Planning of Compliant Motions for Fixture Loading

Kyeonah Yu

Abstract : Fixtures are used in almost all phases of machining and assembly to position and hold a part accurately. The class of fixtures which consists of 3 locators and 1 clamp(3L/1C) is known as the minimal set that can provide form closure which is a kinematic constraint condition for preventing all planar motions. This type of fixtures has advantages in terms of the number of fixture elements required, the time for clamping, and so on. However it is not widely used in industry because reliable loading scheme has not been reported. In this paper, we propose a method to load the class of 3L/1C fixtures using compliant motions. The planner is developed for synthesizing compliant motions to achieve precise final fixture configuration in the presence of sensing and control uncertainties. A novel approach to eliminate uncertainty in part orientation by adding one extra fixture element called an aligning pin is proposed.

Keywords : fixture loading, compliant motion planning, robot uncertainty

I. Introduction

A fixture is a device used for locating and holding a part during manufacturing operations such as assembly and machining. Parts must be loaded and unloaded repeatedly and accurately so that such operations can be performed properly. Due to the nature of the fixture, the gap between the fixture and the part is usually very small. Therefore it is hard to load a part into a fixture repeatedly and accurately using a robot, especially in the presence of uncertainties. Even though the design and analysis of fixtures has been extensively studied, little research has been reported on the problem of loading parts into fixtures.

As in Fig. 1, we consider a class of planar fixtures which consists of three round locators and one translating clamp [1, 2]. These 4 point contact fixtures provide *form closure*, which is a kinematic constraint condition that prevents all planar motion. For loading this type of fixtures, the part is inserted into 3 locators first and then the clamp is applied. The 3 locator and one clamp approach has some advantages compared to the traditional fixturing approach as in [3, 4]: 1. The 4 point contact fixtures are known to be a *minimal* set that can provide form closure[5]. 2. It provides rapid loading by reducing time for clamping. 3. There exists a complete algorithm for generating this type of fixtures[1]. In spite of these advantages, it is not widely accepted in industry because loading this type of fixtures is not trivial.

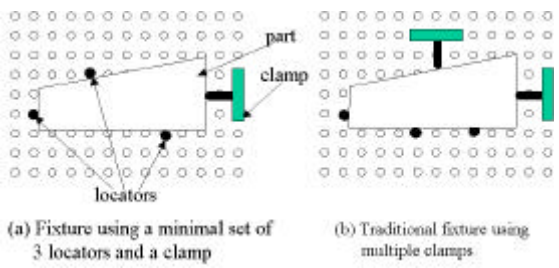


Fig. 1. Examples - 4 point contact minimal fixture and a traditioanl fixture.

Manuscript received : July. 19, 1999., Accepted : March. 20, 2000.
Kyeonah Yu : Dept. of computer science, Duksung Women's University
※This work was supported by the institute of natural science research fund, Duksung Women's University in 1999.

In this paper, we develop the planner synthesizing compliant motions for loading the class of fixtures, which are composed of 3 locators and 1 clamp (3L/1C). The planner will generate fixture loading plans that guarantee successful loading even in the presence of uncertainties. The uncertainties considered in this paper are as follows: the initial position and orientation of the part which cannot be known exactly due to sensing error prior to loading, and robot control error that makes it impossible to achieve a precise commanded velocity of the part. When involving part orientation, the configuration space obstacles become 3 dimensional, which is known to be hard to analyze. Therefore a unique method to eliminate part orientation uncertainty is proposed: using an *aligning pin*. An aligning pin is identical to a locator but not necessarily part of the fixture. It is used for eliminating orientation uncertainty of the part during loading.

Under compliant motions, the part can move to the goal not only directly, but also indirectly by sliding on the surfaces of obstacles, thereby we call them *guides*. In the fixture loading problem, because round locators are used as guides, it is hard to achieve pure translational compliance. Therefore special mechanism is devised called *selective compliance mechanism* (SCM), which is stiff in rotation during translational motions and rotational compliance is provided selectively. Also, simple on/off sensors are integrated to locators to detect the contacts between the part and locators during execution.

1. Previous work

The framework on how to synthesize compliant motions was introduced by Lozano-Perez, Mason, and Taylor in [6]. They considered a class of the peg-in-hole assembly problems: the part can reach the goal not only directly but also indirectly by sliding on the surface of the hole. Their framework is called *preimage backchaining*, where the *preimage* of the goal is defined as a range of positions that can reach the goal in the presence of uncertainty. Erdmann implemented this method by introducing *backprojections* which are defined as computed preimages. As stated above, locators are not polygonal, the configuration space(*Cspace*) obstacles of the fixture loading problem are not polygonal. Instead they become *generalized polygons* (GPs), with linear

and circular edges [7]. To deal with GP obstacles, the *Cspace* representation method by Lozano-Perez [8] and the backprojection algorithm by Erdmann [9] cannot be used directly to handle GP obstacles.

Applying the backprojection algorithm to the fixture loading problem was originally proposed by Yu and Goldberg in [10]. Fig. 2(a) describes possible loading paths as suggested in [10] for a part and a fixture given in Fig.1(a) when the part is initially in the desired orientation. In [10], they mainly focused on extending the existing algorithms[9,6,11] to cover up non-polygonal obstacles without losing the *completeness* of the original algorithms. Orientation uncertainty was not considered when synthesizing compliant motions and it was attempted to be removed by pushing as shown in Fig. 2(b) during loading. However this method to eliminate orientation error is not sufficient because it is not possible to compute the exact pushing direction and force to make the part slide into the goal position. (Refer to [12] for the reasons.) As a result, the part often sticks on the locators and fails to reach the final position. Therefore we propose a new method to eliminate uncertainty in the part orientation in this paper using an *aligning pin*.

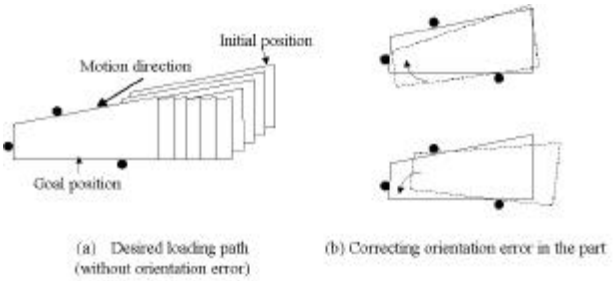


Fig. 2. Loading examples proposed in [10].

The overview of the planner and the executor are briefly explained in section **II**. The modified backprojection algorithm to handle non-polygonal obstacles is explained in section **III**. Section **IV** describes the method for finding an aligning pin to remove uncertainty in the initial part orientation. The simulation results of the planner are demonstrated in section **V**, followed by the conclusion in section **VI**.

II. The overview of fixture loading

There are two main components in fixture loading: the off-line planner which synthesizes a fixture loading plan and the executor which actually executes the plan. The executor consists of a robot which actually executes the plan, the passive compliance mechanism which is specially devised to execute compliant motions generated by the planner, and the sensor-based fixtures to detect contact state between the part and fixture element(fixel). Their relationship is described in Fig. 3.

1. The planner

The planner assumes that position uncertainty of the part is bounded by a ball of radius ε_p, velocity uncertainty is bounded by a semi-infinite cone whose axis is the commanded velocity, whose apex is the initial position, and whose angle is $2\varepsilon_v$, and friction between the part and each fixel conforms to Coulomb's law. The input to the planner is a planar polygonal part represented by an ordered set of vertices, the initial configuration of the fixture with a clamp loosened, and the desired final configuration of the part in the fixture.

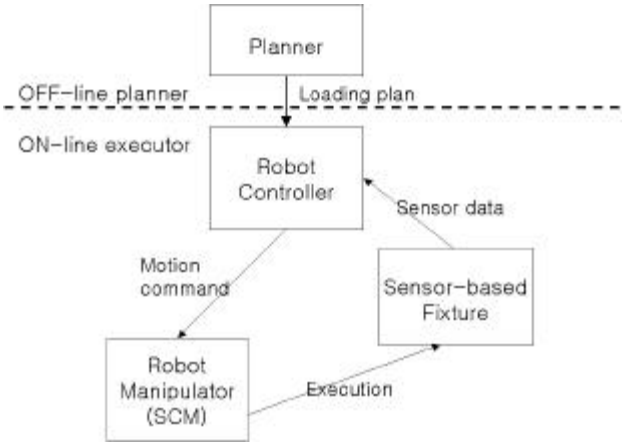


Fig. 3. Components of a fixture loading plan.

Planning Steps

Step 1 : Reduce the original fixture loading problem to the problem of motion planning of a point in generalized polygonal *Cspace*. Then abutting edges to the goal are disposed, which can be used as guides to the goal. These edges are candidates for an aligning pin.

Step 2 : Select an aligning pin. We call an aligning pin and a locator on the same edge an *aligning pair*.

Step 3 : Compute a range of initial positions for a part and a commanded velocity to reach the aligning pair which is turned to an *intermediate goal* or a *subgoal* of the fixture loading (Fig. 4(a)).

Step 4 : Compute a commanded velocity to reach the goal from the subgoal (Fig. 4(c)).

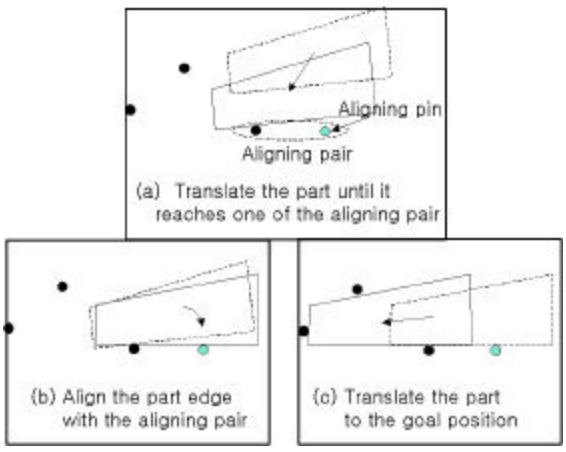


Fig. 4. Example executing a loading plan generated by the planning algorithm.

2. The executor

The executor consists of three components as in Fig. 3. In sensor-based fixture, each fixture element is equipped with a simple binary sensor which turns ON when it makes contact with the part and turns OFF when the contact breaks [10, 13]. This sensor information is forwarded to the robot controller and the robot controller command motions according to this information. The loading plan generated by the planner is executed as follows:

Execution Steps

Step 1 : Move the part into a subgoal from a given initial position by translational motion(Fig. 4(a)).

Step 2 : Align the part with the aligning pair(Fig. 4(b)).

Step 3 : Translate the part along with the aligning pair to the goal(Fig. 4(c)).

The special robotic manipulator is required for such execution that must satisfy the followings: ① It must be stiff in rotation during translational motion as in Fig. 4(c) to prevent the part pivoting about the locator. ② Rotational compliance can be activated selectively in the horizontal plane. In other words, in the example of Fig. 4, rotational compliance should be allowed in (b). This type of compliance is achieved by attaching a selective compliance mechanism to the wrist of the manipulator.

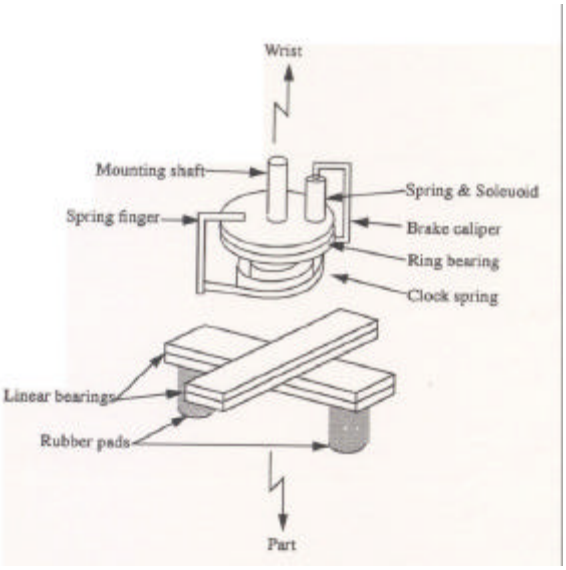


Fig. 5. Selective compliance mechanism.

The SCM designed to meet these conditions is shown in Fig. 5. There are two linear bearings above the rubber pads, normal to each other, and parallel to the fixture platten. Above the linear bearings, there is a selectable rotational compliance mechanism. When selected, the device is compliant to rotational (about the vertical axis) motion, and when deselected, it is rotationally stiff. A clutch mechanism, activated by a compressed air cylinder or electrical solenoid, is used to select and deselect the rotational compliance. The linear compliance is always present. The stack order of the devices is not critical to the operation, but it will be simpler to implement if the active device is above the passive devices and hence, closer to the manipulator base as shown in Fig. 5.

When contact with the first fixel is achieved, the part surface should be aligned with the virtual edge connecting the locator and the aligning pin by using the rotational compliance of the SCM. The rotational compliance mechanism clutch is released at this time, allowing the part to rotate as it is pushed into contact with the aligning pin. The rotational centering spring has a very low spring constant so that torsion from the spring is much lower than torque from sliding friction. The only function of the rotational centering spring is to bring the manipulator pads to a known orientation when the manipulator is raised.

Translational compliance is achieved by the two linear bearings. When the manipulator pushes the part against the fixel(s), the part slides in a direction parallel to the contacting part edge without rotating. The two centering springs on the linear bearings are chosen with spring constants so that sufficient sliding force will be generated by modest displacement input from the manipulator.

III. Algorithm for planning translational motions

In this section, the *Cspace* representation algorithm for planning step 1 and the backprojection algorithm for planning step 3 and 4 are explained. In other words, the method of reducing the problem of fixture loading to the problem of motion planning of a point in generalized polygonal *Cspace* in section 3.1, and the backprojection algorithm for computing a commanded velocity and a range of initial positions for reaching a given goal is explained in section 3.2.

1. Problem representation in generalized polygonal configuration space

We convert the problem of moving a part to the equivalent problem of moving a point in *Cspace*. Fixels are treated as obstacles. If we let li be fixels and P a part and $COP(li)$ be the *Cspace* obstacle for li, $COP(li)$ can be computed by the equation, $COP(li) = \ominus P \oplus li$ where $\oplus$ and $\ominus$ are Minkowski operators defined as $P \oplus li = \{p+l \mid p \in P, l \in li\}$ and $\ominus P = \{-p \mid p \in P\}$ [8]. To compute *Cspace* obstacles for locators, we first expand the part by the locator radius[14]. The expanded part, EP, is not polygonal, but generalized polygonal and locators can be treated as points. Then $COP(li)$ for locators is simply obtained by translating $\ominus EP$ by (x, y) where (x, y) are the coordinates of li. The *Cspace* representation for multiple obstacles can be obtained by the union of individual obstacles, $COP(L) = \cup_{i=1,3} COP(li)$, where $L = \cup_{i=1,3} li$. This process is depicted in Fig. 6. The figure at the bottom represents the problem of moving a point equivalent to the figure at the top(original fixture loading problem). G marked with "■" is the goal of the converted problem.

2. GP backprojection algorithm

In the fixture loading problem, the goal is a point because the part configuration in the fixture must be unique in its position and orientation. A *preimage* of the goal is defined as a set of initial positions of the part that are

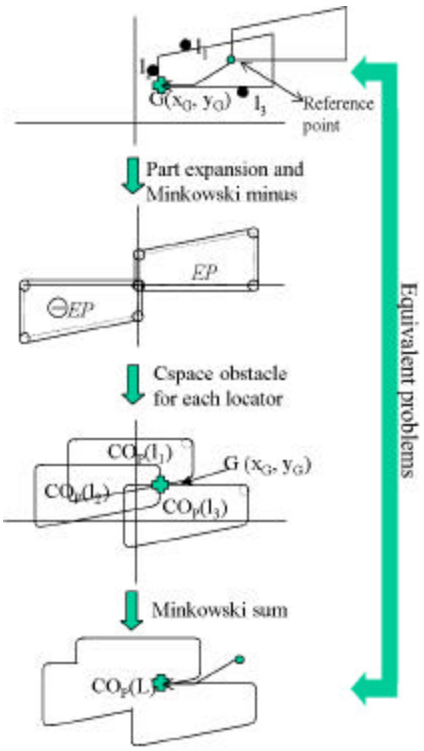


Fig. 6. Generalized polygonal Cspace obstacle.

guaranteed to reach the goal. A *backprojection* represents a restricted class of computable preimages and is defined as a region of *C*space such that the part can reach the goal with a commanded velocity. We apply Erdmann's backprojection algorithm [9] to compute initial positions for a part for a specific velocity direction.

We state the following property similar to Laumond's [15] to extend the algorithm to GP environments.

Property : *The boundary of backprojections of generalized polygonal Cspace obstacles consists of generalized polygonal lines which are linear segements and arcs of a circle*.

Proof : Let *BP* be a backprojection and *CO* Cspace obstacles. The set of boundaries of *BP* is a union of rays and a subset of boundaries of *CO*. Hence the boundary of *BP* consists of generalized polygonal lines.

GP backprojection algorithm

Step 1 : In *COP*, mark every non-goal vertex(or endpoints of an arc) at which sticking is possible. Mark every goal vertex(or endpoints of an arc) if sliding away or sticking is possible on an adjacent non goal edge.

Step 2 : At every marked vertex(or endpoints of an arc), erect the inverted velocity cone. The interior side of the cone is the illegal region that causes sticking at its apex.

Step 3 : Trace out the backprojection region starting from the goal, G. Refer Fig. 7.

For vertices, sticking is possible if the inverted velocity cone overlaps with the friction cone. It is determined whether sticking is possible on an arc as follows. For arcs, the normal to the surface continuously changes between two normals of adjacent line segments. So does the friction cone. The axis of the friction cone on an arc is continuous

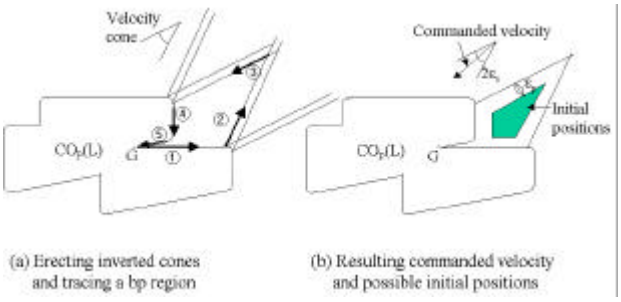


Fig. 7. Backprojection for a generalized CO.

between two normals of the adjacent line segments. Therefore we compare the inverted velocity cone and the range of friction cones of an arc which corresponds to the *convex combination* of the normals of the adjacent line segments. If the overlap between the inverted cone and the range is non null, sticking is possible on an arc. In other words, the exact range of an arc on which sticking may occur can be computed. However as no vertex contact is allowed for fixturing, we can simplify the process for computing backprojections by erecting the inverted velocity cone only at two end points of an arc at any point of which sticking is possible.

IV. Algorithm for eliminating uncertainty in part orientation

Consider the fixture consisting of a polygonal locator and a cylindrical locator. Uncertainty in part orientation can be eliminated by using the surface of the polygonal locator. By activating the rotational portion of the SCM, the part surface can be aligned with the surface of the locator. And then the part is moved only by translational compliance to the goal (Fig. 8(a)).

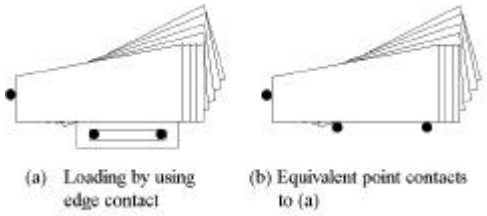


Fig. 8. Loading by using edge-contact and its equivalent point contact.

We can implement the kinematic equivalent contact of an edge contact by using a pair of round locators: we put two round locators into contact with the part surface (Fig. 8(b)). The part can be aligned to the virtual edge connecting two round locators. This leads us to introduce an extra pin to fixtures for the sole purpose of eliminating uncertainty in part orientation. We call it an *aligning pin*.

An aligning pin is identical to a locator but not necessarily part of the fixture. It is used for eliminating orientation uncertainty of the part during loading. After achieving a desired part orientation by aligning the part surface with the virtual edge connecting the locator and the aligning pin, the part moves only by translations by locking

the rotational mechanism of the SCM.

Aligning pin selection algorithm

Step 1 : Find abutting edges to the goal(G) from *COF(L)*. One of these edges will guide the part into the goal. In Fig 9, e1 and e2 are abutting edges to G.

Step 2 : For each edge from step 1, extend it from the vertex(other than G) by the distance of that edge. See Fig. 9(a). Then pin holes(the center of holes) on the extended edges are candidates for an aligning pin(Fig. 9(b)).

Step 3 : Select one of candidates. If the aligning pin is too close to the locator(say h1 in the example in Fig 9), it is not completely free from pivoting. If the aligning pin is too far fromthe locator(say h8 in the example), achieving the subgoal is not easy. Therefore when multiple candidates exist, one in the middle is chosen.

Step 4 : Then the subgoal is the overlapped portion of the existing *COF(L)* and the *Cspace* obstacles added newly. Fig. 11 explains this when h4 is selected as an aligning pin.

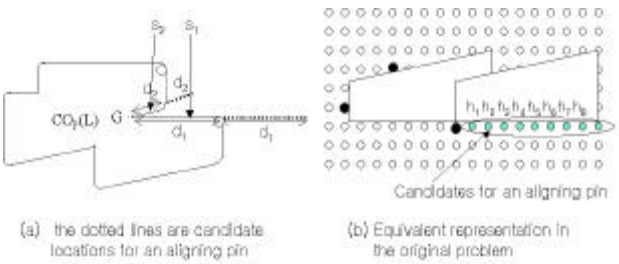


Fig. 9. Candidate locations for the aligning pin.

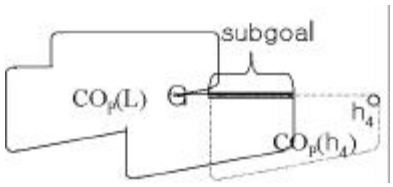


Fig. 10. Modified CO by adding an aligning pin.

V. Simulation results

The fixture loading planner and executor simulator has been implemented using Visual C++ on the Microsoft Windows 98 environment. The planner takes two input files: One has coordinates of vertices of the part and locations of three locators in order. The other has values of the radius of locators, ε_p, ε_v, ε_o, and the coefficient of friction μ. The program consists of individual modules reading inputs and computing GP Cspace obstacles, backprojections, and an aligning pin. Fig. 11 shows the execution of the Cobstacle module and the backprojection module for a fixture configuration for the part in Fig 1.

Simulation results tested on some other polygons are shown in Fig. 12. We can see that the aligning pin which is marked with a empty circle is used only for loading and is not part of the fixture in the final configuration. When the part makes contact with either the aligning pin or the first locator as it is moved in the direction computed by the backprojection algorithm, the rotational compliance of the SCM is selected and the part is rotated freely until the part is aligned along the reference edge. Then the part is moved only by the translational compliance with the rotational compliance locked.

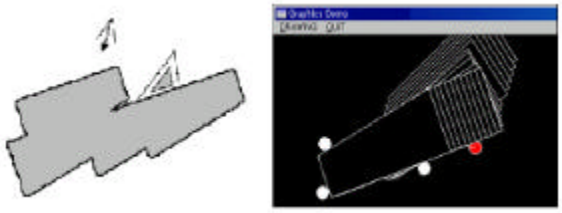


Fig. 11. Backprojection computed for the part in Fig. 1 and the corresponding loading path.

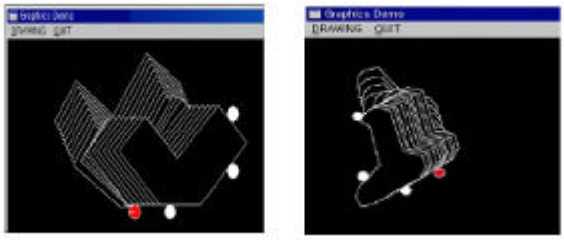


Fig. 12. Simulation results tested on a 7-gon and a 14-gon(glue gun case).

Test fixtures were chosen on three polygonal parts shown in Fig. 11 and 12. For each part, ten, eleven, and four sample fixtures were given to the planner as input to synthesizing a loading plan. These samples were selected such that there are only topologically distinguished fixture configurations in test sets. Also we excluded fixture configurations with no translational freedom. See Fig. 13.

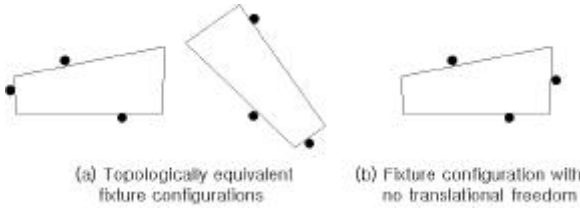


Fig. 13. Examples excluded from the test sets.

Planning results were compared with the previous work in terms of two factors: the success rate of planning a method for eliminating orientation uncertainty and the allowable orientation error bound. The success rates are summarized in table 1.

The method used in [10] failed in many cases for which it can't find any pushing point on the part or any pushing direction. Refer to [10]. The proposed method sometimes failed in finding a proper hole for an aligning pin when the modular fixture system with lattice holes is used. However if we use the reconfigurable fixture system without lattice [3] in which the location for a fixture element is not restricted to the lattice, the proposed method never failed for all the cases of our simulation.

Table 1. Comparison results based on the success rate of planning.

Parts	Number of test fixtures	Success rate of planning		
		Without an aligning pin	With an aligning pin (on the lattice)	With an aligning pin (on arbitrary place)
4-gon	10	60%	80%	100%
7-gon	11	55%	91%	100%
14-gon	4	50%	50%	100%

The aligning pin method is less sensitive to the initial part orientation error than the method of [10]. The former tries to eliminate orientation uncertainty after making contact with one locator whereas the latter does after achieving the second contact with another locator. In other words, the process of eliminating orientation uncertainty occurs under less restricted condition with the aligning pin method. Therefore the aligning pin method is more tolerable about the degree of the initial orientation error and allows larger error bound than the previous method in [10]. Fig. 14 shows loading examples by two methods when the part is oriented by -7.5o initially. With the method without an aligning pin, the part can be stuck as in Fig. 14(a) depending on the degree of the initial orientation error, but with the other one, the part is not stuck as in Fig. 14(b).

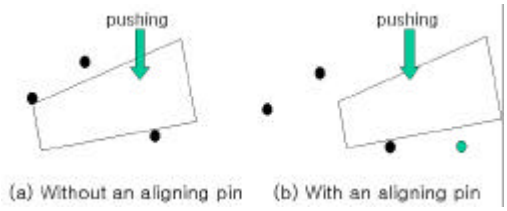


Fig. 14. Comparison of eliminating orientation error - (a) shows the case that the part is stuck due to a larger error value than the allowable error bound and (b) shows that the loading process can be successful in the same situation when using an aligning pin.

VI. Conclusions

In this paper, a new method to load the class of 3L/1C fixtures in the presence of uncertainties using compliant motions is proposed. A unique approach to eliminate uncertainty in part orientation is also proposed. The simulation results show that the proposed method can eliminate uncertainty in part orientation easily while performing compliant motions generated by the planner just by using *one extra pin*. Compared to the pushing method used in [10], this method is less restricted to the exact location and moving direction for the SCM. Moreover, the proposed method is more tolerable about uncertainty in initial part orientation. However this method has a limitation that it may fail to find lattice holes for the aligning pin due to the discreteness of the grid holes. Therefore it is desirable that the planning results of fixture loading are forwarded to the fixture designer so that this limitation can be considered in fixture designing phase.

References

[1] R. Brost and K. Goldberg, "A complete algorithm for synthesizing modular fixtures for polygonal parts," *IEEE Trans. Robotics Automation*, vol. 12(1), pp. 31-46, Feb, 1996.

[2] Y. C. Chou, V. Chandru, and M. M. Barash, "A mathematical approach to automatic configuration of machining fixtures: analysis and synthesis," *Journal of Engineering for Industry*, vol. 111, pp. 299-306, 1989.

[3] H. Asada and A. By, "Kinematic analysis of workpart fixturing for flexible assembly with automatically reconfigurable fixtures," *IEEE Trans. Robotics and Automation*, vol. 1(2), pp. 85-93, 1985.

[4] M. Mani, "Automated design of workholding fixtures using kinematic constraint synthesis," *Ph.D thesis, Dept. of Mechanical Engineering*, Northwestern university, June 1988.

[5] X. Markenscoff, L. Ni, and C.H. Papadimitrou, "The geometry of grasping," Int. Journal of Robotics Research, vol 9(1), pp. 28-35, 1990.

[6] T. Lozano-Perez, M. Mason, and R. Taylor, "Automatic synthesis of fine motion strategies for robots," *Int. Journal of Robotics Research*, vol. 3(1), pp. 3-24, 1984.

[7] A. Rao and K. Goldberg, "Orienting generalized polygonal parts," *Proc. IEEE International Conference on Robotics and Automation*, CA, April, 1992, pp. 2263-2268.

[8] T. Lozano-Perez, "Spatial planning: A configuration space approach," *IEEE Trans. Computers*, vol. 32(2), pp. 108-120, 1983.

[9] M. Erdmann, "Using backprojections for fine motion planning with uncertainty," *Int. Journal of Robotics Research*, vol. 5(1), pp. 19-45, 1986.

[10] K. Yu and K. Goldberg, "A complete algorithm for fixture loading," *Int. Journal of Robotics Research*, vol. 17(11), pp. 1214-1224, 1998.

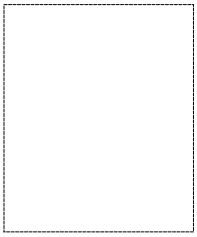
[11] B. R. Donald, "Error-detection and recovery in robotics," *Springer-Verlag*, 1989.

[12] M. T. Mason, "Compliance and force control for computer controlled manipulators," *IEEE Transactions on Systems*, Man, and Cybernetics, 11(6):418-432, 1981.

[13] B. Benhabib, K. Chan, and M. Dai, "A modular programmable fixturing system," *ASME J. Eng. Industry*, vol. 113(1), pp. 93-100, 1991.

[14] J. Rossignac and A. G. Requicha, "Offsetting operations in solid modelling," *Computer Aided Geometric Design*, vol. 3, pp. 129-148, 1986.

[15] J. P. Laumond, "Obstacle growing in a nonpolygonal world," *Inform. Processing Letter*, vol. 25(1), pp. 41-50, 1987.



Kyeonah Yu

Kyenah Yu was born in Seoul, Korea, on January 27, 1964. She received B.S. and M.S. degrees in the department of control and instrumentation engineering at Seoul National University in 1986 and 1988, respectively and Ph.D degree in the department of computer science at University of Southern California in 1995. She joined the department of Computer Science at Duksung Women's University in 1996, where she is an assistant professor. Her current research interests include Robot algorithms, Intelligent agent, and Knowledge-based systems.