Pose Estimation for Planar Contact Manipulation with Manifold Particle Filters

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

{mkoval,nsp,siddh}@cs.cmu.edu
The Robotics Institute, Carnegie Mellon University

Abstract

We investigate the problem of using contact sensors to estimate the pose of an object during planar pushing by a fixed-shape hand. Contact sensors are unique because they inherently discriminate between "contact" and "no-contact" configurations. As a result, the set of object configurations that activates a sensor constitutes a lower-dimensional *contact manifold* in the configuration space of the object. This causes conventional state estimation methods, such as the particle filter, to perform poorly during periods of contact due to particle starvation.

In this paper, we introduce the *manifold particle filter* as a principled way of solving the state estimation problem when the state moves between multiple manifolds of different dimensionality. The manifold particle filter avoids particle starvation during contact by adaptively sampling particles that reside on the contact manifold from the *dual proposal distribution*. We describe three techniques—one analytical, and two sample-based—of sampling from the dual proposal distribution and compare their relative strengths and weaknesses. We present simulation results that show that all three techniques outperform the conventional particle filter in both speed and accuracy. Additionally, we implement the manifold particle filter on a real robot and show that it successfully tracks the pose of a pushed object using commercially available tactile sensors.

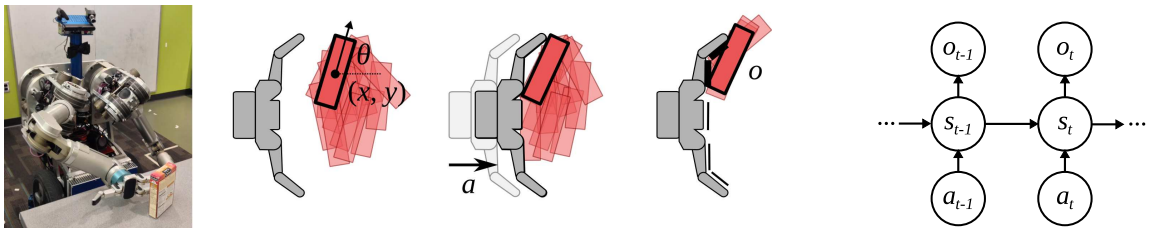
1 Introduction

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* during which we make *persistent contact* with the environment. During contact, tactile feedback is critical to localize the object being manipulated.

Armed with real-time observations from tactile sensors (Odhner et al. 2013, Tenzer et al. 2014, Fishel & Loeb 2012), manipulators should also be able to estimate the pose of the manipulated object. Early work attempted to solve this problem by deriving analytical state estimators to track and, in some cases, control the pose of an object from contact positions based on simple models of physics (Jia & Erdmann 1999). However, these models fail to accurately capture the reality of manipulation because 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 a particle filter to estimate the pose (Corcoran & Platt 2010, Zhang & Trinkle 2012), contact state (Gadeyne et al. 2005, Meeussen et al. 2007), and physical properties (Zhang et al. 2013) of an object during manipulation. However, the conventional particle filter (CPF, Section 4) suffers from a startling problem: *the CPF systematically performs worse as the sensor resolution and update frequency increases* (Section 4.4).

This problem arises because contact sensing accurately *discriminates* between contact and no-contact. Topologically, the set of states that are consistent with a contact observation lies in the lower dimensional *contact manifold* (Section 3.1) embedded in the configuration space of the object. Particles sampled from the state space have a low probability of being on the contact manifold and, as a result, *particle starvation* (Thrun et al. 2005, Thrun, Fox & Burgard 2000) occurs in the vicinity of the true state. In the limit, when the sensor can perfectly localize contact along the hand, this region shrinks to a zero measure set (Section 9.4) and the CPF is completely ineffective.

In this paper, we consider the pose estimation for contact manipulation problem (Figure 1) as one of Bayesian estimation (Section 3) and introduce the *manifold particle filter* (Koval, Dogar, Pollard & Srinivasa 2013, Koval, Pollard & Srinivasa 2013) (MPF, Section 5) as a principled way of solving the problem of particle starvation during contact. We specifically apply the MPF to the case of a fixed-shape hand equipped with accurate contact sensors pushing an object in the plane. The MPF addresses the state estimation problem when the state moves between multiple manifolds of possibly different dimensions. In the case of contact manipulation, this occurs when the object makes or breaks contact with a contact sensor.



(a) Contact Manipulation Problem (b) Bayes Network

Figure 1: The contact manipulation problem. (a) HERB (Srinivasa et al. 2012) pushing a rectangular box across the table. The state $s \in S$ is the pose of the box relative to the hand. An action $a \in A$ is a relative motion of the hand. After taking action a, HERB receives an observation $o \in O$ indicating where the object touched the hand. (b) The Bayes filter uses the Markov property to recursively compute $b(s_t)$ from $b(s_{t-1})$

The gist of our algorithm is quite simple: we factorize belief into the marginal probability of being on a manifold and the probability of the state conditioned on that manifold. We first choose a manifold, then sample a particle from that manifold. The key result of this factorization is that we can apply a different sampling technique to each manifold. In our case, we sample from free space using the conventional proposal distribution (Thrun et al. 2005) and from the contact manifold using the *dual proposal distribution* (Thrun, Fox & Burgard 2000).

The dual proposal distribution avoids particle starvation by sampling particles from the contact manifold that are consistent with the most recent observation. When the object-hand geometry consists of polygons in $\mathbb{R}^2$, an *analytic representation* (AM, Section 6.3.3) of the contact manifold can be computed using the Minkowski sum (Lozano-Pèrez 1983, Wein 2013). This provides a computationally efficient way of sampling from the contact manifold and, thus, implementing the dual proposal distribution. However, computing an analytic representation of the contact manifold is not always possible.

It is possible to apply the MPF to arbitrary planar geometry by approximating the contact manifold with a set of weighted samples. We present two such representations. The *rejection sampled representation* (RS, Section 6.3.1) distributes samples uniformly in the ambient space near the manifold and, thus, is agnostic to the policy followed during execution. The *trajectory rollout representation* (TR, Section 6.3.2) concentrates samples on the regions of the manifold that we are most likely to encounter during execution.

Our simulation results (Section 7) confirm that the MPF outperforms the CPF in terms of both estimation accuracy and computational efficiency. We show that the MPF, in contrast to the CPF, scales favorably with increases in the robot's sensor resolution and update frequency (Section 7.5). We also analyze the relative performance of the RS, TR, and AM manifold representations. As expected, all three representations outperform the CPF and the AM and TR representations both outperform RS (Section 7.6). Surprisingly, however, our experiments show that TR performs as well as AM. By focusing samples on likely regions, TR saturates these regions at a resolution indistinguishable from the analytic solution.

We support these results with an implementation of the MPF on Andy Bagnell et al. (2012) (Section 8), a bimanual manipulator equipped with the Barrett WAM arm (Salisbury et al. 1988) and the i-HY end-effector (Odhner et al. 2013). Using the MPF, Andy successfully estimated the pose of several objects while executing a pushing action using feedback from the i-HY hand's tactile sensors (Tenzer et al. 2014). These experiments demonstrate that the MPF is able to successfully estimate the pose of an object using commercially available sensors.

The contact manipulation problem exhibits unique structure that makes it fundamentally different from most state estimation and planning problems. By exploiting the structure of the contact manifold, we are able to significantly outperform standard state estimation techniques. Furthermore, by exploiting the geometry of the hand-object interaction, the trajectory rollout representation achieved performance comparable to that of the analytic solution.

However, the implementation of MPF discussed in this paper has several limitations: we only consider planar manipulation with quasistatic physics (Section 9.5.1), assume that the hand has a fixed shape (Section 9.5.2), and do not refine our estimate of the physical properties of the environment (Section 9.5.4) during execution. We also assume that contact sensors are discriminative (Section 9.5.3), i.e. are capable of

accurately differentiating between contact and no-contact. We are interested in addressing all four of these limitations in future work. Finally, we are excited to use the belief state estimated by the MPF as feedback for closed-loop manipulation primitives (Section 9.5.5).

2 Related Work

This paper builds on a long history of research on planar manipulation (Section 2.1) and tactile sensing (Section 2.2) as sensor feedback. Our choice of the particle filter was inspired by recent work on object pose estimation (Section 2.3) and contact state estimation (Section 2.4) for manipulation. Finally, we adapt the dual and mixture proposal distributions (Section 2.5) used for mobile robot localization to the contact manipulation problem.

2.1 Manipulation via Pushing

Our focus is on contact manipulation and, in particular, planar pushing actions. Pushing enables robots to perform a wide variety of tasks that are not possible through pick-and-place manipulation alone: pushing can move objects that are too large or heavy to be grasped (Dogar & Srinivasa 2011), is effective at manipulating objects under uncertainty (Brost 1988, Dogar & Srinivasa 2010), and can be used as *pre-grasp manipulation* to bring objects to configurations where they can be easily grasped (Chang et al. 2010, Kappler et al. 2010, King et al. 2013). Additionally, pushing can be used to simultaneously move multiple objects (Dogar et al. 2012).

Since pushing offers such a dramatic expansion of manipulation skills, there has been extensive research on the fundamental mechanics of pushing (Mason 1986, Lynch & Mason 1995, Howe & Cutkosky 1996, Lynch & Mason 1996) and on the planning of planar pushing operations (Lynch & Mason 1996, Akella & Mason 1998). Recently, there has been interest in generating push trajectories using sampling based planners (Lau et al. 2011, Cosgun et al. 2011), trajectory optimization (King et al. 2013), and learning methods (Zito et al. 2012). We leverage this work by using the quasistatic physics model (Lynch et al. 1992, Howe & Cutkosky 1996), the same model used by much of this prior work (Dogar & Srinivasa 2010, Dogar et al. 2012, Dogar & Srinivasa 2011), to estimate the motion of the object.

Most of these techniques, however, employ pushing as an open-loop operation and are sensitive to object pose uncertainty. One notable exception is the *push-grasp* (Dogar & Srinivasa 2010, Dogar et al. 2012), which reasons about pose uncertainty during the planning process to generate a straight-line action that funnels the object into the hand. This work is complementary to our own: tracking the pose of an object during the execution of a push-grasp would allow the robot to cope with larger amounts of uncertainty and detect success (or failure) more quickly. In the future, we plan to use the state estimate produced by the MPF to adapt the robot's motion in real-time. We took a first step in this direction in Koval et al. (2014).

2.2 Tactile Sensing

Contact sensing is an attractive type of feedback during manipulation because it directly observes the robot's interaction with the environment. Contact sensors come in many forms, including binary switches (Edin et al. 2006), pressure-sensitive pads (Tenzer et al. 2014), and complex fingertips (Fishel & Loeb 2012) with multimodal sensing capabilities. The MPF can accommodate any type of sensor that accurately discriminates between contact and no-contact and can be characterized by a probabilistic model.

One method of using tactile sensors during manipulation is to create a feedback controller that directly maps sensor readings to actions. For example, a robot can use the tactile Jacobian to servo its end-effector to a desired contact state (Zhang & Chen 2000, Li et al. 2013). These controllers are effective for specific tasks, such as following a contour (Zhang & Chen 2000) or locally refining the quality of a grasp (Platt et al. 2010). Another approach is to learn a task-specific policy (Pastor et al. 2011) from demonstration. Unfortunately, it is difficult to generalize these techniques to the full spectrum of manipulation tasks. Our method explicitly estimates the state of the object, which can then be used by a higher-level planning algorithm to achieve an arbitrary goal.

Another approach to using contact sensors is to first localize the object, then grasp it. This approach is commonly implemented by executing a sequence of move-until-touch actions (Petrovskaya & Khatib 2011,

Javdani et al. 2013, Hebert et al. 2013, Hsiao 2009) that localize the object within some tolerance, then execute an open-loop trajectory to achieve a grasp. These techniques generally assume that the object does not move (Javdani et al. 2013, Petrovskaya & Khatib 2011) or use a simple motion model that causes actions to "bump" the object by a small amount (Hsiao 2009). The MPF solves a fundamentally different problem: it estimates the pose of an object during manipulation and does not plan any actions. Additionally, the MPF reasons about the motion of the object using an accurate physics model.

2.3 Object Pose Estimation

There is a rich history of using recursive estimators to track the pose of objects for manipulation (Harris 1992, Drummond & Cipolla 2002). Recently, there has been interest in integrating models of physics into visual tracking algorithms to improve their performance. Duff et al. (2010) integrated the PhysX simulator (NVIDIA Corporation 2014) into a RANSAC-based (Fischler & Bolles 1981) tracker and significantly outperformed several physics-agnostic baseline algorithms. The same physics model was later used for the transition model of a particle filter (Gordon et al. 1993) that uses edge likelihood measurements (Mörwald et al. 2009) to track an object's three-dimensional pose (Duff et al. 2011). Similar to this work, the MPF uses a physics model (Lynch et al. 1992) as the transition model in a particle filter for object pose estimation. However, the MPF uses observations from contact sensors—instead of vision—for feedback.

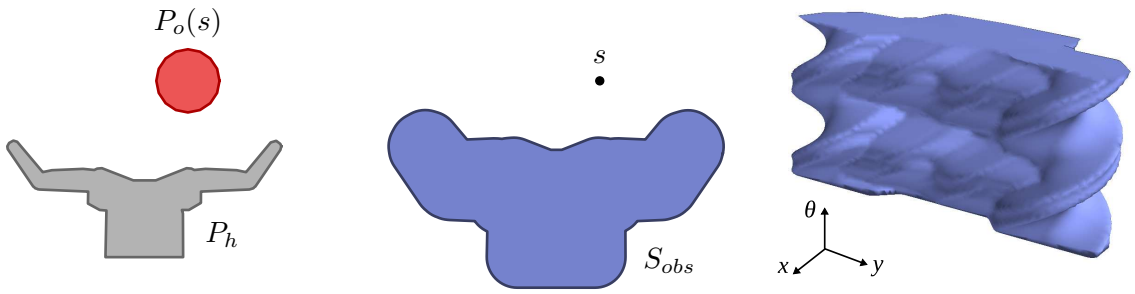
More recently, Zhang & Trinkle (2012) used a particle filter to combine contact sensing, visual pose estimates, and an NCP-based physics model (Ferris & Munson 1999) to track the pose of an object being pushed by a manipulator in the plane. Their results demonstrate that contact sensing can significantly improve visual pose estimation accuracy, particularly during extended periods of visual occlusion. In later work, the same authors used a Rao-Blackwellized (Blackwell 1947) particle filter to simultaneously estimate object pose and the value of spatio-temporally varying parameters (Zhang 2013); e.g. friction coefficients. Unfortunately, the experiments in Zhang & Trinkle (2012) show that this state estimator too slow to run in real-time, even when applied to a simple hand with three contact sensors. The MPF does not estimate physical parameters of the environment (see Section 9.5.4 for future work), but is able to use a smaller set of particles to achieve near real-time performance on complex sensor configurations.

The authors of both of these particle filters (Zhang & Trinkle 2012, Duff et al. 2011) note that the performance of the estimator heavily depends on the type of uncertainty introduced into the transition model. Adding noise to the output of the physics model can produce inter-object penetration and physically-infeasible motion. Instead, noise should be introduced to the input of the simulator by applying random forces to the simulated object (Duff et al. 2011, Duff 2011) or by adding noise to the model's parameters (Zhang & Trinkle 2012). We adopt the latter technique in the MPF by sampling the hand-object friction coefficient and the radius of the object's pressure distribution—the parameters of our quasistatic physics model—from probability distributions.

Modifying noise in the transition model does not, however, address the particle starvation problems inherent to contact sensing. The problem of particle starvation when using contact sensors in a particle filter have been recognized several times in the literature (Gadeyne et al. 2005, Zhang & Trinkle 2012, Zhang 2013). This problem is commonly addressed by "smoothing" the observation model with artificial noise that spreads contact observations over a non-infinitesimal, full-dimensional region of the state space (Zhang 2013, Javdani et al. 2013, Zhang & Trinkle 2012, Corcoran & Platt 2010). This approach—while sometimes effective—scales poorly to high-resolution sensors and discards the most important property of contact sensors: the difference between contact and no-contact. Additionally, this assumption can lead to belief states that drift arbitrarily far from those generated by true belief dynamics over time (Thrun, Fox & Burgard 2000). The MPF solves the particle starvation issue by sampling from the dual proposal distribution, which is theoretically sound and suffers neither of these issues.

2.4 Contact State Estimation

For some applications, e.g. learning a compliant controller (Schutter & Brussel 1988), estimating the contact state between an object and the environment is equally as important as estimating its pose. The contact state is typically represented as a contact formation (Xiao 1993), which is the set of elementary contacts (e.g. face-vertex, face-edge, etc) between the robot and the environment. The contact manifold (Section 3.1) used



(a) Workspace Geometry (b) C-Space Geometry (c) Contact Manifold

Figure 2: (a) Workspace and (b) C-space geometry for a hand pushing a bottle. The contact manifold S_c is the lower-dimensional boundary between S_{invalid} and S_{free}. (c) In the case of the asymmetric object in Figure 1, the contact manifold is a two-dimensional manifold embedded in $SE(2)$. This figure was generated discretizing the object's orientation and computing analytic Minkowski sums as described in Section 6.3.3.

by the MPF is equivalent to the projection of all possible contact formations into the configuration space of the object relative to the hand.

Gadeyne et al. (2005) used a particle filter to track a hybrid discrete-continuous probability distribution over a small set of contact states (discrete) and object pose (continuous). The estimator was later scaled to the full set of possible contact states by using a pre-computed contact state graph (Xiao 1993) to generate a sparse transition model between discrete contact state (Meeussen et al. 2007). Constructing the contact state graph offline mirrors the MPF's pre-computation of the contact manifold (Section 6.3).

The key difference between the algorithms is that the MPF tracks a single mixed-dimensional distribution over object pose (Section 9.4) that implicitly encodes contact state. In contrast, the algorithm presented in Meeussen et al. (2007) explicitly maintains a distribution over contact states. Tracking a single distribution guarantees that MPF's pose estimate incorporates the contact state constraints imposed by contact observations. This allows the MPF to use a small number of particles to track the pose of the object, enabling us to use a computationally expensive physics model in the transition model. This comes at the cost of assuming that it is known whether the object lies on the observable contact manifold; e.g. by using discriminative contact sensors (Section 6.1).

2.5 Bayesian Estimation

The MPF, along with the other Bayesian state estimation algorithms described above, build on a rich history of Bayesian estimation research. The Kalman filter (Kalman 1960), extended Kalman filter, and unscented Kalman filter (Julier & Uhlmann 1997) have been shown to be effective on problems with Gaussian belief states.[1] Unfortunately, none of these techniques are directly applicable to the contact manipulation problem: pushing and tactile sensing are both highly non-linear and frequently produce non-Gaussian and multi-modal belief states.

Instead, similar to the prior work described in Sections 2.3 and 2.4, we track the pose of the object using a particle filter (Gordon et al. 1993). We borrow the concepts of the *dual* and *mixture proposal distributions* from mobile robot localization literature (Thrun, Fox & Burgard 2000, Montemerlo et al. 2003). Particle filters in this domain suffer from a similar particle starvation problem on robots using very high-accuracy depth rangefinders or cameras. The dual proposal distribution solves this problem by sampling particles directly from the observation model. This is possible because the vision and depth sensors used on mobile robots provide high-accuracy readings independent of the true state. Conversely, contact sensors only provide accurate readings when the object is in contact with the sensor. Therefore, the MPF must arbitrate between particles sampled from the conventional and dual proposal distributions.

[1] The Kalman filter requires the system to have a linear transition model with additive Gaussian noise and a linear observation model corrupted by Gaussian white noise. The extended Kalman relaxes this requirement to arbitrary differentiable functions. The unscented Kalman filter further relaxes the requirement to arbitrary functions. However, all three algorithms assume that posterior belief state can be approximated as Gaussian.

3 Pose Estimation for Contact Manipulation

Let $s \in S$ be the state of a dynamical system which evolves over time under actions $a \in A$ and produces observations $o \in O$. The *state estimation problem* addresses the computation of the *belief state* $b(s_t)$, the probability distribution over the state s_t at time t

$$b(s_t) = p(s_t | a_{1:t}, o_{1:t}) \tag{1}$$

given the history of actions $a_{1:t} = (a_1, \ldots, a_t)$ and observations $o_{1:t} = (o_1, \ldots, o_t)$ (Thrun et al. 2005).

We focus on the problem of *pose estimation for contact manipulation*, where the goal is to estimate the pose of an object relative to the hand. In this paper, we specifically consider the problem of planar contact manipulation with quasistatic physics (Lynch et al. 1992) and a fixed hand shape. The *quasistatic assumption* states that an object will stop moving as soon as it leaves contact with the hand. Prior work has shown that this is a good approximation for the planar manipulation of many household objects (Dogar & Srinivasa 2010, 2011, Dogar et al. 2012, Dogar & Srinivasa 2012).

As a result of this assumption, state $s \in S = SE(2)$ is the pose of the object relative to the hand (Figure 1a-Left) and an action $a = (v, \Delta t) \in A$ is the relative velocity of the hand $v \in se(2)$ applied for a duration $\Delta t \in \mathbb{R}^+$ (Figure 1a-Middle). During contact, the object moves according to a stochastic *transition model* $p(s_t | s_{t-1}, a_t)$ that encodes the motion of the object in response to pushing action a_t. We model uncertainty in the physics model by drawing the model's parameters from a known distribution. Adding noise to the input—instead of the output—of a physics model has been shown to avoid inter-object penetration and ensure that the object's motion remains physically feasible (Duff 2011).

After taking action a_t, contact sensors provide an observation $o_t \in O$ (Figure 1a-Right). This observation is either a contact observation ($o_t \in O_c$) or a no-contact observation ($o_t \in O_{nc} = O \setminus O_c$). If $o_t \in O_c$, then a contact sensor has fired and the observation o_t may provide additional information about the pose of the object. Otherwise, if $o_t \in O_{nc}$, the observation indicates that contact has not occurred. Both of these properties are combined into the stochastic *observation model* $p(o_t | s_t, a_t)$ as the probability of state s_t generating observation o_t after executing action a_t.[2]

3.1 Contact Manifold

Contact manipulation poses a unique state estimation challenge because the state evolves on a lower-dimensional manifold embedded in S. The state space S naturally partitions into: (1) penetrating contact S_{invalid}, (2) non-penetrating contact S_c, and (3) no contact S_{free}. These three sets are defined by the interplay between the geometry of the object and the geometry of the hand.

Let $P_h \subseteq \mathbb{R}^2$ be the geometry of the hand and $P_o(s) \subseteq \mathbb{R}^2$ be the geometry of the object at configuration $s \in S$. The set of all object poses that are in collision with the hand form the *configuration space obstacle* (Lozano-Pèrez 1983)

$$S_{\text{obs}} = \text{CObstacle}_o(P_h) = \{s \in S : P_h \cap P_o(s) \neq \emptyset\}$$

of the hand in the object's configuration space.

Any configuration in $S_{\text{invalid}} = \text{int}(S_{\text{obs}})$ is invalid because the object penetrates the hand. Conversely, any configuration in $S_{\text{free}} = S \setminus S_{\text{obs}}$ is in *free space* where the object is out of contact with the hand. Therefore, any valid object configuration of the object that is in contact with the hand must lie on the *contact manifold* $S_c = S_{\text{obs}} \setminus \text{int}(S_{\text{obs}})$ that forms the boundary between S_{invalid} and S_{free}.

Figure 2 shows the geometry of the workspace (Figure 2a) and configuration space (Figure 2b) of a BarrettHand manipulating a circular bottle. Since the object is radially symmetric, S is simply the set of (x, y) positions of the object relative to the hand. If the object is not symmetric, such as the elongated box shown in Figure 1, S is the set of of three-dimensional (x, y, θ) coordinates of the object relative to the hand, and the contact manifold (Figure 2c) is a two-dimensional structure embedded in $SE(2)$. The structure

[2] We define the observation model as $p(o_t | s_t, a_t)$ instead of the more traditional $p(o_t | s_t)$. We do so to recognize the fact that—unlike in many applications of the Bayes filter—o_t is strongly influenced by the most recent action a_t. This is equivalent to constructing an augmented state space $S' = S \times A$ and an augmented transition model that stores a_t in the successor state s_{t+1}.

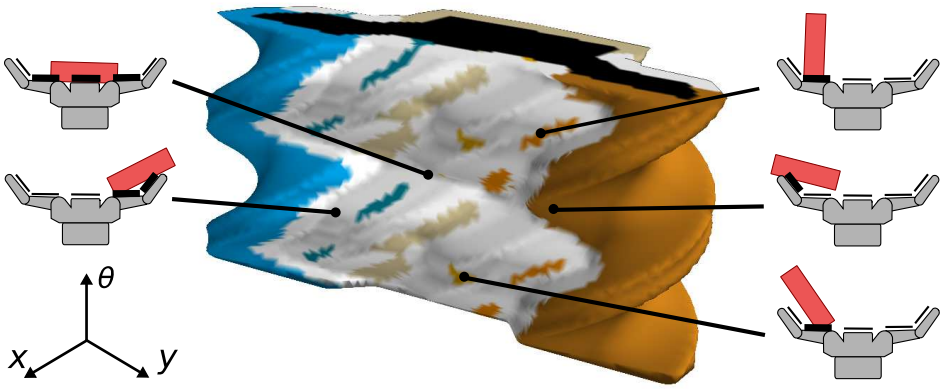


Figure 3: Observable contact manifold S_o for a two-dimensional BarrettHand pushing a rectangular box. Each point corresponds to a configuration of the object $s \in S_c$ that is in non-penetrating contact with the hand and is uniquely colored by the active contact sensors. Configurations that are in contact with multiple sensors are white. This figure was generated using the analytic representation of the contact manifold described in Section 6.3.3.

shown in Figure 2c is the C-obstacle S_{obs} of the hand in the object's configuration space. Points inside the obstacle are is $S_{invalid}$, points outside the obstacle is S_{free}, and the surface separating $S_{invalid}$ from S_{free} is the contact manifold S_c. In this case, S_c is repeated twice along the θ-axis because the box exhibits rotational symmetry.

3.2 Observable Contact Manifold

We know that $s \in S_c$ during periods of contact. However, our contact sensors may not be able to sense contact over the entire surface of the hand. We define the *observable contact manifold* $S_o \subseteq S_c$ as the set of object poses that are capable of generating contact observations $o \in O_c$.

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

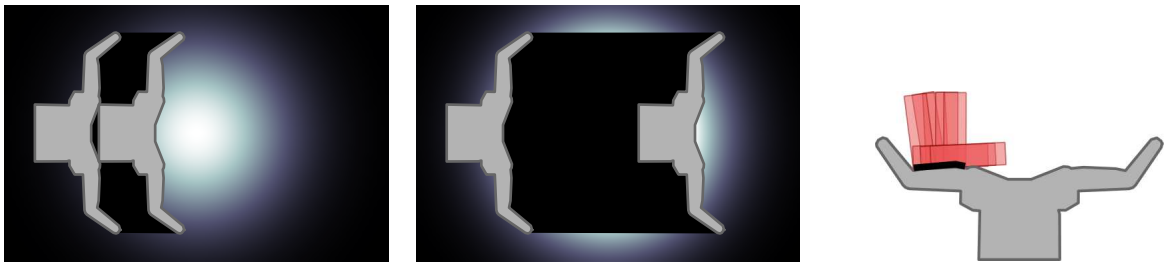
$$S_s = \text{CObstacle}_o(P_s) = \{s \in S : P_s \cap P_o(s) \neq \emptyset\}$$

of the sensors in the object's configuration space. The *observable contact manifold* $S_o = S_s \cap S_c$ consists of the set of valid object configurations that have high probability of generating a contact observation $o_t \in O_c$. Intuitively, S_o is the set of object poses that are in non-penetrating contact with one or more contact sensors.

Figure 3 shows the contact manifold colored by which sensors are active at each point. For example, states in the large, dark orange region of the manifold are in contact with—and, thus, are likely to activate—the left distal contact sensor. The two disjoint, light orange patches on the top-right of the manifold contain the two configurations of the box shown in Figure 4c. Similarly, states in the central tan region of the manifold are in contact with the palm sensor. Regions of the contact manifold that are in simultaneous contact with multiple sensors are drawn as white.

4 Conventional Particle Filter

In this section, we provide a brief introduction to Bayesian estimation (Section 4.1) and the conventional particle filter (Thrun et al. 2005) (CPF, Section 4.2). We show how the CPF can be applied to the contact manipulation problem. Unfortunately, we also demonstrate that the algorithm inherently suffers from particle starvation (Section 4.4) during periods of contact.



(a) $o_1 \in O_{\text{nc}}$ (b) $o_1, \ldots, o_n \in O_{\text{nc}}$ (c) $o_1, \ldots, o_n \in O_{\text{c}}$

Figure 4: The contact manipulation problem commonly produces non-Gaussian and multi-modal belief states. (a) Receiving a single no-contact observation results in a non-Gaussian posterior belief state. (b) Continuing to receive no-contact observations results in a multi-modal distribution. (c) Contact observations can also result in a multi-modal distribution over the object's pose.

4.1 Bayes Filter

The *Bayes filter* is the most general algorithm for recursively filtering a belief state $b(s_t)$ (Thrun et al. 2005) given an initial belief state $b(s_0)$ and the history of actions and observations. This filter assumes that the system satisfies the *Markov property*. This property, $s_t \perp (a_{1:t-1}, o_{1:t-1})|s_{t-1}$, says that state is a sufficient statistic for all previous actions and observations. The Markov property is drawn as a Bayes network in Figure 1b.

We derive the Bayes filter by considering our belief $b(s_{0:t}) = p(s_{0:t}|a_{1:t}, o_{1:t})$ over the trajectory of states $s_{0:t}$ reached by starting in state s_0 and executing the sequence of actions $a_{1:t}$. By applying Bayes rule and the Markov property, we can derive the recursive update rule

$$\begin{aligned} b(s_{0:t}) &= p(s_{0:t}|a_{1:t}, o_{1:t}) \\ &= \eta\, p(o_t|s_{0:t}, a_{1:t}, o_{1:t-1}) p(s_{0:t}|a_{1:t}, o_{1:t-1}) && \triangleright \text{ Bayes rule} \\ &= \eta\, p(o_t|s_{0:t}, a_{1:t}, o_{1:t-1}) p(s_t|s_{0:t-1}, a_{1:t}, o_{1:t-1}) p(s_{0:t-1}|a_{1:t}, o_{1:t-1}) \\ &= \eta\, p(o_t|s_t, a_t) p(s_t|s_{t-1}, a_t) p(s_{0:t-1}|a_{1:t-1}, o_{1:t-1}) && \triangleright \text{ Markov property}^3 \end{aligned} \tag{2}$$

where $\eta = [p(o_t|a_{1:t}, o_{1:t-1})]^{-1}$ is a normalization factor (Thrun et al. 2005). This is equal to the probability of receiving the sequence of observations $o_{1:t}$ given our history.

Equation (2) recursively defines the t-step joint belief $b(s_{0:t})$ in terms of the $(t-1)$-step joint belief $b(s_{0:t-1})$. Unfortunately, this means that representing $b(s)$ requires memory that is exponential in the time horizon t. We shrink the size of our belief to constant by computing the marginal

$$\begin{aligned} b(s_t) &= \eta\, p(o_t|s_t, a_t) \int_S p(s_t|s_{t-1}, a_t) \int p(s_{0:t-1}|a_{1:t-1}, o_{1:t-1}) ds_{0:t-2} ds_{t-1} \\ &= \eta\, p(o_t|s_t, a_t) \int_S p(s_t|s_{t-1}, a_t) b(s_{t-1}) ds_{t-1}, \end{aligned} \tag{3}$$

of the joint belief $b(s_{0:t})$ over the history $s_{0:t-1}$. This equation recursively constructs $b(s_t)$ from $b(s_{t-1})$ and, by doing so, forms the basis of dynamic Bayesian estimation.

There are several ways of implementing the Bayes update (Equation 3) depending upon the properties of the system. The Kalman filter (Kalman 1960) is optimal when the $b(s_0)$ is Gaussian, the transition model is linear, and observations are corrupted by additive Gaussian white noise. The extended (Kalman 1960) and unscented (Julier & Uhlmann 1997) Kalman filters relax the constraint that the system is linear, but still assume that the belief state is Gaussian.

However, none of these assumptions are valid for the contact manipulation problem. Even in the simplest quasistatic case the transition model is a function of the contact physics between the hand, object, and

[3] The Markov property, as stated in the text, does not directly imply that $p(s_{0:t-1}|a_{1:t}, o_{1:t-1}) = p(s_{0:t-1}|a_{1:t-1}, o_{1:t-1})$. We additionally assume that $s_{0:t-1} \perp a_t|(a_{1:t-1}, o_{1:t-1})$; i.e. action a_t does not affect states in the past. This is true if the policy used to select a_t is a function of only the history $a_{1:t-1}$ and $o_{1:t-1}$ or, more commonly, the belief $b(s_{t-1})$.

Algorithm 1 Conventional Particle Filter

Input: action $a_t \in A$ and observation $o_t \in O$
Input: particles $S_{t-1} = \{\langle s_{t-1}^{[i]}, w_{t-1}^{[i]} \rangle\}_{i=1}^{n}$ from time $t-1$ such that $S_{t-1} \sim b(s_{t-1})$
Output: particles $S_t = \{\langle s_t^{[i]}, w_t^{[i]} \rangle\}_{i=1}^{n}$ at time t such that $S_t \sim b(s_t)$

```
Ŝ_t ← ∅
for i = 1, ..., n do
    s_t^[i] ~ p(s_t^[i] | s_{t-1}^[i], a_t)
    w_t^[i] ← w_{t-1}^[i] p(o_t | s_t^[i], a_t)
    Ŝ_t ← {⟨s_t^[i], w_t^[i]⟩} ∪ Ŝ_t
end for
S_t ← Resample(Ŝ_t)
```

support surface. This includes the hand-object geometry and discrete transitions between contact states. As a result, the transition model is non-linear and lacks analytic derivatives Zhang & Trinkle (2012). Similarly, the observation model is highly non-linear because the probability of an observation sharply changes between S_{free} and S_{c}.

Even worse, the belief state quickly becomes non-Gaussian even if $b(s_0)$ is Gaussian: a single no-contact observation $o_1 \in O_{\text{nc}}$ assigns $b(s_1)$ zero probability in the swept volume of the contact sensors (Figure 4a). Furthermore, $b(s_t)$ becomes multi-modal (Figure 4b) if the hand continues to receive no-contact observations. The belief state can also become multi-model after receiving a contact observation that does not unambiguously resolve the object's orientation. Figure 4c shows one example where pushing straight causes an object to settle into one of two stable configurations.

4.2 Particle Filter

The *particle filter* (Gordon et al. 1993, Thrun et al. 2005), shown in Algorithm 1, is a non-parametric realization of the Bayes filter that represents the belief state $b(s_t)$ with a discrete set of samples. The samples $s_t^{[1]}, \ldots, s_t^{[n]} \in S$, along with their weights $w_t^{[1]}, \ldots, w_t^{[n]} \in \mathbb{R}^{\geq 0}$, are called *particles* $S_t = \{\langle s_t^{[i]}, w_t^{[i]} \rangle\}_{i=1}^{n}$ and are distributed according to $b(s_t)$. The particle filter implements the Bayesian update by recursively constructing S_t from S_{t-1} using importance sampling (Smith & Gelfand 1992).

The key insight behind the particle filter is that it is difficult to directly sample from the *target distribution* (Equation 3), but we can instead sample from a *proposal distribution* (Thrun et al. 2005, Thrun, Fox & Burgard 2000)

$$s_t^{[i]} \sim q(s_t)$$

that we choose to be easy to sample from. We make no assumption about the distribution $q(s_t)$, except the support of q is a superset of the support of $b(s_t)$; i.e. $b(s_t) > 0 \implies q(s_t) > 0$. Intuitively, q represents a "guess" at the target distribution that is easier to sample from than the true target distribution.

Next, the particle filter corrects for the mismatch between the proposal distribution $q(s_t)$ and the target distribution by computing *importance weights*. The importance weight $w_t^{[i]}$ for sample $s_t^{[i]}$ is

$$w_t^{[i]} = \frac{b(s_t^{[i]})}{q(s_t^{[i]})},$$

the ratio of the target distribution to the proposal distribution. Intuitively, importance weights decrease the influence of particles that are over-represented ($q(s_t) > b(s_t)$) and increase the influence of particles that are under-represented ($q(s_t) < b(s_t)$) by the proposal distribution.

Given any function $f : S \to \mathbb{R}$, we can use S_t to approximate the expectation

$$E_{s_t \sim b(s_t)}\left[f(s_t)\right] \approx \sum_{i=1}^{n} w_t^{[i]} f\left(s_t^{[i]}\right) \tag{4}$$

under $b(s_t)$, assuming the weights are normalized such that $\sum_{i=1}^{n} w_t^{[i]} = 1$. In the limit as $n \to \infty$, the right-hand side of Equation 4 converges to $E[f(s_t)]$ for $s_t \sim b(s_t)$. This property allows us to treat the weighted set of samples S_t as a finite-dimensional approximation of the true belief $b(s_t)$. We use the notation $S_t \sim b(s_t)$ to denote that the set of particles S_t has this property.

This sampling strategy is known as *sequential importance sampling* (SIS) because it sequentially constructs $S_t \sim b(s_t)$ from $S_{t-1} \sim b(s_{t-1})$. Unfortunately, it has been shown (Thrun et al. 2005) that SIS causes the variance of the weights to increase over time and, consequently, only one particle has non-zero weight in the limit. This issue is solved by using *sequential importance resampling* (SIR) to periodically resample (Algorithm 1, Line 7) S_t with replacement in proportion to their weights. In practice, we use low-variance resampling (Thrun et al. 2005) to implement the resampling step. After resampling, S_t is distributed with respect to $b(s_t)$ with unit weights.

4.3 Conventional Particle Filter

Implementing the particle filter requires choosing a proposal distribution that satisfies two properties. First, it should be easy to sample $s_t^{[i]} \sim q(s_t)$ from the proposal distribution. Second, it must be possible to compute the importance weight $w_t^{[i]} = b(s_t)/q(s_t)$.

The most commonly used proposal distribution, which we refer to as the *conventional proposal distribution*, is

$$q(s_t) = \int_S p(s_t|s_{t-1}, a_t) b(s_{t-1}) ds_{t-1}, \tag{5}$$

which is equal to the belief state after taking action a_t, but before receiving observation o_t. Sampling from Equation 5 is implemented by forward simulating each $s_{t-1}^{[i]} \in S_{t-1}$ to time t using the transition model $s_t^{[i]} \sim p(s_t^{[i]}|s_{t-1}^{[i]}, a_t)$ (Algorithm 1, Line 3). Since $S_{t-1} \sim b(s_{t-1})$, the output of this operation is distributed according to $q(s_t)$. We refer to any particle filter that samples from $q(s_t)$ as a *conventional particle filter* (CPF).

Next, the CPF computes an importance weight $w_t^{[i]}$ (Algorithm 1, Line 4) equal to the ratio of the target distribution (Equation 3) to the proposal distribution (Equation 5)

$$w_t^{[i]} = \frac{b(s_t^{[i]})}{q(s_t^{[i]})} = \frac{\eta\, p(o_t|s_t^{[i]}, a_t) \int_S p(s_t^{[i]}|s_{t-1}, a_t) b(s_{t-1}) ds_{t-1}}{\int_S p(s_t^{[i]}|s_{t-1}, a_t) b(s_{t-1}) ds_{t-1}} = \eta\, p(o_t|s_t^{[i]}, a_t) \tag{6}$$

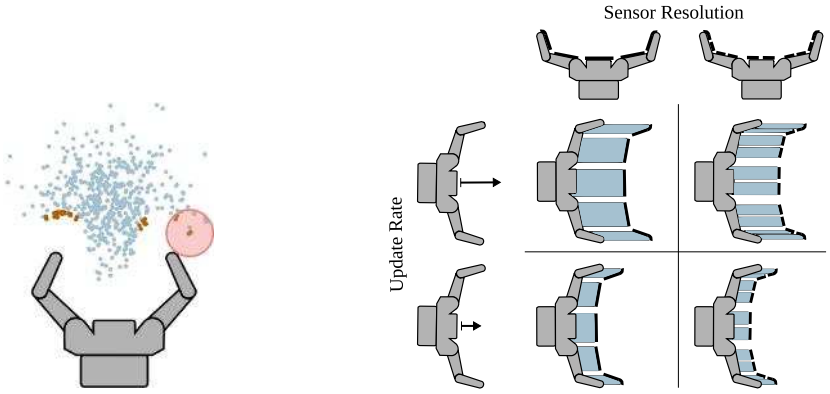
to compensate for the mismatch between the proposal and target distributions. In the general case, where S_{t-1} has non-uniform weights, the weight for particle $s_t^{[i]}$ is given by $w_t^{[i]} = \eta\, w_{t-1}^{[i]} p(o_t|s_t^{[i]}, a_t)$ where $w_{t-1}^{[i]}$ is the weight of the particle $s_{t-1}^{[i]}$ that was forward-simulated by the transition model. The re-weighting step incorporates the observation o_t by assigning higher weights to particles that are consistent with the observation.

4.4 Particle Starvation During Contact

The CPF is agnostic to the observation model and has been applied to a variety of domains (Montemerlo et al. 2003, Gadeyne et al. 2005, Zhang & Trinkle 2012). However, the contact manipulation problem is unique because: (1) the state may become concentrated on the lower-dimensional contact manifold S_c and (2) contact sensors accurately discriminate between contact and no-contact.

During periods of contact observations are discriminative and the observation model $p(o_t|s_t, a_t)$ is peaked on S_o. Since S_o is a lower-dimensional manifold, the set of observations with non-trivial probability form a zero measure set. As a result, the conventional proposal distribution (Equation 5) is a poor approximation of the target distribution (Equation 3) during contact; i.e. no particles in S_t will agree with o_t with high probability.

In practice, the particle filter is updated in discrete steps. Executing an action pushes all states that occupy the swept volume of the hand onto the contact manifold. As a result, the hand's contact sensors gain full dimensionality and the CPF is not completely ineffective. Unfortunately, as Figure 5a shows, the CPF



(a) Swept Volume of the Hand (b) Spatio-Temporal Sensor Resolution

Figure 5: (a) Only the small number of particles (dark orange) that are in the swept volume of the sensors generate contact observations. Most particles (light blue) generate no-contact observations. Therefore, the conventional proposal distribution performs poorly during contact. The light orange circle shows the geometry of the object for one particular particle. (b) Increasing the sensor's resolution or update rate reduces the swept volume of the sensors. This exacerbates the problem of particle starvation.

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 starvation* during periods of contact: there are often no particles in the vicinity of the true state.

Figure 6-Top shows the effect of particle starvation 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, $b(s_t)$ is concentrated on S_o and importance sampling fails to accurately represent the distribution. As a result, the CPF converges to the erroneous belief that the box has rolled off the finger tip instead of settling into the palm.

Surprisingly, this effect causes the CPF to *perform worse as sensor resolution or the update frequency increases*. We illustrate the reason for this unintuitive result in Figure 5b and demonstrate this effect occurs in simulation experiments (Section 7.5). As sensor resolution increases (left-to-right), the swept volume of each sensor becomes narrower. As the update frequency increases (top-to-bottom), the distance traveled by the hand between updates decreases, and the swept volume becomes shorter. As a result, the particle filter requires a large number of particles to successfully track the state.

5 Manifold Particle Filter

Suppose the state space S is partitioned into m disjoint components $M = \{M_j\}_{j=1}^m$ such that $\bigcup_{j=1}^m M_j = S$ and $M_i \cap M_j = \emptyset$ for $i \neq j$. In this situation, we can express the belief state as the weighted sum

$$b(s_t) = \sum_{j=1}^{m} b(s_t|M_j) b(s_t \in M_j) \tag{7}$$

over the components $j = 1, \ldots, m$. The term $b(s_t|M_j)$ is the conditional belief over M_j and $b(s_t \in M_j)$ is the marginal belief that s_t is on component M_j.[4] This factorization is motivated by the the case where $M_1, \ldots, M_{m-1}$ are lower-dimensional manifolds and $M_m = S \setminus \bigcup_{j=1}^{m-1} M_j$ is the remaining ambient space.

The MPF, shown in Algorithm 2, represents $b(s_t)$ using a single set of particles $S_t \sim b(s_t)$. After taking action $a_t \in A$ and receiving observation $o_t \in O$, the MPF uses importance sampling to recursively construct S_t from $S_{t-1} \sim b(s_{t-1})$ (Algorithm 2, Lines 4 and 5) just like the CPF. The key insight behind the MPF is to factor the belief state across manifolds, as shown in Equation 7, and perform a separate importance

[4]We use the notation $b(s_t|M_j)$ to denote the probability distribution $b(s_t)$ restricted to M_j. Formally, $b(s_t|M_j) = \eta' b(s_t)$ for $s_t \in M_j$ and $b(s_t|M_j) = 0$ for $s_t \notin M_j$. The normalization factor $\eta' = [\int_{M_j} b(s_t) ds_t]^{-1}$ is chosen such that $\int_{M_j} b(s_t|M_j) ds_t = 1$.

Algorithm 2 Manifold Particle Filter

Input: action $a_t \in A$ and observation $o_t \in O$
Input: proposal distribution $q(s_t|M_j)$ and number of samples n_j for $j = 1, \ldots, m$
Input: particles $S_{t-1} = \{\langle s_{t-1}^{[i]}, w_{t-1}^{[i]} \rangle\}_{i=1}^{n}$ from time $t-1$ such that $S_{t-1} \sim b(s_{t-1})$
Output: particles $S_t = \{\langle s_t^{[i]}, w_t^{[i]} \rangle\}_{i=1}^{n}$ at time t such that $S_t \sim b(s_t)$

1: **for** $j = 1, \ldots, m$ **do**
2: $\quad S_t^{M_j} \leftarrow \emptyset$
3: $\quad$ **for** $i = 1, \ldots, n_j$ **do**
4: $\quad\quad s_t^{M_j[i]} \sim q(s_t|M_j)$
5: $\quad\quad w_t^{M_j[i]} \leftarrow b(s_t|M_j)/q(s_t|M_j)$
6: $\quad\quad S_t^{M_j} \leftarrow \left\{ \langle s_t^{M_j[i]}, w_t^{M_j[i]} \rangle \right\} \cup S_t^{M_j}$
7: $\quad$ **end for**
8: **end for**
9: $\hat{S}_t \leftarrow \sum_{j=1}^{m} b(s_t \in M_j) S_t^{M_j}$
10: $S_t \leftarrow \text{Resample}(\hat{S}_t)$

sampling step for each manifold $M_j \in M$. This factorization enables the MPF to use a different sampling technique to sample from each conditional belief $b(s_t|M_j)$, which may tailored to the particular structure of M_j. Finally, the MPF combines the sets of samples drawn from each manifold to form S_t (Algorithm 2, Line 9).

Let $S_t^{M_j} = \{\langle s_t^{M_j[i]}, w_t^{M_j[i]} \rangle\}_{i=1}^{n_j}$ be the set of n_j particles that we sample from manifold $M_j \in M$.[5] First, we sample the state of each particle $s_t^{M_j[i]} \sim q(s_t|M_j)$ from the manifold-dependent proposal distribution $q(s_t|M_j)$ (Algorithm 2, Line 4). Then, we compute the corresponding importance weight $w_t^{M_j[i]} = b(s_t|M_j)/q(s_t|M_j)$ as the ratio of the target distribution to the proposal distribution (Algorithm 2, Line 5). Just as before, we make no assumption about $q(s_t|M_j)$ except that $b(s_t|M_j) > 0 \implies q(s_t|M_j) > 0$.

Finally, we construct a unified set of particles $S_t \sim b(s_t)$ from the m individual sets of particles $S_t^{M_j} \sim b(s_t|M_j)$ (Algorithm 2, Line 9). We do so by computing the mixture

$$S_t = \sum_{j=1}^{m} b(s_t \in M_j) S_t^{M_j},$$

where the sum $aX + cY$ of the sets of particles $X = \{\langle x^{[i]}, w_x^{[i]} \rangle\}_{i=1}^{n_x}$ and $Y = \{\langle y^{[i]}, w_y^{[i]} \rangle\}_{i=1}^{n_y}$ with non-negative scale factors $a, c \in \mathbb{R}^{\geq 0}$ is defined to be $aX + cY = \{\langle x^{[i]}, a w_x^{[i]}/W_x \rangle\}_{i=0}^{n_x} \cup \{\langle y^{[i]}, c w_y^{[i]}/W_y \rangle\}_{i=0}^{n_y}$. The variables $W_x = \sum_{i=1}^{n_x} w_x^{[i]}$ and $W_y = \sum_{i=1}^{n_y} w_y^{[i]}$ denote the total weight of X and Y, respectively. Since each set $S_t^{M_j} \sim b(s_t|M_j)$ is individually distributed according to the conditional belief, the mixture $S_t \sim b(s_t)$ is distributed according to the target distribution.

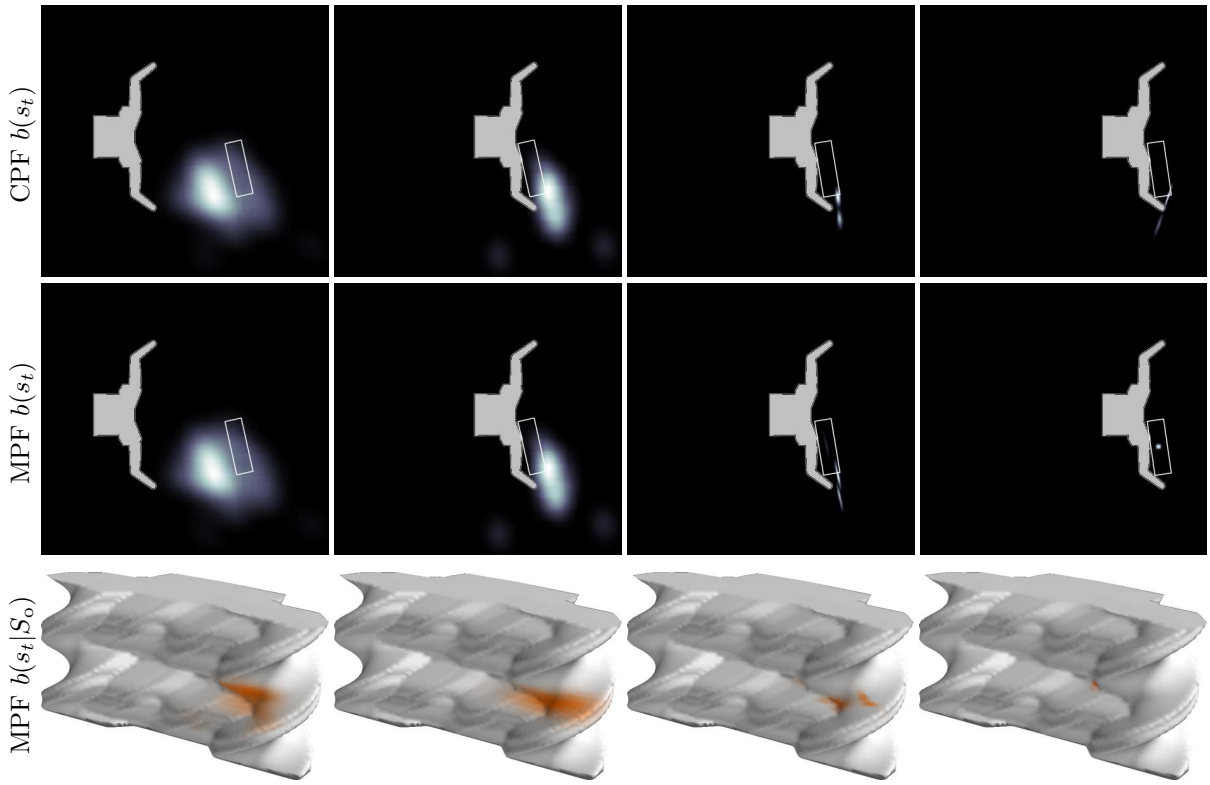
5.1 Marginal Belief Over Manifolds

In order to reconstruct the full belief $b(s_t)$ using Equation 7, we must also know the marginal belief $b(s_t \in M_j)$ over manifolds. Ideally, we would compute

$$b(s_t \in M_j) = \int_{M_j} b(s_t) ds_t,$$

by marginalizing the current belief state $b(s_t)$ over each manifold. Unfortunately, computing this marginal requires knowledge of $b(s_t)$: precisely the distribution that we are trying to estimate!

[5] In this section we assume $b(s_t \in M_j) > 0$ to insure that the conditional belief $b(s_t|M_j)$ is defined. When $b(s_t \in M_j) = 0$, we simply do not sample from M_j and assign $S_t^{M_j} = \emptyset$. See Section 9.4 for a unified measure-theoretic treatment of this case.



(a) Prior Belief (b) Pre-Contact (c) Post-Contact (d) Final Belief

Figure 6: Snapshots of the CPF and MPF, using analytic representation of the contact manifold, during execution. Unlike the CPF, the MPF avoids particle starvation by explicitly tracking the probability distribution on the observable contact manifold S_o.

It may seem reasonable to approximate $b(s_t \in M_j) \approx \int_{M_j} b(s_{t-1})ds_{t-1}$ by computing the marginal over the previous belief state $b(s_{t-1})$. This, however, is not the case: this approximation treats $b(s_t \in M_j)$ as a stationary distribution and performs poorly when probability mass transitions between manifolds: one of the same situations that cause the CPF performs poorly.

Instead, we rely on domain-specific structure of the problem to estimate the marginal. In the case of contact manipulation, we use the discriminative nature of contact sensors to estimate the marginal (Section 6.1).

5.2 Number of Particles Per Manifold

Our above analysis made no assumption about the number of particles n_j sampled from each manifold. It is generally advisable choose $n_j \approx n \cdot b(s_t \in M_j)$ to avoid sampling too many particles in low-probability manifolds. This strategy is equivalent to using systematic sampling to first sample a manifold $M_j \sim b(s_t \in M_j)$ for each particle, then using importance sampling to sample the particle from the corresponding conditional belief $b(s_t|M_j)$.

However, if domain-specific knowledge is available, it may be desirable to manually specify the number of particles n_j sampled from M_j. We demonstrate one example of this technique in Section 6.5. In this case, the set of particles produced by the MPF will be of size $|S_t| = \sum_{j=1}^{m} n_j$. If $|S_t| > |S_{t-1}|$, then each update of the particle filter will require additional memory and computation time to complete. To avoid this, we enforce the invariant $|S_t| = |S_{t-1}|$ by resampling S_t with replacement (Algorithm 2, Line 10).

6 Manifold Particle Filter for Contact Manipulation

In this section we apply the MPF to the contact manipulation problem by defining the observable contact manifold $M_1 = S_o$ and the ambient space $M_2 = S \setminus S_o$ as the relevant partition of S. We show that, given this partition, it is possible to compute $b(s_t \in M_j)$ using the discriminative nature of contact sensors (Section 6.1).

Given this partition, the MPF uses the conventional proposal distribution to sample from $S \setminus S_o$ and the dual proposal distribution (Section 6.2) to sample from S_o. We propose three representations of the observable contact manifold (Section 6.3) that can be used to implement the dual proposal distribution. We also present a technique that uses kernel density estimation for approximating the dual importance weights (Section 6.4). Finally, we show how to efficiently mix particles from the CPF and MPF to achieve better performance than either the CPF or the MPF in isolation (Section 6.5).

Figure 6 shows the performance of the MPF and the CPF on the same stream of actions and observations. Before contact (a)–(b), $b(s_t \in S_o) \approx 0$ and both filters update using the conventional proposal distribution. After contact (c)–(d), $b(s_t \in S_o) \approx 1$ and the manifold particle filter samples from the dual proposal distribution. Sampling from this distribution allows the MPF to accurately track the object's pose during persistent contact.

6.1 Discriminative Observation Model

Contact sensors accurately discriminate between contact and no-contact. An observation model is *discriminative* if it has a low probability ϵ of generating false-positive or false-negative observations of contact. Formally, we call an observation model discriminative if we can partition the set of observations O into sets of contact $O_c \subseteq O$ and no-contact $O_{nc} = O \setminus O_c$ observations such that $\Pr(o \in O_c | s_t \in S_o, a_t) > 1 - \epsilon$ during contact and $\Pr(o_t \in O_{nc} | s_t \notin S_o, a_t) > 1 - \epsilon$ during no-contact.

If a sensor is perfectly discriminative, i.e. $\epsilon = 0$, then the marginal

$$b(s_t \in M_j) = \int_{M_j} b(s_t) ds_t = \eta \int_{M_j} p(o_t | s_t, a_t) \int_S p(s_t | s_{t-1}, a_t) b(s_{t-1}) ds_{t-1} ds_t$$

is binary because $s_t \in O_{nc} \implies p(o_t | s_t, a_t) = 0 \; \forall s_t \in S_o$ and $s_t \in O_c \implies p(o_t | s_t, a_t) = 0 \; \forall s_t \notin S_o$. As a result, the MPF samples entirely from the dual proposal distribution during periods of contact. Otherwise, the MPF samples from the conventional proposal distribution.

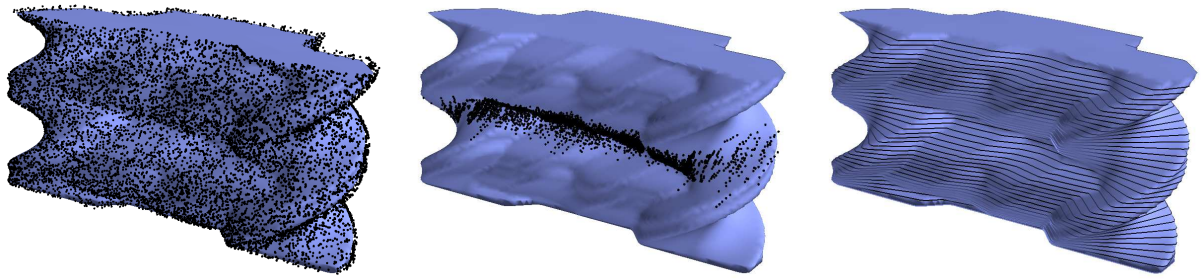
For small values of $\epsilon > 0$, we approximate the marginal by the probability $b(s_t \in S_o) \propto \int_{S_o} p(o_t | s_t, a_t) ds_t$ of the single most recent observation o_t. This approximation is equivalent to ignoring the history encoded in $b(s_{t-1})$ while computing $b(s_t \in S_o)$. This is a reasonable approximation of the true marginal for the first few critical post-contact timesteps, but accumulates bias over time. We suggest two potential solutions to this problem in Section 9.5.3.

We make no assumptions about the observation model during contact, i.e. $p(o_t | s_t, a_t)$ for $s_t \in S_o$. This distribution models the information provided by the sensors while contact is being observed. In the case of binary sensors, such as those used in Sections 7 and 8, $p(o_t | s_t, a_t)$ is uniform over the set of states that are in non-penetrating contact with the active sensors. In the case of a more sophisticated sensor, like a six-axis force/torque sensor, this distribution encodes a non-uniform probabilistic model of the sensor.

6.2 Dual Proposal Distribution

When $s_t \in S_o$ we know that the conventional proposal distribution is a poor approximation for the posterior and particle starvation will occur. Instead, we sample from the *dual proposal distribution* (Thrun, Fox & Burgard 2000)

$$q(s_t) = \frac{p(o_t | s_t, a_t)}{p(o_t | a_t)} \tag{8}$$



(a) Rejection Sampling (RS) (b) Trajectory Rollouts (TR) (c) Analytic Representation (AM)

Figure 7: Three representations of the contact manifold. The (a) rejection-sampled and (b) trajectory rollout representations approximate S_c with discrete sets of samples. The (c) analytical representation explicitly solves for exact orientation iso-contours of the manifold.

to generate a sample $s_t^{[i]} \sim q(s_t)$ that is consistent with the latest observation o_t.[6] As in prior work (Thrun, Fox, Burgard & De 2000), we assume that $p(o_t|a_t)$ is finite. Sampling this proposal distribution is non-trivial and may require domain-specific knowledge (Thrun, Fox & Burgard 2000). In the case of contact manipulation, we build an approximate representation of the observable contact manifold (Section 6.3) to facilitate this sampling.

Just as before, we can find the corresponding *dual importance weights*

$$w_t^{[i]} = \frac{b(s_t^{[i]})}{q(s_t^{[i]})} = \frac{\eta\, p(o_t|s_t^{[i]}, a_t) \int_S p(s_t^{[i]}|s_{t-1}, a_t) b(s_{t-1}) ds_{t-1}}{p(o_t|s_t^{[i]}, a_t)/p(o_t|a_t)} = \eta' \int_S p(s_t^{[i]}|s_{t-1}, a_t) b(s_{t-1}) ds_{t-1}, \qquad (9)$$

with normalization factor $\eta' = \eta\, p(o_t|a_t)$. We obtained this equation by dividing the target distribution (Equation 3) by the proposal distribution (Equation 8). We discuss how to approximate these weights using kernel density estimation in the next section (Section 6.4).

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. Sampling from the dual proposal distribution is effective when $p(o_t|s_t, a_t)$ is peaked around the true state (Thrun, Fox & Burgard 2000).

6.3 Representing the Contact Manifold

Sampling a particle $s_t^{[i]} \sim q(s_t)$ requires generating particles that lie on the observable contact manifold S_o. To do so, we compute an approximate representation $\tilde{S}_o \approx S_o$ of the observable contact manifold as a pre-computation step. Then, at runtime, we sample from a distribution over $\tilde{S}_o$ weighted by $p(o_t|s_t, a_t)/p(o_t|a_t)$.

We describe three possible representations of $\tilde{S}_o$. Two of these, the rejection sampling (Section 6.3.1) and trajectory rollout (Section 6.3.2) representations, approximate S_o with large set of discrete samples $\tilde{S}_o \approx S_o$. The third technique (Section 6.3.3) takes advantage of additional structure in object-hand geometry to solve for a continuous, analytic representation of S_o.

6.3.1 Rejection Sampling

The most straightforward way of sampling from $S_o \subset S$ is by rejection sampling from the ambient space S. *Rejection sampling* iteratively samples candidate states $s^{[i]} \sim \text{uniform}(S)$ until it finds a sample $s^{[i]} \in S_o$ in the desired set. Using this technique, we can generate a large set of samples $\tilde{S}_o = \{s^{[i]}\}_{i=1}^n \subset S_o$ that densely cover S_o in a pre-computation step. At runtime, we sample from the discrete set $\tilde{S}_o$ weighted by $p(o_t|s_t, a_t)/p(o_t|a_t)$.

[6] Since we are only using the dual proposal distribution to sample from S_o, the proposal distribution needs to be restricted to S_o. We intentionally omit the conditioning notation adopted in Section 5 for the remainder of this section for simplicity.

Unfortunately, rejection sampling fails for the same reason as the conventional particle filter: S_o is a measure-zero set and there is zero probability of successfully sampling an $s^{[i]} \in S_o$. Instead, we rejection sample the set

$$\tilde{S}_o = \left\{ s \in S : \min_{p_s \in P_s, p_o \in P_o(s)} ||p_s - p_o|| \leq \epsilon \right\}$$

of object configurations that are within distance $\epsilon \in \mathbb{R}^+$ of the hand. The set $\tilde{S}_o$ is a reasonable approximation for S_o when ϵ is on the same order of magnitude as the numerical inaccuracies of the motion and observation models.

Figure 7a shows S_o covered by a set of 10,000 rejection-sampled configurations $\tilde{S}_o$ of the BarrettHand in contact with the rectangular box shown in Figure 3. The samples $\tilde{S}_o$ are not exactly on S_o and are distributed uniformly over the ambient space S, instead of uniformly across the surface of the manifold. This is, in most cases, an acceptable approximation for a true uniform distribution over S_o.

Sampling from the dual proposal distribution is implemented by importance sampling from the set $\tilde{S}_o$ using the weights given in Section 6.4. In the worst case, generating these samples requires evaluating the importance weight of all $|\tilde{S}_o|$ particles. Our experimental results (Sections 7 and 9.2) suggest that the computational cost of evaluating the dual importance weights is insignificant compared to other parts of the algorithm. This can be further reduced to sub-linear complexity using a spatial index, e.g. k-d tree (Bentley 1975), if the kernel used to compute importance weights has finite support.

6.3.2 Trajectory Rollouts

Rejection sampling attempts to densely cover all of S_o with samples that are independent of the prior belief $b(s_0)$. As a result, many of the samples generated by rejection sampling will be found in regions of S_o that remain low (or, in the extreme case, zero) probability during execution. We can exploit this structure by concentrating more samples in the regions of S_o that we are likely to encounter during execution.

We can generate samples $\tilde{S}_o$ that are biased towards these regions by performing trajectory rollouts from the initial belief $b(s_0)$. We begin by sampling a particle from the prior $s_0^{[i]} \sim b(s_0)$. Next, we forward-simulate the particle for T steps using the motion model $s_t^{[i]} \sim p(s_t|s_{t-1}, a_t)$ with $a_t \sim \pi(b(s_t))$ chosen according to the same policy π that will be run during execution.[7] Finally, we add any $s_t^{[i]} \in S_o$ to $\tilde{S}_o$. This process repeats until $|\tilde{S}_o|$ reaches the desired size.

Figure 7b shows 10,000 samples taken from 2000 trajectory rollouts with a fixed "move straight" action and $b(s_0)$ 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 $b(s_0)$—at the cost of little-to-no coverage of the rest of the manifold.

Unfortunately, the non-uniformity of our samples means that $\tilde{S}_o$ is biased towards absorbing regions of the state space; e.g. the configurations where the object rests stably against the hand in Figure 4c. We compensate for this bias through importance sampling: we assign each $s^{[i]} \in \tilde{S}_o$ an importance weight $p(o|s,a)/[p(o|a)\tilde{p}(s)]$ where $\tilde{p}(s)$ is the density of $\tilde{S}_o$ at s. We estimate $\tilde{p}(s)$ using a standard kernel density estimation technique (Rosenblatt 1956) on $\tilde{S}_o$.[8] Once these weights have been computed, we use the same technique as described in Section 6.3.1 to sample from the dual proposal distribution.

6.3.3 Analytic Representation

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

[7] If the initial belief $b(s_0)$ is not known, then we may substitute an alternative belief $\tilde{b}(s_0)$ where $b(s_0) > 0 \Rightarrow \tilde{b}(s_0) > 0$. Similarly, if π is not known, we may substitute an alternative policy $\tilde{\pi}$ where $\pi(b) > 0 \Rightarrow \tilde{\pi}(b) > 0$. These conditions guarantee that the policy used for constructing $\tilde{S}_o$ will eventually visit all states that can be encountered at runtime.

[8] Note that $w_t^{[i]}$ is undefined if $\tilde{p}(s) = 0$. This cannot happen if we choose a kernel with sufficiently broad support.

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

$$S_{\text{obs}}(\theta) = P_{\text{h}} \oplus -P_{\text{o}}\left([0, 0, \theta]\right)$$

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 $P_{\text{o}}([0, 0, \theta])$ are polygonal, $S_{\text{obs}}(\theta)$ is also polygonal and can be computed in closed form via a convolution of P_{h} and $P_{\text{o}}([0, 0, \theta])$ (Wein 2013). The contact manifold $S_{\text{c}}(\theta)$ at orientation θ simply consists of the line-string boundary of the polygon $S_{\text{obs}}(\theta)$. Figure 7c shows several θ-isocontours of S_{c} superimposed over a high-resolution polyhedral approximation of the contact manifold. The same process can be repeated with P_{h} and P_{s} to construct an analytic representation of $S_{\text{o}}(\theta)$.

Finally, we approximate the observable contact manifold as a union $\tilde{S}_{\text{o}} = \bigcup_{\theta \in \Theta} S_{\text{o}}(\theta)$ over a large, discrete set of orientations Θ.[9] Discretizing θ approximates S_{o} with a polyhedron $\tilde{S}_{\text{o}}$ that shares the same iso-contours as S_{o} at all $\theta \in \Theta$.

Sampling an $s^{[i]} \sim \tilde{S}_{\text{o}}$ is possible by first sampling a $\theta \in \Theta$, then uniformly sampling an $s^{[i]}$ from our analytical representation of $S_{\text{o}}(\theta)$. Alternatively, one could sample from an approximate, polyhedral representation of S_{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 S_{o}.

6.4 Dual Importance Weights

Regardless of the method we use to sample from the dual proposal distribution, we must weight each sample $s_t^{[i]}$ with its corresponding importance weight $w_t^{[i]} = \int_S p(s_t|s_{t-1}, a_t) b(s_{t-1}) ds_{t-1}$. Intuitively, the importance weight integrates our prior belief $b(s_{t-1})$ and the effect of taking action a_t into $b(s_t)$ (Thrun, Fox & Burgard 2000). This is the logical dual of the conventional importance weights, which serve to integrate the observation o_t into the posterior (Section 4.3).

We evaluate $w^{[i]}$ by forward propagating each particle $s_{t-1}^{[i]} \in S_{t-1}$ from time $t-1$ to time t using the transition model $s_t^{[i]} \sim p(s_t^{[i]}|s_{t-1}^{[i]}, a_t)$. This set of samples, which we denote by S_{t-1}^{+}, is distributed according to our belief state after taking action a_t, but before receiving the next observation o_t. Then, we use S_{t-1}^{+} to approximate the importance weight $w^{[i]} = \int_S p(s_t|s_{t-1}, a_t) b(s_{t-1}) ds_{t-1}$ using a density estimation technique (Thrun, Fox & Burgard 2000).

Ideally, we would compute a density estimate over the manifold S_{o}. Unfortunately, while there has been some work on density estimation on Riemannian manifolds (Pelletier 2005), it is difficult to apply these techniques to the approximate and sample-based representations of S_{o} described above. This is exacerbated by the fact that many of our forward-simulated particles will not lie precisely on S_{o}.

Instead, we use kernel density estimation (Rosenblatt 1956) to promote S_{t-1}^{+} into a full-dimensional distribution over S and evaluate $w_t^{[i]}$ using the density estimate over the full space. The belief given by our forward propagated particles $S_{t-1}^{+} = \langle s_{t-1,+}^{[i]}, w_{t-1,+}^{[i]} \rangle_{i=1}^{n}$ is

$$b(s_{t-1,+}) \approx \sum_{i=1}^{n} w_{t-1,+}^{[i]} K\left(s_{t-1,+} - s_{t-1,+}^{[i]}\right),$$

where $K = \delta(\cdot)$ is the Dirac delta function. This distribution has discrete support because $b(s_t) = 0$ for all $s_t \notin S_{t-1}^{+}$. Applying kernel density estimation to S_{t-1}^{+} replaces $\delta(\cdot)$ with a kernel function $K(\cdot)$ with broad support; e.g. an Epanechnikov (Epanechnikov 1969) or Gaussian kernel. This allows us to evaluate $b(s_{t-1,+})$ for the particles S_t that we sampled from the dual proposal distribution.

In practice, we choose $K(\cdot)$ to be a Gaussian kernel and select the bandwidth matrix using a multivariate generalization of Silverman's rule of thumb (Silverman 1981). Our estimate is effectively restricted to S_{o} because it is only evaluated on the samples drawn from the dual proposal distribution. Figure 6 shows an example of the resulting density estimate over S_{free} (Figure 6-Middle) and S_{o} (Figure 6-Bottom) computed using this technique.

[9] 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).

6.5 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 (Thrun, Fox & Burgard 2000). The MPF uses the dual proposal distribution to sample from S_o and, as a result, shares the same weakness.

We use a *mixture proposal distribution* (Thrun, Fox & Burgard 2000) to mitigate this effect by combining both sampling techniques. Instead of sampling all of the particles from the MPF, we sample some particles $|S_{\text{cpf}}| = n$ from the CPF and the remaining particles $|S_{\text{mpf}}| = d$ from the MPF. The mixture proposal distribution combines the two sets of particles with the weighted sum $(1-\phi)S_{\text{cpf}} + \phi S_{\text{mpf}}$ with $0 \leq \phi \leq 1$ before resampling.[10]

We seamlessly combine the mixture proposal distribution and the MPF's manifold mixing step (Algorithm 2, Line 9) into a single update. To do so, we rewrite the mixture proposal distribution as

$$\begin{aligned} S_t &= (1-\phi)S_{\text{cpf}} + \phi\left[b(s_t \in M_1)S_t^{M_1} + b(s_t \in M_2)S_t^{M_2}\right] \\ &= (1-\phi)S_{\text{cpf}} + \phi\left[b(s_t \in M_1)S_t^{M_1} + b(s_t \in M_2)(S_{\text{cpf}} \cap M_2)\right] \\ &= (1-\phi)(S_{\text{cpf}} \cap M_1) + \left[1-\phi+\phi\, b(s_t \in M_2)\right](S_{\text{cpf}} \cap M_2) + \phi\, b(s_t \in M_1)S_t^{M_1} \end{aligned}$$

by partitioning S_{cpf} into the particles $S_{\text{cpf}} \cap M_1$ on the observable contact manifold $M_1 = S_o$ and those $S_{\text{cpf}} \cap M_2$ in free space. This factorization is possible because both $S_{\text{cpf}} \cap M_2$ and $S_t^{M_2}$ are both generated by sampling from the conventional proposal distribution. This combined update rule can be interpreted as assigning additional weight to the particles $S_{\text{cpf}} \cap M_2$ in the ambient space to avoid biasing S_t towards S_o.

The parameters n and d can be interpreted as the minimum number of samples necessary to cover the high-probability regions of, respectively, the ambient space $M_2 = S \setminus S_o$ and the observable contact manifold S_o. The *mixing rate* $0 \leq \phi \leq 1$ parameter allows the algorithm to smoothly transition between the CPF ($\phi = 0$) to the MPF ($\phi = 1$). Increasing ϕ provides better performance when transitioning between manifolds, but only at the cost of becoming more sensitive to erroneous observations (Thrun, Fox & Burgard 2000).

The output of the mixture is a set of $|S_t| = n+d$ particles distributed according to the target distribution (Equation 3). Then, as described in Section 5, we resample S_t with replacement to enforce the invariant that $|S_t| = |S_{t-1}| = n$. This invariant is critical to ensure the MPF, just like the CPF, can be recursively updated without increasing in computational complexity.

Due to this resampling, the MPF fundamentally differs from related particle filtering techniques (Gadeyne et al. 2005, Meeussen et al. 2007) that track a distribution over contact formations (Xiao 1993). In the general case, estimating the distribution $b(s_t \in M_j)$ over manifolds is as hard as solving the filtering problem itself (Section 5.1). In this paper we specifically consider the case where contact sensors are discriminative (Section 6.1) and $b(s_t \in M_j)$ is binary. In future work, we are interested in using these complementary techniques to estimate to estimate $b(s_t \in M_j)$ for non-discriminative sensors.

The MPF uses this estimate of $b(s_t \in M_j)$ to maintain a single set of particles that span all manifolds. It is not meaningful to identify whether a particular particle was sampled from the "conventional" or "dual" proposal distribution since they are seamlessly mixed as part of the same posterior distribution.

7 Simulation Experiments

We designed a set of simulation experiments to compare the MPF with the CPF for the state estimation for contact manipulation problem (Sections 7.4 and 7.5). We also ran experiments to explore the differences between the three representations of the contact manifold (Section 7.6) and the effect of the mixing rate parameter (Section 7.7).

7.1 Hypotheses

Based on the particle starvation problem described in Section 4.4, we hypothesize that:

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

[10] We use the notation defined in Section 5 to represent the mixing of multiple sets of particles.

Increasing the sensor resolution or update rate should make this difference more pronounced because it reduces the swept volume of the sensors. As this happens, the CPF will begin to suffer from particle starvation. Therefore, we hypothesize:

H2. *The CPF will perform worse as sensor resolution increases; the MPF will not.*

H3. *The CPF will perform worse as the sensor update rate increases; the MPF will not.*

All of the above hypotheses (H1–H3) should be true regardless of which representation of the contact manifold is used by the MPF. Since AM faithfully represents the continuous manifold, we hypothesize:

H4. *The analytic representation of the contact manifold will perform best.*

Surprisingly, our results suggest that H4 is false: TR outperforms AM despite the fact that it is a sample-based approximation of the true contact manifold. We discuss a possible explaination of this result in Section 7.6.

Between the sample-based representations, we expect TR to outperform RS. RS attempts to represent the contact manifold at a uniform resolution. In contrast, TR focuses samples on regions of the contact manifold that we are likely to encounter during execution. Therefore, we hypothesize:

H5. *The trajectory rollout representation will outperform the rejection sampled representation.*

7.2 Experimental Design

We implemented the CPF and MPF 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 $b(s_0) = \mathcal{N}(\bar{s}_0, \Sigma)$ with covariance $\Sigma^{1/2} = \text{diag}[5\text{ cm}, 5\text{ cm}, 20°]$. The mean $\bar{s}_0 = (\bar{x}_0, \bar{y}_0, \bar{\theta}_0)$ was placed a fixed distance $\bar{x}_0 = 20$ cm from the hand and was assigned a random lateral offset $\bar{y}_0 \sim \text{uniform}[-10\text{ cm}, 10\text{ cm}]$ and orientation $\bar{\theta}_0 \sim \text{uniform}[0°, 360°]$ for each trial.

We simulated the motion of the object using a penetration-based quasistatic physics model (Lynch et al. 1992) with a 1 mm step size. Before each step, the finger-object coefficient of friction μ_f and the radius of the object's pressure distribution c were sampled from the Gaussian distributions $\mu_\text{f} \sim N(0.5, 0.2^2)$ and $c \sim N(0.05, 0.01^2)$ truncated to enforce $\mu_\text{f}, c > 0$. Binary observations were simulated for each of the hand's sensors, which were uniformly distributed across the front surface of the hand, by computing the intersection of each sensor with the object. Observations were assumed to be perfectly discriminative, but the observation model had a 10% chance of generating an incorrect observation during contact; i.e. an incorrect sensor would fire. These observations were simulated by applying the same observation model to a special "ground truth" particle s_t^* sampled from $s_0^* \sim b(s_0)$.

7.3 Dependent Measures

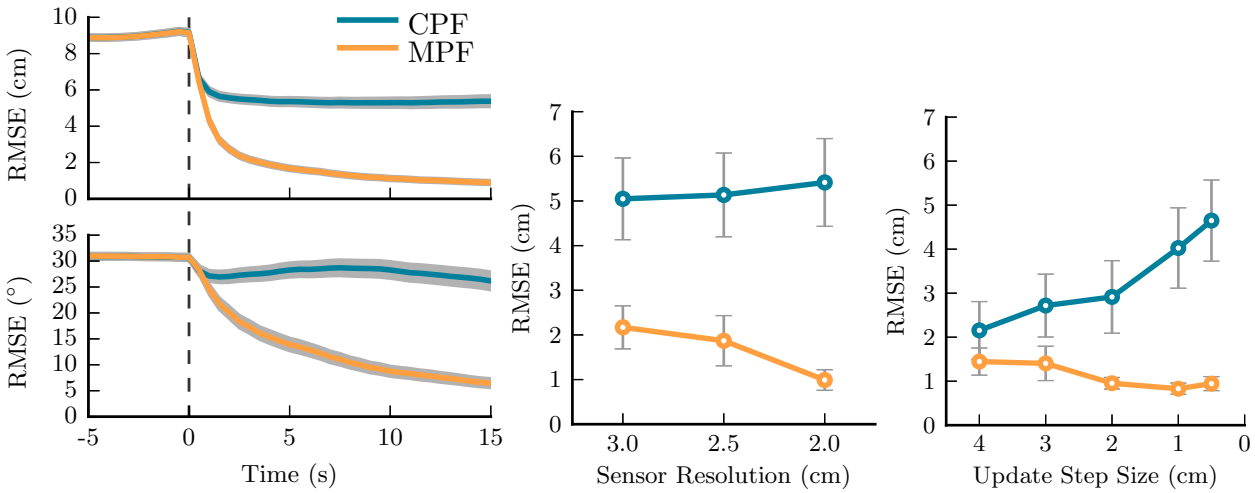
We evaluate the performance of an estimator by computing the *root mean square error*

$$\text{RMSE}(S_t, s_t^*) = \sqrt{\frac{\sum_{i=1}^{n} (s_t^{[i]} - s_t^*)^2 w_t^{[i]}}{\sum_{i=1}^{n} w_t^{[i]}}}$$

of the particles $S_t = \{\langle s_t^{[i]}, w_t^{[i]} \rangle\}_{i=1}^n$ with respect to the true state s_t^* at time t. Instead of combining the position error (measured in centimeters) with the orientation error (measured in degrees), we report separate RMSE values for position and orientation.

7.4 Conventional vs. Manifold Particle Filter (H1)

We ran the CPF with $n = 100$ particles and the MPF with $n = 100$ conventional, $d = 25$ dual particles, and a mixing rate of $\phi = 0.1$. We intentionally chose the same value of n for both algorithms—despite the addition of d dual particles for the MPF—because the dual sampling step adds negligible overhead to the runtime of



(a) Estimation Error (b) Spatial Resolution (c) Temporal Resolution

Figure 8: Comparison between the CPF and the MPF-AM in simulation. (a) The CPF and MPF perform identically before contact, but the MPF significantly outperforms the CPF post-contact. (b) The MPF improves as spatial sensor resolution increases, whereas the CPF declines in performance. (c) Similarly, the MPF improves and the CPF declines when faced with a faster update frequency. Note that resolution improves when moving from left-to-right in (b) and (c). In all cases, error bars indicate a 95% confidence interval.

the algorithm (Section 9.2). This means that both the CPF and MPF are tuned to run at approximately the same update rate. The MPF sampled from an analytic representation of the contact manifold that was pre-computed with 1 mm linear and 1.15° angular resolution.

Figure 8a shows the performance of both filters averaged over 900 trials. These results show that—as expected—both filters behave similarly before contact ($t \leq 0$) and there was not a significant difference in RMSE. After contact ($t > 0$), the MPF quickly achieves 4.4 cm less RMSE than the CPF. Figure 6 shows one example where the MPF achieved a significantly better pose estimate than the CPF. These results support hypothesis H1: the MPF achieves lower post-contact error than the CPF.

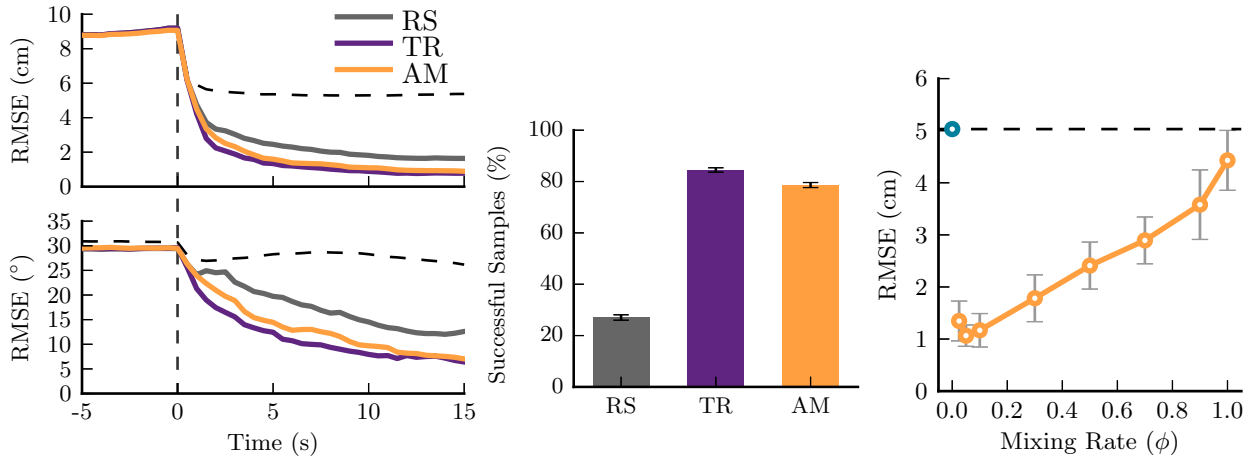
7.5 Spatio-Temporal Sensor Resolution (H2–H3)

We evaluated the effect of sensor resolution on estimation accuracy by varying the resolution of binary contact sensors. In all cases, the sensors are distributed uniformly over the front surface of the hand. Figure 8b shows the relative performance of the CPF and MPF for three different resolutions averaged over 95 trials. As expected, the CPF performs worse as the spatial sensor resolution increases. In contrast, the MPF performs better. This confirms hypothesis H2.

Additionally, we measured the performance of the filters as we varied the distance traveled between sensor updates from 5 mm to 4 cm. Since the hand was moving at a constant velocity, this corresponds to changing the sensor's update frequency. Figure 8c shows the performance of the CPF and MPF averaged over 95 trials. As expected, the CPF performs worse as the update frequency increases. In contrast, the MPF performs better and confirms hypothesis H3.

7.6 Contact Manifold Representation (H4–H5)

We also compared the performance of the MPF using the RS, TR, and AM representations of the observable 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 using the same physics model as used during



(a) Manifold Representations (b) Sampling Success Rate (c) Mixing Rate

Figure 9: (a) 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 $t = 0$. (b) Percent of the time that the MPF succeeded at sampling from the dual proposal distribution during contact. (c) Performance of the MPF-AM as a function of the mixing rate $0 \leq \phi \leq 1$. In all three figures the performance of CPF is plotted as a dotted line and error bars denote a 95% confidence interval.

execution. The AM was built using the parameters described in Section 7.2.

Figure 9a shows the performance of the three representations averaged over 900 trials. The MPF outperformed the CPF with all three representations. As expected, the data supports hypothesis H5: MPF-AM and MPF-TR both outperform MPF-RS. This occurs because the RS representation attempts to cover the entire surface S_o with a coarse set of samples. In contrast, the TR representation focuses the same number of samples on the smaller region of S_o that is encountered during execution.

Surprisingly, however, hypothesis H4 was not supported by the data: MPF-AM did not achieve lower error than MPF-TR representation. This is partially explained by same reasoning as above: the TR representation was able to densely cover the reachable subset of S_o at a resolution indistinguishable from that of the AM representation. Additionally, we know that every sample drawn from the TR representation must be reachable from the initial belief $b(s_0)$. This means that MPF-TR does not waste samples from the dual proposal distribution in regions of S_o that are known to be unreachable from $b(s_0)$.

Our intuition is that the relatively poor performance of the MPF-RS is a result of it frequently failing to sample from the dual proposal distribution. Sampling from the dual proposal distribution fails when all particles sampled from $\tilde{S}_o$ have low probability $p(o_t|s_t, a_t)$ of generating o_t. This occurs when the high-probability regions of $p(o_t|s_t, a_t)$ are not represented by our approximation $\tilde{S}_o$. In the case of binary contact sensors, a sampling failure typically occurs when several sensors are simultaneously active at runtime that were never observed to be simultaneously active while pre-computing $\tilde{S}_o$.

Figure 9b shows the rate of sampling failures for the MPF-AM, MPF-RS, and MPF-TR computed over 900 trials. We formally define a sampling failure as an update where $p(o_t|s_t, a_t) < 0.1$ for all $s_t \in \tilde{S}_o$. Since there is a 10% chance of receiving an erroneous observation during contact (Section 7.2), this corresponds to $\tilde{S}_o$ containing no states that are consistent with o_t. Under this metric, the TR and AM representations fail to sample from the dual proposal distribution for $< 30\%$ of updates. Conversely, the RS representation fails to sample $> 70\%$ of updates. When sampling fails the MPF behaves identically to the CPF and suffers from the same problem of particle starvation. As a result, MPF-RS performs poorly compared to MPF-AM and MPF-TR.

7.7 Mixing Rate

In addition to the manifold representation selected, the mixing rate parameter ϕ also has a strong impact on the performance of the MPF. We repeated the experiments described in Section 7.4 while varying the

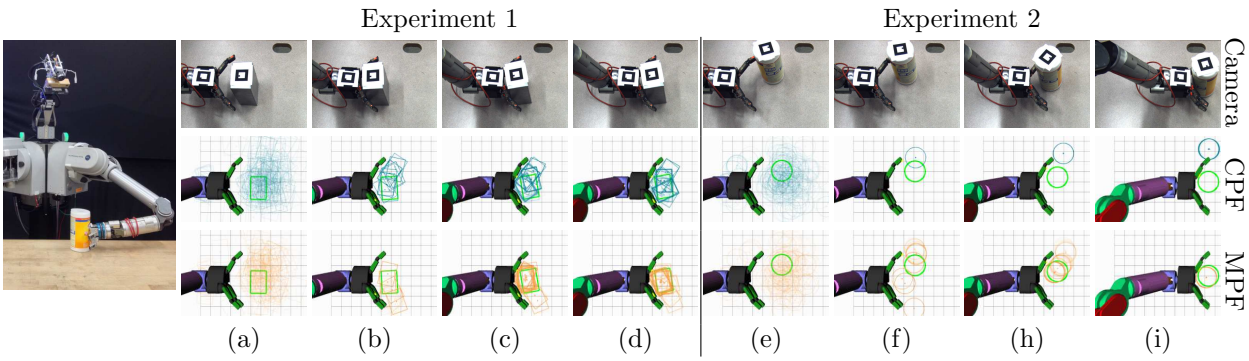


Figure 10: Andy pushing a box (a)–(d) and cylinder (e)–(i) across the table. The top row shows a video of the experiment from an overhead camera. The bottom two rows show the belief state estimated by the CPF (middle, dark blue) and MPF (bottom, light orange) as a cloud of particles. Ground truth is shown as a thick green outline. In both cases, the belief state estimated by the MPF is more accurate than that estimated by the CPF.

MPF-AM's mixing rate over the set $\phi = \{0.025, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$. Note that $\phi = 1$ corresponds to the pure MPF. Figure 9c shows the post-contact performance of the MPF averaged over 150 trials and plotted as a function of ϕ. The performance of the CPF ($\phi = 0$) is plotted as a horizontal dotted line.

As expected, the MPF outperforms the CPF for all $\phi > 0$. Surprisingly, however, the optimal value of ϕ falls into the lowest range of ϕ values $0.025 \leq \phi \leq 0.1$ that we tested. Increasing ϕ out of this range leads to a predictable, linear increase in error. This occurs for two reasons. First, the dual proposal distribution performs poorly when there is observation noise (Thrun, Fox & Burgard 2000). Second, the MPF samples from an approximation of the dual proposal distribution that has higher variance than the true posterior belief. Reducing the mixing rate decreases the rate at which this variance grows. See Section 9.3 for more discussion of this phenomenon.

8 Real-Robot Experiments

We evaluated the CPF and MPF on Andy (Bagnell et al. 2012), a bimanual manipulator developed for the DARPA ARM-S competition. Andy used a Barrett WAM arm (Salisbury et al. 1988) equipped with the i-HY (Odhner et al. 2013) end-effector to push an object across a table. The i-HY's palm (48 tactels), interior of the proximal links (12 tactels each), interior of the distal links (6 tactels each), and fingertips (2 tactels each) were equipped with an array of tactile sensors (Tenzer et al. 2014) based on MEMS barometer technology. The tactels were grouped into 39 vertical stripes to compensate for dead tactels and to simplify the observation model.

Figure 10 shows two representative runs of the state estimator on Andy. The ground-truth pose of the object was tracked by an overhead camera using a visual fiducial. Both filters were run with 250 particles, with $n = 250$ for the CPF and $n = 225$, $d = 25$, $\phi = 0.1$ for the MPF, and were updated after each 5 mm of end-effector motion. With the speed of the arm, this corresponded to an update rate of approximately 5–15 Hz.

In Experiment 1, Andy pushed a metal box that made initial contact with the right proximal link (b) and rolled into the palm (c). The CPF did not have any particles in the small observation space and, thus, failed to track the box as it rolled into the palm (d). The MPF successfully tracked the box by sampling particles that agree with the observation. Note that the MPF was able to exploit the observation of simultaneous contact on the palm and distal link to correctly infer the orientation of the box.

In Experiment 2, Andy pushed a cylindrical container that made initial contact with its left fingertip (e). The cylinder rolled down the distal (f) and proximal (h) links to finally settle in the palm (i). Both the CPF and MPF made use of the initial contact observation to localize the container near the robot's left fingertip. However, the CPF's few remaining particles incorrectly rolled off of the fingertip and outside the hand. The MPF avoided particle starvation near the true state and was able to successfully track the container for the

duration of contact.

9 Discussion

In this section, we discuss the MPF in greater detail, discuss its limitations, and outline several directions for future work. In particular, we analyze the computational complexity of the MPF (Section 9.2) and formally ground the MPF using measure theory (Section 9.4).

9.1 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. 2014) 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 S_{c} and the observable contact manifold $S_{\mathrm{o}} \subseteq S_{\mathrm{c}}$. The geometry of the non-observable region of the contact manifold $S_{\mathrm{c}} \setminus S_{\mathrm{o}}$ impacts the difficulty of the state estimation for contact manipulation problem. Any stable states in that are not in contact with a sensor, e.g. those that come to rest against a flat surface, will accumulate belief when receiving a series of no-contact observations. The only way to sense an object in one of these poses is to perform an action that moves the object out of S_{nc} by pushing it into contact with a sensor.

9.2 Computational Complexity

One practical advantage of the MPF over the CPF becomes apparent when profiling the operations performed by the two algorithms. We will express the complexity of the two algorithms in terms of several basic operations: (1) the number of samples drawn from the transition model, (2) the number of times the observation model was evaluated, (3) number of samples drawn from the contact manifold, and (4) other operations.

Each update of the CPF begins by drawing n samples from the transition model (Table 1a). Then, we compute an importance weight (Table 1b) for each of the n samples by evaluating the observation model. Finally, we must perform $O(n)$ operations to re-sample the particles with unit weight.

Given a fixed number of particles, the MPF both achieves better performance and has higher complexity than the CPF. The MPF samples from the conventional proposal distribution and still requires drawing n samples from the transition model and evaluating the observation model n times. Additionally, the MPF samples d from the contact manifold (Table 1c). Computing the importance weights for these particles (Table 1d) involves evaluating an $O(n)$ kernel density estimate for each of the d particles, resulting in $O(nd)$ total complexity. Mixing and resampling the resulting particles requires an additional $O(n + d)$ time.

In practice, we have found that the large increase in performance provided by the MPF dramatically outweighs the small increase in computational complexity. This occurs because, as shown in Table 1, the four operations described above take dramatically different amounts of time. Sampling from the transition model dominates the majority (79.35%) of the runtime because each sample involves running a computationally-expensive physics simulation. Evaluating the observation model consumes much (17.82%) of the remaining runtime because of the large number of collision checks required to simulate an observation. All remaining operations—including the overhead incurred by the MPF when sampling from the dual proposal distribution—is negligible (1.38%) compared to these two operations.

Note that the time required to evaluate the transition and observation models depends largely on the composition of $b(s_t)$. When $b(s_t \in S_{\mathrm{free}})$ is high, then few particles are likely to be touching the hand and the transition and observations models can be sped up through intelligent use of conservative, broad-phase checks (e.g. bounding box intersection queries). This suggests that particles on S_{c} are "more expensive" than those in the ambient space S_{free}. It may be possible to leverage this insight by modifying the resampling step

			No Contact		Contact	
	Operation	Complexity	Time (ms)	Percent	Time (ms)	Percent
(a)	Conventional Proposal	$O(n)$	20.36	65.85%	130.37	79.35%
(b)	Conventional Weights	$O(n)$	8.77	28.36%	29.28	17.82%
(c)	Dual Proposal	$O(n+d)$	–	–	0.02	0.01%
(d)	Dual Weights	$O(nd)$	–	–	2.25	1.37%
(e)	Other	–	1.80	5.79%	2.37	1.45%
	Total		30.92	100%	164.29	100%

Table 1: Time required to perform the sampling stages of the (a)–(b) CPF and (a)–(d) MPF. Nearly all time is spent evaluating the (a)–(b) transition and observation models necessary to implement the conventional proposal distribution. In comparison, the time required to sample from the (c)–(d) dual proposal distribution is negligible. Timing information was collected on one core of a 3.4 GHz Intel i7 processor.

to keep more particles in S_{free}. Those particles would be down-weighted to avoid biasing the distribution towards no-contact.

In addition to the runtime performance described above, the MPF incurs a one-time cost to build a representation of the contact manifold. Building the contact manifold is only required once per hand-object pair and is done as an offline pre-computation step that does not affect the speed of the algorithm at runtime. For the parameters described in Section 7, constructing the contact manifold took 51.96 s for the analytic representation, 451.45 s for the rejection-sampled representation, and 429.49 s for the trajectory rollout representation. In all three cases, the resolution parameters were intentionally tuned to maximize the estimator's runtime performance with no regard for pre-computation time. Reducing the resolution of the manifold could dramatically reduce pre-computation time, while having a relatively small effect on the MPF's accuracy.

9.3 Mixing Rate Parameter

Our experimental results (Section 7) show that, surprisingly, the the optimal choice of mixing rate ϕ is much closer to $\phi = 0$ than $\phi = 1$. This may seem counter-intuitive: the MPF outperforms the CPF, so one would expect increasing ϕ to improve performance. Two competing forces partially mitigate this effect: (1) the susceptibility of the dual proposal distribution to observation noise and (2) the tendency for the dual proposal distribution to increase the variance of the posterior distribution.

First, the dual proposal distribution performs poorly when confronted with sensing errors (Thrun, Fox & Burgard 2000). Receiving an erroneous observation can cause the majority of the particles sampled from the dual proposal distribution to lie in the wrong region of the state space. This stems from the same underlying problem described in Section 4.4: when an observation error occurs, the proposal distribution is a poor approximation of the target distribution. As a result, it would take a prohibitively large number of samples to to faithfully represent the posterior. Using the mixture proposal distribution (Section 6.5) leverages the complementary nature of the conventional and dual proposal distributions to avoid the worst-case behavior of either distribution (Thrun, Fox & Burgard 2000).

Second, the belief state tracked by the pure MPF tends to increase in variance over time. This occurs because we use kernel density estimation to compute the importance weights for particles sampled from the dual proposal distribution. Kernel density estimation, as described in Section 6.4, replaces the Dirac delta function in the filtering distribution with a kernel that has broad support. As a result, particles sampled from the dual proposal distribution generally have higher variance than those sampled from the true posterior distribution. This variance increases over time as the estimator is recursively updated. Assigning a low weight to the particles sampled from the MPF—by choosing a low value of ϕ—reduces the rate at which the variance grows.

These results suggest that the mixing rate should vary between update steps. The mixture rate should be high when sampling from conventional proposal distribution performs poorly; e.g. when transitioning

from no-contact to contact or moving between contact sensors. Otherwise, ϕ should kept near $\phi = 0$ to avoid introducing variance into the posterior. We are interested in exploring this idea in future work. For example, we could vary ϕ as a function of the number of effective particles (Liu 2008) in the CPF posterior.

9.4 Measure-Theoretic Considerations

Regardless of whether the observation model is discriminative, our derivation of the MPF in Section 5 relied on our ability to factor the belief state $b(s_t) = \sum_{j=1}^{m} b(s_t|M_j)b(s_t \in M_j)$ into a m separate conditional probability distributions and a marginal distribution $b(s_t \in M_j)$ over the manifolds. However, this definition introduces an apparent inconsistency: How can $b(s_t \in M_j)$ possibly be non-zero when M_j is a lower-dimensional manifold?

We can answer this question using measure theory. Formally, our probability space (S, F, μ) consists of the sample space S, the σ-algebra of events $F \subseteq 2^S$, and a probability measure $\mu : F \to \mathbb{R}$. Most applications assume that μ is isomorphic to the Lebesgue measure $\lambda : F \to \mathbb{R}$ over the unit interval (Rokhlin 1962). Any measure that satisfies this property would assign $\mu(M_j) = 0$ for the lower-dimensional manifolds $i < m$. Unfortunately, this not the case for the MPF because non-zero probability is concentrated on the lower-dimensional manifolds $M_1, \ldots, M_{m-1} \subset S$. Since the n-dimensional Lebesgue measure assigns zero measure to any set with dimension less than n, there does not exist a measurable map between μ and λ. For example, it is possible that $\mu(M_j) > 0$ and $\lambda(M_j) = 0$ for $j < m$.

However, we can express the probability measure $\mu = \sum_{j=1}^{m} \mu_j$ as the sum of the m measures $\mu_1, \ldots, \mu_m$ where each $\mu_j : 2^{M_j} \cap F \to \mathbb{R}$ is a measure over M_j. If the measures $\mu_1, \ldots, \mu_m$ are *partial probability measures*[11] and satisfy $\sum_{j=1}^{m} \mu_i(M_j) = 1$, then μ is a probability measure over S. Any probability distribution μ' over $M_1, \ldots, M_m$ can be defined in terms of a *probability density function* $p(s)$ with respect to μ. In this case, the probability density function is the Radon–Nikodym derivative of μ' and $\int_A p(s)d\mu(s)$ is the Lebesgue integral of $p(s)$ over $A \subseteq S$, both taken with respect to measure μ (Resnick 1999). The derivation of the MPF in Section 5 implicitly assumes that all densities and integrals are defined in this way.

We intuitively arrived at the same understanding by factoring the belief state as $b(s_t) = \sum_{j=1}^{m} b(s_t|M_j)$. The marginal $b(s_t \in M_j) = \int_{M_j} b(s_t)d\mu(s_t)$ is the total probability contributed by the partial probability measure μ_i. This value represents the probability of s_t residing on M_j. Each conditional distribution $b(s_t|M_j)$ is simply the measure of the corresponding partial probability measure normalized such that it sums to one.[12]

9.5 Limitations and Future Work

We made several simplifying assumptions when applying the MPF to contact manipulation. We focused on the problem of planar manipulation in a quasistatic environment (Section 9.5.1), assumed the shape of the hand is fixed (Section 9.5.2), and only consider discriminative contact sensors (Section 9.5.3). We are interested in relaxing all three of these assumptions in future work.

We are also interested in extending the MPF to estimate uncertain physical properties of the environment (Section 9.5.4) during execution. Finally—and most importantly—we are excited by the prospect of using the MPF's state estimate for real-time feedback (Section 9.5.5).

9.5.1 Manipulation in Higher Dimensions

The MPF, as described in this paper, assumes that the robot is performing planar manipulation in a quasistatic (Lynch et al. 1992) environment. Applying the MPF to a three-dimensional environment would require filtering in $S = SE(3)$. Relaxing the quasistatic assumption again doubles the dimensionality of the problem by expanding the state space to the full tangent bundle $S = SE(3) \times se(3)$.

Increasing the dimensionality of the state space causes an exponential increase in the number of samples required by the the rejection-sampled representation to approximate S_o. Similarly, building an analytic

[11] A measure μ_j is a partial probability measure if $\mu_j(\emptyset) = 0$, $\mu_j(M_j) \leq 1$, and μ_j is σ-additive.

[12] Technically, the conditional belief $b(s_t|M_j)$ is undefined if $b(s_t \in M_j) = 0$. This is why we defined μ as the sum of partial probability measures instead of as the convex combination of full probability measures. However, this is not a practical concern because the MPF will never sample from $b(s_t|M_j)$ if $b(s_t \in M_j) = 0$.

representation of S_o is not computationally tractable because it would entail computing polygonal Minkowski sums over a large number of discretized orientation parameters (Varadhan & Manocha 2006).

It may, however, be possible to extend the trajectory rollout representation (Section 6.3.2) to higher dimensions. This representation, unlike the RS and AM representations, scales only with the size of the region of the state space that we encounter with non-trivial probability during execution. Furthermore, it may be possible to avoid sampling S_o during a pre-computation step and, instead, dynamically generate samples from S_o online. Generating these samples is potentially much less expensive than running the MPF with more particles. Unlike particles, which must always be distributed according to $b(s_t)$, these samples only are required to uniformly cover S_o. This may allow us to bypass additional evaluations of the expensive transition model (Section 9.2) in favor of a more efficient sampling technique.

9.5.2 Manipulation with an Articulated Hand

Sampling from the dual proposal distribution requires using one of the contact manifold representations described in Section 6.3. All three of these implementations assume that the hand is a rigid body. This means that applying the MPF to an articulated hand requires pre-computing a separate contact manifold for all possible hand configurations that may be encountered during execution. The timing information shown in Section 9.2 shows that this is only tractable for a small, discrete set of hand shapes.

We are interested in applying the MPF to articulated hands by building a representation of the contact manifold that can efficiently adapted to multiple hand shapes. This may be possible by pre-computing multiple contact manifolds—one for each rigid component of the hand—and combining them at runtime. The key challenge with this approach is to efficiently generate the lower-dimensional events that occur when an object settles into multi-point contact with the hand. Solving this problem would allow the MPF to be applied during the full grasping process, instead of only to planar pushing.

9.5.3 Non-Discriminative Contact Sensors

One major limitation of our current implementation of the MPF is that it relies on having a discriminative observation model. This is a valid assumption in many cases, but fails when manipulating light objects or using unreliable sensors. When this is the case, we can no longer approximate the marginal distribution $b(s_t \in S_o)$ using the latest observation and must find some other method of approximating the marginal.

In theory, we could approximate the marginal using the set of particles $S_t = \{\langle s_t^{[i]}, w_t^{[i]} \rangle\}_{i=1}^n$ sampled from the conventional proposal distribution as

$$b(s_t \in S_o) \approx \frac{\sum_{i=1}^n w_t^{[i]} I(s_t^{[i]} \in S_o)}{\sum_{i=1}^n w_t^{[i]}}$$

where $I(\cdot)$ is the indicator function. In practice, however, our results with the CPF shows that S_t is a poor approximation for $b(s_t)$: we simply do not have enough samples to accurately estimate this probability. However, we may be able to crudely estimate $b(s_t \in S_o)$ with sufficient regularization (Liu & West 2001).

Alternatively, it may be possible to approximate $b(s_t \in S_o)$ directly in terms of $b(s_{t-1})$ by backwards-propagating $\tilde{s}_{t-1}^{[i]} \sim p(s_{t-1}|s_t^{[i]}, a_t)$ each sample $s_t^{[i]}$ to the the previous timestep. The corresponding importance weights are proportional to $b(\tilde{s}_{t-1}^{[i]})$, which can be approximated using the density estimation techniques described in Section 6.4. Thrun, Fox & Burgard (2000) show that this technique performs well on mobile robot localization. Unfortunately, this algorithm may be difficult to implement because it is challenging to sample from the inverse transition model $p(s_{t-1}|s_t, a_t)$ in the case of contact manipulation.

9.5.4 Parameter Estimation

Much of the noise in the transition model $p(s_t|s_{t-1}, a_t)$ may actually result from unknown—but static—properties $\gamma \in \Gamma$ of system. This is equivalent to writing the transition model as the marginal $p(s_t|s_{t-1}, a_t) = \int_\Gamma p(s_t|s_{t-1}, a_t, \gamma)p(\gamma)d\gamma$ of an underlying transition model $p(s_t|s_{t-1}, a_t, \gamma)$ over the unknown parameters γ. In the case of the quasistatic motion model, γ is the finger-object coefficient of friction and the radius of the object's pressure distribution (Lynch et al. 1992, Dogar & Srinivasa 2010). Prior work has shown that

it is possible to estimate static (Zhang & Trinkle 2012) and spatio-temporally varying (Zhang et al. 2013) friction coefficients using visual and tactile data. Adding this capability to the MPF would require a method of assigning parameters to particles sampled from the dual proposal distribution.

Another important case is when γ includes the hand-object geometry and the goal is shape estimation (Allen 1984, Bjorkman et al. 2013, Ilonen et al. 2014) or object identification (Xu et al. 2013, Schneider et al. 2009, Chitta et al. 2011). This case is particularly important when using compliant end-effectors (e.g. i-HY hand (Odhner et al. 2013)) or manipulating unmodeled 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 create a "fuzzy" contact manifold that consists of the union of several hypothesized contact manifolds.

The MPF, as presented in this paper, assumes that the hand-object geometry is known with certainty and that $p(\gamma)$ is a known, stationary distribution. Instead of estimating $b(s_t)$, we could estimate the joint belief $b(s_t, \gamma)$ over the state $s_t \in S$ and the parameter $\gamma \in \Gamma$ values. In principle, this could be accomplished by filtering in the augmented state space (S, Γ) with a trivial transition model for γ. In future work, we are interested in extending the dual sampling step in the MPF to support these types of parameters. This may be challenging because it is difficult to implement the Bayes update with continuous γ without suffering from particle starvation (Liu & West 2001).

9.5.5 Real-Time Feedback

Finally, we would like to use the belief state estimated by the MPF for real-time feedback. This problem can be naturally formulated as a *partially observable Markov decision process* (POMDP) (Kaelbling et al. 1998) with the transition and observation models defined in Section 3 and a reward function that assigns positive reward to achieving the problem-specific goal.

Optimally solving a general POMDP is PSPACE-complete (Littman 1996). However, we are optimistic that the structure of the contact manipulation problem will enable us to efficiently find approximate solutions. Contact sensors provide little information before contact and, as a result, the problem is nearly deterministic. Once contact occurs, the discriminative nature of contact sensors means that the belief state exhibits sparse support (Lee et al. 2007) that is constrained to the contact manifold. It may be possible to leverage this knowledge in a special-purpose POMDP solver. We are encouraged by recent work—including our own (Koval et al. 2014)—that has achieved promising results in grasping (Hsiao 2009, Platt et al. 2011) and non-prehensile manipulation (Horowitz & Burdick 2013) using a POMDP formulation of the problem.

10 Conclusion

In this paper, we investigated the problem of using contact sensors to estimate the pose of an object during planar manipulation (Section 3). We showed that the conventional particle filter (Section 4) performs poorly on this problem because the state lies on the lower-dimensional contact manifold during periods of contact.

We introduced the manifold particle filter (Section 5) as a solution to this problem and showed how it can be applied to the contact manipulation problem (Section 6) using three different representations of the contact manifold. Our simulation results (Section 7) show that the CPF significantly outperforms the MPF and that the gap widens further as sensor resolution and update step size decreases. Finally, we implemented the MPF on a real robot (Section 8) and showed that the MPF is able to successfully track an object using commercially available tactile sensors.

Acknowledgments

This material is based on work supported by NSF award IIS-1218182, the DARPA Autonomous Robotic Manipulation Software Track (award DARPA-BAA-10-28), a NASA Space Technology Research Fellowship (award NNX13AL62H), and the Toyota Motor Corporation. We would like to thank Drew Bagnell, Mehmet Dogar, Geoff Gordon, Jen King, and the members of the Personal Robotics Lab for their helpful input.

Finally, we would like to thank Michael Dawson-Haggerty for his invaluable help with conducting the real-robot experiments.

References

Akella, S. & Mason, M. (1998), 'Posing polygonal objects in the plane by pushing', *International Journal of Robotics Research* **3**(1), 2255–2262.

Allen, P. (1984), Surface descriptions from vision and touch, *in* 'IEEE International Conference on Robotics and Automation'.

Bagnell, J., Cavalcanti, F., Cui, L., Galluzzo, T., Hebert, M., Kazemi, M., Klingensmith, M., Libby, J., Liu, T., Pollard, N., M. Pivtoraiko, J.-S. V. & Zhu, R. (2012), An integrated system for autonomous robotics manipulation, *in* 'IEEE International Conference on Robotics and Automation'.

Bentley, J. (1975), 'Multidimensional binary search trees used for associative searching', *Communications of the ACM* **18**(9), 509–517.

Bjorkman, M., Bekiroglu, Y., Hogman, V. & Kragic, D. (2013), Enhancing visual perception of shape through tactile glances, *in* 'IEEE/RSJ International Conference on Intelligent Robots and Systems'.

Blackwell, D. (1947), 'Conditional expectation and unbiased sequential estimation', *The Annals of Mathematical Statistics* .

Brost, R. (1988), 'Automatic grasp planning in the presence of uncertainty', *International Journal of Robotics Research* **7**(1), 3–17.

Chang, L., Srinivasa, S. & Pollard, N. (2010), Planning pre-grasp manipulation for transport tasks, *in* 'IEEE International Conference on Robotics and Automation'.

Chitta, S., Sturm, J., Piccoli, M. & Burgard, W. (2011), 'Tactile sensing for mobile manipulation', *IEEE Transactions on Robotics* **27**(3), 558–568.

Corcoran, C. & Platt, R. (2010), Tracking object pose and shape during robot manipulation based on tactile information, *in* 'IEEE International Conference on Robotics and Automation'.

Cosgun, A., Hermans, T., Emeli, V. & Stilman, M. (2011), Push planning for object placement on cluttered table surfaces, *in* 'IEEE/RSJ International Conference on Intelligent Robots and Systems'.

Dogar, M., Hsiao, K., Ciocarlie, M. & Srinivasa, S. (2012), Physics-based grasp planning through clutter, *in* 'Robotics: Science and Systems'.

Dogar, M. & Srinivasa, S. (2010), Push-grasping with dexterous hands: Mechanics and a method, *in* 'IEEE/RSJ International Conference on Intelligent Robots and Systems'.

Dogar, M. & Srinivasa, S. (2011), A framework for push-grasping in clutter, *in* 'Robotics: Science and Systems'.

Dogar, M. & Srinivasa, S. (2012), 'A planning framework for non-prehensile manipulation under clutter and uncertainty', *Autonomous Robots* **33**(3), 217–236.

Drummond, T. & Cipolla, R. (2002), 'Real-time visual tracking of complex structures', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**(7), 932–946.

Duff, D. (2011), Visual motion estimation and tracking of rigid bodies by physical simulation, PhD thesis, University of Birmingham.

Duff, D., Stolkin, T. M. R. & Wyatt, J. (2011), Physical simulation for monocular 3D model based tracking, *in* 'IEEE International Conference on Robotics and Automation'.

Duff, D., Wyatt, J. & Stolkin, R. (2010), Motion estimation using physical simulation, *in* 'IEEE International Conference on Robotics and Automation'.

Edin, B., Beccai, L., Ascari, L., Roccella, S., Cabibihan, J. & Carrozza, M. (2006), Bio-inspired approach for the design and characterization of a tactile sensory system for a cybernetic prosthetic hand, *in* 'IEEE International Conference on Robotics and Automation'.

Epanechnikov, V. (1969), 'Non-parametric estimation of a multivariate probability density', *Theory of Probability & Its Applications* **14**(1), 153–158.

Farahat, A., Stiller, P. & Trinkle, J. (1995), 'On the geometry of contact formation cells for systems of polygons', *IEEE Transactions on Robotics* .

Ferris, M. & Munson, T. (1999), 'Interfaces to PATH 3.0: Design, implementation and usage', *Computational Optimization and Applications* **12**(1-3), 207–227.

Fischler, M. & Bolles, R. (1981), 'Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography', *Communications of the ACM* **26**(6), 381–395.

Fishel, J. & Loeb, G. (2012), 'Sensing tactile microvibrations with the BioTac – comparison with human sensitivity', *IEEE/RAS-EMBS BioRob* .

Gadeyne, K., Lefebvre, T. & Bruyninckx, H. (2005), 'Bayesian hybrid model-state estimation applied to simultaneous contact formation recognition and geometrical parameter estimation', *International Journal of Robotics Research* **24**(8), 615–630.

Gordon, N., D.J., S. & Smith, A. (1993), Novel approach to nonlinear/non-Gaussian Bayesian state estimation, *in* 'IEE Proceedings F'.

Harris, C. (1992), Geometry from visual motion, *in* A. Blake & A. Yuille, eds, 'Active Vision', MIT Press, chapter 16.

Hebert, P., Howard, T., Hudson, N., Ma, J. & Burdick, J. (2013), The next best touch for model-based localization, *in* 'IEEE International Conference on Robotics and Automation'.

Horowitz, M. & Burdick, J. (2013), Interactive non-prehensile manipulation for grasping via POMDPs, *in* 'IEEE International Conference on Robotics and Automation'.

Howe, R. & Cutkosky, M. (1996), 'Practical force-motion models for sliding manipulation', *International Journal of Robotics Research* **15**(6), 557–572.

Hsiao, K. (2009), Relatively robust grasping, PhD thesis, Massachusetts Institute of Technology.

Ilonen, J., Bohg, J. & Kyrki, V. (2014), 'Three-dimensional object reconstruction of symmetric objects by fusing visual and tactile sensing', *International Journal of Robotics Research* **33**(2), 321–341.

Javdani, S., Klingensmith, M., Bagnell, J., Pollard, N. & Srinivasa, S. (2013), Efficient touch based localization through submodularity, *in* 'IEEE International Conference on Robotics and Automation'.

Jia, Y. & Erdmann, M. (1999), 'Pose and motion from contact', *International Journal of Robotics Research* **18**(5), 466–490.

Julier, S. & Uhlmann, J. (1997), A new extension of the Kalman filter to nonlinear systems, *in* 'International Symposium on Aerospace/Defense Sensing, Simulation, and Controls'.

Kaelbling, L., Littman, M. & Cassandra, A. (1998), 'Planning and acting in partially observable stochastic domains', *AI* .

Kalman, R. (1960), 'A new approach to linear filtering and prediction problems', *Journal of Basic Engineering* .

Kappler, D., Chang, L., Przybylski, M., Pollard, N., Asfour, T. & Dillmann, R. (2010), Representation of pre-grasp strategies for object manipulation, *in* 'IEEE-RAS International Conference on Humanoid Robots'.

King, J., Klingensmith, M., Dellin, C., Dogar, M., Velagapudi, P., Pollard, N. & Srinivasa, S. (2013), Pregrasp manipulation as trajectory optimization, *in* 'Robotics: Science and Systems'.

Koval, M., Dogar, M., Pollard, N. & Srinivasa, S. (2013), Pose estimation for contact manipulation with manifold particle filters, *in* 'IEEE/RSJ International Conference on Intelligent Robots and Systems'.

Koval, M., Pollard, N. & Srinivasa, S. (2013), Manifold representations for state estimation in contact manipulation, *in* 'International Symposium of Robotics Research'.

Koval, M., Pollard, N. & Srinivasa, S. (2014), Pre- and post-contact policy decomposition for planar contact manipulation under uncertainty, *in* 'Robotics: Science and Systems'.

Lau, M., Mitani, J. & Igarashi, T. (2011), Automatic learning of pushing strategy for delivery of irregularhaped objects, *in* 'IEEE International Conference on Robotics and Automation'.

LaValle, V. (2006), *Planning algorithms*, Cambridge University Press.

Lee, W., Rong, N. & Hsu, D. (2007), What makes some POMDP problems easy to approximate?, *in* 'Advances in Neural Information Processing Systems'.

Li, Q., Schürmann, C., Haschke, R. & Ritter, H. (2013), A control framework for tactile servoing, *in* 'Robotics: Science and Systems'.

Littman, M. (1996), Algorithms for Sequential Decision Making, PhD thesis, Brown University.

Liu, J. (2008), *Monte Carlo Strategies in Scientific Computing*, Springer.

Liu, J. & West, M. (2001), Combined parameter and state estimation in simulation-based filtering, *in* 'Sequential Monte Carlo methods in practice', Springer.

Lozano-Pèrez, T. (1983), 'Spatial planning: A configuration space approach', *IEEE Transactions on Computers* .

Lynch, K., Maekawa, H. & Tanie, K. (1992), Manipulation and active sensing by pushing using tactile feedback, *in* 'IEEE/RSJ International Conference on Intelligent Robots and Systems'.

Lynch, K. & Mason, M. (1995), 'Pulling by pushing, slip with infinite friction, and perfectly rough surfaces', **14**(2), 174–183.

Lynch, K. & Mason, M. (1996), 'Stable pushing: Mechanics, controllability, and planning', *International Journal of Robotics Research* **15**(6), 533–556.

Mason, M. (1986), 'Mechanics and planning of manipulator pushing operations', *International Journal of Robotics Research* **5**(3), 53–71.

Meeussen, W., Rutgeerts, J., Gadeyne, K., Bruyninckx, H. & De Schutter, J. (2007), 'Contact-state segmentation using particle filters for programming by human demonstration in compliant-motion tasks', *IEEE Transactions on Robotics* **23**(2), 218–231.

Montemerlo, M., Thrun, S., Koller, D. & Wegbreit, B. (2003), FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges, *in* 'International Joint Conference on Artificial Intelligence'.

Mörwald, T., Zillich, M. & Vincze, M. (2009), Edge tracking of textured objects with a recursive particle filter, *in* 'Graphicon'.

NVIDIA Corporation (2014), 'PhysX SDK 2.8', http://www.geforce.com/hardware/technology/physx. Accessed: 2014-11-18.

Odhner, L., Jentoft, L., Claffee, M., N.Corson, Tenzer, Y., Ma, R., Buehler, M., Kohout, R., Howe, R. & Dollar, A. (2013), 'A compliant, underactuated hand for robust manipulation', *CoRR* .

Pastor, P., Righetti, L., Kalakrishnan, M. & Schaal, S. (2011), Online movement adaptation based on previous sensor experiences, *in* 'IEEE/RSJ International Conference on Intelligent Robots and Systems'.

Pelletier, B. (2005), 'Kernel density estimation on Riemannian manifolds', *Statistics & Probability Letters* **73**(3), 297–304.

Petrovskaya, A. & Khatib, O. (2011), 'Global localization of objects via touch', *IEEE Transactions on Robotics* **27**(3), 569–585.

Platt, R., Fagg, A. & Grupen, R. (2010), 'Nullspace grasp control: theory and experiments', *IEEE Transactions on Robotics* **26**(2), 282–295.

Platt, R., Kaelbling, L., Lozano-Pèrez, T. & Tedrake, R. (2011), Simultaneous localization and grasping as a belief space control problem, *in* 'International Symposium of Robotics Research'.

Resnick, S. (1999), *A Probability Path*, Birkhäuser.

Rokhlin, V. (1962), *On the fundamental ideas of measure theory*, American Mathematical Society.

Rosenblatt, M. (1956), 'Remarks on some nonparametric estimates of a density function', *The Annals of Mathematical Statistics* **27**(3), 832–837.

Salisbury, K., Townsend, W., Eberman, B. & DiPietro, D. (1988), Preliminary design of a whole-arm manipulation system (WAMS), *in* 'IEEE International Conference on Robotics and Automation'.

Schneider, A., Sturm, J., Stachniss, C., Reisert, M., Burkhardt, H. & Burgard, W. (2009), Object identification with tactile sensors using bag-of-features, *in* 'IEEE/RSJ International Conference on Intelligent Robots and Systems'.

Schutter, J. D. & Brussel, H. V. (1988), 'Compliant motion I, II', *International Journal of Robotics Research* .

Silverman, B. (1981), 'Using kernel density estimates to investigate multimodality', *Journal of the Royal Statistical Society. Series B (Methodological)* pp. 97–99.

Smith, A. & Gelfand, A. (1992), 'Bayesian statistics without tears: a sampling–resampling perspective', *The American Statistician* .

Srinivasa, S., Berenson, D., Cakmak, M., Collet, A., Dogar, M., Dragan, A., Knepper, R., Niemueller, T., Strabala, K. & Vande Weghe, M. (2012), 'HERB 2.0: Lessons learned from developing a mobile manipulator for the home', *Proceedings of the IEEE* **100**(8), 1–19.

Tenzer, Y., Jentoft, L. & Howe, R. (2014), Inexpensive and easily customized tactile array sensors using MEMS barometers chips, *in* 'IEEE Robotics and Automation Magazine'.

Thrun, S., Burgard, W. & Fox, D. (2005), *Probabilistic robotics*, MIT Press.

Thrun, S., Fox, D. & Burgard, W. (2000), Monte Carlo localization with mixture proposal distribution, *in* 'National Conference on Artificial Intelligence'.

Thrun, S., Fox, D., Burgard, W. & Dellaert, F. (2000), Robust Monte Carlo localization for mobile robots, Technical Report CMU-CS-00-125, Computer Science Department,Carnegie Mellon University, Pittsburgh, PA.

Varadhan, G. & Manocha, D. (2006), 'Accurate Minkowski sum approximation of polyhedral models', *Graphical Models* **68**(4), 343–355.

Wein, R. (2013), 2D Minkowski sums, *in* 'CGAL User and Reference Manual', 4.3 edn, CGAL Editorial Board.

Xiao, J. (1993), Automatic determination of topological contacts in the presence of sensing uncertainties, *in* 'IEEE International Conference on Robotics and Automation'.

Xu, D., Loeb, G. & Fishel, J. (2013), Tactile identification of objects using Bayesian exploration, *in* 'IEEE International Conference on Robotics and Automation'.

Zhang, H. & Chen, N. (2000), 'Control of contact via tactile sensing', *Robotics and Automation, IEEE Transactions on* **16**(5), 482–495.

Zhang, L. (2013), Physics-empowered perception for robot grasping and dexterous manipulation, PhD thesis, Rensselaer Polytechnic Institute.

Zhang, L., Lyu, S. & Trinkle, J. (2013), A dynamic bayesian approach to simultaneous estimation and filtering in grasp acquisition, *in* 'IEEE International Conference on Robotics and Automation'.

Zhang, L. & Trinkle, J. (2012), The application of particle filtering to grasping acquisition with visual occlusion and tactile sensing, *in* 'IEEE International Conference on Robotics and Automation'.

Zito, C., Stolkin, R., Kopicki, M. & Wyatt, J. (2012), Two-level RRT planning for robotic push manipulation, *in* 'IEEE/RSJ International Conference on Intelligent Robots and Systems'.