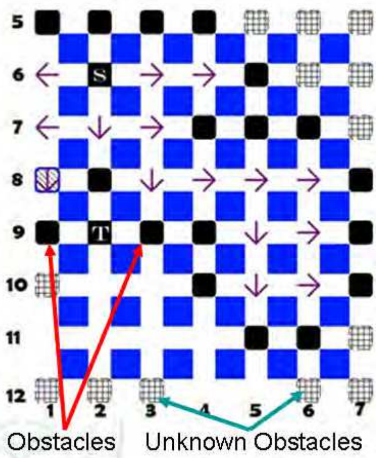
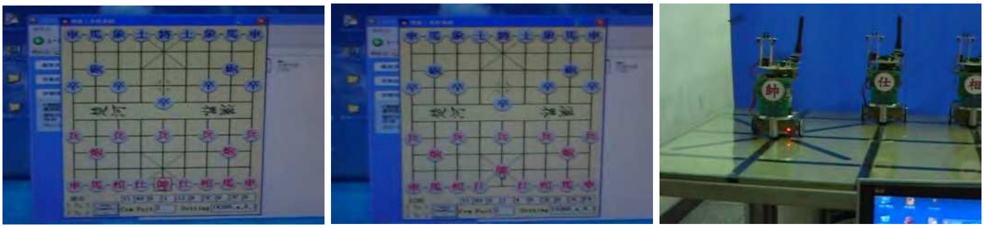


Fig. 11. The searching result for case 2.

6. Experimental results

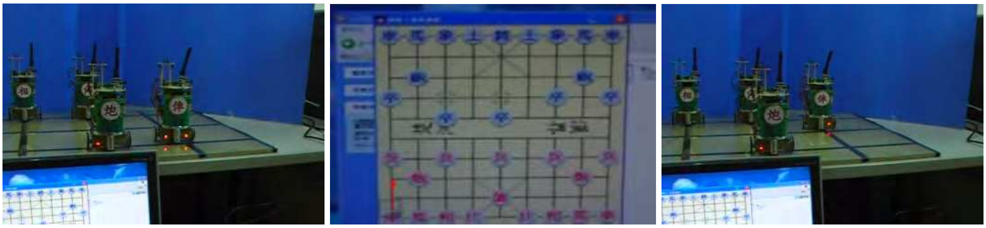
We execute some experimental scenarios using the MBR-I robot on the chessboard for the Chinese chess game system, and test the movement functions of some chesses (mobile robots). The two players are located by two sides, and move the chess one time each other. One moves red chesses; the other moves black chesses. The first experimental scenario is "red king". The player moves forward the chess "red king" using the mouse to be shown in Fig. 12 (a). The chess "red king" moves one grid to the target position on the user interface to be shown in Fig. 12 (b). The supervised computer orders the command to the mobile robot "red king" moving forward to the target position via wireless RF interface. The mobile robot can calculates the movement displacement using the encoder module, and speaks the movement status using the voice module. The experimental result is shown in Fig. 12 (c).

The second experimental scenario is "red rook". The player moves forward chess "red rook" using the mouse to be shown in Fig. 13 (a). The supervised computer orders the command



(a) (b) (c)

Fig. 12. The experimental result for “red king”.



(a) (b) (c)

Fig. 13. The experimental result for “red rook”.

to the mobile robot “red rook”. The mobile robot “red rook” moves to the target position (8,2) from the start position (8,0). The mobile robot can calculates the movement displacement using the encoder module, and stops at the target position. The experimental result is shown in Fig. 13 (b) and (c).erwe

The third experimental scenario is “red cannon”. The player moves the chess “red cannon” to right side using the mouse. The chess “red cannon” moves to the target position (4,2) from the start position (1,2) to be shown in Fig. 14 (a) and (b). The supervised computer orders the command to the mobile robot “red cannon” moving to the target position. The mobile robot receives the command to turn right, and moves to the target position. Then it can turn left and face to the black chess. The mobile robot calculates the movement displacement and the orientation using the encoder module, and speaks the movement status using the voice module. The experimental results are shown in Fig. 14 (c) and (d).

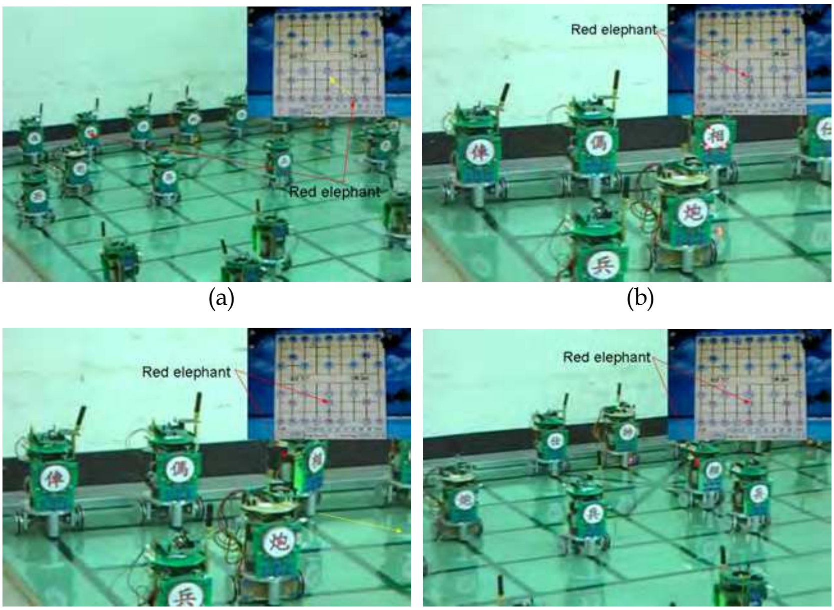
(a) (b)

Fig. 14. (Continued)

(c) (d)

Fig. 14. The experimental result for “red cannon”.

We implement the functions of the mobile robot to obey the movement rules of Chinese chess game. Then we execute experimental scenarios of the multiple mobile robots based Chinese chess game system. The first experimental scenario is “red elephant” that is moved by the player (red side). The player moves the chess “red elephant” to the target position (4,2) using the mouse to be shown in Fig. 15 (a) and (b). The supervised computer orders the command to the mobile robot “red elephant” via wireless RF interface. The start position of the mobile robot is (6,0). The movement orientation of the mobile robot is west-north. The mobile robot receives the command to turn left on 90^0, and moves forward to the target position according to the Chinese chess rules. The mobile robot speaks the movement status using voice module, and calculates the movement displacement using the encoder module, and moves to the target position (4,2). The experimental results are shown in Fig. 15 (c) and (d).

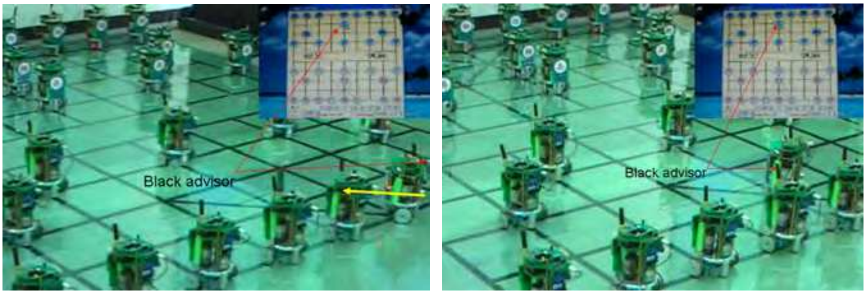


(a) (b)

(c) (d)

Fig. 15. The experimental result for “red elephant”.

The second experimental scenario is "black advisor" that is moved by the other player (black side). The player moves the chess "black advisor" using the mouse to be shown in Fig. 16 (a). The start position of the "black advisor" chess is (3,9). The supervised computer orders the command to the mobile robot "black advisor". The chess turn left on 90^0, and moves to the target position (4,8) and stop. The mobile robot can calculates the movement displacement using the encoder module, and speak the movement status using voice module, too. The experimental result is shown in Fig. 16 (b). The next step must moves red chess by the player (red side). The players (red side and black side) move one chess step by step on the chessboard each other. Finally, the user interface can decides which player to be a winner, and closes the game.



(a) (b)

Fig. 16. The experimental result for "black advisor".

The controller of the mobile robots calculates the movement displacement to be error, or loses pulse numbers from the encoder module. Mobile robots can't stay at the cross point of two black lines on the chessboard for a long time. We design the grid based chessboard platform to improve the weakness, and implement some experimental results using the multiple mobile robots (MBR-II) based Chinese chess game system. The first experimental scenario is "red elephant".

The user moves the chess "red elephant" using the mouse to be shown in Fig. 17 (a). The start position of the "red elephant" chess is (6,0). The supervised computer orders the command to the mobile robot "red elephant". The chess moves forward two grids, and turn left angle to be 90^0. Then the mobile robot moves two grids to the next position (4,2), and turn right angle 90^0 to face the black side and stop. The user interface of the Chinese chess game system uses multiple mobile robots to be shown in Fig. 17(a). The mobile robot can calculates the movement displacement via encoder of DC servomotor, and speaks the movement status of the mobile robot using voice module, too. The experimental scenarios use the mobile robot to execute the motion path of the chess "red elephant" to be shown in Fig. 17 (b)-(e). There is not obstacle on the motion path of the chess. We program the shortest path using A* searching algorithm.

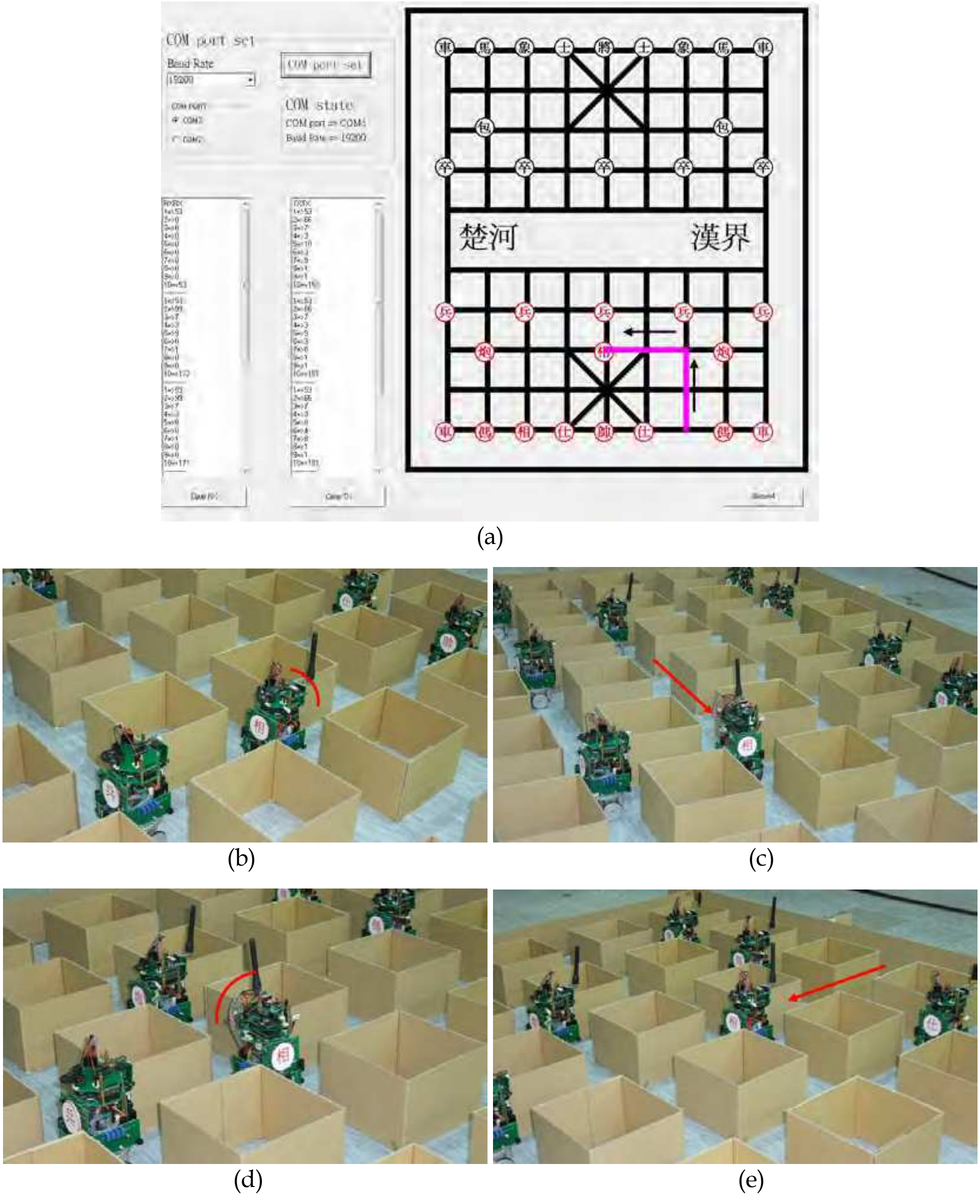


Fig. 17. The experimental results for the chess “red elephant”.

The second experimental scenario is “red cannon”. The user moves the chess “red cannon” using the mouse to be shown in Fig. 18 (a). The start position of the “red cannon” chess is (1,2), and the target point is (1,6). The mobile robot “red cannon” can’t moves forward to the target position (6,1). The motion path of the mobile robot has an obstacle (chess) at the position (1,5). The supervised computer reconstructs all possible paths using A* searching algorithm for red cannon step by step. Finally, we can find out the shortest path to avoid the obstacle (chess). The path can displays on the interface using red line. The supervised computer controls the mobile robot moving to the target point from the start point using the shortest path to avoid the obstacle. The supervised computer calculates the cost *f(n)* to be listed on the left side of the Fig. 18 (a).

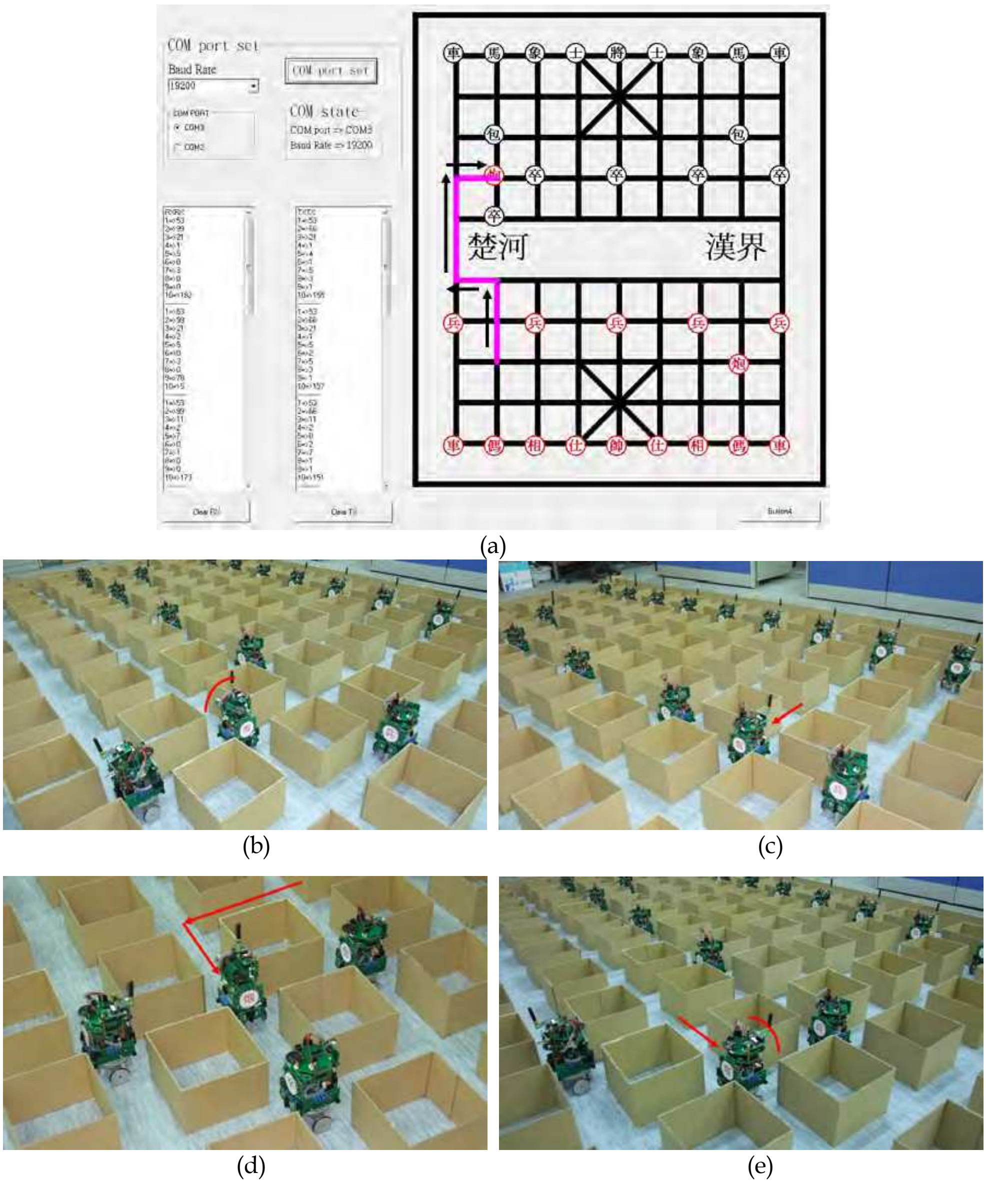


Fig. 18. The experimental results for the chess "red cannon".

The supervised computer orders the command to the mobile robot "red cannon". The chess piece moves forward two grids, and turn left angle to be 90^0. Then the mobile robot moves one grid to, and turn right angle 90^0. Then the mobile robot moves two grids, and turn right angle 90^0. Finally the mobile robot moves one grid to the target position (1,6), and turn left angle 90^0 to face the black side and stop. Finally, the experiment scenarios are shown in Fig. 18(b)-(e).

7. Conclusion

We have presented the experimental scenario of the Chinese chess game system using multiple mobile robots. The system contains a supervised computer, a image system, some wireless RF modules and thirty-two mobile robots. Mobile robots are classified red side and black side. We design two types' mobile robots (MBR-I and MBR-II) for Chinese chess game. The two mobile robots have the shape of cylinder and its diameter, height and weights is 8cm, 15cm and 1.5kg, and execute the chess attribute using two interfaces. One is wireless RF interface, and the other is voice interface. We develop the user interface on the supervised computer for the Chinese chess game. The supervised computer can programs the motion paths of the mobile robots using evaluation algorithm according to the rule of Chinese chess game, and receive the status of mobile robots via wireless RF interface. The MBR-I calculates the movement displacement using the pulse numbers from reflect IR sensors of the encoder module. Then we develop the MBR-II robot to implement scenario of the Chinese chess game on the grid based chessboard platform. The MBR-II robot calculates the movement displacement using encoder of DC servomotor. The chess (mobile robot) uses A* searching algorithm to program the shortest path moving to the target point. The supervised computer can controls mobile robots, and receives the status of mobile robots via wireless RF interface. Players can move the chess piece using the mouse on the supervised computer, and obey the game rules. The supervised computer can controls the mobile robot moving to the target position. The mobile robot speaks Chinese language for the movement status.

8. Acknowledgment

This work was supported by the project "Development of an education robot", under National Science Council of Taiwan, (NSC 99-2622-E-224-012-CC3).

9. References

S. J. Yen, J. C. Chen, T. N. Yang and S. C. Hsu, "Computer Chinese Chess," ICGA Journal, Vol.27, No. 1, pp.3-18, Mar, 2004.

P. Darwen and X. Yao, "Coevolution in Iterated Prisoner's Dilemma with Intermediate Levels of Cooperation: Application to Missile Defense," International Journal of Computational Intelligence Applications, Vol. 2, No. 1, pp.83-107, 2002.

C. H. Yong and R. Miikkulainen, "Cooperative Coevolution of Multi-agent Systems," University of Texas, Austin, USA, Tech. Rep. AI01-287,2001.

C. S. Ong, H. Y. Quek, K. C. Tan and A. Tay, "Discovering Chinese Chess Strategies Through Coevolutionary Approaches," IEEE Symposium on Computational Intelligent and Games, pp.360-367, 2007.

J. Wang, Y. H. Luo, D. N. Qiu and X. H. Xu, “Adaptive Genetic Algorithm’s Implement on Evaluation Function in Computer Chinese Chess,” Proceeding of ISCIT 2005, pp.1206-1209.

W. P. Lee, L. J. Liu and J. A. Chiou, “A Component-Based Framework to Rapidly Prototype Online Chess Game for Home Entertainment,” IEEE International Conference on System, Man and Cybernetics, pp.4011-4016, 2006.

W. Zhou, J. L. Zhang and Y. Z. Wang, “The Iterative Sort Search Techniques Based on Percentage Evaluation,” Chinese Control and Decision Conference (CCDC 2008), pp.5263-5266.

Kuo-Lan Su, Sheng-Ven Shiau, Jr-Hom Guo and Chih-Wei Shiau, “Mobile Robot Based Onlin Chinese Chess Game,” The Fourth International Conference on Innovative Computing, Information and Control, pp.63, 2009.

C. Buiu and N. Popescu, Aesthetic emotions in human-robot interaction implications on interaction design of robotic artists, International Journal of Innovative Computing, Information and Control, Vol.7, No. 3, pp.1097-1107, 2011.

T. Song, X. Yan, A Liang and K. Chen, A distributed bidirectional auction algorithm for multi-robot coordination, IEEE International Conference on Research Challenges in Computer Science, pp.145-148, 2009.

R. Grabowski and L. Navarro-Serment, C. Paredis, and P. Khosla, Heterogeneous teams of modular robots for mapping and exploration, Autonomous Robots, Vol.8, No. 3, pp.293-308, 2000.

E. J. Chung, Y. S. Kwon, J. T. Seo, J. J. Jeon, andH. Y. Lee, Development of a multiple mobile robotic system for team work, SICE-ICASE International Joint Conference, pp.4291-4296, 2006.

T. Balch, and R. Arkin, Behavior-based formation control for multirobot teams, IEEE Trans. on Robotics and Automation, Vol.14, No. 6, pp.926-939, 1998.

R. Alami, S. Fleury, M. Herrb, F. Ingrand, and F. Robert, Multi-robot cooperation in the MARTHA project, IEEE Robotics and Automation Magazine, Vol. 5, No. 1, pp.36-47, 1998.

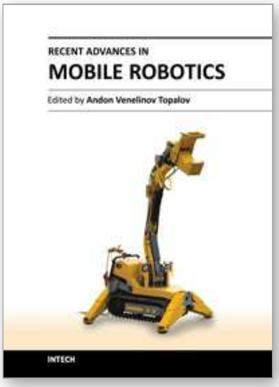
Y. Cao *et al*, Cooperative mobile robotics: antecedents and directions, Autonomous Robots, Vol.4, No. 1, pp.7-27, 1997.

J. H. Guo and K. L. Su, Ant system based multi-robot path planning, ICIC Express Letters, Part B: Applications, Vol. 2, No. 2, pp. 493-498, 2011.

W. Burgard, M. Moors *et al*, Coordinated Multi-Robot Exploration, IEEE Transaction on Robotics, Vol. 21, No. 3, pp.376-386, 2005.

K. L. Su, C. Y. Chung, Y. L. Liuao and J. H. Guo, A* searching algorithm based path planning of mobile robots, ICIC Express Letters, Part B: Applications, Vol. 2, No. 1, pp. 273-278, 2011.

Y. Saber and T. Senjyu (2007) Memory-bounded ant colony optimization with dynamic programming and A^* local search for generator planning, IEEE Trans. on Power System, Vol.22, No. 4, pp.1965-1973



Recent Advances in Mobile Robotics

Edited by Dr. Andon Topalov

ISBN 978-953-307-909-7

Hard cover, 452 pages

Publisher InTech

Published online 14, December, 2011

Published in print edition December, 2011

Mobile robots are the focus of a great deal of current research in robotics. Mobile robotics is a young, multidisciplinary field involving knowledge from many areas, including electrical, electronic and mechanical engineering, computer, cognitive and social sciences. Being engaged in the design of automated systems, it lies at the intersection of artificial intelligence, computational vision, and robotics. Thanks to the numerous researchers sharing their goals, visions and results within the community, mobile robotics is becoming a very rich and stimulating area. The book Recent Advances in Mobile Robotics addresses the topic by integrating contributions from many researchers around the globe. It emphasizes the computational methods of programming mobile robots, rather than the methods of constructing the hardware. Its content reflects different complementary aspects of theory and practice, which have recently taken place. We believe that it will serve as a valuable handbook to those who work in research and development of mobile robots.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Jr-Hung Guo, Kuo-Lan Su and Sheng-Ven Shiau (2011). Path Searching Algorithms of Multiple Robot System Applying in Chinese Chess Game, Recent Advances in Mobile Robotics, Dr. Andon Topalov (Ed.), ISBN: 978-953-307-909-7, InTech, Available from: http://www.intechopen.com/books/recent-advances-in-mobile-robotics/path-searching-algorithms-of-multiple-robot-system-applying-in-chinese-chess-game