Tactile Grasp Refinement using Deep Reinforcement Learning and Analytic Grasp Stability Metrics

Alexander Koenig[1,2], Zixi Liu[2], Lucas Janson[3] and Robert Howe[2,4]

***Abstract*— Reward functions are at the heart of every reinforcement learning (RL) algorithm. In robotic grasping, rewards are often complex and manually engineered functions that do not rely on well-justified physical models from grasp analysis. This work demonstrates that analytic grasp stability metrics constitute powerful optimization objectives for RL algorithms that refine grasps on a three-fingered hand using only tactile and joint position information. We outperform a binary-reward baseline by 42.9% and find that a combination of geometric and force-agnostic grasp stability metrics yields the highest average success rates of 95.4% for cuboids, 93.1% for cylinders, and 62.3% for spheres across wrist position errors between 0 and 7 centimeters and rotational errors between 0 and 14 degrees. In a second experiment, we show that grasp refinement algorithms trained with contact feedback (contact positions, normals, and forces) perform up to 6.6% better than a baseline that receives no tactile information.**

I. Introduction

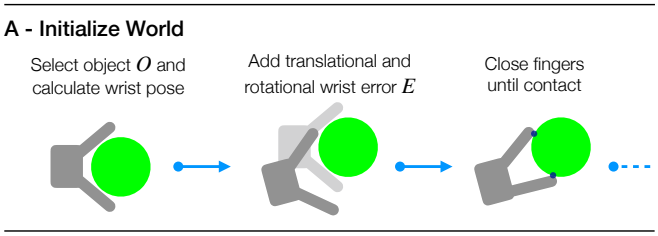
Most modern grasping systems rely on computer vision to plan a grasp and an open-loop controller to execute the generated trajectory. However, open-loop controllers often fail when a grasp is subject to calibration errors. Such errors typically arise due to misaligned coordinate frames or inaccuracies in the object pose and geometry estimation. Computer vision is often not suitable to recover from calibration errors due to occlusion. Hence, there is excellent potential for tactile sensing in closed-loop robotic grasp refinement.

Recent advances in reinforcement learning make the technique increasingly attractive for robotic grasping. Several recent works process tactile information from multi-fingered hands with RL algorithms [1], [2], [3], [4] for robotic grasping. A critical part of every RL algorithm is the reward function [5]. Table I shows an overview of the reward functions used in these related works.

While some reward functions in Table I encode the experiment outcome [2], [3] others consist of manually engineered cues (e.g., number of contacts [2], [4]). However, such cues often have no well-justified relation with grasp stability: a grasp with many contact points can easily fail if the contact forces are insufficient to perform the task. Moreover, the rich body of research on grasp analysis, contact modeling, and grasp quality metrics [6] is not leveraged by current reward functions. Hence, in our first contribution, we demonstrate

This material is based upon work supported by the US National Science Foundation under Grant No. IIS-1924984 and by the German Academic Exchange Service.

[1] Department of Informatics, Technical University of Munich
[2] School of Engineering and Applied Sciences, Harvard University
[3] Departments of Statistics, Harvard University
[4] RightHand Robotics, Inc., 237 Washington St, Somerville, MA 02143 USA. Robert Howe is corresponding author howe@seas.harvard.edu.



Stage	Refine	Lift	Hold	End
Steps	15	6	6	-
Duration	5 s	2 s	2 s	-
ϵ *and* δ	$\epsilon_f + \alpha_1\epsilon_\tau + \alpha_2\delta_{task}$	$\epsilon_f + \alpha_1\epsilon_\tau + \alpha_2\delta_{cur}$	$\epsilon_f + \alpha_1\epsilon_\tau + \alpha_2\delta_{cur}$	0
δ	δ_{task}	δ_{cur}	δ_{cur}	0
ϵ	$\epsilon_f + \alpha_1\epsilon_\tau$	$\epsilon_f + \alpha_1\epsilon_\tau$	$\epsilon_f + \alpha_1\epsilon_\tau$	0
β	0	0	0	{0,1}

Fig. 1: Overview of one algorithm episode. (A) Initialization of hand and object. (B) We split the grasp refinement algorithm into four stages and compare four reward frameworks: (1) ϵ *and* δ, (2) only δ, (3) only ϵ and (4) the binary reward framework β. The weighting factors of $\alpha_1 = 5$ and $\alpha_2 = 0.5$ were empirically determined.

the potential of analytic grasp stability metrics in the task of tactile grasp refinement. In these experiments, as shown in Fig. 1, we first randomly select an object O and a wrist error E to simulate calibration errors. The hand consequently closes its fingers in this initial grasp configuration. In the grasp refinement episode, the algorithm uses only contact and finger joint position data to refine the grasp by iteratively updating the wrist and finger positions. The algorithm lifts and holds the object to evaluate the grasp's stability. We compare three types of rewards: a quality metric ϵ based on the largest-minimum resisted wrench [7], a force-based metric δ that evaluates the distance of the contact forces to the friction cone, and a binary task execution metric β.

Several recent works demonstrated that RL algorithms benefit from contact feedback when grasping [2] and when performing in-hand manipulation tasks [8], [9]. However, the same studies [8], [9] also revealed that models trained with binary contact signals perform equally well as models that receive accurate normal force information. This result is coun-

TABLE I: Reward functions of related works.

Paper	Reward
Chebotar 2016 [1]	Maximize predicted grasp success from learned stability predictor
Merzić 2019 [2]	Maximize (1) number of links in contact and (2) binary drop test reward Minimize (1) distance object to gripper, (2) distance fingertips to object, (3) joint torques and (4) object velocity
Wu 2019 [3]	Maximize binary pick-up reward at episode end Minimize finger reopening
Hu 2020 [4]	Maximize (1) number of contact points and (2) number of object key-points contained in convex hull of hand and finger key-points Minimize (1) distance from hand key-points to object key-points, (2) angle between hand key-point normals and vectors pointing from hand key-points to object center, (3) number of contacts on outside of fingers and (4) object linear velocity

terintuitive since, for the studied in-hand manipulation tasks (e.g., pen or block rotation), the magnitudes of the contact forces are undoubtedly relevant. In our second contribution, we quantify the benefit of contact sensing in tactile grasp refinement and analyze whether we reach similar results as in [8], [9] using physically inspired reward functions in the grasp refinement and lifting experiments from Fig. 1. In addition to the binary and normal force frameworks, we also include a framework in our comparison that receives the full contact force vector, which we hypothesize to be especially helpful while lifting the object where tangential forces are more prominent. These results can be a valuable guide towards robotic hand design and RL research in general.

II. Grasp Stability Metrics

A. Largest-minimum resisted forces and torques

Ferrari and Canny [7] define grasp quality as the largest-minimum perturbing wrench that the grasp can resist given the grasp's force constraints. Ferrari's metric [7] suffers from the non-comparability of forces (in N) and torques (in Nm). Hence, Mirtich and Canny [10] refine this popular metric by decoupling the wrench space into a force and torque component, and thereby evaluate how well a grasp resists pure forces and torques.

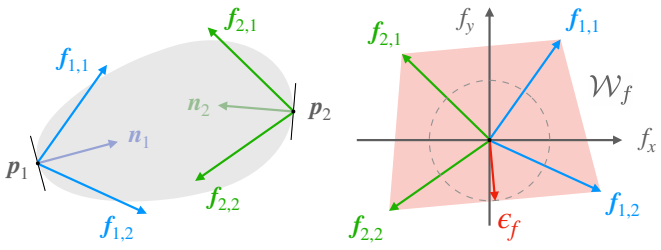


Fig. 2: Left: a grasp with two contact points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, contact normals $\boldsymbol{n}_i$ and friction cones. Right: the quality metric ϵ_f is the radius of the largest ball contained in the convex hull $\mathcal{W}_f$ over the set of resisted forces.

Let us examine how to measure resistance to disturbing forces. The contact force $\boldsymbol{f}_i$ at each contact i is constrained via the friction cone $f_{i,t} \leq \mu f_{i,n}$, where μ is the coefficient of friction and $f_{i,t}$ and $f_{i,n}$ are the tangential and normal components of $\boldsymbol{f}_i$, respectively. The friction cone is discretized using m edges $\boldsymbol{f}_{i,j}$. The set of forces $\mathcal{W}_f$ that the contacts can apply to the object is $\mathcal{W}_f = \text{ConvexHull}\left(\bigcup_{i=1}^{n}\{\boldsymbol{f}_{i,1},\ldots,\boldsymbol{f}_{i,m}\}\right)$, where n is the number of contacts. Finally, the quality metric ϵ_f in equation (1) is the shortest distance from the origin to the nearest hyperplane of $\mathcal{W}_f$. Hence, the metric defines a lower bound on the resisted force in all directions. As shown in Fig. 2, ϵ_f can be geometrically interpreted as the radius of the largest ball centered at the origin and contained inside $\mathcal{W}_f$.

$$\epsilon_f = \min_{\boldsymbol{f}\in\partial\mathcal{W}_f} \|\boldsymbol{f}\| \tag{1}$$

This concept is easily extended to the torque domain. The reaction torque $\boldsymbol{\tau}_{i,j}$ resulting from a friction cone edge $\boldsymbol{f}_{i,j}$ is calculated by $\boldsymbol{f}_{i,j} = \boldsymbol{r_i} \times \boldsymbol{f_{i,j}}$, where $\boldsymbol{r_i}$ is a vector pointing from the object's center of mass to the contact point $\boldsymbol{p_i}$. The set of torques $\mathcal{W}_\tau$ that the grasp can resist is defined by $\mathcal{W}_\tau = \text{ConvexHull}\left(\bigcup_{i=1}^{n}\{\boldsymbol{\tau}_{i,1},\ldots,\boldsymbol{\tau}_{i,m}\}\right)$. The metric ϵ_τ in equation (2) evaluates the grasp's quality by identifying the magnitude of the largest-minimum resisted torque.

$$\epsilon_\tau = \min_{\boldsymbol{\tau}\in\partial\mathcal{W}_\tau} \|\boldsymbol{\tau}\| \tag{2}$$

B. Minimum distance to the friction cone

The quality metrics ϵ_f and ϵ_τ analyze the forces that each contact can theoretically exert on the object. However, these metrics do not consider the actual contact forces that the contacts apply to the object. To this end, we define two force-based quality metrics δ_{cur} and δ_{task}. While δ_{cur} is a general-purpose grasp quality metric, δ_{task} is applicable when a task definition exists.

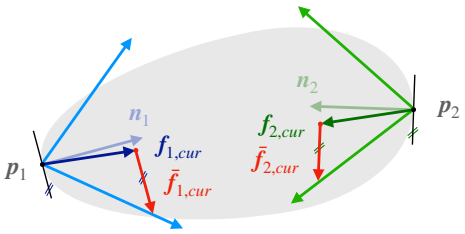


Fig. 3: Grasp with current contact forces $\boldsymbol{f}_{i,cur}$ and tangential force margins $\bar{\boldsymbol{f}}_{i,cur}$ to the friction cones.

Similar to Buss et al. [11], we measure grasp stability in terms of how far the contact forces are from the friction limits. Fig. 3 shows a grasp with the current contact forces $\boldsymbol{f}_{i,cur}$ and the tangential force margins $\bar{\boldsymbol{f}}_{i,cur}$. The vectors $\bar{\boldsymbol{f}}_{i,cur}$ are forces in the tangential direction that point from $\boldsymbol{f}_{i,cur}$ to the closest point on the friction cone, thereby identifying the direction in which the contact can take the least tangential force before slipping. A grasp with large tangential force margins $\bar{\boldsymbol{f}}_{i,cur}$ is desirable since the contacts are less prone to sliding when an object wrench is applied. Hence, the metric δ_{cur} in equation (3) measures the average magnitude of the safety margins $\|\bar{\boldsymbol{f}}_{i,cur}\|$. Contacts with larger forces contribute more to grasp stability because they can have larger tangential force margins $\bar{\boldsymbol{f}}_{i,cur}$ and

can thereby compensate more disturbing object wrenches. Therefore, we weigh the safety margins $\|\bar{\boldsymbol{f}}_{i,cur}\|$ by their respective contact force magnitudes $\|\boldsymbol{f}_{i,cur}\|$.

$$\delta_{cur} = \frac{\sum_{i=1}^{n_c} \|\boldsymbol{f}_{i,cur}\| \|\bar{\boldsymbol{f}}_{i,cur}\|}{\sum_{i=1}^{n_c} \|\boldsymbol{f}_{i,cur}\|} \tag{3}$$

In many grasping tasks, a clear task definition exists. Let $T = \{\boldsymbol{w_1}, \boldsymbol{w_2}, \ldots, \boldsymbol{w_m}\}$ be the set of task wrenches that the grasp must resist during task execution (e.g., object weight or wrenches from expected collisions). Several task-oriented quality metrics measure how well a convex set of T is contained within the convex set of all wrenches that the grasp can resist [12], [13], [14]. However, since these approaches reason about the theoretically applicable contact forces, which are commonly bounded to unity [6], [7], it is not possible to evaluate whether the *current* contact forces of a grasp are suitable to balance the anticipated task wrenches.

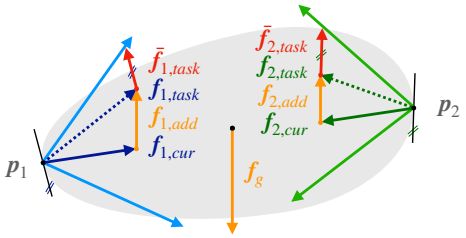


Fig. 4: Grasp with predicted task contact forces $\boldsymbol{f}_{i,task}$ after mapping the task force $-\boldsymbol{f}_g$ onto the contacts.

We define an alternative task-oriented metric δ_{task}. With $\mathbf{G}^{+}\boldsymbol{w} = (\ \boldsymbol{f}_{1,add}^T \quad \boldsymbol{f}_{2,add}^T \quad \cdots \quad \boldsymbol{f}_{n_c,add}^T\ )^T$ where $\mathbf{G}^{+}$ is the pseudoinverse of the grasp matrix we calculate the additional contact force $\boldsymbol{f}_{i,add}$ that each contact i must react with to compensate the task wrench $\boldsymbol{w} \in T$. Fig. 4 shows that the task contact force $\boldsymbol{f}_{i,task} = \boldsymbol{f}_{i,cur} + \boldsymbol{f}_{i,add}$ is the sum of the current contact force $\boldsymbol{f}_{i,cur}$ and $\boldsymbol{f}_{i,add}$ which results from a task wrench (here the object weight $-\boldsymbol{f}_g$). We use a hard contact model and assume that the internal grasp forces stay the same after applying $\boldsymbol{f}_{i,add}$. The metric δ_{task} in equation (4) measures the expected grasp stability during task execution by computing the average magnitude of the tangential force margins $\|\bar{\boldsymbol{f}}_{i,task}\|$ of the task contact forces $\boldsymbol{f}_{i,task}$. The metric δ_{task} is a lower bound over all task wrenches $\boldsymbol{w} \in T$ and we thereby identify the worst-case task wrench.

$$\delta_{task} = \min_{\boldsymbol{w} \in T} \frac{\sum_{i=1}^{n_c} \|\boldsymbol{f}_{i,task}\| \|\bar{\boldsymbol{f}}_{i,task}\|}{\sum_{i=1}^{n_c} \|\boldsymbol{f}_{i,task}\|} \tag{4}$$

III. Reward Design and Grasp Refinement

A. Simulation Environment

We simulate the grasp refinement episodes of the three-fingered ReFlex TakkTile hand using a custom robotics simulator based on the Gazebo [15] simulation environment, the DART [16] physics engine, and the ROS [17] communication framework. We model the under-actuated distal flexure as a rigid link with two revolute joints (one between the proximal and one between the distal finger link). Further, we approximate the finger geometries as cuboids to reduce computational load. We activate simulated gravity in our experiments (unlike in [2]), and the object can freely interact with the hand and the world. Our source code is publicly available under github.com/axkoenig/grasp_refinement.

B. Training Dataset

Each training sample consists of a tuple (O, E), where O is the object, and E is the wrist pose error sampled uniformly before every episode. There are three object types (cuboid, cylinder, and sphere) with a mass $\in [0.1, 0.4]$ kg and randomly sampled sizes. Fig. 5 visualizes the minimum and maximum object dimensions. The wrist pose error E consists of a translational and a rotational error. We uniformly sample the translational error (e_x, e_y, e_z) from $[-5, 5]$ cm and the rotational error (e_ξ, e_η, e_ζ) from $[-10, 10]$ deg for each variable, respectively.

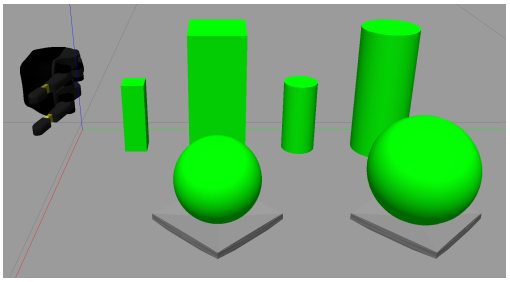


Fig. 5: Minimum and maximum object sizes. We place the spheres on a concave mount to prevent rolling.

C. Test Dataset

We define 8 different wrist error cases for the test dataset. Let $d(a, b, c) = \sqrt{a^2 + b^2 + c^2}$ be the L2 norm of the variables (a, b, c). Table II shows the wrist error cases, where case A corresponds to no error and case H means maximum wrist error. Fig. 6 visualizes two wrist error cases. The test dataset consists of 30 random objects O (10 cuboids, 10 cylinders, and 10 spheres). Per object O, we randomly generate the eight wrist error cases $\{A, B, \ldots, H\}$ from Table II. Hence, we run $30 \times 8 = 240$ experiments to test one model.

TABLE II: Wrist error cases

Wrist Error Case	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
$d(e_x, e_y, e_z)$ in cm	0	1	2	3	4	5	6	7
$d(e_\xi, e_\eta, e_\zeta)$ in deg	0	2	4	6	8	10	12	14

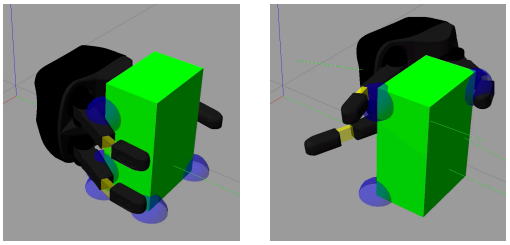


Fig. 6: Left: wrist error case **A** (no wrist error), Right: wrist error case **H** (maximum wrist error). Contact points in blue.

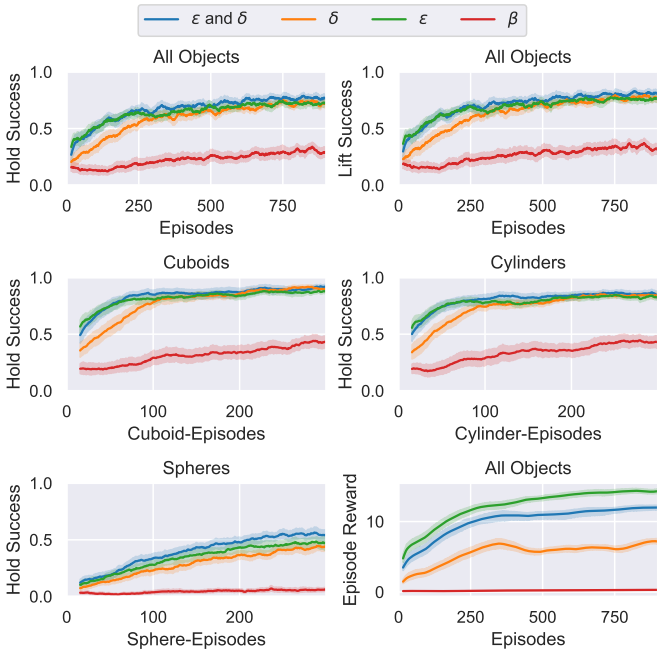


Fig. 7: Training results for reward frameworks.

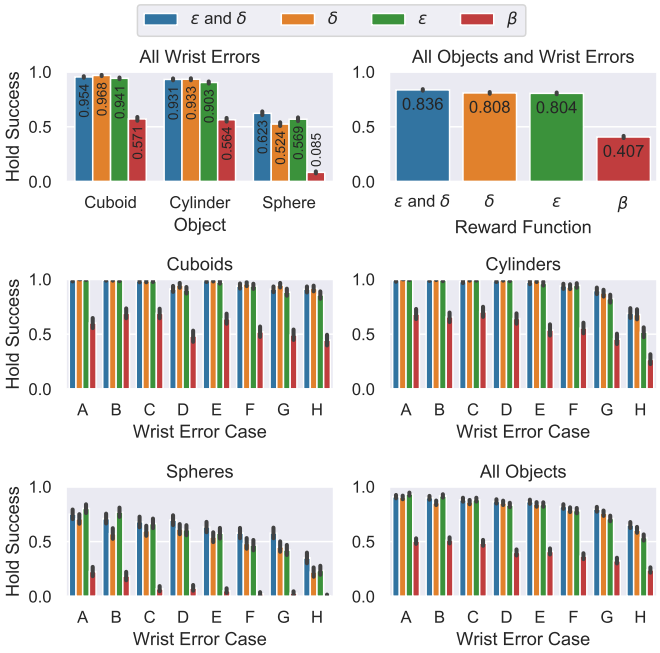


Fig. 8: Test results for reward frameworks.

D. State and Action Space

The state vector $\boldsymbol{s}$ consists of 7 joint positions (1 finger separation, 3 proximal bending, 3 distal bending degrees of freedom), and 7 contact cues (3 on proximal links, 3 on distal links, and 1 on palm) that include contact position, contact normal and contact force, which have 3 (x, y, z) components each. The dimension of the state vector is $\boldsymbol{s} \in \mathbb{R}^{7+7\times(3\times3)=70}$. Note that we do not assume any information on the object (e.g., object pose, geometry, or mass) in the state vector, unlike related works [2], [4].

The action vector $\boldsymbol{a}$ consists of 3 finger position increments, 3 wrist position increments and 3 wrist rotation increments. The action vector's dimension is $\boldsymbol{a} \in \mathbb{R}^{3+3+3=9}$.

The policy $\pi_{\boldsymbol{\theta}}$ is parametrized by a neural network with weights $\boldsymbol{\theta}$. The network is a multi-layer perceptron (MLP) with four layers [70, 256, 256, 9] where the input layer matches the size of the state vector $\boldsymbol{s}$ and the output layer matches the size of the action vector $\boldsymbol{a}$. We use the `stable-baselines3` [18] implementation of the soft actor-critic (SAC) [19] framework to train the stochastic policy $\pi_{\boldsymbol{\theta}}$. We evaluate the policy deterministically when testing.

E. Algorithm Overview

Fig. 1 shows an overview of one training episode. Before starting the control algorithm, we reset the world. Thereby, we randomly generate a new object, wrist error tuple (O, E) (or we select one from the test dataset) and close the fingers of the robotic hand in the erroneous wrist pose until the fingers make contact with the object. Consequently, the grasp refinement episode starts. We divide each episode into three stages, as displayed in Fig. 1. Firstly, the policy $\pi_{\boldsymbol{\theta}}$ *refines* the grasp in five seconds and 15 algorithm steps. Afterward, the agent *lifts* the object by 15 cm via hard-coded increments to the wrist's z-position in two seconds and six algorithm steps. Finally, the policy *holds* the object in place for two seconds and six algorithm steps to test the grasp's stability. The policy $\pi_{\boldsymbol{\theta}}$ can update the wrist and finger positions while lifting and holding. The control frequency of the policy in all stages is 3 Hz, while the update frequency of the low-level proportional–derivative (PD) controllers in the wrist and the fingers is 100 Hz.

Each episode can last at most $15 + 6 + 6 = 27$ algorithm steps. We end the episode earlier if the hand shifts the object by more than 10 cm during the refinement stage to discourage excessive movement of the object. Furthermore, we terminate refinement if one of the fingers exceeds a joint limit of 3 radians. We do not enter the holding stage if the object dropped after the lifting stage. The algorithm trains for 25000 steps, which corresponds to approximately 1000 training episodes depending on the episode lengths.

As shown in the table of Fig. 1, we use the analytic grasp stability metrics from section II as reward functions. We compare the following reward configurations: (1) both ϵ *and* δ, (2) only ϵ, (3) only δ and (4) the baseline β. While ϵ *and* δ, δ, and ϵ provide feedback about grasp stability after every algorithm step, the baseline β gives a sparse reward after the holding stage, indicating if the object is still in the hand (1) or not (0). Since the SAC algorithm is sensitive to reward scaling [19], we normalize the rewards, which are based on grasp quality metrics.

F. Results

Fig. 7 shows the training results of the four reward frameworks. For all experiments in this paper, we average over 40 models trained with different seeds for each framework and smooth training curves with a moving average filter of kernel size 30. The error bars in all plots represent ± 2

standard errors. It takes approximately 20 hours to train one model on a machine with 4 CPUs. We realize from Fig. 7 that the algorithms trained with grasp stability metrics are more sample efficient and reach higher success rates than β within the defined training steps. We also notice that the combination between ϵ *and* δ is particularly helpful for spheres. The algorithms trained with β especially struggle to grasp spheres. Furthermore, the reward framework ϵ initially trains faster than the reward frameworks that include the force agnostic metric δ. Lastly, we recognize that the *Hold Success* and *Lift Success* graphs in Fig. 7 are very similar.

Fig. 8 summarizes the test results. All test results in this paper stem from 38400 grasps (40 models with different seeds $\times$ 4 frameworks $\times$ 240 test cases). Our main observation is that combining the geometric grasp stability metric ϵ with the force-agnostic metric δ yields the highest average success rates of 83.6% across all objects (95.4% for cuboids, 93.1% for cylinders, and 62.3% for spheres) over all wrist errors. The ϵ *and* δ framework outperforms the binary reward framework β by 42.9%. As expected, performance decreases for larger wrist errors. We show results of a one-sided, paired t-test in Table III (mean of framework x is μ_x and '$\approx$ 0.0' means that value was numerically zero).

TABLE III: Results of t-test for reward comparison.

Result	$\mu_{\epsilon \text{ and } \delta} > \mu_{\delta}$	$\mu_{\epsilon \text{ and } \delta} > \mu_{\epsilon}$	$\mu_{\epsilon \text{ and } \delta} > \mu_{\beta}$
p-value	$3.1681\ 10^{-10}$	$2.0510\ 10^{-12}$	≈ 0.0

G. Discussion

This study investigates the potential of analytic grasp stability metrics for robotic grasp refinement. From the results of the t-test in Table III, we conclude that the claim 'the combination of ϵ *and* δ outperforms all other rewards frameworks' is statistically significant ($p < 0.01$ for all comparisons). The results demonstrate that the grasp stability metrics ϵ and δ encode different information and that the algorithm learns to integrate both types of feedback into a stronger overall policy. The low success rates for the spheres may be because they can roll and are therefore harder to grasp (cuboids and cylinders move comparatively less when touched by fingers or the palm). The observation that success rates after the *lift* and the *hold* stage are almost identical means that once the hand successfully lifts the object, the grasp is usually also stable enough to keep the object in hand until the very end of the grasp refinement episode.

The β framework performs worst after the defined number of training steps, which is unsurprising because shaped rewards are known to be more sample efficient than sparse rewards [20]. The β framework may not constitute the best-performing alternative that is not based on analytic techniques from grasp analysis. However, it should be considered as a baseline often integrated into reward functions of related works [2], [3]. Furthermore, the performance of the β framework in Fig. 7 continues to rise slowly, and it would be interesting to evaluate at which success rates it plateaus.

IV. Contact Sensing and Grasp Refinement

A. Experimental Setup

In a second experiment, we investigate the effect of contact sensing on grasp refinement. We compare four contact sensing frameworks. The *full* contact sensing framework receives the same state vector $\boldsymbol{s} \in \mathbb{R}^{70}$ as in section III-D. In the *normal* framework, we only provide the algorithm with the contact normal forces and omit the tangential forces ($\boldsymbol{s} \in \mathbb{R}^{56}$). In the *binary* framework we only give a binary signal whether a link is in contact (1) or not (0) ($\boldsymbol{s} \in \mathbb{R}^{56}$). Finally, we solely provide the joint positions in the *none* framework ($\boldsymbol{s} \in \mathbb{R}^{7}$). We adjust the size of the input layer of the neural network from section III-D to match the size of the state vector of each framework. We keep the rest of the network's architecture fixed to allow a fair comparison. The reward function in these experiments is ϵ *and* δ from Fig. 1. Hence, all contact sensing frameworks receive contact information indirectly via the reward.

B. Results

Fig. 9 shows the training performance of the contact sensing frameworks. Note that the *full* framework is the same as the ϵ *and* δ framework from section III. We can observe that the *none* framework initially learns faster than the other frameworks. However, after approximately 250 episodes, the frameworks that receive contact feedback outperform the *none* framework, which plateaus at a lower success rate.

Fig. 10 compares the test results of the different contact sensing frameworks. We observe that the frameworks which receive contact feedback (*full*, *normal*, *binary*) outperform the *none* framework by 6.3%, 6.6% and 3.7%, respectively. Providing the algorithm with *normal* force information yields a performance increase of 2.9% compared to the *binary* contact sensing framework. However, training with the *full* contact force vector only increases the performance by 2.6% compared to the *binary* framework. Furthermore, the success rates for cuboids and cylinders are higher than for spheres (for the *normal* force framework the success rates are 96.8%, 93.7%, 61.3%, respectively). We show the results of a one-sided, paired t-test in Table IV.

TABLE IV: Results of t-test for contact sensing comparison.

Result	$\mu_{normal} > \mu_{full}$	$\mu_{normal} > \mu_{binary}$	$\mu_{normal} > \mu_{none}$
p-value	0.2232	$7.0177\ 10^{-11}$	$1.3087\ 10^{-46}$

C. Discussion

The second experiment analyzes the effect of contact sensing modalities on grasp refinement performance. Specifically, we test whether the findings in [8], [9] (models trained with normal force feedback perform approximately as well as ones trained with binary contact signals) are reproducible for the grasp refinement task. Similar to other work in the field [2], [8], [9], our main conclusion is that tactile sensing improves performance when training RL agents to grasp. We relate the differences in learning speed to the size of the state vector. The *none* framework has a smaller state vector and

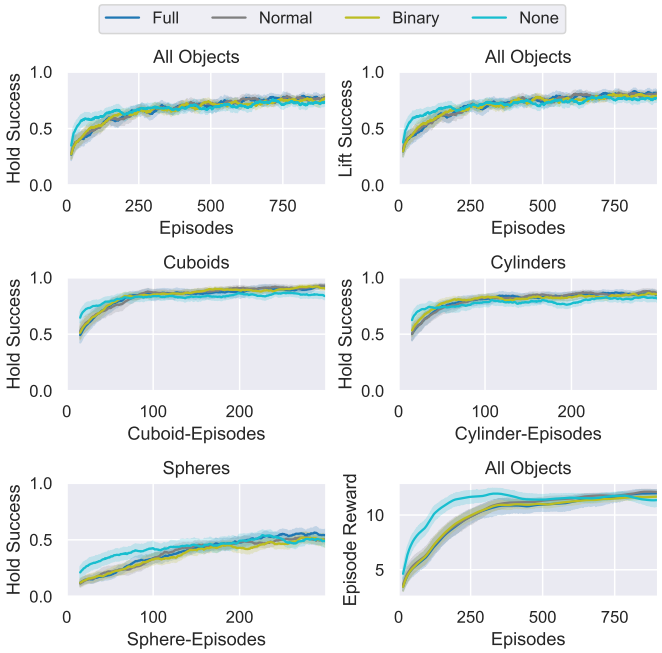


Fig. 9: Training results for contact sensing frameworks.

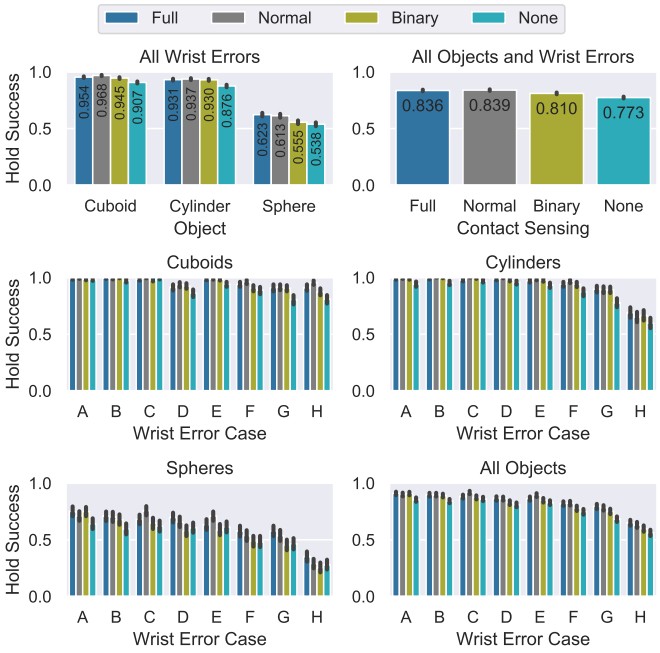


Fig. 10: Test results for contact sensing frameworks.

can hence learn faster, while the frameworks that process contact information require more training data to converge. Furthermore, the surprisingly good performance of the *none* framework means that agents can refine grasps solely based on the crude contact feedback of finger joint position data when trained with rewards that encode grasp stability.

The training curves of the *full*, *normal* and *binary* frameworks in Fig. 9 are hard to distinguish which is also visible in the plots of [8] and [9]. Each data point in the training curves includes the outcome of only one grasp refinement episode per model (one object O and one wrist error W). This punctual evaluation poorly reflects on the *overall* model performance. Therefore, we should focus our analysis on the test results from the 240 experiments per model over multiple objects and wrist errors which provide a more comprehensive model evaluation. In the test results, we observe statistically significant improvements for the *normal* force framework when compared to the *binary* and *none* frameworks (p-values in Table IV < 0.01). However, the accurate *normal* force readings only improve the *binary* framework by a small margin of 2.9%. Hence, our results closely resemble the findings in [8], [9], which concluded that *normal* and *binary* frameworks perform approximately equally well for in-hand manipulation tasks.

Counterintuitively, the algorithms trained with the *full* force vector perform approximately on par with the ones that receive the *normal* force information (the small difference in success rates of 0.3% is not statistically significant because p-value > 0.01 in Table IV). This observation could be due to three reasons. (1) The *full* force framework is the framework with the largest state vector (see section IV-A) and therefore requires the most training data because it has the most network parameters. Future experiments should run more training steps. (2) The models trained with the *full* framework will have to internally represent the concept of the friction cone, which may be a complex notion to learn from discontinuous contact data (sometimes there is contact on a link, sometimes there is not). An alternative representation of the tangential forces could be an exciting avenue for research (e.g., provide margin to the friction cone instead of tangential force vector). (3) Lastly, contact forces in simulated environments are known to be unstable [21], especially when simulating robotic grasping [22]. Hence, another reason for our observation (and for the results in [8], [9]) may be that since simulated contact forces are not always physically meaningful, they may not necessarily constitute a good proxy of grasp success in simulation.

V. Conclusion

This paper investigated the potential of analytic grasp stability metrics as reward functions for RL algorithms that perform tactile grasp refinement on three-fingered robotic hands. We found that the rich body of research in grasp analysis is a valuable toolbox to construct meaningful and sound optimization objectives for RL. Furthermore, we investigated the effect of different contact sensing modalities on grasp refinement performance and raised interesting questions on tactile data processing with RL.

There are several exciting directions for future work. We want to test the learned policies on the real robotic hand and evaluate their sim-to-real performance. Specifically, we would like to investigate whether some reward frameworks transfer better to the real world than others. Future reward functions should also contain a force minimization term. This work mainly examined the effect of the representation of contact forces on grasp refinement. Therefore, future ablation studies should quantify the relevance of contact normal and position sensing.

REFERENCES

[1] Y. Chebotar, K. Hausman, Z. Su, G. S. Sukhatme, and S. Schaal, "Self-supervised regrasping using spatio-temporal tactile features and reinforcement learning," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 1960–1966.

[2] H. Merzić, M. Bogdanović, D. Kappler, L. Righetti, and J. Bohg, "Leveraging contact forces for learning to grasp," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3615–3621.

[3] B. Wu, I. Akinola, J. Varley, and P. Allen, "Mat: Multi-fingered adaptive tactile grasping via deep reinforcement learning," *arXiv preprint arXiv:1909.04787*, 2019.

[4] W. Hu, C. Yang, K. Yuan, and Z. Li, "Reaching, grasping and regrasping: Learning multimode grasping skills," 2020.

[5] D. Silver, S. Singh, D. Precup, and R. S. Sutton, "Reward is enough," *Artificial Intelligence*, vol. 299, p. 103535, 2021.

[6] M. A. Roa and R. Suárez, "Grasp quality measures: review and performance," *Autonomous robots*, vol. 38, no. 1, pp. 65–88, 2015.

[7] C. Ferrari and J. Canny, "Planning optimal grasps," in *Proceedings 1992 IEEE International Conference on Robotics and Automation*, 1992, pp. 2290–2295 vol.3.

[8] A. Melnik, L. Lach, M. Plappert, T. Korthals, R. Haschke, and H. Ritter, "Tactile sensing and deep reinforcement learning for in-hand manipulation tasks," in *IROS Workshop on Autonomous Object Manipulation*, 2019.

[9] ——, "Using tactile sensing to improve the sample efficiency and performance of deep deterministic policy gradients for simulated in-hand manipulation tasks," *Frontiers in Robotics and AI*, vol. 8, p. 57, 2021.

[10] B. Mirtich and J. Canny, "Easily computable optimum grasps in 2-d and 3-d," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*. IEEE, 1994, pp. 739–747.

[11] M. Buss, H. Hashimoto, and J. Moore, "Dextrous hand grasping force optimization," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 3, pp. 406–418, 1996.

[12] Z. Li and S. Sastry, "Task-oriented optimal grasping by multifingered robot hands," *IEEE Journal on Robotics and Automation*, vol. 4, no. 1, pp. 32–44, 1988.

[13] N. Pollard, "Synthesizing grasps from generalized prototypes," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, 1996, pp. 2124–2130 vol.3.

[14] C. Borst, M. Fischer, and G. Hirzinger, "Grasp planning: how to choose a suitable task wrench space," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 1, 2004, pp. 319–325 Vol.1.

[15] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, Sep 2004, pp. 2149–2154.

[16] J. Lee, M. X. Grey, S. Ha, T. Kunz, S. Jain, Y. Ye, S. S. Srinivasa, M. Stilman, and C. K. Liu, "Dart: Dynamic animation and robotics toolkit," *Journal of Open Source Software*, vol. 3, no. 22, p. 500, 2018.

[17] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, Kobe, Japan, May 2009.

[18] A. Raffin, A. Hill, M. Ernestus, A. Gleave, A. Kanervisto, and N. Dormann, "Stable baselines3," https://github.com/DLR-RM/stable-baselines3, 2019.

[19] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," 2018.

[20] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *In Proceedings of the Sixteenth International Conference on Machine Learning*. Morgan Kaufmann, 1999, pp. 278–287.

[21] J. M. Hsu and S. C. Peters, "Extending open dynamics engine for the darpa virtual robotics challenge," in *Proceedings of the 4th International Conference on Simulation, Modeling, and Programming for Autonomous Robots - Volume 8810*, ser. SIMPAR 2014. Berlin, Heidelberg: Springer-Verlag, 2014, p. 37–48.

[22] J. R. Taylor, E. M. Drumwright, and J. Hsu, "Analysis of grasping failures in multi-rigid body simulations," in *2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, 2016, pp. 295–301.