Task Planning of Assembly of Flexible Objects and Vision-Based Verification

Jun Miura
Department of Computer-Controlled Mechanical Systems, Osaka University,
2-1, Yamadaoka, Suita, Osaka 565, Japan
jun@mech.eng.osaka-u.ac.jp
Katsushi Ikeuchi
Institute of Industrial Science, The University of Tokyo,
7-22-1, Roppongi, Minato, Tokyo 106, Japan

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 vision-guided 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.*

Robotica, Vol. 16, No. 3, pp. 297-307, 1998. in the special issue on "Intelligent Robotic Assembly."

List of Figures

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][2][3] and development of analytical models of flexible objects [4][5][6].

Takahashi et al. [1] developed a machine which can assemble rubber belts with pulleys for cassette tape recorders. Schraft et al. [2] have developed a special tool for O-ring assembly. In this line of approach, new tools, sensors and algorithms are necessary for each task [3]. Development of such task-specific systems requires a significant amount of effort and hardware.

Zheng et al. [4] have analytically developed a model of flexible beam and a control strategy based on that model for achieving the insertion of the beam into a hole. Villarreal and Asada [5] have proposed a concept of buffer zones for efficiently generating assembly strategies for compliant parts based on an analytical model of flexible object. Hirai et al. [6] have proposed a systematic approach to modeling of bendable thin objects such as paper and a sheet metal. In these approaches, planning of manipulation strategies rely mainly upon an analytical model of flexible object. In general, the development of such a model is 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 [7]. 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. The ability to focus attention is essential to realizing efficient and robust sensor-guided manipulation. Inoue and Inaba [8] demonstrated a rope-handling system based on visual feedback. By employing an exploratory visual search in a limited area, the state of a rope was determined. 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 vision-guided 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 an 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 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.

In the problem, we assume a rubber belt is flexible enough in the axial direction to be stretched up to a necessary extent. We set this assumption because of the limitation of available hardware. We can eliminate the assumption if: (1) pulleys are movable and enough number of fingers are available; or (2) fingers have a special mechanism (or skill) which enables fingers to change the grasping point while keeping grasping the belt.

The rest of the paper is organized as follows. Section 2 describes a constraint used in the belt-pulley problem. Using the constraint, we can introduce the notion of "state" into the problem, thereby decomposing the original planning problem into state transition planning and trajectory planning between states, which are solved hierarchically. Section 3 describes the operators and the algorithm used in the high-level planning (state-transition planning). Section 4 describes the low-level planning which actually generates finger trajectories between two states by considering the stability of the belt and the collision possibility. Section 5 explains how to focus visual attention to informative parts of the belt in each state transition. Section 6 describes the experimental setup and the experimental results. Section 7 summarizes the paper and discusses potential applicability of the proposed method to other kinds of sensor-based assembly of flexible objects.

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 *controlled state*. Note that a statically balanced state is not necessarily a controlled state. For example, the state of the belt held by a parallel-jaw gripper in Fig. 1 is not a controlled one. Although the belt is in equilibrium (static balance), the position and the shape of the belt cannot be determined only from the position of the gripper without an exact model of the belt.

The condition that the belt is in a controlled state is represented by

$$\sum_i l_i > l_{thresh}, \tag{1}$$

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 controlled 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 [7][9].

(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 controlled 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 sub-problems:

- 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).

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

The belt-pulley problem may be solved by carefully specifying the movement of fingers. Making an executable finger movement from scratch, however, requires a large amount of effort. Moreover, the generated movement tends to be specific to each problem and may be difficult to be reused for the different configuration of pulleys. In contrast, in our approach, the hierarchical decomposition of the problem using the appropriate constraints reduces the search space at every level of planning, thereby making it easier to automatically generate solutions.

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

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 whether 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 $P_1 F_1 F_2 F_3^*$. 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 (see Fig. 4):

- $ADD\text{-}FINGER\text{-}INSIDE(x, y, z)$: finger x enters the inside of the belt and touches the belt at the position between y and z.
- $ADD\text{-}PULLEY\text{-}INSIDE(x, y, z)$: pulley x enters the inside of the belt and touches the belt at the position between y and z.

- $ADD\text{-}FINGER\text{-}OUTSIDE(x, y, z)$: 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.

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 determined 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 which have the same minimum steps, and then passes them to the low-level planner. If none of them is found to be feasible (executable) 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. Fig. 5 illustrates the process of plan generation, where the two planners are sequentially activated. This process continues until one feasible final plan is generated.

The worst-case computational complexity in generating all candidates of length $2p$ is roughly given by $\mathcal{O}(b^p)$, where b is the average branching factor. Although the actual computation time in the current problem is almost negligible, the computational cost could be high when the minimum length p of the feasible high-level plan is large.

One approach to reducing the complexity is to interleave the high-level and the low-level planner, i.e., to have the high-level planner pass a partial (incomplete) sequence of state transitions to the low-level planner to check its feasibility. This approach can reduce the complexity because many infeasible branches in the search tree are pruned by the low-level planner. However, we have determined to divide the two planners hierarchically for the clarity of the system configuration.

Another approach is the use of macro operators, which are composed of sequences of primitive operators. Macro operators are either prepared beforehand or learned from frequently repeated sequences of operators in the history of solving many planning problems [10].

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. 6. Also, let the initial state and the goal state be P_1F_1 and P_1P_2, respectively (see Fig. 7).

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. 8 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. 9 shows a transition graph representing the second-shortest plans; the number of steps is four; there are twelve candidate plans.

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. The planner generates a *feasible* (executable) trajectory which satisfies the following conditions:

- *collision-free condition*: a finger must not collide with other fingers, pulleys and other objects;
- *entering condition*: when a finger or a pulley enters inside the belt, it must not touch the belt.
- *controlled-state condition*: while a finger is touching the belt, it must move so that the belt is at the current controlled state (i.e., equation (1) is satisfied) at any moment of the movement.

4.1 Representation of Work Space, Free Spaces, and Trajectories

We discretize the 3-D work space horizontally as shown in Fig. 10, for the simplicity of trajectory generation. A point in the 3-D space is specified by a triplet (i, j, z), where (i, j) indicates the position of each discretized block (called $grid$) and z indicates the vertical position in that grid.

To check the *collision-free condition* for a finger, we examine its movable range of height in each grid. Since fingers approach the assembly downwards from the ceiling in our belt-pulley problem, it is enough, for the purpose of collision detection, to record the minimum collision-free height $z_h(i, j)$ of the finger in each grid (i, j). Using this information, we can generate a 2-D map of free spaces for any designated height of the finger.

A *trajectory* of a finger is represented by a series of points traversed. Let $T = \{(i_k, j_k, z_k)|k = 1,\ldots,N\}$ denote a trajectory; N is the number of points including the start and the final point. We currently limit the movement of a finger to those which are composed of movements parallel to the i, j, and z axes. Under this limitation, trajectory T is collision-free if

$$\forall k\,[\,z_k > z_h(i_k, j_k)\,] \tag{2}$$

holds.

4.2 Trajectory Generation

A trajectory generator is prepared for each operator listed in Section 3.2. The state transition indicated by an operator can be decomposed into several consecutive *steps*. For example, ADD-$FINGER$-$INSIDE$ takes the three steps as shown in Fig. 4(a): (1) move a finger above the region formed by the belt; (2) lower the finger below the height of the belt; and (3) pull the belt outward. The trajectory for each step is first generated, and then all partial trajectories are connected to form the final trajectory to realize the given state transition.

In horizontal motions (e.g., steps (1) and (3) of ADD-$FINGER$-$INSIDE$), at the end of each step, a certain configuration of fingers should be achieved which satisfies the corresponding *entering condition* or *controlled-state condition*. A set of satisfactory configuration can be represented by a 2-D region in the i-j space. Thus, such a region is first calculated. Then, the final point of the trajectory is selected in the region. Finally the trajectory connecting the current position and the final position is calculated in the corresponding 2-D free space map. Since the trajectory is composed of straight motions along the axes, a simple trajectory generation strategy is employed which minimizes the number of motions.

For vertical motions (e.g., step (2) of $ADD\text{-}FINGER\text{-}INSIDE$), several predetermined heights are used as goal heights. The rod finger uses two heights: one for touching the belt, and the other for moving horizontally above the belt. The parallel-jaw gripper also uses two heights: one for grabbing the belt, and the other for moving horizontally above the belt. A special treatment is necessary for operators $ADD\text{-}PULLEY\text{-}INSIDE$ and $ADD\text{-}PULLEY\text{-}OUTSIDE$. These operators need to adjust the height of the belt exactly to that of the ditch of the pulley. This height adjustment is performed with visual feedback; its detailed process is described in Section 5.2.

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

4.3 Example

Suppose we are generating a trajectory for $ADD\text{-}FINGER\text{-}INSIDE(F_2, P_1, F_1)$ with the initial state P_1F_1. Fig. 11 shows the state before the transition. Fig. 12 shows the region where F_2 does not collide with other objects (*collision-free condition*) when F_2 is at the height for touching the belt. Fig. 13 shows the regions for F_2 to realize the above-mentioned three steps in this state transition; region *R-1* indicates the region where F_2 can enter the inside of the belt from the above (*entering condition*); region *R-2* indicates the area where the controlled state $P_1F_2F_1$ is realized (*controlled-state condition*). Considering these areas, the final trajectory for this operator is determined as shown in Fig. 14.

5 Visual Verification Planning

Vision is used for the following two purposes:

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

5.1 Verifying Completion of State Transition

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 the added pulley or finger is examined (see Fig. 15(a)). If a pulley or a finger is removed, the part of the belt around the position where the removed pulley or finger was touching is examined (see Fig. 15(b)). The visual verification planner generates a region of interest to be examined; we call this region a *verification window*. The size of verification window is determined in advance from the observation procedure described in Section 6.1. Only the 2-D information is enough for this verification.

Each operator has knowledge of which part of the belt should be observed. We call such a part an *informative part*. The desired position of an informative part can be calculated from the position of fingers and pulleys (because the belt is in a controlled state). The verification window is *feasible* if it can cover the informative part, and if the correpsonding sensor position does not cause the collision between the sensor and other objects. The planner first selects several points on the informative part as candidate positions of the verification window, and then chooses the one which is feasible *and* nearest to the center of the informative part.

If the belt is not in the desired state, the planner considers that the previous operator failed, and retries it.

5.2 Verifying Precondition for State Transition

The state transitions that require the verification of precondition are ADD-$PULLEY$-$INSIDE$ and ADD-$PULLEY$-$OUTSIDE$. 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. 16. For this purpose, the position of the part of the belt to be inserted (called *inserted part*) should be measured. The difference between this verification and the previous one is that not the 2-D but the 3-D position of the inserted part is necessary.

As in the case of verification of state transitions, each operator has knowledge of which part of the belt should be observed (see Fig. 17). If a feasible verification window can be set on the inserted part (*window 1* in Fig. 17), that window is used; otherwise, the position of the inserted part is calculated from the position of the neighbouring parts of the belt (*window 2* and *window 3* in Fig. 17).

At execution time, adjustment of the belt height with visual information is repeated until the precondition shown in Fig. 16 is satisfied.

6 Experiment

6.1 Experimental Setup

Fig. 18 shows the experimental setup. Three overhead modules in the RobotWorld [11] are used for the parallel-jaw gripper (F_1), the rod finger (F_2), and the line laser range finder [12] (see Fig. 19). In the figure, a large-scaled 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 first measured while the range finger moves horizontally as shown in Fig. 20. Then, a line is fitted to the measured points because in controlled states, the portion of the belt between pulleys and fingers is always a straight line. 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. 7 is generated as follows.

The low-level planner first examined the feasibility of two-step plans shown in Fig. 8, and found that neither of two operators $ADD\text{-}PULLEY\text{-}INSIDE(F_1, P_2, P_1)$ and $ADD\text{-}PULLEY\text{-}INSIDE(P_1, P_2, F_1)$ 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. 9 and found four out of twelve candidates for state transition sequences were feasible.

Fig. 21 shows the generated plan of finger movement and visual verification from one of four feasible sequences, $P_1F_1 \to P_1F_2F_1 \to P_1F_2P_2F_1 \to P_1F_2P_2 \to P_1P_2$. Due to the collision possibility, feasible verification windows for the second state transition ($ADD\text{-}PULLEY\text{-}INSIDE(P_1, F_2, P_2, F_1)$) could not been obtained. Instead, we considered that the verification for the next (third) transition ($REM\text{-}FINGER\text{-}INSIDE(F_1)$) also verified the second transition. Fig. 22 shows the actual process of the operation; the belt was successfully set around the two pulleys.

The trajectory in the operation from the initial *not-controlled* state (F_1) to the initial *controlled* state (P_1F_1) was designed manually, although the visual verification for this step was planned by the low-level planner. Note that if we have more than two fingers and a rubber belt is initially in a controlled state, we can generate a plan composed only of controlled states for any problems.

Fig. 23 shows another successful belt-pulley operation.

7 Conclusion and Discussion

This paper has presented a novel approach to vision-guided 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 *controlled state* and by restricting the possible states of the belt to controlled 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 can be applicable to some other assembly processes which involve manipulation of flexible objects. By abstracting the original problem space using appropriate constraints, a feasible plan including sensing plan can be efficiently generated in a hierarchical way.

Finding appropriate constraints is, however, a hard problem. Although we were able to easily find a useful constraint in the belt-pulley problem, that constraint will not work for other problems. Nevertheless, we believe that by observing human manipulating a flexible object, 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 *controlled* 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 controlled state.

Once we have collected enough constraints to describe a whole assembly sequence in a discrete state space, we can then automatically generate a plan at least for that assembly process. A future work is to develop a method to autmatically extract useful constraints for the given task. The Assembly Plan from Observation (APO) paradigm [13], in which the system observes a human performing an assembly task, understands it, and generates a robot program to perform the same task, will be an appropriate framework for this future work.

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. They would also like to thank the members of the Task-Oriented Vision Laboratory, The Robotics Institute, Carnegie Mellon University, for fruitful discussions.

References

[1] N. Takahashi et al, "Automatic Rubber Belt Assembly Machine", In *Proceedings of the 4th Int. Conf. on Precision Eng.*, 896–900 (1980)

[2] 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*, 37–52 (1988)

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

[4] 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.*, 2598–2603 (1991)

[5] 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.*, 2708–2715 (1991)

[6] 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.*, 2955–2960 (1994)

[7] E.D. Sacerdoti, "Planning in Hierarchy of Abstraction Spaces", *Artificial Intelligence*, 5, 115–135 (1974)

[8] H. Inoue and M. Inaba, "Hand-Eye Coordination in Rope Handling", In M. Brady and R. Paul, eds., *Robotics Research*, 163–174 (The MIT Press, 1983)

[9] 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)

[10] R.E. Fikes and N.J. Nilsson, “Learning and Executing Generalized Robot Plans”, *Artificial Intelligence*, 3, 251–288 (1972)

[11] V. Scheinman. “A Multiple Robot Vision Guided Assembly System”, In R. Bolles and B. Roth, eds., *Robotics Research 4*, 23–27, (The MIT Press, 1987)

[12] 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*, 5–51–5–60 (1988)

[13] K. Ikeuchi and T. Suehiro, “Toward an Assembly Plan from Observation Part I: Task Recognition With Polyhedral Objects”, *IEEE Trans. on Robotics and Automat.*, 10(3), 368–385 (1994)

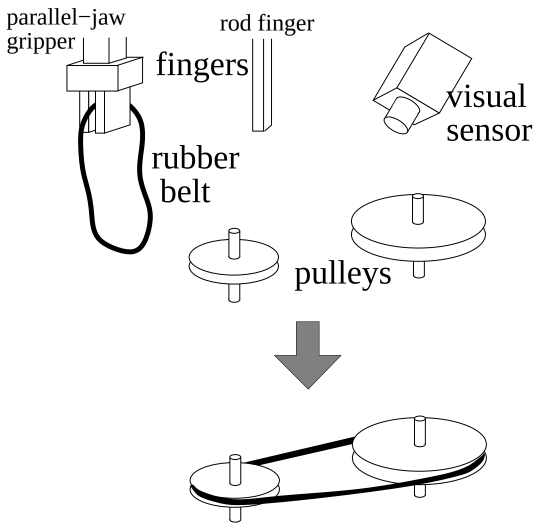


Figure 1: The belt-pulley problem.

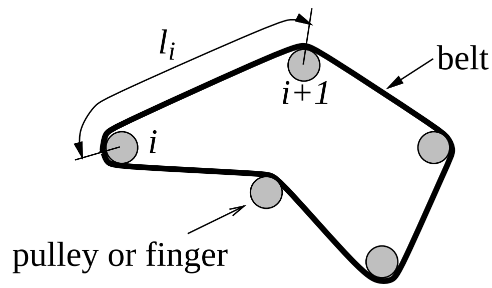


Figure 2: A controlled state.

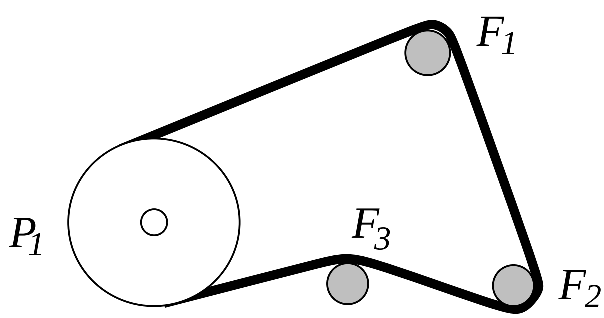
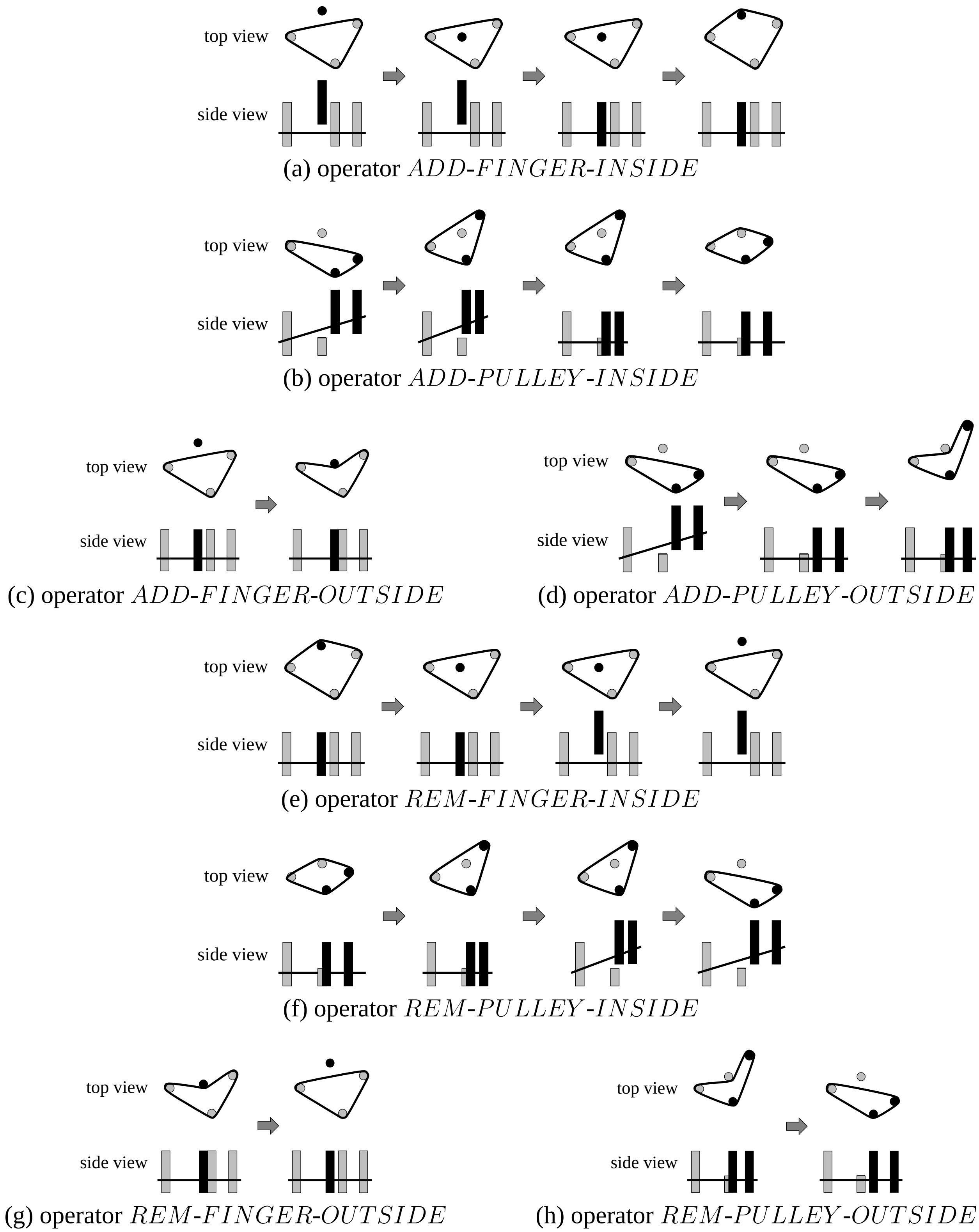


Figure 3: An example state. This state is represented as $P_1F_1F_2F_3^*$.



(a) operator $ADD\text{-}FINGER\text{-}INSIDE$

(b) operator $ADD\text{-}PULLEY\text{-}INSIDE$

(c) operator $ADD\text{-}FINGER\text{-}OUTSIDE$

(d) operator $ADD\text{-}PULLEY\text{-}OUTSIDE$

(e) operator $REM\text{-}FINGER\text{-}INSIDE$

(f) operator $REM\text{-}PULLEY\text{-}INSIDE$

(g) operator $REM\text{-}FINGER\text{-}OUTSIDE$

(h) operator $REM\text{-}PULLEY\text{-}OUTSIDE$

Figure 4: Operators for state transition. Black lines indicate the belt; black figures indicate fingers moved; shaded figures indicate pulleys or stationary fingers.

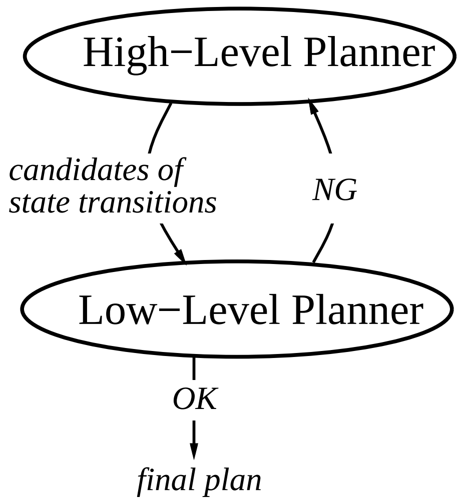


Figure 5: Hierarchical planning.

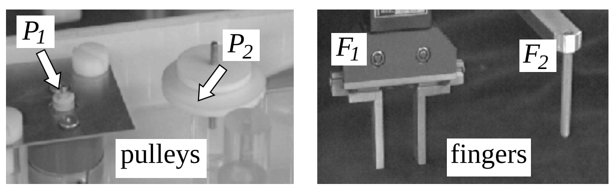


Figure 6: Pulleys and fingers used in an example problem.

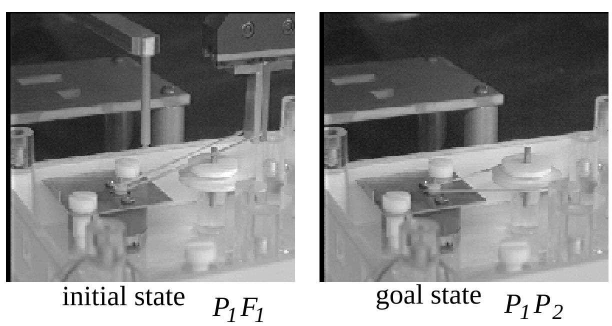


Figure 7: The initial and the goal states.

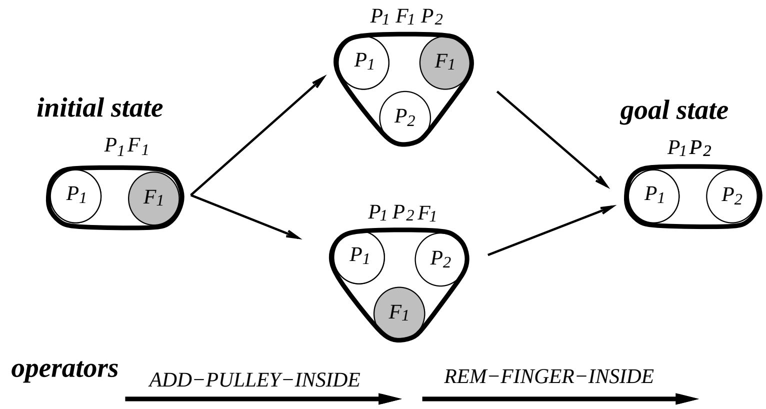


Figure 8: Transition graph for the shortest plans of state transition. Applied operators are indicated at the bottom.

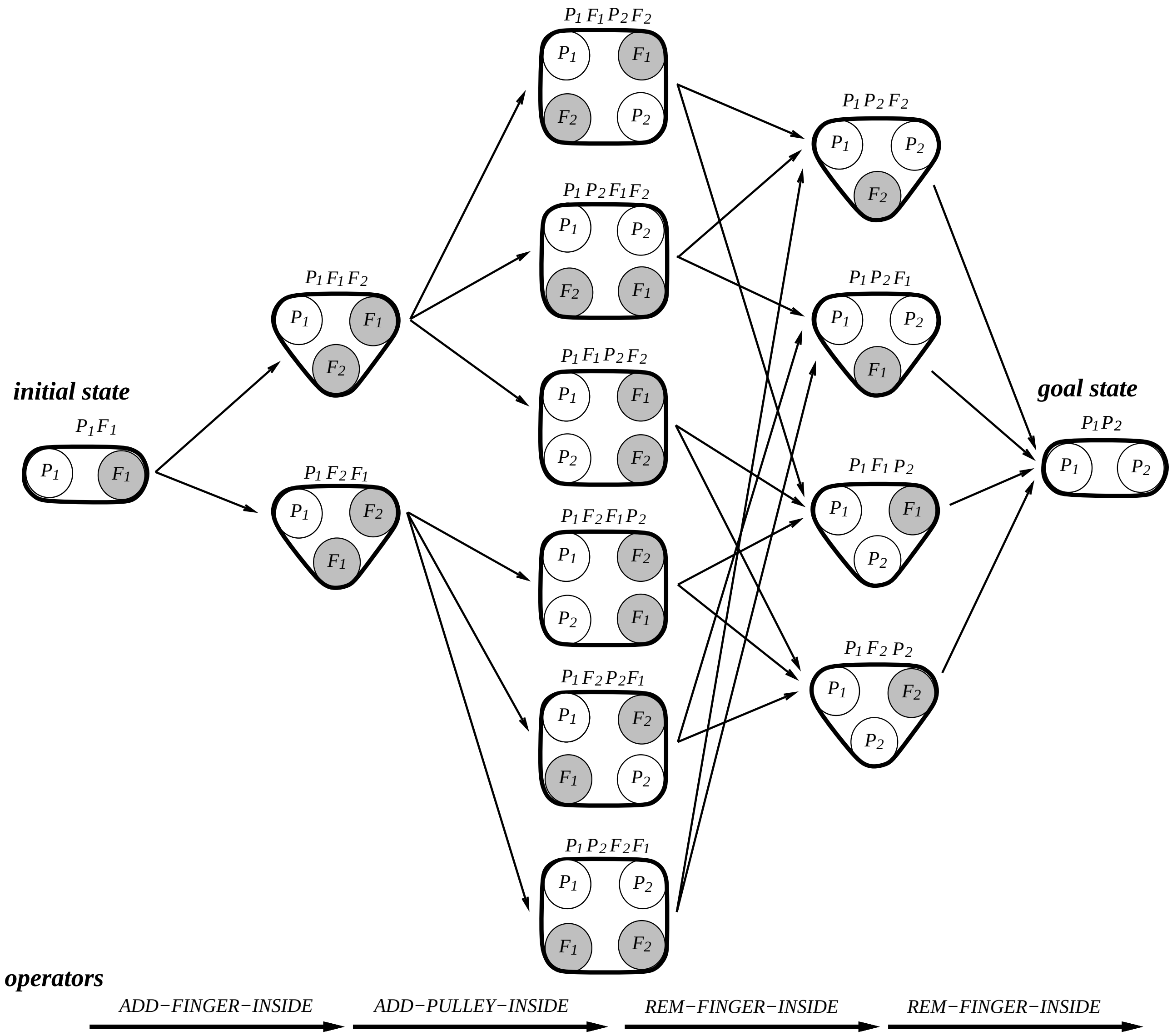


Figure 9: Transition graph for the second-shortest plans of state transition. Applied operators are indicated at the bottom.

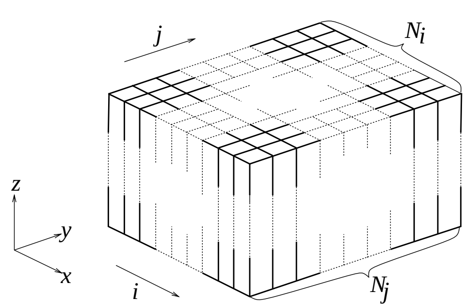


Figure 10: (i, j, z) representation of the 3-D work space.

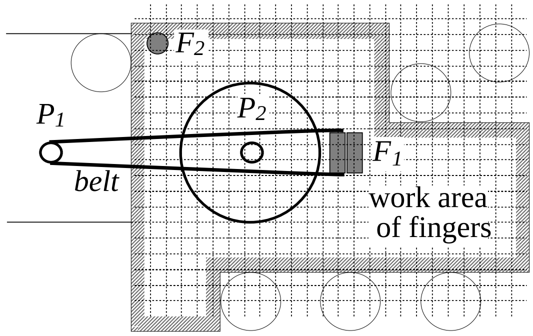


Figure 11: An example state P_1F_1 before the transition. The belt is over the pulley P_2. Predetermined grids are indicated with dotted lines.

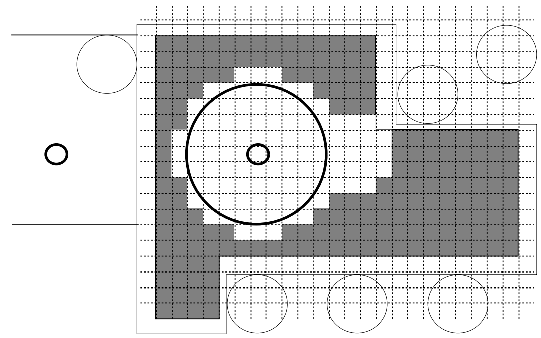


Figure 12: Shaded grids indicate the position where F_2 does not collide with other objects.

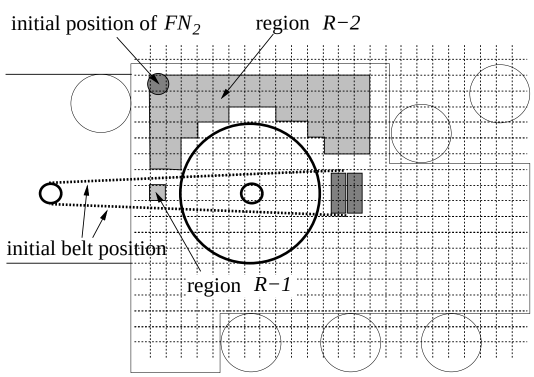


Figure 13: Two regions. *R-1* for inserting the F_2 inside the belt, *R-2* for achieving the controlled state $P_1F_2F_1$.

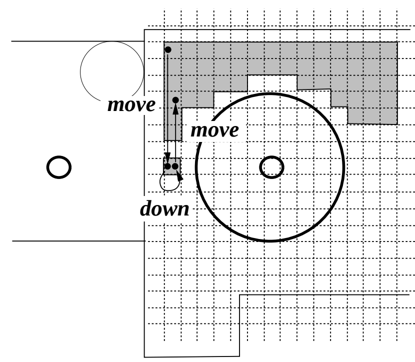


Figure 14: A generated trajectory of F_2 for $ADD\text{-}FINGER\text{-}INSIDE(F_2, P_1, F_1)$.

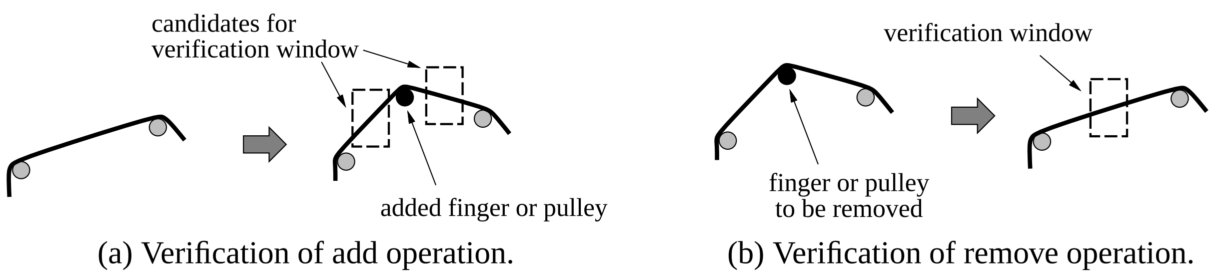


Figure 15: Regions to be examined for verification (verification window).

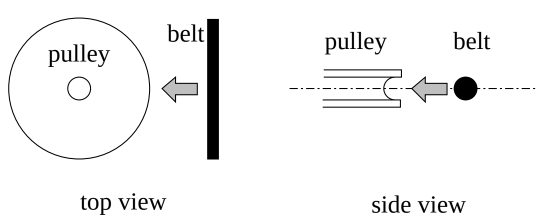


Figure 16: Insertion of the belt into the ditch of a pulley. The height of the belt needs to be adjusted to the height of the pulley before mating.

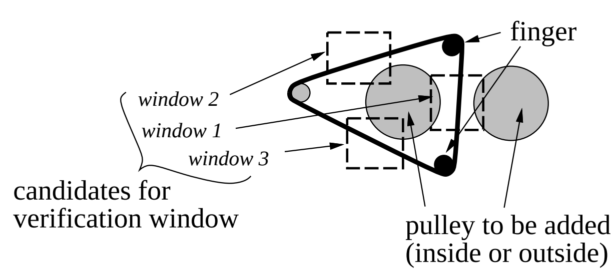


Figure 17: Candidate verification windows for checking precondition.

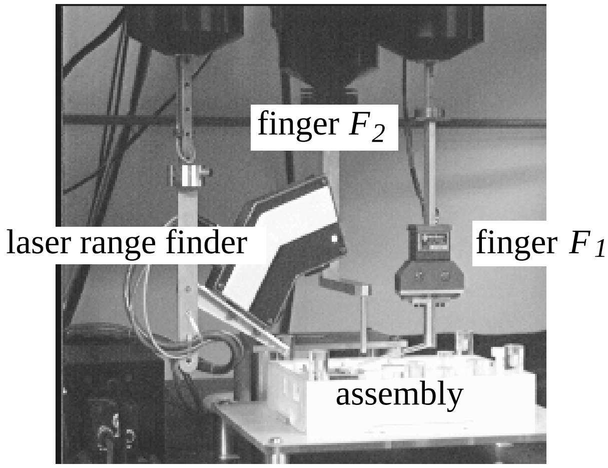


Figure 18: Experimental setup. Three RobotWorld mobile modules are used for two fingers and a laser range finger.

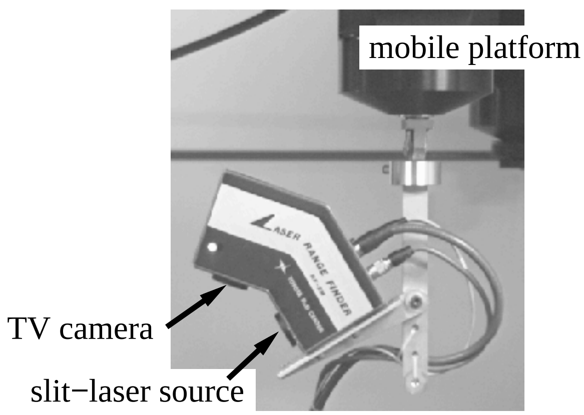


Figure 19: A laser range finder attached to a manipulator.

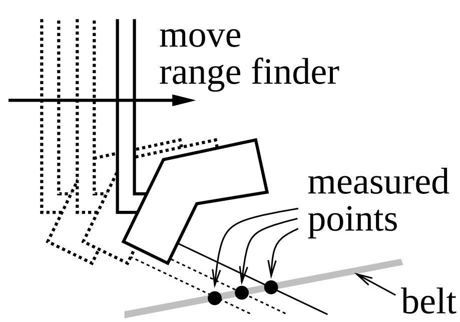


Figure 20: A strategy to observe the belt position.

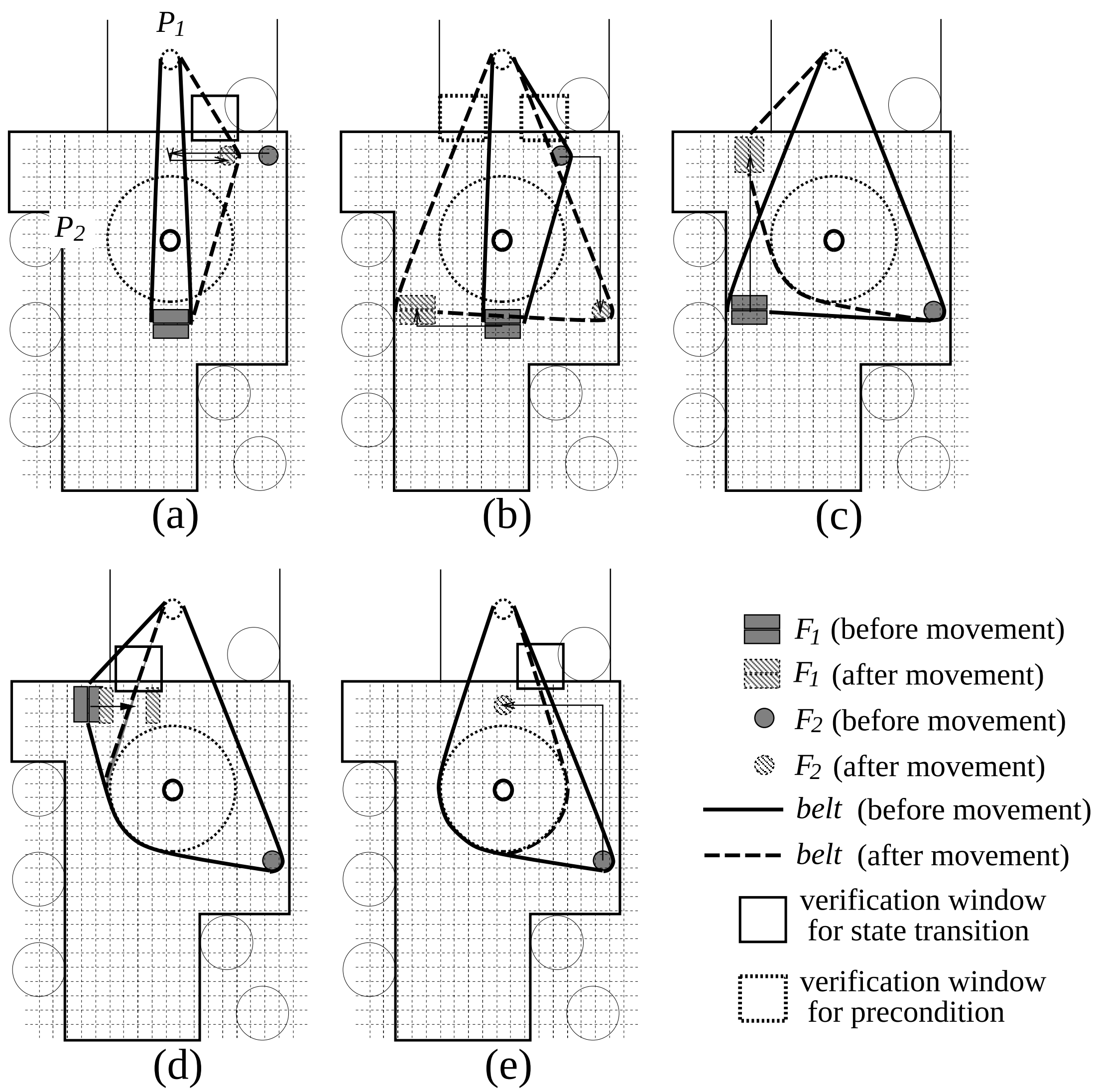


Figure 21: Final plan: (a) $ADD\text{-}FINGER\text{-}INSIDE(P_1, F_2, F_1)$ is realized by moving F_2, lowering F_2 and moving back F_2. (b) The first part of $ADD\text{-}PULLEY\text{-}INSIDE(F_2, P_2, F_1)$ is realized by moving and lowering F_1 and F_2 so that P_2 enter the inside of the belt. The 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.

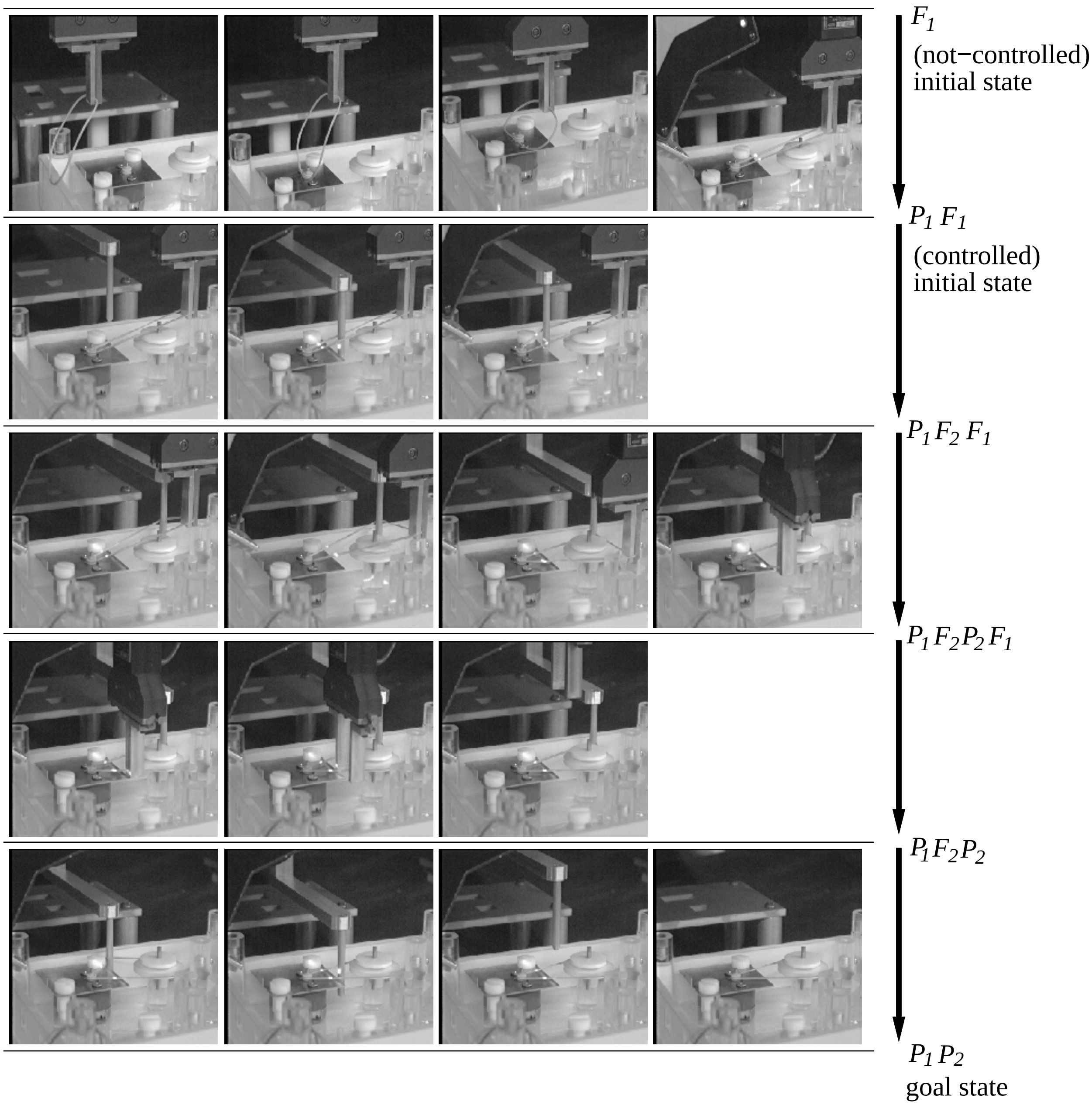


Figure 22: A successful belt-pulley operation. State transition is explained on the right side.

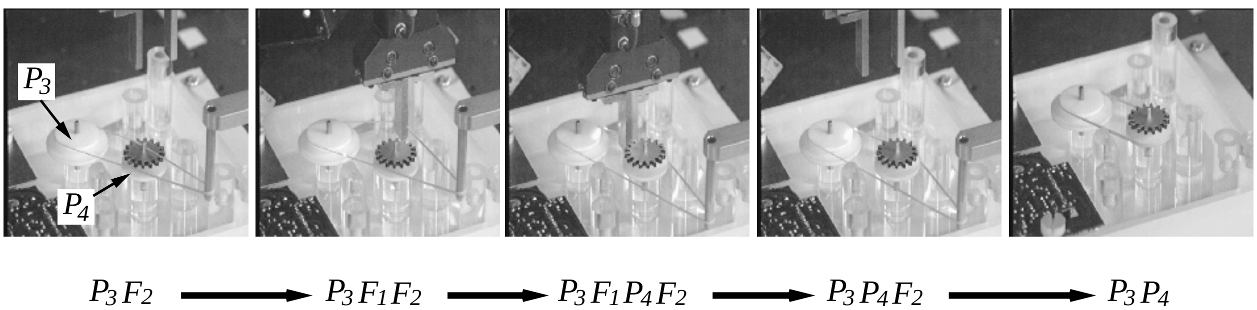


Figure 23: Another successful belt-pulley operation.