Appeared In
Proc. International Symposium on Robotics Research, 2015

Learning Dynamic Robot-to-Human Object Handover from Human Feedback

Andras Kupcsik, David Hsu, Wee Sun Lee

Abstract Object handover is a basic, but essential capability for robots interacting with humans in many applications, e.g., caring for the elderly and assisting workers in manufacturing workshops. It appears deceptively simple, as humans perform object handover almost flawlessly. The success of humans, however, belies the complexity of object handover as collaborative physical interaction between two agents with limited communication. This paper presents a learning algorithm for *dynamic* object handover, for example, when a robot hands over water bottles to marathon runners passing by the water station. We formulate the problem as contextual policy search, in which the robot learns object handover by interacting with the human. A key challenge here is to learn the latent reward of the handover task under *noisy* human feedback. Preliminary experiments show that the robot learns to hand over a water bottle naturally and that it adapts to the dynamics of human motion. One challenge for the future is to combine the model-free learning algorithm with a model-based planning approach and enable the robot to adapt over human preferences and object characteristics, such as shape, weight, and surface texture.

1 Introduction

In the near future, robots will become trustworthy helpers of humans, performing a variety of services at homes and in workplaces. A basic, but essential capability for such robots is to fetch common objects of daily life, e.g., cups or TV remote controllers, and hand them to humans. Today robots perform object handover in a limited manner: typically the robot holds an object statically in place and waits for

Andras Kupcsik
School of Computing, National University of Singapore, e-mail: kupcsik@comp.nus.edu.sg

David Hsu
School of Computing, National University of Singapore, e-mail: dyhsu@comp.nus.edu.sg

Wee Sun Lee
School of Computing, National University of Singapore, e-mail: leews@comp.nus.edu.sg

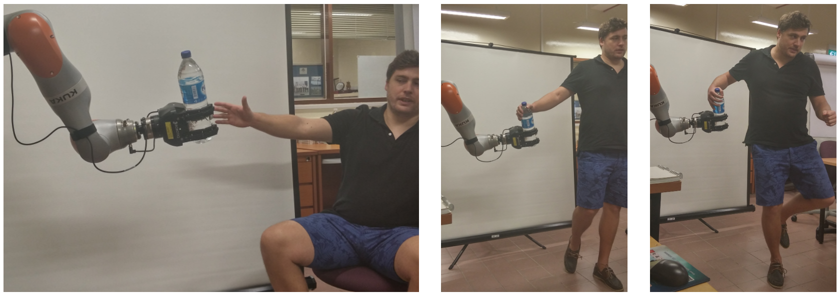

Fig. 1 Hand over a water bottle to a person sitting, walking, or running.

the human to take it. This is far from the fluid handover between humans and is generally inadequate for the elderly, the very young, or the physically weak who require robot services. The long-term goal of our research is to develop the algorithmic framework and the experimental system that enable robots to perform *fluid* object handover in a *dynamic* setting and to adapt over human preferences and object characteristics. This work takes the first step and focuses on a robot handing over a water bottle in a dynamic setting (Fig. 1), e.g., handing over flyers to people walking by or handing over water bottles to marathon runners.

Object handover appears deceptively simple. Humans are experts at object handover. We perform it many times a day almost flawlessly without thinking and *adapt* over widely different contexts:

- *Dynamics:* We hand over objects to others whether they sit, stand, or walk by.
- *Object characteristics:* We hand over objects of different shape, weight, and surface texture.
- *Human preferences:* While typical human object handover occurs very fast, we adapt our strategy and slow down when handing over objects to the elderly or young children.

The success of humans, however, belies the complexity of object handover as collaborative physical interaction between two agents with limited communication. Manually programming robot handover with comparable robustness and adaptivity poses great challenge, as we lack even a moderately comprehensive and reliable model for handover in a variety of contexts.

Alternatively, the robot can learn the handover skill by interacting with the human and generalize from experience. In this work, we formulate the learning task as *contextual policy search* [19]. Policy search is a general approach to reinforcement learning and has been very successful in skill learning for robot with many degrees of freedom [11]. Policy search algorithms parametrize robot control policies and search for the best parameter values by maximizing a reward function that captures the policy performance. Contextual policy search introduces a set of *context variables* that depend on the task context, e.g., object type or size for the handover task, and the policy parameters are conditioned on the context variables.

A reward function that accurately measures policy performance is key to the success of policy search. However, handcrafting a good reward function is often tedious and error-prone, in particular, for learning object handover. It is unclear what quantitative measures capture fluid object handover. Instead, we propose to learn the latent reward function from human feedback. Humans are experts at object handover and can easily provide reward feedback. However, the feedback is often noisy. To be robust against noise and avoid overfitting, we apply a Bayesian optimization approach to latent reward learning. Importantly, our learning algorithm allows for both *absolute feedback*, e.g., "Is the handover good or bad?", and *preference feedback* , e.g., "Is the handover better than the previous one?". Combining latent reward learning and policy search leads to a holistic contextual policy search algorithm that learns object handover directly from human feedback. Our preliminary experiments show that the robot learns to hand over a water bottle naturally and that it adapts to the dynamics of human motion.

2 Related Work

2.1 Object Handover

Object handover has intrigued the research community for a long time from the both physical and social-cognitive perspectives. Early work on handover dates back to at least 1990s [1, 21]. Recent work suggests that object handover consists of three stages conceptually: approach, signal, and transfer [26]. They do not necessarily occur sequentially and may partially overlap. In the first stage, the giver approaches the taker and poses the object to get ready for handover [4, 20, 25]. In the second stage, the giver and taker signal to each other and exchange information, often through non-verbal communication, such as motion [12], eye gaze, or head orientation [13], in order to establish joint intention of handover. In the final stage, they complete the physical transfer of the object. The transfer stage can be further divided into two sub-stages, before and after the giver and the taker establish joint contact of the object, respectively. Earlier work on object transfer generally assumes that the object remains stationary once joint contact is established and relies on handcrafted controllers [1, 6, 14, 21]. Our work focuses to the final physical transfer stage only. The algorithm learns a controller directly from human feedback. It does not make the stationary assumption and caters for dynamic handover. Object transfer is an instance of the more general problem of cooperative manipulation [3]: it involves two asymmetric agents with limited communication.

Human-human object handover provides the yardstick for handover performance. Understanding how humans perform handover (e.g., [5, 15]) paves the way towards improved robot handover performance.

2.2 Policy Search

Robot skill learning by policy search has been highly successful in recent years [11]. Policy search algorithms learn a skill represented as a probability distribution over parameterized robot controllers, by maximizing the expected reward. To allow robot skills to adapt to different situations, contextual policy search learns a contextual policy that conditions a skill on context variables [8, 9, 19].

To represent robot skills, policy search typically makes use of parametrized controllers, such as *dynamic movement primitives* [16] or *interaction primitives* [2]. The latter is well-suited for human-robot interaction tasks. Our work, on the other hand, exploits domain knowledge to construct a parameterized impedance controller.

To learn robot skills, policy search requires that a reward function be given to measure learning performance. However, handcrafting a good reward function is often difficult. One approach is inverse reinforcement learning (IRL), also called inverse optimal control, which learns a reward function from expert demonstration [22, 24]. Demonstrations by human experts can be difficult or tedious to acquire, in particular, for robot-human object handover. An alternative is to learn directly from human feedback, without human expert demonstration. Daniel et al. use reward feedback from humans to learn manipulation skills for robot hands [10]. Wilson et al. consider learning control policies from trajectory preferences using a Bayesian approach without explicit reward feedback [27]. Jain et al. learn manipulation trajectories from human preferences [17]. Preference-based reinforcement learning algorithms generally do not use absolute reward feedback and rely solely on preference feedback [28]. Our algorithm combines both absolute and preference feedback in a single Bayesian framework to learn a reward function and integrate with policy search for robot skill learning.

3 Learning Dynamic Handover from Human Feedback

3.1 Overview

Assume that a robot and a human have established the joint intention of handover. Our work addresses the physical transfer of an object from the robot to the human. The robot controller $u(\cdot;\omega)$ specifies the control action u_t at the state x_t at time t for $t = 1, 2, \ldots$. The controller $u(\cdot;\omega)$ is parametrized by a set of parameters ω, and the notation makes the dependency on ω explicit. A reward function $R(\omega)$ assigns a real number that measures the performance of the policy $u(\cdot;\omega)$. To handle the dynamics of handover, we introduce a context variable s representing the velocity of the human hand and condition the controller parameters ω on s, giving rise the reward function $R(\omega, s)$. In general, context variables may include other features, such as human preferences and object characteristics as well. A contextual policy $\pi(\omega|s)$ is a probability distribution over parametrized controllers, conditioned on

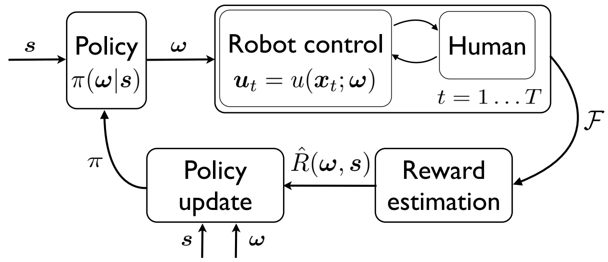

Fig. 2 The human-robot handover skill learning framework. The robot observes context s, then samples ω using the policy $\pi(\omega|s)$. The experiment is executed with a robot controller with parametrization ω. The robot controller $u(x;\omega)$ provides deterministic control signals u given the state of the robot and its environment x. After the experiment the human provides a high-level feedback $\mathscr{F}$, which is used the estimate the latent reward $\hat{R}(\omega,s)$. Finally, the policy is updated with the latest data.

the context s. Our goal is to learn a contextual policy that maximizes the expected reward:

$$\pi^* = \arg\max_{\pi} \int_s \int_{\omega} R(\omega,s)\pi(\omega|s)\mu(s)\, \mathrm{d}\omega\, \mathrm{d}s, \tag{1}$$

where $\mu(s)$ is a given prior distribution over the contexts.

Contextual policy search iteratively updates π so that the distribution peaks up on controllers with higher rewards. In each iteration, the robot learner observes context s and samples a controller with parameter value ω from the distribution $\pi(\cdot|s)$. It executes the controller $u(\cdot|\omega)$ and observes the reward $R(\omega,s)$. After repeating this experiment L times, it updates π with the gathered data $\{\omega_i, s_i, R(\omega_i, s_i)\}_{i=1}^{L}$ and proceeds to the next iteration. See Fig. 2 for the overall learning and control architecture and Table 1 for a sketch of our learning algorithm.

The reward function $R(\omega,s)$ is critical in our algorithm. Unfortunately, it is difficult to specify manually a good reward function for learning object handover, despite the many empirical studies of human-human object handover [4, 5, 15, 26]. We propose to learn a reward function $\hat{R}(\omega,s)$ from human feedback. Specifically, we allow both *absolute* and *preference* human feedback. Absolute feedback provides direct assessment of the robot controller performance on an absolute scale from 1 to 10. Preference feedback compares one controller with another relatively. While the former has higher information content, the latter is usually easier for humans to assess. We take a Bayesian approach and apply Gaussian process regression to latent reward estimation. The learned reward model $\hat{R}(\omega,s)$ generalizes the human feedback data. It provides estimated reward on arbitrarily sampled (ω,s) without additional experiments and drastically reduces the number of robot experiments required for learning a good policy.

The C-REPS Algorithm with Human Feedback

Input: relative entropy bound ε, initial policy $\pi(\omega|s)$, maximum number of policy updates H.
for $h = 1, \ldots, H$
  Collect human feedback data:
    Observe context $s_i \sim \mu(s)$, $i = 1, \ldots, L$
    Draw parameters $\omega_i \sim \pi(\omega|s_i)$
    Collect human feedback $\mathscr{F}_i$
  Estimate latent rewards of all previously seen samples $\{\omega_i, s_i, \mathscr{F}_i\}_{i=1}^{E}$
  Predict rewards:
    Draw context $s_j \sim \hat{\mu}(s), \; j = 1, \ldots, Q$
    Draw parameters $\omega_j \sim \pi(\omega|s_j)$
    Predict $\hat{R}(\omega_j, s_j)$ *with reward model*
  Update policy:
    Update policy $\pi(\omega|s)$ *using C-REPS with samples* $\{\omega_j, s_j, \hat{R}(\omega_j, s_j)\}_{j=1}^{Q}$
  end
Output: policy $\pi(\omega|s)$

Table 1 The learning framework for human-robot object transfer.

3.2 Representing the Object Handover Skill

In this section we discuss how we encode the handover skill and which parameters ω refers to. In our work we use a trajectory generator, a robot arm controller and a robot hand controller to encode the handover skill. A trajectory generator provides reference Cartesian coordinates for the robot end-effector to follow. In robot learning tasks, Movement Primitives (MP) are often used to encode a trajectory with a limited amount of parameters. MPs encode the shape, speed and magnitude of the trajectory in Cartesian space, or in joint space for each degree of freedom. While MPs can encode a wide variety of skills, they typically require a higher number of parameters to tune, which might slow down the learning process.

For handover tasks however, we can use human expert knowledge to define robot hand trajectories. This approach allows for a more compact representation of the trajectory generator with less parameters to tune. Furthermore, we can address safety by reducing the workspace of the robot and we can easily synchronize with the human motion. In our experiments we use visual data of a Kinect sensor, which tracks the right hand of the human. As soon as the human hand is within d_{max} distance from the robot hand the robot moves the object towards the human hand location. We assume that a path planner computes the reference trajectory from the current robot hand location to the human hand location. The reference trajectory is updated every time the human hand location is updated. As soon as the distance between the human and the robot hand falls below d_{min}, we do not use visual information due to possible occlusion and measurement error. Instead, we use the recorded visual data to predict the human hand trajectory for the next second when the physical interaction is likely to happen. The values of d_{min} and d_{max} may depend on different factors, such as, experiment setup, robot configuration, etc.

In order to ensure robust human-robot handover, we need to allow compliant robot arm motion. We use Cartesian impedance control [3] where the wrench $F_{6\times1}$ concatenating forces and torques exerted in the end-effector frame is computed according to $F = M\Delta\ddot{x} + D\Delta\dot{x} + P\Delta x$, where $\Delta x_{6\times1}$ is the deviation from the reference trajectory. The gain parameters M, D and P will determine the amount of exerted forces and torques. M is typically replaced with the robot inertia at the current state. We choose the damping D such that the closed loop control system is critically damped. We use a diagonal stiffness matrix $P = \mathrm{diag}([p_t^T, p_r^T])$, where p_t is the translational and p_r is the rotational stiffness. Finally, the applied torque commands are $\tau = J^T F + \tau_{ff}$, where J is the Jacobian of the robot and τ_{ff} are feed forward torques compensating for gravity and other nonlinear effects.

Motivated by recent work in human-human handover experiments [5], a robot grip force controller [6] has been proposed $F_g = kF_l + F_{ovl}$, where F_g is the commanded grip force, F_l is the measured load force and F_{ovl} is the preset overloading force. The slope parameter k depends on object properties, such as mass, shape and material properties. When using this controller, the robot will release the object in case the total load force on the robot drops below a threshold value. For robot hands with only finger position control we cannot use the above control approach. Instead, we directly command finger positions by identifying the finger position with minimal grip force that still holds the object. Then, we use a control law to change finger positions linear in the load force $f_{pos} = f_{min} + mF_l$. The value of m depends on many factors, such as, object type, weight and other material properties.

For learning the object handover, we tune 7 parameters of the control architecture. For trajectory generator we tune the minimal and maximal tracking distances d_{min} and d_{max}. For the compliant arm controller we learn the translational stiffness parameters and one parameter for all the rotational stiffness values. Finally, for finger controller we tune the slope parameter. All these parameters are collected in $\omega_{7\times1}$.

3.3 Estimating the Latent Reward Function

In this section we propose a Bayesian latent reward estimation technique based on previous work [7]. Assume that we have observed a set of samples $\{s_i, \omega_i\}_{i=1}^E$ and human feedback $\{\mathscr{F}_i\}_{i=1}^E$, where $\mathscr{F}_i = \tilde{R}(y)$, in case the human gives an absolute evaluation (denoted by $\tilde{R}$) on parametrization ω_i in context s_i, $y = [s^T, \omega^T]^T$. In case of preference feedback $\mathscr{F}_i = y_k \succ y_{i\neq k}$ if y_i is preferred over y_i. Note that for a given sample there may exist both preference and absolute evaluation.

We define the prior distribution over the latent rewards as a Gaussian Process [23], $\hat{R} \sim \mathscr{N}(0, K)$, with $K_{ij} = k(y_i, y_j)$. Without the loss of generality we assume 0 prior mean, but more informative priors can be constructed with expert knowledge. The likelihood function for preference based feedback is given by $p(y_i \succ y_j | \hat{R}) = \Phi((\hat{R}_i - \hat{R}_j)/(\sqrt{2}\sigma_p))$ [7], where $\Phi(\cdot)$ is the c.d.f. of $\mathscr{N}(0,1)$ and σ_p is a noise term accounting for feedback noise. For absolute feedback data we simply define the likelihood by $p(\tilde{R}_i|\hat{R}) = \mathcal{N}(\hat{R}_i, \sigma_r^2)$, where σ_r^2 represents the variance of absolute human feedback. Finally, the posterior distribution of the latent rewards can be approximated by,

$$p(\hat{R}|\mathscr{D}) \propto \prod_{i=1}^{N} p(y_{i,1} \succ y_{i,2}|\hat{R}) \prod_{j=1}^{M} p(\tilde{R}_j|\hat{R}_j, \sigma_r^2) p(\hat{R}|0, K), \tag{2}$$

where we used the notation $p(y_{i,1} \succ y_{i,2}|\hat{R})$ to highlight that $\mathscr{F}_i$ is a preference feedback comparing $y_{i,1}$ to $y_{i,2}$. For finding the optimal latent rewards, we minimize

$$J(\hat{R}) = -\sum_{i=1}^{N} \log \Phi(z_i) + \frac{\sigma_r^{-2}}{2} \sum_{j=1}^{M} (\tilde{R}_j - \hat{R}_j)^2 + \hat{R}^T K^{-1} \hat{R}, \tag{3}$$

with $z_i = (\hat{R}(y_{i,1}) - \hat{R}(y_{i,2}))/(\sqrt{2}\sigma_p)$. It was shown in [7] that minimizing J w.r.t. $\hat{R}$ is a convex problem in case there is only preference based feedback ($M = 0$). However, it easy to see that the Hessian of $J(\hat{R})$ will only be augmented with non-negative elements in the diagonal in case $M > 0$, which will leave the Hessian positive semi-definite and the problem convex. Optimizing the hyper-parameters of the kernel function θ and the noise terms can be evaluated by maximizing the evidence $p(\mathscr{D}|\theta, \sigma_p, \sigma_r)$. While the evidence cannot be given in a closed form, we can estimate it by Laplace approximation.

It is interesting to note that in case there is only preference feedback, that is, $M = 0,\ N > 0$, we obtain the exact same algorithm as in [7]. In the other extreme, in case there is only absolute feedback ($M > 0,\ N = 0$) we get Gaussian Process regression, which provides a closed form solution for $p(\hat{R})$. Overall, our extension provides an opportunity to mix preference and absolute feedback in a unified Bayesian framework.

Also note that after obtaining $p(\hat{R})$ we can use Bayesian linear regression to query the expected reward R^* of unseen samples y^* [7, 23]. We can use the resulting generative model of the reward to query the reward for a large number of samples from the current control distribution $y \sim \mu(s)\pi(\omega|s)$, without the need for real experimental evaluation. Such a data-efficient model-based approach has been demonstrated to reduce the required number of experiments up to two orders of magnitude [19, 10].

3.4 Contextual Relative Entropy Policy Search

To update the policy $\pi(\omega|s)$, we rely on the contextual extension of Relative Entropy Policy Search [19, 11], or C-REPS. The intuition of C-REPS is to maximize the expected reward over the joint context-control parameter distribution, while staying close to the observed data to balance out exploration and experience loss. C-REPS uses an information theoretic approach, where the relative entropy between con-

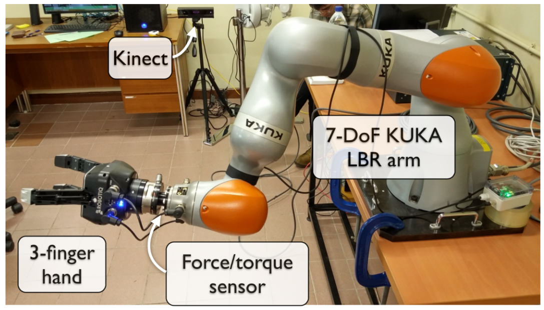

Fig. 3 Robot setup for experiments. We use the 7-DoF KUKA LBR arb with the 3-finger Robotiq robot hand. We use Kinect to track the human hand motion.

secutive parameter distributions is bounded $\int_{s,\omega} p(s,\omega) \log \frac{p(s,\omega)}{q(s,\omega)} ds d\omega \leq \varepsilon$, where $p(s,\omega)$ and $q(s,\omega)$ represent the updated and the previously used context-parameter distributions. The parameter $\varepsilon \in \mathbb{R}^+$ is the upper bound of the relative entropy. The emerging constrained optimization problem can be solved by the Lagrange multiplier method (see e.g. [18]). The closed form solution for the new distribution is given by $p(s,\omega) \propto q(s,\omega) \exp\left((R(\omega,s) - V(s))/\eta\right)$. Here, $V(s)$ is a context dependent baseline, while η and θ are Lagrangian parameters. The baseline is linear in some context features and it is parametrized by θ. To update the policy we use the computed probabilities $p(s,\omega)$ as sample weights and perform a maximum likelihood estimation of the policy model parameters.

4 Experiments

For the handover experiment we use the 7-DoF KUKA LBR arm (Figure 3). For the robot hand we use the Robotiq 3-finger hand. The fingers are position controlled, but the maximum grip force can be indirectly adjusted by limiting the finger currents. In order for accurate measurement of external forces and torques, a wrist mounted force/torque sensor is installed.

4.1 Experimental Setup

An experiment is executed as follows. First, a 1.5l water bottle is placed at a fixed location, which the robot is programmed to pick up. Subsequently, the robot moves the bottle to a predefined position. At this point we enable compliant arm control and we use a Kinect sensor (Figure 3) to track the hand of the human. Subsequently, the human moves towards the robot to take the bottle. While approaching the robot, we use the Kinect data to estimate the hand velocity s of the human, which we assume to be constant during the experiment. We only use data when the human is relatively far (above 1m) from the robot to avoid occlusion. After the context variable is estimated the robot sets its parameter by drawing a controller parametrization $\omega \sim \pi(\omega|s)$. Subsequently, the robot and the human make physical contact and the handover takes place. Finally, the human evaluates the robot performance (preference or absolute evaluation on a 1-10 scale, where 1 is worst 10 is best) and walks away such that the next experiment may begin.

We presented the pseudo code of our learning algorithm in Table 1. As input to the algorithm we have to provide the initial policy $\pi(\omega|s)$, and several other parameters. We use a Gaussian distribution to represent the policy $\pi(\omega|s) = \mathcal{N}(\omega|a + As, \Sigma)$. In the beginning of the learning we set $A = 0$, that is, the robot uses the same controller distribution over all possible context values. During learning all the parameters (a, A, Σ) of the policy will be tuned according to the C-REPS update rule.

The initial policy mean a and the diagonal elements of the covariance matrix Σ are set as follows. For the rotational stiffness we set 2.75 Nm/rad mean and 0.5^2 variance. For the translational stiffness parameters we chose 275, 450, 275 N/m in x, y, and z direction in the hand frame (Fig 4). The variances are $50^2, 75^2$, and 50^2 respectively. For the finger control slope parameter we chose 2.5 1/N with a variance of 0.5^2. This provides a firm grip of the water bottle. The robot will not move the fingers until the force generated by the human hand reaches half the weight of the bottle. With a slope parameter of 0 the robot exerts a minimal grip force that can still support the bottle. With a slope value above 5 the robot only releases the bottle if the human can support $1.2\times$ the object weight. Thus, we can avoid dropping the object, even with the initial policy. Finally as mean we set 200mm and 600mm as minimal and maximal trajectory tracking control distance. As variances we chose 50^2 and 150^2. The parameters are therefore initialized as $a = (2.75,\ 275,\ 450,\ 275,\ 2.5,\ 200,\ 600)^T$, $A = 0$ and $\Sigma = \mathrm{diag}(0.5^2,\ 50^2,\ 75^2,\ 50^2,\ 0.5^2,\ 50^2,\ 150^2)$.

Fig. 4 The robot hand frame orientation.

For the C-REPS learning algorithm in Table 1 we chose $\varepsilon = 0.75$ and we updated the policy after evaluating $L = 10$ human-robot handover experiments. However, before the first policy update we used $L = 40$ handover experiments, such that we have a reliable estimation of the latent rewards. Before each policy update we estimate the latent rewards for all the previously seen experiments $\{\omega_i, s_i, \mathscr{F}_i\}_{i=1}^{E}$. Here, E represents the total number of observed samples. Note, that E is increased by the amount of latest experiments L before each policy update. Therefore, E represents how much experimental evaluation, or information we used to reach a certain level of performance. After estimating the latent rewards we use the resulting generative reward model to evaluate $Q = 500$ *artificial* context-control parameter pairs drawn from $\hat{\mu}(s)\pi(\omega|s)$. We used these artificial samples to update the policy. This way we got a highly data-efficient algorithm, similar to the one in [19]. After the policy is updated, we start a new loop and evaluate L new experiment. We not only use this information to update our dictionary to estimate latent rewards, but also to estimate the performance of the current policy. The performance of the policy is measured by the expected latent reward of the newly evaluated L experiments. We expect the performance measure to increase with the amount of information E and policy updates. After updating the policy H times (Table 1) we terminate the learning.

4.2 Results

As the learning algorithm uses randomly sampled data for policy updates and noisy human feedback, the learned policy and its performance may vary. In order to measure the consistency of the learning progress we repeated the complete learning trial several times. A trial means evaluating the learning algorithm starting with the initial policy and with an empty dictionary, $E = 0$, but using the same parameters for L and ε. We evaluated 5 learning trials and recorded the expected performance of the robot before each policy update. The expected learning performance over 5 trials with 95% confidence bounds against the amount of real robot experiments E used for policy update is shown in Figure 5. We can see that learning indeed improved the performance of the initial policy, which has an expected value of 6.8. Over the learning trials, we noticed that the human mostly gave absolute feedback for very good or bad solutions. This is expected as humans can confidently say if a handover skill feels close to that of a human, or if it does something unnatural (e.g., not releasing the object). By the end of the learning, the expected latent reward rose to the region of 8. Note, that the variance of the learning performance over different trials not only depends on the stochastic learning approach, but also on noisy human feedback. Thus we can conclude that the learning indeed improved the expected latent reward of the policy, but how did the policy and the experiments change with the learning?

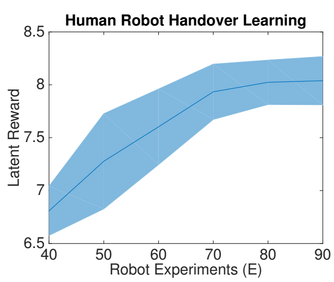

Fig. 5 The expected latent reward mean and standard deviation over 5 independent learning trials.

The learned policy. We first discuss the mean value a of the learned policy and then we show how the policy generalizes to more dynamic tasks. Over several learning trials we observed that a high quality policy provides a lower rotational stiffness compared to the hand-tuned initial policy. We observed that on expectation the learned rotational stiffness is 1.29 Nm/rad, which is lower than the initial 2.75. This helped the robot to quickly orient the object with the human hand upon physical contact. We observed similar behavior in the translational stiffness values in the $x-z$ directions

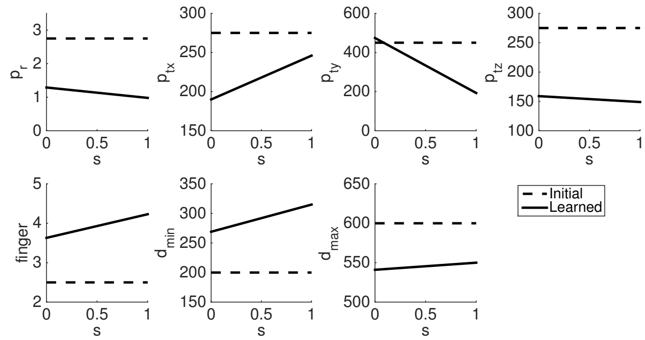

Fig. 6 The initial and the learned policy parameters against the context value. Top row, from left to right: the rotational stiffness, translational stiffness in the x-y-z direction. Bottom row, from left to right: finger control slope, minimal and maximal visual hand tracking distance.

(see Figure 4). The learned values were almost 100 N/m lower compared to the initial values. This helps the robot to become more compliant in horizontal motions. Interestingly, the learned stiffness in y direction became slightly higher (474 N/m) compared to its initial value. During physical interaction the forces acting along the y-axis are mostly responsible for supporting the weight of the object. With a higher stiffness value interaction times became lower and also helped avoiding situations where the robot did not release the object. The learned slope parameter of the finger controller became more conservative (3.63 1/N). This prevented any finger movement until the human force reached at least $0.8\times$ the weight of the object. Finally, the learned minimal and maximal tracking distance on expectation became 269 and 541mm respectively.

The policy generalizes the controller parametrization with mean $a+As$. We discussed above how a changed on expectation after the learning. We now turn our attention to A and show how generalization to more dynamic task happens. We typically executed experiments with hand speed between 0.1 and 1m/s. We observed that on expectation the rotational stiffness values were lowered for more dynamical tasks ($s = 1$m/s) with -0.31 Nm/rad. This helped the robot to orient with the human hand quicker. Interestingly, we observed that the stiffness in x direction is slightly increased with 56 N/m. However, the stiffness in y direction is dramatically decreased with -281 N/m. This reduces forces acting on the human significantly during faster physical interaction. The stiffness in z direction is decreased with -10 N/m, which is just a minor change. Interestingly, the slope parameter of the robot finger controller increases with 0.6 1/N, which leads to an even more conservative finger control. Finally, we observed that on expectation the minimal hand tracking distance is increased by 46mm and the maximal distance remains almost the same with an additional 9mm. A visual representation of the learned parameters against the context value is shown in Figure 6. In the following, we will analyze some static and dynamic handover experiments to give more insight why humans prefer the learned policy as opposed to the initial hand-tuned controller.

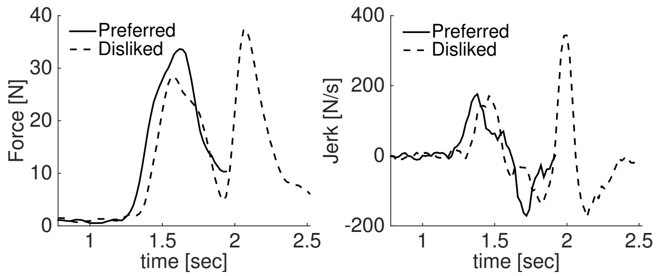

Fig. 7 Two example of experimental results of the forces acting between the human and the robot during physical interaction. The forces are plotted starting right before the physical interaction until the handover is finished.

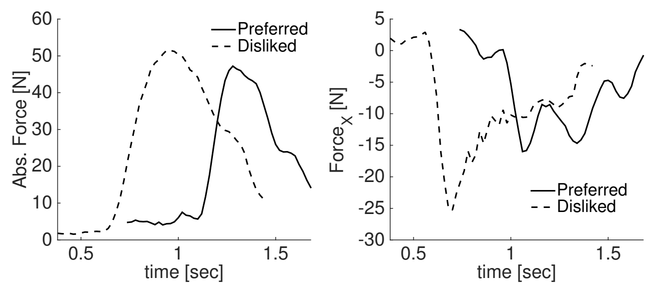

Fig. 8 Two example of experimental results in dynamic handover situations. The forces are plotted starting right before the physical interaction until the handover is finished.

Human preferences for static handovers. For static handover tasks we observed that a robust and quick finger control was always preferred and highly rated. In Figure 7 we can see the forces and jerks of two typical static handover solutions. The weight of the bottle was around 20N. We can see that the preferred solution always maintained a low jerk and forces remained limited. Moreover, a successful handover happens relatively fast. In our experiments we observed that a high quality solution happens within 0.6 seconds and no faster than 0.4 seconds. Similar results have been reported in human-human object transfers experiments [5]. Typically disliked parameterizations have low translational stiffness and a stiff finger control, resulting in the robot not releasing the object quick enough, which is considered a failure. These experiments typically lasted for 1 to 2 seconds until the bottle was released.

Human preferences for dynamic handovers. In dynamic handover situations contact forces and jerks were significantly higher compared to the static case (Figure 8). A typical preferred dynamic handover controller has lower rotational and translational stiffness, and a more firm finger controller. In our experiments the human always came from one direction while taking the bottle from the robot. In the robot hand frame this was the x-direction. As we can see, a preferred controller achieves a significantly lower contact force and jerk in this direction. We noticed that a physical contact time in a dynamic handover scenario is around $0.3-0.6$ sec. Based on the latent rewards, we noticed that there is a strong preference towards faster handovers, as opposed to the static case, where we did not observe such strong correlation in handovers within 0.6 seconds. Interestingly, we noticed that humans preferred stiffer finger controllers in dynamic handovers. We assume that this helps a robust transfer of the object from giver to taker. In a dynamic handover situation vision might not provide enough feedback about the handover situation during physical contact, and thus, an excess of grip force would be necessary to ensure the robust transfer and to compensate for inaccurate position control.

Video footage of some typical experiments before and after the learning is available at www.youtube.com/watch?v=2OAnyfph3bQ.

By analyzing these experiments we can see that the learned policy indeed provides a controller parametrization that decreases handover time, reduces forces and jerks acting on the human over a wide variety of dynamic situations. While the initial policy provides a reasonable performance in less dynamic experiments, learning and generalization significantly improves the performance of the policy. Based on our observations, for static handovers a fast and smooth finger control was necessary for success, while in dynamic handover situation higher compliance and a firm finger control were preferred.

5 Discussion

This paper presents an algorithm for learning dynamic robot-to-human object handover from human feedback. The algorithm learns a latent reward function from both absolute and preference feedback, and integrates reward learning with contextual policy search. Experiments show that the robot adapts to the dynamics of human motion and learns to hand over a water bottle successfully, even in highly dynamic situations.

The current work has several limitations. First, it is evaluated on a single object and a small number of people. We plan to generalize the learning algorithm to adapt over human preferences and object characteristics. While contextual policy search works well for adapting over handover dynamics, object characteristics exhibit much greater variability and may pose greater challenge. Second, our handover policy also does not consider human response during the handover or its change over time. We want to model key features of human response and exploit it for effective and fluid handover. For both, combining model-free learning and model-based planning seems a fruitful direction for exploration.

Acknowledgements. This research was supported in part an A*STAR Industrial Robotics Program grant (R-252-506-001-305) and a SMART Phase-2 Pilot grant (R-252-000-571-592).

References

[1] A. Agah and K. Tanie. Human interaction with a service robot: mobile-manipulator handing over an object to a human. In *Proc. IEEE Int. Conf. on Robotics & Automation*, 1997.

[2] H. Ben Amor, G. Neumann, S. Kamthe, O. Kroemer, and J. Peters. Interaction primitives for human-robot cooperation tasks. In *Proc. IEEE Int. Conf. on Robotics & Automation*, 2014.

[3] S. Bruno and O. Khatib, editors. *Handbook of Robotics*. Springer, 2008.

[4] M. Cakmak, S. Srinivasa, M. Lee, J. Forlizzi, and S. Kiesler. Human preferences for robot-human hand-over configurations. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2011.

[5] W. Chan, C. Parker, H. Van der Loos, and E. Croft. A human-inspired object handover controller. *Int. J. Robotics Research*, 32(8):971–983, 2013.

[6] W. P. Chan, I. Kumagai, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba. Implementation of a robot-human object handover controller on a compliant underactuated hand using joint position error measurements for grip force and load force estimations. In *Proc. IEEE Int. Conf. on Robotics & Automation*, 2014.

[7] W. Chu and Z. Ghahramani. Preference learning with Gaussian processes. In *Proc. Int. Conf. on Machine Learning*, 2005.

[8] B. da Silva, G. Konidaris, and A. Barto. Learning parameterized skills. In *Proc. Int. Conf. on Machine Learning*, 2012.

[9] C. Daniel, G. Neumann, and J. Peters. Hierarchical relative entropy policy search. In *AISTATS*, 2012.

[10] C. Daniel, M. Viering, J. Metz, O. Kroemer, and J. Peters. Active reward learning. In *Proc. Robotics: Science & Systems*, 2014.

[11] M. P. Deisenroth, G. Neumann, and J. Peters. A survey on policy search for robotics. *Foundations and Trends in Robotics*, 2(1-2):1–142, 2013.

[12] A. Dragan and S. Srinivasa. Generating legible motion. In *Proc. Robotics: Science & Systems*, 2013.

[13] E. C. Grigore, K. Eder, A. G. Pipe, C. Melhuish, and U. Leonards. Joint action understanding improves robot-to-human object handover. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 4622–4629. IEEE, 2013.

[14] C.-M. Huang, M. Cakmak, and B. Mutlu. Adaptive coordination strategies for human-robot handovers. In *Proc. Robotics: Science & Systems*, 2015.

[15] M. Huber, A. Kupferberg, C. Lenz, A. Knoll, T. Brandt, and S. Glasauer. Spatiotemporal movement planning and rapid adaptation for manual interaction. *PLoS One*, 2013.

[16] A. J. Ijspeert and S. Schaal. Learning Attractor Landscapes for Learning Motor Primitives. In *Advances in Neural Information Processing Systems*. 2003.

[17] A. Jain, B. Wojcik, T. Joachims, and A. Saxena. Learning trajectory preferences for manipulators via iterative improvement. In *Advances in Neural Information Processing Systems*. 2013.

[18] A. Kupcsik, M. Deisenroth, J. Peters, L. Ai Poh, V. Vadakkepat, and G. Neumann. Model-based contextual policy search for data-efficient generalization of robot skills. *Artificial Intelligence*. to appear.

[19] A. Kupcsik, M. P. Deisenroth, J. Peters, and G. Neumann. Data-efficient contextual policy search for robot movement skills. In *Proc. AAAI Conf. on Artificial Intelligence*, 2013.

[20] J. Mainprice, M. Gharbi, T. Siméon, and R. Alami. Sharing effort in planning human-robot handover tasks. In *Proc. Int. Symp. on Robot & Human Interactive Communication*, 2012.

[21] K. Nagata, Y. Oosaki, M. Kakikura, and H. Tsukune. Delivery by hand between human and robot based on fingertip force-torque information. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 1998.

[22] A. Ng and S. Russell. Algorithms for inverse reinforcement learning. In *Proc. Int. Conf. on Machine Learning*, 2000.

[23] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

[24] N. Ratliff, D. Silver, and J. Bagnell. Learning to search: Functional gradient techniques for imitation learning. *Autonomous Robots*, 27(1):25–53, 2009.

[25] E. Sisbot, R. Alami, T. Siméon, K. Dautenhahn, M. Walters, and S. Woods. Navigation in the presence of humans. In *Proc. IEEE-RAS Int. Conf. on Humanoid Robots*, 2005.

[26] K. Strabala, M. K. Lee, A. Dragan, J. Forlizzi, S. Srinivasa, M. Cakmak, and V. Micelli. Towards seamless human-robot handovers. *J. Human-Robot Interaction*, 2013.

[27] A. Wilson, A. Fern, and P. Tadepalli. A bayesian approach for policy learning from trajectory preference queries. In *Advances in Neural Information Processing Systems*. 2012.

[28] C. Wirth and J. Fürnkranz. Preference-based reinforcement learning: A preliminary survey. In J. Fürnkranz and E. Hüllermeier, editors, *Proc. ECML/PKDD Workshop on Reinforcement Learning from Generalized Feedback: Beyond Numeric Rewards*, 2013.