

Manifold Representations for State Estimation in Contact Manipulation

Michael C. Koval, Nancy S. Pollard, and Siddhartha S. Srinivasa

Abstract We investigate the problem of using contact sensors to estimate the configuration of an object during manipulation. Contact sensing is very discriminative by nature and, therefore, the set of object configurations that a sensor constitutes a lower-dimensional manifold in the state space of the object. This causes conventional state estimation methods, such as particle filters, to perform poorly during periods of contact. The *manifold particle filter* addresses this problem by sampling particles directly from the contact manifold. When it exists, we can sample these particles from an *analytic representation* of the contact manifold. We present two alternative sample-based contact manifold representations that make no assumptions about the object-hand geometry: *rejection sampling* and *trajectory rollouts*. The rejection sampling representation distributes uniformly in the space surrounding the manifold, while the trajectory rollout representation concentrates samples on the regions of the manifold that we are most likely to encounter during execution. We discuss theoretical considerations behind these three representations and compare their performance in an extensive suite of simulation experiments. We show that all three representations enable the manifold particle filter to outperform the conventional particle filter. Additionally, we show that the trajectory rollout representation performs similarly to the analytic method despite its relative simplicity.

1 Introduction

Humans effortlessly use their sense of touch to manipulate objects. Imagine groping around on a nightstand for a glass of water, or feeling around a cluttered kitchen cabinet while searching for the salt shaker. Each of these tasks involves *contact manipulation* where we make *persistent contact* with the environment. Our contact observations are critical for localizing the object as we manipulate it.

Michael C. Koval, Nancy S. Pollard, Siddhartha S. Srinivasa
Robotics Institute, Carnegie Mellon University
e-mail: {mkoval, nsp, siddh}@cs.cmu.edu

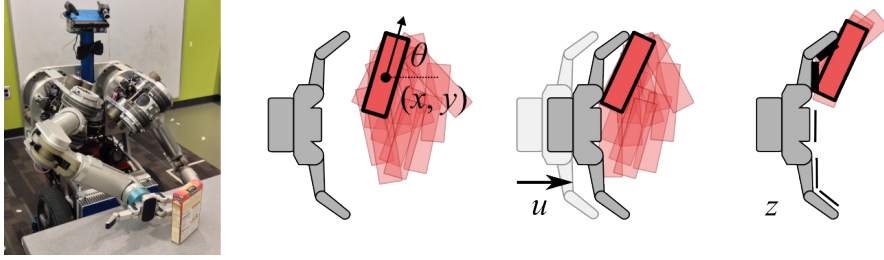


Fig. 1: HERB is pushing a rectangular box across the table. The state $x \in X$ is the pose of the box relative to the hand. An action $u \in U$ is a relative motion of the hand. After taking action u , HERB receives an observation $z \in Z$ indicating where the object touched the hand.

Armed with real-time observations from tactile sensors [Odhner et al., 2013, Tenzer et al., 2012, Fishel & Loeb, 2012], manipulators should be also able to estimate the *state* of the manipulated object—a problem we formalize as the *state estimation for contact manipulation* problem (Section 2).

Early work attempted to solve this problem by deriving analytical state estimators and controllers to track and control the pose of an object from contact positions based on simple models of physics [Jia & Erdmann, 1999]. These models fail to accurately capture the reality of manipulation: there is a large amount of uncertainty in both the object’s motion and the robot’s observations.

Other work has employed a Bayesian approach by using the particle filter to estimate the pose [Zhang & Trinkle, 2012] and physical properties [Zhang et al., 2013] of an object during manipulation. However, our prior work [Koval et al., 2013] revealed that the conventional particle filter (CPF) performs poorly at real-time update rates and suffers from a startling problem: *the CPF systematically performs worse as sensor resolution increases* (Section 3).

This problem arises because contact sensing accurately *discriminates* between contact and no-contact. Topologically, the set of object configurations that agree with a contact observation lies in the lower dimensional *observable contact manifold* embedded in the configuration space of the object (Section 2). Particles sampled from the state space during contact have low probability of falling into the observation space. This results in *particle starvation* in the vicinity of the true state. The *manifold particle filter* (MPF) provides a principled way of solving this problem by sampling particles directly from the observable contact manifold (Section 4).

Applying the MPF to the state estimation for contact manipulation problem requires sampling particles from the observable contact manifold. When it exists, an analytic representation (AM) of the manifold provides an exact and computationally efficient way of sampling from the dual proposal distribution (Section 5). However, computing an analytic representation of the contact manifold is not always possible.

We present two alternative sample-based contact manifold representations that make no assumptions about the object-hand geometry: rejection sampling (RS) and trajectory rollouts (TR). The RS representation distributes samples uniformly in the space surrounding the manifold, while the TR representation concentrates many

samples on the regions of the manifold that we are most likely to encounter during execution (Section 6).

Our results (Section 6) reveal the tradeoffs between these representations. RS performs the worst. By distributing samples uniformly everywhere, even in unlikely regions, RS is relatively sparse everywhere. Surprisingly, TR performed nearly as well as AM. By focusing samples on likely regions, TR was able to saturate those regions at a resolution that was indistinguishable from AM. A key reason is that likely regions occupy only a small portion of the contact manifold in our experiments, where the hand pushes straight towards the object.

Our key takeaway is to exploit structure. By exploiting the manifold structure of the contact state estimation problem, we are able to outperform the CPF. Furthermore, by exploiting the geometry of the hand-object interaction with trajectory rollouts, we are able to perform as well as the analytical method.

We are excited by our future directions. First, in generalizing state to material properties, and even shape, enabling us to simultaneously estimate shape and pose from contact. Second, in closing the loop between state estimation and control, to develop robust closed-loop policies for contact manipulation.

2 State Estimation for Contact Manipulation

Let $x \in X$ be the state of a dynamical system which evolves over time under actions $u \in U$ and produces observations $z \in Z$. The *state estimation problem* addresses the computation of the *belief state*, a probability distribution over the current state x_t given the past history of actions $u_{1:t}$ and observations $z_{1:t}$:

$$b(x_t) = p(x_t | z_{1:t}, u_{1:t}) \quad (1)$$

We focus on the *state estimation for contact manipulation problem*, where the state is the pose x (Fig. 1-Left) of the manipulated object and an action u (Fig. 1-Middle) is a relative motion of the hand. During contact the object moves according to the stochastic *transition model* $p(x_t | x_{t-1}, u_t)$ that encodes the physics of the object’s motion in response to pushing action u_t . The stochasticity of the transition model may be due to unknown physical properties of the object (e.g. friction coefficients), imperfections in the physics simulation, or error in executing u_t .

Contact sensors attached to the surface of the hand provide observations z_t (Fig. 1-Right) that indicate whether the object is touching the sensor. This is equivalent to testing whether $x_t \in X_o$, where $X_o \subseteq X$ is the set of states where the object is in contact with one or more sensors. While $x \in X_o$, z_t may provide additional information about the configuration of the object through noisy measurement of its contact with the hand. Both of these properties are combined into the stochastic *observation model* $p(z_t | x_t, u_t)$ as the probability of state x_t generating observation z_t after executing action u_t .

The Contact Manifold. The key challenge for state estimation arises because state evolves on a lower-dimensional contact manifold. We can partition X into three parts depending upon the type of contact occurring between the hand and the object: (1) penetrating contact X_{pen} , (2) non-penetrating contact X_c , and (3) no contact X_{free} .

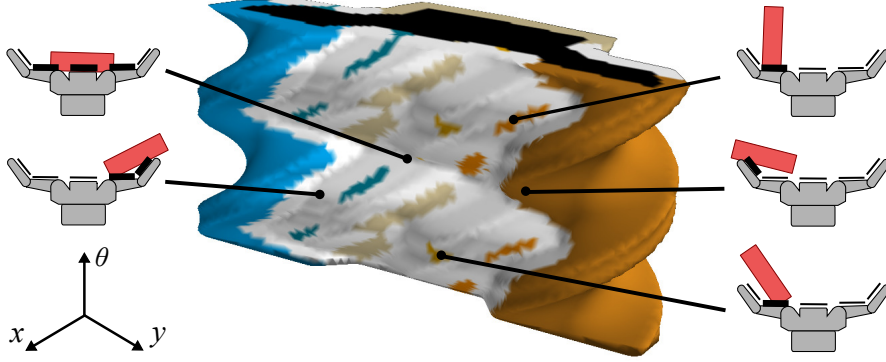


Fig. 2: Contact manifold X_c for a two-dimensional BarrettHand pushing a rectangular box. Each point $x \in X_c$ corresponds to a configuration of the object $x \in X_c$ that is in non-penetrating contact with the hand and is uniquely colored by the active contact sensors. Configurations that are in contact with multiple sensors are colored white.

These three sets are defined by the interplay between the geometry of the object and the geometry of the hand.

Let $P_h \subseteq \mathbb{R}^n$ be the geometry of the hand and $P_o(x) \subseteq \mathbb{R}^n$ be the geometry of the object at configuration $x \in X$. The set of all object poses that are in collision with the hand form the *configuration space obstacle* [Lozano-Pérez, 1983]

$$X_{obs} = CO_o(P_h) = \{x \in X : P_h \cap P_o(x) \neq \emptyset\}$$

of the hand in the object's configuration space.

Any configuration in $X_{pen} = \text{int}(X_{obs})$ is invalid because the object penetrates the hand. Conversely, any configurations in $X_{free} = X \setminus X_{obs}$ is in *free space* where the object is out of contact with the hand. Therefore, any valid object configuration that is in contact with the hand must lie on the *contact manifold* $X_c = X_{obs} \setminus \text{int}(X_{obs})$.

Figure 2 shows the contact manifold for the BarrettHand pushing an elongated rectangular box in $X = SE(2)$. Topologically, the contact manifold is a torus in $SE(2)$ with the top and bottom edges of the θ -dimension identified. The manifold is repeated twice along the θ -axis due to symmetry.

The Observable Contact Manifold. We know that $x \in X_c$ during periods of contact. However, our contact sensors may not be able to observe contact over the entire surface of the hand. We will define the *observable contact manifold* $X_o \subseteq X_c$ as the set of object poses that are capable of generating contact observations $z \in Z_c$.

Let $P_s \subseteq P_o \setminus \text{int}(P_o)$ denote the surface of the hand that is instrumented with contact sensors. The set of observable states X_s that could generate a contact observation is given by the configuration space obstacle

$$X_s = CO_o(P_s) = \{x \in X : P_s \cap P_o(x) \neq \emptyset\}$$

of the sensor in the object’s configuration space. The *observable contact manifold* $X_o = X_s \cap X_c$ consists of the set of valid object configurations that have high probability of generating a contact observation $z \in Z_c$.

Figure 2 shows the contact manifold colored by which sensors are active at each point. Any state in the large, dark orange region of the manifold are in contact with—and, thus, will stimulate, the left distal contact sensor. States in the large white region of the manifold are simultaneously in contact with multiple sensors.

Discriminative Observation Model. Contact sensors accurately discriminate between contact and no-contact. We call an observation model *discriminative* if we can partition the set of observations Z into sets of contact $Z_c \subseteq Z$ and no-contact $Z_{nc} = Z \setminus Z_c$ observations such that there are few false-positive and false-negative indications of contact. Therefore, a discriminative observation model satisfies $\Pr(z \in Z_c | x \in X_o, u) > 1 - \epsilon$ during periods of contact and $\Pr(z \in Z_{nc} | x \notin X_o, u) > 1 - \epsilon$ during no-contact.

3 Conventional Particle Filter

The *Bayes filter* is the most general algorithm for filtering a belief state given a sequence of actions and observations by recursively constructs $b(x_t)$ from $b(x_{t-1})$ using the update rule

$$b(x_t) = \eta p(z_t | x_t, u_t) \int_X p(x_t | x_{t-1}, u_t) b(x_{t-1}) dx_{t-1} \quad (2)$$

where η is a normalization factor. The terms $p(z_t | x_t, u_t)$ and $p(x_t | x_{t-1}, u_t)$ are, respectively, the observation and transition models. The recursion is initialized with a prior belief $b(x_0)$ provided by task-specific knowledge or other sensors (e.g. an object recognition system).

The *particle filter* [Thrun et al., 2005] is a non-parametric formulation of the Bayes filter that represents the belief state $b(x_t)$ with a discrete set of samples. The samples $X_t = \{x_t^{[i]}\}_{i=1}^n$ are called *particles* and are distributed according to the belief state $x_t^{[i]} \sim b(x_t)$. The particle filter implements the Bayesian update (Eq. 2) by recursively constructing X_t from X_{t-1} using a technique called *importance sampling*.

The conventional particle filter (CPF) is summarized in algorithm 1. The key insight behind this realization is that it is difficult to directly sample from the *target distribution* (Eq. 2), but it is relatively easy to sample from the transition model. Therefore, we sample $x_t^{[i]}$ from the *proposal distribution* $\int_X p(x_t | x_{t-1}, u_t) b(x_{t-1}) dx_{t-1}$ (line 3) by forward-simulating X_{t-1} to X_t using the motion model. Next, we compute an *importance weight* $w_t^{[i]} = p(z_t | x_t, u_t)$ for each forward-simulated particle (line 4).

The importance weights result from dividing the target distribution by the proposal distribution. As a result, the samples $x_t^{[i]}$, along with their importance weights $w_t^{[i]}$, are distributed according to the target distribution $b(x_t)$. Intuitively, the weighting step incorporates the observation model into the update by assigning higher weight to particles that are consistent with z_t .

Algorithm 1 Conventional Particle Filter

Require: X_{t-1} , particles from time $t - 1$
Require: $p(x_t|x_{t-1}, u_t)$, transition model
Require: $p(z_t|x_t, u_t)$, observation model
Ensure: X_t , particles sampled from $b(x_t)$
1: $X_t \leftarrow \emptyset$
2: **for all** $x_{t-1}^{[i]} \in X_{t-1}$ **do**
3: Sample $x_t^{[i]} \sim p(x_t|x_{t-1}^{[i]}, u_t)$
4: $w_t^{[i]} \leftarrow p(z_t|x_t^{[i]}, u_t)$
5: $X_t \leftarrow \{x_t^{[i]}\} \cup X_t$
6: $X_t \leftarrow \text{Resample}(X_t)$

Periodically the particle filter *resamples* the set of weighted particles (line 6) with replacement to distribute X_t according to the desired posterior $b(x_t)$ with unit weights. Frequent resampling is necessary to prevent the weights from growing unbounded and degenerating over time [Thrun et al., 2005].

Particle Starvation During Contact. The particle filter, as described above, is agnostic to the observation model and has been applied to a variety of application domains [Montemerlo et al., 2003, Zhang & Trinkle, 2012]. However, contact sensors are unique because they operate in two discrete states: contact and no contact. When $z \in Z_c$, the belief state has a singular component that is concentrated on the lower-dimensional *observable contact manifold*. Conversely, when $z \in Z_{nc}$, $p(z_t|x_t, u_t)$ is uniform over free space and provides little useful information. This property makes contact sensors fundamentally different than cameras and depth sensors, which have relatively smooth observation models.

In practice, particle filters are updated in discrete steps. The execution of an action concentrates any states that penetrate the hand onto the contact manifold. As a result, the hand’s contact sensors gain full dimensionality and the CPF is not completely ineffective at estimating the state. However, the CPF requires a large number of particles to increase the probability that some fall into the small swept volume of each sensor. As a result, the CPF suffers from particle deprivation and may result in there being no particles in the vicinity of the true state during periods of contact.

Figure 3-Top shows an effect that particle starvation has on the post-contact performance of the CPF. The conventional particle filter correctly filters the belief state before contact in (a–b). However, after contact occurs, $b(x_t)$ becomes singular and importance sampling fails. As a result, the CPF converges to an erroneous belief that has the box rolling off of the finger tip instead of settling into the palm.

Surprisingly, this effect causes the particle filter to *perform worse as sensor resolution or the update rate increases*. As sensor resolution increases, the swept volume of each sensor becomes narrower. As the update rate increases, the distance traveled by the hand between updates decreases, and the swept volume becomes shorter. As a result, the CPF requires a large number of particles to successfully track the state.

We have shown that the conventional particle filter is poorly suited for the state estimation for contact manipulation problem because the state evolves on a lower-dimensional manifold.

Algorithm 2 Manifold Particle Filter

Require: X_{t-1} , particles from time $t - 1$
Require: k , number of dual particles to sample
Ensure: $\bar{X}_t$, particles sampled from $b(x_t)$

- 1: $\bar{X}_{M_i} \leftarrow \emptyset$ for $i = 1, \dots, m$
- 2: **for** $1, \dots, k$ **do**
- 3: Sample $M_i \sim \Pr(x_t \in M_i)$
- 4: **if** $M_i \neq M_m$ **then**
- 5: Sample $\bar{x}_t^{[i]} \sim p(z_t | M_i, x_t, u_t) / p(z_t | M_i, u_t)$
- 6: $\bar{w}_t^{[i]} \leftarrow \text{EstimateDensity}(X_{t-1}, u_t, \bar{x}_t^{[i]})$
- 7: $\bar{X}_{M_i} \leftarrow \{\bar{x}_t^{[i]}\} \cup \bar{X}_{M_i}$
- 8: **end if**
- 9: $\bar{X}_{M_m} \leftarrow \text{ConventionalProposal}(X_{t-1}, u_t, z_t) \cap M_m$
- 10: $\bar{X}_t \leftarrow \text{Resample}(\sum_{i=1}^m \Pr(x_t \in M_i) \bar{X}_{M_i})$

4 Manifold Particle Filter

Suppose the state space X is partitioned into m disjoint components $M = \{M_i\}_{i=1}^m$, where $M_1, \dots, M_{m-1} \subseteq X$ are manifolds and $M_m = X \setminus \cup_{i=1}^{m-1} M_i$ is the remaining free space. The belief state $b(x)$ may have a singular component with non-zero probability concentrated on the lower-dimensional manifolds $\{M_i\}_{i=1}^{m-1}$.

We redefine the belief state as the weighted sum

$$b(x_t) = \sum_{M_i \in M} b(x_t | M_i) \Pr(x_t \in M_i) \quad (3)$$

over manifolds, where $b(x_t | M_i)$ is the belief over M_i given that $x_t \in M_i$.¹

The manifold particle filter (MPF), summarized in algorithm 2, also represents the belief using particles. For each particle, we first choose which manifold to sample from according to $M_i \sim \Pr(x_t \in M_i)$. Then, we sample the particle $\bar{x}_t^{[i]} \sim b(x_t | M_i)$ from the corresponding conditional belief using a sampling technique that is appropriate for the structure of M_i .

Ideally, we could compute $\Pr(x_t \in M_i)$ by marginalizing over M_i . Unfortunately, this is fundamentally impossible for two reasons. First, marginalizing requires $b(x_t)$, precisely the distribution that we are trying to estimate. Second, $\int_{M_i} b(x_t) dx_t = 0$ because M_i is a measure zero set.

Instead, we approximate $\Pr(x_t \in M_i)$ using only the most recent observation

$$\Pr(x_t \in M_i) \approx \frac{p(z_t | M_i, u_t)}{p(z_t | u_t)} \quad (4)$$

where $p(z_t | M_i, u_t)$ is the probability that z_t was generated by an $x_i \in M_i$ and $p(z_t | u_t) = \int_X p(z_t | x_t, u_t) dx_t$ is the prior probability of receiving observation z_t . However, Eq. 4 is a good approximation in the case where $p(z_t | x_t, u_t)$ accurately discriminates between manifolds.

¹ From this point forward we will use $b(x_t)$ as shorthand for the weighted sum in Eq. 3.

Finally, we can sample a particle $\bar{x}_t^{[i]}$ according to the belief distribution over the chosen manifold $b(x_t|M_i)$. Our key insight is that we can apply a different sampling technique for each M_i that is specifically designed to take advantage of the structure of the manifold. For the manifolds $\{M_i\}_{i=1}^{m-1}$, we sample from the dual proposal distribution [Thrun et al., 2000] as described below. In the case of the free space M_m , we sample $\bar{x}_t^{[i]}$ with the conventional technique and reject any $\bar{x}_t^{[i]} \in \cup_{i=1}^{m-1} M_i$. This rejection sampling step is necessary to avoid biasing the estimate of $b(x_t)$ towards the manifolds.

Dual Proposal Distribution. Importance sampling from the conventional proposal distribution fails on M_i for $i < m$ because they are lower-dimensional manifolds. In this case, we will sample from the *dual proposal distribution* [Thrun et al., 2000]

$$\bar{x}_t^{[i]} \sim \eta \frac{p(z_t|M_i, x_t, u_t)}{p(z_t|M_i, u_t)}, \quad (5)$$

where η is a normalization constant. We can find the corresponding importance weights

$$\bar{w}_t^{[i]} = \int_{M_i} p(\bar{x}_t^{[i]}|x_{t-1}, u_t) b(x_{t-1}|M_i) dx_{t-1}. \quad (6)$$

by dividing the target distribution (Eq. 2) by the proposal distribution (Eq. 5).

The conventional proposal distribution forward-predicts using the motion model and computes importance weights using the observation model. Conversely, the dual proposal distribution samples particles from the observation model and weights them by how well they agree with the motion model. [Thrun et al., 2000].

Mixture Proposal Distribution. Just as how the conventional proposal distribution performs poorly with accurate sensors, the dual proposal distribution performs poorly when there is observation noise. The MPF uses the dual proposal distribution to sample from the manifolds and, as result, shares the same weakness.

We use a *mixture proposal distribution* [Thrun et al., 2000] to mitigate this effect by combining both sampling techniques. Instead of sampling all of the particles from the MPF, we sample n particles from the CPF and d particles from the MPF. We then combine the two sets of particles with the weighted sum $(1 - \phi)X_t + \phi\bar{X}_t$ before resampling. The *mixing rate* $0 \leq \phi \leq 1$ is a parameter that allows the algorithm to smoothly transition from the CPF ($\phi = 0$) to the MPF ($\phi = 1$).

Intuitively, $d = |\bar{X}_t|$ is the number of particles necessary to simultaneously cover all of the manifolds and $n = |X_t|$ is the number of additional particles necessary to represent $b(x_t)$ in free space.

5 Manifold Particle Filter for Contact Manipulation

In this section, we will apply the MPF to the state estimation for contact manipulation problem. To do so, we will define the observable contact manifold X_o and free space X_{free} as the relevant subsets of X . We will also describe a technique for computing the importance weights $w_t^{[i]}$ using kernel density estimation.

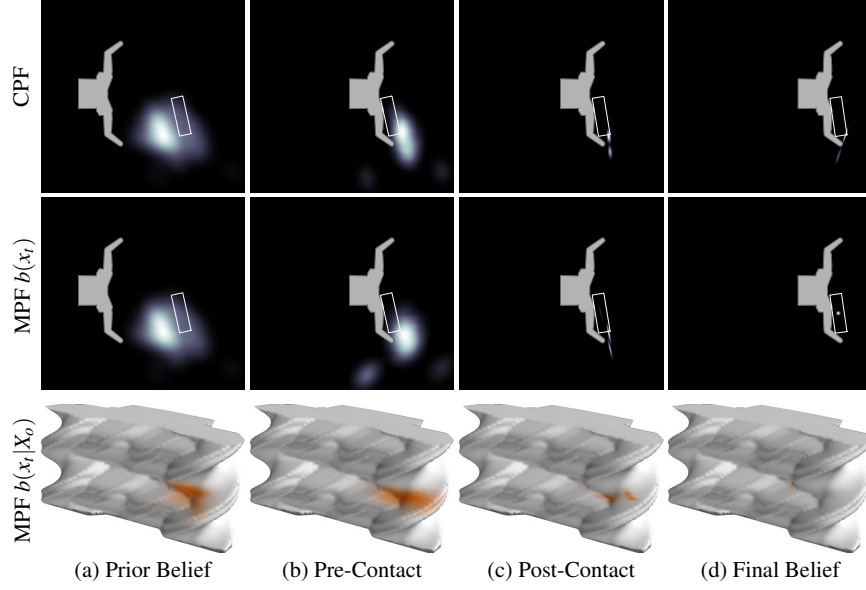


Fig. 3: Snapshots of the MPF during execution. Unlike the CPF, the MPF explicitly tracks the probability distribution on the contact manifold X_c .

Figure 3 shows the performance of the MPF relative to the CPF. Before contact (a–b), $\Pr(x_t \in X_o) \approx 0$ and both filters update using the conventional proposal distribution. After contact (c–d), $\Pr(x_t \in X_o) \approx 1$ and the manifold particle filter begins sampling from X_o . Sampling from the observable contact manifold allows the MPF to accurately track the object’s pose during persistent contact.

Importance Sampling from the Contact Manifold. We must weight the samples drawn from the dual proposal distribution with their corresponding importance weights $\bar{w}^{[i]} = \int_X p(x_t | x_{t-1}, u_t) b(x_{t-1}) dx_{t-1}$. Intuitively, this integrates our belief state $b(x_{t-1})$ prior to taking action u_t into $b(x_t)$.

We evaluate $\bar{w}^{[i]}$ by forward-simulating the previous set of particles X_{t-1} to time t by sampling from $p(x_t | x_{t-1}, u_t)$, then evaluating the density of the distribution at $\bar{x}_t^{[i]}$ using a density estimation technique [Thrun et al., 2000]. Ideally, we would compute a density estimate over the manifold X_o . Unfortunately, while there has been some work on density estimation on Riemannian manifolds [Pelletier, 2005], but it is difficult to apply these algorithms to the approximate and sample-based representations of X_o described below. This is exacerbated by the fact that many of our forward-simulated particles will not lie on X_o .

Instead, we use kernel density estimation to approximate the probability density over X , then restrict the estimate to $X_o \subset X$. Figure 3 shows an example of the resulting density estimate over X_{free} (Fig. 3-Middle) and X_o (Fig. 3-Bottom) computed using Gaussian kernels with bandwidths selected by Silverman’s rule of thumb.

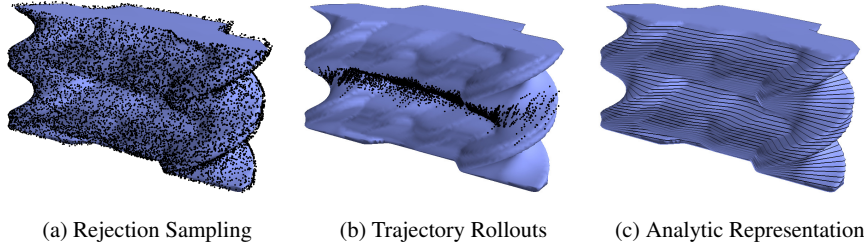


Fig. 4: Several approximate representations of the contact manifold. Representations (a) and (b) approximate X_o with discrete sets of samples. Representation (c) computes an approximate, analytic representation X_o .

6 Representing the Contact Manifold

Implementing the manifold particle filter requires sampling particles from X_o with probability equal to their observation likelihood $x_t^{[i]} \sim \eta p(z_t | x_t, u_t) / p(z_t | u_t)$. Sampling from this distribution requires maintaining a representation of the observable contact manifold X_o .

We will discuss three possible representations of the contact manifold. Two of these, the rejection sampling (Fig. 4a) and trajectory rollout (Fig. 4b) representations, approximate the continuous manifold X_o with large set of discrete samples. The third technique (Fig. 4c) takes advantage of additional structure in geometry of the problem to solve for an analytic representation of X_o .

Rejection Sampling. The most straightforward way of sampling from $X_o \subset X$ is through rejection sampling in the ambient space X . *Rejection sampling* iteratively samples candidate states $x^{[i]} \sim \text{uniform}(X)$ until it finds a sample $x^{[i]} \in X_o$ in the desired set. Using this technique, we can generate a large set of samples $\tilde{X}_o = \{x^{[i]}\}_{i=1}^n \subset X_o$ that densely cover X_o in a pre-computation step. At runtime, we importance sample from the discrete set $\tilde{X}_o$ with $w^{[i]} = p(z_t | x_t, u_t) / p(z_t | u_t)$ as importance weights.

Unfortunately, rejection sampling fails for the same reason as the conventional particle filter: X_o is a measure-zero set and there is zero probability of successfully sampling an $x^{[i]} \in X_o$. We will, instead, rejection sample from the set

$$X_{o,approx} = \{x \in X : \min_{p_s \in P_s, p_o \in P_o(x)} \|p_s - p_o\| \leq \epsilon\}$$

of object configurations that are within distance $\epsilon \in \mathbb{R}^+$ of the hand. The set $X_{o,approx}$ is a reasonable approximation for X_o when ϵ is on the same order of magnitude as the numerical inaccuracies of the motion and observation models (e.g. simulation step size).

Figure 4a shows X_o covered with a set of 10,000 rejection-sampled configurations $\tilde{X}_o$ of the BarrettHand in contact with the rectangular box shown in Fig. 2. The samples $\tilde{X}_o$ are not exactly on X_o and are distributed uniformly over the ambient space X . This is, in most cases, an acceptable approximation for a true uniform distribution over X_o .

Trajectory Rollouts. Rejection sampling attempts to densely cover all of X_o with samples $\tilde{X}_o$ that are independent of the prior belief $b(x_0)$. As a result, many of the samples generated by rejection sampling will be found in regions of X_o that remain low probability during the entire duration of execution. We can exploit this structure by concentrating more samples in the regions of X_o that we are likely to encounter during execution.

We can generate samples $\tilde{X}_o$ that are biased towards these regions by performing trajectory rollouts for a set of sampled beliefs. We begin by sampling a particle from the prior $x_0^{[i]} \sim b(x_0)$. Next, we forward-simulate the particle for T steps using the motion model $x_t^{[i]} \sim p(x_t | x_{t-1}, u_t)$ with u_t chosen according to our policy.² Finally, we add any $x_t^{[i]} \in X_o$ to $\tilde{X}_o$. This process repeats until $|\tilde{X}_o|$ reaches the desired size.

Figure 4b shows 10,000 samples taken from 3000 trajectory rollouts with a fixed “move straight” action and $b(x_0)$ roughly centered in front of the hand. The trajectory rollout technique achieves dense coverage of the reachable area of the state space, which consists of the front of the hand with orientations consistent with $b(x_0)$, at the cost of sparse coverage of the rest of the manifold.

Unfortunately, the non-uniformity of our samples means that $\tilde{X}_o$ is biased towards absorbing regions of the state space. We compensate for the bias through importance sampling: we assign each $x^{[i]} \in \tilde{X}_o$ an importance weight $w^{[i]} = p(z|x, u) / [p(z|u)\tilde{p}(x)]$ where $\tilde{p}(x)$ is the density of $\tilde{X}_o$ at x . We estimate $\tilde{p}(x)$ using a standard kernel density estimation technique and, thus, produce samples that are uniformly distributed over the ambient space X .

Analytic Representation. In some special cases of hand-object geometry we can compute an analytic representation of X_o . This is possible, for example, in the common case where P_h and P_o are polygons in $\mathbb{R}^2$ [Lozano-Pérez, 1983] or polyhedra in $\mathbb{R}^3$ [LaValle, 2006].

Without loss of generality, we will consider polygonal objects in $SE(2)$. In this case, we can geometrically compute the C-obstacle $X_{obs}(\theta)$ for a fixed orientation θ of the object as

$$X_{obs}(\theta) = P_h \oplus -P_o([0, 0, \theta])$$

where $A \oplus B = \{a + b : a \in A, b \in B\}$ denotes the Minkowski sum of sets A and B .

Since P_h and $P_o(\theta)$ are polygonal, $X_{obs}(\theta)$ is also polygonal and we can compute an analytical representation of $X_{obs}(\theta)$ with a convolution [Wein, 2013]. The contact manifold $X_c(\theta)$ at orientation θ simply consists of the perimeter of the polygon $X_{obs}(\theta)$. Figure 4c shows several θ -isocontours of X_{obs} superimposed over a high-resolution polyhedral approximation of X_o . The same process can be repeated with P_s and P_o to construct an analytic representation of $X_o(\theta)$.

We can approximate the C-obstacle $\tilde{X}_{obs} = \cup_{\theta \in \Theta} X_{obs}(\theta)$ with a union over a large, discrete set of orientations Θ .³ Discretizing θ approximates X_{obs} by a polyhedron $\tilde{X}_{obs}$ that shares the same polygonal iso-contours as X_o for all $\theta \in \Theta$.

² If the policy is not known, we sample $u \sim \text{uniform}(U)$.

³ Uniformly discretizing θ may miss critical events where the object first comes into or leaves contact with the hand. If these events are important, it is possible to analytically solve for the critical values of θ through careful analysis of the geometry [Farahat et al., 1995].

Sampling an $x^{[i]} \sim X_o$ is possible by first sampling a $\theta \in \Theta$, then uniformly sampling an $x^{[i]}$ from our analytical representation of $X_o(\theta)$. Alternatively, one could sample from an approximate, polyhedral representation of X_o by interpolating between iso-contours. In both cases, the samples are correctly drawn uniformly with respect to a measure defined over the lower-dimensional X_o .

7 Experiments and Results

We designed a set of simulation experiments to compare the MPF with the CPF for the state estimation for contact manipulation problem, and to explore the differences between the three representations of the contact manifold.

Based on the particle starvation problem, we hypothesize that

H1. *The MPF will outperform the CPF after contact.*

Among the three representations of the contact manifold, we expect the rejection sampling (RS) representation to perform the worst due to its relatively sparse distribution of samples. The trajectory rollout (TR) representation solves this problem by concentrating samples on the regions of the contact manifold that we are most likely to encounter.

Therefore, we hypothesize:

H2. *Trajectory rollouts will outperform rejection sampling.*

H3. *The analytic contact manifold will outperform rejection sampling.*

However, we hypothesize that the analytic representation will outperform both the RS and TR representations because it exactly represents the contact manifold:

H4. *The analytic contact manifold will perform best.*

7.1 Experimental Design

Setup. We implemented the CPF, MPF-RS, MPF-TR, and MPF-TR in a custom two-dimensional kinematic simulation environment with polygonal geometry. Each experiment consisted of a simulated BarrettHand pushing a rectangular box in a straight line at a speed of 1 cm/s for 50 cm. The initial belief state was set to $b(x_0) = \mathcal{N}(0, \Sigma)$ with $\Sigma^{1/2} = \text{diag}[5 \text{ cm}, 5 \text{ cm}, 20^\circ]$.

Motion Model. We simulated the motion of the object using a penetration-based quasistatic physics simulator [Lynch et al., 1992] with a 1 mm step size. During each update, the finger-object coefficient of friction μ and the radius of the object’s pressure distribution c were sampled from the Gaussian distributions $\mu \sim N(0.5, 0.2^2)$ and $c \sim N(0.05, 0.01^2)$ truncated at $\mu, c > 0$.

Observation Model. Binary observations were simulated for each of the hand’s sensors by computing the intersection of the contact sensor with the object’s geometry. Ground-truth observations were simulated by applying the same observation model to a special “ground truth” particle sampled from $b(x_0)$.

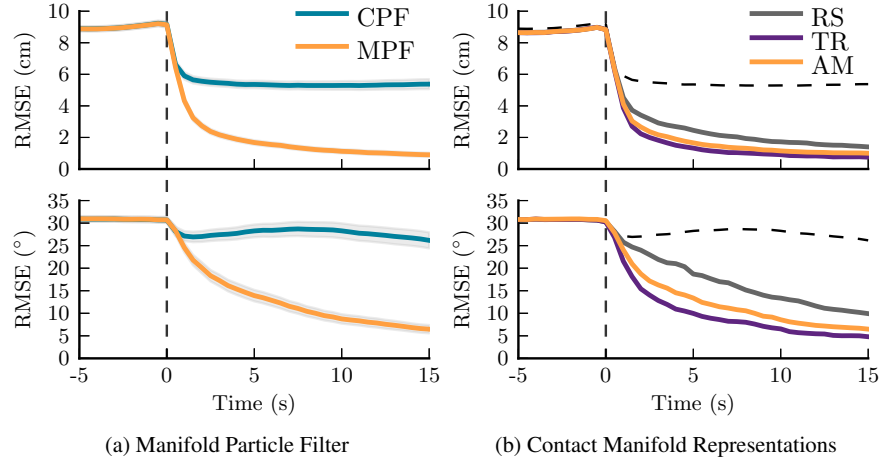


Fig. 5: (a) Estimation error of the CPF and MPF. (b) Performance of MPF using the rejection-sampled (RS), trajectory-rollout (TR), and analytical (AM) manifold representations. In both cases, the data is aligned such that contact occurs at $t = 0$.

Dependent Measure. We measure performance of the estimators by tracking the root mean square error (RMSE) of the object’s position (Fig. 5a-Top) and orientation (Fig. 5a-Bottom) over a large number of experiments

7.2 Conventional vs. Manifold Particle Filter (H1)

Both the CPF and the MPF used 100 particles. The MPF used an analytical representation of the contact manifold and a mixing rate of $\phi = 0.1$

As expected, both filters behave similarly before contact ($t \leq 0$) and there not a significant difference in RMSE. After contact ($t > 0$), the MPF quickly achieves 4.4 cm less RMSE than the CPF. These results support hypothesis H1: the MPF achieves lower post-contact error than the CPF.

7.3 Contact Manifold Representation (H2–H4)

We also compared the RMSE error of the MPF using the rejection sampling (RS), trajectory rollouts (TR), and an analytic (AM) representations of the contact manifold. The RS representation consisted of 10,000 samples that were held constant throughout all of the experiments. The TR representation generated a different set 10,000 samples for each experiment by collecting five samples each from 2000 trajectory rollouts. Finally the AM representation was implemented by sampling from polygonal iso-contours of X_o spaced every 3° of angular resolution.

All three implementations of the MPF outperformed the CPF. As expected, the AM and TR representations both outperformed the RS representation, supporting hypotheses H2 and H3. This occurs because the RS representation attempts to sparsely cover the entire surface X_o with a relatively small number of samples, while the TR representation densely covers the states that we are most likely to reach.

Surprisingly, hypothesis H4 was not supported by the data: the AM representation did not achieve lower error than the TR representation. This occurred because the TR representation was able to saturate the regions of X_o that we are likely to encounter during execution. By doing so, the TR representation achieves such dense coverage of the relevant parts of that it is unlikely to fail at sampling from the dual proposal distribution.

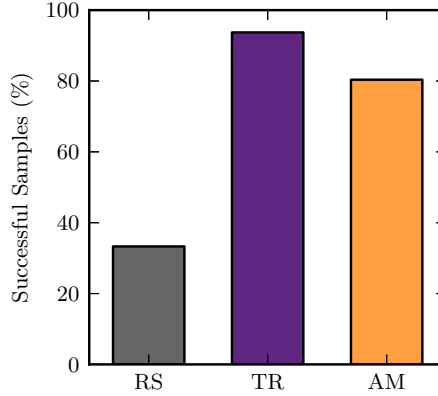


Fig. 6: Percent of the time that the MPF succeeded at sampling from the dual proposal distribution during contact. Sampling fails when all particles sampled from the contact manifold have a low probability $p(z_t|x_t, u_t)$ of generating z_t .

Sampling Failures. Figure 6 supports our intuition that the relatively poor performance of the RS representation is a result of it frequently failing to sample from the dual proposal distribution. The TR and AM representations fail to sample from the dual proposal distribution for only $< 20\%$ of updates. Conversely the RS representation fails to sample $> 60\%$ of the time. When sampling fails, the MPF behaves identically to the CPF and suffers from the same problem of particle starvation. As a result, the RS representation performs relatively poorly compared to the RS and TR representations in Fig. 5b.

8 Discussion

In this section, we discuss how partial sensor coverage and different contact manifold representations effect the manifold particle filter. Additionally, we discuss several possible ways of addressing the limitations of the MPF in future work.

Contact Manifold Representations. We discussed several possible implementations of the contact manifold that can be used to sample from the dual proposal distribution: rejection sampling (RS), trajectory rollouts (TR), and an analytic representation (AM). Each of these representations makes different assumptions about the structure of the problem.

The RS and TR representations approximate X_o with a discrete set of pre-computed samples $\tilde{X}_o$. These techniques make no assumptions about the geometry of the problem and widely applicable. Both techniques outperform the CPF, but MPF-TR outperforms MPF-RS. This occurs because the TR representation concentrates $\tilde{X}_o$ in the regions of the state space that we are most likely to see during execution. As a result, sampling from the dual proposal distribution is less likely to fail with TR than RS.

Unlike RS, the set of samples $\tilde{X}_o$ generated by TR are specific to $b(x_0)$ and cannot be generalized between problem instances. Even worse, pre-computing $\tilde{X}_o$ requires rolling out a large number of trajectories using the computationally expensive motion model. In summary, TR trades more pre-computation time for better online performance.

When it exists, an analytic representation of the contact manifold provides an exact representation of X_o . For polygonal geometry in $SE(2)$, the analytic representation requires minimal pre-processing and can be easily updated in real-time as the geometry of the object and hand changes. Additionally, it is efficient to uniformly sample states from X_o at runtime. Unlike the sample-based representations, these samples will be distributed uniformly with respect to the measure over X_o instead of the underlying space X . Finally, there is no chance of failing to sample from the dual proposal distribution due to a sparsity of samples.

The Observability of Contact Contact sensors frequently do not cover the entire surface of a hand. For example, the proximal links of the BarrettHand are not covered with tactile sensors and the SynTouch BioTac [Fishel & Loeb, 2012] sensor only provides tactile sensing on the interior of the fingertip. Even the iHY hand, [Odhner et al., 2013] which tightly integrates TakkTile sensors [Tenzer et al., 2012] into its mechanical design, does not cover the outside surface of the hand with sensors. As a result, it is important to consider the effect that observability of contact has on our state estimation ability.

The difference between “contact” and “observed contact” is captured in our definitions of the contact manifold X_c and the observable contact manifold $X_c \subseteq X_o$. The geometry of the non-observable region of the contact manifold $X_{no} = X_c \setminus X_o$ impacts the difficulty of the state estimation for contact manipulation problem. Ideally, the transition model will quickly move states out of X_{no} into X_o by pushing them into contact with a sensor. Any stable states in X_{no} , e.g. those that come to rest against a flat surface, will accumulate belief during execution.

Contact with Multiple Objects We implicitly assume that the hand can only contact the object that we are manipulating. This may not be possible in highly cluttered environments where we must contact multiple objects to achieve the desired task [Dogar et al., 2012]. In future work, we hope to explore methods of generalizing the MPF to environments with multiple—both static and movable—objects. We believe it is possible to do so through limited factoring of the belief state (e.g. through Rao-Blackwellization) to avoid requiring exponentially more particles.

Shape Uncertainty We assume that the hand and object both have known geometry. This is often not true when using compliant/under-actuated hands (e.g. the iHY hand [Odhner et al., 2013]) or manipulating un-modeled objects. Small variations of the object-hand geometry can cause large changes in the shape and topology of the contact manifold. We hope to address this additional source of uncertainty in future work by considering distributions over object and hand geometry. This would in effect, create a “fuzzy” contact manifold that consists of the union of several hypothesized contact manifolds.

Acknowledgements This work was supported by a NASA Space Technology Research Fellowship and the DARPA Autonomous Robotic Manipulation Software Track (ARM-S) program. We would also like to thank Mehmet Dogar, Anca Dragan, and the members of the Personal Robotics Lab for their helpful input.

References

- [Dogar et al., 2012] Dogar, M., K. Hsiao, M. Ciocarlie, & S.S. Srinivasa 2012. Physics-Based Grasp Planning Through Clutter. In RSS.
- [Farahat et al., 1995] Farahat, A. O., P. F. Stiller, & J. C. Trinkle 1995. On the geometry of contact formation cells for systems of polygons. IEEE T-RO.
- [Fishel & Loeb, 2012] Fishel, J. A., & G. E. Loeb 2012. Sensing Tactile Microvibrations with the BioTac Comparison with Human Sensitivity.
- [Jia & Erdmann, 1999] Jia, Y., & M. Erdmann 1999. Pose and motion from contact.
- [Koval et al., 2013] Koval, M.C., M.R. Dogar, N.S. Pollard, & S.S. Srinivasa 2013. Pose Estimation for Contact Manipulation with Manifold Particle Filters. In IEEE/RSJ IROS.
- [LaValle, 2006] LaValle, V. M. 2006. Planning algorithms. Cambridge university press.
- [Lozano-Pérez, 1983] Lozano-Pérez, T. 1983. Spatial Planning: A Configuration Space Approach. Computers, IEEE Transactions on.
- [Lynch et al., 1992] Lynch, K. M., H. Maekawa, & K. Tanie 1992. Manipulation and active sensing by pushing using tactile feedback.
- [Montemerlo et al., 2003] Montemerlo, M., S. Thrun, D. Koller, & B. Wegbreit 2003. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges.
- [Odhner et al., 2013] Odhner, L., L. P. Jentoft, M. R. Claffee, N. Corson, Y. Tenzer, R. R. Ma, M. Buehler, R. Kohout, R. D. Howe, & A. M. Dollar 2013. A Compliant, Underactuated Hand for Robust Manipulation. CoRR.
- [Pelletier, 2005] Pelletier, B. 2005. Kernel density estimation on Riemannian manifolds. Statistics & Probability Letters.
- [Tenzer et al., 2012] Tenzer, Y., L. P. Jentoft, & R. D. Howe 2012. Inexpensive and Easily Customized Tactile Array Sensors using MEMS Barometers Chips. IEEE.
- [Thrun et al., 2005] Thrun, S., W. Burgard, & D. Fox 2005. Probabilistic robotics. MIT Press.
- [Thrun et al., 2000] Thrun, S., D. Fox, & W. Burgard 2000. Monte Carlo localization with mixture proposal distribution.
- [Wein, 2013] Wein, R. 2013. 2D Minkowski Sums. In CGAL User and Reference Manual. CGAL Editorial Board, 4.3 edition.
- [Zhang et al., 2013] Zhang, L., S. Lyu, & J. Trinkle 2013. A Dynamic Bayesian Approach to Simultaneous Estimation and Filtering in Grasp Acquisition. In IEEE ICRA.
- [Zhang & Trinkle, 2012] Zhang, L., & J. C. Trinkle 2012. The application of particle filtering to grasping acquisition with visual occlusion and tactile sensing.