Proc. 1995 IEEE Int. Conf. on Intelligent Robots and Systems
vol.2, pp.77-83, Pittsburgh, PA, August 1995

Assembly of Flexible Objects without Analytical Models*

Jun Miura

Department of Mechanical Engineering
for Computer-Controlled Machinery
Osaka University
jun@cv.ccm.eng.osaka-u.ac.jp

Katsushi Ikeuchi

School of Computer Science
Carnegie Mellon University
ki@cs.cmu.edu

Abstract

The ability of manipulating flexible objects, such as rubber belts and paper sheets, is important in automated manufacturing systems. This paper describes a novel approach to assembly of flexible objects. The operation dealt with in this paper is to assemble a rubber belt with fixed pulleys. By analyzing possible states of the belt based on the empirical *knowledge of the belt, we can derive a method to have not only the action planning but also the visual verification planning. We have implemented a belt assembly system using the two manipulators and a laser range finder as the sensor, and succeeded in performing the belt-pulley assembly. Extension of our approach to other kinds of assembly of flexible objects is also discussed.*

1 Introduction

Many manufacturing processes involve assembly of flexible objects such as rubber belts and paper sheets. There is a need for the ability of manipulating flexible objects to realize automated manufacturing systems. Past research on this subject can be roughly divided into two approaches: development of task-specific systems [1] [8] [9] and development of analytical models of flexible objects [2] [10] [11]. Development of a task-specific system is difficult. Construction of analytical models of flexible object is also hard; in addition, its algorithmic implementation is computationally expensive.

On the other hand, a human can relatively easily manipulate flexible objects without explicit analytical models of the objects. We attribute this mainly to the following two human abilities: the ability of *problem abstraction* and the ability of *efficient sensory feedback*.

The ability of problem abstraction plays an important role in human problem solving [6]. In the case of assembly of flexible objects, by capturing qualitative behavior of flexible object based on empirical knowledge of the object, a human can segment a continuous problem space into a set of discrete *states*. By reasoning in this abstract state space, a feasible plan is then obtained efficiently.

Sensory feedback is also important for successful manipulation of flexible objects (e.g., [4]). The ability to focus attention is essential to realizing efficient and robust sensor-guided manipulation. The more restrictive constraints we have on the possible state of a manipulated flexible object, the more efficiently sensor data can be collected.

Based on the conjecture of the human abilities, we propose a novel approach to assembly of flexible objects. The operation dealt with in this paper is to assemble a rubber belt with fixed pulleys (see Fig. 1). Fingers (a parallel-jaw gripper or a auxiliary rod finger) and visual sensors are used for the operation. The shape and the position of pulleys and other solid objects are described in a CAD-based world model. We assume a rubber belt is flexible enough in the axial direction to be streched up to a necessary extent. We call this problem the *belt-pulley problem.* We will show that by introducing an appropriate constraint on the possible state of the belt, the original complex belt-pulley problem can be significantly simplified in terms of both action planning and visual verification planning.

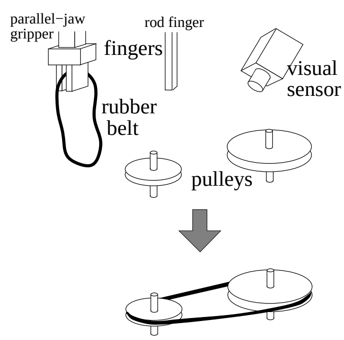


Fig. 1: The belt-pulley problem.

*This research was sponsored by the Advanced Research Projects Agency under the Department of the Army, Army Research Office under grant number DAAH04-94-G-0006.

2 Constraint for Simplifying the Problem

This section introduces a constraint for simplifying the belt-pulley problem. We use the following *empirical* knowledge about the rubber belt:

> "If a belt is pulled *taut* by fingers and/or pulleys, the belt is stationary and the shape and the position of the belt is uniquely determined from the shape and the position of the fingers and/or pulleys."

We call the above state a *stable state*. The condition that a state is stable is represented by

$$\sum_i l_i > l_{thresh}, \qquad (1)$$

where l_i is the length of the ith part (between the ith and the $i+1$th pulley or finger), l_{thresh} is the predetermined threshold (see Fig. 2). The threshold is determined according to the length of the belt when no pulling force is imposed on it.

Based on this knowledge, we introduce the following constraint:

"A belt is manipulated via stable states only."

This constraint significantly simplifies the problem in the following two points:

(1) *Search space of actions becomes small enough to be tractable.*
By introducing the notion of *state*, the planning problem can be decomposed into two levels: state transition planning and trajectory planning between states. By solving these problems hierarchically, the search space is greatly reduced [3] [6].

(2) *Visual verification planning becomes easier.*
Since the shape and the position of the belt is uniquely determined from the position of fingers and pulleys in a stable state, exploratory visual sensing is unnecessary. The only task of the vision system is to verify if the shape and the position of the belt are as desired.

Under the above constraint, the belt-pulley problem is decomposed into the following three subproblems:

- find the sequence of state transitions from the initial state to the goal state (see Section 3);
- find the appropriate trajectories of fingers for the given state transition (see Section 4); and
- find the appropriate visual verification plan to ensure the state transition (see Section 5).

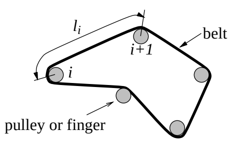


Fig. 2: A stable state.

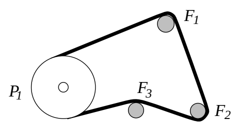


Fig. 3: State $P_1F_1F_2F_3^*$.

If the initial state is not a stable state, we treat that state as an exception. We assume that a certain procedure to move from the initial state to some stable state is given in such a case.

3 High-Level Planning: Generating Candidates for State Transition Sequences

The high-level planner generates the candidates for state transition sequences from the initial stable state to the goal state.

3.1 State Representation

In this level of planning, we represent the state qualitatively; we consider only two properties of pulleys and fingers: one is the qualitative position (order) of a pulley (or a finger) in the top view; the other is if a pulley (or a finger) is inside the belt.

Let P_i $(i = 1, \ldots, m)$ and F_i $(i = 1, \ldots, n)$ denote the ith pulley and the ith finger, respectively. A state is represented by listing clockwise the symbols of pulleys or fingers which are touching the belt. If a pulley or a finger is outside the belt, the superscript "$*$" is attached to the symbol. For example, the state shown in Fig. 3 is represented as $P_1F_1F_2F_3^*$. Any symbol in the list can be the starting one; all of its cyclic permutations are equivalent.

3.2 Operators

We then define *operators* to describe the transition between states. We currently use the following eight operators (Fig. 4 explains some of them):

- $ADD\text{-}FINGER\text{-}INSIDE(x, y, z)$: finger x enters the inside of the belt and touches the belt at the position between y and z (see Fig. 4(a)).
- $ADD\text{-}PULLEY\text{-}INSIDE(x, y, z)$: pulley x enters the inside of the belt and touches the belt at the position between y and z (see Fig. 4(b)).
- $ADD\text{-}FINGER\text{-}OUTSIDE(x, y, z)$: finger x approaches the belt from the outside and touches it at the position between y and z.
- $ADD\text{-}PULLEY\text{-}OUTSIDE(x, y, z)$: pulley x approaches the belt from the outside and touches it at the position between y and z.
- $REM\text{-}FINGER\text{-}INSIDE(x)$: finger x inside the belt detaches from the belt and exits to the outside.
- $REM\text{-}PULLEY\text{-}INSIDE(x)$: pulley x inside the belt detaches from the belt and exits to the outside.
- $REM\text{-}FINGER\text{-}OUTSIDE(x)$: finger x detaches from the belt.
- $REM\text{-}PULLEY\text{-}OUTSIDE(x)$: pulley x detaches from the belt.

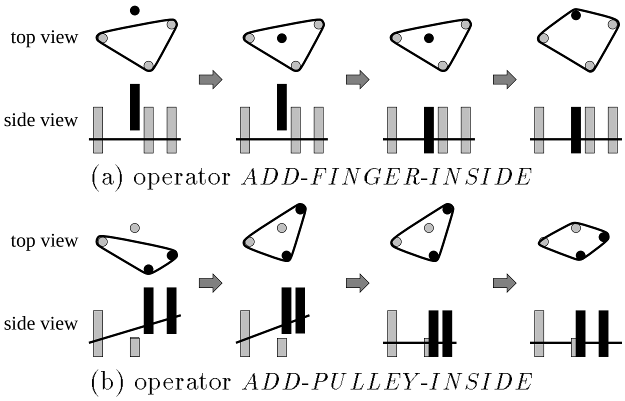


Fig. 4: Examples of operators for state transition. Black lines indicate the belt; black figures indicate fingers moved; shaded figures indicate pulleys or stationary fingers.

3.3 Generation of Candidate State Transition Sequences

The high-level planner generates candidates of operator sequences which can move from the initial state to the goal state. By combining the forward breadth-first search from the initial state and the backward breadth-first search from the goal state, the planner searches for the minimum-step plans. Since the feasibility of each high-level plan cannot be decided without taking care of low-level details such as generation of collision-free trajectories, the high-level planner generates all possible candidates of high-level plans, and then passes them to the low-level planner. If none of them is found to be feasible by the low-level planner, the high-level planner will generate the second-shortest plans and passes the new plans to the low-level planner again. This final step is repeated until the satisfactory final plan is generated.

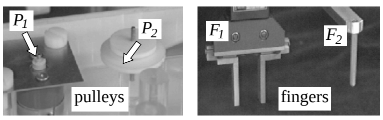


Fig. 5: Pulleys and fingers used in an example problem.

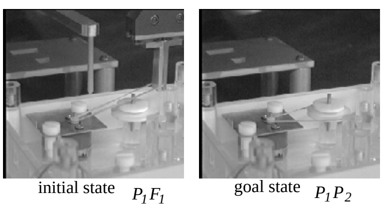


Fig. 6: The initial and the goal states.

3.4 Example

Suppose we have two pulleys (P_1 and P_2) and two fingers (F_1 and F_2) as shown in Fig. 5. Also, let the initial state and the goal state be P_1F_1 and P_1P_2, respectively (see Fig. 6).

The result of search is represented as a directed graph, which includes the initial state, the goal state, and several transitional states. We call this graph a *transition graph*. Each possible path connecting the initial and the goal states represent a candidate plan of state transition.

Fig. 7 shows a transition graph representing the shortest plans obtained by the high-level planner; the number of steps is two; there are two candidate plans. If neither of the plans is found to be feasible by the low-level planner, the second-shortest plans are generated. Fig. 8 shows a transition graph representing the second-shortest plans; the number of steps is four; there are twelve candidate plans.

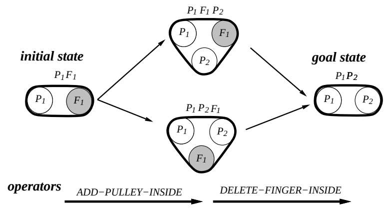


Fig. 7: Transition graph for the shortest plans of state transition. Applied operators are indicated at the bottom.

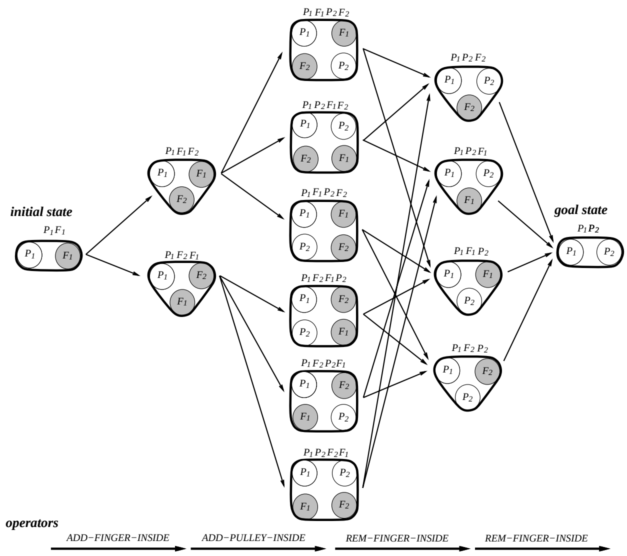


Fig. 8: Transition graph for the second-shortest plans of state transition. Applied operators are indicated at the bottom.

4 Low-Level Planning: Generating Finger Trajectory

The low-level planner generates trajectories of fingers based on the state transition sequence obtained in the high-level planning.

4.1 Representation of Trajectory and Constraints on Trajectory Generation

We discretize the 3-D space for simplicity of trajectory generation. We first vertically divide the 3-D space in which fingers moves into slices of horizontal 2-D space. We then divide each 2-D space into a set of square grids and check the possibility of collision at the center of each grid. A trajectory consists of a series of the grids traversed. We currently limit the movement of a finger to those which are composed of movements parallel to grid axes.

A trajectory generator is prepared for each operators listed in Section 3.2. Each state transition indicated by an operator can be decomposed into several consecutive steps. For example, *ADD-FINGER-INSIDE*(x, y, z) roughly takes the three steps shown in Fig. 4(a): move a finger above the region formed by the belt; lower the finger below the height of the belt; and pull the belt outward. To realize each step, a certain configuration of fingers should be achieved. Thus, in trajectory generation for a state transition, such a configuration is determined first. Then, actual trajectories connecting consecutive configurations are generated by considering the following two conditions:

- A finger must not collide with other fingers, pulleys or other objects.
- If a finger is touching the belt, it must move so that the belt is at the current stable state (i.e., equation (1) is satisfied) at any moment of the finger movement.

Before generating a trajectory, a set of grids is enumerated which satisfies the above two conditions. Then, from this set and the possible position set of the intermediate states, a feasible sequence of grids is collected as the final trajectory.

If no feasible sequences of grids can be obtained, the current sequence of state transition, which is given by the high-level planner, is considered to be infeasible.

4.2 Example

Suppose we are generating a trajectory for *ADD-FINGER-INSIDE*(F_2, P_1, F_1) with the initial state P_1F_1. Fig. 9 shows the state before the transition. Fig. 10 shows the area where F_2 does not collide with other objects. Fig. 11 shows the feasible configurations of F_2 to realize the above-mentioned three steps in this state transition; position set *PS-1* indicates the area where F_2 can enter the inside of the belt from the above; position set *PS-2* indicates the area where the stable state $P_1F_2F_1$ is realized. Considering these areas, the final trajectory for this operator is determined as shown in Fig. 12.

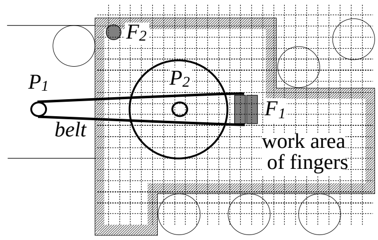


Fig. 9: An example state P_1F_1 before the transition. The belt is over the pulley P_2. Predetermined grids are indicated with dotted lines.

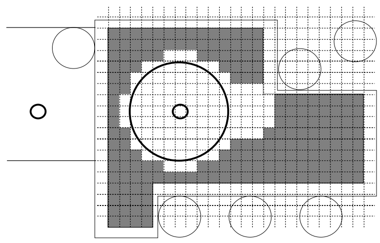


Fig. 10: Shaded grids indicate the position where F_2 does not collide with other objects.

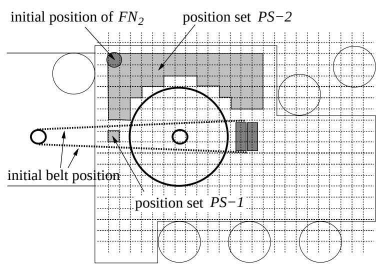


Fig. 11: Two position sets: *PS-1* for inserting the F_2 inside the belt, *PS-2* for achieving the stable state $P_1F_2F_1$.

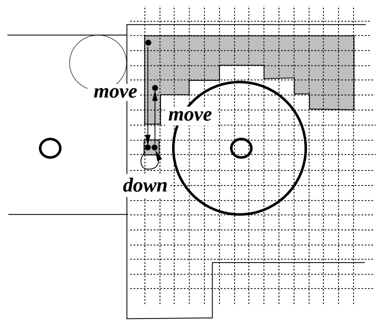


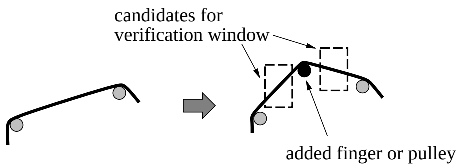
Fig. 12: A generated trajectory of F_2 for *ADD-FINGER-INSIDE*(F_2, P_1, F_1).

5 Visual Verification Planning

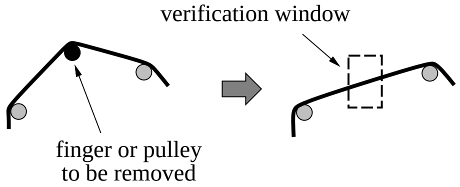
Vision is used for the following two purposes:

- verify that a state transition has been accomplished; and
- verify that the precondition for a state transition is established.

For the verification of completion of a state transition, the position of a certain part of the belt is measured where some change should occur, and is compared with the desired position. If a pulley or a finger is added, the part of the belt corresponding to either side of that added pulley or finger is examined (see Fig. 13(a)). If a pulley or a finger is removed, the part of the belt around the position where that removed pulley or finger was touching is examined (see Fig. 13(b)). The visual verification planner generates a region of interest to be examined; we call this region a *verification window*. At execution time, the visual sensor is placed so that the part of the belt inside the verification window can be observed, and so that the sensor does not collide with other objects. Only the two-dimensional position is enough for this verification.



(a) verification of add operation



(b) verification of remove operation

Fig. 13: Verification windows.

The state transitions that require the verification of precondition are *ADD-PULLEY-INSIDE*(x, y, z) and *ADD-PULLEY-OUTSIDE*(x, y, z). In order to correctly set the belt into the ditch of the pulley, the height of the belt needs to be adjusted as shown in Fig. 14. For this purpose, several parts of the belt which are supported by fingers are examined (see Fig. 15). Since the position of the pulleys is known in advance, we can calculate the desired position of the belt. At execution time, adjustment of the belt height with visual information is repeated until the precondition shown in Fig. 14 is satisfied.

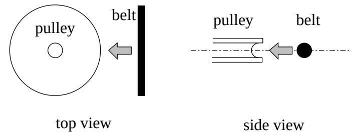


Fig. 14: The height of the belt needs to be adjusted to the height of the pulley before mating.

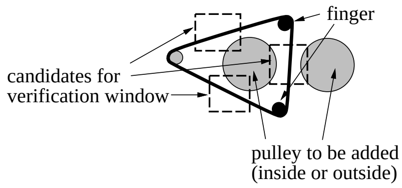


Fig. 15: Candidate verification windows for checking precondition.

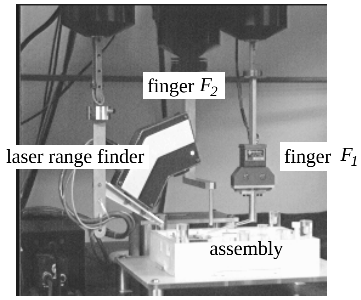


Fig. 16: Experimental setup.

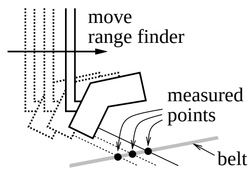


Fig. 17: A strategy to observe the belt position.

6 Experiment

6.1 Experimental Setup

Fig. 16 shows the experimental setup. Three overhead modules in the RobotWorld [7] are used for the parallel-jaw gripper (F_1), the rod finger (F_2), and the line laser range finder [5]. In the figure, a model of cassette tape recorder is being assembled. The assembly process of the tape recorder includes the belt-pulley problem, which is the focus of this paper.

In order to calculate the position of the belt, several points on the belt are measured while the range finger moves horizontally as shown in Fig. 17, and a line is fitted to the measured points.[1] We can calculate the three-dimensional equation of the line.

6.2 Generated Plan and Actual Sequence of Operations

The plan for the problem shown in Fig. 6 is generated as follows.

[1] In stable states, the portion of the belt between pulleys and fingers is always a straight line.

The low-level planner first examined the feasibility of two-step plans shown in Fig. 7, and found that neither of two operators *ADD-PULLEY-INSIDE*(F_1, P_2, P_1) and *ADD-PULLEY-INSIDE*(P_1, P_2, F_1) is applicable to the initial state (P_1F_1) because pulley P_2 is too large to enter the inside of the belt formed by P_1 and F_1 without touching the belt. Then, the low-level planner examined the second-shortest plans, i.e., plans shown in Fig. 8 and found four out of twelve candidates for state transition sequences were feasible.

Fig. 18 shows the generated plan of finger movement and visual verification from one of four feasible sequences, $P_1F_1 \rightarrow P_1F_2F_1 \rightarrow P_1F_2P_2F_1 \rightarrow P_1F_2P_2 \rightarrow P_1P_2$. Due to the collision possibility, feasible verification windows for the second state transition (*ADD-PULLEY-INSIDE*(P_1, F_2, P_2, F_1)) could not been obtained. Instead, we considered that the verification for the next (third) transition (*REM-FINGER-INSIDE*(F_1)) also verified the second transition. Fig. 19 shows the actual process of the operation; the belt was successfully set around the two pulleys. The trajectory in the operation from the initial *unstable* state (F_1) to the initial *stable* state (P_1F_1) was designed manually[2], although the visual verification was planned by the low-level planner. Fig. 20 shows another successful belt-pulley operation.

7 Conclusion and Discussion

This paper has presented a novel approach to assembly of flexible objects without analytical models of the objects. We dealt with the belt-pulley problem as an example. By introducing the notion of *stable state* and by restricting the possible states of the belt to stable ones, the original complex problem was significantly simplified in terms of both action planning and visual verification planning. We have implemented an experimental system that succeeded in assembling a belt with pulleys.

Our approach might be applicable to other assembly processes which involve manipulation of flexible objects. By abstracting the problem using appropriate constraints on the original problem, a feasible plan could be efficiently generated in a hierarchical way.

Finding appropriate constraints is, however, a hard problem. Although we were able to easily find what constraint to use in the belt-pulley problem, that constraint will not work for other problems. Nevertheless, by observing human manipulating a flexible object, we believe we can learn some ideas about what constraint to use. When we manipulate a flexible object, we usually keep the manipulated object in some sort of *stable* state so that we can easily visually guide the manipulation. For example, when we hook a cord on a nail, we hold the cord at two points with fingers, and move the cord downward by keeping the relative distance between the two hands almost constant. In this operation, the part of the cord between the two held points can be easily visually located. This is because the region where that part can exist is sufficiently constrained by its length and the position of the hands. That part of the cord is considered to be in a stable state. Once we have collected enough constraints to describe a whole assembly sequence in a discrete state space, we could then automatically generate a plan for that assembly process without analytical models of the manipulated flexible objects.

[2] If we have more than two fingers and a rubber belt is initially stable, we can generate a plan composed only of stable states for any problems.

Acknowledgement

The authors would like to thank Dr. Sing Bing Kang for helpful comments on the draft of this paper, and Mr. Masato Kawade for the experimental setup.

This research was conducted while the first author was with Carnegie Mellon University. His stay was supported by the Telecommunications Advancement Foundation, Tokyo, Japan.

References

[1] N. Takahashi et al. Automatic Rubber Belt Assembly Machine. In *Proceedings of the 4th Int. Conf. on Precision Eng.*, 1980.

[2] S. Hirai, H. Wakamatsu, and K. Iwata. Modeling of Deformable Thin Parts for Their Manipulation. In *Proceedings of 1994 IEEE Int. Conf. on Robotics and Automat.*, pages 2955–2960, 1994.

[3] S.A. Hutchinson and A.C. Kak. SPAR: A Planner that Satisfies Operational and Geometric Goals in Uncertain Environments. *AI Journal*, 11(1):30–61, 1990.

[4] H. Inoue and M. Inaba. Hand-Eye Coordination in Rope Handling. In M. Brady and R. Paul, editors, *Robotics Research*, pages 163–174. The MIT Press, 1983.

[5] O. Ozeki, K. Higuchi, and S. Yamamoto. Automated Dimension Inspection System for Automotive Plastic Parts with a Laser Probe. In *Proc. of ROBOTS 12 and Vision '88 Conference*, pages 5–51–5–60, Detroit, MI, 1988.

[6] E.D. Sacerdoti. Planning in Hierarchy of Abstraction Spaces. *Artificial Intelligence*, 5:115–135, 1974.

[7] V. Scheinman. A Multiple Robot Vision Guided Assembly System. In R. Bolles and B. Roth, editors, *Robotics Research 4*. The MIT Press, 1987.

[8] R.D. Schraft, M. Schweizer, and G.E. Fischer. Robotized Assembly of O-Ring Seals: Bin Picking, Handling and Flexible Joining into Different Groove Contours. In *Proceedings of the 18th Int. Symp. on Industrial Robots*, pages 37–52, 1988.

[9] P.M. Taylor. *Sensory Robotics for the Handling of Limp Materials.* Springer-Verlag, 1990.

[10] A. Villarreal and H. Asada. A Geometric Representation of Distributed Compliance for the Assembly of Flexible Parts. In *Proceedings of 1991 IEEE Int. Conf. on Robotics and Automat.*, pages 2708–2715, 1991.

[11] Y.F. Zheng, R. Pei, and C. Chen. Strategies for Automatic Assembly of Deformable Objects. In *Proceedings of 1991 IEEE Int. Conf. on Robotics and Automat.*, pages 2598–2603, 1991.

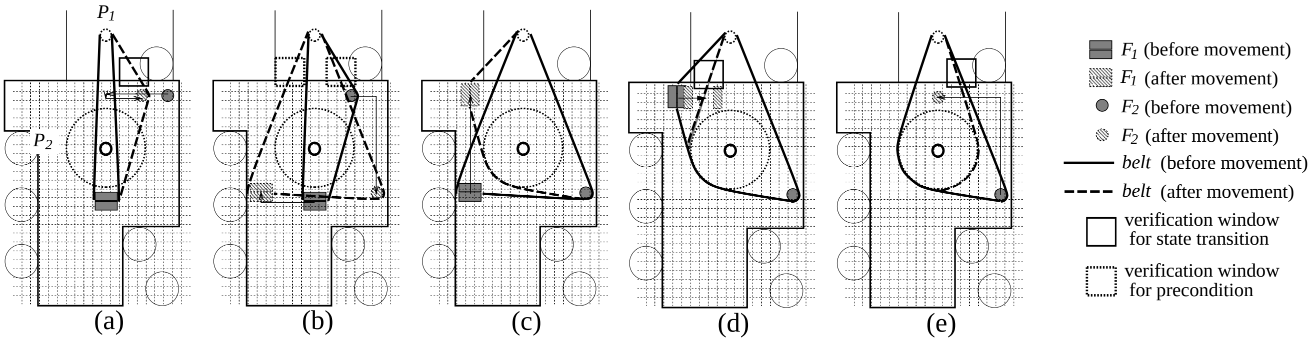
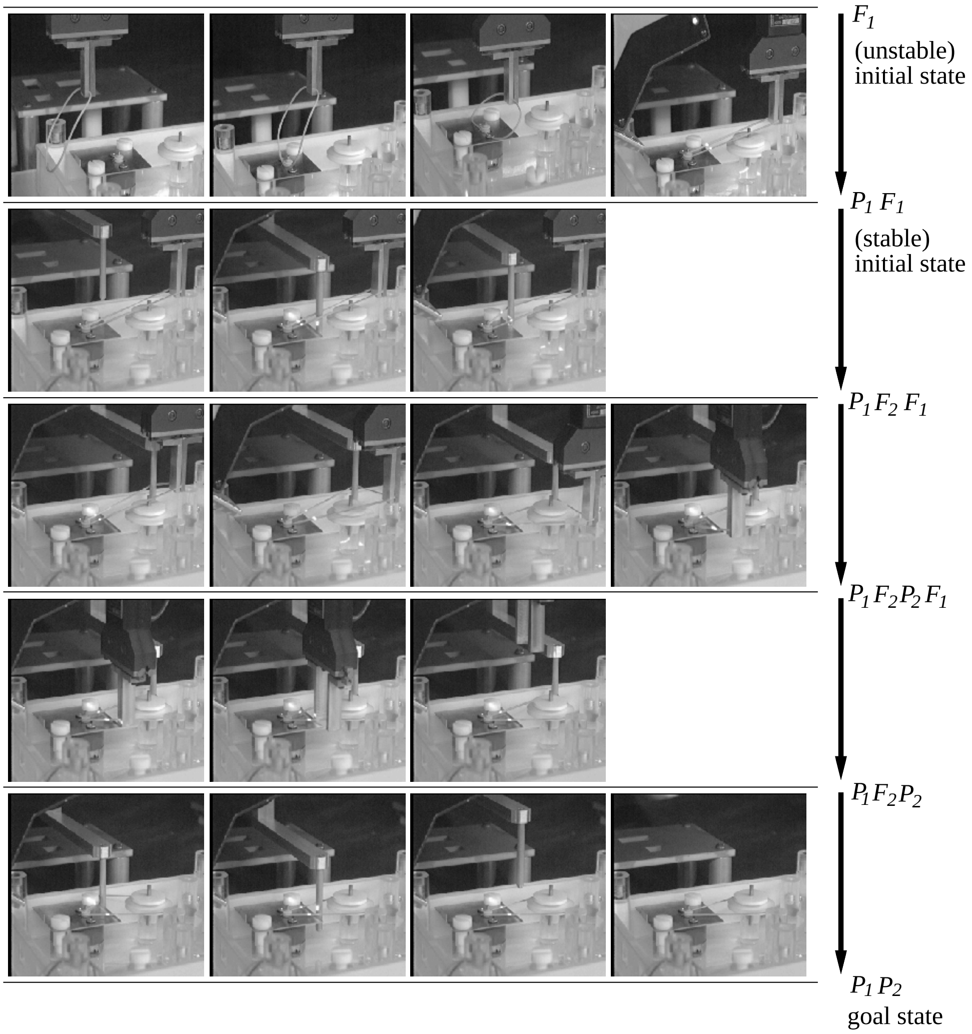


Fig. 18: Final plan: (a) $ADD\text{-}FINGER\text{-}INSIDE(P_1, F_2, F_1)$ is realized by moving F_2, lowering F_2 and moving back F_2. (b) The first part of $ADD\text{-}PULLEY\text{-}INSIDE(F_2, P_2, F_1)$ is realized by moving and lowering F_1 and F_2 so that P_2 enter the inside of the belt. The height of the belt is adjusted by observing two parts of the belt. (c) The second part of $ADD\text{-}PULLEY\text{-}INSIDE(F_2, P_2, F_1)$ is realized by rotating and moving F_1. (d) $REM\text{-}FINGER\text{-}INSIDE(F_1)$ is realized by moving F_1 and opening the parallel gripper. (e) $REM\text{-}FINGER\text{-}INSIDE(F_2)$ is realized by moving F_2.



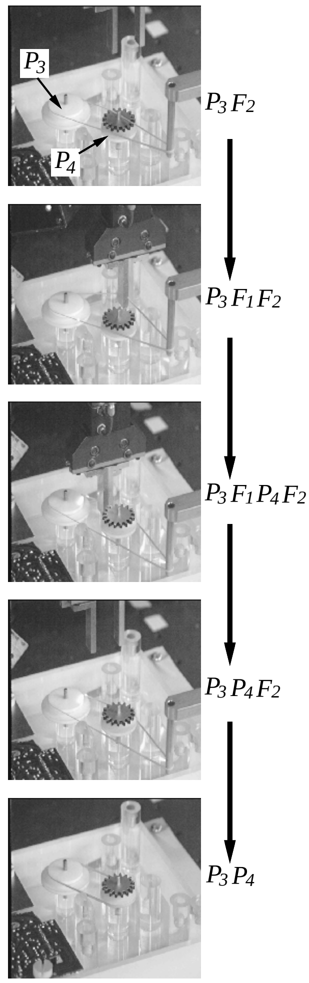


Fig. 19: A successful belt-pulley operation. State transition is explained on the right side.

Fig. 20: Another example.