UNARM System to Decide Units Locations of Cell-type Assembly Machines with Robots Arms

Hirotaka Moribe
Department of Human Information Systems, Gifu University, Yanagido 1-1,
Gifu, Gifu 501-1194, JAPAN

Hidehiko Yamamoto
Department of Mechanical Engineering, Gifu University, Yanagido 1-1
Gifu, Gifu 501-1194, JAPAN

Takayoshi Yamada
Department of Mechanical Engineering, Gifu University, Yanagido 1-1
Gifu, Gifu 501-1194, JAPAN

E-mail: u3128029@edu.gifu-u.ac.jp, yam-h@gifu-u.ac.jp, yamat@gifu-u.ac.jp

Abstract

This study develops the system called UNARM which assists the unit placement decision of the automatic assembling equipments to assemble efficiently. The unit is a part supply or a robot hand constituting assembling equipment. It is the system which decides these units placement where of the assembling equipment you place the interference of the both arms of the robot using reinforcement learning in spite of being a guard.

Keywords: Reinforcement Learning, Unit placement decision, Double arm robot, Assembly machine

1. Introduction

In recent years, automations of the assembly works with robots have been performed in the production area. As for the automatic assembly machine with robots, the production efficiency changes by the placement of each unit. However, because the placement of units is determined by the experience of engineers in many factories, it is questionable whether the determined placement is good or not. To solve the problem, we develop the system, Units-layout Nomination for Assembly with Robot Mechanism (UNARM), to assist the unit placement decision of the assembly machine with robots. In addition, we apply UNARM to a cell type assembly machine with robot both arms and ascertain the usefulness of UNARM.

2. UNARM

UNARM is the system to decide each unit placement of the assembly machine using reinforcement learning. For the placement place of each unit chosen by every learning, UNARM calculates the assembly time and improves the unit placement using machine learning and the calculate time.

2.1. *Structure of UNARM*

UNARM consists of two modules, (ⅰ)the conditions making module and (ⅱ)the learning module as shown in Fig.1.

After the condition making module reads the work data and the possible area of unit placements, the module gets ready to learn in a learning module based on it. The work data means the time between chuck and unchuck, the coordinates that both arms move and the work sequence. The learning module uses reinforcement learning based on values and repeats the four processes, ①to choose the tentative placements, ②to calculate the working time, ③to evaluate the acquired cycle time, and ④ to repeat the four reward distribution operations. Finally, the most suitable placement of the units can be decided.

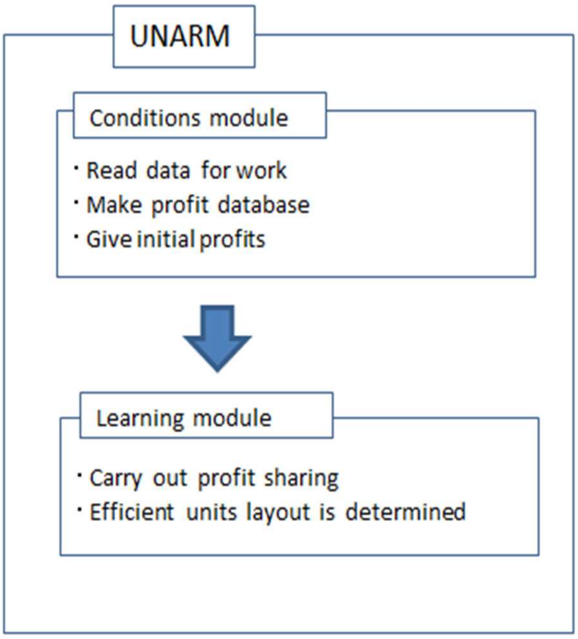

Fig. 1. System construction of UNARM.

2.2. *Condition making module*

The condition making module carries out the following processes.

Step1. Read each work data of the robot two arms.

Step2. Read the possible area of each unit placement.

As shown in Fig.2, the possible area of each unit placement is divided as a lattice and the center of each lattice becomes the center of the candidate place.

Step3. Give initial values to the candidate place created in Step2.

2.3. *Learning module*

The learning module carries out the following processes.

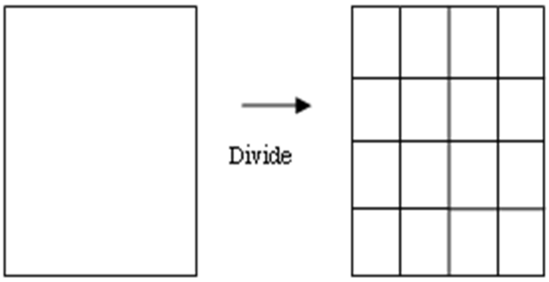

Fig. 2. Divided placements.

10	10	10	10
10	10	10	10
10	10	10	10
10	10	10	10

Fig. 3. Distributions of values.

Step1. Make an initial values of the values.

Step2. Decide the placement of the unit by a roulette selection.

Step3. Calculate the coordinates of each part.

Step4. Calculate the moving distances of the robot arms.

Step5. Calculate the working time of the robot arms.

Step6. Compare the calculated working times with the standard value.

Step7. If the time is smaller, give the reward.

Step8. If the working time is smaller, update the standard value.

Step9. Repeats Step2 from Step8 till a decided reputation times.

We adopt the concept of waiting time not to produce arm interferences because the two arms move at the same time in Step5. In step6, the concept chooses the longer working time as the next evaluation time.

3. Moving method of robot arms

To calculate the working time of the robot arms in Step5, the robot arm moving speed is necessary to calculate.

Therefore we define the arm moving locations as follows, Apprpach-1: the upper point of the part that a robot will take, Approach-2: the upper point closer to Approach-1 and Chuck point: the point that the arms chuck, unchuck or put a part. We also define the arm moving speed among the upper three points. The speed between Approach-2 and Approach-2 is defined a feet speed. The speed between Approach-2 and Approach-1 as approach speed I, the speed between Approach-1 and chuck point as approach speed II. In addition, as these three kinds of speed have different speeds whether the arm grasp a part or not, the six kinds of speed value definitions in total must be chosen. In this study, we adopt that the robot arms always move by regular linear motion.

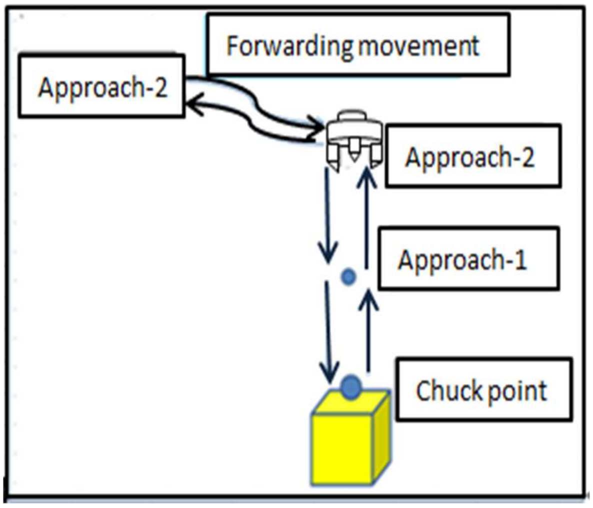

Fig. 4. Robot arm movements.

4. Concept of waiting time

The robot of this study has both arms. When we use the both arms, the most difficult point is to interfer both arms each other. In other words, it is necessary to decide the work sequence not to make the interference of the right arm and the left. Therefore we adopt the concept, the waiting time, to prevent arm interferences in calculating working times of one cycle in Step5. in the learning module. For example, the assembly sequence exists that the assembly work A must be set after the completion of the assembly work X. In this case, when one of the two arms performs assembly work A, the other arm must confirm the completion of assembly work X. In other words, the assembly work A must start after the completion of assembly work X. We define this time lag as the waiting time.

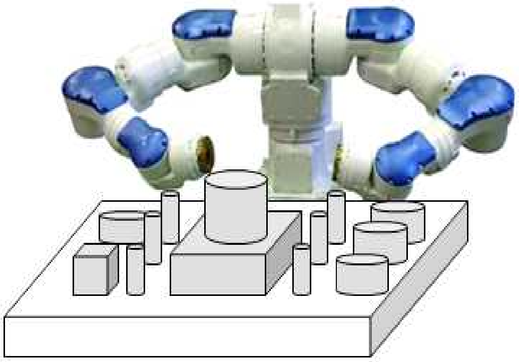

Fig. 5. Both arms of robot.

The following shows the waiting time.We define the two flags, Lp and P.

Lp(Look at Point): Job that starts to need the waiting time.
P(Point): Work corresponding to the waiting time of Lp.

In order not to make arm interferences, the works that need the waiting time are given Lp and the corresponding works are given P.

For example, as shown in Fig.6, when the left arm starts Lp, the left arm can continue the work as the right arm still finished P. This is because waiting time is not given to the left arm.

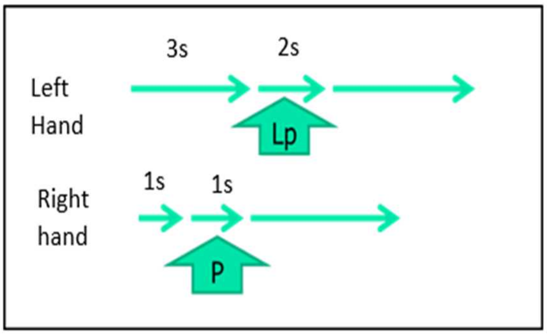

Fig. 6. No waiting time.

On the contrary, in the case of Fig.7, when the left arm starts Lp, the work of P of the right arm is not finished. This is because the left hand has to wait seven seconds to start Lp. The seven seconds correspond to waiting time.

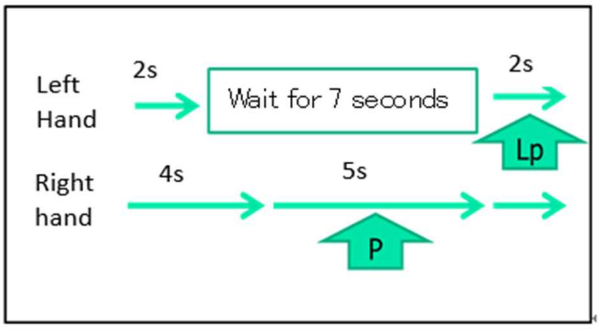

Fig. 7. Waiting time.

5. Simulation applications

We applied UNARM to the cell type assembly machine with the both arms of the robot to ascertain the usefulness of UNARM. The application example adopted eight units (two of the eight are trays that many parts were put), 10 initial values, 10 reward levels, 600,000 times learning number, 36 left arm assembly processes and 40 right arm assembly processes. Table 1 shows the moving speed of the robot arms. In addition, in order to compare the learning results, the simulations randomly to choose unit placements were also carried out.

Table 1. Moving patterns of arms.

	Without part	With part
Approach-2—Approach-2	770mm/s	380mm/s
Approach-2—Approach-1	400mm/s	250mm/s
Approach1—Chuck point	25mm/s	50mm/s

5.1. *Simulation results*

We carried out ten times simulations for each condition and showed the mean and the minimum of two kinds simulations in Table2.

Table 2. Simulation results.

	UNARM(s)	Random(s)
Average time	237.4228	243.9498
The smallest time	237.2925	241.9623

Acquired placements of the minimum evaluation time are shown in Fig.8. The placements acquired by random selections are shown in Fig.9.

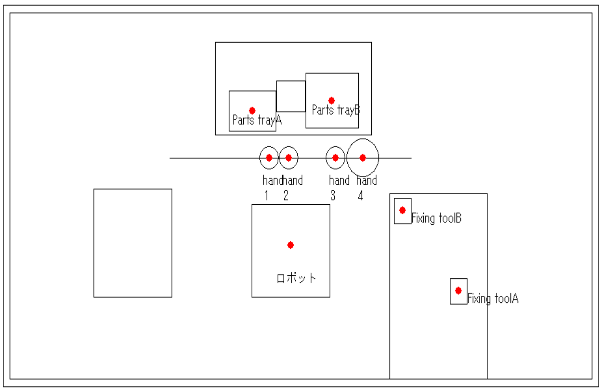

Fig. 8. Placements of the minimum.

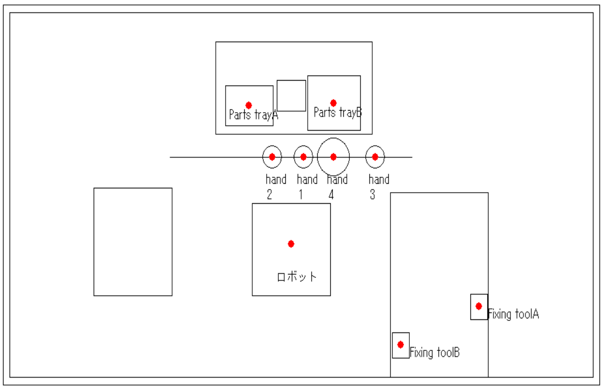

Fig. 9. Placements by a random method.

Comparing the results of Table2, UNARM's evaluation times are better than the random ones. Especially, the mean evaluation time is shorter six seconds. It is said that UNARM can continuously acquire the better assembly cycle time corresponding to the efficient units placement.

6. Conclusions

In this study, we developed UNARM system which finds the better unit placements and prevents the interferences of both arms. To do this, UNARM adopted the concept of waiting time. By applying UNARM to the cell-type assembly machine that includes the both arms robot, UNARM acquired the good unit placements. It is ascertained that UNARM is useful.