Robust Model-Based In-Hand Manipulation with Integrated Real-Time Motion-Contact Planning and Tracking

Yongpeng Jiang

***Abstract*— Robotic in-hand manipulation, involving fingers making and breaking contacts, advances toward human-like dexterity in real-world robotic interactions. While learning-based approaches have recently shown promising performance, they face bottlenecks due to high data requirements and lengthy training times. Although model-based methods have the potential to overcome these limitations, they struggle with efficient online planning and handling modeling errors, which limits their real-world applications. This paper proposes a novel approach for in-hand manipulation that addresses the limitations of both learning-based and model-based methods. The key feature of our approach is the integrated real-time motion-contact planning and tracking, achieved through a hierarchical structure. At the high level, finger motion and contact force references are jointly generated using contact-implicit model predictive control (MPC). At the low level, these combined references are tracked with tactile feedback. Extensive experiments demonstrate that our approach outperforms existing methods in terms of accuracy, robustness, and real-time performance. It successfully completes all 6 challenging tasks in real-world environments, even under significant external disturbances. The full paper and video are available on `https://director-of-g.github.io/in_hand_manipulation_2/`.**

I. Introduction

In-hand manipulation refers to changing the position of grasped objects using fingers, the palm, and external contacts [1], [2],. This capability is essential for enabling versatile and dexterous robotic interaction with the real world [3], [4]. In-hand manipulation can be categorized into in-grasp manipulation [5], where hand-object contacts are maintained, and manipulation involving finger making and breaking contacts (regrasping) [2], [6]. This paper focuses on the latter, which is challenging due to two key aspects. First, modeling errors are unavoidable due to the difficulty of accurately modeling the nonlinear, contact-rich dynamics [7]–[9], compounded by sensor noise and variability in object properties and hand structures. Robust planning and control, incorporating contact state monitoring with tactile feedback, as well as visual and proprioceptive signals, are essential. Second, real-time planning is challenging due to the high degrees of freedom in multi-fingered hands and the need to coordinate numerous contacts. External disturbances and stochastic contact dynamics [10] require fast online re-planning to update contact sequences and recover from perturbations, especially in long-horizon tasks with regrasping.

To address these issues, considerable works have been reported, divided into learning-based and model-based methods. Reinforcement learning (RL) achieves state-of-the-art

The author is with the Department of Automation, Beijing National Research Center for Information Science and Technology, Tsinghua University, China.

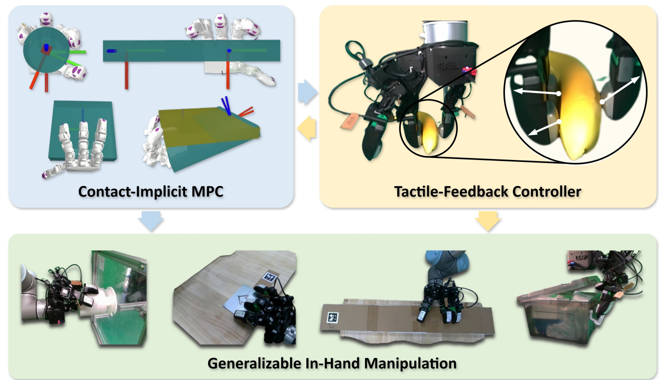

Fig. 1. Overview of the proposed framework for generalizable in-hand manipulation. The framework is model-based and organized as a hierarchical structure. In the high level, a contact-implicit MPC generates real-time motion-contact plans where the fingers make and break contacts. In the low level, a tactile-feedback controller tracks the high-level plans while compensating for the modeling errors by exerting desired contact force.

performance through parallel simulation and domain randomization [11]–[14]. However, RL's generalization requires extensive data, posing significant challenges for further deployment. In contrast, model-based methods offer training-free generalization. These methods can be further categorized into contact-explicit and contact-implicit methods. Contact-explicit methods transform manipulation into a hybrid problem, solving discrete contact sequences and continuous control inputs [15]–[18]. However, to avoid the locality of solutions, problems are often solved considering the complete manipulation sequence, making online re-planning time-consuming and susceptible to perturbations. Contact-implicit methods instead directly plan through contacts without explicitly considering contact sequences [19]–[23]. To ensure real-time performance, contact-implicit methods typically adopt simplified and approximate models, avoiding the use of accurate but difficult-to-solve complementary constraints, which results in imperfect physical fidelity. Consequently, Methods of this kind are susceptible to modeling errors.

This paper addresses the in-hand manipulation problem with regrasping, emphasizing robust, long-horizon manipulation in the presence of external disturbances and significant object pose changes. It proposes a hierarchical framework combining real-time integrated motion-contact planning and tactile-feedback tracking control. At the high level, a contact-implicit model predictive control (MPC) scheme computes reference finger motions and contact forces using a differential dynamic programming (DDP) algorithm and an implicit contact model. At the low level, these references are tracked with MPC-based hybrid force-motion control

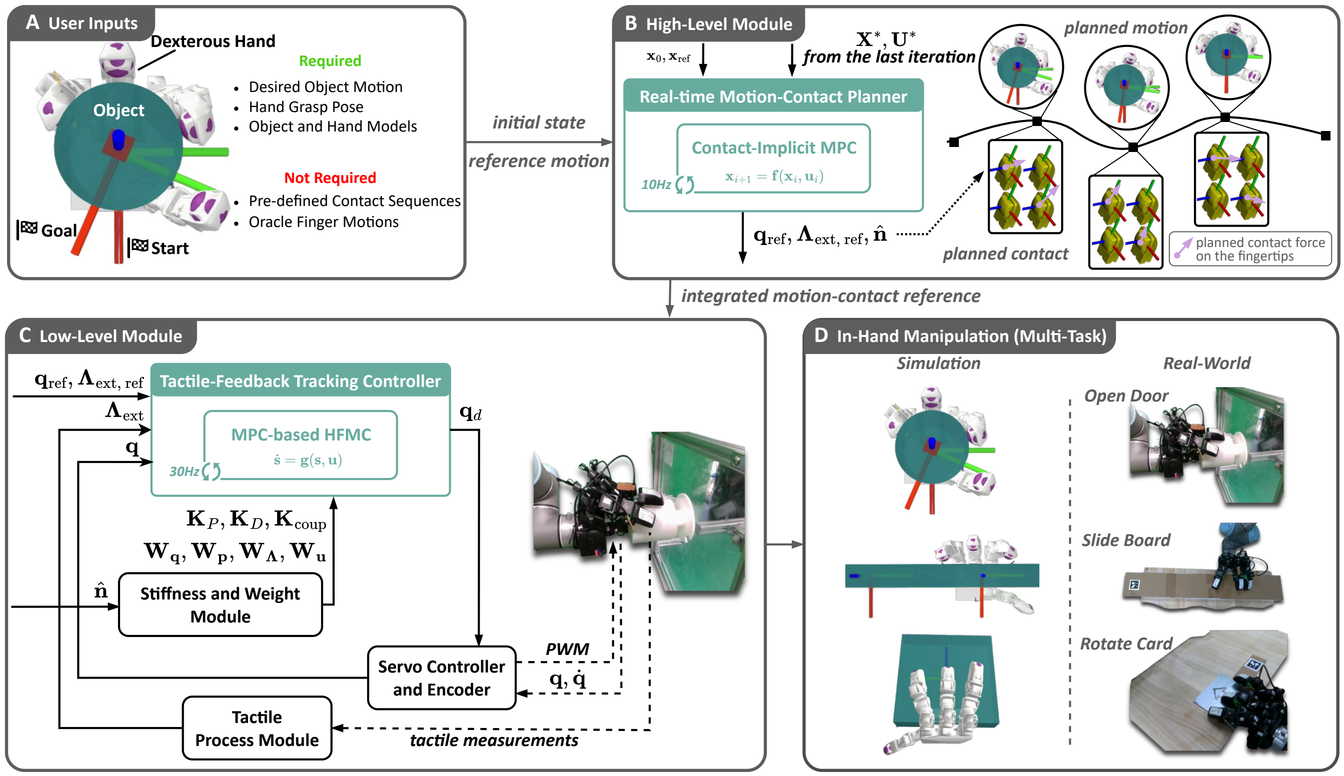

Fig. 2. Proposed integrated motion-contact planning and tracking framework. (A) The user inputs the desired object motion, hand grasp pose, and corresponding models. (B) The high-level real-time motion-contact planner employs contact-implicit MPC to generate motion-contact references from the initial state $\boldsymbol{x}_0$, state reference $\boldsymbol{x}_{\text{ref}}$, and the previous iteration's solution $\boldsymbol{X}^*, \boldsymbol{U}^*$. (C) The low-level tactile-feedback tracking controller uses tactile feedback to track these references jointly. The core algorithm is an MPC-based HFMC. (D) Together, these modules ensure robust and precise in-hand manipulation across multiple tasks.

(HFMC) incorporating tactile feedback. The high-level module enables real-time planning, while the low-level module ensures robust execution and addresses modeling errors like the force-at-a-distance effect[1] caused by modeling errors. We conduct extensive simulations and real-world experiments to validate the accuracy, robustness, and real-time performance of the proposed framework, with the first two metrics outperforming existing methods.

II. Method

This section provides an overview of the proposed framework for dexterous in-hand manipulation, as illustrated in Fig. 2. The framework takes as input the desired object motion, the hand grasp pose, and the models of both the object and hand. It eliminates the need for predefined contact sequences or predetermined finger motions, thanks to real-time planning with implicit contact models. The framework has a hierarchical structure, including a high-level real-time integrated motion-contact planner and a low-level tactile-feedback tracking controller. The two modules run in parallel and are related through the motion-contact references. Note that different dynamics models are used at different levels. At the high level, we are concerned about the full system dynamics with combined motion-contact planning ability. Thus, the smoothed CQDC model $\boldsymbol{f}$ [20] is used. At the low level, we are concerned about the computation efficiency and local force-motion relationship of hand contacts. Thus, a simpler model $\boldsymbol{g}$ is used, which focuses solely on the hand's dynamics without accounting for object movement. In addition, the models are updated with proprioception and object perception.

The high-level planner generates coarse finger motions to establish specific contacts and drive the object to follow the desired motion. Finger motions, contact locations, and contact forces are jointly planned. However, due to modeling errors, the force-at-a-distance effect can occur, leading to insufficient contact force with pure motion tracking. The low-level module addresses modeling errors, including the force-at-a-distance effect, by jointly tracking planned motions and contact forces using tactile feedback. Consequently, the actual motion is adjusted to ensure that the actual contact forces closely match the planned forces. The output of the low-level module is then converted into position commands and sent to the hardware. Please refer to Fig. 3 for a detailed view of the function at different levels. Please refer to the **full paper** for more details.

III. Simulations

We choose the **Rotate Sphere** task to compare different methods. Two representative approaches from existing work were chosen: 1) executing generated finger motions in an

[1] Also refered to as the ”boundary layer” effect in [20]

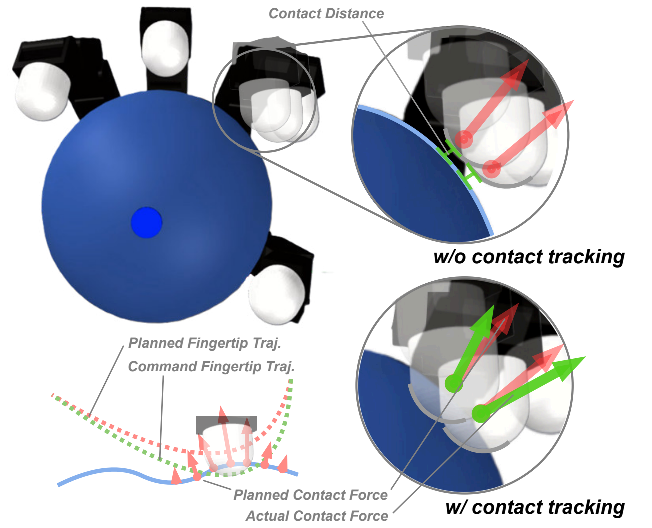

Fig. 3. Detailed view of the proposed framework. The top figures show the high-level integrated motion-contact planning module, which generates real-time finger motions and contact information (visualizing only the index finger and forces). The top-right figure illustrates how modeling errors lead to the force-at-a-distance effect, where non-zero planned forces appear even when contact is inactive. Modeling errors can be mitigated through low-level motion-contact tracking (shown in the bottom-right figure). The bottom-left figure shows how contact tracking is achieved by deforming the command fingertip trajectory. Best viewed in color.

open-loop manner (**openloop**); 2) predictive sampling (PS) using Cross Entropy Methods (**CEM**) or gradient-based methods (**iLQG**). We generate 100 random target orientations, of which the rotation from the initial orientation is no more than 90 degrees. For each target orientation, we record the sphere orientation and joint positions within the first 60s. Please refer to the **full paper** for all results.

1) Results and Discussions: As shown in Tab. I, the proposed method achieves the highest precision and success rate with the smoothest finger motions. Compared with the **openloop** baseline, our method has a lower task error since the tactile-feedback controller tracks desired contact forces and avoids missing contacts. The **openloop** baseline demonstrates poorer performance compared to [20]. This is attributed to the omission of the additional trajectory optimization process, ensuring a fair comparison as the experiments are conducted in real-time. Furthermore, we employ the more accurate MuJoCo simulator instead of a quasi-static simulator[2]. For the same reason, the **openloop** baseline has the lowest task error S.D. Compared with MJPC (CEM), our method has a lower task error, since sampling-based methods have poorer performance especially near convergence. However, MJPC quickly reduces task error, resulting in the shortest task time. This is attributed to the precise dynamics model utilized by MJPC. Besides, our method has much lower task error S.D. and joint acceleration, which show a potential advantage for hardware deployment. MJPC (iLQG) has the worst performance, since the gradients computed through finite difference often vanish for contact-rich manipulation, which indicates the importance of smoothing. Compared to manipulation tasks, MJPC (iLQG) performs relatively better on locomotion tasks [24], [25], as foot contacts are easily established due to gravity and the gradients are typically non-zero.

Remark. *We exclude direct comparisons with learning-based methods due to their limited generalization capability to novel task setups. In contrast, the proposed model-based method seamlessly applies to modified task setups without extensive training. However, we note that RL-based methods with large-scale training typically outperform in highly dynamic tasks [26], [27], which are challenging for the proposed method with simplified dynamics and relatively low control frequency.*

The other simulation results demonstrate that: 1) the proposed framework is robust against sensor noises, owing to the online planning and long-term predictive ability that reduce sensitivity to noisy observations; 2) the proposed framework achieves higher accuracy and lower variance under modeling errors, compared to the predictive sampling baseline; 3) the proposed low-level controller achieves precise tracking of contact forces and exhibits robust generalization capabilities across various object shapes in the static grasping experiment; 4) the proposed low-level controller effectively tracks

[2]In [20], an additional trajectory optimization with a smaller time step refines the planned trajectory to mitigate the "boundary layer" effect. Moreover, [20] employs a custom quasi-static simulator, which is less accurate than alternatives.

TABLE I

COMPARISON BETWEEN DIFFERENT METHODS IN THE SPHERE ROTATION TASK IN MUJOCO SIMULATION.

Methods	Success rate ↑	Average task error (rad) ↓	Average task error of successful cases (rad) ↓	Average task error S.D. of successful cases ($\times 10^{-1}$rad) ↓
ours (planning)[a]	100 / 100	4.954×10^{-5}	4.954×10^{-5}	0.003
ours	**100 / 100**	**0.024**	**0.024**	0.034
openloop	14 / 100	0.644	0.069	**0.004**
MJPC (CEM)	83 / 100	0.099	0.050	4.505
MJPC (iLQG)	14 / 100	0.464	0.084	5.601

Methods	Average task time of successful cases (s) ↓	Average joint acceleration (rad/s^2) ↓	Average control frequency high-level/low-level (Hz)
ours (planning)[a]	28.824	0.247	10.007 / N/A
ours	35.550	**0.618**	9.884 / 30.006
openloop	47.801	0.651	9.456 / N/A
MJPC (CEM)	**21.926**	174.578	99.996 / N/A
MJPC (iLQG)	24.999	198.478	99.998 / N/A

[a] ours (planning) refers to testing the high-level planning module without MuJoCo simulation.

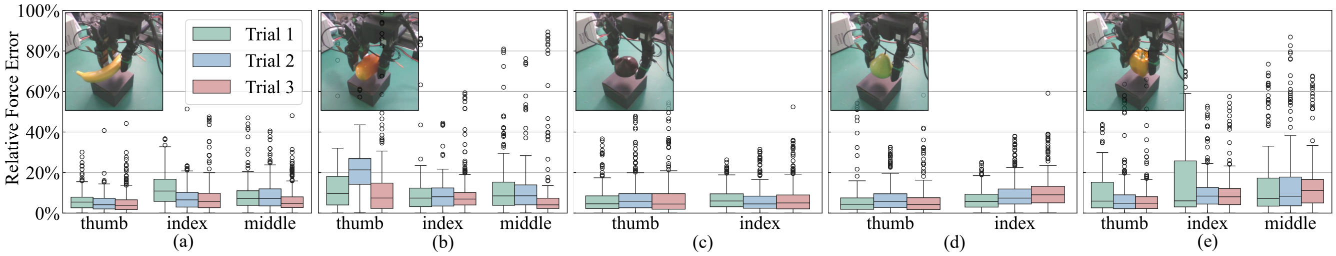

Fig. 4. Relative force tracking error and snapshots of the grasping experiment. The objects are: (a) banana, (b) mango, (c) onion, (d) pear, (e) pepper.

both finger motion and contact forces in the in-hand object movement experiment.

IV. Real-World Experiments

The real-world experiments demonstrate that: 1) the proposed method executes all five tasks—grasping, opening the door, rotating the card, sliding the board, and opening the box—with remarkable precision and robustness; 2) the proposed method converges more quickly and demonstrates a lower orientation error compared to the baseline (without contact tracking) in the task of opening the door. Please refer to the **full paper** for all results.

A. Experiment Setup

We attach the LEAP Hand to the flange of a UR5 arm, which serves as a movable base, and the wrist movements are not utilized during in-hand manipulation. The original fingertips of LEAP Hand are replaced with four vision-based tactile sensors Tac3D [28], which estimate the contact normals and forces at 30Hz.

B. Real-World Grasping Experiment

We test the low-level controller in a real-world grasping experiment, and the results are shown in Fig. 4. We execute grasping 3 times for each of the 5 objects. The relative force tracking error is small and remains consistent between multiple trials, which accords with the simulation. The results demonstrate that the proposed tactile-feedback controller is reliable to track desired contact forces.

C. Experiments of the Open Door Task

We customize a door model with cylindrical handle and a hinge joint to accommodate of the LEAP Hand, as shown in Fig. 5 (a)(b). The task first requires in-hand manipulation to turn the door handle 180 degrees so that the notch is aligned with the door latch, which requires high precision. The hand then pulls open the door as shown in the last column of the snapshots. The rotation of the door handle during four trials is shown in Fig. 5 (c). We exert human interference in Trial 1 and 2, as shown in the peaks. The door handle is turned to the target orientation even under disturbances. The planned and real contact forces during Trial 4 are visualized in Fig. 5 (d)(e). The boxes with solid borders and the shadows filled in corresponds to time intervals where the planned and real contact forces exceed the given threshold $0.1\,\mathrm{N}$ (time intervals that last less than $1\,\mathrm{s}$ are regarded as noises and are neglected). If the planned forces are ideally tracked, the shadows will fill up 100% of the boxes. However, there is an obvious tracking delay especially when the planned forces do not maintain for sometime (i.e., high-frequency oscillation). The reasons for the imperfection include the low sampling frequency and the servo characteristics which are not accurately modeled.

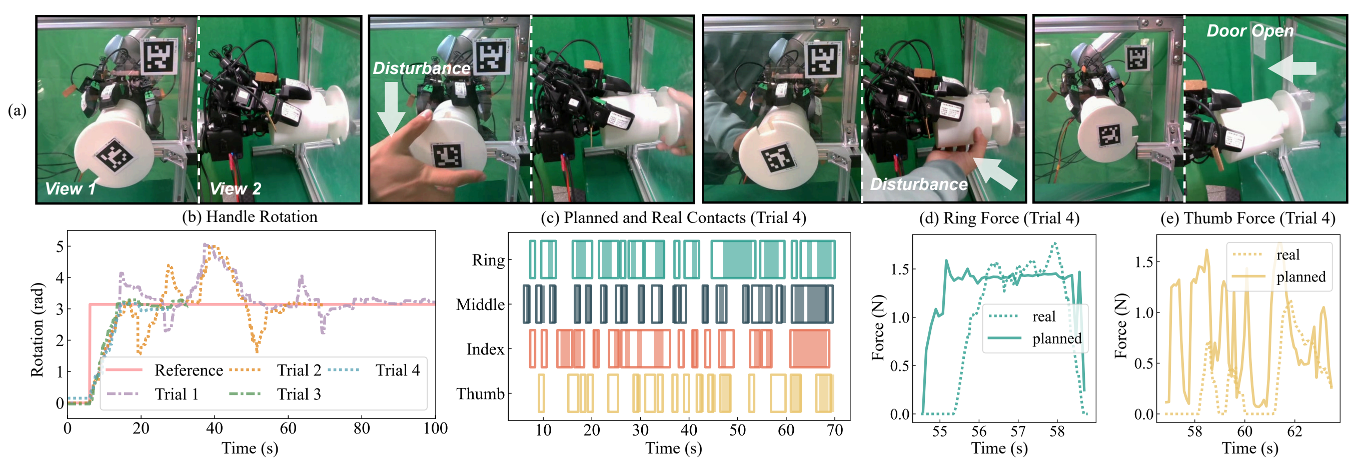

Fig. 5. Experiment results of the open door task with human disturbance. (a) Snapshots from two different views. (b) Door handle rotation of different trials. (c) Planned and real contacts (Trial 4) represented as time intervals where the contact forces exceed a given threshold. (d)(e) The planned and real contact forces.

References

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

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

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

[4] Z. Gu, J. Li, W. Shen, W. Yu, Z. Xie, S. McCrory, X. Cheng, A. Shamsah, R. Griffin, C. K. Liu, *et al.*, "Humanoid locomotion and manipulation: Current progress and challenges in control, planning, and learning," *arXiv preprint arXiv:2501.02116*, 2025.

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

[6] ——, "Geometric in-hand regrasp planning: Alternating optimization of finger gaits and in-grasp manipulation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 231–238.

[7] Q. Le Lidec, W. Jallet, L. Montaut, I. Laptev, C. Schmid, and J. Carpentier, "Contact models in robotics: a comparative analysis," *IEEE Transactions on Robotics*, 2024.

[8] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.

[9] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[10] L. Röstel, J. Pitz, L. Sievers, and B. Bäuml, "Estimator-coupled reinforcement learning for robust purely tactile in-hand manipulation," in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*. IEEE, 2023, pp. 1–8.

[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 *Conference on Robot Learning*. 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," *Science Robotics*, vol. 8, no. 84, p. eadc9244, 2023.

[13] A. Handa, A. Allshire, V. Makoviychuk, A. Petrenko, R. Singh, J. Liu, D. Makoviichuk, K. Van Wyk, A. Zhurkevich, B. Sundaralingam, *et al.*, "Dextreme: Transfer of agile in-hand manipulation from simulation to reality," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5977–5984.

[14] M. Yang, chenghua lu, A. Church, Y. Lin, C. J. Ford, H. Li, E. Psomopoulou, D. A. Barton, and N. F. Lepora, "Anyrotate: Gravity-invariant in-hand object rotation with sim-to-real touch," in *CoRL Workshop on Learning Robot Fine and Dexterous Manipulation: Perception and Control*, 2024. [Online]. Available: https://openreview.net/forum?id=Bt5upUsbvu

[15] S. Cruciani, C. Smith, D. Kragic, and K. Hang, "Dexterous manipulation graphs," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2040–2047.

[16] C. Chen, P. Culbertson, M. Lepert, M. Schwager, and J. Bohg, "Trajectotree: Trajectory optimization meets tree search for planning multi-contact dexterous manipulation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8262–8268.

[17] X. Cheng, S. Patil, Z. Temel, O. Kroemer, and M. T. Mason, "Enhancing dexterity in robotic manipulation via hierarchical contact exploration," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 390–397, 2023.

[18] H. Zhu, A. Meduri, and L. Righetti, "Efficient object manipulation planning with monte carlo tree search," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 10 628–10 635.

[19] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.

[20] 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 Transactions on robotics*, 2023.

[21] V. Kurtz, A. Castro, A. Ö. Önol, and H. Lin, "Inverse dynamics trajectory optimization for contact-implicit model predictive control," *arXiv preprint arXiv:2309.01813*, 2023.

[22] A. Aydinoglu, A. Wei, W.-C. Huang, and M. Posa, "Consensus complementarity control for multi-contact mpc," *IEEE Transactions on Robotics*, 2024.

[23] 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," *arXiv preprint arXiv:2402.18897*, 2024.

[24] T. Howell, N. Gileadi, S. Tunyasuvunakool, K. Zakka, T. Erez, and Y. Tassa, "Predictive sampling: Real-time behaviour synthesis with mujoco," *arXiv preprint arXiv:2212.00541*, 2022.

[25] G. Kim, D.-K. Kang, J. ha Kim, S. Hong, and H.-W. Park, "Contact-implicit model predictive control: Controlling diverse quadruped motions without pre-planned contact modes or trajectories," *The International Journal of Robotics Research*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:266209996

[26] Y. Chen, Y. Geng, F. Zhong, J. Ji, J. Jiang, Z. Lu, H. Dong, and Y. Yang, "Bi-dexhands: Towards human-level bimanual dexterous manipulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 5, pp. 2804–2818, 2023.

[27] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar, "Eureka: Human-level reward design via coding large language models," *arXiv preprint arXiv:2310.12931*, 2023.

[28] L. Zhang, Y. Wang, and Y. Jiang, "Tac3d: A novel vision-based tactile sensor for measuring forces distribution and estimating friction coefficient distribution," *arXiv preprint arXiv:2202.06211*, 2022.