2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, November, 2013.

Randomized Planning and Control Strategy for Whole-Arm Manipulation of a Slippery Polygonal Object

Tasuku Yamawaki and Masahito Yashima

***Abstract*— The present paper introduces a planning and control strategy for whole-arm manipulation of a slippery polygonal object. Randomized planning methods are first proposed in order to generate contact state transitions, which help not only to reduce the amount of calculation required, but also to handle a hybrid system composed of a continuous system and a discrete system, which has a large search space and complicated constraints. Second, a novel control strategy, which can switch manipulation modes among quasi-static, dynamic, and caging manipulation depending on the situation, is proposed. This strategy not only can cope with changes in the mechanics of the system caused by contact state transitions, but also can increase the manipulation feasibility and reliability. The validity of the proposed methods is verified through simulations and experiments.**

I. Introduction

Robotic hands are widely used for automated assembly, manufacturing, and packing. Despite promising progress in research on robotic hands, challenges in industrial applications remain. One important task is to rotate objects into different orientations. Traditional fingertip manipulation is suitable for precise local manipulation. However, this requires repeated grasping and releasing of objects for large reorientation and there are problems with respect to reliability because traditional fingertip manipulation is easily affected by uncertainties. In contrast, whole-arm manipulation, which involves wrapping multiple fingers and palms around objects, is expected to perform large reorientation of objects continuously as well as robustly, while allowing the objects to slip on the surface of fingers and palms. However, research has primarily addressed the problem of obtaining stable grasps of objects so that an object cannot slide within the grasp for any external forces applied [4], [19].

The present paper deals with the problem of manipulating polygonal objects with fingers and fixed palms from an initial configuration to a target configuration. Since large reorientation of polygonal objects relative to the fingers and palm is performed, a change in the contact states (i.e., vertex-edge and edge-edge contacts) occurs among all of the bodies in the system. Several problems related to the planning and design of control systems remain. Examples of such problems include (i) whole-arm manipulations, which are regarded as a hybrid system, consisting of a continuous system that expresses motion states and a discrete system that expresses the change in the contact states; and (ii) contact state transitions, which change the kinematics and mechanics of the manipulation system.

T. Yamawaki and M. Yashima are from the Dept. of Mechanical Systems Engineering, National Defense Academy of Japan, 1-10-20, Hashirimizu, Yokosuka, Japan. {yamawaki, yashima} @ nda.ac.jp

A. Related Research

There have been few studies on planning and control of whole-arm manipulation [7], [8], [12], [13], [15], [16], [17]. Song [7], [8] proposed dynamics simulators of whole-arm manipulation, which can deal with inconsistencies in the forward dynamics problem. However, the simulators are limited to local manipulation of smooth objects without contact state transitions in a plane. Yashima [16], [17] proposed a planning algorithm for the global manipulation of 3D smooth objects based on randomized techniques. Trinkle [12], [13] discussed the global manipulation of polygonal objects with contact state transitions. However, a detailed planning algorithm was not presented, which motivates our investigation of these issues.

In an assembly task, polygonal objects are required to be constrained by making contact with various fixed environments in an orderly manner, the concept of which is related to the problem discussed here, which involves contact state transitions. With the goal of automating assembly tasks, algorithms that can generate the contact state graph using Petri nets [3], for example, are proposed. However, these algorithms require a huge amount of calculation.

There have been a number of studies on the conventional manipulation of polygonal objects by robots. Manipulation planning based on a quasi-static model was discussed in [2], [18]. However, the applicability of quasi-static manipulations to various tasks is limited for the case in which equilibrium is not satisfied. In an attempt to remove such limitations, dynamic manipulations were discussed in [1], [9]. Since manipulation performance is greatly affected by modeling errors, dynamic manipulations may not be used as the primary manipulation method. Caging manipulations are proposed in [5], [6], which are limited to a horizontal motion. However, the use of external forces such as gravity has not been discussed.

B. Contribution

The main contributions of the present paper are as follows. First, randomized planning for generating contact state transitions for whole-arm manipulations of a polygonal object in contact with the environment is proposed. The randomized planning method helps not only to reduce the amount of calculation required, but also to handle a hybrid system that has a large search space and complicated constraints.

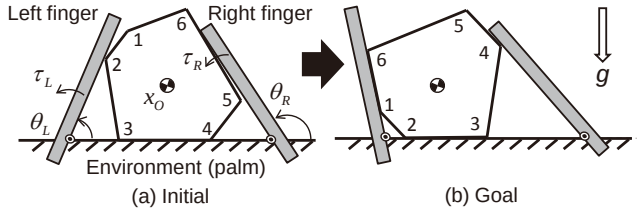

Fig. 1. Initial and goal configurations of whole-arm manipulation

Second, a novel control strategy, which can switch manipulation modes among quasi-static, dynamic, and caging manipulation modes depending on the situation, is proposed. Quasi-static manipulation is the primary manipulation mode. The manipulation mode is switched to dynamic manipulation when the quasi-static manipulation cannot be continued because equilibrium is not satisfied. In particular, caging manipulation is performed when the gravity force applied to the object can be used to realize the rotational motion of the manipulation. This control strategy not only can cope with changes in the mechanics of the system caused by contact state transitions, but also can increase the manipulation feasibility and reliability.

As shown in Fig. 1, the present paper discusses a planning and control strategy for whole-arm manipulation of a polygonal object from the initial configuration to the goal configuration, including contact state transition using a hand system composed of two one-degree-of-freedom fingers and a palm, which is regarded as an environment. Section II presents the quasi-static model and the conditions for the quasi-static manipulation. Section III shows the manipulation planning, which is composed of two phases. The first phase obtains a contact state transition graph, and the second phase generates the desired trajectory by switching manipulation modes. In Section IV, the validity of the proposed methods is demonstrated through simulations and experiments. Finally, we present our conclusions and areas for future research.

II. Quasi-static Model

A. Assumption

In order to simplify the discussion, we assume that 1) the object, fingers, and environment are rigid polygons, 2) the friction between all bodies is negligible, 3) edge-to-edge contacts can be decomposed into two vertex-to-edge contacts, 4) the geometries of all bodies are known, 5) a vertex-to-vertex contact is negligible, 6) the object makes contact with at least one point with each finger and the environment, 7) each joint is velocity controlled or torque controlled, 8) velocity-controlled joints and torque-controlled joints perform non-compliant and compliant motion, respectively.

B. Formulation

The kinematics and statics of the two-fingered hand system are formulated to obtain a quasi-static manipulation model.

Assuming that the object makes contact with the hand system at n frictionless points of contact, the kinematic constraints can be expressed as follows:

$$\begin{bmatrix} \boldsymbol{G}_n^T & -\boldsymbol{J}_n \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{x}}_O \\ \dot{\boldsymbol{\theta}} \end{bmatrix} = \boldsymbol{0} \tag{1}$$

where $\boldsymbol{x}_O \in \Re^3$ is the position and orientation of the object, $\boldsymbol{\theta} \in \Re^2$ is the vector of the joint angles, $\boldsymbol{G}_n \in \Re^{3\times n}$ is the normal contact wrench matrix, and $\boldsymbol{J}_n \in \Re^{n\times 2}$ is the normal Jacobian matrix.

If the object is manipulated quasi-statically by the hand, then the object and the hand system must satisfy the following equilibrium equation at every instant:

$$\begin{bmatrix} \boldsymbol{G}_n \\ -\boldsymbol{J}_n^T \end{bmatrix} \boldsymbol{f}_n = \begin{bmatrix} -\boldsymbol{g}_O \\ \boldsymbol{g}_h - \boldsymbol{\tau} \end{bmatrix} \tag{2}$$

where $\boldsymbol{f}_n \in \Re^n$ is the normal contact force vector, $\boldsymbol{g}_O \in \Re^3$ is the gravity force applied to the object, $\boldsymbol{\tau} \in \Re^2$ is the joint driving torque vector, and $\boldsymbol{g}_h \in \Re^2$ is the joint torque vector induced by gravity.

C. Condition for Quasi-static Manipulation

In order to perform a quasi-static manipulation, the velocity of the object must be uniquely determined for a given velocity of a subset of joints. In addition, the remaining finger is required to apply the joint driving torque to satisfy static equilibrium between the object and the hand system [14].

In order to determine the object's motion uniquely for a given finger motion, (1) should have a solution other than $\dot{\boldsymbol{x}}_O = \boldsymbol{0}$ and $\dot{\boldsymbol{\theta}} = \boldsymbol{0}$. The goal of manipulation is different from that of grasping an object, which requires $\dot{\boldsymbol{x}}_O = \boldsymbol{0}$ in order to fully constrain the motion of an object grasped by fingers. Therefore, the matrix $\begin{bmatrix} \boldsymbol{G}_n^T & -\boldsymbol{J}_n \end{bmatrix}$ in (1) should be full row rank in the case of manipulation.

A square block matrix is extracted from $\begin{bmatrix} \boldsymbol{G}_n^T & -\boldsymbol{J}_n \end{bmatrix}$ by dividing the two fingers into a velocity-controlled finger and a torque-controlled finger. Equation (1) can be rewritten as

$$\begin{bmatrix} \boldsymbol{G}_n^T & -\boldsymbol{J}_{nT} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{x}}_O \\ \dot{\theta}_T \end{bmatrix} = \boldsymbol{J}_{nV} \dot{\theta}_V \tag{3}$$

where $\dot{\theta}_V$ and $\dot{\theta}_T$ are the joint velocities of the velocity-controlled finger and torque-controlled finger, and $\boldsymbol{J}_{nV}$ and $\boldsymbol{J}_{nT}$ are the Jacobian matrices corresponding to $\dot{\theta}_V$ and $\dot{\theta}_T$, respectively.

If the matrix $\begin{bmatrix} \boldsymbol{G}_n^T & -\boldsymbol{J}_{nT} \end{bmatrix} \in \Re^{n\times 4}$ in (3) is nonsingular and the Jacobian matrix $\boldsymbol{J}_{nV}$ is full row rank, then the velocity of the object $\dot{\boldsymbol{x}}_O$ can be determined uniquely for an arbitrary joint velocity $\dot{\theta}_V$ of the velocity-controlled finger.

On the other hand, the hand system is required to maintain a quasi-static equilibrium by pressing the object against the environment and the velocity-controlled finger with the torque-controlled finger. The equilibrium can be expressed by extracting the equations of the object and the torque-controlled finger from (2) as follows:

$$\begin{bmatrix} \boldsymbol{G}_n \\ -\boldsymbol{J}_{nT}^T \end{bmatrix} \boldsymbol{f}_n = \begin{bmatrix} -\boldsymbol{g}_O \\ g_T - \tau_T \end{bmatrix} \tag{4}$$

$$\boldsymbol{f}_n > \boldsymbol{0} \tag{5}$$

where τ_T and g_T are the joint driving torque and the gravity forces of the torque-controlled finger, respectively. The contact force must be compressive, as shown in (5).

If there exists a joint driving torque τ_T of the torque-controlled finger, which satisfies (4) and (5), the hand system can maintain quasi-static equilibrium.

The above formulations for kinematics and statics imply that the joint velocity $\dot{\theta}_V$ of the velocity-controlled finger and the joint torque τ_T of the torque-controlled finger can be regarded as inputs of the system.

In summary, the conditions for the quasi-static manipulation are such that

1 $\begin{bmatrix}\boldsymbol{G}_n^T & -\boldsymbol{J}_{nT}\end{bmatrix}$ is nonsingular.
2 $\boldsymbol{J}_{nV}$ is full row rank.
3 Joint torque τ_T of the torque-controlled finger, which can satisfy (4) and (5), exists.

If the above conditions are satisfied, a compliant manipulation can be performed by pressing the object against the hand system with the torque-controlled finger in accordance with the motion of the velocity-controlled finger.

III. Manipulation Planning

Contact states (i.e., vertex-edge and edge-edge contacts) among all bodies are changed by the large reorientation of a polygonal object relative to fingers and a fixed environment. The proposed manipulation planning for whole-arm manipulation consists of two phases. The first phase obtains a contact state transition graph connecting the initial and goal configuration, and the second phase generates the desired trajectory for the fingers, which can perform the desired transition of the contact state obtained in the first phase.

The system has a total of five degrees of freedom (DOF), consisting of two DOF of the two 1-DOF fingers and three DOF of the object, and thus may be represented by a five-dimensional configuration space. If the object is constrained by the hand system with four contact points ($n = 4$), the dimension of the configuration space can be reduced to one. This means that if the one-dimensional joint velocity $\dot{\theta}_V$ of the velocity-controlled finger is given, the remaining velocity with four DOF, including the velocity of the object, may be determined uniquely. We hereinafter consider the manipulation of an object that contacts the hand at four points.

A. Phase 1: Search for Contact State Transition

1) Subgoal Network: This phase obtains the contact state transition graph connecting the initial and goal configurations using a randomized algorithm. This approach constructs a subgoal network, as shown in Fig. 2, based on the generation of subgoals and the connection between subgoals.

All of the subgoals are classified as either a four-contact subgoal $\boldsymbol{X}$ or a five-contact subgoal $\boldsymbol{Y}$. The initial and goal configurations are denoted by $\boldsymbol{X}_{init}$ and $\boldsymbol{X}_{goal}$, respectively. A pair of four-contact subgoals, $\boldsymbol{X}_{start}$ and $\boldsymbol{X}_{end}$, which are connected by a directed tree, is reachable from $\boldsymbol{X}_{start}$ to $\boldsymbol{X}_{end}$ with the same contact state. However, a subgoal $\boldsymbol{X}_{through}$ is generated when the contact state is changed

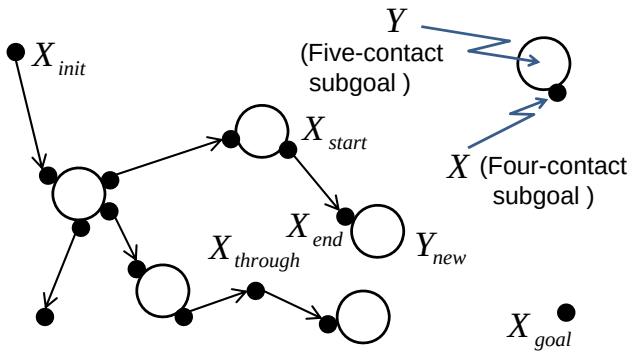

Fig. 2. Generation of subgoals in search space, where $\boldsymbol{X}$ and $\boldsymbol{Y}$ are four-contact and five-contact subgoals, respectively.

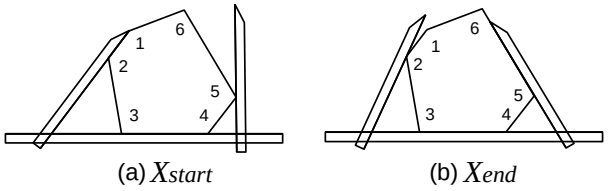

Fig. 3. Illustration of generation of subgoal. By rotating the right finger counterclockwise from a four-contact subgoal $\boldsymbol{X}_{start}$ while maintaining the current contact state, a new four-contact subgoal $\boldsymbol{X}_{end}$ is generated.

along the way from $\boldsymbol{X}_{start}$ to $\boldsymbol{X}_{end}$. The four-contact subgoals attached to the identical five-contact subgoal have the same configuration but have different contact states.

Fig. 3 illustrates the generation of $\boldsymbol{X}_{start}$ and $\boldsymbol{X}_{end}$. Suppose that $\boldsymbol{X}_{start}$ is generated by selecting four contact points at vertexes #2–5 among the five existing contact points and assigning counterclockwise rotation to the velocity-controlled right finger. The object slides toward the left along the environment by moving the right finger while maintaining the current contact state. When the object gains a new edge-edge contact with the right finger with five contact points, a four-contact subgoal $\boldsymbol{X}_{end}$ which has the same contact state as $\boldsymbol{X}_{start}$ is generated. This process is repeated until reaching $\boldsymbol{X}_{goal}$.

The proposed algorithm uses randomized methods for extending subgoals, which is very effective for motion planning with a large dimensional search space [16]. The randomized methods may be suitable for the case of whole-arm manipulation because transitions of the contact state occur frequently and a number of subgoals are generated.

2) Data Structure: The subgoals $\boldsymbol{X}$ and $\boldsymbol{Y}$ are described by structured data type. Each subgoal consists of structure members, such as the position and orientation of the object, $\boldsymbol{x}_O$, the joint angles of the fingers, $\boldsymbol{\theta}$, and the contact state matrix $\boldsymbol{C}$, which is used to qualitatively describe the contact state between all bodies. In addition, $\boldsymbol{X}$ has the rotational directions of the velocity-controlled finger, dir_joint, and its parent subgoal number as structure members.

3) Algorithm of Randomized Planning: The randomized algorithm is described in detail by the

MAIN_RANDOM_SEARCH function shown below. Either a velocity-control mode or a torque-control mode is assigned to each finger temporarily in this phase.

(Steps 1 and 5) Iterate this process until completing a subgoal network connecting the initial and goal configuration or until reaching the maximum number of iterations.

(Step 2) A subgoal $\boldsymbol{Y}_{rand}$ is selected at random from among the existent five-contact subgoals, including the initial subgoal.

(Step 3) A four-contact subgoal $\boldsymbol{X}_{start}$ is generated by selecting four contact points from the contact points of $\boldsymbol{Y}_{rand}$ at random without overlapping with the existing four-contact subgoals attached to $\boldsymbol{Y}_{rand}$ if a five-contact subgoal is selected in step 1. Moreover, the rotational direction of the velocity-controlled finger, dir_joint, is determined randomly.

(Step 4) The velocity-controlled finger is moved in the direction of dir_joint with a joint velocity input $\dot{\theta}_V$ from the configuration of $\boldsymbol{X}_{start}$ to generate new subgoals. If the object is moved to the configuration of the object of $\boldsymbol{X}_{goal}$, the search is terminated. The NEW_SUBGOAL function is described in detail in the next section.

MAIN_RANDOM_SEARCH
1 **for** $i = 1$ to i_{max} **do**
2 $\quad \boldsymbol{Y}_{rand} \leftarrow$ RANDOM_SUBGOAL_Y();
3 $\quad (\boldsymbol{X}_{start}, dir\_joint) \leftarrow$ RANDOM_SUBGOAL_X($\boldsymbol{Y}_{rand}$);
4 $\quad$ **if** (NEW_SUBGOAL($\boldsymbol{X}_{start}, \dot{\theta}_V, dir\_joint$) = *Reached*) **break;**
5 **end for**

4) Generation of New Subgoals: The NEW_SUBGOAL function, which generates new subgoals $\boldsymbol{X}$ and $\boldsymbol{Y}$ by moving the velocity-controlled finger in the direction of dir_joint, is described in detail below.

(Step 1) Set the system configuration, $\mathcal{S} = (\boldsymbol{x}_O, \boldsymbol{\theta}, \boldsymbol{C})$, of a starting subgoal $\boldsymbol{X}$.

(Step 2) Check whether the hand system can satisfy the equilibrium condition by solving (4) and (5).

(Step 3) The forward kinematics problem (3) is solved by giving the joint velocity input $\dot{\theta}_V$ of the velocity-controlled finger in the direction of dir_joint in order to obtain the velocity of the object $\dot{\boldsymbol{x}}_O$ and the remaining joint velocity $\dot{\theta}_T$. The joint velocity $\dot{\theta}_V$ with a constant value may be given in this phase.

(Step 4) Check whether caging manipulation, which depends on the current object's orientation and rotational direction, is feasible. The caging manipulation is described in detail in Section III-B.3.

(Step 5) If neither equilibrium nor caging manipulation are feasible, then no subgoals are generated and return to the main program, otherwise go to the next step.

(Step 6) A new configuration $\mathcal{S}$ is obtained by integrating $\dot{\boldsymbol{x}}_O$ with respect to time Δt.

(Step 7) If the current configuration of object, $\boldsymbol{x}_O$, is equivalent to that of $\boldsymbol{X}_{goal}$, then return to the main program to terminate the search, otherwise go to the next step to generate new subgoals.

(Step 8) If the number of contact points becomes five, then new subgoals $\boldsymbol{Y}_{new}$ and $\boldsymbol{X}_{end}$ are generated. Return to the main program.

(Step 9) If the number of contact points is four and the contact state is changed, then a new four-contact subgoal $\boldsymbol{X}_{through}$ is generated. Return to step 2 to restart the search from $\boldsymbol{X}_{through}$ without changing the rotational direction of the velocity-controlled finger.

(Step 10) If the number of contact points is four and the contact state is not changed, then return to step 2 to find a new subgoal.

NEW_SUBGOAL $(\boldsymbol{X}, \dot{\theta}_V, dir\_joint)$
1 $\mathcal{S} = (\boldsymbol{x}_O, \boldsymbol{\theta}, \boldsymbol{C}) \leftarrow$ SetConfig $(\boldsymbol{X})$;
2 EquilFlag $\leftarrow$ Check_Equilibrium $(\mathcal{S})$;
3 $(\dot{\boldsymbol{x}}_O, \dot{\theta}_T) \leftarrow$ FwdKinematics $(\dot{\theta}_V, dir\_joint)$;
4 CagingFlag $\leftarrow$ Check_Caging $(\boldsymbol{x}_O, \dot{\boldsymbol{x}}_O)$;
5 **if** (EquilFlag = False **and** CagingFlag = False) **return** *Unreached*;
6 $\mathcal{S} = (\boldsymbol{x}_O, \boldsymbol{\theta}, \boldsymbol{C}) \leftarrow$ Get_NewConfig $(\dot{\boldsymbol{x}}_O, \Delta t)$;
7 **if** $(\boldsymbol{x}_O = \boldsymbol{X}_{goal}.\boldsymbol{x}_O)$ **return** *Reached*;
8 **else if** (five contacts) $(\boldsymbol{Y}_{new}, \boldsymbol{X}_{end}) \leftarrow$ Generate_Y$(\mathcal{S})$; **return** *Unreached*;
9 **else if** (four contacts **and** contact state changed) $\boldsymbol{X}_{through} \leftarrow$ Generate_Xthrough$(\mathcal{S})$; go to step 2;
10 **else if** (four contacts **and** contact state unchanged) go to step 2;

B. Phase 2: Trajectory Generation

This phase generates the desired trajectories, which can perform the desired transition of the contact state obtained in phase 1. Three manipulations are applied: a quasi-static manipulation, a dynamic manipulation, and a caging manipulation.

1) Quasi-Static Manipulation: A quasi-static manipulation is the primary manipulation. The joint torque of the torque-controlled finger for a quasi-static manipulation is derived. First, the maximum and minimum joint torques, τ_{max} and τ_{min}, which can satisfy the equilibrium conditions (4) and (5), are obtained by solving the following quadratic programming problem:

$$\begin{aligned} \text{maximize} \quad & \tau_T^2 \quad (\text{or} \quad \text{minimize} \quad \tau_T^2) \\ \text{subj. to} \quad & \text{Eqs. (4) and (5)} \\ & -\tau_{limit} \le \tau_T \le \tau_{limit} \end{aligned} \tag{6}$$

where τ_{limit} is the limit torque of the actuators.

Any joint torque between τ_{max} and τ_{min} is applicable for the quasi-static manipulation. As shown in Fig. 4, the applicable torque range is described by the hatched region bounded by τ_{max} and τ_{min}. The desired joint torque τ is obtained by adding a margin torque τ_{margin} to the minimum joint torque in view of modeling errors.

The conditions for the feasibility of the contact state transition with a quasi-static manipulation are as follows:

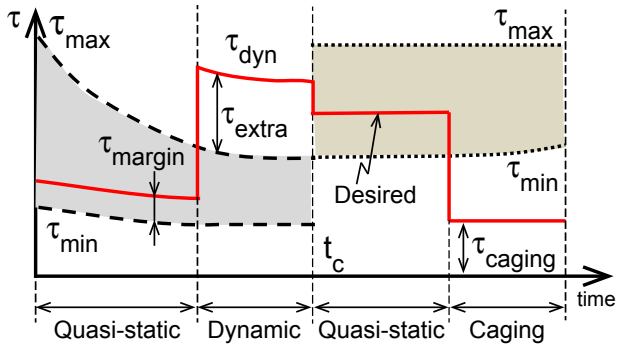

Fig. 4. Desired torque trajectory generated based on three manipulation modes: quasi-static manipulation, dynamic manipulation, and caging manipulation.

1) There exists a joint velocity of the velocity-controlled finger that makes it possible to uniquely determine the velocities of an object just before and after contact state transition.
2) There exists a joint torque of the torque-controlled finger that makes it possible to simultaneously satisfy the equilibrium states just before and after contact state transition.

Condition 1) should be satisfied if the contact state transition is obtained in phase 1 because the forward kinematics problem (3) is solved. In contrast, condition 2) is not always satisfied. This situation occurs at $t=t_c$ in Fig. 4, where the applicable torque range is separated.

2) Dynamic Manipulation: When the contact state transition with a quasi-static manipulation is not feasible, a dynamic manipulation should be applied. As shown in Fig. 4, a joint torque τ_{dyn} that exceeds the maximum joint torque by an extra torque τ_{extra} is applied before $t=t_c$ in order to perform the dynamic manipulation by breaking equilibrium. In order to verify whether applying τ_{dyn} causes the contact state transition dynamically, the forward dynamics problem is solved, which can be described as a linear complementarity problem formulated by combining the motion equations of the object and the torque-controlled finger, as well as the acceleration kinematics. This approach is widely used to simulate contact phenomena in many fields, including robotics and mechanics [10], [11]. The forward dynamics problem is not described in detail here due to its length.

3) Caging Manipulation: A caging manipulation, in which the object is caged by the fingers, is used to rotate the object around the vertex contacting the environment through gravity. The advantage of this manipulation is that precise force control is not needed because the fingers only have to be position-controlled in order to achieve a caging configuration. After the center of gravity of the object crosses the contact normal of the vertex during the rotation, the manipulation mode is switched from quasi-static manipulation to caging manipulation. The object is rotated through gravity until another edge of the object comes into contact with the environment. To stop the object stably, it is assumed that the center of gravity of the object is located between the two normals of the ends of the edge contacting the environment. The constant torque τ_{caging}, which is less than the minimum joint torque, is applied in order that the torque-controlled finger may not prevent the object from moving, as shown in Fig. 4.

4) Control Mode: Either a velocity-control mode or a torque-control mode is assigned to each finger. The geometric constraint is imposed on the object not only by the environment but also by the velocity-controlled finger. The velocity-controlled finger may cause jamming due to position errors in actual experiments as the number of contact points between the velocity-controlled finger and the object increases. It is required to properly assign the velocity-control mode to either finger depending on a current system configuration.

The control modes are assigned based on the following steps. First, calculate the degree of constraint (DOC) [18] of the object imposed by the velocity-controlled finger and the environment of a current system configuration for two cases in which the velocity control mode is assigned to either the right finger or the left finger. And prefer the one whose DOC is smaller. This method is not described in detail here due to its length.

5) Algorithm of Trajectory Generation: The algorithm of the trajectory generation is described in detail by the MAIN_TRAJECTORY function shown below. Before starting the algorithm, control modes are assigned to each finger, as described above.

(Step 1) Read a series of subgoals connecting the initial and goal configurations obtained in phase 1.

(Steps 2 and 3) Set subgoals $\boldsymbol{X}_A$ and $\boldsymbol{X}_B$, which correspond to both ends of the trajectory of the velocity-controlled finger, where $\boldsymbol{X}_B$ is assigned a subgoal just before the rotational direction of the velocity-controlled finger is reversed, or is assigned $\boldsymbol{X}_{goal}$ if the rotational direction is not reversed between $\boldsymbol{X}_A$ and $\boldsymbol{X}_{goal}$.

(Step 4) Check whether the condition for quasi-static transition of the contact state mentioned in Section III-B.1 can be satisfied between $\boldsymbol{X}_A$ and $\boldsymbol{X}_B$.

(Step 5) The trajectory $\theta_V(t)$ of the velocity-controlled finger during the segment connecting $\boldsymbol{X}_A$ and $\boldsymbol{X}_B$ is calculated. The velocity and acceleration constraints at both ends are zero, and the duration of this segment is ΔT.

(Step 6) The forward kinematics problem (3) is solved by giving the derived joint velocity $\dot{\theta}_V(t)$ in order to obtain the velocity of the object $\dot{\boldsymbol{x}}_O(t)$ and the remaining joint velocity $\dot{\theta}_T(t)$.

(Step 7) The joint torque $\tau(t)$ of the torque-controlled finger is calculated based on the current manipulation mode, the details of which are presented in the next section.

(Step 8) A new configuration, $\mathcal{S}(t)=(\boldsymbol{x}_O(t),\boldsymbol{\theta}(t),\boldsymbol{C}(t))$, is calculated by integrating $\dot{\boldsymbol{x}}_O$ with respect to time Δt.

(Step 9) If the joint angle $\theta_V(t)$ reaches the target angle of the current segment, then the initial subgoal of the next segment is given and go to step 3.

(Steps 10 and 11) In addition, if the connected subgoal $\boldsymbol{X}_B$ is equivalent to $\boldsymbol{X}_{goal}$, then planning is terminated, otherwise go to step 5 to calculate trajectories at time $t+\Delta t$.

MAIN_TRAJECTORY

1 $\mathcal{X} = (\boldsymbol{X}_{init}, \cdots, \boldsymbol{X}_{goal}) \leftarrow$ Read_Subgoal();
2 $\boldsymbol{X}_A \leftarrow \boldsymbol{X}_{init}$;
3 $\boldsymbol{X}_B \leftarrow$ Set_BoundarySubgoal $(\boldsymbol{X}_A)$;
4 untransitFlag $\leftarrow$ Check_Transition$(\boldsymbol{X}_A, \boldsymbol{X}_B)$;
5 $(\theta_V(t),\ \dot{\theta}_V(t))$
$\leftarrow$ Make_JointTraj $(\boldsymbol{X}_A, \boldsymbol{X}_B, \Delta T, t)$;
6 $(\dot{\boldsymbol{x}}_O(t), \dot{\theta}_T(t)) \leftarrow$ FwdKinematics$(\dot{\theta}_V(t))$;
7 $\tau(t) \leftarrow$ GET_TORQUE (untransitFlag);
8 $\mathcal{S}(t) \leftarrow$ Get_NewConfig $(\dot{\boldsymbol{x}}_O, \Delta t)$;
9 **if** $(\theta_V(t) = \boldsymbol{X}_B.\theta_V$ **and** $\boldsymbol{X}_B \neq \boldsymbol{X}_{goal})$
$\boldsymbol{X}_A \leftarrow$next subgoal of $\boldsymbol{X}_B$;
go to step 3;
10 **else if** $(\theta_V(t) = \boldsymbol{X}_B.\theta_V$ **and** $\boldsymbol{X}_B = \boldsymbol{X}_{goal})$
Stop;
11 **else** $t = t + \Delta t$; go to step 5;

6) Derivation of Joint Torque: The GET_TORQUE function, which generates the desired joint torque for the torque-controlled finger, is described in detail below.

(Step 1) The joint torque for a quasi-static manipulation is derived.

(Steps 2 and 3) If caging manipulation can be applied, then the manipulation mode is switched from quasi-static manipulation to caging manipulation. The constant joint torque τ_{caging} is applied.

(Step 4) If the transition between contact states is not feasible with a quasi-static manipulation, a dynamic manipulation is performed. The joint torque $\tau_{dyn}(t)$ for the dynamic manipulation is calculated and applied to the torque-controlled finger before the object arrives at the configuration in which the contact state transition is not feasible. Whether the application of $\tau_{dyn}(t)$ generates the desired contact state transition is then verified.

(Step 5) Return the calculated joint torque $\tau(t)$.

GET_TORQUE (untransitFlag)
1 $\tau(t) \leftarrow$ Get_TorqueQuasiStatic (τ_{margin});
2 CagingFlag $\leftarrow$ Check_Caging $(\boldsymbol{x}_O(t),\ \dot{\boldsymbol{x}}_O(t))$;
3 **if** (CagingFlag = TRUE) $\tau(t) \leftarrow \tau_{caging}$;
4 **else if** (untransitFlag = TRUE **and**
near untransitable configuration)
$\tau_{dyn}(t) \leftarrow$ Get_TorqueDyn (τ_{extra});
DynFlag $\leftarrow$ Check_Dyn $(\tau_{dyn}(t))$;
if (DynFlag = TRUE) $\tau(t) = \tau_{dyn}(t)$;
5 **return** $(\tau(t))$;

IV. Simulations and Experiments

We conducted simulations and experiments on the whole-arm manipulation shown in Fig. 1.

A. Two-fingered Robotic Hand System

The two-fingered robotic hand system shown in Fig. 5 was developed for the whole-arm manipulation. Two aluminum fingers 15 cm in length were actuated by direct-drive motors with encoders. The length between the joint axes of the fingers was 20 cm. A support plane, which was tilted by 15 degrees from the vertical plane, was used to restrict the motion of the object to a planar motion caused by gravity. The friction between the object and the support plane was negligible because the support plane was coated with Teflon®. The asymmetric hexagonal object was made of plastic resin and had a mass of 353 g. There were no sensors to measure the position and orientation of the object.

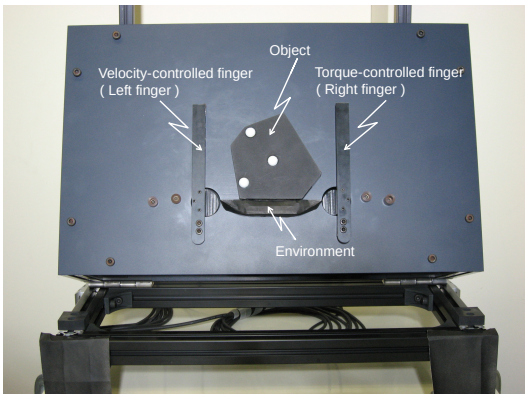

Fig. 5. Two-fingered robotic hand system developed for whole-arm manipulation

B. Results of the Search for the Contact State Transition (Phase 1)

By applying the proposed randomized algorithm, we obtain the contact state transition graph from the initial configuration to the goal configuration, as shown in Fig. 6. A velocity-control mode is assigned to the right finger temporarily in this phase.

Each figure shows the configuration of either $\boldsymbol{X}_{start}$ or $\boldsymbol{X}_{through}$. The subgoal number and the rotational direction of the right finger are shown at the top left and top right corners, respectively. The bold lines at the contact points indicate the contact normals. The reachable subgoals are connected by solid lines.

The subgoals #1 and #2 correspond to the initial configuration $\boldsymbol{X}_{init}$ and are distinguished as different subgoals because the rotational direction of the right finger is different. Subgoal #54 corresponds to the final configuration $\boldsymbol{X}_{goal}$. A total of 54 subgoals are generated randomly to connect $\boldsymbol{X}_{init}$ and $\boldsymbol{X}_{goal}$. Backtracking from #54 to #2 yields the desired transition of the contact state, which is indicated by the thick red lines between subgoals #2, 24, 29, 30, 36, 37, 43–45, 53, and 54.

C. Results of Trajectory Generation (Phase 2)

The control mode for each finger is assigned based on the concept mentioned in Section III-B.4 by taking into consideration the contact state obtained in phase 1. In this example, the torque control mode is applied to the right finger, and the velocity control mode is applied to the left finger.

Fig. 7(a) shows the obtained torque trajectory of the torque-controlled finger and the manipulation mode. Let the parameters be $\tau_{margin} = 3.5$ Nm, $\tau_{caging} = 0.5$ Nm, and $\tau_{extra} = 2.0$ Nm. The hatched region bounded by the maximum and minimum torque shows the joint torque, which

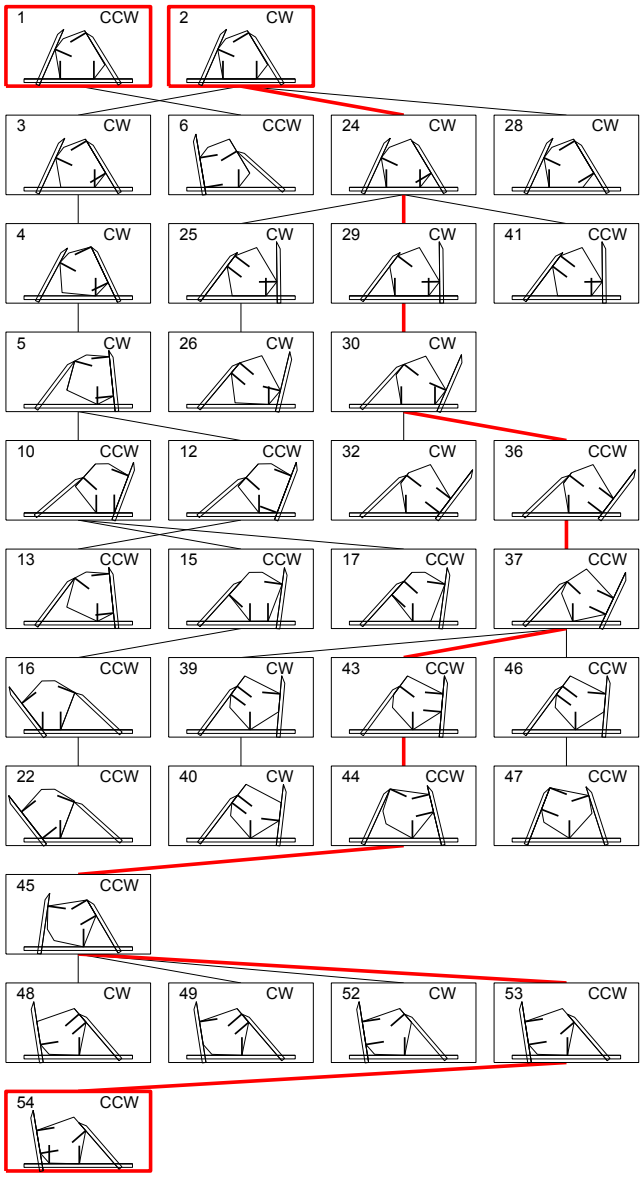

Fig. 6. Contact state transition graph obtained in phase 1. Transitions from $\boldsymbol{X}_{init}$ to $\boldsymbol{X}_{goal}$ are indicated by thick red lines. The rotational directions of the right finger are shown at the top right. The bold lines at the contact points indicate the contact normals.

is applicable to quasi-static manipulation. The grasp has force closure when the maximum torque reaches the limit of the actuator torque $\tau_{limit} = 7.0$ Nm, otherwise the grasp is in equilibrium. The solid red line is the desired torque trajectory τ_R obtained in this phase. Fig. 7(b) shows the joint angle trajectory of the velocity-controlled finger, which is obtained using a fifth-order polynomial with respect to time.

As shown in Fig. 7(a), the contact state transition with the quasi-static manipulation is impossible at $t = 0.65$ s because condition 2) in Section III-B.2 is not met. At that time, the lower right edge of the object makes contact with the link of the right finger (see $\#36$ in Fig. 6). In order to avoid this configuration, a dynamic manipulation is performed between $t = 0.51$ and 0.65 s by applying a joint torque τ_{dyn}, which exceeds the maximum joint torque. The caging manipulation

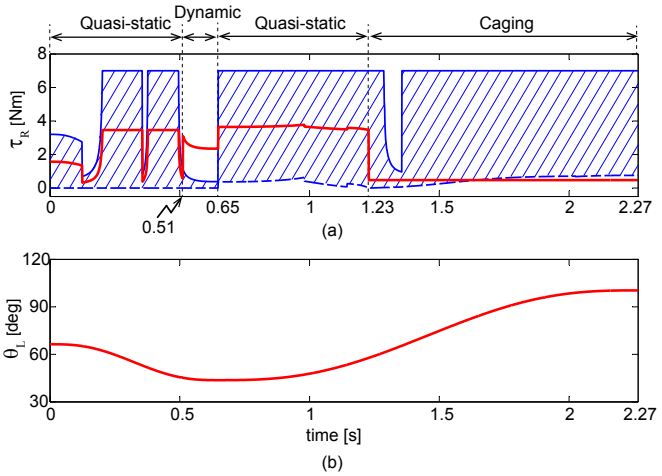

Fig. 7. Desired trajectories obtained in phase 2. (a) Joint torque trajectory of the torque-controlled finger (right finger) and the manipulation mode. (b) Joint angle trajectory of the velocity-controlled finger (left finger).

is performed between $t = 1.23$ and 2.27 s by applying the constant joint torque τ_{caging} because the center of gravity comes across the contact normal of the vertex contacting the environment (see $\#43$–$45, 53$, and 54 in Fig. 6).

D. Experimental Results and Discussions

In order to verify the validity of the proposed methods, we conducted manipulation experiments. Fig. 8 shows photographs of the movement of the object and fingers. The velocity-controlled finger is controlled by a simple PD feedback controller, and the torque-controlled finger is controlled by an open-loop controller.

The object starts to slide along the environment toward the right by rotating the velocity-controlled left finger clockwise while maintaining equilibrium in the quasi-static manipulation mode. The right-most contact on the environment breaks at $t = 0.51$ s by starting the dynamic manipulation. By pushing the object firmly with the torque-controlled right finger, the object rotates counterclockwise around the left-most vertex contacting the environment. When the lower right edge of the object makes contact with the link of the right finger at $t = 0.65$ s, the manipulation mode is switched to the quasi-static mode. Then, the caging manipulation starts at $t = 1.23$ s. Although the left finger breaks contact with the object at $t = 1.34$ s, the caging manipulation enables the object to continue rotating under the force of gravity. During the rotation of the object around its vertex contacting the environment, the contact point slides on the environment because of frictionless contact. Finally, the object reaches the goal configuration at $t = 2.27$ s.

The transition to and from the dynamic manipulation might not be made smoothly. The dynamic manipulation is crucial to successful planning because this manipulation is performed during the period when the quasi-static manipulation cannot be applied. However, this manipulation is easily affected by uncertainties and requires the value of joint torque τ_{extra} and the timing of the application of the joint torque to be set carefully based on experiments

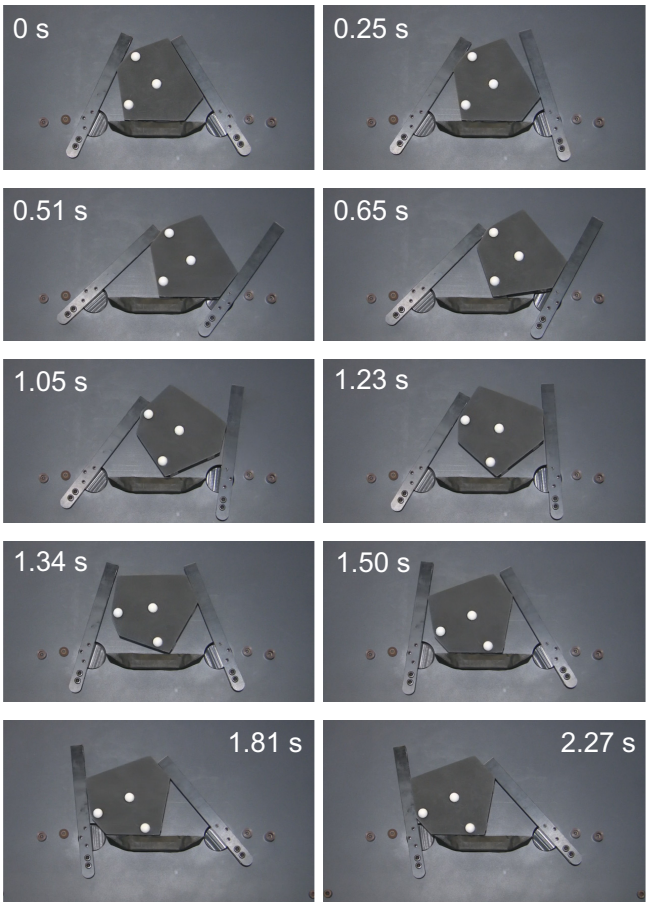

Fig. 8. Photographs of the manipulation experiment involving an asymmetric hexagonal object

in order to successfully perform the dynamic manipulation. We are considering control strategies that can make smooth transitions to and from the dynamic manipulation mode, even if there are slight control errors, for example.

The attached videos show the results of the two experiments. The first video shows the results of the experiment shown in Fig. 8, and the second video shows the results of the experiment using a regular hexagonal object having a mass of 335 g. These results reveal the validity of the proposed manipulation planning and its adaptability to various object shapes.

V. Conclusions and Future Research

The present paper discussed a planning and control strategy for whole-arm manipulations. A novel randomized planning was proposed in order to obtain a transition graph of the contact states, which can reduce the amount of computation required and can handle a hybrid system with a large search space and complicated constraints. This method can be applied to various planning problems, such as the planning of assembly tasks. Moreover, switching manipulation modes among the quasi-static, dynamic, and caging manipulation depending on the situation was demonstrated to be effective in increasing manipulation feasibility and reliability.

In the future, we intend to (i) develop optimization algorithms for the contact state transition because randomized planning cannot assure the optimality of the planned transition, (ii) design manipulation planning and control, which is currently affected by uncertainties, in the dynamic manipulation mode, and (iii) validate the proposed method for a more complicated manipulation system composed of multi-DOF fingers in a 3D space.

References

[1] K. M. Lynch and M. T. Mason, “Dynamic Nonprehensile Manipulation: Controllability, Planning, and Experiments, The International Journal of Robotics and Research, vol. 18, no. 1, pp. 64-92, 1999.

[2] Y. Maeda, T. Nakamura and T. Arai, “Motion Planning of Robot Fingertips for Graspless Manipulation,” Proc. of the 2004 IEEE Int. Conference on Robotics and Automation, pp. 2951-2956, 2004.

[3] B. J. McCarragher and H. Asada, “The Discrete Event Control of Robotic Assembly Tasks,” Trans. of the ASME Jouranal of Dynamic Systems, Measurement, and Control, vol. 117, pp. 384-393, Sep. 1995.

[4] T. Omata and K. Nagata, “Rigid Body Analysis of the Indeterminate Grasp Force in power Grasps,” Proc. of the 1996 IEEE International Conference on Robotics and Automation, pp.1787-1794, 1996.

[5] G. A. Pereira, M. F. Campos and V. Kumar, “Decentralized Algorithms for Multi-Robot Manipulation via Caging,” The International Journal of Robotics and Research, vol. 23, no. 7/8, pp. 783-795, 2004.

[6] A. Sudsang, J. Ponce, M. Hyman and D.J. Kriegman “On Manipulating Polygonal Objects with Three 2-DOF Robots in the Plane,” Proc. of the 2002 IEEE International Conference on Robotics and Automation, pp. 2227-2234, 1998.

[7] P. Song, M. Yashima and V. Kumar, “Dynamic Simulation for Grasping and Whole Arm Manipulation,” Proc. of the 2000 IEEE International Conference on Robotics and Automation, pp. 1082-1087, 2000.

[8] P. Song, M. Yashima and V. Kumar, “Dynamics and Control of Whole Arm Grasps,” Proc. of the 2001 IEEE International Conference on Robotics and Automation, pp. 2229-2234, 2001.

[9] S. S. Srinivasa, M. A. Erdmann and M. T. Mason, “Control Synthesis for Dynamic Contact Manipulation,” Proc. of the 2005 IEEE International Conference on Robotics and Automation, pp. 2523-2528, 2005.

[10] D. Stewart and J. Trinkle, “An implicit time-stepping scheme for rigid-body dynamics with inelastic collisions and Coulomb friction,” Int. J. of Numerical Methods Engineering, vol. 39, pp. 2673-2691, 1996.

[11] E. Todorov, “Implicit Nonlinear complementarity: A New Approach to Contact Dynamics,” Proc. of the 2010 IEEE International Conference on Robotics and Automation, pp.2322-2329, 2010.

[12] J. C. Trinkle and J. J. Hunter, “A Framework for Planning Dexterous Manipulation,” Proc. of the 1991 IEEE International Conference on Robotics and Automation, pp.1245-1251, 1991.

[13] J. C. Trinkle, R. C. Ram, and A. O. Farahat, “Dexterous Manipulation Planning and Execution of an Enveloped Slippery Workpiece,” Proc. of the 1993 IEEE International Conference on Robotics and Automation, pp.442-448, 1993.

[14] J. C. Trinkle, D.C. Zeng, “Prediction of the Quasistatic Planar Motion of a Contact Rigid Body,” IEEE Trans. on Robotic and Automation, vol. 11, no. 2, pp. 229-246, 1995.

[15] T. Watanabe, K. Harada, T. Yoshikawa, and Z. Jiang, “Towards Whole Arm Manipulation by Contact State Transition,” Proc. of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5682-5687, 2006.

[16] M. Yashima, Y. Shiina and H. Yamaguchi, “Randomized Manipulation Planning for A Multi-Fingered Hand by Switching Contact Modes,” Proc. of the 2003 IEEE International Conference on Robotics and Automation, pp. 2689-2694, 2003.

[17] M. Yashima, “Manipulation Planning for Re-Orientation based on Randomized Techniques, ” Proc. of the 2004 IEEE International Conference on Robotics and Automation, pp.1245-1251, 2004.

[18] Y. Yu and T. Yoshikawa, “Evaluation of Contact Stability between Objects,” Proc. of the 1997 IEEE International Conference on Robotics and Automation, pp. 695-702, 1997.

[19] X. Y. Zhang, Y. Nakamura and K. Yoshimoto, “Robustness of Power Grasp,” Proc. of the 1994 IEEE International Conference on Robotics and Automation, pp.2828-2835, 1994.