

Motion-Data Driven Grasp/Assembly Planner

Kensuke Harada¹², Natsuki Yamanobe², Weiwei Wan¹²
Kazuyuki Nagata², Ixchel G. Ramirez-Alpizar¹, and Tokuo Tsuji³

¹ Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka 560-8531, Japan

² Intelligent Systems Research Institute
National Institute of Advanced Industrial Science and Technology
1-1-1 Umezono, Tsukuba 305-8568, Japan

³ Institute of Science and Technology, Kanazawa University
Kadoma-cho, Kanazawa 920-1192, Japan

E-mail: harada@sys.es.osaka-u.ac.jp, n-yamanobe@aist.go.jp,
wan@sys.es.osaka-u.ac.jp, k-nagata@aist.go.jp,
ramirez@sys.es.osaka-u.ac.jp, tokuo-tsuji@se.kanazawa-u.ac.jp

Abstract : This paper proposes a robotic grasp/assembly planner using a database of complex and skillfull motion. In our proposed planner, if a skillfull motion is included in the solution path, motion data included in the database is used. In this paper, we first explain the three-layered structure of the motion data. Then, we explain the proposed grasp/assembly planner. Finally, we show a numerical example to confirm the effectiveness of our proposed method.

1. Introduction

In factory environment, industrial robots are expected to assemble a product. However, robotic assembly is difficult since a robot has to assemble a variety of objects. Sometimes, a robot has to realize a very skillfull and complex motion.

On the other hand, this research aims to fully automate the product assembly process and proposes a novel robotic grasp/assembly planner. In conventional grasp/assembly planners, we simultaneously planned the grasping posture of a part, the assembly sequence and the motion of a robot. However, conventional grasp/assembly planners can just realize a series of pick-and-place motion of a robot. For the purpose of realizing more complex and skillfull motion by using a grasp/assembly planner, we consider incorporating the database of skillfull motion. The motion data can be generated by using several methods such as human motion capture and robotic motion generation methods. Especially, our motion generation tool utilizes the previously proposed three-layered structure of motion data. In our proposed grasp/assembly planner, if the solution path includes complex and skillfull motion, such motion is realized by using the database of skillfull motion.

This paper briefly outlines our novel grasp/assembly planner using the motion data and shows a simple numerical example. The rest of this paper is organized as follows; we first explain the three-layered data structure in Section 2. The robotic grasp/assembly planner is explained in Section 3. Application of motion data to the grasp/assembly planner is explained in Section 4. A numerical example is shown in Section 5.

2. Three-Layered Data Structure

In this section, we explain the structure of the motion data and method for storing skillfull motion to the database [1]. An assembly task is composed of multiple task elements where each task element is composed of a set of motion sequence. According to this classification, the first, the second and the third layers are respectively named as "work", "task" and "action" layers. Fig. 2. shows an example of assembly of a toy airplane. Since a single action usually includes multiple objects to be assembled, the object held by a robot is named as the "main object" and the object assembled to the main object is named as the "sub object". The following is a set of information included in the database:

- Motion strategy (A sequence of task/motion)
- Human (Identified name etc.)
- Robot (Name, type, hand)
- Object Name
- Object category (e.g., JICFS code)
- Object shape
- Object parameters (Mass, CoG, material, stiffness, etc.)
- Grasping configuration (Main object)
- Fixture configuration (Sub object)
- Time series of the main object's pose w.r.t the sub object
- Control method and its parameters
- Force information applied by the hand

The data can be obtained by several methods such as a human motion capture and a motion generation tool. The data is stored in the YAML format where some examples are shown in Fig. 2. On the other hand, we have constructed a software tool manually constructing a skillfull motion and store it to the database [1]. Among the three-layered structure of database, our proposed grasp/assembly planner especially uses the action data corresponding to the edge of the AND/OR graph of assembly which will be explained in the next section.

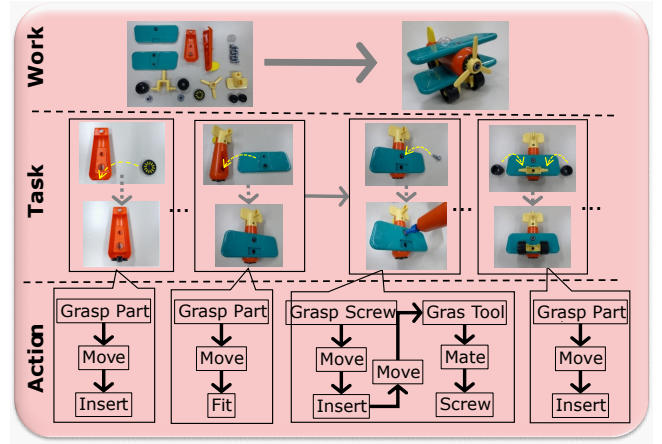
3. Grasp/Assembly Planner Definition

This section shows the overview of the grasp/assembly planner which has proposed in [2, 3]. Let us consider a product composed of m parts $P = (P_1, \dots, P_m)$. Let us also consider assembly of parts by $\mathcal{C} = (\mathcal{C}_1, \dots, \mathcal{C}_r)$. For example, the assembly \mathcal{C}_l composed of the parts P_u, P_v , and P_w can be defined as

$$\mathcal{C}_l = \{(P_u, P_v, P_w), (\mathbf{T}_{uv}, \mathbf{T}_{uw}), (\mathbf{n}_{uv}, \mathbf{n}_{uw})\} \quad (1)$$

where \mathbf{T}_{uv} and \mathbf{n}_{uv} denote the homogenous matrix and the approach vector, respectively, of P_v relative to P_u used for an assembly task.

To assemble a product, a gripper has to grasp an assembly of parts. We consider a set of n grippers $H = (H_1, \dots, H_n)$ where H_j denote the j -th gripper. We consider using a grasp planner to calculate a grasping posture where G_{ljk} denotes the k -th grasping posture of the assembly \mathcal{C}_l by the gripper H_j . By iteratively calculating grasping posture with changing the approach



Yaml Format

```

type: work
name: "AirplaneAssembly"
objective: "Assemble a toy airplane"
objects:
- toyAirplane.yaml
- driver.yaml
sequence:
- insertPropellerBaseToBody.yaml
- assembleAirPlaneBodyParts.yaml
...

type: task
name: "insertPropellerBaseToBody"
objective: "Insert a propeller base to a body part"
objects:
- propellerBase.yaml
- upperBody.yaml
sequence:
- grasp***.yaml
- move***.yaml
- insert###.yaml

type: action
name: "insert###"
objective: "Insert a propeller base to a body part"
operator:
type: robot
info: nextage.yaml
objects:
manipulatedObject : propellerBase.yaml
environmentObject : upperBody.yaml
sequence:
-
time: 0
pose: [x, y, z, r00, r01, r02, r10, r11, r12, r20, r21, r22]
manipulatedObject :
prehesion :
type: prehension
frame: [x, y, z, r00, r01, r02, r10, r11, r12, r20, r21, r22]
graspPoints :
- [x, y, z]
- [x, y, z]
- [x, y, z]
environmentObject :
prehesion :
type: fix
...

```

Fig. 1: Three Layered Data Structure and Data Expression using YAML format

position, we consider constructing a set of grasping postures $G_{lj} = (G_{lj1}, \dots, G_{lj_k})$ of the assembly \mathcal{C}_l grasped by the gripper H_j .

Every possible assembly sequence can be expressed by using the AND/OR graph $G(V, E)$ where the node V is composed of the assembly of parts \mathcal{C}_l $l = 1, \dots, r$ (Fig. 3.). In the grasp/assembly planner introduced in this section, we assume that the AND/OR graph is defined in advance. To obtain robot motion to assemble a given product, we first solve the AND/OR graph[4, 5] and obtain a possible assembly sequence. Then, we consider constructing the manipulation graph from the assembly sequence including a set of grasping posture. By solving the manipulation graph, we simultaneously

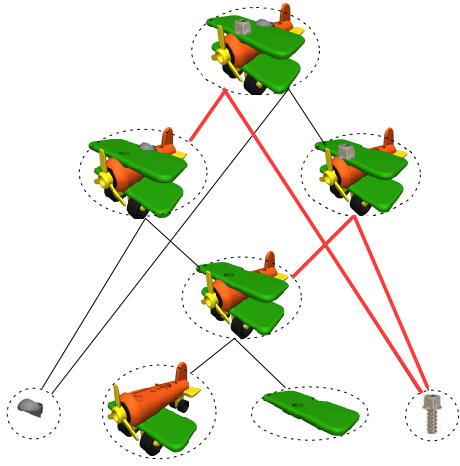


Fig. 2: AND/OR Graph of our assembly task where the red lines denotes the edges with which the action data is associated

plan grasping posture and robot motion. The method is detailed in our previous papers [2, 3].

4. Application of Assembly Motion Data

The grasp/assembly planner explained in the previous subsection just can realize a series of pick-and-place motion. We consider extending this grasp/assembly planner for more complex tasks by using the database of skillfull assembly motion data. In the AND/OR graph shown in Fig.3., the wing is picked by the gripper and put on the body, and then the cockpit can be picked and put on the wing. However, the bolt cannot be assembled to the wing just by using a pick-and-place motion. The edges marked in red shows that the assembly motion data is required for realizing this assembly sequence. We consider associating the action Yaml file corresponding to this assembly sequence. To obtain an assembly sequence, we consider solving the AND/OR graph. If an action Yaml file is associated to the solution edge, we consider using the action Yaml file to generate the robot motion. Otherwise, we consider constructing the manipulation graph and solving it to obtain the robot motion.

5. Example

We consider performing the robotic assembly of a toy airplane. We prepared the skillfull motion of the

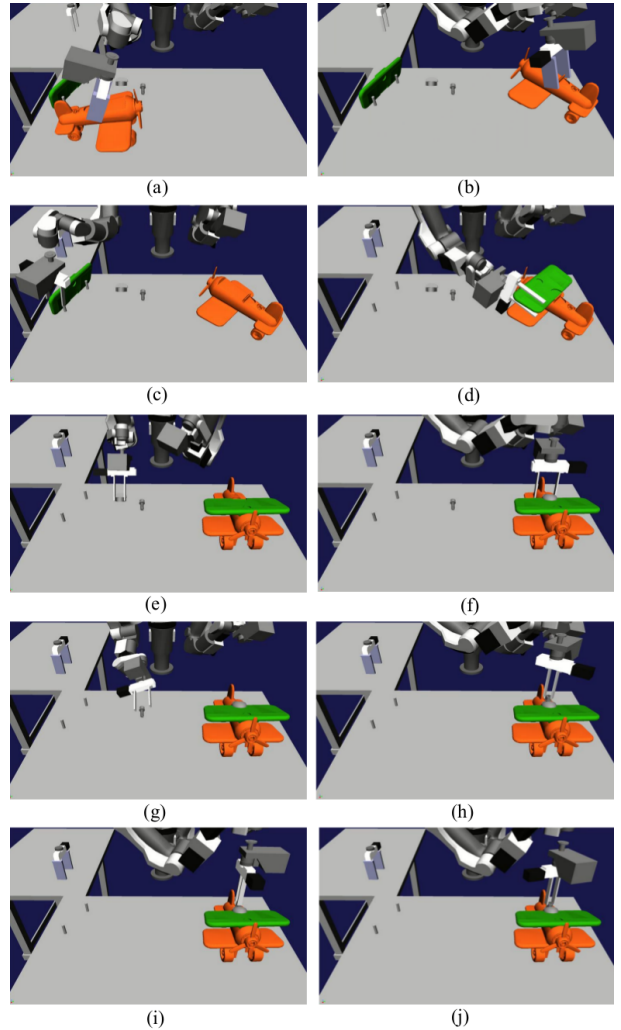


Fig. 3: Simulation result of toy airplane assembly

robot rotating the bolt stored in the database by the YAML format. We first obtained the solution path of the AND/OR graph. Since the solution path includes the screwing motion of the bolt, we used the action YAML data of the screwing motion. The result is shown in Fig. 5.. As shown in the figure, when the robot assembles the bolt to the wing, the robot tries to rotate the bolt by using the task motion data.

6. Conclusions

In this paper, we proposed a motion-data driven robotic grasp/assembly planner. A motion data includes an information on skillfull motion which cannot simply be realized by using conventional grasp/assembly planners. This paper first explained the structure of motion data. Then, we explained the proposed grasp/assembly planner. Finally, we showed a numerical example of a toy

airplane assembly.

This research was conducted under the support of NEDO AI-Robotics Project.

References

- [1] N. Yamanobe, T. Tsuji, K. Harada, K. Nagata, R. Hanai, and W. Wan, and I.G. Ramirez-Alpizar, "Construction of Cloud Database for Task Motion Generation", Annual Conf. on the Robotics Society of Japan, 2016.
- [2] K. Harada, K. Nakayama, W. Wan, K. Nagata, N. Yamanobe, T. Tsuji, and I.G. Ramirez-Alpizar, "Tool Exchangeable Grasp/Assembly Planner", Proc. of Int. Conf. on Intelligent Autonomous Systems, 2018.
- [3] W. Wan and K. Harada, "Developing and Comparing Single-arm and Dual-arm Regrasp," Robotics and Automation Letter, vol. 1, no. 1, pp. 243-250, 2016.
- [4] L.S.M. de Mello and A.C. Sanderson, "And/Or Graph Representation of Assembly Plans", CMU Research Showcase, CMU-RI-TR-86-8, 1986.
- [5] L.S.M. de Mello and A.C. Sanderson, "Automatic Generation of Mechanical Assembly Sequences", CMU Research Showcase, CMU-RI-TR-88-19, 1988.
- [6] T. Siméon, J.P. Laumond, J. Cortés, and A. Aahbani, "Manipulation Planning with Probabilistic Roadmaps", Int. J. Robotics Research, vol. 23, no. 7-8, pp. 729-746, 2004.
- [7] K. Harada, T. Tsuji, and J.-P. Laumond, "A Manipulation Motion Planner for Dual-arm Industrial Manipulators", *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 928-934, 2014.