Journal of Signal Processing, Vol.19, No.4, pp.139-142, July 2015

SELECTED PAPER AT NCSP'15

Design and Evaluation of Behavior Control Algorithm for Multiple-Mobile-Robot System in Panel Cruising Problem

Takuya Matsumoto and Takashi Yasuno

Graduate School of Advanced Technology and Science, Tokushima University
2-1 Minami-Josanjima, Tokushima 770-8506, Japan
Phone/FAX:+81-88-656-7458
E-mail: {matsumoto-t,yasuno}@ee.tokushima-u.ac.jp

Abstract

In this paper, we describe the design and evaluation of a behavior control algorithm for a multiple-mobile-robot system in a panel cruising problem. In order to achieve given tasks, fundamental functions such as path planning, path tracking, obstacle avoidance, and energy conservation are required for each mobile robot. Therefore, we propose an intelligent behavior control algorithm that was derived by analyzing the operator's manipulation data. In particular, we focus on the implementation of decision-making skills making it possible to maintain a balance between selfish and altruistic behaviors in a multiple-mobile-robot system, and confirm the usefulness of the proposed behavior control algorithm from several computer simulations.

1. Introduction

Recently, the technical advantages and the diversification of robots have become advanced. Therefore, the complexity of tasks required of robots has increased. However, the scale of a task that is possible to be accomplished by a single robot is limited. In addition, high cost, high technology and considerable time are required for manufacturing a robot that can deal with all tasks. Therefore, multiple-mobile-robot systems are of interest and are being studied actively[1, 2].

Until now, for the panel cruising problem using multiple mobile robots, each mobile robot must accomplish a given task while avoiding collision and interference with obstacles or other robots. We have proposed a method of acquiring the traffic control rules by genetic programming (GP)[3]. However, it is very difficult to define a suitable evaluation function for learning in GP. On the other hand, although the data mining algorithm C4.5 can generate a decision tree from the operator's manipulation data[4], the size of the generated traffic control rules, as the decision tree, becomes so huge that its implementation to real robots is difficult owing to the limited memory capacity.

From the above background, we take up a panel cruising problem and develop a virtual mobile robot control simulator (client-server-type network program) on a PC for the purpose of acquiring manipulation data by several operators at the same time. In addition, we derive the behavior control algorithm, expressed in a tree structure, on the basis of the acquired data and discuss the effectiveness of the proposed method referring to several experimental results.

2. Problem Statement

As shown in Fig. 1, a two-dimensional workspace with several static obstacles is virtually divided into 10 × 10 cm square panels. Each panel is assigned a value of 0, 1, 2, 4-32 or 64 points. The task of the mobile robots is to acquire as many points as possible without collision with another robot or the walls. In order to accomplish this task, each mobile robot can obtain environmental information on the local area enclosed by the red line in Fig. 2 as follows.

- Point assignment of local area
- Positions of static obstacles in local area
- Positions of other robots existing in local area

In addition, the robot can carry out forward, back, turn left, turn right, and stop actions. The traffic control method using shared point maps and the behavior control algorithm are treated as the panel cruising problem in this paper.

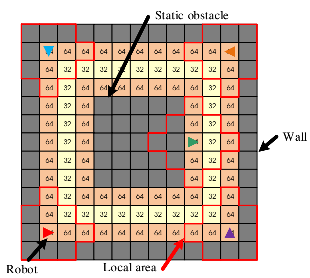


Figure 1: Workspace

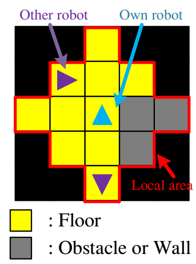


Figure 2: Local area

3. Data Acquisition System

Figure 3 shows the developed system for the collection of an operator's decision-making data. The developed system has a network configuration of the server-client type. The operator manipulates a robot (own robot) while looking at a virtual space expressed in 3D-CG. At this time, the operator's manipulation data is transmitted to the server via a network (TCP & UDP protocol). By adopting the server-client system, it becomes possible to obtain the interference among the intentions of operators and to collect a large amount of data in a short time. In addition, mistakes of robot manipulation can be reduced by using a game pad rather than a keyboard.

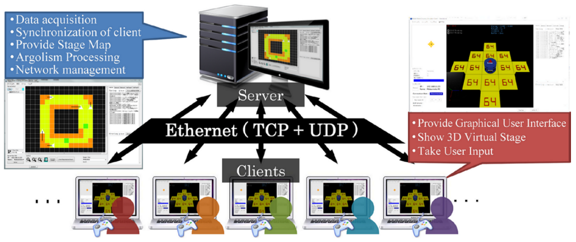


Figure 3: Configuration of developed data acquisition system

4. Behavior Control Algorithm of Mobile Robot

4.1 Analysis of operator's manipulation data

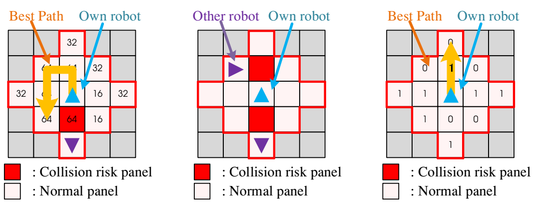
To derive a behavior control algorithm from the operator's manipulation data collected using our developed data acquisition system, we first focus on special cases where several mobile robots are close together in a small area. Secondly, we analyze the environmental conditions around the mobile robot and the selected action of each mobile robot in order to extract common characteristic features. As a result, behaviors of the mobile robot manipulated by the operator can be classified into the following three rules.

Rule I Obtain a higher number of points with top priority and by a smaller number of actions

Rule II Do not move to a panel with a high risk of collision

Rule III Expand the moving area when a higher-point panel does not exist around the mobile robot

Figures 4(a)-4(c) show examples of situations classified into the three rules mentioned above.



(a) Rule I (b) Rule II (c) Rule III

Figure 4: Classified behavior rules

4.2 Proposed behavior control algorithm

To implement the three rules of behavior into a real mobile robot, it is necessary to express them as computer algorithms. Moreover, the behavior control algorithm should be simple so that the implementation conditions satisfy the given hardware specifications. In our previous study, we confirmed that Dijkstra's method is useful for the obstacle avoidance problem of an unmanned ground vehicle. Dijkstra's method is one of the best-first search algorithms and is widely known as a route search algorithm applicable to mobile robots. In this paper, we propose a behavior control algorithm with a tree structure by extending the theory of the best-first search algorithm.

Figure 5 shows the concept of the proposed behavior control algorithm, which mainly consists of two parts. One is the process of making the tree structure that records the robot action patterns, and the other is the process of searching for the most suitable node from its tree structure.

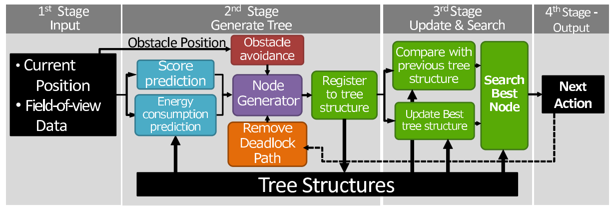


Figure 5: Concept of proposed behavior control algorithm

4.2.1 Generation method of tree structure

The proposed generation method of the tree structure consists of four stages, as shown in Fig. 5. The details of each stage are listed below.

Stage 1 Identify the point assignment and obstacles in the local area. Simultaneously, mark the high-risk panel of collision as a keep-out panel in the local area, as shown in Figs. 4(a) and 4(b).

Stage 2 Compute the number of points acquired and energy consumption for the cases of moving forward, back, left turn, and right turn. The number of points acquired is appended to the tree structure as a node value.

Stage 3 Calculate the travel distance and energy consumption. The travel distance means the length of the path from the center position in Fig. 6. The energy consumption values for each action are listed in Table 1.

Stage 4 Add the current node to the parent node. However, the node is excluded under the following conditions.

- Entry into the keep-out panel
- Entry into the same coordinates as the parent node
- Continuous turning action at the same panel
- Continuous moving action only between 2 panels

By repeating the above four stages one by one until all nodes become leaves, the tree structure is generated, as shown in Fig. 7.

4.2.2 Searching algorithm to obtain optimal node

An optimal node search from the generated tree structure is performed using the expanded best-first search algorithm. Here, the evaluation function $f(n)$ is given by

$$f\,(n) = \frac{1}{n} \sum\nolimits_{i=0}^{n} P_i \tag{1}$$

where n is the node depth and P_i is the number of points associated with the node. If multiple optimal nodes are found, the best optimal node is selected as the most energy-saving node. In addition, in order to implement the behavior rule shown in Fig. 4(c), the point acquisition situation of the past and present is compared.

If the number of acquired points decreases, the evaluation function $f(n)$ is changed to $g(n)$.

$$g(n) = \sum_{i=0}^{n} D_i \tag{2}$$

where D_i is the distance from the center position in Fig. 6. As a consequence, wide-area cruising can be performed. Furthermore, the size of the tree structure becomes an average of about 500 nodes. In most cases, it takes less than 100 ms to generate the tree structure. The required memory capacity for processing is small, so it is also possible to incorporate the generated tree structure into a microcontroller.

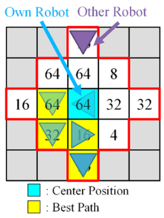


Figure 6: Example situation

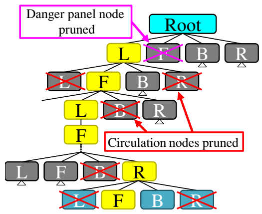


Figure 7: Planning tree

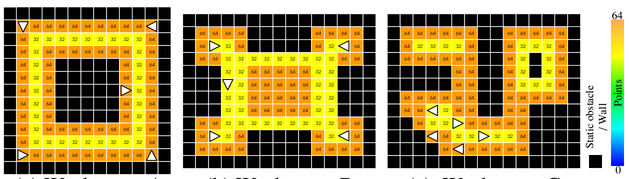
Table 1: Energy consumption values for each action

Action	Energy consumption
forward/back	10
rotate	8
stop	0

5. Experimental Results

To ensure the validity of the proposed intelligent behavior control algorithm, a comparative experiment with one operator's decision making was conducted in three kinds of workspaces, as shown in Figs. 8(a)-8(c). The experimental conditions are listed below.

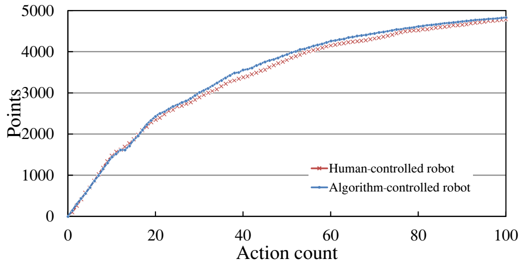
- The initial position and direction of virtual mobile robots are set to be the same as those of human-controlled robots (HUM-robots) and the robots controlled using the developed behavior control algorithm (ALG-robots).
- All virtual mobile robots move synchronously.
- The number of mobile robots is set to five.
- The total number of actions is limited to 100.
- Fifteen subjects act as operators. All subjects carried out given missions two times (100 actions for one mission) to confirm the basic manipulation of the virtual robot before the experiments.



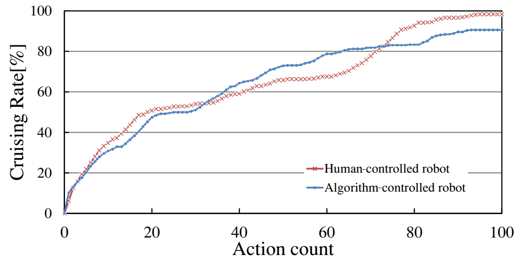
(a) Workspace A (b) Workspace B (c) Workspace C

Figure 8: Simulated workspaces

Figures 9 to 11 show the transitions of the number of acquired points and the cruising rate (CR) for the HUM-robots and the ALG-robots. Here CR means the ratio of non occupied panels without static obstacles in a local area that when updated after every action become occupied to the total number of panels in the workspace. In the case of simple workspace A, the transition of the CR for the HUM-robots and the ALG-robots show approximately the same characteristics. On the other hand, in the complex workspaces B and C, different characteristics are confirmed from Figs. 9(a) and 9(b). The CR of the ALG-robots increased more slowly, whereas, there is no decisive difference between the number of points acquired by the HUM-robots and the ALG-robots.

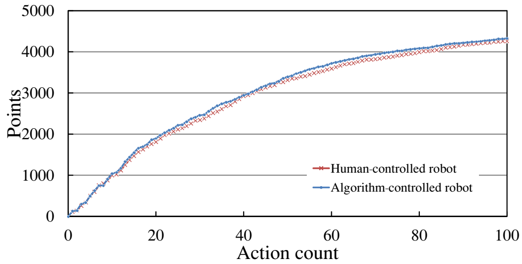


(a) Transition of number of points in workspace A

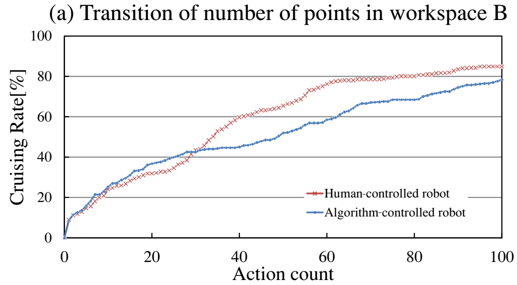


(b) Transition of cruising rate in workspace A

Figure 9: Results in the case of workspace A

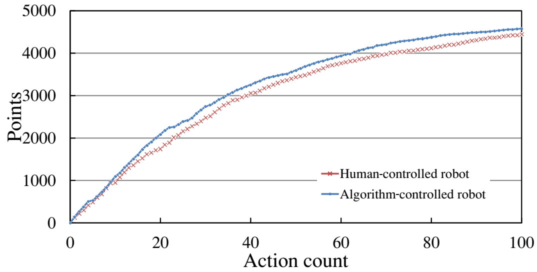


(a) Transition of number of points in workspace B

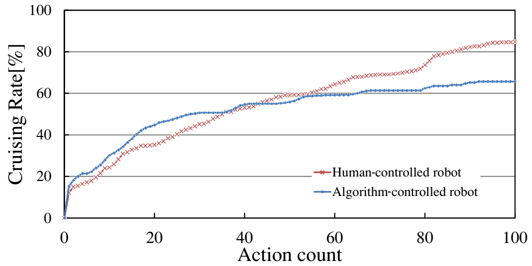


(b) Transition of cruising rate in workspace B

Figure 10: Results in the case of workspace B



(a) Transition of number of points in workspace C



(b) Transition of cruising rate in workspace C

Figure 11: Results in the case of workspace C

Next, to evaluate the task performance of the mobile robot, we define the point taken ratio (PTR), which is calculated by

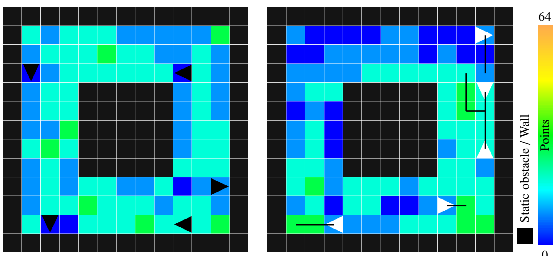
$$\mathrm{PTR} = \frac{\sum_{i=1}^{N} \delta_i d_i}{\sum_{i=1}^{N} \delta_i k_i} \times 100 \begin{cases} \text{obstacle panel} : \delta_i = 1 \\ \text{not obstacle panel} : \delta_i = 0 \end{cases} \quad (3)$$

where N is the workspace size, k_i is the number of panel points before the experiment and d_i is the number of panel points after the experiment. From Table 2, we can see that the ALG-robot's PTR is higher than that of the HUM-robots. Therefore, the proposed behavior control algorithm has an excellent task performance ability compared with human decision-making skills.

Finally, Figs. 12 to 14 show the point assignment situation in the workspace after 100 actions. We can easily confirm that the mobile robots controlled by the proposed behavior control algorithm can achieve the given task successfully.

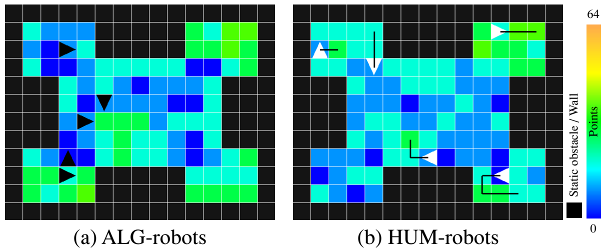
Table 2: Result of PTR for each workspace

Workspace	A	B	C
ALG-robot's PTR [%]	94.2	93.0	94.2
HUM-robot's PTR [%]	93.4	91.2	91.3



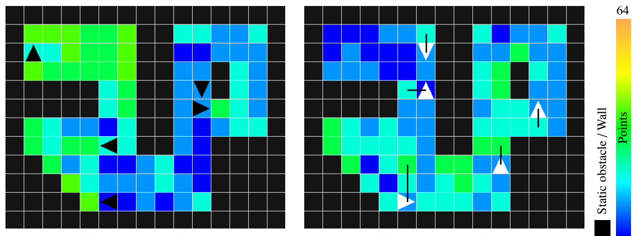
(a) ALG-robots (b) HUM-robots

Figure 12: Point collection results in workspace A



(a) ALG-robots (b) HUM-robots

Figure 13: Point collection results in workspace B



(a) ALG-robots (b) HUM-robots

Figure 14: Point collection results in workspace C

6. Conclusions

In this work, we proposed a design method of an intelligent behavior control algorithm for a multiple-mobile-robot system based on human decision-making skills. From the experimental results, it was confirmed that the point collection performance of the algorithm is higher that of human decision-making skills.

Future works are planned to confirm the repeatability of the proposed behavior control algorithm through experiments using the developed real multiple-mobile-robot system, and to implement the behavior control rules considering the amounts of remaining energy.

References

[1] H. Suzuki, T. Yasuno, S. Urushihara and H. Harada: Design and implementation of cooperative conveyance patterns for multiple mobile robots using neural network, Journal of Signal Processing, Vol. 11, No. 6, pp. 489-495, 2007.

[2] M. Morimoto, T. Yasuno, E. Yasuno, H. Suzuki and H. Harada: Traffic control of multiple sweeping robots in panel cruising problem, Journal of Signal Processing, Vol. 12, No. 4, pp. 331-334, 2008.

[3] T. Kagawa, T. Yasuno, S. Urushihara, E. Yasuno and A. Kuwahara: Traffic control of multiple sweeping robots using genetic programming, RISP International Workshop on NCSP, pp. 415-418, Mar. 2009.

[4] K. Koide, T. Yasuno, T. Kagawa, E. Yasuno and A. Kuwahara: Acquisition method of human decision-making skills for panel cruising problem with multiple robots, RISP International Workshop on NCSP, pp. 215-218, Mar. 2010.