Demonstration-Guided Deep Reinforcement Learning of Control Policies for Dexterous Human-Robot Interaction

Sammy Christen[1], Stefan Stevšić[1], Otmar Hilliges[1]

***Abstract*— In this paper, we propose a method for training control policies for human-robot interactions such as handshakes or hand claps via Deep Reinforcement Learning. The policy controls a humanoid Shadow Dexterous Hand, attached to a robot arm. We propose a parameterizable multi-objective reward function that allows learning of a variety of interactions without changing the reward structure. The parameters of the reward function are estimated directly from motion capture data of human-human interactions in order to produce policies that are perceived as being natural and human-like by observers. We evaluate our method on three significantly different hand interactions: handshake, hand clap and finger touch. We provide detailed analysis of the proposed reward function and the resulting policies and conduct a large-scale user study, indicating that our policy produces natural looking motions.**

I. Introduction

Dexterous humanoid hands, such as the Shadow Dexterous Hand [1], are becoming very sophisticated. Improvements in mechatronics have enabled very compact systems that have more than twenty degrees of freedom (DoF). However, controller design remains very challenging and has been shown to be a very complex problem [2], [3]. Recently, model-free deep reinforcement learning (DRL) algorithms have been applied to the control of humanoid hands, albeit on relatively simple tasks such as grasping or door opening [2] and in simulation only. In [4], a controller trained in simulation has been transferred to a real humanoid hand. This opens up the door to learn policies for natural physical human-robot interactions. In particular, we are interested in learning a control policy for diverse hand interactions, such as handshakes or hand claps. The handshake is the most common greeting gesture throughout the world, therefore it has received a lot of attention in the robotics community [5], [6], [7], [8], [9]. In this paper, we present a method for training a control policy for human-robot hand interactions, using data from human demonstrations in combination with deep reinforcement learning. We test our method on the simulated model of the Shadow Dexterous Hand.

To train a control policy using a DRL algorithm, one of the main issues is the definition of a reward function. For simple tasks, like grasping or pick and place tasks, the goal is obvious and the reward can be easily shaped. For our task, however, it is not obvious how to shape a reward function.

[1]AIT Lab, Department of Computer Science, ETH Zurich, 8092 Zurich, Switzerland sammy.christen | stefan.stevsic | otmar.hilliges @inf.ethz.ch

This work was supported in parts by the Swiss National Science Foundation (UFO 200021L_153644). We thank the NVIDIA Corporation for the donation of GPU servers used in this work.

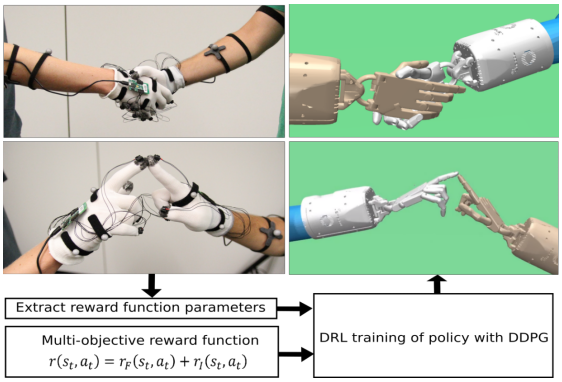

Fig. 1. Approach overview. The proposed multi-objective reward function and extracted parameters from human interaction data are used to train a human-robot interaction control policy via DRL.

The reward needs to result in motions that are perceived as natural, the hand needs to reach a desired contact profile and precise position, while dealing with complex contact dynamics. To produce natural looking motions of animated characters in [10], the reward is based on tracking position and angle references from motion capture data. However, the authors only consider motions in open space and hence their reward function cannot be transfered to our task. Thus, we investigate important terms to construct a reward function and compare the influence of different reward terms in an ablation study. To enable generalization to different hand interactions, we define a parametrized reward function. We extract most of the reward function parameters from motion capture data, leading to only six parameters that are relatively easy to adjust. One could argue that the policy could be learned directly from data via Inverse Reinforcement Learning (IRL). However, state-of-the-art IRL methods [11], [12] have not been applied to tasks that require precise positioning or challenging contact dynamics. Furthermore, these methods can be unstable when applied to motion capture data [12]. To ensure training convergence, we propose a specialized training method. Standard DRL algorithms work out of the box on benchmark problems [13], but for more complex problems additional training details, such as randomization or early stopping are important [10], [14], [2]. We propose a training method which works in combination with DDPG, resulting in stable convergence properties.

This work presents a method for learning control policies for dexterous human-robot interaction. More specifically, we contribute the following: (i) A multi-objective reward function for DRL algorithms. We show how reward func-

tion parameters are extracted from motion capture data and provide detailed analysis of how different parts of the reward influence the resulting control policy. (ii) A training method which works in combination with standard DRL algorithms. (iii) A dataset of human hand interactions. (iv) A large-scale user study showing that adding imitation reward to the policy results in motions that are perceived as more natural.

II. Related Work

A. Human Hand Interaction

Different aspects of the human-robot handshake problem were investigated in the robotics community, e.g., force properties of a human handshake [5], the possibility to recognize personality and gender from a handshake [6], or the design of a compliant controller for handshakes [7]. Previous work mostly focuses on the handshake properties after the contact phase. However, producing a handshake movement is equally important [8], [9]. When humans establish a handshake, one person requests the handshake by holding out one hand, while the other person responds by grabbing the hand [9]. Based on this observation, [9] tries to model the appropriate time to request a handshake.

To achieve a handshake with humanoid hands, usually the robot requests the handshake and closes the fingers when the human hand is in contact [7]. To the best of our knowledge, our method is the first that treats the problem in the case where the human requests the handshake. This case is harder from a control perspective, because it requires coordination with the human hand, the robot needs to produce natural looking motions and it still involves physical contact as in the previous case. The control of robotic arm movement is investigated in [8], [9], but these papers do not control humanoid hands and do not observe complex contact dynamics. Furthermore, we investigate the possibility of performing different hand interactions, such as hand claps or finger touches, which are relatively under-researched.

B. Control of Dexterous Humanoid Hands

Dexterous humanoid hands are a highly complex mechanical systems [15]. Due to their high complexity, the control problem is shown to be very challenging [2], [3]. Most approaches therefore use trajectory optimization to provide a controller [3], [16], [17]. These approaches require a precise model of the system, which makes them hard to transfer to real robots. Leveraging real robot data for model learning was proposed in [18], but the method is limited to slow in-hand manipulation of a pole. Contrary, model-free DRL does not require a model of the robot dynamics, i.e., all information is obtained through multiple episodes of trial-and-error. The only input to the algorithm is a reward function. Recently, model-free DRL has been applied to the control problem of humanoid hands [2], achieving impressive results in a simulated environment. Furthermore, [4] demonstrates the possibility of transferring a policy trained in simulation to the real Shadow Dexterous Hand.

For non-linear control tasks, model-free DRL algorithms have shown impressive performance [19], [20], [21] on the OpenAI gym benchmark problems [13]. Furthermore, when applied to low degree of freedom robotic manipulators (7-10 DoF), like robotic hands with grippers, model-free DRL has been leveraged successfully [22], [14], [23]. However, the problem becomes more challenging when DRL is applied to systems with higher DoF. A hand control problem [2] or natural movement character control problem [10] deal with such systems. In [10], the authors carefully design and adjust the weights of the reward to achieve the desired performance. To improve convergence properties, the authors use two training techniques: setting the initial state on the demonstration trajectory and early stopping. Human demonstration can be used to accelerate the convergence rate [2]. In [2], a human operator provides demonstrations via teleoperation, which are used to initialize the policy. Alternatively, IRL methods [24], [25], [26] can learn the reward function from demonstration data. However, they require running the RL algorithm in the inner loop of an iterative method, which is not feasible with DRL algorithms. To overcome this issue, IRL is posed as an adversarial imitation problem [12], [11], but these methods are prone to instability.

Our method is inspired by previous DRL approaches, but our task requires significant changes of existing methods. In [2], the demonstrations are provided by teleoperating a simulated robot hand, operating in isolation. For our case, it is impractical to collect demonstrations this way because it requires interactions between two humans. Thus, we capture real interactions using a motion capture system. Our method is more similar to [10], which is not designed for humanoid hands. In [10], the authors use motion capture data directly in the reward function formulation. Contrary, we extract the final pose parameters from data, while we similarly use motion capture data to produce natural looking motions. Additionally, we add contact patterns as an objective to the reward function, and provide a different training method.

III. Preliminaries

A. Deep Reinforcement Learning

Our control problem can be formalized using Markov Decision Processes (MDP), defined as a tuple $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \rho_0, \gamma\}$. We describe an environment with a set of states $\mathcal{S}$, a set of actions $\mathcal{A}$, a reward function $\mathcal{R} = r(s_t, a_t)$, transition dynamics $\mathcal{T} = p(s_{t+1}|s_t, a_t)$, an initial state distribution $\rho_0 = p(s_1)$ and a discount rate $\gamma \in [0, 1]$, where $s_t \in \mathcal{S}$ and $a_t \in \mathcal{A}$. Model-free RL does not require knowledge of transition dynamics and requires only sampling from the transition dynamics probability distribution. We define the return R_t as a discounted sum of future rewards:

$$R_t = \sum_{i=t}^{T} \gamma^{i-t} r(s_i, a_i). \tag{1}$$

The controller is defined as a control policy π, which maps states to actions $\pi : \mathcal{S} \rightarrow \mathcal{A}$. The goal of the reinforcement learning algorithm is to learn a policy π which maximizes the expected return from the start distribution:

$$J = \mathbb{E}_{\mathcal{M}} [R_1] \tag{2}$$

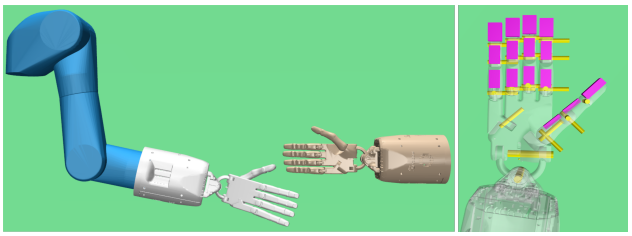

Fig. 2. Agent arm with the Shadow Dexterous Hand. The system has 28 DoF. The arm, shown in blue in the left figure, has 4 Dof. The hand has 24 Dof, depicted as yellow cylinders in the right figure, and 20 actuators. Contact sensors are marked in purple (right).

We define the action-value or Q-function, which describes the expected return under a policy π when taking action a_t from state s_t, also called state-action pair, as follows:

$$Q(s_t, a_t) = \mathbb{E}_{\mathcal{M}} \left[R_t | s_t, a_t \right]. \tag{3}$$

To solve the given problem, we define the Q-function and policy as neural network function approximators parametrized with θ^Q and θ^π. This is known as an actor-critic type of RL algorithm, since we learn both an actor function, i.e., the policy, and a critic function, i.e., the Q-function. More specifically, we use the DDPG algorithm [19] to compute the gradients for updating the neural network parameters. To update θ^Q, we minimize the loss:

$$L(\theta^Q) = \mathbb{E}_{\mathcal{M}, a_t} \left[\left(Q(s_t, a_t; \theta^Q) - y_t \right)^2 \right] \tag{4}$$

$$y_t = r(s_t, a_t) + \gamma Q(s_{t+1}, a_{t+1}; \theta^Q) \tag{5}$$

To update the actor parameters θ^π, we compute the gradients:

$$\nabla_{\theta^\pi} J = \mathbb{E}_{\mathcal{M}} \left[\nabla_{\theta^\pi} Q(s_t, a_t; \theta^Q) | s_t, a_t = \pi(s_t; \theta^\pi) \right], \tag{6}$$

which are applied to the actor neural network. For both networks we use three fully connected layers with 256 neurons and ReLu activation functions.

To ensure convergence of the policy, we apply all techniques from the DDPG paper to stabilize convergence properties. This includes a replay buffer, batch normalization and target networks. DDPG is an off-policy algorithm, thus we define the exploration policy as:

$$a_t = \pi(s_t) + \mathcal{N}, \tag{7}$$

where $\mathcal{N}$ is a sample from zero mean Normal distribution. More implementation details can be found in [19].

B. Simulation Environment

Our simulation environment consist of two robots: the agent, controlled by the policy, and the target hand. The agent consists of a 4 DoF robotic arm and the Shadow Dexterous Hand with 24 DoF. The Shadow Dexterous Hand is controlled by 20 actuators (cf. Fig. 2). We use the same hand model as stand-in for a human hand for convenience, since this model is easy to pose in different configurations. However, this model can be replaced with a human hand model, which should not influence the results of our experiments since the hand is not actuated, as usually assumed for a hand requesting an interaction [8], [9]. The robot models are taken from the OpenAI gym framework [13].

The input to the control policy are joint angles, joint velocities and contact sensor readings (cf. Fig. 2) of the agent hand. Additional inputs are the positions of the target hand links and the origin of each rigid body on the hand. The policy outputs are control signals that actuate the agent's arm and hand. Control signals are setpoints for the joint angles scaled in the range from -1.0 to 1.0.

IV. Method

Our method is able to learn control policies of hand interactions using motion capture data of human demonstrations. We assume the following setting: the first participant requests the interaction while the second is executing the interaction sequence. In the example of a handshake, the first participant stretches out the hand to request the handshake, while the other responds by grabbing the hand. The robot learns to perform the behavior of the second participant. The goal is to produce the desired interaction *and* motions perceived as natural. We propose a *single* reward function that can be applied to various hand interactions. The parameters of the reward function are extracted directly from the motion capture dataset using Alg. 1. The policy is trained via a DDPG based training method, as explained in Sec. IV-B.

A. Reward Function

Our proposed reward function consists of two terms:

$$r(s_t, a_t) = r_F(s_t, a_t) + r_I(s_t, a_t), \tag{8}$$

where $r_F(s_t, a_t)$ is the final state reward, which is used to reward the correct end configuration, and the imitation state reward $r_I(s_t, a_t)$, which provides trajectory guidance to make the interactions look more natural. The final state reward itself consists of four terms:

$$r_F(s_t, a_t) = r_p(s_t) + r_\alpha(s_t) + r_c(s_t) + r_a(a_t), \tag{9}$$

where $r_p(s_t)$ is a position reward, $r_\alpha(s_t)$ is an angle reward, $r_c(s_t)$ is a contact reward, and $r_a(a_t)$ penalizes high action inputs. Experimentally, we determined that the most important position features are the fingertip positions of the agent hand (total number $N_f = 5$). Regarding the angles, we use all joint angles of the robot hand to compute the angle reward (total number $N_\alpha = 24$). Position and angle rewards are defined as a negative l_2 norm of position features and angle errors:

$$r_p(s_t) = -\sum_{i=1}^{N_f} \omega_p^i ||p_g^i - p_{rt}^i|| \tag{10}$$

$$r_\alpha(s_t) = -\sum_{i=1}^{N_\alpha} \omega_\alpha^i ||\alpha_g^i - \alpha_{rt}^i||. \tag{11}$$

The vector p_g^i is the goal position of each position feature and p_{rt}^i is the current position of the respective feature. Similarly, α_g^i is the goal joint angle and α_{rt}^i the current joint angle on the robot hand. The weights ω_p^i, ω_α^i determine the importance

of the specific goal, which we define via algorithm described in Sec. IV-A.1. When using only position and angle rewards, the robot hand only roughly reaches the desired end configuration (cf. Fig 4, Baseline 2). Our task requires accurate hand positioning, which is hard to achieve in tasks that involve contacts. We achieve the desired hand position by adding a contact reward $r_c(s_t)$, which forces the desired contact profile. For more details about the influence of the reward terms we refer to Sec. V-A. Additionally, a control input reward is added to prevent high control signals. Contact and input rewards are defined as:

$$r_c(s_t) = \sum_{i=1}^{N_c} \omega_c^i \mathbb{1}_{ct}^i \quad , \tag{12}$$

$$r_a(a_t) = -\sum_{i=1}^{N_a} \omega_a^i ||a^i||^2 \quad , \tag{13}$$

where the indicator function $\mathbb{1}_{ct}^i$ outputs 1 in case the contact sensor is active and 0 otherwise. The weight ω_c^i determines the importance of each contact sensor. The system has $N_c = 15$ contact sensors. a^i is the control input signal for each actuator and ω_a^i is the respective weight. The system has $N_a = 24$ control inputs.

The imitation reward consist of two further terms:

$$r_I(s_t, a_t) = r_{pI}(s_t) + r_{\alpha I}(s_t). \tag{14}$$

The difference compared to the final state reward is that the goal position p_{gt}^i and goal angles α_{gt}^i depend on the timestep:

$$r_{pI}(s_t) = -K_{pI} \sum_{i=1}^{N_f} \omega_p^i ||p_{gt}^i - p_r^i|| \quad , \tag{15}$$

$$r_{\alpha I}(s_t) = -K_{\alpha I} \sum_{i=1}^{N_\alpha} \omega_\alpha^i ||\alpha_{gt}^i - \alpha_r^i||. \tag{16}$$

These rewards are scaled with the weights K_{pI} and $K_{\alpha I}$.

The final state reward function is often enough to complete the interaction, i.e., to reach the final pose. However, the most direct trajectory often does not look natural (cf. Fig. 6). We overcome this issue by adding an imitation reward. As shown in Sec. V-B, imitation reward significantly improves that the hand motion is perceived as natural. However, when the starting hand pose is far away from poses in demonstration examples, the policy may produce non-smooth motions. We evaluate these examples also in Sec. V-B.

1) Reward function parameters: To define the terms of the reward function, we use a motion capture dataset $\mathcal{D} = \{(p_t^i, p_t^j, \alpha_t^i, \alpha_t^j)\}_{t=1}^T$ of an interaction sequence. The dataset provides positions of rigid bodies p_t^i, p_t^j and joint angles α_t^i, α_t^j of two human hand models at timestep t: the target hand, denoted with superscript j, which requests the interaction and the actor hand, denoted with superscript i, which the robot imitates. From $\mathcal{D}$, we can calculate distances between the rigid bodies d_t^{ij} and the relative position calculated in the coordinate frame of the target hand body Δp_t^{ij}. We use fingertip positions on both hands, plus the palm position on the target hand. For the joint angles, we use all joints from the human data that have a corresponding joint on the robot hand. Based on the minimum distance, we set the reference frame $j_{\min}$ and reference timestep $t_{\min}$, as shown in Alg. 1. Using these references, we compute the goal positions and goal angles. The position goals are defined in a goal centric way, which enables us to calculate them for a randomly positioned hand in the simulation p_s^j, as shown in line 5 of Alg. 1, where R_s^j is the rotation matrix of the target hand rigid body with index j.

Algorithm 1 Reward Parameters

1: **Input:** $\mathcal{D}, d_t^{ij}, \Delta p_t^{ij}$
2: $t_{\min} \leftarrow \arg\min d_t^{ij}, \quad j_{\min} \leftarrow \arg\min d_t^{ij}$
3: $p_s^{j_{\min}} \leftarrow$ position of a $j_{\min}$ link in simulation
4: $R_s^{j_{\min}} \leftarrow$ rotation matrix of a $j_{\min}$ link in simulation
5: $p_g^i \leftarrow p_s^{j_{\min}} + R_s^{j_{\min}} \Delta p_{t_{\min}}^{ij_{\min}}, \quad \alpha_g^i \leftarrow \alpha_{t_{\min}}^i$
6: $p_{gt}^i \leftarrow p_s^{j_{\min}} + R_s^{j_{\min}} \Delta p_t^{ij_{\min}}, \quad \alpha_{gt}^i \leftarrow \alpha_t^i$
7: **return** $p_g^i, \alpha_g^i, \{\alpha_{gt}^i | t = 1..t_{\min}\}, \{p_{gt}^i | t = 1..t_{\min}\}$

Position weights are calculated using the equation:

$$\omega_p^i = K_p \frac{d_{\min}}{d_{t_{\min}}^{ij_{\min}}}. \tag{17}$$

The angle and control weights all have the same value $\omega_\alpha^i = K_\alpha$, $\omega_a^i = K_a$. We set the contact weights ω_c^i to K_c for all sensors that should be in contact. This can be done by asking participants where they feel the pressure during interactions. Alternatively, one could use the method from [5], which simply applies color to the target hand and measures contact area from paint marks.

To train the policy, we need to set just six weights in the reward function ($K_p, K_\alpha, K_c, K_a, K_{pI}, K_{\alpha I}$). In all our experiments, K_a is set to 1, while K_c is roughly set to $\frac{N_o}{5}$, where N_o is the number of sensors that should be in contact.

B. Training

To train the policy, we first need to position the target hand. For a single training episode, the target hand stays fixed. Although we train the policy with a static hand, we experimentally show that our policy generalizes to moving hands (cf. V-C). The joint angles of the target hand are set according to the joints of the human target hand at a timestep which occurs prior to the interaction timestep $t_{\min}$. This ensures that the target hand is not closed, thus allowing the robot to interact. Our reward function is defined in a goal centric way. This enables randomization of the target hand position, performed at the beginning of each episode. To calculate the reward function parameters, we pick an interaction sequence uniformly at random.

We randomize the robot hand position additionally to target hand randomization. For imitation reward to be effective, the robot hand should be positioned in the same configuration as the human hand at the start of the imitation trajectory. Thus, we position the robot wrist to the position of the human wrist at timestep t_s, augmented with random Gaussian noise. The timestep t_s is selected uniformly at random from a

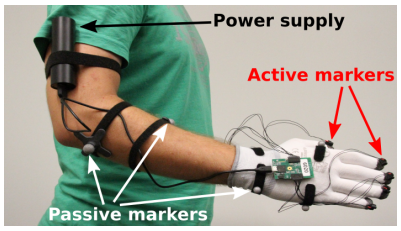

Fig. 3. Participants wear 5 active markers on the fingertips and 6 passive markers on the palm and forearm of the right hand.

set $\{t_k | k = 1..(t_{\min} - t_{\text{off}})\}$. We use a small offset t_{off} because hands can collide in the last part of the trajectory. If this position is not reachable, we start from the closest reachable position. Contrary to [10], we cannot position the agent exactly in the human pose because of the configuration differences of the human and robot arms. Furthermore, our task is driven by a goal pose, which means that starting only from demonstration trajectories, as in [10], will result in poor generalization. After each N_e steps, we update the network using the DRL algorithm described in Sec. III-A.

C. Data Collection

To collect data, we use the OptiTrack motion capture system. Each participant is equipped with markers as shown in Fig. 3. We only track the right hand of each participant. Hand tracking is prone to marker mislabeling, with fingertip markers being most problematic. We use active markers, which can be uniquely identified by their blinking pattern, on the fingertips. The OptriTrack software fits a model of the human hand to the markers, providing the position of each link and joint angle of the human hand. For each interaction, we recorded five demonstrations. We will release the dataset and simulation environment for further research (https://ait.ethz.ch/projects/2019/DRL-handshake/).

V. Results

We conducted experiments in simulation to evaluate our method. We extensively test our policy on three different hand interactions: handshake, hand clap, and E.T. greeting (e.g. index finger touching), see Fig. 4. These three interactions are diverse: the handshake requires grasping of the target hand in a specific way, the hand clap has a characteristic motion prior to contact, while the E.T. greeting requires precise positioning of the index finger.

A. Ablation Study on Reward Function

In a first experiment, we intend to show the influence of the different parts of our reward function on the resulting control policy. For this, we do an ablation study on our reward function. We compare the full final state reward $r_F(s_t, a_t)$ with two baselines. Baseline 1 uses only the relative position of the palm instead of the fingertips as a goal position feature, while keeping angle, contact and input reward the same. In Baseline 2, we remove the contact reward from the reward function. Furthermore, we examine the influence of adding the imitation reward to the final state reward.

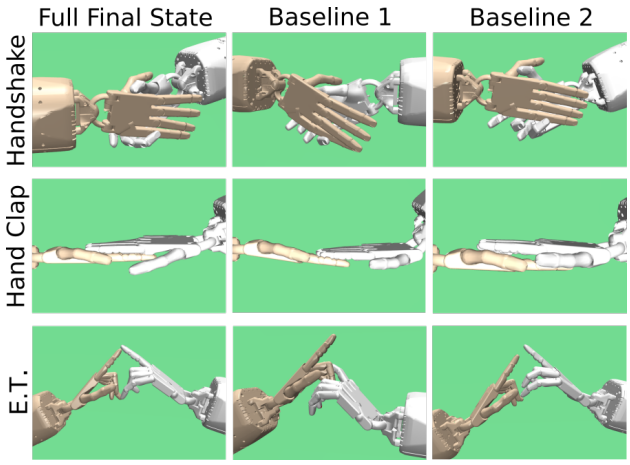

Fig. 4. Final poses of the hands for different hand interactions.

Our final state reward function shows overall better performance than both baselines (c.f. Fig. 5). The influence of the position reward can be seen by comparing Baseline 2 to Baseline 1 for handshake and E.T. interactions. Although both baselines have low success rates, Baseline 2 results in final configurations closer to the desired ones as shown in Fig. 4. Adding contact reward to Baseline 2, i.e. using our reward function, removes these errors. The importance of the contact reward can be also seen in Fig. 5 in case of the hand clap. Since precise positioning is less important here, Baseline 1 achieves high success rate because of the contact reward. After adding the imitation reward, we observe that the success rates do not significantly change.

B. Evaluation of Imitation Training

To qualitatively assess the impact of the imitation reward, we conduct a large-scale user study ($N = 116$). We present 11 video sequences of policy outputs with and without imitation reward side-by-side . We keep the initial conditions for each sequence-pair the same and randomly assign videos to the left or right. The participants state which video is perceived as more natural on a forced alternative choice 5-point scale. The five responses are: "Left sequence looks much more natural", "Left sequence looks more natural", "Both the same", "Right sequence looks more natural", "Right sequence looks much more natural".

Assuming equidistant intervals, we mapped user responses onto a scale from -2 to 2, where positive values mean that the user prefers the policy generated with imitation reward and vice versa. The imitation policy can generate non-smooth motions when the starting pose is far away from the recorded human trajectory. To evaluate these examples, we compare two sequences including obvious non-smooth motions.

Generally speaking, participants favor policies generated with imitation reward (c.f. Table I). For hand claps, the differences are easy to spot (see Fig. 6). Hence, human raters strongly prefer the imitation based policy. For the handshake, the differences are harder to see, resulting in significant amount of participants selecting "Both the same"

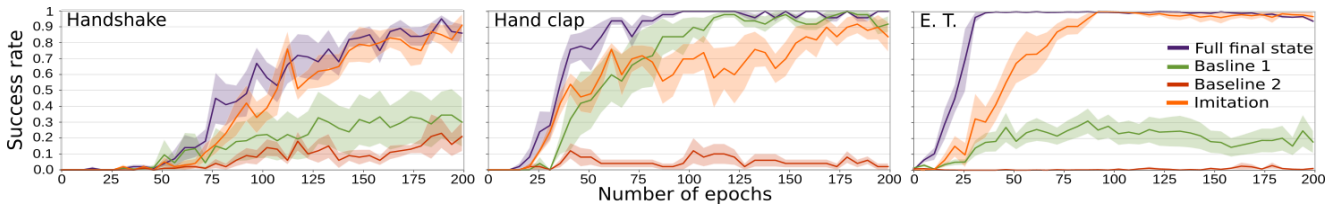

Fig. 5. Baseline comparison. We measure the success rate of the policy every five epochs. Success is estimated from the contact profile on the hand. We show average results from 5 random seeds with the standard error indicated by the shaded area.

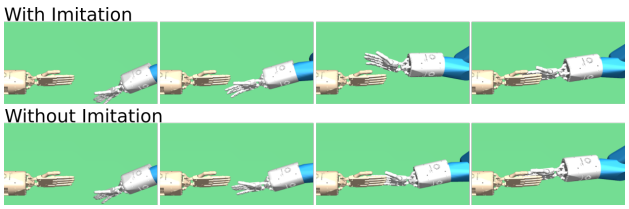

Fig. 6. Imitation reward. The policy trained with imitation reward produces a characteristic motion prior to the hand clap.

TABLE I

USER STUDY RESULTS OF PEOPLE VOTING ON 5 POINT SCALE FROM -2 (NO IMITATION) TO 2 (IMITATION) LOOKING MUCH MORE NATURAL

Score	−2	−1	0	1	2	mean
Handshake	4.9%	26.2%	15.2%	**45.1%**	8.6%	0.26
Hand clap	0.0%	2.3%	3.4%	41.9%	**52.3%**	1.44
E.T.	3.4%	17.2%	**54.3%**	19.0%	6.0%	0.07
Handshake non-smooth	23.3%	**43.1%**	12.9%	15.5%	5.1%	−0.64
Hand clap non-smooth	0.8%	7.0%	12.9%	**49.1%**	30.2%	1.0

(15.2%). However, the majority of the participants still prefer the imitation based policy. For the E.T. interactions, there are no observable differences. Thus, the majority of participants answered with "Both the same" (54.3%). For non-smooth handshakes, the results indicate that participants prefer smooth motions. However, for non-smooth hand claps, participants prefer imitation features, although the mean score is lower than in the case of smooth hand claps.

C. Policy Evaluations

To evaluate the robustness of policies created with our method, we test the reaction to perturbations in orientation and velocity of the target hand. During training, we only randomize the position of the target hand, while the orientation stays the same. We position the target hand in the reachable workspace of the agent and measure the success rate, while changing the yaw and pitch angles of the target hand. In a realistic scenario, the human hand will not be perfectly still. Furthermore, the human can react when the contact is imminent by closing the hand or approaching the agent hand. Thus, we conduct a second experiment where the target hand is moving at constant speed, changing the direction at random every half second (see Fig. 7).

The experiments indicate that our method is robust to perturbations in orientation and velocity of the target hand. This shows that our reward function generates policies that generalize well to unseen scenarios. We also tested our policy with changing configurations of the target hand, i.e., closing fingers in a handshake policy, and did not observe any major changes. Demonstrations of these experiments can be seen in the accompanying video (https://youtu.be/ZSgEqyltaN4).

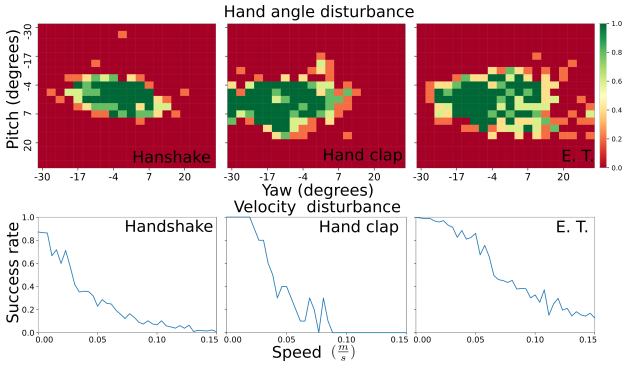

Fig. 7. Robustness experiments. Top: The average success rate for different target hand angles (the color bar shows the success rate). Down: The average success rate for different hand velocities.

VI. DISCUSSION AND CONCLUSION

Control of dexterous humanoid hands is a challenging problem, especially when it involves contact dynamics. In this paper, we demonstrate that a single parametrized reward function can be used for different hand interactions. To define parameters of the reward function, we use a simple algorithm to extract parameters from motion capture data. We show that policies generated with our method produce more natural looking trajectories, and generalize well to different orientations and velocities of the target hand.

Our results are shown only in simulation and with a static target hand as an initial step towards natural human-robot hand interactions. To achieve this level of performance on a real robot, transfer learning methods, such as the one suggested in [4], could be applied. We show that our policy reacts well to small velocity disturbances. However, humans can perform synchronous hand motions prior to interaction. This problem should be investigated in more detail. Our method only considers contacts, but we never investigated forces acting on the hand. [5] emphasizes the importance of forces applied during handshakes. According to our measurements, forces applied to the target hand are in the range of a normal handshake. Compliant behavior is important for hand interactions [7]. However, evaluation of the robot hand compliance is outside the scope of our work.

This paper shows how natural human-robot hand interaction can be learned using DRL. To the best of our knowledge, this is the first paper that uses a dexterous humanoid hand for human-robot hand interactions. This opens up the possibility to achieve natural hand interactions on a real humanoid robot.

REFERENCES

[1] "Shadow dexterous hand," https://www.shadowrobot.com/products/dexterous-hand/, accessed: 2018-08-21.

[2] A. Rajeswaran, V. Kumar, A. Gupta, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," *arXiv preprint arXiv:1709.10087*, 2017.

[3] I. Mordatch, Z. Popović, and E. Todorov, "Contact-invariant optimization for hand manipulation," in *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*. Eurographics Association, 2012, pp. 137–144.

[4] M. Andrychowicz, B. Baker, M. Chociej, R. Jzefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba, "Learning dexterous in-hand manipulation," *arXiv preprint arXiv:1808.00177*, 2018.

[5] E. Knoop, M. Bächer, V. Wall, R. Deimel, O. Brock, and P. Beardsley, "Handshakiness: Benchmarking for human-robot hand interactions," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 4982–4989.

[6] P.-H. Orefice, M. Ammi, M. Hafez, and A. Tapus, "Let's handshake and i'll know who you are: Gender and personality discrimination in human-human and human-robot handshaking interaction," in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*. IEEE, 2016, pp. 958–965.

[7] M. Arns, T. Laliberté, and C. Gosselin, "Design, control and experimental validation of a haptic robotic hand performing human-robot handshake with human-like agility," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 4626–4633.

[8] T. Shu, X. Gao, M. S. Ryoo, and S.-C. Zhu, "Learning social affordance grammar from videos: Transferring human interactions to human-robot interactions," *arXiv preprint arXiv:1703.00503*, 2017.

[9] M. Jindai, S. Ota, Y. Ikemoto, and T. Sasaki, "Handshake request motion model with an approaching human for a handshake robot system," in *Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), 2015 IEEE 7th International Conference on*. IEEE, 2015, pp. 265–270.

[10] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *arXiv preprint arXiv:1804.02717*, 2018.

[11] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Advances in Neural Information Processing Systems*, 2016, pp. 4565–4573.

[12] J. Merel, Y. Tassa, S. Srinivasan, J. Lemmon, Z. Wang, G. Wayne, and N. Heess, "Learning human behaviors from motion capture by adversarial imitation," *arXiv preprint arXiv:1707.02201*, 2017.

[13] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *CoRR*, vol. abs/1606.01540, 2016. [Online]. Available: http://arxiv.org/abs/1606.01540

[14] M. Vecerík, T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. Heess, T. Rothörl, T. Lampe, and M. A. Riedmiller, "Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards," *CoRR, abs/1707.08817*, 2017.

[15] V. Kumar, Z. Xu, and E. Todorov, "Fast, strong and compliant pneumatic actuation for dexterous tendon-driven hands," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1512–1519.

[16] Y. Bai and C. K. Liu, "Dexterous manipulation using both palm and fingers," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1560–1565.

[17] V. Kumar, Y. Tassa, T. Erez, and E. Todorov, "Real-time behaviour synthesis for dynamic hand-manipulation," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6808–6815.

[18] V. Kumar, E. Todorov, and S. Levine, "Optimal control with learned local models: Application to dexterous manipulation," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 378–383.

[19] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[20] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International Conference on Machine Learning*, 2015, pp. 1889–1897.

[21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[22] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 3389–3396.

[23] A. A. Rusu, M. Večerík, T. Rothörl, N. Heess, R. Pascanu, and R. Hadsell, "Sim-to-real robot learning from pixels with progressive nets," in *Conference on Robot Learning*, 2017, pp. 262–270.

[24] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 1.

[25] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning." in *Aaai*, 2008, pp. 1433–1438.

[26] M. Wulfmeier, P. Ondruska, and I. Posner, "Maximum entropy deep inverse reinforcement learning," *arXiv preprint arXiv:1507.04888*, 2015.