Rotary Surface Object Manipulation by Multifingered Robot Hand

Qiang Li, Robert Haschke and Helge Ritter
Research Institute for Cognition and Robotics(Cor-Lab), Bielefeld University

Bram Bolder
Honda Research Institute Europe (HRI-EU), Offenbach, Germany

Abstract

We propose a simple but efficient control strategy to manipulate objects of unknown shape, weight, and friction properties – prerequisites which are necessary for classical offline grasping and manipulation methods. With this strategy, the object can be manipulated in hand in a large scale,(eg. to rotate the object 360 degree) regardless whether there is rolling or sliding motion between the fingertips and object. The proposed control strategy employs estimated contact point locations, which can be obtained from modern tactile sensors with good spatial resolution. The feasibility of the strategy is proven in simulation experiments employing a physics engine providing exact contact information. However, to motivate the applicability in real world scenarios, where only coarse and noisy contact information will be available, we also evaluated the performance of the approach when adding artificial noise.

1 Introduction

We consider the challenging task of dexterously manipulating an object within a multi-fingered robot hand, i.e. moving the object with respect to the hand.

There exists a considerable amount of work to analytically describe the motion of the object, finger tips and contact points during manipulation. These theoretical approaches assume various things to be known: the hand kinematics, object properties like shape, mass and mass distribution, the contact locations and friction coefficients, and the local surface geometry of both the object and finger tips. Based on this knowledge it is possible to compute joint-level finger trajectories in an offline fashion, and even determine slipping and rolling motions of the fingertips [3].

Recent approaches to object-in-hand manipulation avoid the complex analysis of geometric relations and apply state-of-the-art *motion planning* methods [6] like RRT [7] and PRM [4] to the manipulation problem. All these approaches attempt to find feasible motion trajectories in an *offline fashion* utilizing a physics simulation to model the outcome of random actions. Again, this requires a considerable amount of prior knowledge about the manipulated object. Employing fast tactile feedback, Ishihara et. al [1] propose a control law to spin a pen of known shape at an impressive speed. Tahara et. al [5] point out a method to manipulate objects of unknown shape. They use a virtual object frame determined by the triangular finger-tip configuration of a three-fingered hand to derive a control law to manipulate the object's pose. However, without explicit sensory feedback, their method is limited in accuracy.

The latter two approaches propose a *reactive* control law for object manipulation, which is in our opinion a major prerequisite for robust object manipulation. Feedforward execution of manipulation trajectories obtained in an offline optimization process cannot account for real-life deviations from the planned trajectory: The initial object pose might be estimated incorrectly, fingers might unpredictedly slide or roll or even loose contact at all. Consequently, we also propose a reactive control strategy based on pose feedback. Currently, we obtain this feedback from a physical simulation, which is used to show the feasibility of the approach. However, the pose feedback can also be estimated from visual features. To confirm the applicability of our method in noisy real-world scenarios, we add artificial noise to the accurate sensor readings obtained from simulation.

Conceptually, the object manipulation process can be divided into two stages: a local manipulation controller and a globally acting regrasp planner. The local controller reactively moves the object by a small amount only. Regrasp planning is employed to adapt the grasp configuration. Subsequently, local manipulation is continued. Our previous work [2] has shown the feasibility of small scale local manipulation of unknown objects. In this paper, we use an rotary object large scale manipulation to show the strategy can be used to perform the complex manipulation in hand. Our contributions are described a follows:

- Based on the local manipulation controller, we extract abstract action primitives which can be *organized* to perform the complex manipulation in-hand task. These action primitives can be instantiated by assigning the parameters defined in Cartesian space, contact force space and feature space.
- We propose that FSM(Finite State Machine) serves

regrasp planner to manage the discrete manipulation state and its transition during the course of manipulation. With this FSM, multifingers are coordinated to grip the object, actively contact the object's surface and rotate the object.

- We demonstrate the instantiated primitives action and a large scale object manipulation by physics simulation. We analyze the hardware and software requirement for the real world robot hand to use our strategy.

The paper is organized as following. In section 2, we summarize the local reactive manipulation strategy. In section 3, we introduce discrete state space, action primitives definition and how to use finite state machine(FSM) to manage state transition. In section 4, the physical simulation setup used for evaluation is introduced and simulation result is shown. Discussion is given on the complex object manipulation and how to transfer current simulation work to the real-world robot hand. Finally, section 5 summarizes our work.

2 Local Reactive Manipulation Strategy

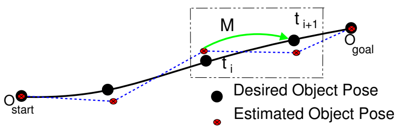

Figure 1: Incremental manipulation of object pose O.

Conventional grasp and manipulation planning methods uncoupled the planning from the control stage. The planning stage strongly depended on global knowledge about the geometry and friction issue of the object and fingertips. As the responding to the unknown geometry and friction issue of object surface, we use the point contact model in the strategy. We do not explicitly model friction properties. However,the physical simulation adopts a Coloumb friction model approximating circular friction cones by four-sided pyramides. We employed micro manipulation assumption and rapid feedback loop to plan contact position and force, not the accurate contact interaction geometry plan.
We assume that we can estimate the current 3D pose O of the object and object's target pose O' comes from high level global planner. eg. desired task, its decomposed subtask and instantiated action primitives. These pose is defined in the continuous Cartesian space which can be described using arbitrary object Cartesian space configuration description method(homogeneous transform matrix, euler angle or Quaternion). Here we use homogeneous transform matrix description because it is easy to integrate this description into the exist matrix calculation library and also easy to transfer to the euler angle which will facilitate the visualizing the manipulation result. Based on O and O', we derive the required object motion M to realize the target pose within the next control cycle (cf. Fig. 1). Current local manipulation controller works in the *pure reactive* mode and it does not explicitly consider the reachable problem of the next desired object's pose in the continuous Cartesian space. This working mode can work well when the object surface is rotary, continuous and no edge occurring during the course of manipulation. While the complex object(eg.polyhedron) is manipulated, the robot hand should have cognition capability to predict the edge from the sensors feedback. Using this information to replan the manipulation task will be the part of our future work.
The contact point position planing in Cartesian space is described as following: Denoting the current and targeted object pose with O and O' resp, we can easily compute the transformation matrix M describing the required finite object motion:

$$O' = O \cdot M \qquad \Leftrightarrow \qquad M = O^{-1} \cdot O' \,. \qquad (1)$$

Assuming, that contact positions do not move relative to the object within the control cycle, we can calculate the new contact positions $\vec{p}_i'$ (w.r.t. the palm). The contact positions $\vec{p}_i^o$ expressed relative to the object frame stay fixed:

$$\vec{p}_i' = O' \cdot \vec{p}_i^o = O \cdot M \cdot O^{-1} \vec{p}_i \,. \qquad (2)$$

From this we can compute the required positional changes $\Delta\vec{p}_i = \vec{p}_i' - \vec{p}_i$ for all contact points as input to the inverse hand kinematics. Because the local object and finger tip geometries as well as grasp stability measures are not explicitly taken into account, the actual grasp configuration might have changed after application of the computed hand pose. This corresponds to sliding or rolling contacts or even to a loss of a contact. On the contrary with the feedforward control which require one complete world model, our manipulation strategy just use the fast feedback to solve the unexpected rolling/sliding phenomenon.

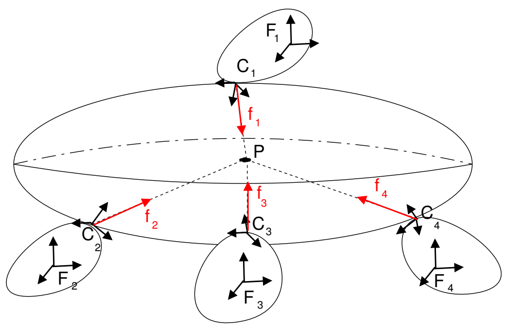

Figure 2: Force planner employs centroid $\bar{\vec{p}}$ of contact locations.

In order to maintain stable contacts anyway, we apply a force-control scheme additionally to the position-controlled object manipulation. Conventional contact force planners strive for a globally optimal contact force distribution ensuring grasp stability, i.e. all contact forces staying within corresponding friction cones, the totally applied force exactly resisting external forces (e.g. gravity), and limiting local contact forces. This general solution is meaningful only if the contact force is controllable. However, we assume that there is no 3D contact force feedback (obtained directly or indirectly), but only the force magnitude is available from tactile sensors. Following concepts from [5] the central idea is to plan the force direction such that the resultant moment will be zero, and to plan the force magnitudes along these directions such that the resultant force applied to the object becomes zero. Obviously the resultant moment is zero, if the contact force directions of all fingers intersect in one point. The force planner is illustrated in Fig. 2. The output of force planner is desired contact force along the *contact direction*, so the deviation force between current contact force and desired force can calculated. Two deviation from position planner and force planner are sent to composite position/force controller to calculate the composite contact position deviation which will be sent to Inverse Kinematics Module. One thing needed to be paid attention is that composite controller is composited by P/PI controller in position and force channel resp. in order to keep the no static error in contact force closed loop control.

All components of the local manipulation controller and robot hand angle angle position servo control are summarized in Fig. 3. More detailed description about the local manipulation controller can be found in [2].

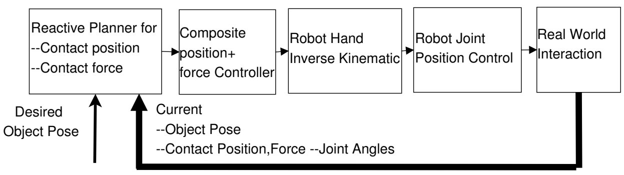

Figure 3: Components of closed loop control scheme.

3 Regrasp Planner

Local reactive manipulation is not enough for the multifingered robot hand to implement large scale object manipulation task. When the local manipulation is end, the robot hand posture must be changed in order to facilitate the new cycle local manipulation. This will be done by the regrasp planner. The basic idea of regrasp planner is to use three grip fingers to stably grasp and rotate the object and one exploration finger to search and contact new contact point in order to facilitate new cycle local manipulation. The grip fingers' motion follows the local(position/force) manipulation controller and exploration finger contact and move on the surface of the object in order to find a new grasp point which makes the new cycle manipulation more *comfortable*. In this part, we introduce that employing the simple action primitives and the finger role switching mechanism managed by FSM to perform an exemplary complex action – large scale object rotation manipulation. Our strategy consider fingers as multi-actuators and provide a kind of engineering solution for robot hand manipulation. Four fingers are enough to prove this idea.

3.1 Definition of State and Action Primitives

Intuitive object manipulation in human hand shows us that object complex manipulation task can be decomposed into some simple action primitives. It will be helpful to extract the primitives which are shared by many given real application tasks. Such primitive can be executed in sequence or simultaneously to finish the complex task. These primitives can be abstract, but should be realized by the current perception and control technology. We use the developed local manipulation controllers to define two action primitives.

(1)multifinger coordinately hold/move/rotate the object locally.

(2)single finger explore the object surface with the desired force.

According to how to select grip/rotation fingers and exploration finger, the two action primitives can be instantiated and the instantiated primitives are coordinated and spanned to the action space defined as following.

$A1$: **TFMR Rotate** $A2$: **TFM Rotate**
$A3$: **R Grip** $A4$: **R Exploration**
$A5$: **TFR Rotate** $A6$: **M Grip**
$A7$: **M Exploration** $A8$: **TMR Rotate**
$A9$: **F Grip** $A10$:**F Exploration**
$A11$: **FMR Rotate** $A12$: **T Grip**
$A13$: **T Exploration**

Where T: Thumb; F: Forefinger; M: Middlefinger; R:Ringfinger. "Exploration" means that the finger serves as exploration role and contact the object with very small contact force and explore the neighbor feasible contact points. "Grip" means that the exploration finger changes it role to grip finger and contacts the object with desired contact force planed by contact force planner. "Rotate" means that grip fingers locally rotate the object. Full 3D object manipulation(not only object rotational motion but also translational motion) will span a larger action space comparing with only rotational manipulation, which is the intuitive extension of current orientation maniplation. eg.

$A14$: **TFMR Move** $A15$: **TFM Move**
$A16$: **TFR Move** $A17$: **TMR Move**
$A18$: **FMR Move**

We don't explicitely describe such actions in FSM because we just need to use orientation manipulation to prove the feasible of our strategy.

Precondition of action primitive is the current manipulation state, and trigger event is the starting of the new cycle local manipulation or the accomplishment of one action primitive. We propose to use the finger role to define the manipulation state because such definition can constrain object manipulation into finite discrete state space, which can largely reduce the search space of global regrasp planner. The fingers role can be classified into two classes.

(1)grip fingers. Their main functions are to grasp and locally manipulate the object.

(2)exploration finger. Its function is to search new feasible stable grasp point.

The role of finger is exclusive, and one finger can not be grip finger and exploration finger at the same time. In temporal domain, however, the finger role can be changed. When the exploration finger find the new feasible grasp point, the role of exploration finger is switched to grip finger and planner will select another proper grip finger to serve exploration finger. The switch of finger role means the state transition in state space. We define the state space as following.

$S1$: **All are grip fingers**

$S2$: **TFM grip fingers and R exploration finger**

$S3$: **TFR grip fingers and M exploration finger**

$S4$: **TMR grip fingers and F exploration finger**

$S5$: **FMR grip fingers and T exploration finger**

3.2 Finite State Machine

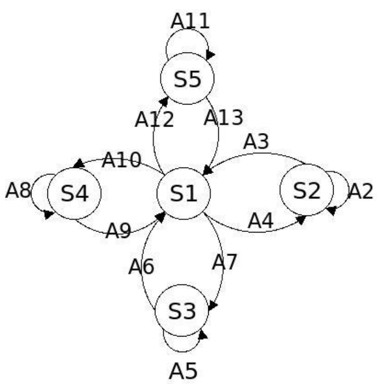

Figure 4: FSM serve as the manager of finger gaits.

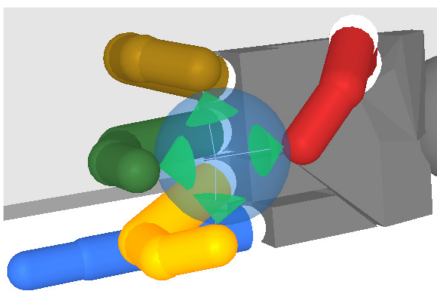

Figure 5: Simulation scenario

The whole object manipulation process is shown in Fig. 4, which can be formalized as a FSM. This is a closed statechart and it's possible to transit from current state to the other arbitrary state in the statechart. Such state transition must pass by the medium state–four finger holding the object.

FSM starts from all fingers grasping the object in a *comfort* manipulation posture and all fingers are defined as grip fingers. Every finger use the same finger switching process, so only ring finger switch is analyzed here as an example. Initially manipulation state is S1, and action 4 (A4) is taken after high level task is given(eg.rotate the object 10 degree). After this action is finished, the process goes into state 2 (S2) and ring finger serves as exploration finger and contacts the object with a small enough desired force(e.g 0.1). TFM are grip fingers and their desired force/position are planned by the local force/position planner to stably grasp object. Small enough contact force of exploration finger is used in order to not damage the other three finger stable grasping. It also provides the contact information about the exploration finger, which is helpful for searching new feasible grasp point. At state 2 (S2), action 2 (A2) is taken and TFM–three grip fingers are used to rotate the object. This is a local rotation manipulation, we can use the local manipulation algorithm developed in [2]. The exploration finger still contact the rotary object with small contact force and try to explore its neighbor feasible contact points in the object surface. After a small rotation is finished, action 3 (A3) is taken and ring finger releases its exploration finger role and changes to be the grip finger. Its desired contact force is changed from small enough to the desired contact force planed by local contact force planner. In this case, manipulation process reaches back to state 1 (S1). Other fingers use the same principle to realize the finger switch.

3.3 Global Object Manipulation in Hand Flowchart

A general global regrasp planner has been described in the previous part. In theory, many AI planners can be used to plan the complex object manipulation task. However, most of these planners can only solve the structured and determinated task planning problem. They are not fit for the dynamic and uncertainty object manipulation task.

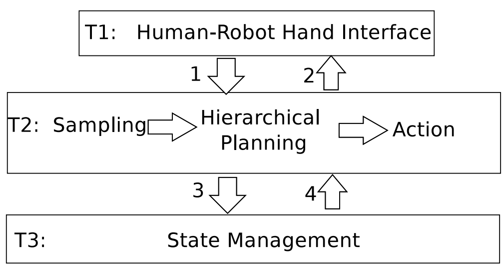

Figure 6: Object manipulation in hand flowchart

In this paper, we focus on a simple and intuitive hard-coding way to solve the given object large scale manipulation task(360 degree rotation manipulation). Some more complex tasks, eg. dynamic changed object manipulation task by user/ controlling the object interacting with the environment etc. need a formal automatic hierarchical planner, we will devote such work in future. Currently, the working threads are defined as 6.

$T1, T2, T3$ are three synchronization working thread. $T1$ is in charge of accepting the human high level task(arrow 1) and visualization of the manipulation(arrow 2). $T2$ is in charge of hierarchical planning to generate the desired joints level control input. $T3$ runs a FSM to manage the state transition(Fig. 4).

Take 360 degree rotation task as an example. Firstly, the initial pose of the object is defined as reference pose. The maximum rotation angle of the object in every local manipulation is limited as 10 degree. One of finger is selected as exploration finger(eg.ring finger). Primitive 1 is applied on the other three grip fingers and primitive 2 is applied on the ring finger. After $A4$ is finished, $A2$ is taken and the object is locally rotateed 10 degree. In order to rotate the object smoothly, we plan the rotation performed in 100 control step(Fig. 3). At every control step, 0.1 degree incremental is sent as the desired input. Euler angle description is changed to homogeneous description in order to facilitate the calculation in Eq. 1, Eq. 2. After $A2$ is finished, $A3$ is taken. It's a new instantiated primitive 2, the desired contact position is previous contact position of ring finger in global frame. This is a period of ring finger gait. Middle finger, index finger and thumb in sequence use the same principle to realize their own gaits and finally to perform 360 degree rotation of the object.

4 Simulation

4.1 Simulation Result

The object manipulation algorithm is validated in a physical simulation experiment. We use the Vortex physics engine to obtain real-time contact information (i.e. contact position and contact force magnitude), and the object's pose (object position and orientation). Currently two geometric primitives, namely sphere (radius=2.5cm), cylinder (radius=2.5cm, height=9cm) are evaluated. The objects are sized middle-scale compared to the robot hand, so rolling and slipping between the fingertips and the object will occur during the course of the manipulation. Object parameters information(radius of the ball and cylinder) is not available to the manipulation strategy. The simulation scenario is shown in Fig. 5 resembling our real robot setup to facilitate future transfer into real world, once the required tactile feedback is robustly available from finger tip sensors.

We assume that the object has been successfully grasped and grasp points are *comfort* for the manipulation. Because the object has the revolution surface, the finger can use the previous grasp point as the new cycle grasp point when its role is changed from "exploration" to "grip".

In the following we present results for the exemplary manipulative motions to show the large scale object manipulation feasibility with our strategy.

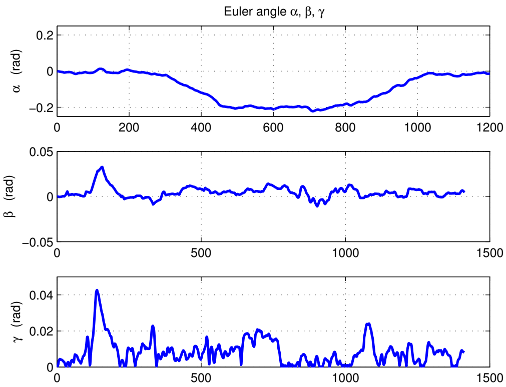

Figure 7: Object rotation around x-axis

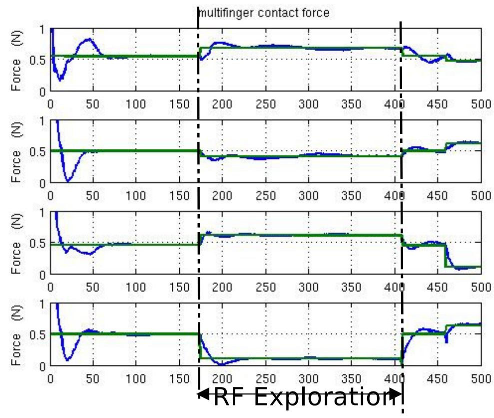

Figure 8: Exploration(ring) finger contact the object.

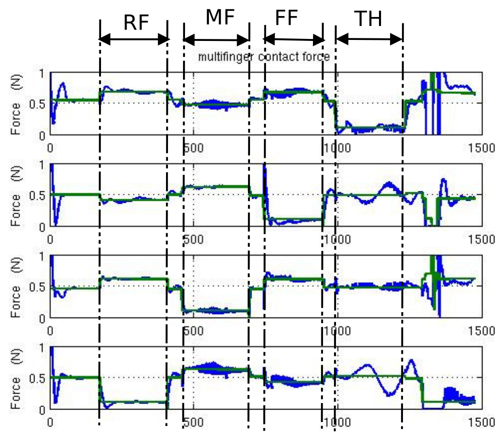

Figure 9: Finger roles switching.

(1) Action primitive I - small rotational movement of 0.2 rad around the x-axis. In simulation, we superimposed artificial measurement noise to the object position value from the physics engine. The standard deviations of the added

Gaussian noise are: 0.5cm. In Fig. 7 it shows that object is rotated from 0 to 0.2 rad and back again.
(2) Action primitive II - exploration finger contacts the object with the desired force. In Fig. 8, ring finger is exploration finger and it contact the object with small contact force – 0.1. Other three fingers use the contact planner to design the desired contact force.
(3) Continuously rotate the object. Here only one periodical of finger role switching is shown in Fig. 9. When the desired contact force is small enough (eg.0.1), it represents this finger is exploration finger. From Fig. 9, it shows that ring finger, middle finger, forefinger and thumb serve exploration finger in sequence to implement the continuous rotation of the object.

4.2 Discussion

Current work can be extended along two paths.

1. Transfer the simulation result to real world robot hand.

   The input of our current algorithm are fingers joints angle, contact points position, contact force, palm position and object pose. Usually, most of state of the art robot hands are equipped with joints angle sensors. They, however, should be calibrated before feedback value can be used. Palm position and object pose can be obtained using the robot vision by adding marker on the palm and object. the contact point position can be calculated using the forward kinematics model by combining the known hand kinematics model, tactile sensors distribution model, and joint angle. Contact force model is one mapping from raw tactile sensors to calibrated force.

2. Complex geometry object manipulation.

   With current algorithm, we try to solve a *unknown* simple rotary surface object manipulation in hand. *Unknown* means that the size, weight and friction coefficient are unknown for the robot hand. We, however, exploit the prior knowledge – rotary object and initial good grasp points, and it's not necessary to use the exploration finger to explore the object surface to find the new good grasp points. In this way, we can focus our work on the local manipulation and manually FSM to realize the object large scale manipulation. When the object becomes more and more complex, the exploration finger must have the capability to find the feasible grasp points and the general finger gaits planner should be developed to autonomously plan the finger gaits by fusing data from low level, state machine and current task. Object properties cognition is the prerequisite of complex geometry object manipulation. The object properties have to be represented and fused into the planner in order to realize *real unknown* object manipulation in hand.

5 Summary

We propose a reactive control strategy to realize manipulation motions for rotary objects. In contrast to traditional manipulation strategies, which require a lot of information about the object and plan in an offline fashion, our plan method developed in local manipulation level and global manipulation level is in an online fashion and employs minimal sensory information. In physical simulation experiments we proved the feasibility of the method to perform complex manipulation task – large scale rotate objects.

ACKNOWLEDGEMENT

Qiang Li gratefully acknowledges the financial support from Honda Research Institute Europe for the project "Autonomous Exploration of Manual Interaction Space".

References

[1] T. Ishihara, A. Namiki, M. Ishikawa, and M. Shimojo. Dynamic pen spinning using a high-speed multifingered hand with high-speed tactile sensor. In *Humanoid Robots,IEEE International Conference on*, pages 258 –263, 2006.

[2] Qiang Li, Robert Haschke, Helge Ritter, and Bram Bolder. Simulation results for manipulation of unknown objects in hand. In *Robotics and Biomimetics, IEEE International Conference on*, 2011.

[3] R.M.Murray, Z.X.Li, and S.S.Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[4] J.P. Saut, A. Sahbani, S. El-Khoury, and V. Perdereau. Dexterous manipulation planning using probabilistic roadmaps in continuous grasp subspaces. In *Intelligent Robots and Systems, IEEE International Conference on*, pages 2907–2912, 2007.

[5] K. Tahara, S. Arimoto, and M. Yoshida. Dynamic object manipulation using a virtual frame by a triple soft-fingered robotic hand. In *Robotics and Automation, IEEE International Conference on*, pages 4322 –4327, 2010.

[6] Z. Xue, J.M. Zollner, and R. Dillmann. Dexterous manipulation planning of objects with surface of revolution. In *Intelligent Robots and Systems, IEEE International Conference on*, pages 2703–2708, 2008.

[7] M. Yashima. Manipulation planning for object re-orientation based on randomized techniques. In *Robotics and Automation, IEEE International Conference on*, volume 2, pages 1245–1251, 2004.