2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Dexterous Manipulation Using Both Palm and Fingers

Yunfei Bai and C. Karen Liu

Abstract— **This paper focuses on the problem of manipulating the orientation of a polygonal object in hand. We propose a control technique which integrates the use of palm and fingers to pick up a given object on the table, to drop it on a specific spot on the palm, and to let it roll continuously and controllably on the palm, subject to the gravitational and contact forces. We formulate a simple and fast algorithm to control the tilting angle of the palm based on the conservation of mechanical energy and an empirical model of energy dissipation due to collisions. We also develop a multifingers controller for stable grasp and for correcting the rolling motion. The proposed technique is demonstrated on a simulated robotic hand manipulating a wide range of objects with different geometry and physical properties.**

I. INTRODUCTION

Using multifingered hands for dexterous tasks has many potential advantages. In addition to efficiency and versatility, multifingered hand dexterity provides additional degrees of freedom to increase the workspace of a manipulator [1]. On the other hand, not using fingers to grasp can also be an effective manipulation strategy in practice. For example, the absence of grippers largely simplifies the mechanical design, while increasing the flexibility to manipulate objects with various sizes and shapes [2].

While imitating anthropomorphic hands is arguably not an optimal solution to many practical applications, it is evident that a wide range of human manipulation tasks can benefit from integrative collaboration between appendages that emulate fingers and a surface that emulates the palm. In this paper, we focus on the problem of manipulating the orientation of a polygonal object in hand. That is, given an initial orientation of the object on the table, can the robotic hand pick up the object and re-orient it to a desired configuration in hand? Our approach leverages a palm-like surface for dynamic nonprehensile manipulation and finger-like appendages to perform simple grasp and corrective control. By integrating the use of a "palm" and "fingers", our approach is able to efficiently and robustly re-orient objects with different geometry and physical properties.

We propose a control technique for a robotic hand to pick up a given object on the table, to drop it on a specific spot on the palm, and to let it roll continuously and controllably on the palm, subject to the gravitational and contact forces. We formulate a simple and fast algorithm to control the tilting angle of the palm based on the conservation of mechanical energy and an empirical model of energy dissipation due to collisions. While this nonprehensile approach requires minimal sensing and actuation capability, the object might not be executed precisely due to unexpected perturbations and inconsistency between the model and the real world. To mitigate execution errors, we develop another multifingers controller which corrects errors as the object rolls on the palm. Our method requires the geometry information of the object to be known a priori, but has no limitation on the convexity and symmetry of the shape, nor the location of the center of mass (COM).

The approach was demonstrated on a Shadow Dexterous Hand$^{\mathrm{TM}}$ simulated using Gazebo simulator [3] with DART physics engine [4]. We showed that the robotic hand was able to manipulate a wide range of objects with different shapes, masses, moment of inertias, and friction coefficients, including nonconvex, irregular objects with an offset center of mass. We also evaluated the proposed technique with noisy vision sensor input and objects with unsmooth surfaces.

II. RELATED WORK

Problems involving dexterous manipulation have been frequently addressed in the context of robotics research. A wide range of manipulation strategies, such as finger gaiting [5], finger pivoting [6], rolling/sliding [7]–[10], and regrasping [11], have been proposed for achieving different dexterous tasks. Although a precise definition of dexterous manipulation is still open to interpretation, many previous reviews provided nice discussion to summarize a variety of dexterous robotic systems based on their functionalities, hardware designs, and planning strategies. In particular, Bicchi [12] made a distinction between anthropomorphic hands to mimic the human anatomy and "minimalistic" hands to meet practical requirements, and argued for the necessity of hand dexterity. Ma and Dollar [1] argued that a simple gripper and a dexterous arm is sufficient for many applications, but a dexterous end-effector can increase the workspace of the arm. Instead of using a dexterous end-effector to make up for the limitations in arm functionality, our technique leverages nonprehensile manipulation on the palm to expand the possible motions of the object, while using multifingers mainly for the purpose of stable grasp.

An alternative approach is to manipulate objects without grasping them. Nonprehensile manipulation allows a robot to use fewer actuated degrees of freedom to manipulate an object, increasing the set of reachable configurations of the object for a simple manipulator. A variety of strategies have been proposed, such as tumbling [13], tilting [14], pivoting [15], tapping [16], [17], two-pin manipulation [18], and two-palm manipulation [19], [20]. The earlier work

*This work was supported by NSF CAREER under Award CCF 0742303 and Alfred P. Sloan Fellowship.

[1]The authors are with School of Interactive Computing, Georgia Institute of Technology. ybai30@mail.gatech.edu

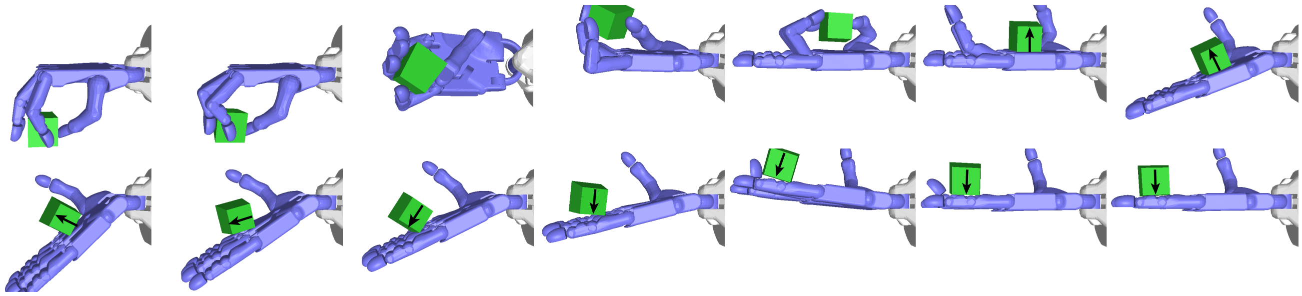

Fig. 1. Grasp and roll. Screenshots from a simulated sequence of the Shadow Hand grasping and rolling an object to manipulate its orientation. The arrow represents the orientation of the object.

done by Erdmann *et al.* [21] solved for a sequence of tilting angles such that a polygonal object on the table can be orientated into a set of desired configurations. Later, Lynch and Mason [22] considered dynamic forces on the object when planning the trajectories for the manipulator. By leveraging dynamics, they showed that an underactuated manipulator could snatch, roll, throw, and catch an object. Srinivasa *et al.* [23] takled the problem of rolling a block sitting on the palm by 90 degree using a trajectory planning technique. They further extended the optimal trajectory to an optimal feedback controller using dynamic programming. This paper addressed a similar nonprehensile manipulation problem. However, instead of planning the trajectory of the tilting angle using optimization techniques, we proposed a different approach based on energy formulation to handle continuous rolling with multiple contacting faces.

III. Problem Statement and Assumptions

The problem we focus in this paper is formulated as follows. Given a polygonal object rested on an arbitrary face on the table, the robotic hand must manipulate the object such that it rests on a desired face on the hand (Figure 1). We propose an approach in which a robotic hand picks up the object from the table, drops it on the palm, and rolls the object to reach the desired contacting face while keeping the object in hand. Our approach utilizes both the palm for rolling the object and the fingers for grasping and correcting the orientation of the object.

Our algorithm makes the following assumptions:

1) The input object has a prism-like shape. That is, the object is a polyhedron with two polygonal bases joined by a set of parallel edges.
2) The prior knowledge about the object includes the position of the center of mass, the mass, the moment of inertia, and every vertex and edge expressed in the object frame.
3) The algorithm requires a vision sensor providing 3D coordinates of at least three vertices of the object in the world frame at all times.
4) The friction coefficient is sufficiently large such that the object does not slide on the robotic hand.

IV. The Palm

We first describe the algorithm to control an object rolling on the palm from an initial contacting face to a desired contacting face. The algorithm assumes that the object has been placed on the palm with its joining edges aligned with the x-axis of the palm (Figure 2). Because the object has the shape of a prism and its joining edges are perpendicular to the rolling direction, without loss of generality, we can reduce the 3D problem to rolling a 2D cross-section of the object on a plane. If the cross-section is concave, we simply take its convex hull and use it to represent the object.

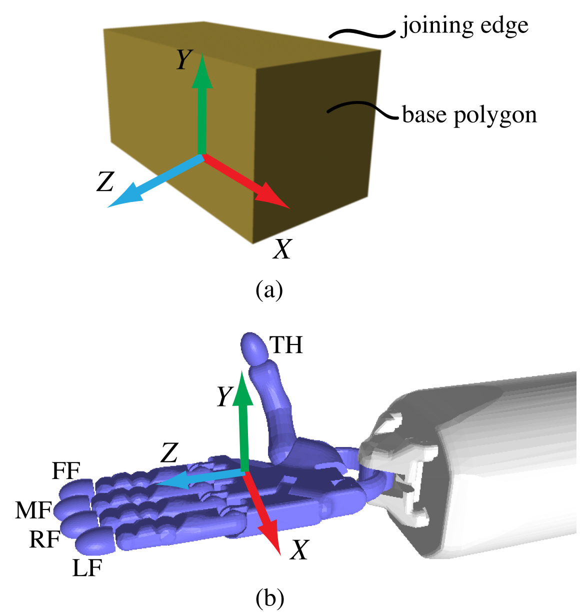

Fig. 2. The object and the hand. (a) The object has a prism-like shape, which consists of two base polygons and a set of parallel joining edges. The object frame is illustrated in the figure. (b) The Shadow Dexterous Hand with 24 degrees of freedom. Five fingers are indicated by TH, FF, MF, RF, lF. The frame of the hand is illustrated in the figure.

The goal of the algorithm is to control the tilting angle of the palm (θ) such that the polygon can continuously roll across multiple edges and stop at the desired contacting edge. We break rolling motion to a sequence of contact cycles. Each contact cycle is associated with a contacting edge $\mathbf{e}$

between two vertices $\mathbf{r}_0$ and $\mathbf{r}_1$, which form a triangle with the center of mass $\mathbf{x}$. We define the length of the two edges of the triangle adjacent to $\mathbf{x}$ as d_0 and d_1, and the two angles adjacent to $\mathbf{e}$ as ϕ_0 and ϕ_1 (Figure 3).

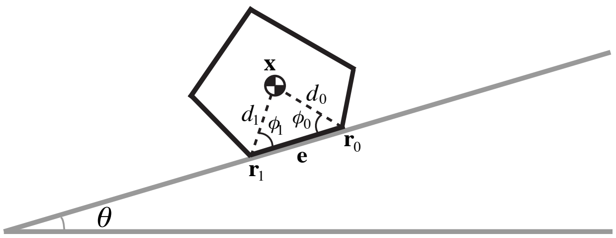

Fig. 3. Notations for one contact cycle.

An naïve approach considers each contact cycle individually and sets θ to be greater than $90 - \phi_1$ for each contact cycle. Because the center of mass is not supported by the contact, the object will roll to the next contacting edge. However, this approach does not take into account the dynamics of the object and the gravitational force, resulting in a constantly accelerating rolling motion difficult to control and stop at the end.

We propose an algorithm that yields a more conservative rolling motion by considering the kinetic energy of the system. Our algorithm can be viewed as solving a sequence of inverted pendulum problems, each of which describes the motion of the center of mass of the polygon rotating about a vertex. If the pendulum at the apex has nonzero kinetic energy, the polygon will continue to roll to the next contacting edge. Based on this simple condition, we compute a sequence of θ to achieve continuous rolling to the desired contacting edge.

A contact cycle consists of three phases: dropping, colliding, and lifting (Figure 4). The dropping phase begins when $\mathbf{x}$ is at the apex and $\mathbf{r}_0$ is the contacting vertex. The colliding phase begins when $\mathbf{r}_1$ establishes contact with the hand. The lifting phase begins when $\mathbf{r}_0$ breaks the contact. When $\mathbf{x}$ reaches the next apex with $\mathbf{r}_1$ as the contacting vertex, the next contact cycle begins. We define the kinetic energy at a few key moments of a contact cycle as follows:

- E^0: The beginning of a contact cycle.
- E^-: The end of the dropping phase right before the collision.
- E^+: The moment after the collision and the beginning of the lifting phase.
- E^1: The end of the current contact cycle and the beginning of the next contact cycle.

During the dropping phase, the conservation of mechanical energy demands that the change of kinetic energy is equal to the change of potential energy.

$$E^- - E^0 = mgd_0(1 - \sin(\phi_0 - \theta)) \quad (1)$$

where mg is the gravitational force applied on the object.

The colliding phase models the dissipation of kinetic energy due to collision. We apply the empirical law for collision with a coefficient ϵ ($0 \le \epsilon \le 1$), which measures the kinetic energy dissipation. We set ϵ to 0.5 in our experiments.

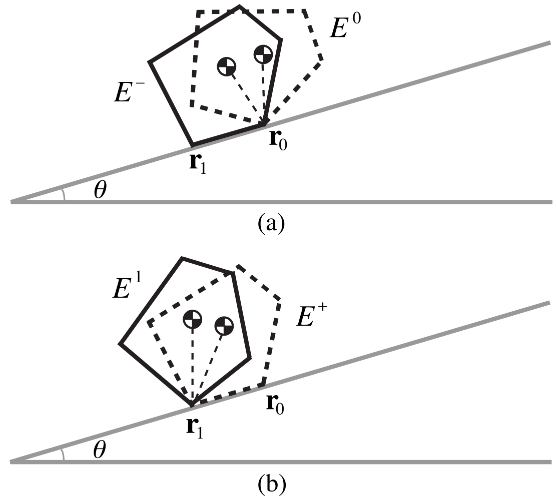

Fig. 4. A contact cycle. (a) Dropping phase: The kinetic energy at the beginning (dashed figure) and end (solid figure) of the dropping phase are E^0 and E^- respectively. (b) Lifting phase: The kinetic energy at the beginning (dashed figure) and end (solid figure) of the lifting phase are E^+ and E^1 respectively.

$$E^+ = \epsilon E^- \quad (2)$$

During the lifting phase, the polygon rolls about $\mathbf{r}_1$ until $\mathbf{x}$ reaches the apex, as the kinetic energy transforms into potential energy.

$$E^+ - E^1 = mgd_1(1 - \sin(\phi_1 + \theta)) \quad (3)$$

Using Equation 1, Equation 2, Equation 3, and the minimal kinetic energy condition $E^1 \ge 0$, the following inequality constraint on θ is derived for continuous rolling:

$$d_1 \sin(\phi_1 + \theta) - \epsilon d_0 \sin(\phi_0 - \theta) \ge d_1 - \epsilon d_0 - \frac{\epsilon}{mg} E^0 \quad (4)$$

where the kinetic energy E^0 is computed based on the state of the polygon at the beginning of the contact cycle:

$$\frac{1}{2}(m\mathbf{v}^2 + \mathbf{I}\omega^2) \quad (5)$$

where $\mathbf{v}$ and ω are the linear and angular velocity of the polygon approximated by finite differencing the current and the previous positions.

Using angle transformation formulas, we rewrite Equation 4 as follows:

$$A \sin(\theta) + B \cos(\theta) \ge C \quad (6a)$$
$$A = d_1 \cos(\phi_1) + \epsilon d_0 \cos(\phi_0) \quad (6b)$$
$$B = d_1 \sin(\phi_1) - \epsilon d_0 \sin(\phi_0) \quad (6c)$$
$$C = d_1 - \epsilon d_0 - \frac{\epsilon}{mg} E^0 \quad (6d)$$

We apply the rule for linear combination in trigonometry to obtain the following equation:

$$A \sin(\theta) + B \cos(\theta) = k \sin(\theta + \varphi) \quad (7)$$

, where $k = \sqrt{A^2 + B^2}$, and φ is the unique angle satisfying following three conditions: 1) $-\pi < \varphi \leq \pi$; 2) $\sin(\varphi) = B/k$; 3) $\cos(\varphi) = A/k$.

The first and the last contact cycles are two special cases. For the last contact cycle, we simply negates the kinetic energy condition to $E^1 < 0$, which stops $\mathbf{x}$ from reaching the next apex after the object hits the desired contacting face. For the first contact cycle, we do not consider the dropping phase and colliding phase because both $\mathbf{r}_0$ and $\mathbf{r}_1$ are already in contact with the hand. The desired angle satisfies the minimal kinetic energy condition as in Equation 4 with E^+ replaced by the initial kinetic energy E^0 :

$$d_1 \sin(\phi_1 + \theta) \geq d_1 - \frac{E^0}{mg} \tag{8}$$

The algorithm solves for θ at the beginning of the contact cycle and commands the palm to achieve the new θ before the colliding phase starts. In theory, we need to adjust both the translation and the rotation of the wrist during the dropping phase, so that the palm reaches θ while $\mathbf{r}_0$ remains stationary in the world frame. In practice, however, we can directly set the wrist angle to θ without translating it because small motion at $\mathbf{r}_0$ has little impact on the rolling.

V. The Fingers

The rolling algorithm described in Section IV utilizes passive dynamics so that the object can be manipulated by only the palm. To achieve manipulation robustly in a real scenario, however, the object needs to be first transported to the palm and the rolling motion sometimes needs to be corrected due to unexpected collisions (e.g. the palm or the object has a rough surface), inconsistency between the assumed model and the real object, or noisy vision sensing input. We propose to use fingers to complement the palm for more versatile manipulation. In particular, we use fingers to grasp the object from the table, drop it on the palm, and provide corrective control to prevent object from overshooting or deviating from the plan.

We implemented a simple grasp algorithm by controlling the pose of the hand and the force generated by the end-effectors. The algorithm first computes the desired contact points for each finger and applies the inverse kinematics method (IK) to generate a desired pose for the hand. The desired contact points can be in any locations on the object surface as long as they provide stable grasp and allow the joining edges of the object aligned with the x-axis of the hand when the object is dropped on the palm. We propose one possible way to achieve this goal: pick two opposing faces that are not base polygons, and place TH on one face and MF and RF on the other face. If FF and LF can reach the base polygons, we add additional contact points for more support. Once the object is in a stable grasp, the wrist of the robotic hand is commanded to turn $180°$ to a palm-up position.

To move the object towards the dropping location, we control the amount of force that fingers apply to the object. The total desired force $\bar{\mathbf{F}}$ and torque $\bar{\tau}$ are determined by the

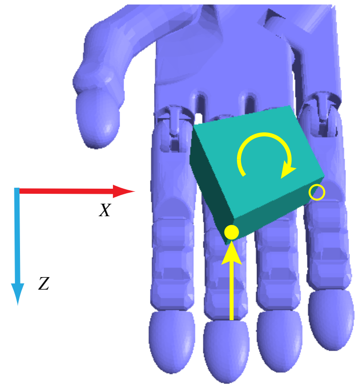

Fig. 5. Angular deviation in the y-axis. A finger is used to create a torque on the object in the opposite direction of the deviation angle. In this case, the desired torque direction is indicated as the yellow arc arrow. The bottom left corner (shown as the solid yellow dot) is selected as the point of application on which a contact force (shown as the yellow arrow) will induce a torque in the desired direction. The closest finger, MF, is selected to provide the contact force.

deviation between the current object state and the dropping location and orientation through a feedback equation:

$$\bar{\mathbf{F}} = k_p(\bar{\mathbf{u}} - \mathbf{u}) \tag{9}$$

$$\bar{\tau} = k_o \alpha(\mathbf{v} \times \bar{\mathbf{v}}) \tag{10}$$

$\mathbf{u}$ and $\bar{\mathbf{u}}$ are the current and the desired positions of the object respectively. $\mathbf{v}$ and $\bar{\mathbf{v}}$ are the current and the desired directions of a vector fixed in the local coordinate frame of the object. And α is the angle between $\mathbf{v}$ and $\bar{\mathbf{v}}$. k_p and k_o are proportional gains for position and orientation. For n contact points on the object, we compute the desired contact force $\mathbf{f}_i$ at each contact point $\mathbf{p}_i$ by solving the following optimization problem:

$$\min_{\mathbf{f}_1 \ldots \mathbf{f}_n} \omega_1 \sum_{i=1}^{n} -\frac{\mathbf{f}_i}{\|\mathbf{f}_i\|} \cdot \mathbf{n}_i + \omega_2 \sum_{i=1}^{n} \|\mathbf{f}_i\|^2 \tag{11a}$$

$$\begin{pmatrix} \mathbf{I} & \cdots & \mathbf{I} \\ [\mathbf{p}_1 - \mathbf{x}]_\times & \cdots & [\mathbf{p}_n - \mathbf{x}]_\times \end{pmatrix} \begin{pmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_n \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{F}} \\ \bar{\tau} \end{pmatrix} \tag{11b}$$

The first term of the objective function minimizes the angle between the contact force and the contact normal, where $\mathbf{n}_i$ is the contact normal at the contact point $\mathbf{p}_i$. The second term minimizes the magnitude of the contact forces. We set the weight ω_1 and ω_2 to be 50 and 1 respectively. The equality constraint requires the total effect of contact forces equals to the desired force and torque. In the equality constraint, $\mathbf{I}$ is the 3×3 identity matrix, and $[\mathbf{p}_i - \mathbf{x}]_\times$ is the skew-symmetric matrix representing the cross product of the vector from the center of mass of the object, $\mathbf{x}$, to $\mathbf{p}_i$. We use Jacobian transpose scheme [24] to control finger joints to exert desired contact forces.

$$\tau_{int} = \sum_{i=1}^{n} J_i^{\mathsf{T}} \mathbf{f}_i, \tag{12}$$

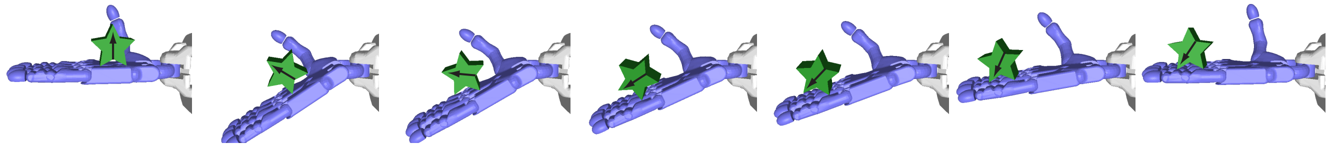

Fig. 6. A nonconvex object. Screenshots from a simulated sequence of rolling a nonconvex object. The arrow represents the orientation of the object.

where τ_{int} indicates the control forces of the hand in generalized coordinates and J_i is the Jacobian matrix evaluated at $\mathbf{p}_i$. When the object reaches the dropping location, the fingers release the object at once. The dropping location is predefined in the coordinate frame of the hand.

Our algorithm also uses fingers to prevent overshooting and angular deviation about the y-axis of the hand. If an overshooting is detected at the last contact cycle ($E^+ > 0$), all four fingers are commanded to bend with a small angle ($10°$). To correct the angular deviation in the y-axis, we use fingers to create a contact force which induces a torque on the object in the opposite direction of the deviation angle. Depending on the desired direction of the torque, the algorithm selects one of the two extreme points in the x-axis as the point of application on the object (Figure 5). The closest finger to the selected point of application is commanded to bend forward until it strikes the object.

VI. Results

We demonstrate our algorithm by simulating the Shadow Dexterous Hand manipulating a variety of objects in different scenarios. All the motions are simulated using the physics engine DART in Gazebo. DART is a multibody dynamic simulator formulated by Lagrange's equations in generalized coordinates. It handles collision and contact using an implicit time-stepping, velocity-based LCP (linear-complementarity problem) to guarantee non-penetration, directional friction, and approximated Coulomb friction cone conditions.

For all the results shown in the paper and the accompanying video, the simulator integrated at 1000Hz (i.e. the integration time step is 1 millisecond), but the controller was running at much lower frequency as it only sent one command to the robotic hand per contact cycle. The motions shown in the video were recorded in realtime. We set the friction coefficient μ to 1.5 in all the results to prevent slippage. We also tested smaller μ such as 1.0. While the titling angle is thus constrained to be less than $45°$, it is successful for cases such as rolling the cube twice on the palm.

Our experiments showed that the control algorithm was able to manipulate a variety of convex and nonconvex objects, such as a cube, a trapezoidal, or a star-shape prism (nonconvex base polygon, Figure 6). We also tested the algorithm on objects with mass ranging from $0.1kg$ to $1.0kg$, as well as objects with offset center of mass. Figure 7 shows the trajectory of θ when manipulating objects with different physical properties. In all cases, the algorithm was successful in rolling the object across multiple faces as desired, provided that the robotic hand is longer than the required rolling distance. If the object is initially placed closer to finger tips, the same algorithm can roll the object in the negative z-axis direction.

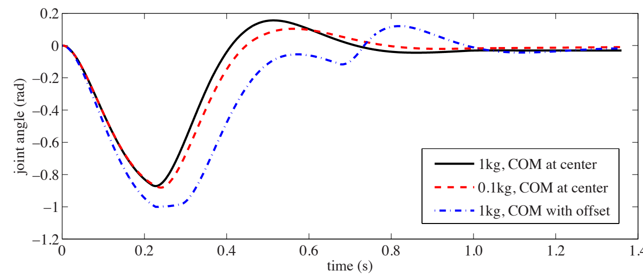

Fig. 7. Trajectory of the wrist angle. Three sequences of rolling a cube across two contacting faces were simulated. Black line: $1kg$ cube with center of mass at the geometry center. Red dashed line: $0.1kg$ cube with center of mass at the geometry center. Blue dot line: $1kg$ cube with offset center of mass.

Two assumptions of the algorithm were relaxed during simulation. First, we allowed the object to have slightly non-parallel joining edges (Figure 8 (a)). We also used objects with rough surfaces instead of analytical shapes considered by the algorithm (Figure 8 (b)). The results showed that the violation of the assumption did not affect rolling significantly and the small errors could be corrected by the fingers. Second, we took into account the noise in the vision sensor input (Figure 8 (c)). To emulate the imperfect vision sensors in the simulation, we added Gaussian noise to the positions of vertices in the world frame and used the corrupted positions to approximate the frame of the object. With the variance of the noise being $5mm$, the successful rate of the control algorithm was still above 80%. More failure cases occurred when we increased the noise. Most failure cases were due to the erroneous center of mass approximation which caused the palm to tilt too early or too late. The example shown in the accompanying video demonstrated one of the failure cases when the approximated center of mass fell behind the actual center of mass.

VII. Discussion and Conclusions

We presented a technique to manipulate the orientation of an object using both palm and fingers of a robotic hand. The experiments with the Shadow Dexterous Hand model showed that the hand was able to pick up a given object on the table, to drop it on a specific spot on the palm, and to let it roll continuously and controllably on the palm, subject to the gravitational and contact forces. We formulated a simple algorithm to control the tilting angle of palm and demonstrated it can be applied to different types of objects

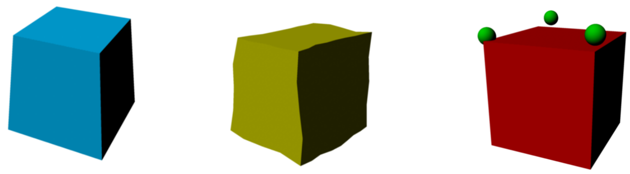

Fig. 8. (a) A non-prism object of which the base polygons are not parallel. (b) An object with rough surfaces. (c) The green dots indicate the three vertices detected by an imperfect vision sensor.

rolling from an initial contacting face to a desired contacting face. Additionally, we developed a corrective controller using fingers to improve the robustness against unexpected collision, irregular object, and noisy vision sensing input. The computation is simple and the controller can run in realtime on a robot.

One limitation of our approach is that the detailed information about the object must be known in advance. This requirement could be problematic for applications when robots need to manipulate unknown objects. Although we tested the algorithm with non-prism and unsmooth shapes, the algorithm is still likely to fail on an object too different from a prism. For example, rolling a key on the palm would be a challenging case. Our algorithm cannot handle objects with curvy surface. Another limiting factor of our algorithm is that the wrist tilting angle is bound by the friction coefficient: $\mu \geq \tan(\theta)$, to prevent slippage.

The proposed technique is general for robotic hand with a palm-like flat surface and finger-like appendages. We expect that the same control algorithm can be applied to other robotic hands. One future direction is to evaluate the algorithm on a physical system and utilize the experimental results to improve the simulation and the control algorithms.

ACKNOWLEDGMENT

The authors would like to thank Jie Tan and Jeff Bingham for their thoughtful discussions and helpful feedback throughout the completion of this work.

REFERENCES

[1] R. R. Ma and A. M. Dollar, “On dexterity and dexterous manipulation,” *2011 15th International Conference on Advanced Robotics (ICAR)*, pp. 1–7, June 2011.

[2] M. T. Mason, “Progress in nonprehensile manipulation,” *The International Journal of Robotics Research*, 1999.

[3] “Gazebo,” 2013. [Online]. Available: http://gazebosim.org/

[4] “DART: Dynamic Animation and Robotics Toolkit,” 2013. [Online]. Available: https://github.com/dartsim/dart/

[5] L. Han and J. Trinkle, “Dextrous manipulation by rolling and finger gaiting,” *Proceedings of 1998 IEEE International Conference on Robotics and Automation*, vol. 1, no. May, pp. 730–735, 1998.

[6] D. Rus, “Dexterous rotations of polyhedra,” *Proceedings of 1992 IEEE International Conference on Robotics and Automation*, pp. 2758–2763, 1992.

[7] A. Bicchi and R. Sorrentino, “Dexterous manipulation through rolling,” *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, vol. 1, pp. 452–457, 1995.

[8] L. Han, Y. Guan, and Z. Li, “Dextrous manipulation with rolling contacts,” *Proceedings of 1997 IEEE International Conference on Robotics and Automation*, pp. 2–7, 1997.

[9] D. Brock, “Enhancing the dexterity of a robot hand using controlled slip,” *Proceedings of 1988 IEEE International Conference on Robotics and Automation*, no. 7, pp. 249–251, 1988.

[10] M. Dogar and S. Srinivasa, “Push-grasping with dexterous hands: Mechanics and a method,” in *Proceedings of 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*, October 2010.

[11] P. Tournassoud, T. Lozano-Perez, and E. Mazer, “Regrasping,” *Proceedings of 1987 IEEE International Conference on Robotics and Automation*, pp. 1924–1928, 1987.

[12] A. Bicchi, “Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity,” *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, pp. 652–662, 2000.

[13] N. Sawasaki, M. Inaba, and H. Inoue, “Tumbling objects using a multi-fingered robot,” *Proceedings of the 20th International Symposium on Industrial Robots and Robot Exhibition*, pp. 609–616, 1987.

[14] M. Erdmann and M. Mason, “An exploration of sensorless manipulation,” *IEEE Journal on Robotics and Automation*, vol. 4, no. 4, pp. 369–379, 1988.

[15] Y. Aiyama, M. Inaba, and H. Inoue, “Pivoting: A new method of graspless manipulation of object by robot fingers,” *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 00, no. C, pp. 136–143, 1993.

[16] W. Huang and M. T. Mason, “Mechanics, planning, and control for tapping,” *International Journal of Robotics Research*, vol. 19, no. 10, pp. 883–894, October 2000.

[17] ——, “Experiments in impulsive manipulation,” *Proceedings of 1998 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1077 – 1082, May 1998.

[18] T. Abell and M. A. Erdmann, “Stably supported rotations of a planar polygonwith two frictionless contacts,” *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1995.

[19] M. A. Erdmann, “An exploration of nonprehensile two-palm manipulation: Planning and execution,” *Proceedings of International Symposium on Robotics Research*, 1995.

[20] N. B. Zumel and M. A. Erdmann, “Nonprehensile two palm manipulation with non-equilibrium transitions between stable states,” *Proceedings of 1996 IEEE International Conference on Robotics and Automation*, pp. 3317–3323, 1996.

[21] M. A. Erdmann, M. T. Mason, and G. Vaněček, “Mechanical Parts Orienting : The Case of a Polyhedron on a Table,” *Proceedings of 1996 IEEE International Conference on Robotics and Automation*, pp. 360–365, 1991.

[22] K. M. Lynch and M. T. Mason, “Dynamic Nonprehensile Underactuated Manipulation,” *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996.

[23] S. Srinivasa, M. A. Erdmann, and M. T. Mason, “Using projected dynamics to plan dynamic contact manipulation,” *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.

[24] C. Sunada, D. Argaez, S. Dubowsky, and C. Mavroidis, “A coordinated jacobian transpose control for mobile multi-limbed robotic systems,” in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*. IEEE, 1994, pp. 1910–1915.