ContactMPC: Towards Online Adaptive Control for Contact-Rich Dexterous Manipulation

Arjun S. Lakshmipathy[1], Nancy S. Pollard[1,2]
[1]Computer Science Department, [2]Robotics Institute, Carnegie Mellon University
{aslakshm, npollard}@andrew.cmu.edu

***Abstract*—Model-based controllers are appealing due to their speed, transparency, and ability to adapt to new situations without strong reliance on extensive previously compiled data. However, online model-based control for dexterous manipulation remains challenging due to the need to navigate a large number of degrees of freedom through complex contact dynamics in real time. In this paper, we show that a simple sampling-based online model-predictive control (MPC) framework can be made to work even for complex multi-step tasks such as picking up a stapler from a table and reconfiguring it in-hand for use. We show that three elements are key to making this approach work in practice. First, a single reference motion is provided, which can come from a source other than the robot. In our case, we provide human motion capture data as reference for control of an Allegro robot hand. Second, contact area information from the reference motion is digested and made part of the loss function during sampling. Third, samples for exploration are taken in PCA basis directions derived from the reference motion. Ablation tests show that all three of these elements are needed to obtain successful results. In particular, sampling without contact information and sampling in the default configuration space of the robot result in uniformly low success rates for multiple tasks. We show examples of the Allegro hand performing several contact-rich motions in simulation, including twisting and pulling a doorknob, lifting an apple, lifting and pouring from a water bottle, and lifting and operating a stapler. We conclude with a discussion of implications for dexterous robotic interaction in the real world.**

I. INTRODUCTION

Dexterous manipulation remains among the most difficult and long-standing grand challenges in robotics. Two challenges in particular contribute significantly to its complexity: constant making, breaking, and intricate evolution of contacts, and controlling the high degrees-of-freedom (DOFs) of dexterous hands performing the tasks.

Despite these complications, there have been impressive demonstrations in recent years utilizing anthropomorphic [1, 17, 4, 8, 3, 19, 14] and non-anthropomorphic [18, 5] dexterous hands. Simpler hand have also been shown performing increasingly complex tasks by exploiting extrinsic dexterity [12, 28] and through reasoning about current and future contact states [6, 29]. However, many of these works require extensive offline computation, exploitation of task-specific assumptions (e.g. up-facing palm, primarily power grasps, constant number of maintained contacts, primitive objects, etc.), or significant upfront engineering costs (e.g. tele-operation).

But what if we could build controllers cheaply from a single arbitrarily sourced manipulation demonstration conducted with a possibly different hand? In particular, demonstrations for human hand manipulations are already publicly available in high quality motion capture datasets [26, 9] and lower quality video datasets [10, 7]; however, adapting such demonstrations to robot hands is a well-known challenge due to the retargeting gap. A common strategy is to first kinematically retarget the human trajectory to the robot hand using fingertip keypoints [22, 8], keyvectors [11, 24], or direct joint angles [2], and then solve for the control policy with the retargeted trajectory as a baseline using imitation [22, 21] or reinforcement [23, 8] learning. But a drawback is that these retargeting techniques often result in significant artifacts such as non-trivial motion misalignment or poor hand-object contact when applied to previously collected data [16], and generating the controller requires considerable compute time and resources. The resulting motion can also sometimes yield unexpected behaviors when rolled out online (e.g. object throwing), which can be challenging to foresee and triage due to the slow development cycle and black-box nature of the learned policy.

We instead propose a sampling-based online MPC approach that explicitly formulates the control policy as the solution to an auto-regressive optimal control problem. Starting from a single motion capture demonstration from the publicly available GRAB dataset [26], we first retarget both the motion trajectory and time-indexed contact distributions following an existing standardized framework [16]. The generated kinematic baseline is then used as the policy seed. Our approach then generates a physically simulated policy using local trajectory improvement, which we demonstrate can be solved online in real time on a single laptop without GPU support. To do so, we propose and evaluate two specific contributions with respect to the standard optimal control problem formulation:

- time-indexed *expected corresponding contact distributions* between the hand and object as a cost term
- a *trajectory improvement exploration strategy* over B-Spline control points of the retargeted principal components computed using the desired robot hand

We show the resulting control problem can be solved over complex, contact-rich manipulation tasks with straightforward max-selection predictive sampling, even under limited time horizons, exploration parameters, and state information variables. We then conclude with a discussion of current limitations and necessary steps to deploy such controllers on real systems.

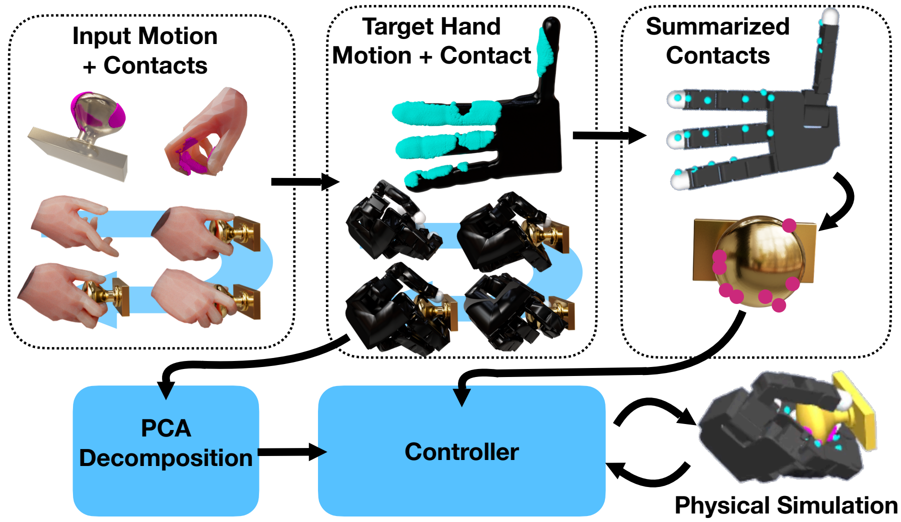

Fig. 1: Overview of our proposed framework. Starting with a single motion capture sequence with corresponding dense contact areas, we retarget both the motion and contact disbributions to obtain a baseline kinematic trajectory. Contact regions are then downsampled to maintain realtime speeds, while the retargeted motion sequence is decomposed into PCs. The motion baseline seeds the controller, while corresponding contact distributions are used to evaluate sampling costs.

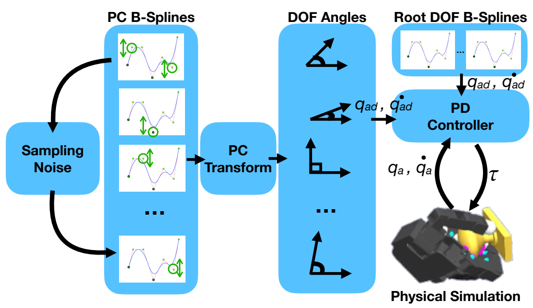

Fig. 2: Detailed overview of our sampling-based controller overview. In our case, samples are used to adjust control points of the B-Splines representing PC-curves. Final joint torques τ are computed by re-assembling the PC values back into DOF space and executing the desired joint angles via PD control.

II. METHODS

Figure 1 illustrates our processing pipeline, while Figure 2 provides a more detailed look at our proposed controller. Our goal is to generate an online control policy for a target hand which can be used to execute a task sourced from a single human motion capture sequence. Our framework can be roughly divided into three stages:

- extract a set of *dense corresponding contact regions* between the human hand and object over the motion sequence
- retarget the contacts and generate a C2 differentiable kinematic trajectory for the target hand
- solve an online optimal control problem to execute the task in physical simulation using the kinematic trajectory as an initial guess

We discuss the stages in the proceeding subsections, with a larger focus on the third stage for this work.

A. *Kinematic Contact and Motion Retargeting*

The first two stages of the pipeline are executed using existing techniques [16, 15]. Using object contacts sourced from the GRAB dataset [26], we compute corresponding human hand contacts using raycasting to guarantee a 1:1 correspondance.

Next, contacts areas are retargeted from the human hand to the robot hand using non-isometric shape matching. Starting from a set of one-time artist-annotated landmarks, we procedurally transfer contacts across the entire time series in a manner that guarantees a 1:1 correspondence while preserving semantic relationships. To reduce the overhead of storing thousands of contacts online, we additionally summarize contact areas by first projecting all points onto the nearest hands links and then taking the mean of the resulting projected distributions per link per timestep. We subsequently obtain corresponding object contact points by performing closest point queries between the summarized link contacts and the object.

Then, using both the retargeted contacts and a set of one-time artist-annotated virtual markers on the back of each hand, we compute a retargeted kinematic trajectory for the target hand. The final motion curves for each hand DOF are fit using B-Splines to both guarantee C^2 continuity and reduce motion sequences, often comprised of hundreds or thousands of frames, to a space of significantly fewer control points that scale with motion complexity rather than duration. For more details on the contact and motion retargeting process, please see existing work [16].

Lastly, we compute the principal components of the resulting retargeted hand trajectory and decompose the original motion curves into a new set of time-indexed “PCA-curves”. We then fit B-Splines to each PCA-curve using the same number of control points as the original curves, which combined serve as the initial guess to our online control problem.

B. Problem Formulation

We assume a standard policy function π which, assuming system state x, agent controls u, and system dynamics f, is defined as $\pi(x_t, u_{t-1}) \rightarrow u_t \ s.t. \ x_{t+1} = f(x_t, u_t)$. In our case, x_t is comprised of the agent configuration $q_a(t)$, agent spatial velocity $\dot{q_a}(t)$, object configuration $q_o(t)$, and object spatial velocity $\dot{q_o}(t)$. The general online control problem can thus be defined as:

$$\begin{aligned} &\underset{x_{1:T}, u_{1:T}}{\operatorname{argmin}} \sum_{t=0}^{T} l(x_t, u_t) \\ \text{s.t.} \quad & x_{t+1} = f(x_t, u_t), \ \text{given } x_0 \end{aligned} \tag{1}$$

where l is the loss function and T is a constant finite time horizon. π is thus the running solution to Eq. 1 such that evaluating π from $t...T$ results in minimal cumulative loss.

We define the loss function as:

$$\begin{aligned} l(x_t, u_t) = \lambda_o ||\hat{q_o}(t) - q_o(t)||^2 + \\ \lambda_c ||c(q_a(t)) - c(q_o(t))||^2 + \lambda_v ||\dot{q_a}||^2 \end{aligned} \tag{2}$$

where $\hat{q_o}(t)$ is the expected object configuration at time t, $c(q_a(t))$ and $c(q_o(t))$ are corresponding time-indexed expected contact distributions on the agent and object respectively, and $\lambda_o, \lambda_c, \lambda_v$ are weighting coefficients.

One notable difference in our formulation is the control signal u. Typically, u directly controls the agent actuators (e.g. DC motors, servos, etc.); however, we instead define u as the function:

$$u_t = \boldsymbol{P}^T \begin{bmatrix} B_M(PC_0)(t) \\ B_M(PC_1)(t) \\ ... \\ B_M(PC_k)(t) \end{bmatrix} + \boldsymbol{p} \tag{3}$$

where $B_M(PC_i)(t)$ represents ith B-Spline PCA-curve evaluated at time t, $\boldsymbol{P}$ is the PC-projection matrix, $\boldsymbol{p}$ is the mean-centering offset vector, and k is the total number of principal components used, which by default is set to the total number of agent DOFs. All B-Splines are cubic, uniform, and represented using a fixed number of control points M pre-determined during the kinematic retargeting process.

C. Online Evaluation

We use max-selection predictive sampling to solve Eq. 1, which is both trivial to implement and allows f to be non-differentiable. For a given sample s taken at time t, we first determine the control points impacted by domain $[t, t+T]$ for each PCA-curve. We then perturb all impacted control point with Gaussian noise and evaluate the perturbed trajectories across all samples S. The sample of lowest cumulative cost on domain $[t, t+T]$ is selected and the corresponding perturbed control points are committed to the nominal policy π. The process is repeated during every agent planning step throughout the manipulation sequence.

Since B-Splines are entirely determined by control point positions, we can run our simulation at arbitrarily small time steps without escalating problem dimensionality; instead, dimensionality scales with spline degree and time horizon length. For a cubic spline, the entire curve shape of a knot segment can be determined by 4 control points; therefore, if T is sufficiently small, the typical problem dimensionality is $4k$. Our method is especially favorable in comparison to the standard dimensionality of $k \cdot \frac{(T-t)}{dt}$, where dt must be carefully selected with respect to the simulation time step.

Finally, the PC-curves from the nominal policy are converted back to u_t using Eq. 3 to compute the desired trajectory in DOF space. We then use PD control to compute the final signals ultimately deployed to the agent.

III. EXPERIMENTS AND EVALUATION

We select 4 tasks for evaluation: doorknob twisting and pulling, apple pickup and hand off, water bottle lifting and pouring, and stapler pickup and wielding. We select the Allegro Hand as the target agent. To avoid workspace discrepencies between human and robot arms, we allow the wrist to float with three linear actuators controlling the position and three motors controlling the orientation synchronized using attitude control. We use 128 predictive samples, a 2.5 second time horizon, and 1 and 10 millisecond timesteps for the simulation and agent planning respectively. We slow down the original human demonstration by a factor between 5 and 10x. We use a standard Gaussian noise kernel $\mathcal{N}(0, \sigma^2)$ with $\sigma^2 = 0.2$ for all PCs and $\sigma^2 = 0$ for root DOFs, effectively limiting sampling only to hand poses. All experiments were run on a single Apple M1 chip using 10 parallel threads, of which 7 are used for predictive sampling. All simulations are run using the Mujoco Engine [27] and MujocoMPC framework [13].

A. Quantitative Evaluation

We consider the time-to-failure metric (TTF) [20] for evaluating policy performance, with longer TTFs indicating better task performance. In our case, a task is considered failed when the hand fails to maintain any contact with the object for 300 milliseconds despite more than six down-sampled expected contacts existing during those timesteps. The assumption then is that the hand has completely lost control of the object and will be unable to recover. Each task was repeated 50 times.

Green boxplots in Figure 3 illustrate the results of our method across all tasks. We also ablate our approach by considering an alternate cost formulation which does not include a contact correspondence term, as well as a more typical sampling strategy over the original joint configuration space with each DOF $\sigma^2 = 60°$. Blue and red boxplots in Figure 3 illustrate results for each ablation respectively. Although we observed that ablations were competitive on "simpler" tasks such as the environmentally constrained doorknob, performance rapidly deteriorated with increasing task

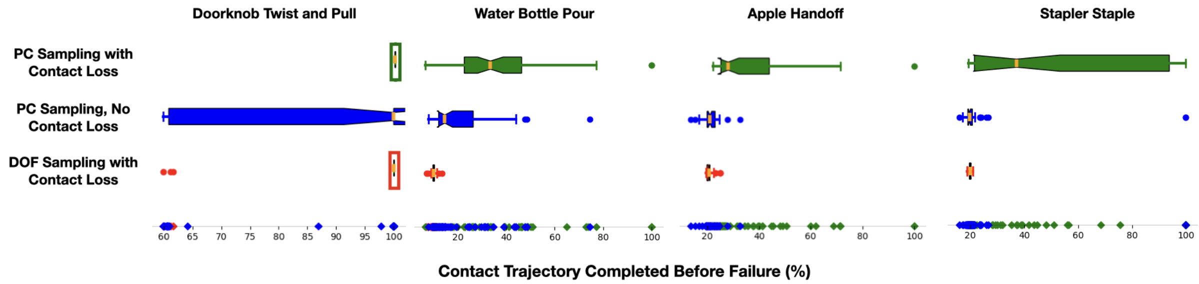

Fig. 3: Plots of time-to-failure metrics across all 4 tasks expressed as a percentage of time between the first and last expected frames of contact per task. Tasks are ordered from left to right by relative complexity in relation to system dynamics, with the doorknob being the simplest and the stapler being the most complex.

complexity. However, the inclusion of our contributions consistently resulted in longer TTFs across the entire task suite.

B. Qualitative Evaluation

Despite plots in Figure 3 indicating non-zero TTFs for ablations, these results are somewhat misleading since contact between the hand and object does not necessarily indicate meaningful task progress. We illustrate this discrepancy qualitatively in the supplementary video. Results in Figure 3 are therefore overestimates of successful execution - in reality, ablating our contact cost term and especially the PC sampling strategy results in extremely poor performance. We also observed and show in the supplementary video that the hand was capable of performing all four tasks to completion under our proposed formulation.

IV. DISCUSSION

We were pleasantly surprised to find that, despite its simplicity, our proposed formulation was capable of executing non-trivial tasks. The massive performance gain in comparison to ablated baseline policies was especially encouraging to observe and strongly validated the importance of our contributions in relation to getting online MPC from a single demonstration, and particularly from data sources that are already publicly available, to actually work.

We also observed interesting emergent behaviors mid-manipulation, such as spidering, sliding, and other non-trivial finger movements that collectively adjusted objects into more favorable configurations. We argue that expected corresponding hand and object contact distributions are largely responsible for encouraging such behavior since said distributions implicitly inform agents *how* to achieve a desired configuration from a deviant intermediate state. Additionally, we saw that sampling over the PC space provides *better quality samples* of meaningful synergies than the original DOF space, which, combined with our B-Spline formulation, helps combat the high dimensionality problem. However, it is currently unclear *why* that is the case outside empirical observation.

But our method is not yet reliable. Although we observed nearly perfect completion on the doorknob task, completion rates for the remaining tasks were less compelling. These results were unsurprising given the substantially higher difficulty of manipulating free-floating objects compared to those which are environmentally constrained. The highly stochastic nature and poor knowledge retention of rollouts from max-selection predictive sampling over long sequences also severely impacts the reliability of online evaluations.

Our method also has yet to be deployed in the real world. There are two main hurdles to address for real-world deployment: online hand-object state estimation, and the fidelity of future state predictions based on real-world dynamics. Fortunately, both problems are long-standing, widely studied, and have rich existing literatures. Although mutual occlusion problems are unavoidable in vision-based tracking approaches for dexterous manipulation, one particularly exciting body of work that has the ability to jointly assist in both of these challenges is tactile sensing, which has already lead to impressive results in online state estimation [25]. Our method is especially well poised to take advantage of such advancements especially since our proposed cost function is fairly simple and does not assume smooth dynamics, require known object mass-inertial properties, or depend on higher order differential terms outside hand velocity. Since our controller also operates fully online, it is feasible to deploy on device immediately without the need for pre-training in simulation and subsequently dealing with Sim2Real headaches. Finally, our framework's design of dealing directly with previously collected data as input has the potential to completely circumvent the need for tele-operation.

V. CONCLUSION AND FUTURE WORK

We have presented an online adaptive control framework for complex, contact-rich dexterous manipulation tasks from a single human hand motion. To do so, we introduced two simple, but powerful contributions within the context of a classical MPC framework: time-indexed corresponding hand-object distributions as an explicit cost term, and a search strategy over the B-Spline control points of hand pose principal components. Our most important next step is to improve the reliability of our method by considering alternative online evaluation strategies and additional potential cost terms. We are excited to see if such modifications will increase the reliability of our results and ultimately provide a path towards real-world deployment.

VI. Acknowledgements

The authors would like to thank Zachary Manchester, Kevin Tracy, and Taosha Fan for helpful discussions, overviews about long-standing and state of the art methods in MPC, and assistance with setting up MujocoMPC. We would also like to thank Vikash Kumar and Sudeep Dasari for helpful suggestions in aligning Mujoco scenes with motion capture demonstrations as well as pointers to the GRAB dataset. Finally, we would like to thank Mustafa Mukadam for helping narrow the initial project scope. This work was primarily conducted while Arjun Lakshmipathy was a visiting researcher at Meta AI research and was partially funded by an AI Research and Mentorship (AIM) award. This material is also based upon work supported by the AI Research Institutes program supported by NSF and USDA-NIFA under AI Institute for Resilient Agriculture, Award No. 2021-67021-35329 and NSF award CMMI-1925130.

References

[1] 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. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

[2] S. P. Arunachalam, I. Güzey, S. Chintala, and L. Pinto. Holo-dex: Teaching dexterity with immersive mixed reality. *arXiv preprint arXiv:2210.06463*, 2022.

[3] V. Caggiano, S. Dasari, and V. Kumar. Myodex: A generalizable prior for dexterous manipulation. 2023.

[4] T. Chen, J. Xu, and P. Agrawal. A system for general in-hand object re-orientation. *Conference on Robot Learning*, 2021.

[5] 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*, 8 (84):eadc9244, 2023. doi: 10.1126/scirobotics.adc9244.

[6] X. Cheng, E. Huang, Y. Hou, and M. T. Mason. Contact mode guided motion planning for quasidynamic dexterous manipulation in 3d. In *International Conference on Robotics and Automation*, pages 2730–2736, 2022. doi: 10.1109/ICRA46639.2022.9811872.

[7] D. Damen, H. Doughty, G. M. Farinella, A. Furnari, J. Ma, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision (IJCV)*, 130:33–55, 2022. URL https://doi.org/10.1007/s11263-021-01531-2.

[8] S. Dasari, A. Gupta, and V. Kumar. Learning dexterous manipulation from exemplar object trajectories and pregrasps. In *IEEE International Conference on Robotics and Automation*. IEEE/RSJ, 2023.

[9] Z. Fan, O. Taheri, D. Tzionas, M. Kocabas, M. Kaufmann, M. J. Black, and O. Hilliges. ARCTIC: A dataset for dexterous bimanual hand-object manipulation. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[10] K. Grauman et al. Ego4d: Around the World in 3,000 Hours of Egocentric Video. In *IEEE/CVF Computer Vision and Pattern Recognition*, 2022.

[11] A. Handa, K. V. Wyk, W. Yang, J. Liang, Y. W. Chao, Q. Wan, S. Birchfield, N. D. Ratliff, and D. Fox. Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9164–9170, 2019.

[12] Y. Hou, Z. Jia, and M. T. Mason. Manipulation with shared grasping. In *Robotics: Science and Systems (RSS '20)*, June 2020.

[13] T. Howell, N. Gileadi, S. Tunyasuvunakool, K. Zakka, T. Erez, and Y. Tassa. Predictive sampling: Real-time behaviour synthesis with mujoco. dec 2022. doi: 10.48550/arXiv.2212.00541. URL https://arxiv.org/abs/2212.00541.

[14] 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, 2024.

[15] A. S. Lakshmipathy, N. Feng, Y. X. Lee, M. Mahler, and N. S. Pollard. Contact edit: Artist tools for intuitive modeling of hand-object interactions. *ACM Transactions on Graphics*, 42(4), July 2023. ISSN 0730-0301. doi: 10.1145/3592117. URL https://doi.org/10.1145/3592117.

[16] A.S. Lakshmipathy, J.K. Hodgins, and N.S. Pollard. Kinematic motion retargeting for contact-rich anthropomorphic manipulations. 2024. doi: 10.48550/arXiv.2402.04820. URL https://arxiv.org/abs/2402.04820.

[17] T. Lin, Y. Zhang, Q. Li, H. Qi, B. Yi, S. Levine, and J. Malik. Learning visuotactile skills with two multifingered hands. *arXiv:2404.16823*, 2024.

[18] A. S. Morgan, K. Hang, B. Wen, K. E. Bekris, and A. M. Dollar. Complex in-hand manipulation via compliance-enabled finger gaiting and multi-modal planning. *IEEE Robotics and Automation Letters*, 2022.

[19] T. Pang, H. J. 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*, 39(6), 2023. doi: 10.1109/TRO.2023.3300230.

[20] H. Qi, A. Kumar, R. Calandra, Y. Ma, and J. Malik. In-hand object rotation via rapid motor adaptation. In *Conference on Robot Learning (CoRL)*, 2022.

[21] Y. Qin, Y. H. Wu, S. Liu, H. Jiang, R. Yang, Y. Fu, and X. Wang. Dexmv: Imitation learning for dexterous manipulation from human videos. In *European Conference on Computer Vision*, 2021.

[22] Y. Qin, H. Su, and X. Wang. From one hand to multiple hands: Imitation learning for dexterous manipulation from single-camera teleoperation. *IEEE Robotics and Automation Letters*, 7:10873–10881, 2022.

[23] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani,

J. Schulman, E. Todorov, and S. Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Robotics: Science and Systems (RSS)*, 2018.

[24] A. Sivakumar, K. Shaw, and D. Pathak. Robotic telekinesis: Learning a robotic hand imitator by watching humans on youtube. In *Proceedings of Robotics: Science and Systems (RSS)*, 2022.

[25] S. Suresh, H. Qi, T. Wu, T. Fan, L. Pineda, M. Lambeta, J. Malik, M. Kalakrishnan, R. Calandra, M. Kaess, J. Ortiz, and M. Mukadam. Neural feels with neural fields: Visuo-tactile perception for in-hand manipulation. In *arXiv*, 2023.

[26] O. Taheri, N. Ghorbani, M. J. Black, and D. Tzionas. GRAB: A dataset of whole-body human grasping of objects. In *European Conference on Computer Vision (ECCV)*, 2020. URL https://grab.is.tue.mpg.de.

[27] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *International Conference on Intelligent Robots and Systems*. IEEE/RSJ, 2012. doi: 10.1109/IROS.2012.6386109.

[28] W. Zhou and D. Held. Learning to grasp the ungraspable with emergent extrinsic dexterity. In *Conference on Robot Learning (CoRL)*, 2022.

[29] W. Zhou, B. Jiang, F. Yang, C. Paxton, and D. Held. Hacman: Learning hybrid actor-critic maps for 6d non-prehensile manipulation. 2023.