Learning a State Estimator for Tactile In-Hand Manipulation

Lennart Röstel, Leon Sievers, Johannes Pitz, Berthold Bäuml

Learning a State Estimator for Tactile In-Hand Manipulation

Lennart Röstel, Leon Sievers, Johannes Pitz and Berthold Bäuml

***Abstract*— We study the problem of estimating the pose of an object which is being manipulated by a multi-fingered robotic hand by only using proprioceptive feedback. To address this challenging problem, we propose a novel variant of differentiable particle filters, which combines two key extensions. First, our learned proposal distribution incorporates recent measurements in a way that mitigates weight degeneracy. Second, the particle update works on non-euclidean manifolds like Lie-groups, enabling learning-based pose estimation in 3D on SE(3). We show that the method can represent the rich and often multi-modal distributions over poses that arise in tactile state estimation. The models are trained in simulation, but by using domain randomization, we obtain state estimators that can be employed for pose estimation on a real robotic hand (equipped with joint torque sensors). Moreover, the estimator runs fast, allowing for online usage with update rates of more than 100 Hz on a single CPU core. We quantitatively evaluate our method and benchmark it against other approaches in simulation. We also show qualitative experiments on the real torque-controlled DLR-Hand II.**

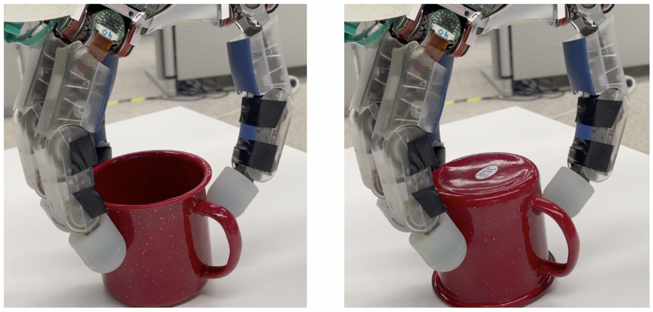

Fig. 1. Purely tactile in-hand pose estimation with the DLR-Hand II [2], where contacts with the fingers are detected based on the hand's torque sensors (see Section IV for details). Many everyday objects, like the mug shown here, exhibit *apparent symmetries*; the same tactile measurement can be compatible with multiple poses, often infinitely many. A state estimator addressing this problem should be able to represent the rich, multimodal belief distributions that arise in these settings.

I. Introduction

Humans are able to locate and manipulate objects only by proprioceptive and haptic feedback. When we manipulate an object in our hands, we don't require constant visual feedback; our hands can even be out of sight completely. In contrast, manipulation in the context of robotics today is often driven by the availability of vision. For example, in their seminal work on fine manipulation, OpenAI [1] train a dedicated model to predict the pose of a Rubic's Cube, where the training data is generated by renderings in simulation. For the transfer to the real-world setup, a rig with three calibrated cameras is required, capturing the scene from multiple angles. However, in many situations encountered in the real world, obtaining visual information may be impracticable. Inspired by this, we explore techniques that enable robotic systems to perform manipulation by only utilizing tactile feedback. More specifically, we train a system that estimates the 3D pose of an object only based on joint measurements (i.e., the configuration of the fingers) and contact information (here via joint torque sensors). Fig. 1 shows a real-world experiment using the DLR-Hand II [2]. Performing purely tactile state estimation brings unique challenges:

- Any single tactile measurement rarely uniquely determines the pose of an object. A key requirement is therefore to model the resulting distributions in the space of object poses. We observe that when manipulating everyday objects in 3D, highly non-Gaussian, often multimodal distributions arise (see Fig. 2).
- The dynamics induced by the contact-based interactions between the fingers and an object are highly non-linear.

We address these challenges by employing a variant of differentiable particle filters (DPF) [3, 4], which we train from data generated in simulation, where ground truth states are available. We show that the obtained algorithm can capture the pose distributions that emerge in the context of tactile state estimation. Our key contributions are:

- Extending existing works on DPFs [3, 4], we present a method for learning more informed proposal distributions that incorporate recent measurements into the propagation of particles while mitigating weight degeneracy.
- We introduce principled update rules for DPFs on non-euclidean manifolds like Lie-groups, enabling learning-based pose estimation in 3D.
- We apply our proposed method to the challenging task of localizing a known object while using only proprioceptive feedback from a multi-fingered humanoid hand. We show that the learned proposal distributions lead to more informed particle updates, improving reliability and performance over alternative approaches for tasks with non-Gaussian, multimodal belief distributions.
- We show that the models, learned entirely in simulation, can be transferred to real robotic in-hand manipulation tasks and that the filter runs online at 100 Hz with 100 particles on a single CPU core.

Related Work

Many previous methods addressing the problem of tactile localization rely on Bayesian filters that are specifically

The authors are with the DLR Institute of Robotics and Mechatronics and with the Deggendorf Institute of Technology (DIT).
Contact: lennart.roestel | berthold.baeuml@dlr.de

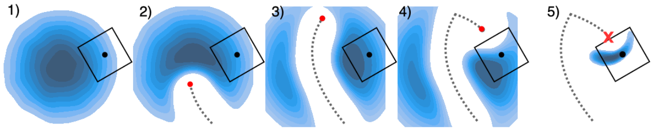

Fig. 2. Illustration of a simplified tactile localization setting. The position of the black rectangle is to be estimated. The ground truth position (black) is not directly observable. The current probability density (belief) for the center of the square is depicted in blue. **1)** Initial estimate of the object pose. **2)** During a first sweep with a touch-sensitive probe (red), no contact is observed. This leads to a transformation of the initial distribution by elimination of hypothesis. **3,4)** After further movements with no contact measurements, the resulting distribution becomes bi-modal and highly non-Gaussian. **5)** The probe hits the target, producing a contact measurement. The resulting distribution in pose space collapses to a submanifold, a so-called *contact manifold* (see Koval et al. [5]).

designed to deal with the challenges of tactile state estimation. For example, Pfanne and Chalon [6] introduce an approach for tracking the pose of a grasped object based on the Extended Kalman Filter (EKF). However, when the initial uncertainty is large or if there are ambiguities due to apparent object symmetries (Fig. 1), belief distributions are, in general, non-Gaussian or even multi-modal, violating the assumption made by the EKF. In these situations, particle filters have been employed [7, 8, 5], enabling the approximation of, in principle, arbitrary belief distributions. Vezzani et al. [8] study pose estimation of static objects with known geometry by obtaining the cartesian position of contact points from capacitive tactile hardware. Our approach can be used in a manner that is agnostic to the exact location of contacts; we perform state estimation only from measured joint angles and control input and also consider dynamic objects. Wirnshofer et al. [7] propose a simulation-based particle filter for pose estimation in contact-rich environments with compliantly controlled robots. Unfortunately, the approach is computationally demanding, requiring a contact-based physics simulation for each particle instance. Conversely, in our learning-based approach, we use a rigid-body simulation only to generate training data a priori, but once employed, the obtained state estimator is computationally cheap, enabling fast online pose estimation with limited compute resources.

A limitation of particle filters that propagate particles by the dynamic model is *particle starvation*; when in any update step the measurement likelihood function is high in a region of the state space in which there are only few particles, the resulting posterior will be poorly approximated. Koval et al. [5] address this issue by sampling particles from so-called *contact manifolds* (see Fig. 2) which are represented as a set of precomputed poses. We address the problem of particle starvation through a learned generative proposal distribution that moves particles towards regions of high likelihood in the state space.

The previously discussed methods are Bayesian filters with carefully handcrafted observation models and motion priors. However, designing these models is challenging in the general case of a multi-fingered hand interacting with arbitrary objects. Differentiable Bayesian filters [9, 10, 3] are a class of learning-based methods which enable data-driven optimization of models while maintaining explainability and structured ways of dealing with uncertainty as provided by Bayesian Filters. Previous works on differentiable particle filters primarily studied visual localization tasks with high-dimensional visual input but often assume simple or known dynamics [3, 4, 11]. In contrast, in case of the tactile state estimation tasks studied here, inputs are lower-dimensional, but the contact-induced dynamics are highly non-linear in a comparably high-dimensional state space. In this work, to our knowledge for the first time, we show how differentiable particle filters can be successfully applied in the regime of in-hand manipulation with a multi-fingered hand, including principled state estimation for non-euclidean manifolds in 3D.

II. Background on Filtering

A. Bayesian Filtering

Given an initial prior distribution $p(x_0)$ as well as observations z_t and control inputs u_t at discrete time steps $t = 1, ..., T$, the goal of filtering is to estimate the posterior or *belief* distribution $\mathcal{B}(x_t) = p(x_t | x_{0:t-1}, z_{1:t}, u_{1:t})$ over the state $x_t \in \mathcal{S}$. Under the assumption that the Markov property holds for the state x_t (i.a. $p(z_t | x_t, z_{1:t-1}) = p(z_t | x_t)$), at any given time step t, this posterior distribution summarizes the information about the history of observations $z_{1:t} = z_1, z_2, ..., z_t$ and control inputs $u_{1:t}$ and is, therefore, a sufficient statistic for the trajectory up to t. Bayes Filters recursively estimate $\mathcal{B}(x_t)$ by combining incoming z_t and u_t with the previous posterior estimate $\mathcal{B}(x_{t-1})$.

B. Particle Filters

Particle filters [12] approximate the belief as a finite set of tuples $\langle x^{(i)}, w^{(i)} \rangle$ where each particle $i = 1, ..., N$ is a weighted sample comprising a state $x^{(i)} \in \mathcal{S}$ and weight $w^{(i)} \in [0, 1]$. Formally, the belief can be written as

$$\mathcal{B}(x_t) \approx \sum_{i=1}^{N} w^{(i)} \delta(x_t - x_t^{(i)}), \tag{1}$$

where δ is the Dirac delta function. $\mathcal{B}(x_t)$ is normalized if $\sum_i w^{(i)} = 1$.

In general, sampling $\mathcal{B}(x_t)$ to obtain samples $x^{(i)}$ is not possible because $\mathcal{B}(x_t)$ is unknown. However, using a known distribution q, we can obtain samples $x^{(i)}$ by importance sampling. In general, q may depend on the full history of observations. Here we consider drawing samples from q of the form

$$x_t^{(i)} \sim q(\cdot | x_{t-1}^{(i)}, z_t, u_t), \tag{2}$$

and compute the importance weights $w_t^{(i)}$ as

$$w_t^{(i)} = w_{t-1}^{(i)} \frac{p(z_t | x_t^{(i)}) p(x_t^{(i)} | x_{t-1}^{(i)}, u_t)}{q(x_t^{(i)} | x_{t-1}^{(i)}, z_t, u_t)}. \tag{3}$$

The conditional distribution $p(z_t | x_t)$ is often referred to as the *observation* or *measurement* model. The distribution $p(x_t | x_{t-1}, u_t)$ predicts the next state given the current state and the control input and is usually referred to as the *motion model*. The recursion of (2) and (3), and the normalization of weights results in the particle filter algorithm. Often an additional *resampling* step is included where particles are duplicated in proportion to their weight and particles with smaller weights are dropped. In (2), in principle, any tractable proposal distribution q can be used for importance sampling. A popular choice is to sample from the motion model, i.e., $q(x_t^{(i)}|x_{t-1}^{(i)}, z_t, u_t) = p(x_t^{(i)}|x_{t-1}^{(i)}, u_t)$. We will refer to this proposal as the *standard* proposal [13].

A major downside of the *standard* proposal is that it is agnostic to observations in the sampling of new particles, resulting in the problem of *particle starvation* in the context of tactile pose estimation [5]. Particle starvation can be mitigated by conditioning the proposal on the most recent observation z_t, allowing particles to be moved towards regions of high observation likelihood in the state space [14, 8]. However, scoring samples drawn from *any* q (compare [15]) and computing importance weights as in (3) can lead to highly noisy weight updates, preventing the emergence of expressive particle distributions. In these regards, a proposal distribution incorporating measurements with particularly nice properties is

$$q(x_t^{(i)}|x_{t-1}^{(i)}, z_t, u_t) = p(x_t^{(i)}|x_{t-1}^{(i)}, z_t, u_t). \tag{4}$$

This proposal is often referred to as the "optimal" proposal[1] [16, 13], where "optimal" refers to the fact that the update step results in minimal variance of the weights for a sample $x_t^{(i)}$ given z_t, u_t and $x_{t-1}^{(i)}$ [16]; the weight update for the "optimal" proposal becomes

$$w_t^{(i)} \propto w_{t-1}^{(i)} p(z_t|x_{t-1}^{(i)}, u_t). \tag{5}$$

Sampling from $p(x_t|x_{t-1}, z_t, u_t)$ and evaluating (5) is, however, notoriously difficult and often only possible for cases where an analytical expression for the "optimal" proposal exists [16]. A core element of the approach presented in this paper is to learn approximations of the "optimal" proposal from data.

C. Differentiable Particle Filters

Differentiable particle filters [3, 4] combine the algorithmic structure of particle filters with neural-network-based function approximation. In their most basic form, DPFs can be parametrized by two learned models: A *generative* proposal distribution F_φ, used for sampling new particles and a *non-generative* update function G_φ which is used to update the weights. The learnable parameters φ can be trained end-to-end to maximize performance through rollouts of the computation graph [3]. Most previous work on DPFs employed the *standard* proposal, i.e. $F_\varphi(x_t|x_{t-1}, u_t) \approx p(x_t|x_{t-1}, u_t)$ and $G_\varphi(x_t, z_t) \propto p(z_t|x_t^{(i)})$ [3, 4, 17, 11]. Jonschkowski et al. [3] additionally learn a dedicated model to propose particles based on observations during an initialization phase, but subsequent particle proposals are done using the *standard* proposal. Chen et al. [15] introduce proposal distributions based on normalizing flows, which may also be conditioned on observations. However, the influence on the variance of weight updates (3) was not studied and the parametrization does not allow for employment on non-euclidean manifolds.

III. Deep Differentiable Proposal Particle Filters (D2P2F)

We present a DPF-variant for state estimation that incorporates recent observations, mitigates weight degeneracy, and naturally allows for deployment on manifolds like SE(3) while maintaining end-to-end differentiability.

A. Learned Models

For our DPF variant, the two learned models are a generative proposal distribution of the form $F_\varphi = F_\varphi(\cdot|x, z, u)$, approximating the "optimal" proposal (4), and a non-generative update model G_φ implementing the weight update (5).

Rather than defining F on the state manifold $\mathcal{S}$ directly, we use samples $\Delta \sim F(\cdot)$ to act locally around a state $x \in \mathcal{S}$. We update the i-th particle as:

$$\Delta_t^{(i)} \sim F_\varphi(\cdot|x_{t-1}^{(i)}, z_t, u_t) \tag{6}$$

$$x_t^{(i)} = x_{t-1}^{(i)} \boxplus \Delta_t^{(i)}, \tag{7}$$

where the $\boxplus$-operator generalizes addition to non-euclidean manifolds as described by Hertzberg et al. [18]: In the simple case that $\mathcal{S} = \mathbb{R}^n$, the $\boxplus$-operator is just the vector addition. If x_t is a rotation matrix, the $\boxplus$-operator has the effect of a rotation around axis Δ by angle $||\Delta||$. The advantage of this method is that F can simply be defined on the vector space, i.e. $\Delta \in \mathbb{R}^n$, yet the updated state is again on the n-manifold ($x_t \in \mathcal{S}$). Hence, any distribution on $\mathbb{R}^n$ that supports the reparametrization trick can be used to represent F. We experimented with parametrizing F as conditional normalizing flows but found no clear advantage over parameterization as a normal distribution, which we use in this work. For sampling from the proposal, for each particle (i), the concatenation of the conditional $(x_{t-1}^{(i)}, z_t, u_t)$ is taken as input to a feedforward network that predicts the parameters of the proposal distribution.

The update model G is implemented as a feedforward neural network,

$$G_\varphi(x_{t-1}, z_t, u_t) : \mathcal{S} \times \mathbb{R}^z \times \mathbb{R}^u \to \mathbb{R} \tag{8}$$

mapping the concatenation of $x_{t-1}^{(i)}$, $z_t \in \mathbb{R}^z$ and $u_t \in \mathbb{R}^u$ to a scalar which can be interpreted as unnormalized log-likelihood. Note that, unlike the update model employed in previous works on DPFs [3, 4], the input is the particle state at the previous time step x_{t-1}, following update rule (5).

We denote the DPF with the above parametrization as Deep Differentiable Proposal Particle Filter (D2P2F) as summarized in Algorithm 1.

[1] In the literature referenced, the definition of the "optimal" proposal commonly is $q = p(x_t^{(i)}|x_{t-1}^{(i)}, z_t)$ because uncontrolled systems are studied. For this paper, we use the more informed variant of the "optimal" proposal $q = p(x_t^{(i)}|x_{t-1}^{(i)}, z_t, u_t)$, which is additionally conditioned on control inputs.

Algorithm 1 Deep Differentiable Proposal Particle Filter

```
Initialize particles: $x_0^{(i)} \sim p(x_0), \quad \forall i = 1, ..., N$
Initialize weights: $\log w_0^{(i)} = -\log(N), \quad \forall i = 1, ..., N$

for $t = 1, 2...$ do
    Obtain: Control input $u_t$, observation $z_t$
    for $i = 1, 2, ..., N$ do
        $\Delta_t^{(i)} \sim F_\varphi(\cdot | x_{t-1}^{(i)}, u_t, z_t)$
        $x_t^{(i)} = x_{t-1}^{(i)} \boxplus \Delta_t^{(i)}$
        $\log \hat{w}_t^{(i)} = \log w_{t-1}^{(i)} + G_\varphi(x_{t-1}^{(i)}, z_t, u_t)$
    end for
    Normalize: $\log w_t^{(i)} \leftarrow \log \hat{w}_t^{(i)} - \mathrm{LSE}_j \log \hat{w}_t^{(j)}$
    $\langle x^{(i)}, w^{(i)} \rangle \leftarrow$ RESAMPLE($\langle x^{(i)}, w^{(i)} \rangle$) (optional)
end for
```

B. Learning Objective and Training

For supervised training of the particle filters, we require that the ground truth target states $\hat{x}_t$ are available. When training data is generated in simulation, this requirement is usually satisfied at no extra cost. We optimize for the learnable parameters with gradient-based methods by differentiating through rollouts of the algorithm over multiple timesteps using automatic differentiation similarly to previous DPF implementations [3, 4]. Gradients through the proposal distribution are calculated using the reparametrization trick. The recursive time stepping scheme of the algorithm amounts to backpropagation through time (BPTT). For our D2P2F, we find that weight degeneracy is less severe (see Section V-B.3). Therefore, we train without resampling, bypassing the issue of the non-differentiable resampling step (compare [3, 4]).

As a learning objective, one could use the distance between $\hat{x}_t$ and the weighted mean of particles [4]. However, we find that training with this objective encourages unimodal particle beliefs at inference, whereas belief distributions that arise in the context of tactile localization are often multimodal. To account for this, we follow Jonschkowski et al. [3] and construct a Gaussian-mixture distribution where each particle constitutes the mean of one component.

We adapt this loss to states on (Lie) manifolds by constructing a zero-mean m-dimensional multivariate normal distribution for each particle (i), where m is the number of manifolds in the state dimension. Then we evaluate the density at $\mathrm{d}(\hat{x}_t, x_t^{(i)}) \in \mathbb{R}^m$, where $\mathrm{d}(\cdot)$ is the distance metric associated with the manifold, leading to the loss function

$$\mathcal{L}_{\mathrm{gm}} = -\frac{1}{T} \prod_{t=1}^{T} \sum_{i=1}^{N} w_t^{(i)} \mathcal{N}(\mathrm{d}(\hat{x}_t, x_t^{(i)}) | 0, \Sigma). \tag{9}$$

On Lie manifolds, evaluating the normal distribution at $\mathrm{d}(\hat{x}_t, x_t^{(i)})$ is justified as long as the covariance matrix $\Sigma \in \mathbb{R}^{m \times m}$ is sufficiently small [18]. This is not a limitation in practice, since by design each particle only contributes to the density in its vicinity. For simplicity, we assume $\Sigma = \mathrm{diag}(\sigma_1, ..., \sigma_m)$ with hyperparameters $\sigma_1, ..., \sigma_m$ that can be adjusted according to the scale of the manifold. We normalize all cartesian dimensions of the state space to zero mean and unit variance and use $\sigma = \exp(-3)$ for all manifolds. $\mathcal{L}_{\mathrm{gm}}$ facilitates the emergence of non-Gaussian belief distributions with multiple modes, which is a critical property for the tracking of multiple hypotheses that arise in the context of tactile localization, as we show in our experiments.

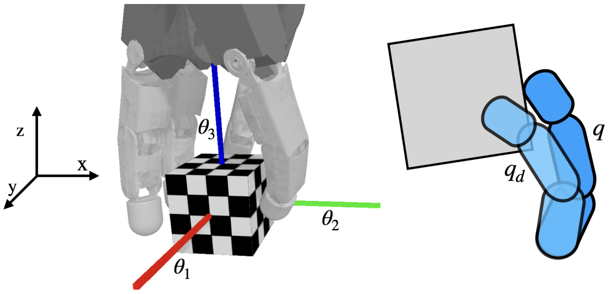

Fig. 3. **Left**: Simulation setup of the cube rotation task. **Right:** The learned model is able to recognize contacts based on the difference between desired joint angles q_d and measured joint angles q.

IV. Application to 3D In-Hand Pose Estimation

We now study applications for pose estimation in 3D using the DLR-Hand II, an anthropomorphic robotic hand [2]. The hand has four identical fingers, each equipped with four joints, three of which are actuated independently. For training in simulation, we use a simulated replica of the hand as described in Sievers et al. [19], where we assume known geometric shapes of the hand and the manipulated object.

The observations consist of the measured joint angles $q \in \mathbb{R}^{12}$ and joint velocities $\dot{q} \in \mathbb{R}^{12}$. The joint measurements are subject to time-invariant Gaussian noise, as well as a systematic error offset which accounts for modeling errors [19]. The hand is torque-controlled with a joint-level impedance control scheme. The control input given to the filter are the desired joint positions $u = q_d \in \mathbb{R}^{12}$ in each time step. In this setting, we can detect contacts indirectly via the high-fidelity torque sensors; the torque τ applied by the controller is proportional to the difference between q_d and q. Therefore, although measured torques are not directly observed by the filter, the presence of contacts can be inferred by observing the discrepancy between q_d and q (see Fig. 3). The full state of the hand-object system is given by the pose of the object in 3D, given as a rigid body transformation $x \in \mathrm{SE}(3)$, the translational and rotational velocities of the object $\dot{x}$, as well as the (ground truth) state of the finger joints $\hat{q}$ and $\dot{\hat{q}}$, i.e.

$$\mathcal{S} = \mathrm{SE}(3) \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^{12} \times \mathbb{R}^{12}. \tag{10}$$

Modeling the intricate dynamics arising from the contact-based interactions between the hand and the object is challenging and, in the general case, requires simulating the interactions using a rigid-body simulator. To account for uncertain system parameters like friction as well as modeling errors, we apply extensive domain randomization in simulation, including randomization of masses, the sizes of objects, and friction coefficients. A more complete description of the setup in PyBullet [20] can be found in Sievers et al. [19].

We study pose estimation in two tasks: grasping a mug with an initially uncertain pose (Section IV-.1) and rotating a cube inside the hand (Section IV-.2). Our primary goal for these tasks is *pose estimation*. Hence, whenever beneficial, we do not actively estimate the full state vector (10). For the learning-based approaches, we empirically find that estimating velocities of objects does not significantly improve (nor degrade) the accuracy of the pose estimate.

1) Grasping a Mug: For this task (see Fig. 1), we aim to estimate the pose of a mug [21] during a predefined grasping motion. The mug is initially placed in an unknown position underneath the fingers of the hand. The initial displacement in the xy-plane is sampled uniformly in $[-3\,\mathrm{cm}, 3\,\mathrm{cm}]$ and the height is randomized in $[-2\,\mathrm{cm}, 2\,\mathrm{cm}]$ relative to the reference frame of the hand. Additionally, the mug is placed upside down by chance ($p = 0.5$) and rotated at random uniformly by $\theta_3 \in [0, 2\pi]$ around the z-axis orthogonal to the surface plane. During the grasping motion, the pose of the mug changes according to the forces exerted by the fingers and gravity. The details of this behavior strongly depend on the initial configuration of the mug, as well as the domain parameters.

2) Rotating a Cube: In this experiment, we estimate the pose of a cube while it is being rotated inside the hand. The policy performing the task is a closed-loop controller that has been obtained by reinforcement learning in simulation by Sievers et al. [19]. The goal of the policy is to rotate the cube around the upwards axis (blue axis in Fig. 3). Observations z_t and control inputs u_t are obtained with a frequency of 100 Hz. We infuse the learned state estimator with knowledge of the rotational symmetry of the cube by mapping a $\pi/2$ rotation in the task space to a full 2π rotation in the state space of the filter. Although we will use this mapping for comparison with baselines, we later show that the DPF-based methods are also able to handle the multimodal beliefs that are induced by non-symmetry-aware representations.

A. Compared Baselines

We compare our approach with multiple applicable baselines, including learning and non-learning-based approaches:

- Our D2P2F as described in Section III, parametrizing the "optimal" proposal $p(x_t|x_{t-1}, z_t, u_t)$, without resampling.
- A DPF parametrizing the *standard* proposal $p(x_t|x_{t-1}, u_t)$ with *soft resampling* [4] whenever the effective sample size [16] falls below the threshold of $N/2$.
- A simulation-based particle filter where the motion model is implemented by a full rigid-body PyBullet [20] simulation instance for each particle (denoted as *sim-PF*). The observation model assigns likelihoods based on the difference between measured joint angles q and joint angles in the i-th simulation instance $q^{(i)}$, as proposed by Wirnshofer et al. [7].
- As a unimodal baseline we implement a model based on recurrent neural networks which parametrizes a normal distribution in state space. Mean and covariance are predicted by a learned decoder from the hidden state output of an LSTM [22]. We use a 6D continuous representation [23] to encode the mean rotation. For comparison with the particle filter-based methods, we train the model by sampling the Gaussian in state space and compute the loss as in (9).

TABLE I

Mug Grasping Metrics

	$\log \mathcal{L}_{\mathrm{gm}}$	$\mathcal{L}_d$ [mm]	$\mathcal{L}_d$ [rad]
DPF	159 ± 85	20 ± 5	1.8 ± 0.4
D2P2F	$\mathbf{82 \pm 43}$	15 ± 5	1.7 ± 0.4
LSTM	560 ± 278	20 ± 6	1.6 ± 0.5
Sim-PF	161 ± 143	22 ± 7	2.2 ± 0.4

The models are trained separately for each task, but for the DPF-based models we use a shared set of hyperparameters. F_φ and G_φ are parametrized by multi-layer perceptrons (MLP) with hidden dimensions $[256, 256, 256]$. We employ dropout with rate 0.2 throughout training and inference for DPF and D2P2F. We optimize using Adam [24] with learning rate $5 \cdot 10^{-4}$. We use $N = 50$ particles for training and $N = 100$ for testing.

B. Performance Metrics

For comparison of the above methods, we employ two performance metrics. As a simple distance metric, we use the weighted distance between particles $x_t^{(i)}$ and ground truth $\hat{x}_t$, averaged over time steps

$$\mathcal{L}_d = \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{N} w_t^{(i)} d(\hat{x}_t, x_t^{(i)}), \qquad (11)$$

reported separately for translational and rotational components of the pose. However, the $\mathcal{L}_d$ metric does not give justice to the nature of the highly non-Gaussian belief distributions in tactile manipulation settings. For this reason, we also report the $\log \mathcal{L}_{\mathrm{gm}}$ metric (9), which better takes into account the underlying belief distribution. For the experiments, we report the above metrics for unseen holdout datasets in simulation, with mean and standard deviation calculated over 100 rollouts with different configurations.

V. Evaluation in Simulation

A. Grasping a Mug

We compare the performance metrics for the mug grasping task in Table I. The D2P2F model performs favorably when considering the $\log \mathcal{L}_{\mathrm{gm}}$ metric. In Fig. 4 we show an example rollout of the D2P2F model on a trajectory from the test set. As can be seen, the model produces sensible belief distributions which capture the multimodal nature of apparent object symmetries. When examining the belief distributions produced by the other models, it becomes

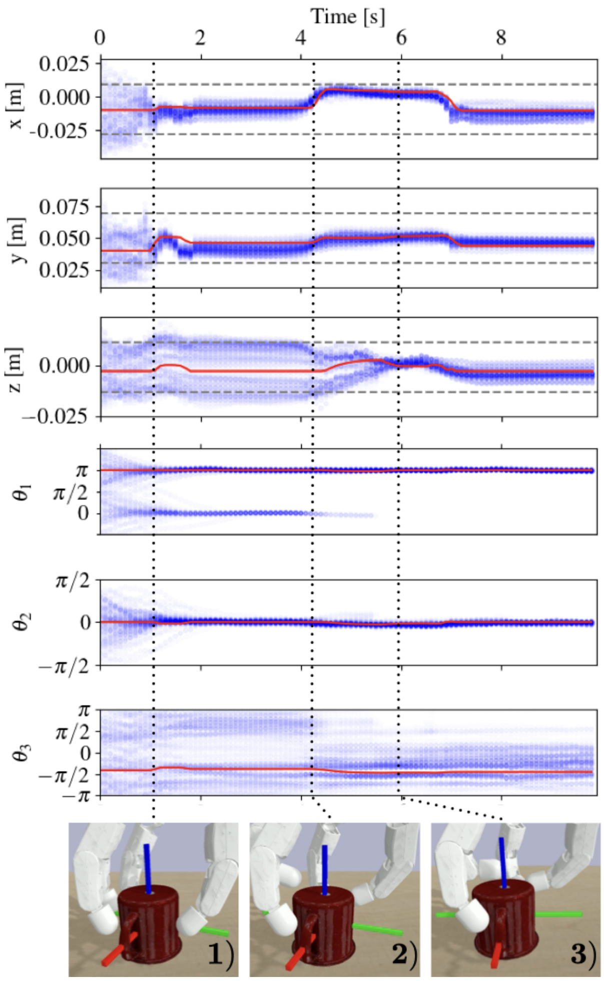

Fig. 4. Exemplary scene from the mug grasping experiment with rollout produced by the D2P2F model. The weighted particle belief is indicated in blue with projections of the particle states to the individual components of the object pose $\mathrm{x}, \mathrm{y}, \mathrm{z}$, and rotations converted to euler angle angles θ_1 (green), θ_2 (red), θ_3 (blue) for visualization purposes. Ground truth trajectory is shown in red. Initially, the cartesian position of the object is unknown, which is reflected by the spread-out particle distribution. For the rotational degrees of freedom, the model produces two opposite modes, corresponding to placements of the cup upright and upside down at $\theta_1 = 0$ and $\theta_1 = \pm\pi$ (compare Fig. 1), which is caused by the prior induced by the dataset used for training (see Section IV-.1). **1)** As the fingers make contact with the mug, the particle belief converges to the ground truth position in x and y components. **2)** One finger approaches the rim of the mug, dissolving the ambiguity of whether or not it was placed upside down (θ_1). **3)** One finger touches the bottom of the mug. From the measured joint positions, the model has learned to infer the relative height of the mug z.

apparent that the unimodal LSTM model fails to account for this multimodality, predicting a mean somewhere in between the two modes which is also reflected in the high $\log \mathcal{L}_{\mathrm{gm}}$.

B. Rotating a Cube

For the cube rotation task, we report the evaluation metrics in Table II. On this task, the D2P2D and the LSTM baseline perform comparably well, indicating that, when using the symmetry-aware representation from Section IV-.2, there is no major advantage in representing highly non-Gaussian beliefs. However, we observe that the sim-PF approach fails to track the target over longer rollouts, which is reflected in the large error rate. The reason is that the underlying policy is a delicate closed-loop controller, for which u_t depends on z_{t-1}. For the sim-PF, this action is then fed to all instances of the simulation $x^{(i)}$. Critically, applying u_t to a simulation with even slightly different initialization of domain parameters results in an open-loop controlled system that can lead to a complete failure of the task (i.e., dropping the cube).

TABLE II
CUBE ROTATION METRICS

	$\log \mathcal{L}_{\mathrm{gm}}$	$\mathcal{L}_d$ [mm]	$\mathcal{L}_d$ [rad]
DPF	74 ± 42	11 ± 3	0.11 ± 0.04
D2P2F	46 ± 27	11 ± 2	0.10 ± 0.03
LSTM	58 ± 40	12 ± 3	0.08 ± 0.03
Sim-PF	320 ± 147	465 ± 1696	0.20 ± 0.07

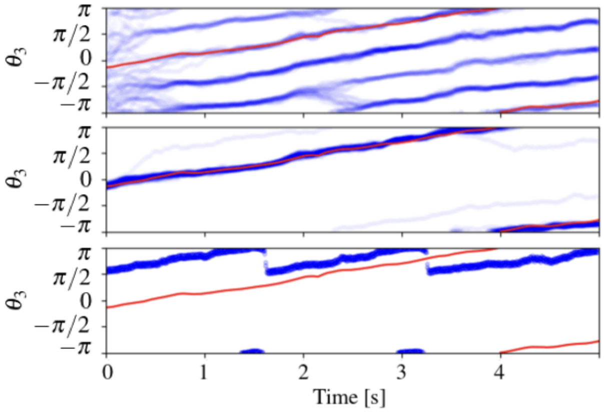

Fig. 5. Angle of rotation θ_3 around the vertical axis for a rollout of the rotation task. Belief distributions produced by learned models (blue) and ground truth trajectory (red). **Top**: D2P2F with particles initialized uniformly. **Center**: D2P2F with particles initialized near ground truth. **Bottom**: Prediction from the LSTM baseline model.

1) Rotational symmetries lead to multimodal belief distributions: In IV-.2 we assumed that two cube states rotated by $\pi/2$ against each other represent the same orientation. However, this may not always be the appropriate choice. For example, if the sides of the cube are colored differently, a visual state estimator could later be fused with the belief from the tactile estimator. In Fig. 5, we show estimated orientations for rollouts of the D2P2F and LSTM models where the output space now covers the full rotation of the cube. The D2P2F is able to handle the multimodality induced by symmetry of the cube and produces a belief distribution with 4 distinct modes, each $\pi/2$ apart. We compare this to a Gaussian belief distribution produced by the LSTM baseline model which predicts a mean which is in-between two modes of the underlying belief, making the estimate highly unreliable.

2) On-the-fly Estimation of System Parameters: The state estimation problem can be easily extended by actively estimating variables of the domain randomization as part of the state. To demonstrate this, we train a D2P2F model to also predict the cube edge length, as well as the mass of the cube. In Fig. 6 we quantify the prediction by calculating

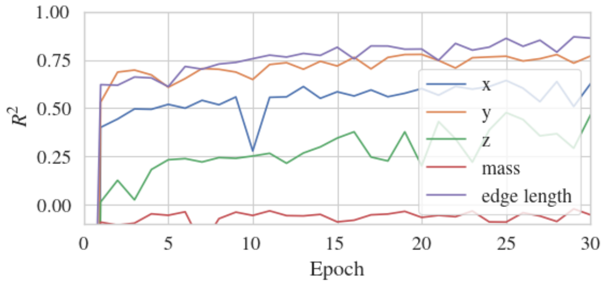

Fig. 6. Coefficient of determination (R^2) during training on the cube rotation task for cartesian dimensions x, y, z, cube mass and cube edge length.

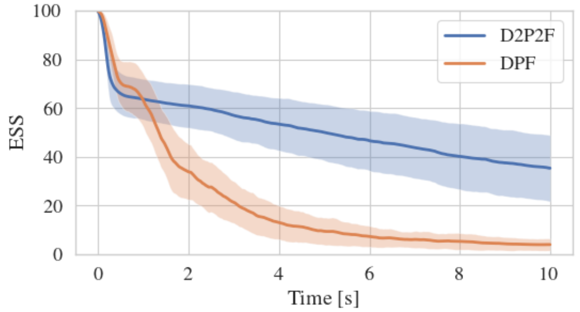

Fig. 7. Effective sample size (ESS) averaged over multiple trajectories vs. time after initialization in the cube rotation task.

the average coefficient of determination over 30 epochs of training, where we take the mean of particles as the model prediction. While the edge length of the cube can be reliably estimated by the D2P2F in an online manner, the mass of the cube can not be predicted by the model from the motion of the rotating policy, which results in a more evenly distributed particle distribution around the data mean in that dimension (not shown).

3) Weight Degeneracy: The quality of approximation of the posterior by weighted samples strongly depends on the distribution of weights. This relation can be characterized by the effective sample size (ESS)[16], where lower values indicate that the belief is dominated by fewer particles, leading to a less expressive posterior distribution. In Fig. 7, we assess the influence of the learned proposal distribution on the ESS. The parametrization used for the D2P2F yields a significantly higher average ESS when compared with the *standard* DPF, even after 1000 timesteps without resampling.

VI. Real World Validation

We transfer the learned models to the physical DLR-Hand II system. We can verify that for the cube rotation tasks, the filters are able to track the rotation of the cube over time frames of several minutes. We also test the real-time capabilities of the algorithm. Although the implementation in PyTorch is not optimized for inference, we measure inference times of $< 10\,\mathrm{ms}$ for 100 particles, enabling fast online state estimation with update rates of $> 100\,\mathrm{Hz}$. In the supplementary video, we show runs of the trained filters for both the mug and the cube tasks.

VII. Conclusion

We proposed a novel differentiable particle filter variant, the deep proposal differentiable particle filter D2P2F, and showed its application in the challenging task of tactile 3D pose estimation with a humanoid hand. We were able to show that incorporating measurements into the propagation of particles can be highly beneficial, especially in situations where belief distributions are non-Gaussian and multimodal. For this, we performed a quantitative comparison in simulation to the *standard* DPF and other learning and non-learning-based estimation methods. Finally, we validated the filter in experiments on a real robotic hand, running the filter in real-time.

In our experiments, contact information was derived solely by using the torque-controlled joints. To further improve the quality of the state estimate, in future work, dedicated tactile hardware like a touch-sensitive skin with high spatio-temporal resolution [25] could be employed, which can be integrated within the same framework described in this paper. Also, we want to be able to actively refine the estimate by using the belief distributions online as control input. To make informed decisions in tactile manipulation tasks, we expect that it is important to respect the multimodal belief distributions, which can be represented by the methods proposed in this work.

References

[1] OpenAI *et al.*, “Solving rubik’s cube with a robot hand,” *arXiv preprint*, 2019.

[2] J. Butterfaß, M. Grebenstein, H. Liu, and G. Hirzinger, “DLR-Hand II: Next generation of a dextrous robot hand,” in *Proc. IEEE International Conference on Robotics and Automation*, 2001, pp. 109–114.

[3] R. Jonschkowski, D. Rastogi, and O. Brock, “Differentiable particle filters: End-to-end learning with algorithmic priors,” in *Proceedings of Robotics: Science and Systems*, 2018.

[4] P. Karkus, D. Hsu, and W. S. Lee, “Particle filter networks with application to visual localization,” in *Conference on robot learning*, 2018.

[5] M. C. Koval, N. S. Pollard, and S. S. Srinivasa, “Pose estimation for planar contact manipulation with manifold particle filters,” *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 922–945, 2015.

[6] M. Pfanne and M. Chalon, “Ekf-based in-hand object localization from joint position and torque measurements,” in *Proc. Int. Conf. Intelligent Robots and Systems*, 2017.

[7] F. Wirnshofer *et al.*, “State estimation in contact-rich manipulation,” in *International Conference on Robotics and Automation*, 2019.

[8] G. Vezzani *et al.*, “Memory unscented particle filter for 6-dof tactile localization,” *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1139–1155, 2017.

[9] T. Haarnoja, A. Ajay, S. Levine, and P. Abbeel, “Backprop kf: Learning discriminative deterministic state estimators,” in *Advances in neural information processing systems*, 2016, pp. 4376–4384.

[10] R. Jonschkowski and O. Brock, “End-to-end learnable histogram filters,” 2016.

[11] A. Kloss, G. Martius, and J. Bohg, “How to train your differentiable filter,” *Autonomous Robots*, pp. 1–18, 2021.

[12] A. Doucet, N. d. Freitas, and N. Gordon, “An introduction to sequential monte carlo methods,” in *Sequential Monte Carlo methods in practice*. Springer, 2001, pp. 3–14.

[13] C. Snyder, T. Bengtsson, and M. Morzfeld, “Performance bounds for particle filters using the optimal proposal,” *Monthly Weather Review*, vol. 143, no. 11, pp. 4750–4761, 2015.

[14] R. Van Der Merwe, A. Doucet, N. De Freitas, and E. Wan, “The unscented particle filter,” *Advances in neural information processing systems*, vol. 13, pp. 584–590, 2000.

[15] X. Chen, H. Wen, and Y. Li, "Differentiable particle filters through conditional normalizing flow," in *FUSION*, 2021.

[16] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and computing*, vol. 10, no. 3, pp. 197–208, 2000.

[17] M. A. Lee *et al.*, "Multimodal sensor fusion with differentiable filters," in *Proc. Int. Conf. Intelligent Robots and Systems*, 2020.

[18] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder, "Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds," *Information Fusion*, vol. 14, no. 1, pp. 57–77, 2013.

[19] L. Sievers, J. Pitz, and B. Bäuml, "Learning purely tactile in-hand manipulation with a torque-controlled hand," in *Proc. IEEE International Conference on Robotics and Automation*, 2022.

[20] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," 2016–2021.

[21] B. Calli *et al.*, "Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set," *IEEE Robotics Automation Magazine*, vol. 22, no. 3, pp. 36–52, 2015.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] Y. Zhou *et al.*, "On the continuity of rotation representations in neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5745–5753.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[25] S. Baishya and B. Bäuml, "Robust material classification with a tactile skin using deep learning," in *Proc. IEEE International Conference on Intelligent Robots and Systems*, 2016.