The Manifold Particle Filter for State Estimation on High-dimensional Implicit Manifolds

Michael C. Koval mkoval@cs.cmu.edu
Matthew Klingensmith mklingen@andrew.cmu.edu
Siddhartha S. Srinivasa siddh@cs.cmu.edu
Nancy S. Pollard nsp@cs.cmu.edu
Michael Kaess kaess@cmu.edu

***Abstract*—We estimate the state of a noisy robot arm and underactuated hand using an implicit Manifold Particle Filter (MPF) informed by contact sensors. As the robot touches the world, its state space collapses to a contact manifold that we represent implicitly using a signed distance field. This allows us to extend the MPF to higher (six or more) dimensional state spaces. Earlier work, which explicitly represents the contact manifold, was only capable of scaling to three dimensions. Through a series of experiments, we show that the implicit MPF converges faster and is more accurate than a conventional particle filter during periods of persistent contact. We present three methods of drawing samples from an implicit contact manifold, and compare them in experiments.**

I. Introduction

Robots often have imperfect proprioception. This may arise from difficult-to-model transmissions, underactuated degrees of freedom, or poorly calibrated sensors. Fig. 1 (Top) shows a Barrett WAM arm [1] touching a box. The WAM measures its joint angles through a cable drive transmission that suffers from hysteresis caused by stretch in the cables [3]. Fig. 1 (Bottom) shows the BarrettHand [2] grasping a bottle. The position of the distal finger joints depends on the state of a mechanical clutch that engages when a torque is met.

In both cases, the configuration reported by the robot's sensors is in error. In Fig. 1 (Top), the robot believes that it is several centimeters above the box even though a finger is in contact. In Fig. 1 (Bottom), the robot computed the configuration under the assumption that the clutch did not engage even though all three fingers made contact with the object. In both cases, the reported joint positions are inconsistent with the robot's readings from its contact sensors.

Our goal is to use contact sensors to refine a robot's estimate of its configuration. Like related work (Section II) dating back to the 1970s [4], we frame this as a problem of Bayesian estimation. State is the configuration of the robot, an action is a commanded change in configuration, and an observation is a measurement from the robot's proprioceptive and contact sensors (Section III).

Traditional Bayesian estimation techniques perform poorly on this domain because a contact observation constrains the set of feasible states to a lower dimensional *contact manifold*

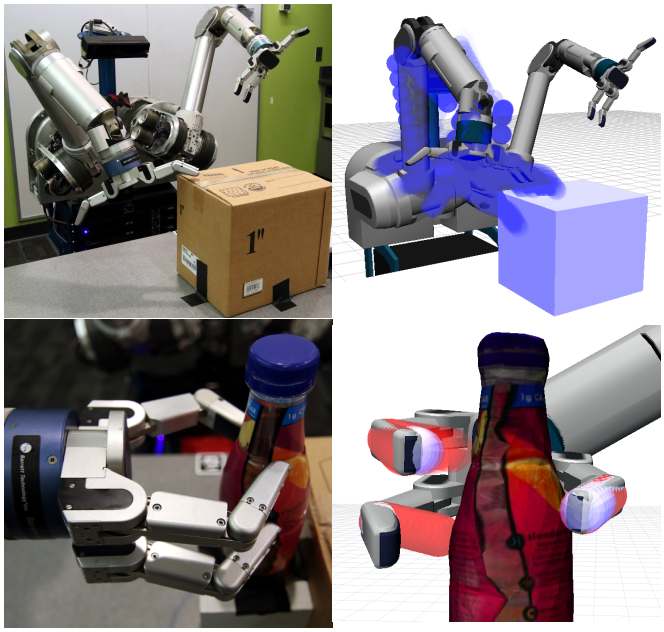


Fig. 1: Two examples of imperfect proprioception. (Top) The Barrett WAM [1] touching a box. The solid render shows the configuration of the arm estimated by the WAM's encoders and the blue renders are particles. Note that there are two modes in the distribution. (Bottom) The BarrettHand [2] grapsing a bottle with no position sensors on its distal joints. The solid render shows the configuration of the hand when the distal joints are assumed to be fixed. The red render shows the mode estimated from the particles.

that place the active sensors in non-penetrating contact with the environment. The extended [5] and unscented [6] Kalman filters assume a Gaussian distribution over state, which cannot accurately represent a probability distribution over this manifold. The conventional particle filter (CPF) [7] suffers from particle deprivation because there the manifold has zero measure: there is zero probability of drawing a state from the ambient space that lies on it.

Instead, we use the *manifold particle filter* (MPF) [8], which adapts its sampling strategy based on the most recent observation. When the robot has not observed contact, the MPF is identical to the CPF. When the robot does sense contact, the MPF draws samples from the contact manifold and re-weights them to enforce temporal consistency. This algorithm has been successfully used to estimate the pose of an object relative to the hand during planar manipulation by

All authors are affiliated with the Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh PA, 15213.

This work was supported by a NASA Space Technology Research Fellowship (award NNX13AL62H), the National Science Foundation (awards IIS-1218182, IIS-1409003, and IIS-1426703), the U.S. Office of Naval Research (award N000141210613), and the Toyota Motor Corporation.

using an explicit representation of the contact manifold.

However, these explicit representations do not scale to the high-dimensional state space of robot configurations. Our key insight is to build an *implicit representation of the contact manifold* using a signed distance field (SDF) and constraint projection (Section IV). First, we draw a set of particles uniformly from the region of state space near the previous set of particles. Then, we use the SDF and manipulator Jacobian to project those samples onto the manifold.

We demonstrate the efficacy of this technique in simulation (Section V) and real-robot (Section VI) experiments for two problem domains. First, we consider a two, three, or seven degree-of-freedom arm with noisy positions sensors moving in a static environment. Second, we consider an underactuated robotic hand grasping a static object. In both cases, we show that the proposed technique outperforms the CPF.

We believe that the proposed approach is applicable to a wide variety of problem domains. However, it has one key limitation: it requires a known, static environment. We plan to relax this assumption in future work by incorporating the pose of dynamic objects into the filter's state space (Section VII).

II. Related Work

There are a variety of approaches that use feedback from contact sensors for manipulation. One approach is to plan a sequence of move-until-touch actions that are guaranteed to localize an object to the desired accuracy [9], [10], [11]. Other approaches formulate the problem as a partially observable Markov decision process [12] and solve for a policy that optimizes expected reward [13], [14], [15]. These algorithms require an efficient implementation of a Bayesian state estimator that can be queried many times during planning. Our estimator could be integrated into one of these algorithms.

Recent work has used the CPF to estimate the pose of an object while it is being pushed using visual and tactile observations [16], [17]. Unfortunately, the CPF performs poorly because the set of feasible configurations lie on a lower-dimensional contact manifold. The MPF avoids particle deprivation by sampling from an approximate representation of this manifold [8]. However, building an explicit representation is only feasible for low-dimensional, typically planar, problems.

In this work, we extend the MPF to estimate the full—typically six or more dimensional—configuration of a robot under proprioceptive uncertainty. Depth-based trackers, such as articulated ICP [18], [19], GMAT [20], and DART [21], can track the configuration of a robot using commercially available depth sensors. DART has been extended to incorporate contact observations using a method similar to our constraint projection [22]. However, all of these methods maintain a uni-modal state estimate and, thus, perform best when the robot is visible and un-occluded. In contrast, the MPF maintains a full distribution and only requires contact sensors.

The work most similar to our own aims to localize a mobile robot in a known environment using contact with the environment. Prior work has used the CPF to localize the pose of a mobile manipulator by observing where its arm contacts the environment [23]. The same approach was used to localize a quadruped on known terrain by evaluating its static stability [24]. Estimating the base pose of a mobile robot is equivalent to solving the problem formulated in this paper with the addition of a single, unconstrained six degree-of-freedom joint that attaches the robot to the world.

Techniques for estimating the configuration of an articulated body from contact sensors are applicable to humans as well as robots. The computer graphics community has instrumented objects with contact sensors [25] and used multi-touch displays [26] to reconstruct the configuration of a human hand from contact observations. These approaches generate natural hand configurations by interpolating between data points collected on the same device. Other work has used machine learning to reconstruct the configuration of a human from measured ground reaction forces [27]. We hope that our approach is also useful in these problem domains.

Recent work on calibration estimates the robot's joint angles and Denavit Hartenberg parameters from a single self-touch [28]. Our work also estimates the joint angles from touch, but differs in that we use online filtering rather than a batch optimization for offline calibration.

III. Background

We formulate robot configuration estimation as a Bayesian filtering problem. As the robot moves, we estimate its configuration from actions and observations in real-time.

A. Problem Definition

Consider a robot with configuration space Q. At each time step the robot executes a control input $\mathbf{u}_t \in U$, transitions to the successor state $\mathbf{q}_t \sim p(\mathbf{q}_t|\mathbf{q}_{t-1}, \mathbf{u}_t)$, and receives an observation $\mathbf{z}_t \sim p(\mathbf{z}_t|\mathbf{q}_t, \mathbf{u}_t)$. An observation $\mathbf{z}_t = (\mathbf{q}_\mathrm{e}, \mathbf{c}) \in Z$ includes a noisy estimate $\mathbf{q}_\mathrm{e} \in \mathbb{R}^n$ of the robot's configuration and a binary vector of readings $\mathbf{c} \in \{0,1\}^m$ from the robot's m contact sensors.

Our goal is to estimate the *belief state* $\mathrm{bel}(\mathbf{q}_t) = p(\mathbf{q}_t|\mathbf{u}_{1:t}, \mathbf{z}_{1:t})$, the probability distribution over the state $\mathbf{q}_t$ given the history of actions $\mathbf{u}_{1:t} = \mathbf{u}_1, \ldots, \mathbf{u}_t$ and observations $\mathbf{z}_{1:t} = \mathbf{z}_1, \ldots, \mathbf{z}_t$.

1) Transition Model: Our method is applicable to any transition model where it is possible to sample from $p(\mathbf{q}_t|\mathbf{q}_{t-1}, \mathbf{u}_t)$ for given values of $\mathbf{q}_t$ and $\mathbf{u}_t$. We assume that the world does not change in response to the robot's touch.

In our experiments, we choose $\mathbf{u}$ to be commanded joint velocities and define the transition model to be

$$\mathbf{q}_t = \mathbf{F}\left(\mathbf{q}_{t-1} + \mathbf{u}_t \Delta t + \epsilon_\mathrm{a}\right),$$

the forward integration of the control input $\mathbf{u}_t$ corrupted by noise $\epsilon_\mathrm{a} \sim \mathrm{uniform}\, B(0, r_a)$ drawn from a ball B of radius r_a centered at the origin. We implement $\mathbf{F}$ using a physics model that simulates frictionless soft contact with the environment.

2) Observation Model: We assume that proprioceptive and contact sensor observations are conditionally independent given $\mathbf{q}_t$ and $\mathbf{u}_t$. Under this assumption, we can write

$$p(\mathbf{z}_t|\mathbf{q}_t, \mathbf{u}_t) = p(\mathbf{q}_\text{e}|\mathbf{q}_t, \mathbf{u}_t)p(\mathbf{c}|\mathbf{q}_t, \mathbf{u}_t)$$

as the product of two marginal distributions.

The distribution $p(\mathbf{q}_\text{e}|\mathbf{q}_t, \mathbf{u}_t)$ models uncertainty in the robot's joint position sensors. We make no assumptions about this distribution other than that it is possible to evaluate the probability density for given values of $\mathbf{q}_\text{e}$, $\mathbf{q}_t$, and $\mathbf{u}_t$.

The distribution $p(\mathbf{c}|\mathbf{q}_t, \mathbf{u}_t)$ models the robot's contact sensors. Each sensor is a rigid body that is attached to one of the robot's links. The sensor returns "contact" if any part of the sensor touches the environment and otherwise returns "no contact." Similar to prior work [8], we assume that the contact sensors do not generate false positives.

In our experiments, we model $\mathbf{q}_\text{e}$ as a measurement

$$\mathbf{q}_\text{e} = \mathbf{q} + \Delta\mathbf{q} + \epsilon_\text{o}$$

corrupted by noise $\epsilon_\text{o} \sim \mathcal{N}(0, \Sigma_{\Delta\mathbf{q}})$ and a static, but unknown, joint offset $\Delta\mathbf{q}$. The initial offset $\Delta\mathbf{q}_0$ is drawn from a known distribution $p(\Delta\mathbf{q}_0)$ and remains constant.

Since state is determined by the offset and proprioceptive observation, we can rewrite

$$p(\mathbf{q}_\text{e}|\mathbf{q}, \mathbf{u}) = \mathcal{N}(\mathbf{q} + \Delta\mathbf{q}, \Sigma_{\Delta\mathbf{q}}).$$

and estimate a belief over $\Delta\mathbf{q}$. However, whenever state is mentioned in this work, what is meant is $\mathbf{q}$.

If one or more joints are unobserved, such as the distal joints in Fig. 1 (Bottom), we treat the unobserved dimensions of $\mathbf{q}$ as initially having a uniform distribution.

B. Bayes Filter

The *Bayes filter* provides method of recursively constructing $\text{bel}(\mathbf{q}_t)$ from $\text{bel}(\mathbf{q}_{t-1})$. Given an initial belief $\text{bel}(\mathbf{q}_0)$, the Bayes filter applies the update rule

$$\text{bel}(\mathbf{q}_t) = \eta\, p(\mathbf{z}_t|\mathbf{q}_t, \mathbf{u}_t) \int_Q p(\mathbf{q}_t|\mathbf{q}_{t-1}, \mathbf{u}_t)\text{bel}(\mathbf{q}_{t-1})\, d\mathbf{q}_{t-1}$$

where η is a normalization constant. This equation is derived from the definition of the belief state and the Markov property.

There are several ways of implementing the Bayes filter. The discrete Bayes filter represents $\text{bel}(\mathbf{q}_t)$ as a piecewise constant histogram. Discretization is intractable on our problem because Q is typically high dimensional: $n \geq 6$ for a most manipulators.

The Kalman filter, extended Kalman filter [5], and unscented Kalman filter [6] avoid discretization by assuming that $\text{bel}(\mathbf{q}_t)$ is Gaussian. This assumption is not valid for our problem: the observation model $p(\mathbf{c}|\mathbf{q}_t, \mathbf{u}_t)$ is discontinuous and tends to produce multi-modal belief states.

Algorithm 1: CONVENTIONAL PARTICLE FILTER

Input: Q_{t-1} particles sampled from $\text{bel}(\mathbf{q}_{t-1})$
Output: Q_t particles sampled from $\text{bel}(\mathbf{q}_t)$

$Q_t \leftarrow \emptyset$
for $\mathbf{q}_{t-1}^{[i]} \in Q_{t-1}$ **do**
  $\mathbf{q}_t^{[i]} \sim p(\mathbf{q}_t|\mathbf{q}_{t-1}^{[i]}, \mathbf{u}_t)$
  $w_t^{[i]} \leftarrow p(\mathbf{z}_t|\mathbf{q}_t^{[i]}, \mathbf{u}_t)$
end
$Q_t \leftarrow \text{RESAMPLE}(Q_t)$

C. Conventional Particle Filter

Instead, we use the particle filter [7]. The *particle filter* (Alg. 1) is an implementation of the Bayes filter that represents $\text{bel}(\mathbf{q}_t)$ using a discrete set of weighted samples $Q_t = \{\langle \mathbf{q}_t^{[i]}, w_t^{[i]} \rangle\}_{i=1}^k$, known as *particles*. The set of particles Q_t at time t is recursively constructed from the set of particles X_{t-1} at time $t-1$ using importance sampling.

First, the particle filter samples a set of k states $\mathbf{q}_t^{[i]} \sim \rho_\text{conv}(\mathbf{q})$ from a *proposal distribution* $\rho_\text{conv}(\mathbf{q})$. Conventionally, the proposal distribution is chosen to be

$$\rho_\text{conv}(\mathbf{q}_t) = \int_Q p(\mathbf{q}_t|\mathbf{q}_{t-1}, \mathbf{u}_t)\text{bel}(\mathbf{q}_{t-1})\, d\mathbf{q}_{t-1}, \qquad (1)$$

the transition model applied to the previous belief state. This can be implemented by forward simulating Q_{t-1} to time t using the transition model.

Next, the particle filter computes an *importance weight* $w_t^{[i]} = \text{bel}(\mathbf{q}_t^{[i]})/\rho_\text{conv}(\mathbf{q}_t^{[i]})$ to correct for the discrepancy between the proposal distribution $\rho_\text{conv}(\mathbf{q}_t)$ and the desired distribution $\text{bel}(\mathbf{q}_t)$. When using the proposal distribution shown in Eq. 1, the corresponding importance weight is $w_t^{[i]} = \eta\, p(\mathbf{z}_t|\mathbf{q}_t^{[i]}, \mathbf{u}_t)$. This can be thought of as updating Q_t to agree with the most recent observation $\mathbf{z}_t$.

Finally, the particle filter periodically resamples each particle in Q_t with replacement, with probability proportional to its weight. This process is known as *sequential importance resampling* (SIR) and is necessary to achieve good performance over long time horizons.

D. Degeneracy of the Conventional Particle Filter

Prior work [8] has shown that the conventional particle filter (CPF) performs poorly with contact sensors because $\text{bel}(s_t)$ collapses to a lower-dimensional manifold. This leads to *particle deprivation* during contact, where $w_t^{[i]} = 0$ for all but a few particles, because it is vanishingly unlikely that a particle sampled from the transition model will lie on the zero measure contact manifold.

To see why this is the case, consider a 2-D, two jointed and two linked robot with a single point contact sensor on its distal link (Fig. 2). When the robot contacts the environment, the contact state of its sensor changes. Infinitesimal motion along the surface results in the same contact state, but infinitesimal motion away from the surface results in a different contact state. The set of configurations with the same contact

Algorithm 2: MANIFOLD PARTICLE FILTER

Input: $\mathbf{u}_t$, Q_{t-1} particles sampled from $\text{bel}(\mathbf{q}_{t-1})$
Output: Q_t particles sampled from $\text{bel}(\mathbf{q}_t)$

$Q_t \leftarrow \emptyset$
for $\mathbf{q}_{t-1}^{[i]} \in Q_{t-1}$ **do**
  if $\mathbf{c}_t = 0$ **then**
    $\mathbf{q}_t^{[i]} \sim p(\mathbf{q}_t | \mathbf{q}_{t-1}^{[i]}, \mathbf{u}_t)$
    $w_t^{[i]} \leftarrow p(\mathbf{c}_t | \mathbf{q}_t^{[i]}, \mathbf{u}_t)$
  else
    $\mathbf{q}_t^{[i]} \sim \text{uniform } M(\mathbf{c})$
    $w_t^{[i]} \leftarrow \text{KERNELDENSITYESTIMATE}(Q_{t-1}^{+}, \mathbf{q}_t^{[i]})$
  end
end
$Q_t \leftarrow \text{RESAMPLE}(Q_t)$

state form a lower-dimensional manifold in the ambient state space.

E. Manifold Particle Filter

The *manifold particle filter* (MPF, Alg. 2) avoids particle deprivation by operating in two modes [8]. When no contact is observed, the MPF is behaves identically to the CPF by sampling particles from the transition model and weighting them by the observation model. When contact is observed, the MPF switches to sampling particles from the observation model and weighting them by the transition model.

Both modes of the MPF implement importance sampling with different proposal distributions. During contact, the MPF samples particles from the *dual proposal distribution*

$$\rho_{\text{dual}}(\mathbf{q}_t) = \frac{p(\mathbf{z}_t | \mathbf{q}_t, \mathbf{u}_t)}{p(\mathbf{z}_t | \mathbf{u}_t)},$$

where $p(\mathbf{z}_t | \mathbf{u}_t)$ is a normalization constant. Sampling from $\rho_{\text{dual}}(\mathbf{q}_t)$ generates configurations that are consistent with the most recent observation $\mathbf{z}_t$. The remainder of this paper describes how to generate these samples efficiently.

The importance weight for a particle $\mathbf{q}_t^{[i]} \sim \rho_{\text{dual}}(\mathbf{q}_t)$ is

$$w_t^{[i]} = \eta \int_Q p(\mathbf{q}_t | \mathbf{q}_{t-1}, \mathbf{u}_t) \text{bel}(\mathbf{q}_{t-1}) \, d\mathbf{q}_{t-1} \tag{2}$$

where η is another normalization constant. The importance weight $w_t^{[i]}$ incorporates information from $\text{bel}(\mathbf{q}_{t-1})$ into the posterior belief state; i.e. enforces temporal consistency with the transition model.

Computing $w_t^{[i]}$ exactly is not possible with a particle-based representation of $\text{bel}(\mathbf{q}_{t-1})$. Instead, we forward simulate the particles Q_{t-1} by applying the transition model just like we do in the CPF. This set of particles Q_{t-1}^{+} are distributed according to the right-hand side of Eq. 2. We approximate the weight $w_t^{[i]}$ using a kernel density estimate [29] built from Q_{t-1}^{+}.

IV. IMPLICIT CONTACT MANIFOLD REPRESENTATION

Implementing the MPF requires sampling from the lower-dimensional contact manifold associated with the active contact sensors. In this section, we formally define the contact manifold (Section IV-A) associated with contact observation $\mathbf{c}$ and explain why it is infeasible to build an explicit representation of this manifold.

Instead, we implicitly define the contact manifold as the iso-contour of a loss function (Section IV-B) and use a local optimizer to project onto it (Section IV-C) by using a signed-distance field to compute gradient information (Section IV-D). Finally, we describe three different methods of using projection to sample from the observation model (Section IV-E).

A. Contact Manifold

Suppose the robot is in a static environment with obstacles $X_{\text{obs}} \subseteq \mathbb{R}^3$ with boundary $X_{\text{surf}} = X_{\text{obs}} \setminus \text{int}(X_{\text{obs}})$. The robot's i-th contact sensor is a rigid body with geometry $c(\mathbf{q}) \subseteq \mathbb{R}^3$ in configuration $\mathbf{q}$. If the robot senses contact with sensor i, then we know that $c_i(\mathbf{q})$ is in non-penetrating contact with the environment; i.e. $c_i(\mathbf{q}) \cap X_{\text{surf}} \neq \emptyset$.

We define the *sensor contact manifold* M_i of sensor i as

$$M_i = \{\mathbf{q} \in Q : c(\mathbf{q}) \cap X_{\text{surf}} \neq \emptyset\},$$

the set of all configurations that put $c_i(\mathbf{q})$ in contact with the environment. If multiple contact sensors are active, then we know that all active sensors are in non-penetrating contact with the environment. Fig. 2a shows a simple example of this.

The *observation contact manifold* $M(\mathbf{c})$ is given by the intersection of the active sensor contact manifolds

$$M(\mathbf{c}) = \bigcap_{i \in \Phi(\mathbf{c})} M_i,$$

where $\Phi(\mathbf{c})$ denotes the indices of the sensors active in $\mathbf{c}$.

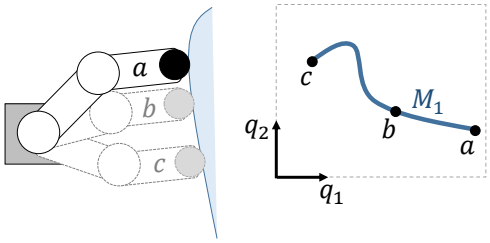
Explicitly representing $M(\mathbf{c})$ is tractable for small problems. For example, Fig. 4a shows a 2-D robot with two joints in an environment consisting of a single point. In this environment, the contact manifold can be computed easily using analytic inverse kinematics. However, as the environment and dimensionality of the problem increase in complexity (Fig. 4b, Fig. 4c), deriving an explicit representation of the manifold becomes computationally infeasible, because it requires computing inverse kinematics solutions for *every* surface point which would cause the contact $\mathbf{c}$.

B. Implicit Representation of the Contact Manifold

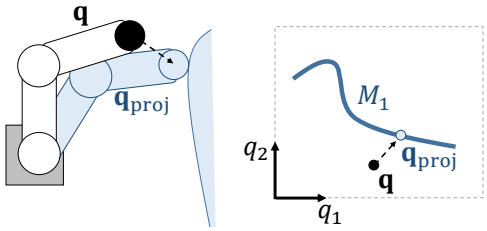
Luckily for the MPF, we do not need to compute an explicit representation of $M(\mathbf{c})$: we only have to be able to draw samples from it. In this work, we first sample from the full state space, and then *project* onto the $M(\mathbf{c})$, which is only represented implicitly as the iso-contour of a loss function. We then reject any sample that is not close enough to the manifold.

We represent the sensor contact manifold M_i as the zero iso-contour $M_i = \{\mathbf{q} \in Q : \text{dist}(c_i(\mathbf{q}), X_{\text{obs}}) = 0\}$ of the *signed distance function*

$$\text{dist}(X, Y) = \min_{x \in X} \begin{cases} \text{dist}(x, Y) & : x \notin Y \\ -\text{dist}(x, Y^c) & : \text{otherwise} \end{cases}$$



(a) Contact manifold



(b) Projection onto the contact manifold

Fig. 2: The contact manifold of a contact sensor (black circle) on a two degree-of-freedom robot. The robot's configuration space, parameterized by q_1 and q_2 is shown on the right. (a) Three different configurations a, b, and c that lie on the contact manifold M_1. (b) The configuration $\mathbf{q}$ is projected to a configuration $\mathbf{q}_{\text{proj}}$ that lies on M_1.

between the sensor and the environment. A signed distance is a positive value equal to the distance between two disjoint sets or negative value equal to the deepest penetration between two intersecting sets. The signed distance $\text{dist}(c_i(\mathbf{q}), X_{\text{obs}})$ is zero iff contact sensor i is in non-penetrating contact with the environment in configuration $\mathbf{q}$.

A configuration $\mathbf{q}$ lies on the observation contact manifold $M(\mathbf{c})$ if the signed distance $\text{dist}(c_i(\mathbf{q}), X_{\text{obs}}) = 0$ for all sensors active $i \in \Phi(\mathbf{c})$ in observation $\mathbf{c}$. We represent this set as the zero iso-contour $M(\mathbf{c}) = \{\mathbf{q} \in Q : D_{\mathbf{c}}(\mathbf{q}) = 0\}$ of the loss function

$$D_{\mathbf{c}}(\mathbf{q}) = \sum_{i \in \Phi(\mathbf{c})} \left[\text{dist}(c_i(\mathbf{q}), X_{\text{obs}})\right]^2,$$

which is zero iff $\mathbf{q} \in M(\mathbf{c})$. Any function that satisfies this property is sufficient. We choose sum-of-squared distances to simplify the projection operator described below.

C. Projecting onto the Contact Manifold

We project a sample from the ambient space $\mathbf{q}$ onto the the contact manifold $M(\mathbf{c})$ by solving the optimization problem

$$\text{proj}(\tilde{\mathbf{q}}, \mathbf{c}) = \underset{\mathbf{q} \in N(\tilde{\mathbf{q}})}{\arg\min}\, D_{\mathbf{c}}(\mathbf{q}) \tag{3}$$

in a neighborhood $N(\tilde{\mathbf{q}}) \subseteq Q$ around an initial configuration $\tilde{\mathbf{q}} \in Q$. If the distance $D_{\mathbf{c}}(\mathbf{q}) = 0$ at the end of the optimization, then we have found a configuration $\mathbf{q} \in M(\mathbf{c})$. Fig. 2b shows an example of the outcome of this process.

We implement the minimization in Eq. 3 using simple gradient descent optimization. The optimizer is initialized with $\mathbf{q}^{(0)} = \tilde{\mathbf{q}}$ and iteratively applies the update rule

$$\mathbf{q}^{(j+1)} = \mathbf{q}^{(j)} - \lambda \nabla D_{\mathbf{c}}^T D_{\mathbf{c}}(\mathbf{q}^{(j)})$$

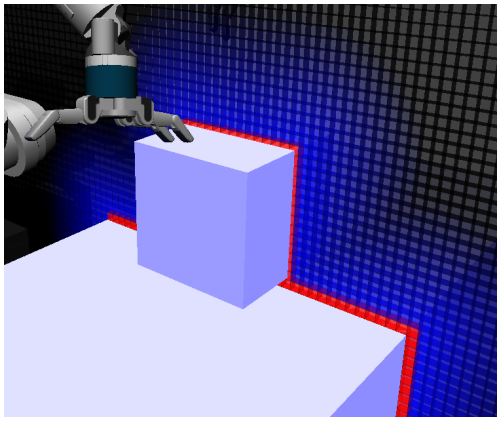


Fig. 3: Voxels used to represent the signed distance field Φ at a 2 cm resolution. Blue voxels have $\Phi(\cdot) > 0$ and red voxels have $\Phi(\cdot) \leq 0$.

until $\mathbf{q}^{(j)}$ has converged, where λ is the learning rate. Performing this update requires computing the gradient

$$\nabla D_{\mathbf{c}} = \nabla_{\mathbf{q}} \,\text{dist}\, \mathbf{c}(\mathbf{q}) \quad = 2 \sum_{i \in \Phi(\mathbf{c})} \nabla_{\mathbf{q}} \,\text{dist}(c_i(\mathbf{q}), X_{\text{obs}}),$$

which, in turn, requires computing the gradient of the signed distance function. We describe how to efficiently compute $\text{dist}(\cdot, \cdot)$ and $\nabla_{\mathbf{q}} \,\text{dist}(\cdot, \cdot)$ in the next section.

Note that this procedure may converge to a configuration where $D_{\mathbf{c}}(\mathbf{q}) \geq \epsilon$ because (1) there is no solution in the neighborhood $N(\tilde{\mathbf{q}})$ or (2) the optimizer reached a local minimum. In either case, the projection fails and needs to be re-initialized with a different $\tilde{\mathbf{q}} \in Q$.

This method of projecting onto the contact manifold using an implicit representation is commonly used in computer graphics to quickly compute contacts for simulated collision resolution and inverse kinematics. The method described here for computing an implicit contact manifold for an articulated body is similar to the one described in [30].

D. Signed Distance Computation

Evaluating $D_{\mathbf{c}}(\mathbf{q})$ requires computing the distance $\text{dist}(c_i(\mathbf{q}), X_{\text{obs}})$ between each contact sensor $c_i(\mathbf{q})$ and the environment X_{obs}. Computing this distance metric is difficult and, potentially computationally expensive, for arbitrary geometric shapes. To avoid this, we approximate the static environment, which may contain arbitrary geometry, with a voxel grid and each contact sensors with a collection of geometric primitives (e.g. spheres, capsules, boxes).

As an offline pre-computation step, we compute a discrete signed distance field over the voxel grid using the technique from Felzenszwalb and Huttenlocher [31]. A *signed distance field* (SDF) is a function $\Phi : x \in \mathbb{R}^3 \mapsto \text{dist}(x, X_{\text{obs}})$ that maps each point x in the workspace to its signed distance to the nearest obstacle. Fig. 3 shows an example of a SDF computed in built in this way.

Without loss of generality, assume that the geometry of each contact sensor is a sphere with center $\mathbf{p}_i(\mathbf{q}) \in \mathbb{R}^3$ and radius $r_i \in \mathbb{R}^+$. The signed distance between the sensor and

the environment is given by

$$\operatorname{dist}(c_i(\mathbf{q}), X_{\text{obs}}) = \Phi(\mathbf{p}_i(\mathbf{q})) - r_i.$$

The gradient of the distance is given by

$$\nabla_{\mathbf{q}} \operatorname{dist}(c_i(\mathbf{q}), X_{\text{obs}}) = [\nabla_{\mathbf{x}} \Phi(\mathbf{p}_i(\mathbf{q}))]\, J_i(\mathbf{q})$$

where $J_i(\mathbf{q}) = \frac{\partial \mathbf{p}_i}{\partial \mathbf{q}}$ is the linear Jacobian of the manipulator evaluated at point $c_i(\mathbf{q})$.

Critically, evaluating Φ requires a single memory lookup, $\nabla_x \Phi(x)$ may be approximated by a finite difference, and the size of the voxel grid is independent of the dimensionality of Q. This is the same representation of the environment used by CHOMP, a gradient-based trajectory optimizer [32].

E. Sampling from the Contact Manifold via Projection

The projection operator described above starts with a single initial configuration $\tilde{\mathbf{q}} \in Q$ and projects it onto the contact manifold $M(\mathbf{c})$. Sampling from the dual proposal distribution requires n such samples distributed uniformly over $M(\mathbf{c})$. We describe three different approaches for selecting the set of initializations $\tilde{Q}_t = \{\tilde{\mathbf{q}}^{[i]}\}_{i=1}^n$ used to generate the set of particles Q_t. If a projection operation fails, i.e. $D_{\mathbf{c}}(\mathbf{q}^{[i]}) \geq \epsilon$, then we generate a new initialization $\tilde{\mathbf{q}}^{[i]}$ and try again.

1) Uniform Projection: The simplest strategy is to sample $\tilde{\mathbf{q}}^{[i]} \sim \operatorname{uniform}(Q)$ uniformly from the robot's configuration space. This method is unbiased with respect to the previous set of particles Q_{t-1}. Unfortunately, since Q is high dimensional, it may take a large number of particles to adequately cover the manifold. This may lead to particle deprivation.

2) Particle Projection: We can focus our samples near Q_{t-1} by directly projecting the previous set of particles $\tilde{Q}_t = Q_{t-1}$ onto the contact manifold. This method tightly focuses samples on the portions of the manifold that will be assigned high importance weights. However, this comes with two downsides: (1) Q_t will have a non-uniform distribution (2) the set of particles may have size $|Q_t| < n$ if projecting any particles fails.

3) Ball Projection: We can combine the advantages of both approaches by uniformly sampling particles $\tilde{\mathbf{q}}_t \sim \operatorname{uniform} R(Q_{t-1})$ from the region of configuration space $R(Q_{t-1})$ near the previous set of particles Q_{t-1}. We define the region

$$R(Q_{t-1}) = \bigcup_{\mathbf{q}^{[i]} \in Q_{t-1}} B(\mathbf{q}^{[i]}, r_\epsilon)$$

where $B(\mathbf{q}^{[i]}, r_\epsilon) = \{\mathbf{q} \in Q : ||\mathbf{q} - \mathbf{q}^{[i]}|| < r_\epsilon\}$ is a ball centered at $\mathbf{q}^{[i]}$ with radius r_ϵ. The set $R(Q_{t-1})$ is the union of all such balls centered at the particles in Q_{t-1}.

This approach is equivalent to particle projection as $r_\epsilon \to 0$. and equivalent to uniform projection as $r_\epsilon \to \infty$. We select the radius r_ϵ such that the transition model has probability $\Pr(\mathbf{q}' \notin R(Q_{t-1})) < \epsilon$ of generating a successor state $\mathbf{q}' \sim p(\mathbf{q}'|\mathbf{q}, \mathbf{u})$ outside of $R(Q_{t-1})$ given any $\mathbf{q} \in Q_{t-1}$.

V. Simulation Experiments

We evaluate the CPF and MPF with $k = 250$ on three different simulated manipulators interacting with known, static environments. Each trial begins by choosing an initial proprioception offset $\Delta\mathbf{q} \sim \mathcal{N}(0, \Sigma_{\Delta\mathbf{q}})$ and executing a predefined sequence of actions. We model motion of the arm using a frictionless contact model that projects the robot out of inter-penetration. After each time step, we simulate an observation with noise $\epsilon_{\text{a}} = \operatorname{uniform} B(0, r_a)$.

We evaluate the efficacy of each algorithm by computing the weighted root-mean square error (W-RMSE) of its particles relative to the ground truth configuration.

A. Two-Dimensional 2-DOF Arm

First, we consider a two degree-of-freedom (2-DOF) manipulator in a two-dimensional (2-D) workspace that contains a point obstacle. The robot has one binary contact sensor on its tip. We set $\Sigma_{\Delta\mathbf{q}}^{1/2} = (2 \text{ rad})I$ and $r_a = 0.05$ rad.

The contact manifold consists of two points in configuration space corresponding to "elbow up" and "elbow down" configurations. Our simulation results confirm this: the robot was able to reduce most of its uncertainty in two touches. We use this fact to implement an explicit representation of the contact manifold using an analytic inverse kinematics solution.

Fig. 4a shows that the CPF performed poorly because particle deprivation caused its state estimate to collapse to one, possibly incorrect, mode. All MPF variants outperform the CPF, but perform similarly: the contact manifold is small, so uniform samples are sufficient to densely cover it.

B. Two-Dimensional 3-DOF Arm

Next, we consider a 3-DOF manipulator in a 2-D workspace that contains an unstructured obstacle. The robot has 20 binary contact sensors spaced evenly along its distal links. We set $\Sigma_{\Delta\mathbf{q}}^{1/2} = (0.8 \text{ rad})I$ and $r_a = 0.05$ rad.

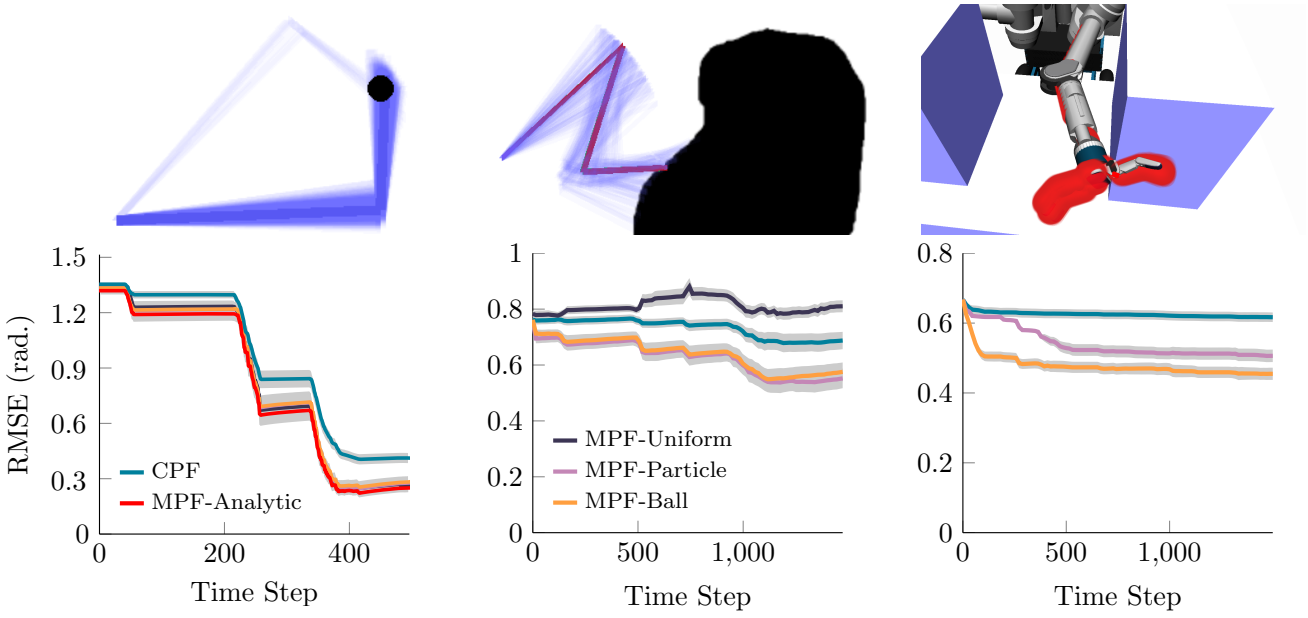
The results show that MPF-Particle and MPF-Ball perform better than CPF, but MPF-Uniform performs worse. MPF-Uniform suffers from particle deprivation because the contact manifold is too large to cover densely with 250 particles. We cannot implement MPF-Explicit in this domain because of the complex shape of the obstacle.

Surprisingly, the results show that MPF-Particle slightly outperforms MPF-Ball. We hypothesize that occurs because the robot maintains long periods of persistent contact with the obstacle. Prior work has shown that the kernel density estimation step of the MPF introduces additional variance into the belief state during persistent contact. This is partially masked by the local optimization performed by MPF-Particle.

C. Three-dimensional 7-DOF Arm

Finally, we consider a 7-DOF Barrett WAM arm with a BarrettHand end-effector in a 3-D workspace that contains boxes. The robot has binary contact sensors on its hand, wrist, and forearm. We set $\Sigma_{\Delta\mathbf{q}}^{1/2} = (0.5 \text{ rad})I$ and $r_a = 0.01$ rad.

The results show that MPF-Ball and MPF-Particle both outperform CPF. MPF-Ball outperforms MPF-Particle because the bias introduced by direct projection prevents the



(a) 2-DOF Manipulator (b) 3-DOF Manipulator (c) 7-DOF Manipuator

Fig. 4: Performance of the MPF using different projection strategies on a 2-DOF, b 3-DOF, and c 7-DOF manipulators. Results are averaged over 100 experiments. All three plots show RMSE and the error bars denote a 95% confidence interval. Best viewed in color.

filter from finding a good solution. MPF-Uniform is omitted from the plot because its large error would distort the scale.

All filters run in real-time (Table I), with most of the time spent computing the observation model (resampling and projection to the contact manifold). MPF-Uniform is particularly slow because it must frequently reject samples that fail to project to the manifold.

VI. Real-Robot Experiments

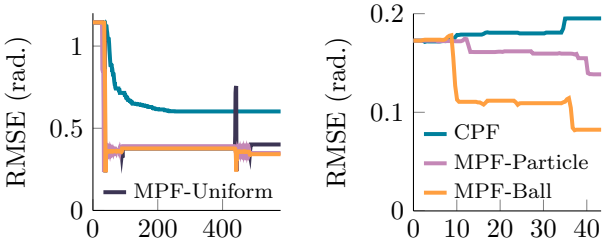
We implemented the CPF and MPF on HERB [33] to estimate the configuration of a BarrettHand [2] end-effector and the Barrett WAM arm [1]. In both cases, HERB used the BarrettHand's strain gauges to detect binary contact with the distal links of the fingers. We measured the actual configuration of the robot using optical joint encoders. These measurements were not available to the estimator. We specifically chose these tasks because ground truth data was available.

In the first experiment (Fig. 5a), we held the arm in a fixed configuration and closed the end-effector around an object. We assumed that the proximal joint angles were known, but the distal joint angles were not. This was a simple estimation problem because each finger was an independent kinematic chain. Fig. 5a shows data from two grasps: one from $t \approx 40$ to $t \approx 100$ and another from $t \approx 450$ to $t \approx 500$. Our results are consistent with the 2-DOF simulation results: all variants of the MPF outperformed the CPF.

In the second experiment (Fig. 5b), we kept the hand open and considered the configuration of the arm to be uncertain. We teleoperated HERB to execute a trajectory in an environment that contained a large box on a table. We used the joint angles measured by encoders located before the cable drive transmission for proprioception. We

Algorithm	Total	Transition	Observation
CPF	4 ± 1 ms	3 ± 1 ms	1 ± 1 ms
MPF-Particle	9 ± 7 ms	2 ± 1 ms	8 ± 7 ms
MPF-Ball	10 ± 7 ms	2 ± 1 ms	8 ± 8 ms
MPF-Uniform	35 ± 12 ms	2 ± 1 ms	34 ± 12 ms

TABLE I: Timing data for the 7-DOF simulation (Fig. 4c) along with a 95% confidence interval. The total time is broken into its component transition and observation model updates. Only iterations of the filter where the robot was in contact are recorded. Times were recorded on a consumer laptop with an Intel core i7 CPU.



(a) Underactuated Hand (b) Noisy Manipulator

Fig. 5: Experiments with real-robot data (Fig. 1) with 250 particles. One run is shown with the mean error and standard deviation of the particle distribution for each filter.

ran the CPF and MPF with $k = 100$, $\Sigma_{\Delta \mathbf{q}} = (0.1 \text{ rad})I$, and $r_a = 0.1$ rad. Fig. 5b supports our simulation results: the MPF achieved lower error than the CPF and MPF-Ball performed best.

VII. Discussion and Future Work

We have shown how the MPF can be extended to high dimensional state spaces by using constraint projection to

draw samples from an implicit representation of the contact manifold. Our simulation and real-robot results show that this approach outperforms the conventional particle filter in several scenarios. This approach can be used to compensate for proprioceptive error in a manipulator or to estimate the configuration of an under-actuated robotic hand.

However, our approach has a key limitation: it requires a known, static environment. We could relax this requirement by incorporating the pose of movable objects in the environment—including the base pose of the robot—into the estimator's state space. This is challenging because: (1) it is no longer possible to pre-compute a signed distance field and (2) the behavior of the projection depends on the parameterization of this configuration state space. We plan to explore these challenges in future work.

Finally, even though an implicit representation of the contact manifold allows us to sample from it efficiently, the samples we draw are biased by the fact that uniform samples of the ambient space will not project uniformly onto the contact manifold for two reasons: (1) because the measures of the full state space and the contact manifold may differ (a problem that even an explicit representation of the contact manifold suffers from) and (2) because projecting samples from the full state space to the contact manifold introduces bias. Our experiments have so far not made clear to what extent this bias degrades performance of the filter. One way of eliminating bias might be by rejecting and resampling particles that fail to meet some criterion of uniformity (*e.g.* Poisson disc sampling). An alternative solution might be sampling from the tangent space of the manifold rather than projecting to it [34], [35].

REFERENCES

[1] K. Salisbury, W. Townsend, B. Eberman, and D. DiPietro, "Preliminary design of a whole-arm manipulation system (WAMS)," in *IEEE ICRA*, 1988. I, 1, VI

[2] W. Townsend, "The BarrettHand grasper–programmably flexible part handling and assembly," *Industrial Robot: An International Journal*, vol. 27, no. 3, pp. 181–188, 2000. I, 1, VI

[3] B. Boots, A. Byravan, and D. Fox, "Learning predictive models of a depth camera & manipulator from raw execution traces," in *IEEE ICRA*, 2014. I

[4] S. Simunovic, "An information approach to parts mating," Ph.D. dissertation, MIT, 1979. I

[5] R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, 1960. I, III-B

[6] S. Julier and J. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *International Symposium on Aerospace/Defense Sensing, Simulation, and Controls*, 1997. I, III-B

[7] N. Gordon, S. D.J., and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," in *IEE-F*, 1993. I, III-C

[8] M. Koval, N. Pollard, and S. Srinivasa, "Pose estimation for planar contact manipulation with manifold particle filters," *IJRR*, vol. 34, no. 7, pp. 922–945, 2015. I, II, III-A2, III-D, III-E

[9] A. Petrovskaya and O. Khatib, "Global localization of objects via touch," *IEEE T-RO*, vol. 27, no. 3, pp. 569–585, 2011. II

[10] S. Javdani, M. Klingensmith, J. Bagnell, N. Pollard, and S. Srinivasa, "Efficient touch based localization through submodularity," in *IEEE ICRA*, 2013. II

[11] P. Hebert, T. Howard, N. Hudson, J. Ma, and J. Burdick, "The next best touch for model-based localization," in *IEEE ICRA*, 2013. II

[12] R. Smallwood and E. Sondik, "The optimal control of partially observable Markov processes over a finite horizon," *Operations Research*, vol. 21, no. 5, pp. 1071–1088, 1973. II

[13] K. Hsiao, L. Kaelbling, and T. Lozano-Pèrez, "Grasping POMDPs," in *IEEE ICRA*, 2007. II

[14] M. Horowitz and J. Burdick, "Interactive non-prehensile manipulation for grasping via POMDPs," in *IEEE ICRA*, 2013. II

[15] M. Koval, N. Pollard, and S. Srinivasa, "Pre- and post-contact policy decomposition for planar contact manipulation under uncertainty," in *R:SS*, 2014. II

[16] L. Zhang and J. Trinkle, "The application of particle filtering to grasping acquisition with visual occlusion and tactile sensing," in *IEEE ICRA*, 2012. II

[17] L. Zhang, S. Lyu, and J. Trinkle, "A dynamic Bayesian approach to simultaneous estimation and filtering in grasp acquisition," in *IEEE ICRA*, 2013. II

[18] S. Pellegrini, K. Schindler, and D. Nardi, "A generalisation of the ICP algorithm for articulated bodies," in *BMVC*, vol. 3, 2008, p. 4. II

[19] M. Krainin, P. Henry, X. Ren, and D. Fox, "Manipulator and object tracking for in-hand 3D object modeling," *IJRR*, vol. 30, no. 11, pp. 1311–1327, 2011. II

[20] M. Klingensmith, T. Galluzzo, C. Dellin, M. Kazemi, J. Bagnell, and N. Pollard, "Closed-loop servoing using real-time markerless arm tracking," in *IEEE ICRA Humanoids Workshop*, 2013. II

[21] T. Schmidt, R. Newcombe, and D. Fox, "DART: Dense articulated real-time tracking," in *R:SS*, 2014. II

[22] T. Schmidt, K. Hertkorn, R. Newcombe, Z. Marton, M. Suppa, and D. Fox, "Depth-based tracking with physical constraints for robot manipulation," in *IEEE ICRA*, 2015. II

[23] M. R. Dogar, V. Hemrajani, D. Leeds, B. Kane, and S. Srinivasa, "Proprioceptive localization for mobile manipulators," Robotics Institute, CMU, Tech. Rep. CMU-RI-TR-10-05, 2010. II

[24] P. V. Sachin Chitta and, R. Geykhman, and D. D. Lee, "Proprioceptive localization for a quadrupedal robot on known terrain," in *IEEE ICRA*, 2007. II

[25] P. G. Kry and D. K. Pai, "Interaction capture and synthesis," *ACM Transactions on Graphics*, vol. 25, no. 3, 2006. II

[26] S.-J. Chung, J. Kim, S. Han, and N. S. Pollard, "Quadratic encoding for hand pose reconstruction from multi-touch input," in *Eurographics Short Paper*, 2015. II

[27] S. Ha, Y. Bai, and C. K. Liu, "Human motion reconstruction from force sensors," in *ACM SIGGRAPH/Eurographics*, 2011. II

[28] A. Roncone, M. Hoffmann, U. Pattacini, and G. Metta, "Automatic kinematic chain calibration using artificial skin: self-touch in the icub humanoid robot," in *IEEE ICRA*. IEEE, 2014, pp. 2305–2312. II

[29] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *The Annals of Mathematical Statistics*, vol. 27, no. 3, pp. 832–837, 1956. III-E

[30] "Efficient global penetration depth computation for articulated models," *Computer-Aided Design*, vol. 70, pp. 116 – 125, 2016. IV-C

[31] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," Cornell Computing and Information Science, Tech. Rep., 2004. IV-D

[32] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "CHOMP: Gradient optimization techniques for efficient motion planning," in *IEEE ICRA*, May 2009, pp. 489–494. IV-D

[33] S. Srinivasa, D. Berenson, M. Cakmak, A. Collet, M. Dogar, A. Dragan, R. Knepper, T. Niemueller, K. Strabala, and M. Vande Weghe, "HERB 2.0: Lessons learned from developing a mobile manipulator for the home," *Proceedings of the IEEE*, vol. 100, no. 8, pp. 1–19, 2012. VI

[34] H. Snoussi, "Particle filtering on riemannian manifolds. application to covariance matrices tracking," in *Matrix Information Geometry*, 2013, pp. 427–449. VII

[35] J. Brookshire and S. Teller, "Articulated pose estimation using tangent space approximations," *IJRR*, vol. 35, no. 1–3, 2016. VII