

Intelligent robot of inclined assembly sequence planning in Industrial 4.0

Yu Cheng Chiang¹, Chumpol Yuangyai², and Chen yang Cheng^{1*}

¹Industrial Engineering Department, National Taipei University of Technology

²Department of Industrial Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand

*cycheng@ntut.edu.tw

Abstract. In the industry 4.0, the Cyber-physical system (CPS) is one of the most important core which makes the manufacturing process more intelligent. Intelligent assembly operation is an important key in intelligent manufacturing of CPS. To complete the intelligent assembly operation, the cooperation between assembly robotic arm and assembly sequence planning (ASP) is necessary. However, the ASP and writing robotic codes manually is time consuming and requires professional knowledge and experience. Because the Local Coordinate System (LCS) is often ignored when checking for interference. If product have inclined interference and without considering LCS and causing and infeasible ASP. Therefore, this paper proposes a LCCPIAS (Local Coordinate Cyber-Physical Intelligent Assembly System) system to achieve three objective functions. First, this paper presents a dual-projected-based interference analysis approach (DPIAA) that analyzes the relations between components. Second, this paper generates optimal assembly sequence automatically to let the assembly sequence more suitable for the robotic arm to perform the assembly operation. The last one is LCS can recognize inclined interference between components and generate feasible ASP. Furthermore, this paper uses CAD model to verify that the DPIAA is faster and consider LCS interference can solve inclined interference problem. In the future assembly factory, the proposed method can help to realize intelligent manufacturing.

1 Introduction

In the manufacturing process, assembly work consume a lot of cost and time of the whole production. The product assembly cost accounts for more than 40% of the total cost in the whole product manufacturing process [2], and the assembly time of the product accounts for the total manufacturing time 20-70%, so intelligent assembly operation is an important key to the CPS intelligent manufacturing. In order to complete the smart assembly, the cooperation between the robot and the ASP is necessary.

If the best assembly sequence of products is automatically generated, it will not only reduce

manufacturing time and costs but also customers' rapidly changing needs can be met immediately [4, 8]. In order to generate the best assembly sequence, several heuristic algorithms are proposed to solve the problem. The three most important ones are Genetic Algorithm (GA) [1], Particle Swarm Optimization (PSO) [3] and ACO [9]. However, the input data of these algorithms such as the contact and interference between parts must be input manually. In addition if the assembly product has inclined assembly and without considering the LCS coordinates of the part itself only considering GCS, it will may generate wrong ASPs and lead to assembly process errors. Therefore some methods to automatically obtain the

part information through CAD are proposed [5-7] .

Therefore, this paper purpose LCCPIAS to solve problem. Firstly, DPIAA method is proposed in this paper. The relationship between the parts in the product is analyzed. According to the output of DPIAA, the ACO algorithm is used to automatically generate Good assembly sequence, so that the assembly sequence is more suitable for assembly of the robot arm operation.

2 Literature discussion

There are probably many assembly sequences for products made up of dozens of parts, so how to find out the best or even the best assembly sequence has been discussed for a long time. There should be no possible collision of assembly sequences, so it is necessary to analyze the interference between the parts. Interference analysis is a common method of calculating the assembly direction of parts[5]. The final pattern of assembly sequence is the optimal assembly sequence, and although the possible assembly sequence allows the assembly operation to be performed without any interference, some assembly sequences still have many repetitive motions during the assembly process and many literature studies Find out the best assembly sequence by different algorithms and different objective functions [1, 9]

3 Research methods

In this paper, the LCCPIAS method was proposed to let the product designer design the product in the CAD software. The designer provided the CAD file to the service provider. The service provider imported the CAD file into the LCCPIAS then it will output assembly sequence automatically to robot.

The method is divided into two parts as shown in Figure 1. Firstly, DPIAA method is used to analyze the contact and interference between parts. After the

analysis is completed, the ACO algorithm is used to generate the best assembly sequence. Because ACO uses the assembly robot’s objective function calculates each feasible assembly sequence, so assembly can be done without human intervention.

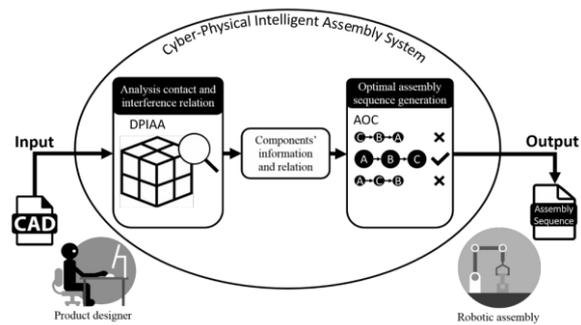


Figure 1 LCCPIAS Method flow chart

3.1 Dual Projection-Based Interferometric Analysis Method: IM is generated by collision detection. Using the proposed DPIAA and combining AABB can reduce the collision detection time.

- Symbol definition:

d_k : Cartesian axis in the direction of k ◦

e_s : Parts “s” to be assembled ◦

n : The total number of parts

CM_{st} : The contact matrix representing the contact relationship between e_s and e_t

DCM_{st}^k : The contact matrix representing the contact relationship between e_s and e_t in the direction d_k

PM_{st}^k : Projection matrix representing the projection of e_s and e_t in the d_k direction.

IM_{st}^k : The interference matrix representing the interference between e_s and e_t in the d_k direction.

LCS_{st}^k : The interference matrix representing the interference between e_s and e_t in the itself direction.

- DPIAA Steps

Step 1: Interfere detection and generation of CM and DCM

In the contact matrix (CM), CM_{st} denotes the contact between e_s and e_t . CM has two kinds of relations: if there is no contact between e_s and e_t , CM_{st} value will be 0, otherwise 1 .Directional

Contact Matrix (DCM), where DCM_{st}^k indicates the contact between e_s and e_t in a certain direction.

There are two types of relationship in DCM. If e_s and e_t have no contact relationship in the d_k direction, DCM_{st}^k value will be 0, otherwise 1.

Step 2 Double projection inspection and generation of PM

Projection Matrix (PM) is generated using dual projection detection. PM_{st}^k represent projection relationship between e_s and e_t . PM used to represent e_s and e_t AABB projection onto the plane is overlapped.

The second projection is the projection of AABB to the moving direction d_k axis. As shown in Figure 2(a), if any part e_u overlaps parts e_s and e_t , e_u will be between two parts (e_s , e_t). It means e_s and e_t is not collision in d_k axis. The projection matrix is modified to 0, otherwise 2,

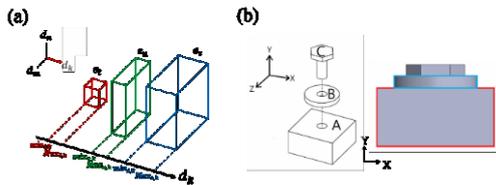


Figure 2 Project AABB to the direction of movement

Step 3 Collision detect

Collision detection is the last method used to determine the collision between parts. If the double projection determines that there is a collision between the parts e_s and e_t , the DPIAA will perform collision detection to verify that the parts e_s and e_t interfere in the direction d_k . If there is interference between e_s and e_t , it will modify the interference matrix value to 1.

In order to generate LCS, collision detection must be performed on all parts of the LCS itself. Considering the order of collision, if e_s and e_t interfere with each other in the LCS direction, If there is any interference with other parts, if interference occurs, test the collision of multiple parts by one part. If there is interference, modify the value of 1 in

LCS_{st}^k .

Step 4 Generate interference matrix

After the DPIAA process is completed, the interference matrix is automatically generated.

3.2 Ant colony algorithm to solve ASP

By simulating the behavior of ants and using the pheromone message to find the best path, the ACO proposed an ACO algorithm for finding the best assembly order. The target of the ACO algorithm is minimum rotation angle and the highest stability. The input assembly required for ACO is CM, DCM, and IM from DPIAA.

Step 1 Initialize the basic data

The first step in this paper is to initialize the basic ACO data. The information to be initialized includes control parameters (such as rotation angle, stability, weight of pheromones, total number of ants and number of iterations), and assembly data (such as the total number of parts n, CM, DCM and IM), the total number of ants m is set by the number of can be disassembled parts.

Step 2 Initialize ant data

Each ant is assigned to a set of IM, CM, and DCM that are automatically generated from DPIAA.

Step3 Find the first moveable part

In each iteration, each ant starts browsing to find out the possible assembly order. Every ant can get all the parts that can be removed by IM. At the beginning of the first iteration, all the movable parts must be located and assigned to different ants, and all the movable parts can be selected according to equation(1).

$$Ant_k^{start} = P_{feasible} \tag{1}$$

Ant_k^{start} represent the ant first choice of parts, ants select the first part of the probability is based on equation(2).

$$P_k^i(0) = \sum_{j=0}^{n-1} \tau_{ij} / \sum_{s \in allowed_k} \sum_{j=0}^{n-1} \tau_{sj} \tag{2}$$

Step 4. Look for detachable parts

After selecting the first detachable part, the ant selects the next part according to equation (3).

$$P_k^i(t) = \tau_{ij}(t)^\alpha * \eta_{ij}^\beta / \sum_{s \in allowed_k} \tau_{is}(t)^\alpha * \eta_{is}^\beta \quad (3)$$

$P_k^i(t)$ represents the probability of ant k from i to j at time t, α and β are the weights and heuristic functions of pheromone, and the heuristic function η_{ij} is composed of the following two criteria:

1. Rotation angle : The rotation angle indicates the combined direction difference between the assembly directions of the parts i and j as equation (4),

$$\begin{cases} 0.1, d_i - d_j = 180^\circ \\ 1, d_i - d_j = 0^\circ \\ 0.5, other\ angle \end{cases} \quad (4)$$

2. Stability : Stability indicates that the part does not fall or displace during disassembly. Heuristic functions for stability (S) are shown in equations(5), (6) and (7)

$$S_{ij} = S_j^1 + S_{ij}^2 (j = 1, \dots, n - 1)(i = j - 1) \quad (5)$$

$$\begin{cases} \frac{CCD_{max} - CCD_j}{CCD_{max}} + \frac{0.1}{DCCD_{G,max}} \frac{DCCD_{G,max} - DCCD_{G,j}}{DCCD_{G,max}}, \begin{cases} \sum_{k=0}^n IM_{j,k}^{+y} \\ \sum_{k=0}^n IM_{j,k}^{-y} \end{cases} \\ other \end{cases} \quad (6)$$

$$S_{ij}^2 = \begin{cases} 0.5, DCM_{ij}^{-y} = 0 \\ 0, other \end{cases} \quad (7)$$

In this paper, we modify the definition of the direction of gravity and increase the stability of the previous component. The multi-objective heuristic function is given by equations(8). S_{ij} represents the stability between i and j, and w_1 and w_2 represent the weight of each target.

$$\eta_{ij} = w_1 * D_{ij} + w_2 * S_{ij} \quad (8)$$

Step5 Pseudo-random scaling rules

After calculating the probability of all moving parts, the ants in the ACO use the pseudorandom scaling rule of equation (9) to select the parts from i to j

$$s = \begin{cases} arg \max_{k \in allowed_k} \tau_{ij}(t)^\alpha * \eta_{ik}^\beta, \\ S \\ if\ q \leq Q \\ other \end{cases} \quad (9)$$

CM, DCM, and IM are updated from the data of each ant, and if there are still parts that have not been selected yet, it will go back to Step 4. Otherwise, when all the ants have finished searching in the current iteration, go to Step 6; otherwise, go to Step 3.

Step 6 Update Pheromone matrix

After all the ants finishes searching, the pheromone matrix is updated with the global update rule given by equation(10) and (11).

$$\tau_{ij}(t + 1) = (1 - p)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (10)$$

$$\Delta\tau_{ij}^k(t) = \begin{cases} (F_{gb})^{-1} \begin{cases} Mod(t,5) = 0 \ \&(i,j) \in Global\ best\ solution \\ (F_{ib})^{-1}, as \ \begin{cases} Mod(t,5) \neq 0 \ \&(i,j) \in Iterated\ optimal\ solution \\ other \end{cases} \end{cases} \\ 0 \end{cases} \quad (11)$$

References

1. Gunji, B., et al., *Hybridized genetic-immune based strategy to obtain optimal feasible assembly sequences*. International Journal of Industrial Engineering Computations, 2017. **8**(3): p. 333-346.
2. Hongbo, S., et al., *Genetic simulated annealing algorithm-based assembly sequence planning*. 2006.
3. Li, M., et al., *A hybrid assembly sequence planning approach based on discrete particle swarm optimization and evolutionary direction operation*. The International Journal of Advanced Manufacturing Technology, 2013. **68**(1-4): p. 617-630.
4. Li, X., et al., *Assembly sequence planning based on an improved harmony search algorithm*. The International Journal of Advanced Manufacturing Technology, 2016. **84**(9-12): p. 2367-2380.
5. Ou, L.-M. and X. Xu, *Relationship matrix based automatic assembly sequence generation from a CAD model*. Computer-Aided Design, 2013. **45**(7): p. 1053-1067.
6. Pan, C., *Integrating CAD files and automatic assembly sequence planning*. 2005.
7. Pintzos, G., et al., *Assembly precedence diagram generation through assembly tiers determination*. International Journal of Computer Integrated Manufacturing, 2016. **29**(10): p. 1045-1057.
8. Yin, Z., et al., *A connector-based hierarchical approach to assembly sequence planning for mechanical assemblies*. Computer-Aided Design, 2003. **35**(1): p. 37-56.
9. Yu, J. and C. Wang, *A max-min ant colony system for assembly sequence planning*. The International Journal of Advanced Manufacturing Technology, 2013. **67**(9-12): p. 2819-2835.