



Hierarchical Tactile-Based Control Decomposition of Dexterous In-Hand Manipulation Tasks

Filipe Veiga^{1*}, Riad Akrou² and Jan Peters^{2,3}

¹ Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology, Cambridge, MA, United States, ² Intelligent Autonomous Systems, Technische Universität Darmstadt, Darmstadt, Germany, ³ Max-Planck-Institut für Intelligente Systeme, Tübingen, Germany

OPEN ACCESS

Edited by:

Subhadeep Chakraborty,
The University of Tennessee,
United States

Reviewed by:

Aravinda Ramakrishnan Srinivasan,
University of Lincoln, United Kingdom
Bin Fang,
Tsinghua University, China

*Correspondence:

Filipe Veiga
fveiga@mit.edu

Specialty section:

This article was submitted to
Sensor Fusion and Machine
Perception,
a section of the journal
Frontiers in Robotics and AI

Received: 19 December 2019

Accepted: 15 October 2020

Published: 19 November 2020

Citation:

Veiga F, Akrou R and Peters J (2020)
Hierarchical Tactile-Based Control
Decomposition of Dexterous In-Hand
Manipulation Tasks.
Front. Robot. AI 7:521448.
doi: 10.3389/frobt.2020.521448

In-hand manipulation and grasp adjustment with dexterous robotic hands is a complex problem that not only requires highly coordinated finger movements but also deals with interaction variability. The control problem becomes even more complex when introducing tactile information into the feedback loop. Traditional approaches do not consider tactile feedback and attempt to solve the problem either by relying on complex models that are not always readily available or by constraining the problem in order to make it more tractable. In this paper, we propose a hierarchical control approach where a higher level policy is learned through reinforcement learning, while low level controllers ensure grip stability throughout the manipulation action. The low level controllers are independent grip stabilization controllers based on tactile feedback. The independent controllers allow reinforcement learning approaches to explore the manipulation tasks state-action space in a more structured manner. We show that this structure allows learning the unconstrained task with RL methods that cannot learn it in a non-hierarchical setting. The low level controllers also provide an abstraction to the tactile sensors input, allowing transfer to real robot platforms. We show preliminary results of the transfer of policies trained in simulation to the real robot hand.

Keywords: tactile sensation and sensors, robotics, in-hand manipulation, hierarchical control, reinforcement learning

1. INTRODUCTION

Dexterous in-hand manipulation is a long studied problem, involving precise movement, inter-finger coordination, and contact management (Okamura et al., 2000). While manipulating objects within a grip is possible with simple grippers, external forces such as gravity or interactions with the environment are necessary to generate the manipulation movements (Dafle et al., 2014; Chavan-Dafle and Rodriguez, 2015; Stork et al., 2015). When considering dexterous hands, the problem complexity greatly increases (Ma and Dollar, 2011), as the additional fingers allow for an increased number of possible solutions for each manipulation action and a larger number of possible interactions with objects. Traditional in-hand manipulation control approaches tackle simplifications of the general problem by attempting small movements or by relying on several strong assumptions regarding contact and the precision of the available robot and object models (Maekawa et al., 1995; Zheng et al., 2000; Bai and Liu, 2014). Even with such simplifications, experiments on real robot platforms are prohibitively hard and thus frequently omitted in the literature (Zheng et al., 2000; Bai and Liu, 2014). Seeing in-hand manipulation as a planning or

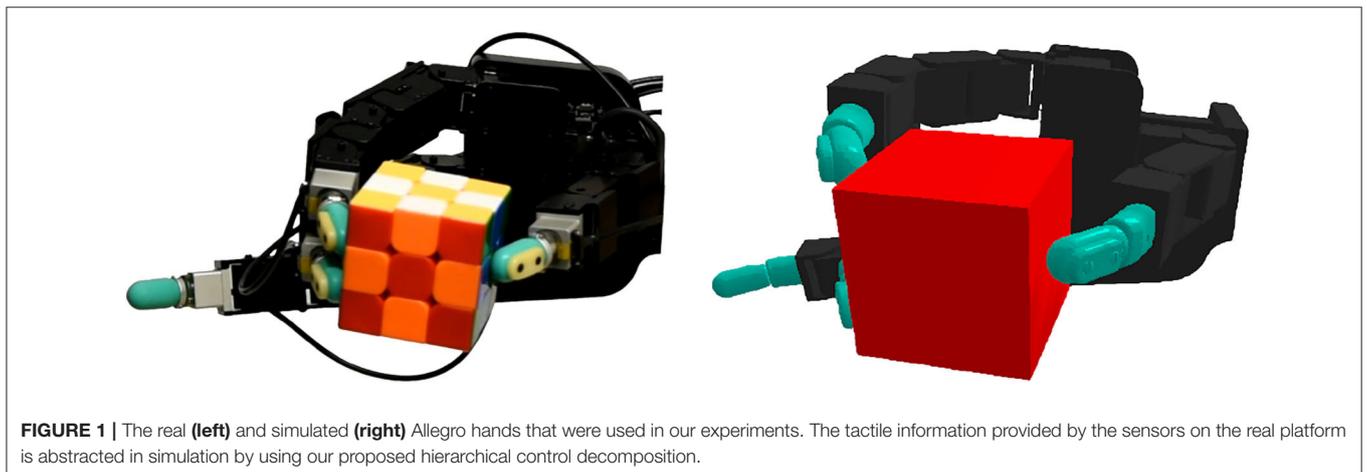
optimization problem provides solutions for more general forms of the problem but most of these solutions integrate very little to no feedback (Cherif and Gupta, 1999; Saut et al., 2007; Mordatch et al., 2012; Sundaralingam and Hermans, 2018). Considering feedback during task execution is crucial to tackle the variability introduced by objects, in the form of distinct shapes, surface properties, target movements, or initial grasp configurations. To achieve a sufficiently general solution to in-hand manipulation, manipulation controllers not only have to generate suitable trajectories that take into account task variability but also have to adapt in accordance with the feedback signals observed during task execution to compensate for unforeseen events, such as object slip.

Tactile sensing is an attractive form of feedback for in-hand manipulation, as it provides information directly from the interaction points. It offers substantial advantages over other forms of feedback such as vision and force, by disregarding effects such as occlusion while providing rich information at high frequencies (Yousef et al., 2011). Additionally, tactile information has been shown to help with the interaction variability required for in-hand manipulation, as it enabled objects to be grasped robustly regardless of their shape or material properties (Veiga et al., 2020). However, integrating high dimensional tactile feedback signals in the control loop of an already complex in-hand manipulation controller is non-trivial.

Reinforcement Learning (RL) has found great success in solving control tasks with large input spaces on both simulated (Mnih et al., 2015; Silver et al., 2016) and physical platforms (Levine et al., 2016). Thus, several approaches based on reinforcement learning (Van Hoof et al., 2015; Popov et al., 2017; Akkaya et al., 2019; Zhu et al., 2019; Andrychowicz et al., 2020), learning from demonstration (LfD) (Li et al., 2014), combinations of RL and LfD (Gupta et al., 2016; Rajeswaran et al., 2017), or optimal control with learned local models (Kumar et al., 2016) have been proposed for in-hand manipulation. Despite this, when considering complex tactile sensors such as the BioTac (Wettels et al., 2014), the richness of the feedback signals leads to considerably more complex state spaces and transition functions, yielding significantly more challenging RL problems. For example, pressure on different contact points

of the BioTac sensors is measured from the displacement of fluid within the fingertip which results from the deformation of its malleable skin. Such a complex physical process is currently impossible to simulate accurately and efficiently. Hence any RL policy learned in simulation using a model of the tactile sensor would not transfer to a physical robot. On the other hand, learning the task directly on the robot is hardly feasible because (i) in-hand manipulation tasks are contact-rich, which creates non-linearities in the state transitions and precludes the learning of a forward dynamics model in a model-based RL setting and (ii) the high dimensionality of the tactile sensors precludes the use of model-free RL directly on the robot due to a prohibitively high sample complexity. Accordingly, only Van Hoof et al. (2015) use RL with integrated tactile information by training the policy directly on a real robot and using very simple tactile information. Additionally, the task is constrained during training by having the object supported by an external surface that prevents it from falling.

Constraining the manipulation task to a position where the object is in a supported position (either by the palm of the robot or by an external support surface), such that the object is less likely to be dropped during exploration, is common among several proposed approaches (Van Hoof et al., 2015; Kumar et al., 2016; Rajeswaran et al., 2017; Akkaya et al., 2019; Andrychowicz et al., 2020). The use of such a constrain is justified by the complex nature of the transition function of in-hand manipulation tasks, even in simulated environments. Indeed, if the robot is holding an object as in **Figure 1**, any exploratory action (a random perturbation to the current joint position) is likely to make the object fall and thus terminates the trajectory after only a few number of steps. We observed that such exploratory behavior could lead to premature convergence of RL to poor local optima where the robot reinforces behaviors that throw the object toward the target. This results in a short term accumulation of rewards at the detriment of the longer term rewards. In addition, we observed that methods such as the ones used in Andrychowicz et al. (2020) to produce impressive results on a real robot with the object supported by the palm, are unable to learn the task when the support is removed.



To address both the learning of in-hand manipulation tasks with rich tactile feedback and the relaxing of the support constraint, we propose a hierarchical control decomposition that relies on a low-level control scheme, composed by a set of independent stabilization controllers, to keep the object firmly gripped during the manipulation actions. These low level stabilization controllers have the benefits of (i) enabling the efficient learning of complex in-hand movements in simulation by maintaining the object in the robot’s hand for a longer period of time, simplifying the transition function and (ii) potentially allowing transfer of policies learned in simulation to physical environments by abstracting tactile information and letting the upper level policy be solely defined on joint information. The stabilization controllers are highly inspired by neurophysiological studies (Johansson, 1996; Flanagan et al., 2006) and have been extensively studied in prior work (Veiga et al., 2020). We show that with the proposed hierarchical decomposition RL methods are able to learn complex and generalizable manipulation actions.

2. HIERARCHICAL CONTROL DECOMPOSITION FOR IN-HAND MANIPULATION

In order to learn general manipulation policies in simulation, that can transfer to a physical robot, we propose a hierarchical control decomposition composed of two control levels: a set of grip stabilization controllers running independently on each finger and a manipulation movement policy that produces the movement trajectory in joint space and trades-off between manipulation and stabilization. We begin by defining the RL problem in a non-hierarchical fashion, followed by a description of the stabilizers that compose the low-level of our proposed hierarchical decomposition and showcase the differences between the non-hierarchical and the hierarchical learning problems.

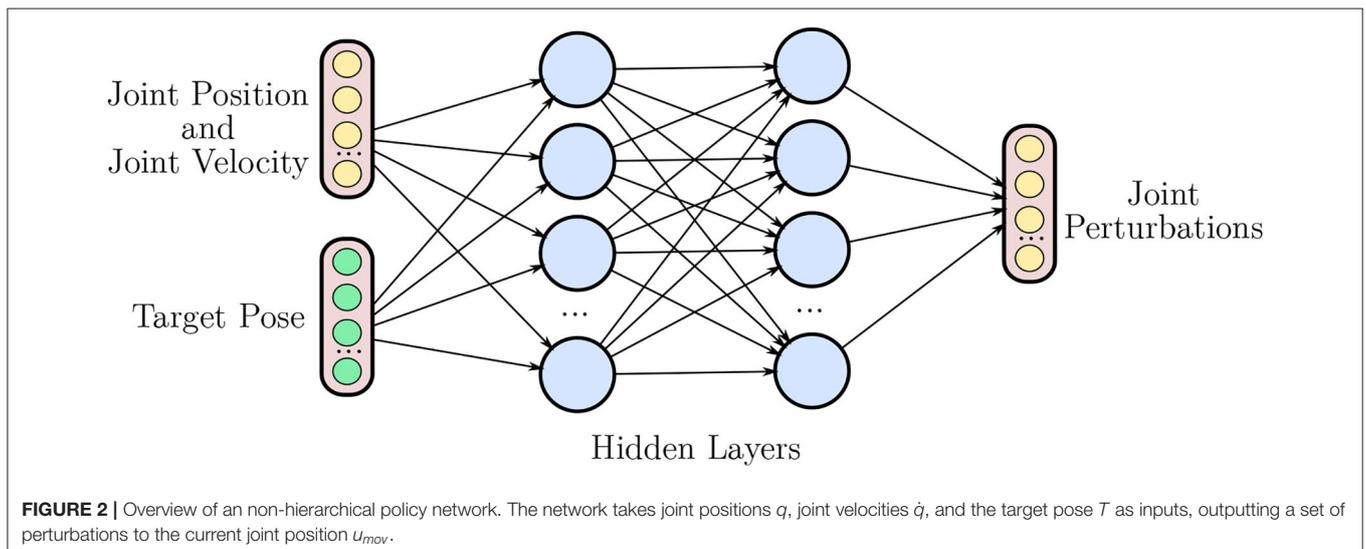
2.1. Reinforcement Learning Problem Definition

Given an initial grasp on an object, we consider the in-hand manipulation task of translating and/or rotating the object to a target pose. We phrase this problem as a Markov Decision Process (MDP), defined by the quintuple $(\mathcal{S}, \mathcal{A}, R, P, \gamma)$, where \mathcal{S} represents the state space, \mathcal{A} the action space, $P(s_{t+1}|s_t, a_t)$ the transition probability, $R(s_t, a_t)$ its associated reward, and γ is the discount factor. In a non-hierarchical RL setting (NH-RL), the state space is comprised of joint positions q , joint velocities \dot{q} and target pose T . The action space is the set of perturbations to the current joint position u_{mov} , constrained by a maximum tolerated velocity. The structure of the non-hierarchical neural network policies is depicted in **Figure 2**. The reward $R(s_t, a_t)$ is inversely proportional to the distance between the current and target object coordinates.

Let π be a stochastic policy giving the probability $\pi(a|s)$ of executing action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$. Let the Q-function be $Q_\pi(s, a) = \mathbf{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) | s_0 = s, a_0 = a]$, where the expectation is taken w.r.t. all random variables s_t and a_t for $t > 0$. Let $V_\pi(s) = \mathbf{E}_{a \sim \pi}[Q_\pi(s, a)]$. The goal of RL is to find the policy maximizing the policy return $J(\pi) = V_\pi(s_0)$ where s_0 denotes some initial state (an initial grasp in our case).

2.2. Independent Grip Stabilization Control

The stabilization controllers that compose the base control level where introduced in Veiga et al. (2020), and are deployed on each finger independently. By interpreting the tactile signals provided by the BioTac sensors (Wettels et al., 2014), these independent finger stabilizers (IFS) locally avoid predicted slip events. This allows them to keep objects stable within multi fingered grips while not being constrained to a particular grasp configuration or a particular distribution of force between the fingers. In a hierarchical setting, the main task of the stabilizers is to ensure grip stability throughout the manipulation action. Formally, provided with a label $c_{t+\tau_f} \in [\text{slip}, \neg\text{contact}, \text{contact}]$ from a



learned tactile based slip predictor, where τ_f is the prediction window of the predictors, the level of a leaky integrator at time t , denoted y_t , is adjusted as follows

$$y_t = \beta y_{t-1} + (1 - \beta)L \quad (1)$$

where β is the leakage at each time step and

$$L = \begin{cases} 1 & \text{if } c_{t+\tau_f} = \text{slip,} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

is the integrator input. The integrator value is then used by the stabilizer to regulate the desired task-space velocity in the contact normal direction, i.e.,

$$\mathbf{v}_{\text{stab}} = \mathbf{N}_t y_t, \quad (3)$$

where \mathbf{N}_t is a unit vector pointing in the contact normal direction. In short, the integrator changes with the predicted contact state, accumulating its response when slip is predicted and leaking if contact. Finally, the stabilization disturbances to the joint positions of the i -th finger $\mathbf{u}_{\text{stab}}^i$ are calculated using inverse kinematics.

There are three differences in implementation pertaining to these controllers between the simulated and real robot environments. The first, is the manner in which the normal contact direction is acquired. In simulation, the contact normal is acquired via the simulator's collision engine while the real robot estimates it via the weighted average of the normal directions of the electrodes. The weights are the activations of each electrode as described in Wettels et al. (2009). The second difference concerns the intensity of the stabilizer response. Due to fluctuations of the fluid of the real sensors, pressure values might indicate that there is no longer contact for one time step, creating jerky responses. As in Veiga et al. (2020), the controllers of the real robot do not immediately stop whenever contact is lost, but have their response smoothly reduced over a period of 200 ms. The final difference concerns the slip signals used by the stabilizers. In simulation, slip signals are provided by a heuristic based slip detector, that observes the changes in relative position and orientation between the fingertip and the object to detect slip. In the real robot, slip is predicted from learned tactile based slip predictors, as described in Veiga et al. (2020), and a prediction window τ_f of ten is used.

The stabilization controllers are independent of the nature of the manipulation task (e.g., nature of the manipulated object, target coordinates, or type of initial grasp) and do not need to be learned. Most importantly, they provide an abstraction to the tactile information provided by the sensors, allowing the high level movement policy to not depend on tactile information while the overall system still reacts to tactile feedback. Being able to learn movement policies with information that is readily available to both the simulated and the real robot facilitates the transfer of policies between the two.

2.3. In-Hand Manipulation Movement Policy

To generate the manipulation movements, a high-level policy π_θ , parameterized by the weights of a neural network θ , is learned in

a simulation environment depicted in **Figure 1**. In contrast to the NH-RL case, in the hierarchical RL (H-RL) setting, the new state space \mathcal{S}' is not only comprised of joint positions q , joint velocities \dot{q} and target pose T but also includes the state $\mathbf{y} = [y^1, \dots, y^4]$ and the state variations $\Delta \mathbf{y} = [y_t^1 - y_{t-1}^1, \dots, y_t^4 - y_{t-1}^4]$ of all the finger stabilizers. The action space is also different, with the new action space $\mathcal{A}' = \mathcal{A} \times [0, 1]^{N_{\text{fing}}}$, now including a set of N_{fing} uni-dimensional merging coefficients α_i , where N_{fing} is the total number of fingers, in addition to the movement commands in the form of perturbations to the hand's joint positions \mathbf{u}_{mov} , that were already included in the action space \mathcal{A} . The merging coefficients α_i regulate the combination of both perturbations, $\mathbf{u}_{\text{stab}}^i$ and \mathbf{u}_{mov} , to compose the final action. Letting \mathbf{u}_i^* be the combined response of each individual finger

$$\mathbf{u}_i^* = \alpha_i \mathbf{u}_{\text{mov}}^i + (1 - \alpha_i) \mathbf{u}_{\text{stab}}^i. \quad (4)$$

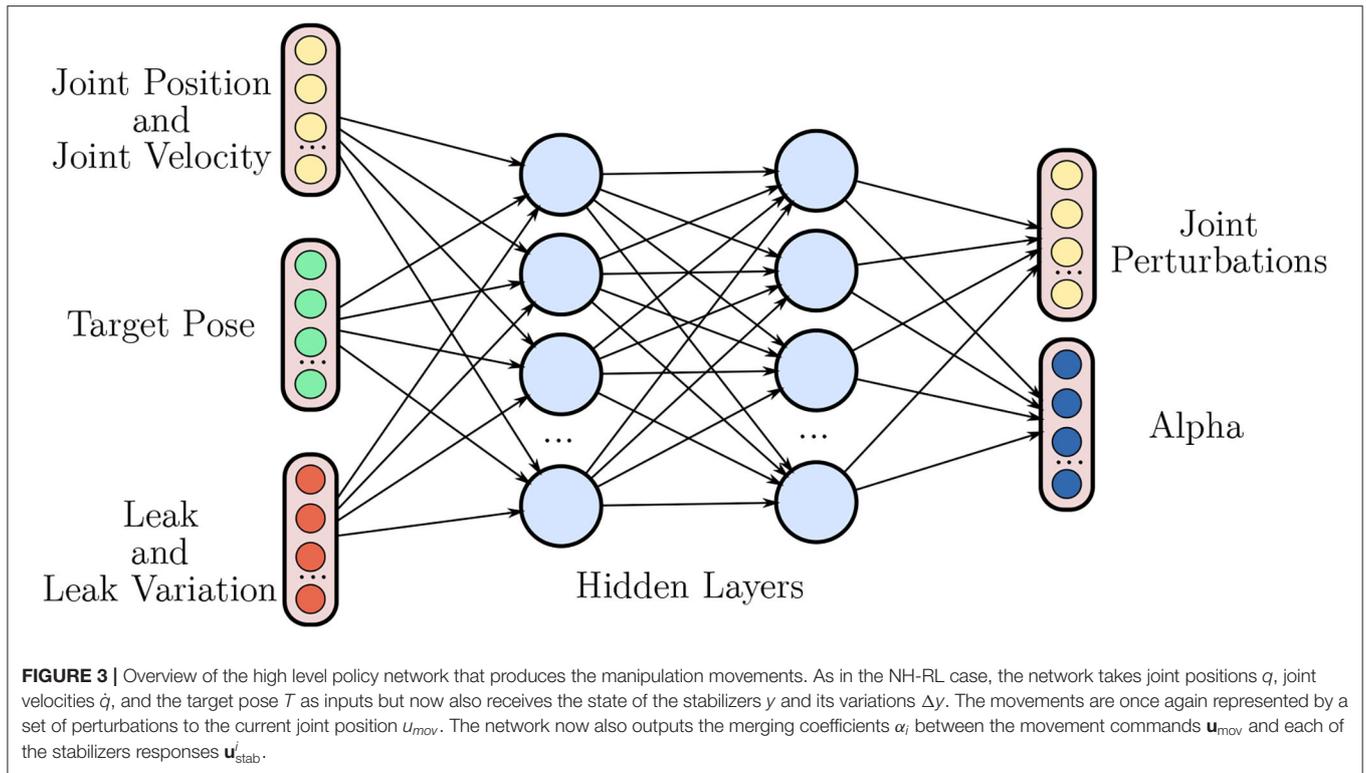
Figure 3 depicts the high-level movement policy of the H-RL setting while **Figure 4** provides an overview of the proposed hierarchy. The latter also re-emphasizes the fact that low-level is designed both in simulation and on the real robot, allowing the high-level policy to rely solely on joint space information.

An important set of hyper-parameters in our hierarchical decomposition is the initial distribution of each α_i , in order to obtain maximum variability in the trajectories of the initial policy and facilitate the RL process. Low values of α_i have the desired effect of stabilizing the grip but dampen the variability of the initial trajectories. Similarly, high values of α_i produce trajectories with low variability as the object falls almost immediately. To find an appropriate trade-off we manually tune the hyper-parameters governing the distribution of α by visual inspection of the resulting initial policy in simulation. The resulting distribution for each α_i is a Gaussian with mean 0.5 and a variance of 0.25. By centering the distribution at the transition point between the stabilization and the movement perturbations, we allow for exploration movements with stabilizer compensation. The variance being relatively low prevents sudden shifts from full movement to full stabilization perturbation and vice-versa.

Any RL algorithm can be applied to this hierarchical decomposition as the actions are not time-extended. Learning proceeds as follow: at the start of an episode a random target coordinate is sampled and the policy is executed until the object falls or 3,000 time steps (10 s) have elapsed. Upon collection of the trajectories we use TRPO (Schulman et al., 2015) to update the neural network policy depicted in **Figure 3**. In our experiments, the same implementation of TRPO (Dhariwal et al., 2017) is used to compare both NH-RL, and the proposed H-RL to in-hand manipulation.

3. EXPERIMENTAL EVALUATION

Using a simulated environment, we evaluate the efficiency of our proposed H-RL when compared to NH-RL and present preliminary results on transferring H-RL policies learned in simulation to a real robot platform.



3.1. Experimental Procedure, Testing Platform, and Tactile Sensors

All experiments are performed either on a simulated or real version of the Allegro Hand that is equipped with BioTac fingertip sensors (SynTouch Inc., www.syntouchinc.com). The Allegro Hand (Wonik Robotics GmbH, www.simlab.co.kr), is a four fingered hand with four joints per finger, for a total of 16 actuated degrees of freedom. With the exception of the thumb, all fingers have two metacarpal joints (rotation and flexing), a proximal joint and a distal joint. The thumb does not have a distal joint having an abduction joint instead. A PD controller was used to control the robot joint positions with a control loop that runs at 300 Hz.

BioTac tactile sensors (Wettels et al., 2014) were used as fingertip sensors. The sensors provide multi-modal responses composed of low and high frequency pressure (P_{dc} and P_{ac}) captured by a pressure transducer, local skin deformations (E) acquired through local impedance changes measured by 19 electrodes scattered across the sensors core surface, as well as temperature and thermal flow (T_{dc} and T_{ac}) measured by a thermometer. All data channels of the sensor are sampled at a rate of 100 Hz. The high frequency pressure is sampled in batches of 22 values at the same frequency. Considering all channels and the P_{ac} batch data, the sensors outputs a total of 44 values every 10 ms.

The PyBullet simulation environment (Coumans and Bai, 2018) is used to simulate the hand and the fingertip sensors. The PD control gains of the hand were tuned in simulation to emulate the behavior of the real hand. The BioTacs are not simulated.

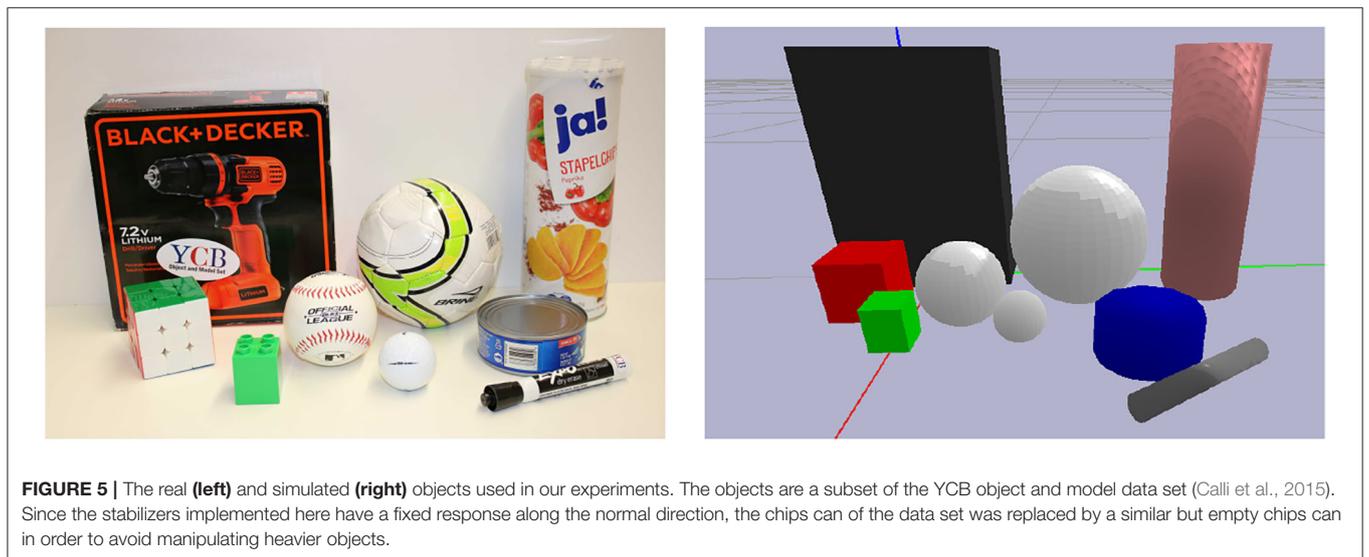
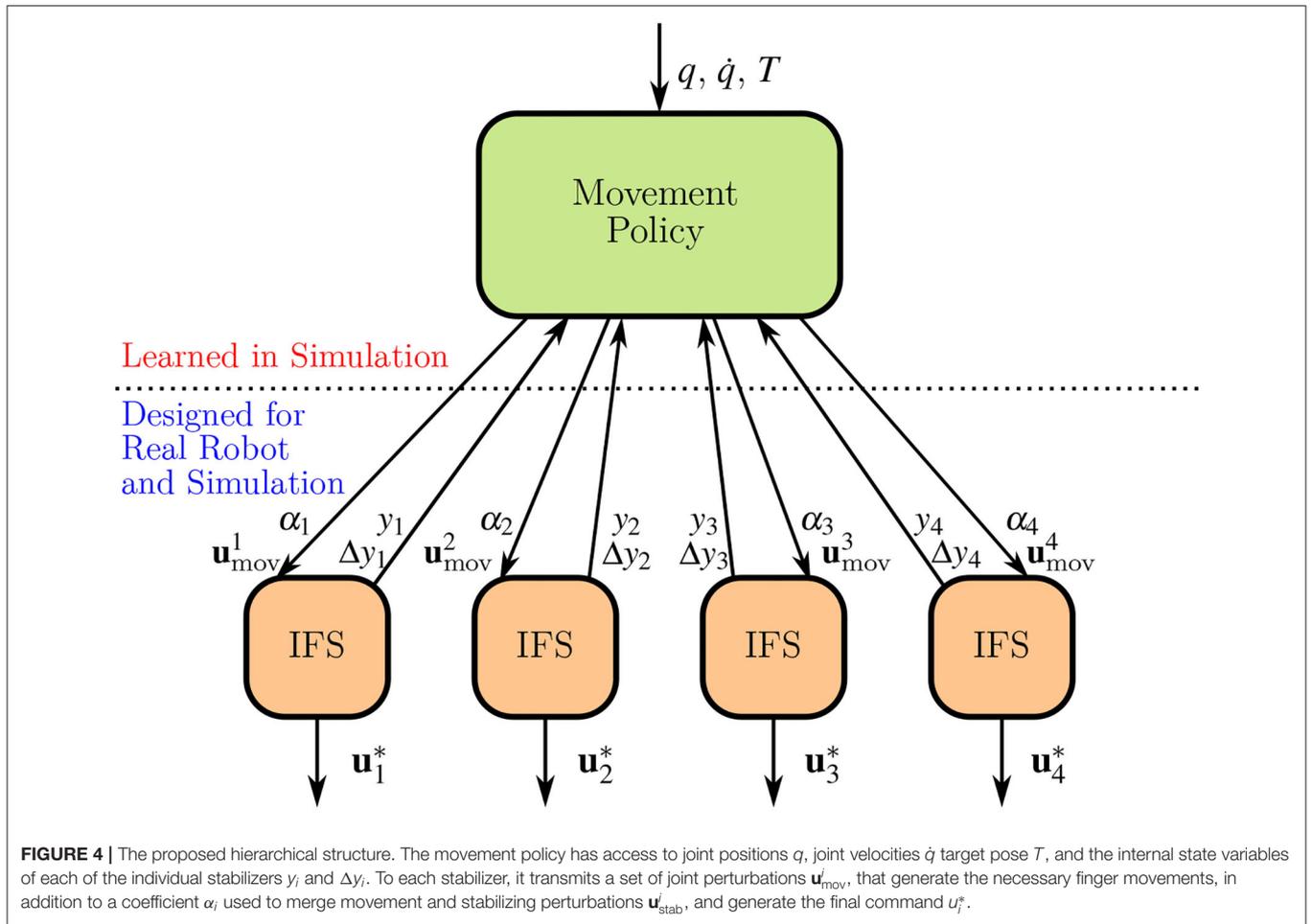
Instead information of contact force and normal direction is obtained directly from the collision engine. In addition to the simulated slip stabilizers, a simplified version of the stabilizers, that uses a constant desired velocity factor β , was implemented and compared with the full stabilizers

$$u_{stab}^i = \beta N_i. \tag{5}$$

After initial testing, it was found that the simple stabilizers would either apply forces that are not sufficiently strong to keep the object in hand or would apply excess force, hindering the manipulation movements. This is due to the simple stabilizers inability to regulate the applied velocity, and suggests that the lower level of the hierarchy requires feedback in order to be beneficial to the systems performance. These observations led us to present all results using the full stabilizers.

All experiments are performed on a subset of objects from the YCB object and model set (Calli et al., 2015), either simulated or on the real robot, as shown in Figure 5.

The simulation experiments considered three possible initial configurations: two fingered grasps for the green Lego brick, the golf ball and the marker, three fingered grasps for the Rubik's cube, the baseball and the tuna can and finally four fingered grasps for the screw-driver box, the small football, and the chips can. Each of these state configurations served as the initial pose for four different manipulation movements. These movements were sampled at the beginning of each trial by setting different target positions and target orientations, both with respect to the initial object position. The position targets are sampled from a



set of two positions, attempting to move the object by 2 cm to the edge of the work space with respect to the y axis. The hand is oriented such that x is the axis moving away from the

palm, y the axis pointing from the palms to the fingers when the fingers are in a stretched position, and z is the height. Rotation targets are either positive or negative $\pi/4$ rotations around the

initial position with the sign sampled uniformly at random. The coordinate frames, position targets, and rotation targets are depicted in **Figure 6**. Having four combinations at the edge of the work space allows all target poses to be consistently observed every episode, simplifying the learning process while potentially allowing the policy to generalize to other intermediate poses. Five learning trials were performed for each combination of manipulation/object configuration and target movements with 50 million samples per trial.

Regarding the learning process, the reward function R is given by

$$R(s_t, a_t) = F(P_{err} + O_{err}) - \mathbf{a}_{cost} - \dot{\mathbf{a}}_{cost} - d \quad (6)$$

where P_{err} and O_{err} respectively correspond to the position and orientation terms

$$P_{err} = e^{-(\mathbf{p}_{curr} - \mathbf{p}_{des})^2} \quad (7)$$

$$O_{err} = e^{-(\mathbf{o}_{curr} - \mathbf{o}_{des})^2}. \quad (8)$$

Using an exponential form for these terms guarantees that the maximum instantaneous reward for each term is one when the error is zero. In order to produce structured manipulation movements, we enforce that the number of fingers in the initial grasp is maintained throughout the manipulation action. This is achieved via F , a ratio between the number of fingers initially in contact with the object K and the current fingers in contact with the object

$$F = \frac{1}{K} \sum_{k=0}^K f_k \quad (9)$$

where f_k equals one if k -th finger is in contact and is zero otherwise. Since both P_{err} and O_{err} respectively increase as the

position and rotation errors decrease, multiplying the sum of these terms by the ratio F effectively limits the instantaneous reward, only providing a fraction of it if the number of fingers is not maintained throughout the trajectory. We also wish to enforce smooth movement during the manipulation action. We do so by applying costs \mathbf{a}_{cost} and $\dot{\mathbf{a}}_{cost}$ on the velocity and acceleration respectively

$$\mathbf{a}_{cost} = \sum_{i=0}^J (a_t^i)^2 \quad (10)$$

$$\dot{\mathbf{a}}_{cost} = \sum_{i=0}^J (a_t^i - a_{t-1}^i)^2 \quad (11)$$

where J is the number of joints. Finally, d is a negative penalty given when the object is dropped.

In addition to the previous reward terms, a specific term is added to the reward calculation in the H-RL setting. This term is an additional cost

$$\dot{\alpha}_{cost} = \sum_{i=0}^{N_{\text{fing}}} (\alpha_t^i - \alpha_{t-1}^i)^2 \quad (12)$$

applied on the variation of the α_i . It serves to penalize policies that shift very abruptly between stabilization and movement commands.

3.2. Hierarchical-RL vs. Non Hierarchical-RL

We compare the average accumulated reward (cumulative reward) achieved by NH-RL and by our proposed hierarchical decomposition H-RL, respectively represented by the blue and orange curves in **Figure 7**. Results show that H-RL performs

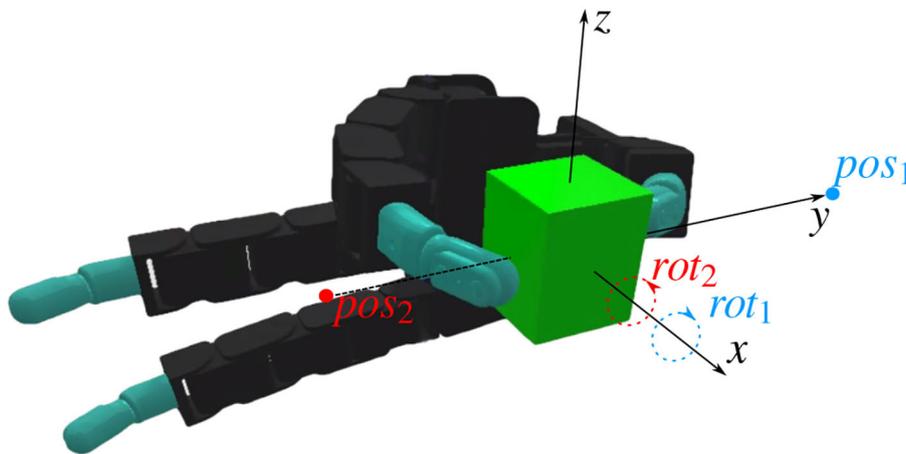
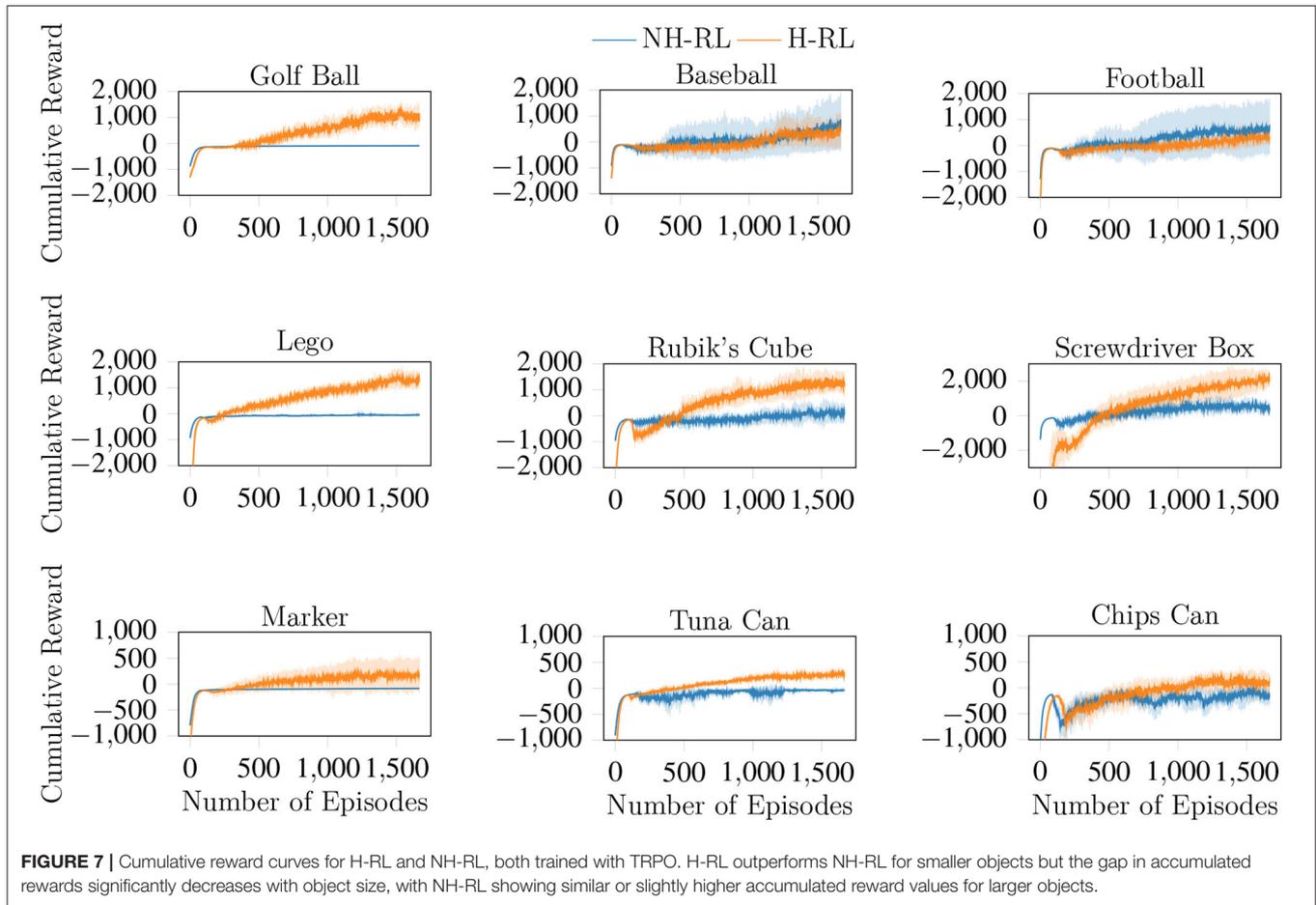


FIGURE 6 | A lego block in the initial grasp position. The coordinate frames defined for the manipulation are shown, as well as the two position targets pos_1 and pos_2 , and the two rotation targets rot_1 and rot_2 . The position targets represent object translations of 2 cm along the y axis. The rotation targets represent clockwise or counter-clockwise rotations of $\pi/4$ radians with respect to the x axis.

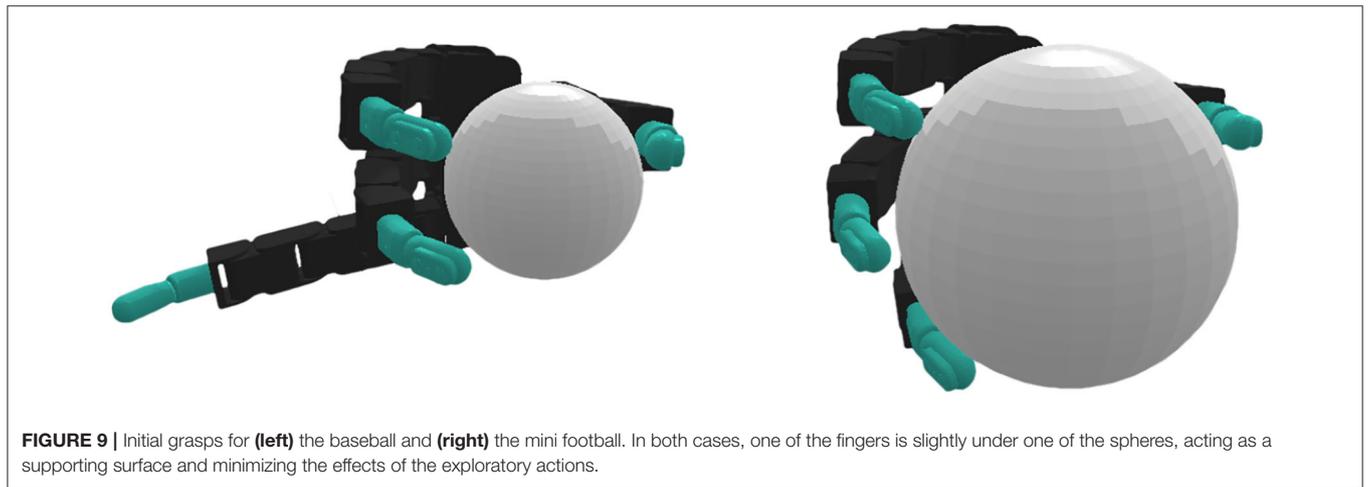
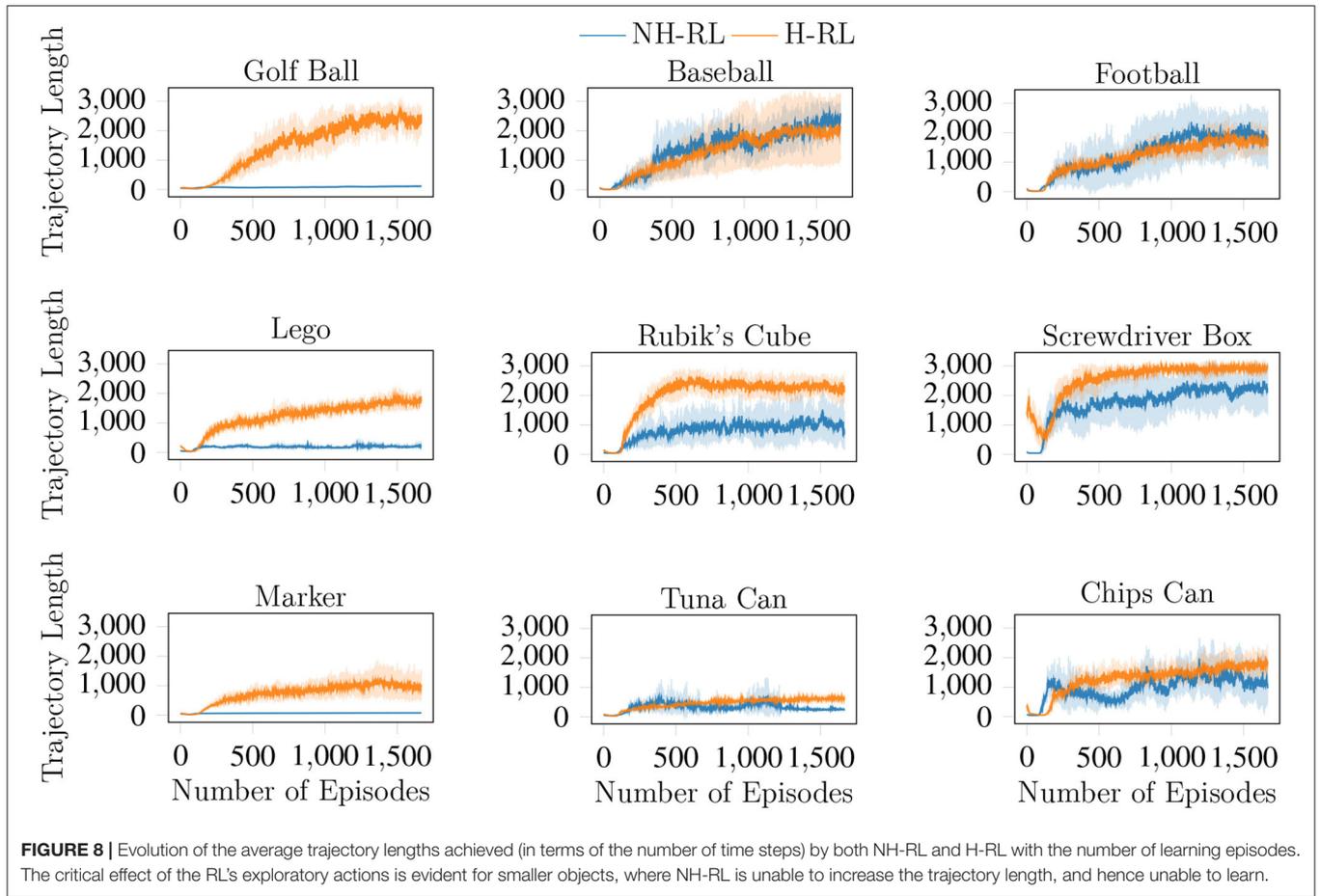


better or on par with NH-RL for all objects. For larger objects such as the football, the screwdriver box, and the chips can, exploratory actions that cause the object to shift in-hand are not as detrimental to the learning episode, as the size of the object allows it to be re-grasped before being dropped. This behavior is shown by a clear correlation between the difference in performance of the two approaches and the size of the object and/or the number of fingers involved in the manipulation action. The impact of bad exploratory actions on the learning process increases as the objects size decreases, rendering NH-RL unable to learn movement policies for smaller objects, while H-RL can learn movement policies for all objects. Moreover, these results are additionally emphasized by the evolution of the trajectory length during learning, shown in **Figure 8**. The average trajectory length for NH-RL policies remains very close to zero in all experiments with smaller objects, where exploratory actions have a critical effect on the movement. In addition to size, the initial grasp configurations can also greatly influence the outcome of the learning. This is the case for the football and the baseball, where one of the fingers is slightly underneath the object as depicted in **Figure 9**, serving as support for the exploration actions.

The effectiveness of the policies learned by both approaches also substantially differs. While NH-RL is capable of learning

policies for the partially supported and for the larger objects, the resulting policies are only capable of maintaining the objects in-hand without any consistent movement toward the target pose. In contrast, the H-RL policies are capable of consistently reorienting the objects to the correct orientations, despite maintaining the initial position error. This behavior, shown in **Figure 10** for the lego block, is observed for both the cubic objects and the cylindrical objects. The spherical objects are kept stable in-hand, with no consistent reduction of position or orientation errors. This behavior results from all the contacts being simulated as contacts between fully rigid objects. This form of contact simulation is particularly relevant for spherical objects where very fine contact management is necessary for repositioning the object.

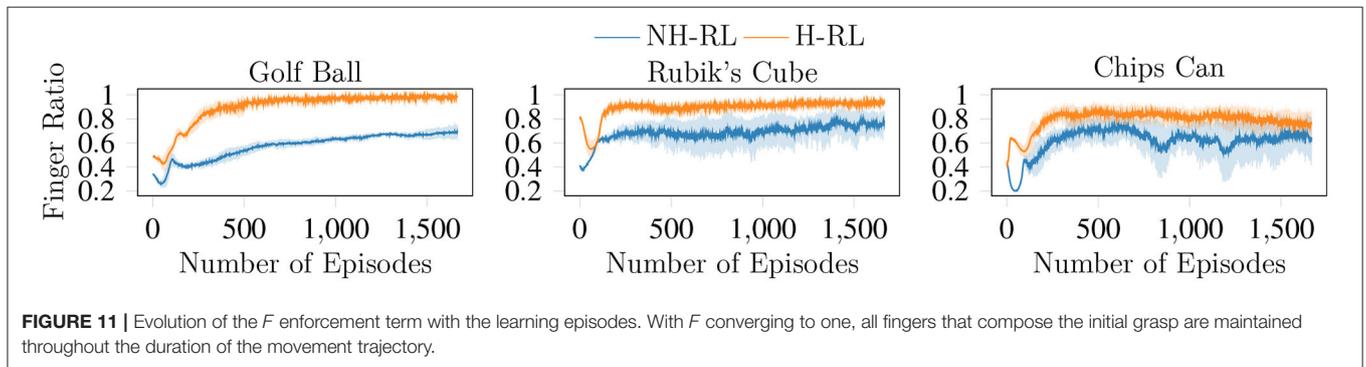
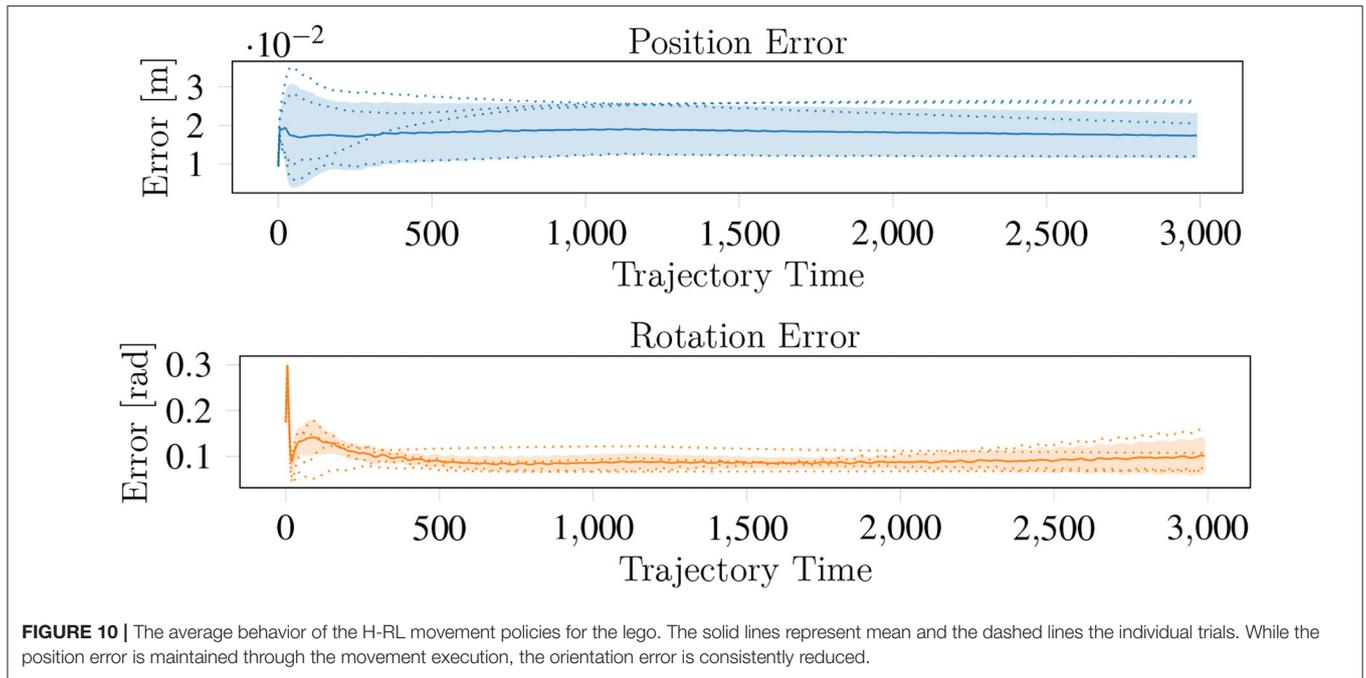
We attribute the inability of the H-RL policies to minimize the position error to (i) these movements requiring a very explicit reduction of the α_i for the fingers for which the movement is in the opposite direction to the contact normal and (ii) this explicit coordination being harder to learn when using a model free approach with opposite movements of the object (movements along either the positive or negative direction along the y axis) in the same batch. In addition, exploration where α_i is substantially skewed toward the upper level policy commands is heavily penalized if the object falls. Despite this, we were able to learn



policies where the position errors were minimized by training only with a single position target in the target set. This suggests that a more complex upper level policy is required to learn how to move the object to arbitrary targets.

Finally, we show the effects of the F term enforcement, where we wish to keep all fingers of the initial grasps involved in the manipulation action. The evolution of the F term with the

learning process shown in **Figure 11** indicates that the ratio between initial and used fingers increases with the number of episodes, converging to values near the maximum value of one, where all fingers in the initial grasp take part in the manipulation action. Another interesting aspect of the results shown in **Figure 11** are the large fluctuations in the F ratio in the initial stages of the learning process. While exploring the state



action space, the initial policies constantly remove fingers from the object, until a balance is reached between improving the error terms and the F ratio simultaneously. This effect is also visible in the cumulative reward curves shown in **Figure 7**.

3.3. Transfer to the Real Robot

Several policies were tested on the real robot in order to assess their transfer capabilities. In **Figure 12**, the movements produced by two policies for the lego block are depicted. These policies were transferred with no further learning on the real robot, displaying similar movements to the ones observed in the simulation environment. While policies correctly transfer to the real robot, the movements are hindered by inaccuracies in the estimated contact normal and by noise on the slip predictors. These estimations are fairly robust for small movements but quickly diverge once contact positions considerably shift. These errors in the contact normal are reduced during the execution of the movement since the α_i values are providing more control to the upper level policy. Once the upper level policy finishes the

desired manipulation movement, and the values of α_i begging to be shift the control to the low level stabilizers, the contact normal errors become more relevant, often resulting in the object being dropped from the grasp. From the policies tested, the ones that better transferred were the ones for the lego block and the Rubik's cube, which is consistent with the manipulation performances observed in simulation.

4. CONCLUSION

We have proposed a hierarchical decomposition for the in-hand manipulation problem in order to enable learning policies for manipulating unsupported objects. The policies learned in simulation are transferred to a real robot platform where similar manipulation movements are observed. Our decomposition is based on low-level per-digit stabilizing controllers that effectively incorporate tactile feedback to ensure a stable grip during object manipulation and a high-level policy that coordinates digit movement and modulates the influence of the individual

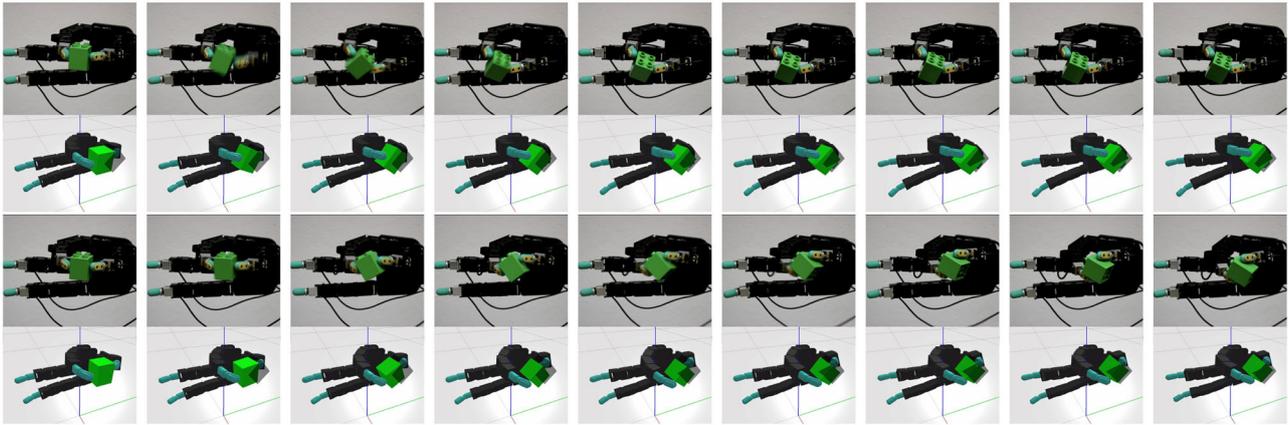


FIGURE 12 | The behavior of two policies trained on the simulator and transferred to the real robot with no additional training. The movements consist of a clockwise (top two rows) or counter clockwise rotation of the object along the axis perpendicular to the palm (bottom two rows).

low-level controllers. Our decomposition allows for efficient training of high-level policies for dexterous manipulation in simulation on a range of different objects achieving faster learning and higher rewards than its non-hierarchical counterpart. By abstracting and encapsulating tactile feedback in the lower-level controllers, the hierarchical decomposition enables direct transfer of policies that were trained in simulation to a physical system.

An interesting direction for future work is to explore the possibility of learning a single policy that is able to perform all achievable translations and rotations of the grasped object by taking inspiration from recent developments in multi-task reinforcement learning.

REFERENCES

- Akkaya, I., Andrychowicz, M., Chociej, M., Litwin, M., McGrew, B., Petron, A., et al. (2019). Solving rubik's cube with a robot hand. *arXiv [Preprint] arXiv:1910.07113*.
- Andrychowicz, M., Baker, B., Chociej, M., Jozefowicz, R., McGrew, B., Pachocki, J., et al. (2020). Learning dexterous in-hand manipulation. *Int. J. Robot. Res.* 39, 3–20. doi: 10.1177/0278364919887447
- Bai, Y., and Liu, C. K. (2014). “Dexterous manipulation using both palm and fingers,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (Hong Kong), 1560–1565. doi: 10.1109/ICRA.2014.6907059
- Calli, B., Singh, A., Walsman, A., Srinivasa, S., Abbeel, P., and Dollar, A. M. (2015). “The YCB object and model set: Towards common benchmarks for manipulation research,” in *2015 International Conference on Advanced Robotics (ICAR)* (Istanbul), 510–517. doi: 10.1109/ICAR.2015.7251504
- Chavan-Dafle, N., and Rodriguez, A. (2015). “Prehensile pushing: in-hand manipulation with push-primitives,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Hamburg), 6215–6222. doi: 10.1109/IROS.2015.7354264
- Cherif, M., and Gupta, K. K. (1999). Planning quasi-static fingertip manipulations for reconfiguring objects. *IEEE Trans. Robot. Autom.* 15, 837–848. doi: 10.1109/70.795789
- Coumans, E., and Bai, Y. (2016–2018). *Pybullet, a Python Module for Physics Simulation for Games, Robotics and Machine Learning*. Available online at: <http://pybullet.org>
- Dafle, N. C., Rodriguez, A., Paolini, R., Tang, B., Srinivasa, S. S., Erdmann, M., et al. (2014). “Extrinsic dexterity: in-hand manipulation with external forces,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (Hong Kong), 1578–1585. doi: 10.1109/ICRA.2014.6907062
- Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., et al. (2017). *Openai Baselines*. Available online at: <https://github.com/openai/baselines>
- Flanagan, J. R., Bowman, M. C., and Johansson, R. S. (2006). Control strategies in object manipulation tasks. *Curr. Opin. Neurobiol.* 16, 650–659. doi: 10.1016/j.conb.2006.10.005
- Gupta, A., Eppner, C., Levine, S., and Abbeel, P. (2016). “Learning dexterous manipulation for a soft robotic hand from human demonstrations,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Daejeon), 3786–3793. doi: 10.1109/IROS.2016.7759557
- Johansson, R. S. (1996). “Sensory control of dexterous manipulation in humans,” in *Hand and Brain*, eds A. M. Wing, P. Haggard, and J. Randall Flanagan (Academic Press), 381–414. doi: 10.1016/B978-012759440-8/50025-6
- Kumar, V., Todorov, E., and Levine, S. (2016). “Optimal control with learned local models: application to dexterous manipulation,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)* (Stockholm), 378–383. doi: 10.1109/ICRA.2016.7487156
- Levine, S., Finn, C., Darrell, T., and Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.* 17, 1334–1373. doi: 10.5555/2946645.2946684

DATA AVAILABILITY STATEMENT

The datasets generated for this study are available on request to the corresponding author.

AUTHOR CONTRIBUTIONS

FV executed the work with the assistance of RA with respect to the reinforcement learning approach. JP with respect to the overall architecture of the control approach. All authors contributed to the article and approved the submitted version.

- Li, M., Yin, H., Tahara, K., and Billard, A. (2014). "Learning object-level impedance control for robust grasping and dexterous manipulation," in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (Hong Kong), 6784–6791. doi: 10.1109/ICRA.2014.6907861
- Ma, R. R., and Dollar, A. M. (2011). "On dexterity and dexterous manipulation," in *2011 15th International Conference on Advanced Robotics (ICAR)* (Tallinn), 1–7. doi: 10.1109/ICAR.2011.6088576
- Maekawa, H., Tanie, K., and Komoriya, K. (1995). "Tactile sensor based manipulation of an unknown object by a multifingered hand with rolling contact," in *Proceedings 1995 IEEE International Conference on Robotics and Automation, 1995, Vol. 1* (Nagoya), 743–750. doi: 10.1109/ROBOT.1995.525372
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level control through deep reinforcement learning. *Nature* 518, 529–533. doi: 10.1038/nature14236
- Mordatch, I., Popović, Z., and Todorov, E. (2012). "Contact-invariant optimization for hand manipulation," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Lausanne: Eurographics Association), 137–144.
- Okamura, A. M., Smaby, N., and Cutkosky, M. R. (2000). "An overview of dexterous manipulation," in *Proceedings of ICRA'00. IEEE International Conference on Robotics and Automation, 2000, Vol. 1* (San Francisco, CA), 255–262. doi: 10.1109/ROBOT.2000.844067
- Popov, I., Heess, N., Lillicrap, T., Hafner, R., Barth-Maron, G., Vecerik, M., et al. (2017). Data-efficient deep reinforcement learning for dexterous manipulation. *arXiv [Preprint] arXiv:1704.03073*.
- Rajeswaran, A., Kumar, V., Gupta, A., Schulman, J., Todorov, E., and Levine, S. (2017). Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv [Preprint] arXiv:1709.10087*. doi: 10.15607/RSS.2018.XIV.049
- Saut, J.-P., Sahbani, A., El-Khoury, S., and Perdereau, V. (2007). "Dexterous manipulation planning using probabilistic roadmaps in continuous grasp subspaces," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007* (Macau), 2907–2912. doi: 10.1109/IROS.2007.4399090
- Schulman, J., Levine, S., Jordan, M., and Abbeel, P. (2015). "Trust region policy optimization," in *International Conference on Machine Learning (ICML)* (Lille), 16.
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., et al. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature* 529, 484–489. doi: 10.1038/nature16961
- Stork, J. A., Ek, C. H., Bekiroglu, Y., and Kragic, D. (2015). "Learning predictive state representation for in-hand manipulation," in *2015 IEEE International Conference on Robotics and Automation (ICRA)* (Seattle, WA), 3207–3214. doi: 10.1109/ICRA.2015.7139641
- Sundaralingam, B., and Hermans, T. (2018). Relaxed-rigidity constraints: kinematic trajectory optimization and collision avoidance for in-grasp manipulation. *Auton. Robots* 43, 469–483. doi: 10.1007/s10514-018-9772-z
- Van Hoof, H., Hermans, T., Neumann, G., and Peters, J. (2015). "Learning robot in-hand manipulation with tactile features," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)* (Seoul), 121–127. doi: 10.1109/HUMANOIDS.2015.7363524
- Veiga, F., Edin, B., and Peters, J. (2020). Grip stabilization through independent finger tactile feedback control. *Sensors* 20:1748. doi: 10.3390/s20061748
- Wettels, N., Fishel, J. A., and Loeb, G. E. (2014). "Multimodal tactile sensor," in *The Human Hand as an Inspiration for Robot Hand Development*, eds R. Balasubramanian and V. J. Santos (Cham: Springer International Publishing), 405–429. doi: 10.1007/978-3-319-03017-3_19
- Wettels, N., Parnandi, A. R., Moon, J.-H., Loeb, G. E., and Sukhatme, G. S. (2009). Grip control using biomimetic tactile sensing systems. *IEEE/ASME Trans. Mechatron.* 14, 718–723. doi: 10.1109/TMECH.2009.2032686
- Yousef, H., Boukallel, M., and Althoefer, K. (2011). Tactile sensing for dexterous in-hand manipulation in robotics—a review. *Sensors Actuat. A* 167, 171–187. doi: 10.1016/j.sna.2011.02.038
- Zheng, X.-Z., Nakashima, R., and Yoshikawa, T. (2000). On dynamic control of finger sliding and object motion in manipulation with multifingered hands. *IEEE Trans. Robot. Autom.* 16, 469–481. doi: 10.1109/70.880798
- Zhu, H., Gupta, A., Rajeswaran, A., Levine, S., and Kumar, V. (2019). "Dexterous manipulation with deep reinforcement learning: efficient, general, and low-cost," in *2019 International Conference on Robotics and Automation (ICRA)* (Montreal, QC: IEEE), 3651–3657. doi: 10.1109/ICRA.2019.8794102

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Veiga, Akrou and Peters. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.