Robotic In-Hand Manipulation for Large-Range Precise Object Movement: The RGMC Champion Solution

Mingrui Yu, Yongpeng Jiang, Chen Chen, Yongyi Jia, and Xiang Li, *Senior Member, IEEE*

***Abstract*—In-hand manipulation using multiple dexterous fingers is a critical robotic skill that can reduce the reliance on large arm motions, thereby saving space and energy. This work focuses on in-grasp object movement, which refers to manipulating an object to a desired pose through only finger motions within a stable grasp. The key challenge lies in simultaneously achieving high precision and large-range movements while maintaining a constant stable grasp. To address this problem, we propose a simple and practical approach based on kinematic trajectory optimization with no need for pretraining or object geometries, which can be easily applied to novel objects in real-world scenarios. Adopting this approach, we won the championship for the in-hand manipulation track at the 9th Robotic Grasping and Manipulation Competition (RGMC) held at ICRA 2024. Implementation details, discussion, and further quantitative experimental results are presented in this work, which aims to comprehensively evaluate our approach and share our key takeaways from the competition. Supplementary materials including video and code are available at `https://rgmc-xl-team.github.io/ingrasp_manipulation`.**

***Index Terms*—Multi-fingered in-hand manipulation, trajectory optimization, Robotic Grasping and Manipulation Competition.**

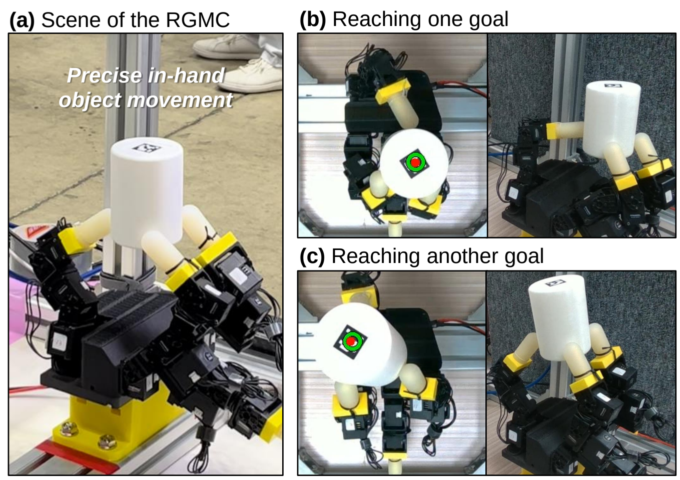


Fig. 1. In-grasp object movement task, where the goal is to manipulate the in-hand object to a desired pose (position) using only finger motions within a stable grasp. (a) Scene of the Robotic Grasping and Manipulation Competition (RGMC) at ICRA 2024, where we won the championship for this task. (b)(c) Precisely moving the object to the desired position in a large in-hand space.

I. Introduction

In-hand manipulation with multi-fingered hands has become increasingly important in recent research on robotic manipulation, as it is crucial for achieving human-level dexterity [1]. Although the advantages of utilizing the high degrees of freedom (DoFs) of multi-fingered hands are attractive, coordinating the fingers to efficiently and robustly manipulate in-hand objects as expected in real-world environments remains a challenging and unresolved issue.

Amid a diverse range of in-hand manipulation tasks, this work focuses on a fundamental task, namely *in-grasp object movement*, which refers to controlling an object's pose (position or orientation) relative to the hand using only finger motions within a stable grasp [2]–[6], as shown in Fig. 1. This task was benchmarked in the in-hand manipulation track of the 9th Robotic Grasping and Manipulation Competition (RGMC) held at ICRA 2024 [7]. Although robot arm motions alone can sometimes achieve similar desired object movements, they consume much more energy than finger motions. Moreover, achieving small desired object movements may require large arm joint motions, which can be problematic in constrained spaces with obstacles.

The authors are with the Department of Automation, Tsinghua University, Beijing, China. Corresponding author: Xiang Li (xiangli@tsinghua.edu.cn).

The significance of *in-grasp* manipulation is that, for many tasks, desired in-hand object motions can be achieved without altering the contact locations. Compared with general in-hand manipulation involving making and breaking contacts, explicitly imposing constant stable grasp constraints can significantly reduce the complexity of problem solving. Moreover, avoiding contact switching can enhance robustness in real-world scenarios, as the complex mechanics of real-world dynamic contacts may exacerbate the sim-to-real (model-to-real) gap.

The challenges in this task stem from several factors. First, a stable grasp with constant contact locations must be strictly maintained throughout the manipulation process; otherwise, the object will fall. Second, the reachable space of the in-grasp object is highly constrained by the limited DoFs (usually ≤ 4) and the joint limit of each finger, especially when the stable grasp constraint is maintained between fingers. Third, real-world task accuracy is affected by imperfect hand kinematics, imperfect hand control, and modeling gap of contacts. Consequently, achieving precise and large-range in-grasp object movement in the real world is non-trivial.

To address the above challenges and achieve deployable multi-fingered in-grasp object movement, this work proposes a simple and practical approach based on trajectory optimization. Specifically, to ensure a constant stable grasp during manipulation and generalization across different goals and objects, we employ geometry-free trajectory optimization with explicit constraints rather than reinforcement learning, which may lack control over accuracy. To expand the object's reachable space, we use global full-trajectory planning and allow rolling contact between the object and all fingertips. Furthermore, to compensate for simplifications in the trajectory optimization and enhance real-world accuracy, we incorporate a closed-loop scheme via re-planning and re-execution.

Our key contributions are highlighted as follows:

1) We propose a simple and practical trajectory optimization approach for multi-fingered in-grasp object movement. The approach does not rely on large-scale training, complex contact models, or even object geometries, making it easily applicable to real-world novel objects. Compared with existing works, it achieves a larger object reachable space while ensuring task accuracy.
2) Beyond the validation of our solution through the RGMC, this work presents a detailed quantitative analysis of our approach through extensive real-world experiments. Moreover, we share the implementation details (including the source code) and practical insights gained from the competition.

Thanks to its accuracy, robustness, and generalizability, our approach won the championship of the in-hand manipulation track of the RGMC [7]. Additionally, our approach was awarded the Most Elegant Solution among all tracks of the RGMC, owing to its concise and novel formulation.

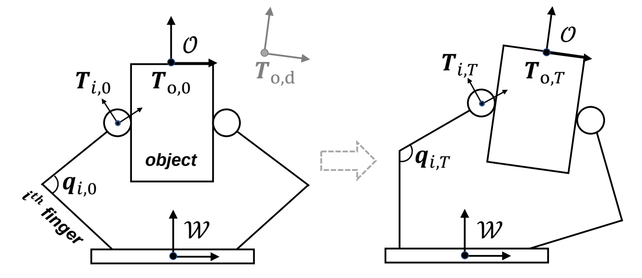
II. Preliminaries

A. Competition Task Setup

The competition task involves two objects: a known cylindrical object and a novel object. The poses of the object are tracked using an attached AprilTag marker. The initial grasp is set by a human operator with no restrictions. For the known object, cylinders of three sizes are provided, and each team can select one based on their hand design. We chose the smallest cylinder with a diameter of 60 mm and a height of 80 mm. The novel object is randomly assigned on-site during the competition. Goal waypoints are given as a sequence of 10 positions relative to the initial object's (AprilTag's) position. After forming the initial grasp, the hand is tasked to autonomously and continuously move the object (AprilTag) through the waypoints one by one. These waypoints are sampled within a $5\times5\times5$ (cm) cubic space centered at the initial object position. Goal object orientations are not assigned in the competition. Each goal waypoint should be reached within a 20-second time budget. The evaluation metric is the accumulated position error across the 10 waypoints. If the object is dropped, it cannot be manually reset to continue with the remaining waypoints. The best of two runs on each object is used to rank the teams. More details of the competition rules can be found in [8].

B. Notations and Definitions

In this work, $\boldsymbol{q}_{i,t}$ represents the joint position vector of the i^{th} finger at time t. For convenience, we denote $\boldsymbol{Q}_t = [\boldsymbol{q}_{1,t};\cdots;\boldsymbol{q}_{n,t}]$, where n is the number of fingers in grasp. The pose of the i^{th} fingertip at time t is denoted as $\boldsymbol{T}_{i,t}$. The object pose at time t is denoted as $\boldsymbol{T}_{\text{o},t}$, with the position and orientation respectively denoted as $\boldsymbol{p}_{\text{o},t}$ and $\boldsymbol{R}_{\text{o},t}$. The desired object pose is denoted as $\boldsymbol{T}_{\text{o,d}}$. The world frame (coordinates) and object frame are denoted as $\mathcal{W}$ and $\mathcal{O}$, respectively. The notations are summarized in Fig. 2. We use $[\boldsymbol{a};\boldsymbol{b}]$ to denote the vertical concatenation of column vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. Following the theory of Lie groups and Lie algebra for robotics [9], we denote the conversion from axis-angle rotation vector $\boldsymbol{r}\in\mathfrak{so}(3)$ to rotation matrix $\boldsymbol{R}\in\mathrm{SO}(3)$ as $\boldsymbol{R}=\exp(\boldsymbol{r}^{\wedge})$ and the inverse conversion as $\boldsymbol{r}=\ln(\boldsymbol{R})^{\vee}$.



(a) Initial grasp (t = 0) **(b)** Goal reaching ($t = T$)

Fig. 2. Formulation of the in-grasp object movement. The objective is to find a hand joint trajectory to move the object, starting from $\boldsymbol{T}_{\text{o},0}$ and reaching $\boldsymbol{T}_{\text{o,d}}$ at time step T while maintaining a constant stable grasp.

In this work, we define the weighted scalar distance d between two poses $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ as

$$d(\boldsymbol{T}_1,\boldsymbol{T}_2,\boldsymbol{W})=\frac{1}{2}\boldsymbol{e}^{\top}\boldsymbol{W}\boldsymbol{e}=\frac{1}{2}\boldsymbol{p}_{\text{e}}^{\top}\boldsymbol{W}_{\text{p}}\boldsymbol{p}_{\text{e}}+\frac{1}{2}\boldsymbol{r}_{\text{e}}^{\top}\boldsymbol{W}_{\text{r}}\boldsymbol{r}_{\text{e}},\quad(1)$$

where $\boldsymbol{e}=[\boldsymbol{p}_{\text{e}};\boldsymbol{r}_{\text{e}}]\in\mathfrak{se}(3)$, in which $\boldsymbol{p}_{\text{e}}=\boldsymbol{p}_1-\boldsymbol{p}_2$ is the position error vector, and $\boldsymbol{r}_{\text{e}}=\ln\left(\exp(\boldsymbol{r}_1^{\wedge})\left(\exp(\boldsymbol{r}_2^{\wedge})\right)^{-1}\right)^{\vee}$ is the rotation error vector. In addition, $\boldsymbol{W}=\text{diag}(\boldsymbol{W}_{\text{p}},\boldsymbol{W}_{\text{r}})$ is a weighting matrix for different dimensions of the errors.

III. Method

A. Trajectory Optimization

We define the full trajectory as a sequence of $T+1$ points. As illustrated in Fig. 2, the objective of the trajectory optimization is to find a hand joint trajectory $\boldsymbol{Q}_{1:T}$ to move the object from the initial configuration ($\boldsymbol{Q}_0$ and $\boldsymbol{T}_{\text{o},0}$) to the desired object pose $\boldsymbol{T}_{\text{o,d}}$ as closely as possible at the end of the trajectory, during which the fingers maintain a stable grasp and avoid self-collisions.

We consider this problem under the following assumptions:

1) The in-hand object is rigid.
2) The initial grasp (defined by $\boldsymbol{Q}_0$ and $\boldsymbol{T}_{\text{o},0}$) is a stable and manipulable grasp.
3) The object surface near the grasp contact points is smooth and exhibits low curvature.
4) The hand and object move at a slow speed, allowing the manipulation process to be treated as quasi-static with negligible inertial effects.

The core idea of our trajectory optimization approach is concise: trying to minimize the terminal object pose error while maintaining constant contact locations. However, constraining the exact contact locations is challenging, as it requires precise initial contact locations and high-fidelity geometries of the object and fingertips, which are hard to obtain in practice. Consequently, we simplify the constant stable grasp constraint as fixing the fingertip positions (i.e., the center of the hemispherical tips) in the object frame $\mathcal{O}$. Note that this simplification implies that we ignore the small changes in contact positions resulting from rolling between the object and fingertips (i.e., rotation of the fingertips in the object frame). We assume that, for objects with low-curvature surfaces, minor changes in contact positions caused by rolling will not significantly affect the manipulation result.

The trajectory optimization problem is specifically formulated as follows. The termination cost regarding the desired object pose is defined as

$$\mathcal{J}_{\text{object}} = d({}^{\mathcal{W}}\boldsymbol{T}_{\text{o},T}, {}^{\mathcal{W}}\boldsymbol{T}_{\text{o,d}}, \boldsymbol{W}_{\text{o}}), \tag{2}$$

where the distance $d(\cdot)$ is defined in (1). If the goal object orientation is not specified, we can assign zero orientation weights to the weighting matrix $\boldsymbol{W}_{\text{o}}$.

The cost for a constant stable grasp is defined as

$$\mathcal{J}_{\text{finger}} = \sum_{t=1}^{T}\sum_{i=1}^{n} d({}^{\mathcal{O}}\boldsymbol{T}_{i,t}, {}^{\mathcal{O}}\boldsymbol{T}_{i,0}, \boldsymbol{W}_{\text{f}}), \tag{3}$$

where ${}^{\mathcal{O}}\boldsymbol{T}_{i,0}$ is the initial fingertip pose in the object frame $\mathcal{O}$. Here, the constant grasp requirement is treated as a soft constraint to avoid strictly infeasible situations. By assigning zero orientation weights to $\boldsymbol{W}_{\text{f}}$, we can fully allow rolling.

Additionally, we include a joint-space penalty to reduce (or restrict) the joint trajectory length and make the waypoints distributed evenly:

$$\mathcal{J}_{\text{joint}} = \lambda \sum_{t=0}^{T-1} \|(\boldsymbol{Q}_{t+1} - \boldsymbol{Q}_t)\|_2^2, \tag{4}$$

where λ is a scalar weight.

The trajectory optimization problem is then formulated as

$$\begin{aligned} \min_{\boldsymbol{Q}_{1:T}, \boldsymbol{T}_{\text{o},1:T}} \quad & \mathcal{J} = \mathcal{J}_{\text{object}} + \mathcal{J}_{\text{finger}} + \mathcal{J}_{\text{joint}} \\ \text{s.t.} \quad & \boldsymbol{Q}^{\min} \preceq \boldsymbol{Q}_t \preceq \boldsymbol{Q}^{\max}, \quad \forall t \in [1, T] \\ & \boldsymbol{F}_{\text{collision}}(\boldsymbol{Q}_t) \preceq \boldsymbol{0}, \quad \forall t \in [1, T], \end{aligned} \tag{5}$$

where the first hard constraint is the joint limit constraint, and the second avoids collisions between fingers. In particular, for the competition, we constrain the distances between four critical points selected on the index and ring fingers. Given the initial stable grasp, the trajectory optimization does not explicitly use the object geometry, allowing it to be applied to novel objects with no need for object reconstruction.

Note that we include the object poses $\boldsymbol{T}_{\text{o},1:T}$ in the optimization variable, different from that in [5] which only included the finger joint angles. This is because the object pose in their work could be represented by the thumb-tip pose under the assumption of rigid thumb-object contact, whereas our object pose cannot be inferred from finger poses due to the rolling contact. Allowing thumb-object rolling enlarges the object's reachable space. The object orientations in the optimization variables are represented by axis-angle rotation vectors $\in \mathfrak{so}(3)$. The analytical gradients, hardware setup, and other implementation details are provided in **Appendix**.

B. Closed-Loop Execution

We adopt a closed-loop execution scheme simply based on re-planning and re-execution to further improve the accuracy. After each iteration of trajectory optimization and actual execution, we plan and execute a new trajectory from the current state to the desired object pose. We repeat this re-planning and re-execution process until any of the following conditions is met: 1) the actual error is smaller than the planned error (i.e., the distance between $\boldsymbol{T}_{\text{o,d}}$ and planned $\boldsymbol{T}_{\text{o},T}$); 2) the predefined maximum number of re-planning attempts N_{replan} is reached; or 3) the time budget is exceeded. This strategy helped us achieve very high precision in the competition.

In addition, the competition requires continuous reaching of a series of waypoints in a large space, which necessitates long-horizon solvability and robustness. Our strategy is to move the fingers back to the initial state (along the forward trajectory) after reaching each waypoint, as the initial state is usually a better start for trajectory optimization to the next goal.

IV. Experimental Results

First, we quantitatively analyze the performance of the proposed approach and the effects of hyper-parameters, using the known cylinder object shown in Fig. 1. Then, we validate the generalization of the approach to novel objects, using the everyday objects shown in Fig. 3. We use the following metrics: 1) planned error: the positional distance between $\boldsymbol{T}_{\text{o,d}}$ and planned $\boldsymbol{T}_{\text{o},T}$ from the initial full trajectory planning (not replanning), and 2) execution error: the distance to the goal after the actual execution.

A. Trajectory Optimization

We first validate the trajectory optimization itself. We choose the eight corners of the $5\times5\times5$ (cm) cubic space as the most representative and challenging goal object positions, and we task the hand to manipulate the cylinder to continuously reach these eight corners for five iterations (giving a total of 40 waypoints without human intervention). No replanning is involved. The planned error, open-loop execution error, and optimization time cost are shown in Fig. 4, where we also explore the effects of the number of trajectory steps T.

The results show that the actual execution errors of the terminal object pose are larger than the planned errors, primarily due to the simplification of rolling in the trajectory optimization and other practical factors; but the open-loop execution errors remain acceptable, averaging less than 1 cm across the 40 waypoints. Second, the different choices of the trajectory steps T have little impact on the execution error. This may be attributed to the physical compliance from the low-level PD controller and the soft fingers, which increase the tolerance of fingertip positions along the trajectory. However, when $T = 1$, we observe that the fingers may excessively compress the object, as the optimization does not account for in-trajectory grasping. Third, the time cost of optimization

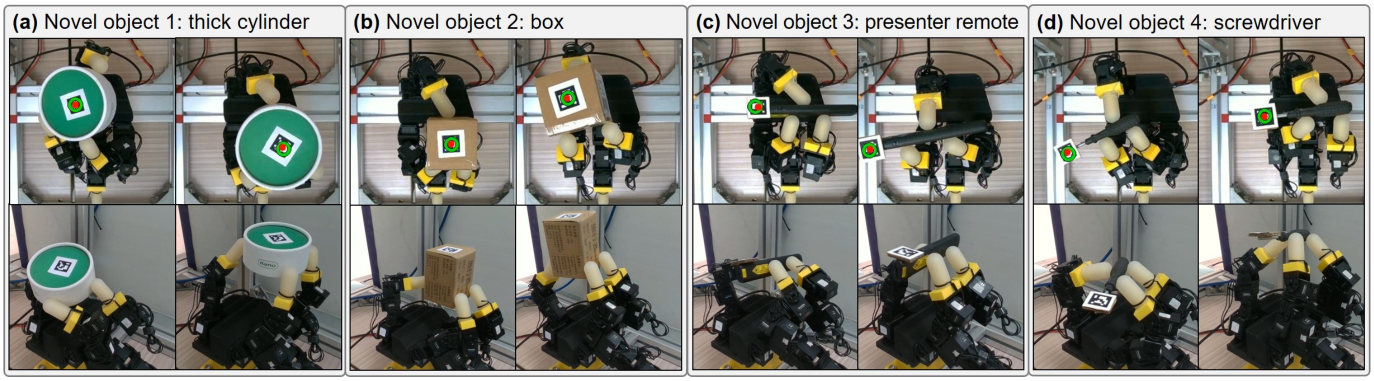


Fig. 3. Experiments of in-grasp object movement with various objects, including a thick cylinder lid, box, presenter remote, and screwdriver. For each object, the images in the first row are from the top-view camera used for object pose tracking, where the red points and green circles represent the AprilTag centers and the desired positions, respectively; the images in the second row are from another camera used only for visualization.

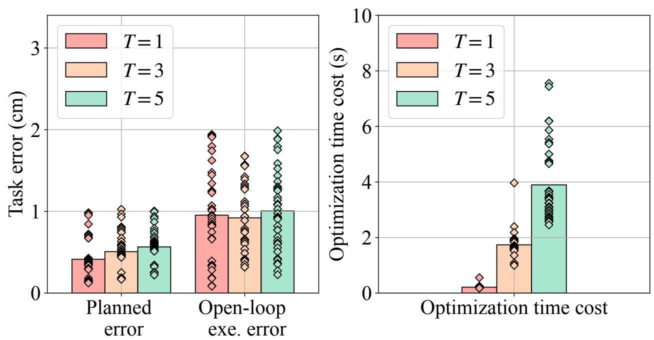


Fig. 4. Evaluation of the trajectory optimization with different numbers of trajectory steps T. Each bar shows the average error/time over 40 waypoints at the corners of the $5 \times 5 \times 5$ (cm) cubic space, and the values for each waypoint are also plotted as the scattered diamond-shaped points.

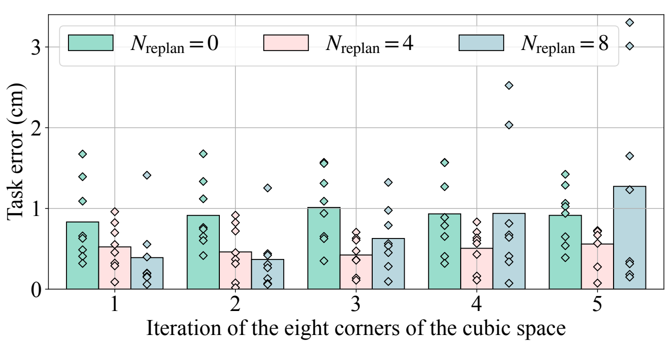


Fig. 5. Evaluation of the closed-loop execution scheme with different maximum re-planning times N_{replan}. Each iteration contains reaching the eight corners of the $5 \times 5 \times 5$ (cm) cubic space, with no human resets between iterations. Each bar shows the average error across the eight corners, and the error for each corner is also plotted as the scattered diamond-shaped points.

increases with the number of trajectory steps T. As a trade-off between resolution and efficiency, we choose $T = 3$ for the competition and subsequent experiments. On average, it takes approximately 2 s to plan a trajectory on a laptop with an Intel i7-9750H CPU (2.6 GHz) and a 16-GB RAM.

B. Closed-Loop Execution

We evaluate the improvements in task accuracy achieved using the closed-loop manipulation scheme. We task the hand to manipulate the cylinder to continuously reach the eight corners of the $5 \times 5 \times 5$ (cm) cubic space for five iterations without human intervention. We also test the effects of the maximum re-planning times allowed N_{replan}.

The results in Fig. 5 indicate that: 1) the closed-loop scheme effectively reduces the final object position errors (from approximately 10 mm to approximately 5 mm); 2) when $N_{\mathrm{replan}} \leq 4$, the system continuously reaches the 40 waypoints without a significant drop in accuracy, demonstrating great long-term robustness; and 3) when $N_{\mathrm{replan}} = 8$, the accuracy in the first and second iteration exceeds that at lower N_{replan}, but the errors in subsequent iterations increase due to slippage between the object and fingertips. Our observations suggest that increased times of re-planning may reduce the contact quality (e.g., slippage or non-tip finger-object contacts). Consequently, as a strategy for the competition, we set $N_{\mathrm{replan}} = 4$ in the first run to ensure more conservative results and $N_{\mathrm{replan}} = 8$ in the second run to aim for higher precision.

C. Additional Evaluations

Appendix contains hyper-parameters, performances competition performances, comparison with the baseline, evaluation of the object reachable space and novel objects, analysis of task error variance, effect of moving back to the initial state, impact of excessive re-planning, effect of cost weight, and experiments with object orientation goals.

V. Conclusion

This work proposes a simple and practical approach for in-grasp object movement via trajectory optimization, which won the ICRA 2024 RGMC in-hand manipulation competition and the Most Elegant Solution award. Our approach is concise and easy to implement, as it requires no pre-training or object geometries. The quantitative experimental results demonstrate that our approach performs effectively and robustly in the real world, as it can continuously reach 40 waypoints in a large $5 \times 5 \times 5$ (cm) in-hand space with an average error of 5 mm. Furthermore, it generalizes well to various everyday objects. We hope that this work provides a comprehensive account of our solution and insights gained from the competition, contributing to future research on dexterous in-hand manipulation with a focus on real-world robustness and practicality.

REFERENCES

[1] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, p. eaat8414, 2019.

[2] K. Hang, W. G. Bircher, A. S. Morgan, and A. M. Dollar, "Hand–object configuration estimation using particle filters for dexterous in-hand manipulation," *Int. J. Robot. Res.*, vol. 39, no. 14, pp. 1760–1774, 2020.

[3] ——, "Manipulation for self-identification, and self-identification for better manipulation," *Sci. Robot.*, vol. 6, no. 54, p. eabe1321, 2021.

[4] J. T. Grace, P. Chanrungmaneekul, K. Hang, and A. M. Dollar, "Direct self-identification of inverse jacobians for dexterous manipulation through particle filtering," in *IEEE Int. Conf. Robot. Autom.* IEEE, 2024, pp. 13 862–13 868.

[5] B. Sundaralingam and T. Hermans, "Relaxed-rigidity constraints: In-grasp manipulation using purely kinematic trajectory optimization," in *Robotics: Science and Systems XIII*, 2017.

[6] ——, "Relaxed-rigidity constraints: kinematic trajectory optimization and collision avoidance for in-grasp manipulation," *Autonomous Robots*, vol. 43, pp. 469–483, 2019.

[7] "9th Robotic Grasping and Manipulation Competition." [Online]. Available: https://cse.usf.edu/~yusun/rgmc/2024.html

[8] "Essential Skill Sub-Track 2: In-Hand Manipulation." [Online]. Available: https://hangkaiyu.github.io/RGMC_in_hand_manipulation_subtrack.html

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

[10] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray *et al.*, "Learning dexterous in-hand manipulation," *Int. J. Robot. Res.*, vol. 39, no. 1, pp. 3–20, 2020.

[11] H. Qi, B. Yi, S. Suresh, M. Lambeta, Y. Ma, R. Calandra, and J. Malik, "General in-hand object rotation with vision and touch," in *Conf. Robot. Learn.* PMLR, 2023, pp. 2549–2564.

[12] T. Chen, M. Tippur, S. Wu, V. Kumar, E. Adelson, and P. Agrawal, "Visual dexterity: In-hand reorientation of novel and complex object shapes," *Sci. Robot.*, vol. 8, no. 84, p. eadc9244, 2023.

[13] J. Pitz, L. Röstel, L. Sievers, D. Burschka, and B. Bäuml, "Learning a shape-conditioned agent for purely tactile in-hand manipulation of various objects," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024.

[14] W. Hu, B. Huang, W. W. Lee, S. Yang, Y. Zheng, and Z. Li, "Dexterous in-hand manipulation of slender cylindrical objects through deep reinforcement learning with tactile sensing," *arXiv preprint arXiv:2304.05141*, 2023.

[15] C. Wang, H. Shi, W. Wang, R. Zhang, L. Fei-Fei, and C. K. Liu, "Dexcap: Scalable and portable mocap data collection system for dexterous manipulation," in *Robotics: Science and Systems (RSS)*, 2024.

[16] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations," in *Robotics: Science and Systems (RSS)*, 2024.

[17] I. Guzey, Y. Dai, B. Evans, S. Chintala, and L. Pinto, "See to touch: Learning tactile dexterity through visual incentives," in *IEEE Int. Conf. Robot. Autom.*, 2024, pp. 13 825–13 832.

[18] C. Chen, P. Culbertson, M. Lepert, M. Schwager, and J. Bohg, "Trajectotree: Trajectory optimization meets tree search for planning multi-contact dexterous manipulation," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 8262–8268.

[19] H. Zhu, A. Meduri, and L. Righetti, "Efficient object manipulation planning with monte carlo tree search," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2023, pp. 10 628–10 635.

[20] X. Cheng, S. Patil, Z. Temel, O. Kroemer, and M. T. Mason, "Enhancing dexterity in robotic manipulation via hierarchical contact exploration," *IEEE Robot. Autom. Lett.*, vol. 9, no. 1, pp. 390–397, 2024.

[21] T. Pang, H. T. Suh, L. Yang, and R. Tedrake, "Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models," *IEEE Trans. Robot.*, 2023.

[22] Y. Jiang, M. Yu, X. Zhu, M. Tomizuka, and X. Li, "Contact-implicit model predictive control for dexterous in-hand manipulation: A long-horizon and robust approach," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024.

[23] W. Jin, "Complementarity-free multi-contact modeling and optimization for dexterous manipulation," *arXiv preprint arXiv:2408.07855*, 2024.

[24] K. Shaw, A. Agarwal, and D. Pathak, "LEAP Hand: Low-cost, efficient, and anthropomorphic hand for robot learning," *Robotics: Science and Systems (RSS)*, 2023.

[25] H. Hartl, "Dextrous manipulation with multifingered robot hands including rolling and slipping of the fingertips," *Robotics Auton. Syst.*, vol. 14, no. 1, pp. 29–53, 1995.

[26] Z. Li, P. Hsu, and S. Sastry, "Grasping and coordinated manipulation by a multifingered robot hand," *Int. J. Robot. Res.*, vol. 8, no. 4, pp. 33–50, 1989.

[27] F. Yang, T. Power, S. A. Marinovic, S. Iba, R. S. Zarrin, and D. Berenson, "Multi-finger manipulation via trajectory optimization with differentiable rolling and geometric constraints," *arXiv preprint arXiv:2408.13229*, 2024.

[28] T. D. Barfoot, *State estimation for robotics*. Cambridge University Press, 2024.

[29] R. S. Jamisola Jr and R. G. Roberts, "A more compact expression of relative jacobian based on individual manipulator jacobians," *Robotics Auton. Syst.*, vol. 63, pp. 158–164, 2015.

[30] D. Kraft, "A software package for sequential quadratic programming," DLR German Aerospace Center — Institute for Flight Mechanics, Koln, Germany, Tech. Rep. DFVLR-FB 88-28, 1988.

[31] M. Yu, B. Liang, X. Zhang, X. Zhu, L. Sun, C. Wang, S. Song, X. Li, and M. Tomizuka, "In-hand following of deformable linear objects using dexterous fingers with tactile sensing," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024.

[32] S. Cruciani, B. Sundaralingam, K. Hang, V. Kumar, T. Hermans, and D. Kragic, "Benchmarking in-hand manipulation," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 588–595, 2020.