Bayesian Tactile Exploration for Compliant Docking with Uncertain Shapes

Kris Hauser
Dept. of Electrical and Computer Engineering, Duke University
Durham, North Carolina 27708
Email: kris.hauser@duke.edu

***Abstract*—This paper presents a Bayesian approach for active tactile exploration of a planar shape in the presence of both localization and shape uncertainty. The goal is to dock the robot's end-effector against the shape – reaching a point of contact that resists a desired load – with as few probing actions as possible. The proposed method repeatedly performs inference, planning, and execution steps. Given a prior probability distribution over object shape and sensor readings from previously executed motions, the posterior distribution is inferred using a novel and efficient Hamiltonian Monte Carlo method. The optimal docking site is chosen to maximize docking probability, using a closed-form probabilistic simulation that accepts rigid and compliant motion models under Coulomb friction. Numerical experiments demonstrate that this method requires fewer exploration actions to dock than heuristics and information-gain strategies.**

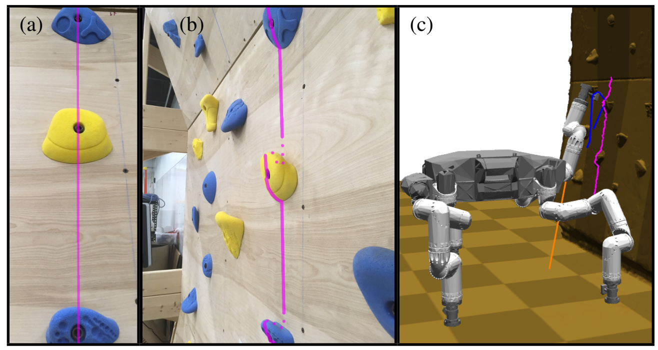

Fig. 1: (a) A climber's view of potential holds from underneath does not reveal the size and shape of pockets. (b) Shown from a side view, occluded regions from the original view are indicated by dotted lines. (c) Tactile exploration can be used by a climbing robot to probe the hidden terrain shape.

I. Introduction

Uncertainty is an inherent challenge in robot manipulation and locomotion; object/terrain geometries are sensed using imperfect sensors, geometric models are usually incomplete due to occlusion, material and friction properties cannot be directly observed, and robots suffer from calibration and localization error. Although humans are adept at using tactile information to infer the shape of objects and adapting their manipulation or locomotion strategies accordingly, robots remain quite far from mastering such behaviors.

In the sport of rock climbing, human climbers look upward to observe a terrain and ask the question: "Would that terrain feature make a good hand hold?" A good hold contains a pocket, ledge, or protrusion with a size and shape suitable for latching onto with fingers or tools, and applying large downward and/or backward forces. However, the parts of the terrain needed to assess quality are *precisely* the parts hidden from view. From below, a deep pocket can appear nearly identical to a useless slope (Fig. 1), so humans use the sense of touch to explore the shape of occluded geometry. If the terrain turns out to be unfavorable, the climber may move on to alternate holds or choose different routes.

The goal of this paper is to enable a climbing robot, equipped with a force/torque sensor, to explore static terrains using tactile sensing in order to "dock" an end effector in a location that resists a given load. Toward this end, three novel technical contributions are presented in this work:

1) A Bayesian geometric shape estimation technique that integrates free-space line segments, contact information, and stick/slip information from tactile sensing together with probabilistic priors. The estimator is statistically consistent and robust to rare events.
2) An analytical probabilistic simulation technique that quickly approximates the docking probability under a compliant robot motion model and Coulomb friction.
3) A fast path planner that optimizes the estimated docking probability for the posterior geometry distribution under Gaussian shape uncertainty.

Experiments suggest that a novel Hamiltonian Monte Carlo (HMC) method for shape estimation outperforms other methods under restrictive shape constraints. Analytical probabilistic simulation is much faster than Monte Carlo simulation methods with comparable accuracy, and hence path planning can be performed very quickly. These contributions are integrated into the inference and planning steps of a tactile exploration controller, which is shown to lead to optimized tactile exploration plans that can dock the end effector (or determine a low docking probability) in only a few attempts. The method is also explored in a realistic physics simulation with the Robosimian quadruped docking a hook end effector in 3D terrain.

II. Related Work

Prior gecko, insect, and snake-like climbing robots [33, 23, 31, 32, 37, 25] use bioinspired controllers and end-effector mechanisms to achieve passive compliance and adhesion to terrain uncertainties. These robots largely rely on gaited locomotion, which does not admit much flexibility in foothold choice. Motion planning has been employed for climbing robots to choose footholds in non-gaited fashion while verifying the existence of equilibrium postures [1], but these algorithms assume precise actuation and terrain modeling.

In the context of legged locomotion on uneven ground, tactile feedback has been explored for state estimation [7] and terrain property estimation [11]. Tactile sensing has proven to be a useful modality in robot manipulation to estimate object properties, such as friction, pose, and shape in the presence of visual sensing error and missing data due to occlusion [15]. Prior work can be grouped into three categories: *passive estimation*, *uncertainty-aware grasping*, and *active exploration*.

Passive contact has been used for state estimation of legged robots by fusing inertial readings with either known terrains [2] or unknown terrain shape observed by sensors [7]. Tactile sensors have been used to classify terrain friction and local shapes of contact points [11]. Machine learning techniques have been used to characterize terrain from visual and tactile sensors, which has been used to predict robustness of footholds [12] or adapt the gaits of a hexapod to optimize movement speed [34]. In manipulation, tactile feedback has been used for localization [27] and shape classification [22] of familiar objects. It has also been used for estimating the location of distinctive features like buttons in textiles [29] and localizing flat objects using texture and high-resolution tactile sensors [21]. A more difficult problem is simultaneous localization and shape estimation, since the unknown shape model must account for collision between fingers and unknown geometry. Prior work in this area typically uses probabilistic point cloud models [15] and Gaussian processes [4, 20, 36] that add point contacts and sensed points as constraints.

Our novel shape inference technique makes use of free space movement and slip detection in addition to contact information. The geometric consistency constraints used in this paper are similar to those proposed by Grimson and Lozano-Perez [8]. However, here they are used in a probabilistic setting to infer distributions of terrain shape rather than binary consistency. Hence, our method is similar to the manifold particle filter method proposed for using contact information for object localization in pushing [19]. The Markov Chain Monte Carlo (MCMC) method proposed here does not permit object movement, but is statistically consistent, i.e., converges to the true probability as more samples are drawn.

Uncertainty-aware grasping incorporates uncertainty into grasp planning by optimizing probabilistic measures (e.g., success probability) for 2D grasps [3] and 3D grasps [14]. Each of these techniques uses sampling for success probability estimation, which is advantageous for parametric object models [10] and deformable object shape [3] because standard simulation techniques can be used for each sample to determine success. However, sampling can be slow, in particular in the absence of good heuristics to restrict number of grasp alternatives [14].

Active tactile exploration schemes can be purely information-gathering or goal-directed. Information gathering has been applied to object shape and friction acquisition using compliant sliding, using Gaussian process models of shape and surface friction [30]. Information gain has been used as a metric for choosing localization actions before manipulating an object [10] and for addressing the exploration vs exploitation tradeoff in goal-directed grasping [5].

Partially-observable Markov decision processes (POMDP) are a principled approach to optimize active goal-directed manipulation [13, 18], but require discrete state, action, and observation spaces. Recent work has developed an RRT-like motion planner for compliant robots that explores continuous state and action spaces, while representing uncertain beliefs using particles [28]. This can be computationally expensive. The current work introduces a fast Gaussian docking probability estimator that is related to the collision probability method of Patil et al [26]. Novel contributions include the simulation of compliant motion with friction, and an improved probability estimate using truncated bivariate Gaussians rather than univariate ones.

III. Summary of Method

A. Problem Setup

1) Probabilistic shape model: The shape $S \subset \mathbb{R}^2$ is represented by its boundary ∂S, which is approximated as a polygonal mesh with vertices $V = (v_1, \ldots, v_n)$ and edges $E \subset V \times V$. The vertices are also represented as a stacked $2n$-dimensional vector x. The true vertex positions are unknown, so X denotes the random variable corresponding to x. The topology of the shape (i.e., E) is assumed known, and edges are oriented in CCW direction around S. The prior joint distribution $P(X)$ includes shape and localization uncertainty.

The elements of X are highly correlated. For example, localization uncertainty makes it more likely to observe a constant shift in translation or rotation across all vertices, rather than a partial shifting of the shape. Also, nearby points on the shape tend to be more correlated than distant points.

$P(X)$ is assumed to be well-approximated by a Gaussian of the form $X \sim N(\mu_x, \Sigma_x)$. We assume $\Sigma_x = A^T A$ is the product of a $2n \times m$ *basis matrix* A so that $X = AZ + \mu_x$ is an affine transform of a zero mean, unit variance normal variable $Z \sim N(0, I_m)$. The basis matrix A provides a convenient form to encode independence assumptions between different sources of uncertainty.

2) Robot motion model: For simplicity, the robot is assumed to be a point and the shape is assumed static. It is possible to relax the point robot assumption to handle a translating polygon, since the C-space obstacle has a polygonal shape. The robot has known position relative to the reference frame. To handle localization uncertainty, this reference frame can be taken as the egocentric frame, while $P(X)$ captures the localization error. The robot moves along a 2D path using guarded moves [16], which trigger a stop when the force felt by the robot exceeds a threshold.

Our method can include a *compliant motion model* that allows compliance perpendicular to the direction of motion. The robot may then slide against the shape, and the contact force obeys Coulomb friction. The surface friction is estimated, but the method is tolerant to errors in friction estimate.

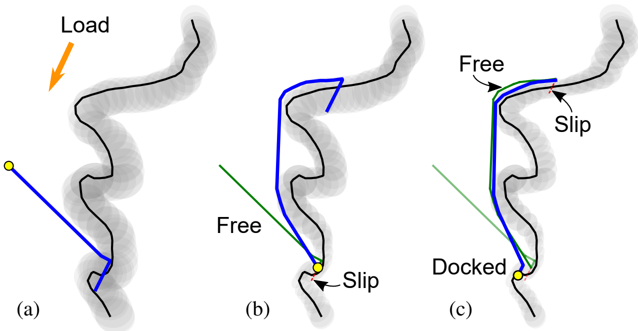

Fig. 2: (a) Bayesian tactile exploration to achieve a diagonally pulling load, with no compliance. The estimated shape distribution is shown in grey and the ground truth is drawn in black. Upon executing the initial plan to the lower ledge, the robot makes contact but slips, and the shape distribution is updated for consistency with the free-space and slip information. (b) An alternate site on the upper edge is chosen. (c) This again slips, and the third plan successfully docks against a notch near the first site.

3) Objective function and sensors: The docking objective is to stop at a point in contact with the shape $p \in \partial S$ such that a desired loading force f_{load} is entirely canceled by the friction forces available at p. In practice this is tested by having the final motion of the robot move in the direction f_{load} and checking if it sticks or jams. The robot attempts to minimize the amount of time before the loading force is acquired.

The information available to the robot is represented by line segments $\overline{ab} \subset \mathbb{R}^2$, annotated by their collision status s ("free" or "colliding"). Free segments help eliminate shape hypotheses in a manner similar to space carving for 3D shape estimation [9]. If the robot's force sensor provides enough information to estimate the stick/slip status of the various segments, we may also represent collision status flags "stick," "slip left," and "slip right." Here left and right indicate CCW and CW from the motion direction, respectively.

B. Bayesian Tactile Exploration Method

During exploration, the robot records the information from sensor readings $I_i = (s_i, a_i, b_i)$, $i = 1, \ldots, k$ as a history variable H, which is initially empty. It repeats several *exploration cycles*, each of which consists of the following steps:

1) Inference: infer the posterior distribution $P(X|H)$ of shape given history. (Sec. IV)
2) Optimization: Optimize the robot's path $p(t)$ to maximize a weighted sum of estimated docking probability $P(\text{dock}|p(t), H)$ and an exploration bonus. (Sec. V)
3) Execution: Execute $p(t)$. If the robot docks successfully, we are done. Otherwise, back up to a non-colliding point, record the sensor data in m new information segments $I_{k+1}, \ldots, I_{k+m}$ into the history: $H \leftarrow H \cup \{I_{k+1}, \ldots, I_{k+m}\}$, and repeat from step 1.

Execution also stops with failure if the docking probability drops below a threshold, which is set to 0.001 in our experiments. A successful three-cycle execution is shown in Fig. 2.

IV. Constrained Monte Carlo Shape Inference

To infer the shape distribution given history, we use Monte Carlo (MC) methods to draw a finite sample set from the true posterior distribution. The sample set will serve as an estimate of the distribution of shape (mean, covariance, and bounds) that improves in accuracy as more samples are drawn.

A. History consistency constraints

The posterior distribution of shapes conditioned on consistency with the sensor history $H = \{I_i = (a_i, b_i, s_i) | i = 1, \ldots, k\}$ can be expressed using Bayes' rule: $P(x|H) = \frac{P(H|x)P(x)}{P(H)}$. Let S_x denote the shape of S given that the vertex positions are given by state x. Under the assumption of perfect sensor information, $P(H|x) = 1$ if H is consistent with S_x, and $P(H|x) = 0$ otherwise. Hence, $P(x|H) \propto P(x)$ if S_x is consistent with H, and $P(x|H) = 0$ otherwise.

History consistency imposes the following conditions:

1) ∂S_x does not overlap any *free* segment $\overline{a_i b_i}$.
2) ∂S_x overlaps all *colliding* segments $\overline{a_i b_i}$.

We represent these conditions as mathematical inequalities. For each free segment, we require that:

$$f_{\text{free},a_i,b_i}(x) = \max_{(u,v)\in E} (-\max_k g_k(a_i, b_i, x_u, x_v)) \leq 0 \quad (1)$$

where x_u, x_v are the endpoints of an edge (u, v) specified by state x, and g_k, $k = 1, \ldots, 4$ are the segment-segment collision constraints as defined in Sec. IV-B (Fig. 3, left). For each colliding segment, we require that

$$f_{\text{coll},a_i,b_i}(x) = \min_{(u,v)\in E} (\max_k g_k(a_i, b_i, x_u, x_v)) \leq 0. \quad (2)$$

Note that there is a nested minimum and maximum in this expression because only one edge of the shape needs to collide. This condition can also be interpreted as a boolean disjunction.

Moreover, if stick/slip information is available for a colliding segment $\overline{a_i b_i}$, then the angle of the shape normal relative to the motion direction is constrained. Specifically:

1) $\overrightarrow{b_i a_i} \in Cone(n_{x,p} + \mu t_{x,p}, n_{x,p} - \mu t_{x,p})$ if $s_i = stick$.
2) $\overrightarrow{b_i a_i} \notin Cone(n_{x,p} + \mu t_{x,p}, n_{x,p} - \mu t_{x,p})$ if $s_i = slip$.

Here, $\overrightarrow{xy} \equiv y - x$, the first point of collision is denoted p, and the normal and tangent directions of ∂S_x at p are denoted $n_{x,p}$ and $t_{x,p}$, respectively. As we shall see in Sec. IV-C, these conditions add 2 additional constraints to (2), for a total of 6 constraints per edge (Fig. 3, right).

Overall, a shape x is history-consistent iff it satisfies

$$f_H(x) = \max_{(s,a,b)\in H} f_{s,a,b}(x) \leq 0. \quad (3)$$

B. Segment-segment collision constraints

Two planar segments $\overline{ab}$ and $\overline{cd}$ collide if and only if there exists a solution (u, v) to the system of equations

$$a + u \cdot \overrightarrow{ab} = c + v \cdot \overrightarrow{cd}, \quad \text{with } 0 \leq u, v \leq 1. \quad (4)$$

Solving for (u, v) via 2x2 matrix inversion we get

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{\alpha} \begin{bmatrix} c_2 - d_2 & d_1 - c_1 \\ a_2 - b_2 & b_1 - a_1 \end{bmatrix} \begin{bmatrix} c_1 - a_1 \\ c_2 - a_2 \end{bmatrix} \quad (5)$$

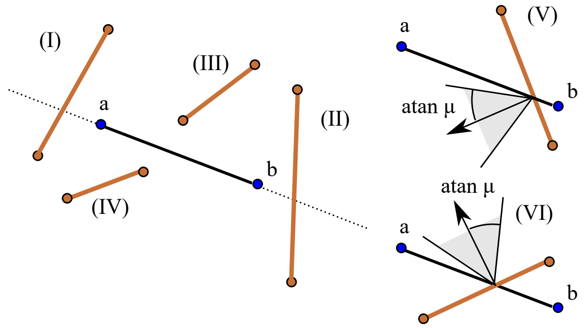

Fig. 3: Left: segment collision violations in cases (I)–(IV) correspond to violations of quadratic and linear inequalities (7)–(10), respectively. Right: stick constraint violations in cases (V) and (VI) correspond to violations of the respective elements of (12).

with $\alpha = (b_1-a_1)(c_2-d_2)-(c_1-d_1)(b_2-a_2)$ the determinant. Assuming $\alpha > 0$, that is, that $\overrightarrow{ab}$ is CCW from $\overrightarrow{cd}$, the original condition can then be rewritten as

$$0 \leq \begin{bmatrix} c_2-d_2 & d_1-c_1 \\ a_2-b_2 & b_1-a_1 \end{bmatrix} \begin{bmatrix} c_1-a_1 \\ c_2-a_2 \end{bmatrix} \leq \alpha \tag{6}$$

which is a quadratic inequality. Specifically, if we let $y = (c_1, c_2, d_1, d_2)$ denote the variables determining the coordinates of $\overline{cd}$, this can be rewritten as two quadratic inequalities and two linear inequalities

$$-y^T Q y + \begin{bmatrix} -a_2 & a_1 & a_2 & -a_1 \end{bmatrix} y \leq 0 \tag{7}$$
$$y^T Q y + \begin{bmatrix} b_2 & -b_1 & -b_2 & b_1 \end{bmatrix} y \leq 0 \tag{8}$$
$$\begin{bmatrix} b_2-a_2 & a_1-b_1 & 0 & 0 \end{bmatrix} y + (a_2 b_1 - a_1 b_2) \leq 0 \tag{9}$$
$$\begin{bmatrix} 0 & 0 & a_2-b_2 & b_1-a_1 \end{bmatrix} y + (a_1 b_2 - a_2 b_1) \leq 0. \tag{10}$$

with Q a constant 4x4 matrix. (It can also be shown that $\alpha \geq 0$ must hold if these equations are simultaneously satisfied.)

C. Segment stick/slip constraints

Let the operator $x^\perp$ on $\mathbb{R}^2$ yield the CCW perpendicular vector $(-x_2, x_1)$. For a motion along $\overline{ab}$, the *stick* condition requires

$$\overrightarrow{ba} \in Cone(\overrightarrow{dc}^\perp + \mu \cdot \overrightarrow{dc}, \quad \overrightarrow{dc}^\perp - \mu \cdot \overrightarrow{dc}), \tag{11}$$

where $Cone$ is the cone of positive combinations of its two arguments and μ is the friction coefficient. The constraint $x \in Cone(y_1, y_2)$ with $x \in \mathbb{R}^2$ is equivalent to two linear inequalities ${x^\perp}^T y_1 \geq 0$, ${x^\perp}^T y_2 \leq 0$ under the condition that $y_1^T y_2 \geq 0$ (i.e., y_2 is clockwise from y_1). This condition holds in (11), so (11) can be rewritten as inequalities

$$\begin{bmatrix} \mu \cdot \overrightarrow{b_2 a_2} - \overrightarrow{b_1 a_1} & -\overrightarrow{b_2 a_2} - \mu \cdot \overrightarrow{b_1 a_1} \\ \mu \cdot \overrightarrow{b_2 a_2} + \overrightarrow{b_1 a_1} & \overrightarrow{b_2 a_2} - \mu \cdot \overrightarrow{b_1 a_1} \end{bmatrix} \overrightarrow{cd} \leq 0 \tag{12}$$

that are linear over the vertex vector $y = (c_1, c_2, d_1, d_2)$. Moreover, the *slip left* condition is equivalent to $\overrightarrow{ba} \in Cone(\overrightarrow{dc}^\perp - \mu\overrightarrow{dc}, -\overrightarrow{dc})$, and *slip right* is equivalent to $\overrightarrow{ba}) \in Cone(\overrightarrow{dc}, \overrightarrow{dc}^\perp + \mu\overrightarrow{dc})$. A similar derivation produces two linear inequalities in y for either case.

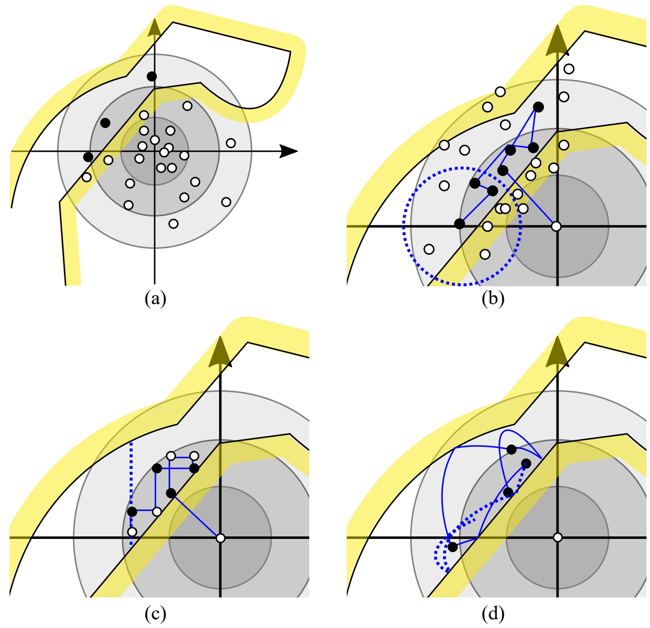

Fig. 4: Monte Carlo methods for constrained shape inference: (a) rejection sampling, (b) Metropolis-Hastings, (c) Gibbs sampling, and (d) Hamiltonian Monte Carlo. Black dots are accepted samples, white dots are rejected samples. The outlined shape illustrates the feasible set, blue paths illustrate a MCMC trajectory, and the dotted lines illustrate a sampling range.

D. Constrained Monte Carlo Sampling

Monte-Carlo methods are the preferred approach to sample from distributions $P(x|H)$ without having to compute $P(H)$. Without loss of generality, we shall sample a sequence $z^{(1)}, \ldots, z^{(N)}$ from the isotropic Gaussian distribution $z \sim N(0, I_m)$ under the restriction $f_H(Az + \mu_x) \leq 0$.

The simplest method for constrained MC is *rejection sampling* (Fig. 4.a), which leads to an i.i.d. sequence. However, procedure can be extremely inefficient, as $P(H)$ is often miniscule, and it will need to draw an expected $N/P(H)$ samples to find N feasible ones. MCMC methods can lower the rejection rate, but at the cost of introducing dependence between subsequent samples (autocorrelation). As a baseline MCMC technique, *Metropolis-Hastings* (MH) takes small perturbations and accepts moves with a given acceptance probability (Fig. 4.b). Our experiments suggests MH performs poorly in constrained sampling due to strong autocorrelation.

We also consider the *Gibbs sampling* technique (Fig. 4.c), which has been applied to Gaussian distributions truncated by linear inequalities [17]. Each iteration samples the posterior distribution of a single element of the state along a given axis, keeping all other elements fixed. Specifically, $z_i^{(j+1)} \leftarrow P(z_i | H, z_1^{(j)}, \ldots, z_{i-1}^{(j)}, z_{i+1}^{(j)}, \ldots, z_m^{(j)})$, with $i = j \mod m$ denoting the chosen element of the state vector. Customarily, every m'th sample is kept and the rest discarded. This approach leads to a constant rejection rate of $(m-1)/m$, which is independent of $P(H)$.

To sample z_i, we determine a feasible range by intersecting the feasible set along the line through $Az + \mu_x$ in direction Ae_i. Specifically, we determine the set of t such that $f_H(A(z+$

$e_i(t - z_i)) + \mu_x) \leq 0$. We discuss how to do so in Sec. IV-E. The range is a set of disjoint intervals, from which t can be sampled from using truncated Gaussian sampling routines that are widely available in scientific computing software libraries.

We finally present a constraint bouncing *Hamiltonian Monte Carlo* (HMC) method (Fig. 4.d), which has been applied to Gaussian distributions under linear and quadratic inequalities [24]. For each iteration, HMC treats the state as a dynamic particle subject to momentum and external force, which has a *momentum vector* p_0 that is sampled independently at random. Starting from $z_0 \equiv z^{(j)}$ and p_0, the method integrates the equations of motion of a dynamic particle subject to the system Hamiltonian $H(z, p)$, which is the sum of a potential energy $U(z) = \log P(z)$ and a kinetic energy $K(p)$ which is a positive definite function of p. The time evolution of the particle follow the coupled ODE:

$$\frac{d}{dt}z = \frac{\partial H}{\partial p}, \quad \frac{d}{dt}p = -\frac{\partial H}{\partial z}. \tag{13}$$

This dynamical system is reversible, and hence integration of these equations for a given timestep T to obtain a proposal state (z', p') can be viewed as a Metropolis-Hastings proposal distribution. In the Gaussian case the integration greatly simplifies [24]. With $\log P(z) = 1/2\|z\|^2$ and setting $K(p) = 1/2\|p\|^2$, the equations of motion trace out an ellipsoid given by the closed form $z(t) = z_0 \cos t + p_0 \sin t$. Moreover, the MH acceptance probability is always 1, so a step $z^{(j+1)} = z(t)$ can always be taken for any value of t. A recommended step size is $t = \pi/2$ because it tends to produce low autocorrelation [24].

A given step along the elliptic trajectory may violate feasibility, so a constraint bouncing method is used. This involves determining the first point in time t_b at which a constraint is violated, advancing the system to t_b, and then reflecting the momentum about the gradient of that constraint. The equations of motion are then integrated forward again until another constraint is hit, or the desired timestep is reached. Again, feasible range determination is used here (Sec. IV-E.)

Note that MCMC methods must begin from a feasible initial seed. We find the seed by random descent of f_H from an initial state sampled from $N(0, I_m)$. If this fails after a given number of iterations, we sample another initial state and repeat.

E. Feasible range determination

Both Gibbs and HMC sampling steps determine the feasible interval set along a state space trajectory $z(t)$. The Gibbs method samples t from all feasible intervals along a line, while HMC t from the feasible interval containing $t = 0$ along an ellipsoidal trajectory. An interval set is a collection of $r \geq 0$ disjoint intervals $[t_1, t_2] \cup [t_3, t_4] \cup \cdots \cup [t_r, t_{r+1}]$, with $t_1 = -\infty$ and $t_{r+1} = \infty$ representing unbounded sets. Interval sets can be solved in closed form by polynomial inequalities denoting intersection with the primitive linear and quadratic constraints (7–10) and (12). Fig. 5 illustrates the process for a linear path and a single collision constraint.

Let us consider a general primitive constraint $g_k(y) = y^T Q y + y^T p + r \leq 0$, with $y = (c_1, c_2, d_1, d_2)$ the coordinates

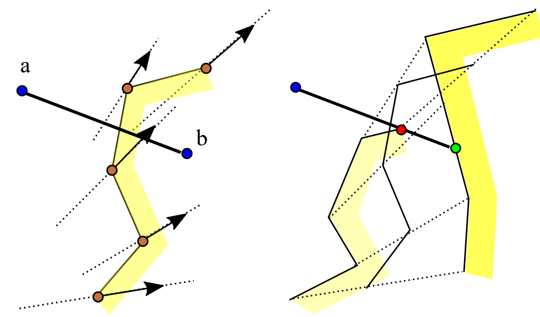

Fig. 5: (a) Given a linear search direction, vertices of the shape will be displaced simultaneously along lines. (b) The range of displacements for which the shape obeys the $(coll, a, b)$ constraint is determined analytically.

of the vertices of an edge. The trajectory $y(t)$ moves along a line / ellipse in $\mathbb{R}^4$ for Gibbs / HMC respectively, since vertices are linear functions of state. We first determine a set of roots in t such that $g_k(y(t)) = 0$ as follows.

For a linear constraint and linear trajectory $y(t) = y_0 + vt$, the root satisfies a linear equation $p^T y_0 + tp^T v + r = 0$ For a quadratic constraint and linear trajectory, the roots of $y_0^T Q y_0 + 2tv^T Q y_0 + t^2 v^T Q v + p^T y_0 + tp^T v + r \leq 0$ are determined by the quadratic equation.

For elliptical trajectories, we solve for roots of $y(t) = y_0 \cos(t) + v \sin(t)$ by introducing variables $c = \cos(t)$, $s = \sin(t)$, with $c^2 + s^2 = 1$. Then, a linear equality can be rewritten to yield a quadratic equation in s. Quadratic constraints can be solved to produce a degree 4 polynomial in s, whose roots are determined using characteristic polynomials. Each root of s yields two possible roots of $t = \pm \sin^{-1}(s)$.

The roots calculated thusly split the number line into sections, and the value of the inequality on each section $[t_i, t_{i+1}]$ could either be positive or negative. Due to numerical errors, best results are achieved by checking the value of the constraint away from the roots, e.g., at interval midpoints. The final feasible set corresponding to (3) is constructed by intersecting ($\max$ operations), unioning ($\min$ operations), and taking the complement (negation) of primitive interval sets.

F. Empirical performance

All methods in this paper are implemented in the Python programming language, and experiments are conducted on a single core of a 2.60GHz Intel Core i7 PC. Note that these algorithms can be almost trivially parallelized, and would also benefit from implementation in a compiled language.

Measuring sampling performance requires accounting for the autocorrelation of the sequence (particularly in the MH algorithm, as illustrated in Fig. 6). We measure the performance of each MCMC technique by the Effective Sample Time (EST), which estimates the amount of computation time needed to generate one effectively independent draw. EST is a function of total computation time T and Effective Sample Size ESS given by $EST = T/ESS$. Fig. 7 reports performance for all four sampling techniques on three problems.

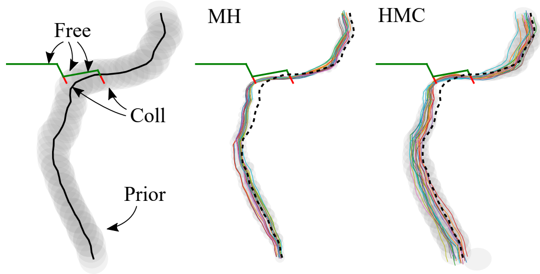

Fig. 6: Illustrating the Sloper problem with five constraint segments. Metropolis-Hastings (MH) samples exhibit strong autocorrelation and bias in estimating the mean on the lower portion of the terrain (dotted line), while the HMC method is far less autocorrelated and biased. Each plot shows 20 samples drawn at random from sets of 1,000 and 100 samples for MH and HMC, respectively.

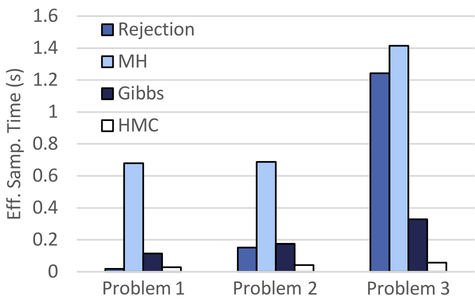

Fig. 7: Effective Sample Time for four sampling techniques (lower is better) over three problems whose constraints are increasingly restrictive. The performance of rejection sampling rapidly degrades when highly constrained, while Gibbs and HMC are more tolerant. Problem 3 is illustrated in Fig. 6.

Problems 1, 2, and 3 have 3, 3, and 5 constraints, respectively, and the fraction of the prior that obeys constraints is approximately 23%, 2.4%, and 0.2%. Rejection sampling performs best on the least restrictive problems, but HMC outperforms all other methods when highly constrained. Although each HMC sample is more costly, it achieves higher ESS because the rejection rate is 0 and autocorrelation is quite close to 0.

V. Optimizing Exploration Plans

Given a path $p(t)$, let E_d denote the event that docking is successful during execution, i.e., f_{load} is resisted at the robot's stopping point. The *docking probability* is given by:

$$P(E_d|p(t),H) = \int_X P(E_d|p(t),x)P(x|H)dx. \tag{14}$$

Since we assume no stochasticity in the robot's motion, $P(E_d|p(t),x)$ is a deterministic function $E_d(p(t),x) \to \{0,1\}$ which can be determined by simulation, because the shape is known given x. The goal of the path planner is to determine $p(t)$ starting at the current state p_0 to maximize the weighted sum of (14) and an exploration bonus.

Minimizing the speed of evaluating (14) is essential because the planner will need to evaluate many potential docking paths. Given the Monte Carlo samples $x^{(1)},\ldots,x^{(N)}$, Eq. (14) could be immediately approximated as:

$$P(E_d|p(t),H) \approx \frac{1}{N}\sum_{i=1}^{N} E_d(p(t),x^{(i)}) \tag{15}$$

which would require N deterministic simulations of the robot's motion model along the path $p(t)$ with given shapes $x^{(i)}$. For a path consisting of m segments, evaluating (15) is an $O(mnN)$ operation. We present a probabilistic simulation method that is more computationally efficient under the assumption that $P(X|H)$ is well-approximated by a Gaussian distribution. This new method runs in $O(mn)$ time per path.

Moreover, we make the assumption that resistance is only desired along the final segment of the path. Our path planner enforces that a path should terminate in a segment that crosses the expected midpoint of an edge e in the direction $-f_{load}$. We call this the *optimal terminal segment* for e. The objective function is nearly unaffected after departing sufficiently far from the shape, so the main question is how to optimize the terminal segment so that estimated docking probability (now an $O(n)$ operation) is maximized and the remainder of the path has nearly 0 probability of collision.

A. Probabilistic Simulation

Probabilistic simulation evaluates the probability of stopping at every vertex or edge of the shape while executing a given path. Remarkably, we are able to do so *without specifying the location of the vertex or edge* under the assumption of a Gaussian shape distribution. The procedure is based on a primitive operation that simulates the execution of a compliant move along a single line segment of the path $\overline{ab}$.

The method is based on summing the probability that the robot stops at a shape feature F (vertex or edge) given that it makes contact with some other feature F', then slides to F and gets stuck. Since the shape has a known winding, we denote these movements as "slide right" and "slide left". We assume for simplicity that the robot does not break then reacquire contact with the shape, although this method could be extended to handle such cases. Let $l(F)$ and $r(F)$ denote the feature immediately to the left and right of F, respectively, as viewed from the exterior. Let us also denote the primitive events C_F, where the motion collides with F; K_F, where the motion sticks on F; SL_F, where the motion slides left on F; and SR_F where the motion slides right on F.

The robot stops at F (event S_F) iff one of the following disjoint events happen:

- $C_F \wedge K_F$: collide with F and gets stuck, or
- $L_{r(F)} \wedge K_F$: slide left from $r(F)$ and get stuck, or
- $R_{l(F)} \wedge K_F$: slide right from $l(F)$ and get stuck.

The robot slides left from F (event L_F) if one of the following disjoint events happen:

- $C_F \wedge SL_F$: collide with F and slide left, or
- $L_{r(F)} \wedge SL_F$: slide left from $r(F)$ and slide left again.

Similarly, it slides right (event R_F) if either $C_F \wedge SR_F$ or $R_{l(F)} \wedge SR_F$ occurs. Because the events are disjoint, we have

- $P(S_F) = P(C_F, K_F) + P(L_{r(F)}, K_F) + P(R_{l(F)}, K_F)$,
- $P(L_F) = P(C_F, SL_F) + P(L_{r(F)}, SL_F)$,
- $P(R_F) = P(C_F, SR_F) + P(R_{l(F)}, SR_F)$,

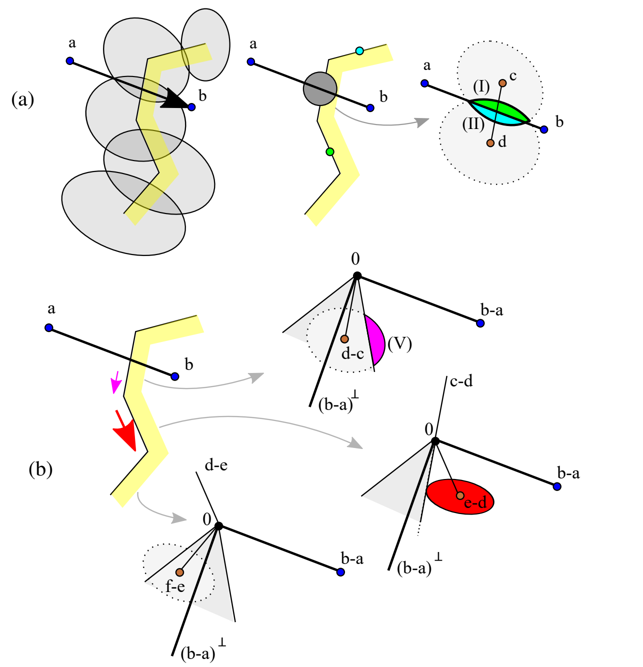

Fig. 8: Illustrating probabilistic simulation. (a) To determine the edge collision likelihoods (illustrated as size of circles) the segment collision conditions are checked against the joint distribution over endpoints c and d. (b) Determining slide-right probabilities for three edges under compliance. The first slip occurs with moderately low probability. The second slide occurs with high probability conditional on the first slip. The third slide probability is nearly 0, since the outgoing edge is far more likely to induce a slip left.

Applying conditioning, we obtain recursive linear equations

$$\begin{aligned} P(S_F) =& P(C_F, K_F) + P(L_{r(F)})P(K_F|L_{r(F)}) \\ &+ P(R_{l(F)})P(K_F|R_{l(F)}) \end{aligned} \tag{16}$$

$$P(L_F) = P(C_F, SL_F) + P(L_{r(F)})P(SL_F|L_{r(F)}) \tag{17}$$

$$P(R_F) = P(C_F, SR_F) + P(R_{l(F)})P(SR_F|R_{l(F)}). \tag{18}$$

Sections V-A1 and V-A2 describe how to calculate $P(C_F, \cdot)$, $P(\cdot|L_{r(F)})$, and $P(\cdot|R_{l(F)})$, with "$\cdot$" standing in for a primitive event. Once calculated, the system of equations can be solved for all features of the shape in $O(n)$ time using sparse matrix inversion. The overall probability of docking is $\sum_{v\in V} P(S_v) + \sum_{e\in E} P(S_e)$.

The system of equations can further be simplified under certain conditions. Because each vertex v has no volume, $P(C_v) = 0$. In a non-compliant motion model, all $P(L_F)$ and $P(R_F)$ probabilities are 0. In the compliant model, $P(K_e|L_{r(e)}) = P(K_e|R_{l(e)}) = 0$ and $P(SL_e|L_{r(e)}) = P(SR_e|R_{l(e)}) = 1$ for all edges e because if a robot slips on a vertex, it will continue until the next vertex. This is because the direction of force application is constant and exceeds the available friction along the entire length of the edge.

1) Probability of Contact: The probability $P(C_e)$ that contact occurs for a edge $e = \overline{cd}$ is approximately the integrated density of $P(c, d|H)$ restricted to the feasible set (7–10) (Fig. 8.a). Adding the stick or slip constraints adds two additional linear inequalities in the form (12).

To estimate the integrated density quickly, for each edge we produce the 4-D Gaussian approximation $P(c, d|H) \approx N(\mu_{cd}, \Sigma_{cd})$. We linearize the quadratic term of (7,8) about $y = \mu_{cd}$ and produces 6 linear inequalities of the form $Ay + b \leq 0$ with $y = (c_1, c_2, d_1, d_2)$. We approximate the probability that the inequalities are satisfied by assuming independence of three pairs of inequalities, but allowing dependence in each pair. To calculate that the probability that a pair of inequalities $a_1^T y + b_1 \leq 0$ and $a_2^T y + b_2 \leq 0$ are mutually satisfied, we define $A_{12} = [a_1 a_2]^T$ and $b_{12} = [b_1 b_2]^T$ and transform the endpoint distribution to a bivariate Gaussian $N(A_{12}\mu_{cd} + b_{12}, A_{12}\Sigma_{cd}A_{12}^T)$ and evaluate the probability integral over the quadrant $(-\infty, 0] \times (-\infty, 0]$. This evaluation can be done accurately using a low degree quadrature [6].

2) Probability of sticking/sliding: The stick event K_v at a vertex v is equivalent to a cone condition at the extrema of the friction cones of the outgoing edges $r(v) = \overline{vw}$ and $l(v) = \overline{uv}$:

$$\overrightarrow{ba} \in Cone(\overrightarrow{wb}^\perp + \mu\overrightarrow{wb}, \overrightarrow{vu}^\perp - \mu\overrightarrow{vu}). \tag{19}$$

SL_v is equivalent to $\overrightarrow{ba} \in Cone(\overrightarrow{vu}^\perp - \mu\overrightarrow{vu}, -\overrightarrow{vu})$, and SR_v is equivalent to $\overrightarrow{ba} \in Cone(\overrightarrow{wb}, \overrightarrow{wb}^\perp + \mu\overrightarrow{wb})$. As before, cones are transformed to inequalities in u, v, and w (Fig. 8.b).

Independence is not appropriate to assume in $P(K_v|L_{r(v)})$ and $P(K_v|R_{l(v)})$, because sliding provides significant information about the normal of the originating edge. Specifically, if $L_{r(v)}$ occurs, then it is certain that $\overrightarrow{ba} \notin Cone(\overrightarrow{wb}^\perp + \mu\overrightarrow{wb}, \overrightarrow{wb}^\perp - \mu\overrightarrow{wb})$. Hence, for K_v to occur after a left slip, the more restrictive condition $\overrightarrow{ba} \in Cone(\overrightarrow{wb}^\perp - \mu\overrightarrow{wb}, \overrightarrow{vu}^\perp + \mu\overrightarrow{vu}^\perp)$ must be satisfied. Similarly, for K_v to occur after $R_{l(v)}$, $\overrightarrow{ba} \in Cone(\overrightarrow{wb}^\perp + \mu\overrightarrow{wb}, \overrightarrow{vu}^\perp + \mu\overrightarrow{vu})$ must be satisfied. Ignoring long-range dependencies, we approximate $P(K_v|L_{r(v)}) \approx P(K_v|SL_{r(v)}) = P(K_v, SL_{r(v)})/P(SL_{r(v)})$.

We apply the same conditional dependency to rightward sliding and continued sliding. $P(SL_v, SL_{r(v)})$ is evaluated using the constraints $\overrightarrow{ba}^{\perp T}\overrightarrow{wb}^\perp + \mu\overrightarrow{wb} \geq 0$ and $\overrightarrow{ba}^{\perp T}\overrightarrow{vu}^\perp + \mu\overrightarrow{vu} \geq 0$. This is exact as long as $\overline{vw}$ does not turn to the left of $\overline{ab}$, e.g., the robot does not separate from the shape at v.

3) Illustration: Fig. 9 plots the simulated distributions for both non-compliant and compliant docking on the 3 Ledges example. Lines give the sticking probability for three different candidate sites. Observe that without compliance, the probability of sticking at any vertex is zero, but with compliance, the end effector can slide after making contact, which increases the overall probability of docking.

B. Optimal Path Planning

Let $\mathcal{P}$ denote the set of previously executed paths. The overall objective function adds to the docking probability an exploration bonus term as follows:

$$J(p) = P(E_d|p, H)B\left(\frac{1}{w}\min_{p\in\mathcal{P}} d(p, p')\right) \tag{20}$$

where w is the bonus weight and $d(p, p')$ measures some notion of path-wise distance. We set $d(p, p')$ to measure the distance between endpoints of the terminal segments of the

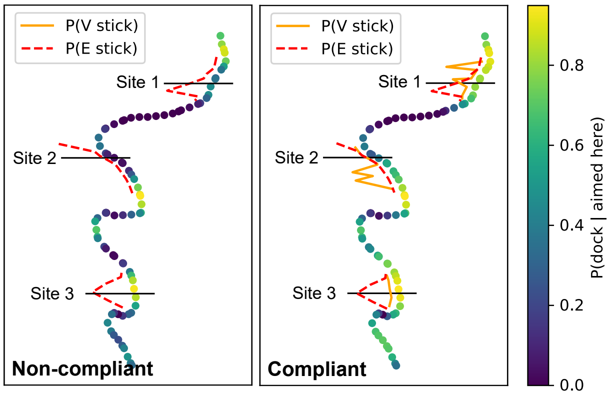

Fig. 9: Probabilistic simulation with non-compliant and linearly compliant motion. The shape distribution is the same as in Fig. 6, but only the mean shape is shown. Three docking sites are chosen, and probability of sticking on nearby features is drawn as horizontally offset lines. Circles are shaded by the probability of docking in a horizontal loading direction, if aimed through the center of circle.

paths. The bonus factor $B(z) = 1 - \exp(-z)$ transforms the domain $[0, \infty)$ to [0,1], with 0 denoting low novelty (e.g., $d(p, p') = 0$) and 1 denoting high novelty. Setting $w = 0$ leads to a greedy approach, but having a small weight helps account for sensor noise and errors in the inference model. We set w equal to the spatial resolution of the contact detector.

The planner maintains an optimal path $p^\star$, docking probability $P^\star_{dock}$, and function value $J^\star$, and proceeds as follows:

1) Initialize $p^\star \leftarrow nil$, $P^\star_{dock} \leftarrow 0$, and $J^\star \leftarrow \infty$.
2) For each edge in order of increasing $P(C_e, K_e)$ for e's optimal terminal segment, repeat:
3) Perform docking probability estimation. If $P(S_e) < P^\star_{dock}$ or $J(p) < J^\star$, it cannot be optimal, so skip to the next edge.
4) Plan a collision-free path p ending in e. If successful, store $p^\star \leftarrow p$, $P^\star_{dock} \leftarrow P(S_e)$, and $J^\star \leftarrow J(p)$.

In Step 4, we first establish a *likely obstacle region* (LOR), a free-space region in which collision has moderate probability. Its complement is the *unlikely obstacle region* (UOR). LOR is obtained by taking convex hulls of each edge over the shape estimation samples, and then performing a union operation. The planner works backward from the terminal point, which lies in LOR. It first finds a path to the boundary of LOR, starting in direction f_{load}, while maintaining the invariant that clearance away from the mean shape is monotonically increasing [35]. Once the path exits LOR, it plans a path to the start point using UOR as free-space. Shortest paths through UOR can be planned to all vertices of LOR quickly using standard methods, e.g., a visibility graph.

C. Experimental results

Fig. 10 shows the docking success rate and cycle count on three problems, where 10 ground truth shapes are sampled at random. In Sloper, only 5/10 ground truth shapes had a feasible solution. Our technique is compared with 1) a growing-window (GW) technique that starts at the site the most likely to dock successfully, then attempts docking at increasingly distant sites, and 2) an information-gain (IG) technique that alternates between one greedy docking step and two information-gain steps, using our HMC estimator to determine a shape distribution. The same path planner is used for all techniques. Note that the standard deviation for cycle count is generally high, since some instances are solved luckily on the first try, while others require dozens of cycles. Our method never failed on a feasible instance, and found a solution with fewer executions than GW or IG. Also, the modest cycle count on Sloper indicates that our method terminates quickly on infeasible problems by correctly estimating a low likelihood of feasibility.

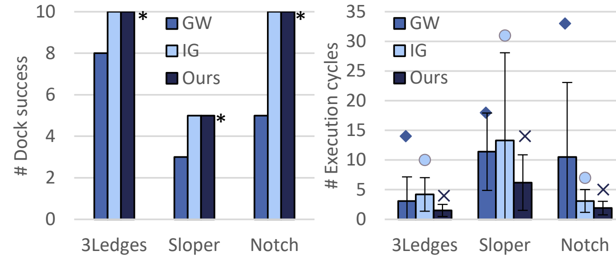

Fig. 10: Docking success rate and number of execution cycles in three problems, over 10 randomly sampled ground truth shapes. The growing-window (GW) heuristic fails in many instances. The information-gain strategy (IG) and our technique (Ours), both using our shape estimator, are more successful. Ours uses fewer execution cycles than both GW and IG. (*: ground truth success rate. Error bars: cycle count std. dev. Dots: max cycles.)

Supplemental videos at http://motion.pratt.duke.edu/locomotion/tactile.html show our technique in action with a model of the Robosimian robot and a 3D climbing wall scan generated via photogrammetry in a realistic physics simulation (Fig. 1). Given a simulated noisy vertical laser scan, our technique generates docking trajectories for a hook end effector along a 2D plane. An operational space controller performs guarded moves using a force sensor to detect collision. The tactile exploration method attempts to dock at multiple sites in response to failed docking moves.

VI. Conclusion

This paper presented a Bayesian tactile exploration controller for docking a point against uncertain shapes. Its two technical contributions include 1) Hamilton Monte Carlo shape sampling, which outperforms other sampling methods, and 2) a probabilistic simulator that quickly computes probability of docking for Gaussian shape models under compliance and friction. The resulting controller is reliable and usually requires few cycles to localize docking sites. In ongoing work, we are attempting to evaluate this technique on the physical Robosimian robot. Future work may consider generalization to other geometric representations, such as point clouds, occupancy grids, and 3D meshes.

Acknowledgment

The author would like to thank Hayden Bader, Gao Tang, and Changrak Choi for helping proofread this manuscript. This research was supported by NSF NRI grant #1527826.

References

[1] T. Bretl. Motion planning of multi-limbed robots subject to equilibrium constraints: The free-climbing robot problem. *Int'l. Journal of Robotics Research*, 25(4):317–342, 2006.

[2] S. Chitta, P. Vemaza, R. Geykhman, and D. D. Lee. Proprioceptive localilzatilon for a quadrupedal robot on known terrain. In *IEEE Int'l. Conf. on Robotics and Automation*, pages 4582–4587. IEEE, 2007.

[3] V. N. Christopoulos and P. Schrater. Handling shape and contact location uncertainty in grasping two-dimensional planar objects. In *IEEE/RSJ Int'l. Conf. Intel. Robots and Systems*, pages 1557–1563, Oct 2007. doi: 10.1109/IROS.2007.4399509.

[4] S. Dragiev, M. Toussaint, and M. Gienger. Gaussian process implicit surfaces for shape estimation and grasping. In *IEEE Int'l. Conf. on Robotics and Automation*, pages 2845–2850. IEEE, 2011.

[5] S. Dragiev, M. Toussaint, and M. Gienger. Uncertainty aware grasping and tactile exploration. In *IEEE Int'l. Conf. on Robotics and Automation*, pages 113–119. IEEE, 2013.

[6] Z. Drezner and G. O. Wesolowsky. On the computation of the bivariate normal integral. *Journal of Statistical Computation and Simulation*, 35(1-2):101–107, 1990.

[7] M. F. Fallon, M. Antone, N. Roy, and S. Teller. Drift-free humanoid state estimation fusing kinematic, inertial and lidar sensing. In *IEEE-RAS Int'l. Conf. on Humanoid Robots (Humanoids)*, pages 112–119. IEEE, 2014.

[8] W. E. L. Grimson and T. Lozano-Perez. Model-based recognition and localization from sparse range or tactile data. *Int'l. Journal of Robotics Research*, 3(3):3–35, 1984.

[9] R. Hadsell, J. A. Bagnell, D. F. Huber, and M. Hebert. Accurate rough terrain estimation with space-carving kernels. In *Robotics: Science and Systems*, volume 2009, 2009.

[10] P. Hebert, T. Howard, N. Hudson, J. Ma, and J. W. Burdick. The next best touch for model-based localization. In *IEEE Int'l. Conf. on Robotics and Automation*, pages 99–106, 05 2013. ISBN 978-1-4673-5641-1.

[11] M. A. Hoepflinger, C. D. Remy, M. Hutter, L. Spinello, and R. Siegwart. Haptic terrain classification for legged robots. In *IEEE Int'l. Conf. on Robotics and Automation*, pages 2828–2833. IEEE, 2010.

[12] M. A. Hoepflinger, M. Hutter, C. Gehring, M. Bloesch, and R. Siegwart. Unsupervised identification and prediction of foothold robustness. In *IEEE Int'l. Conf. on Robotics and Automation*, pages 3293–3298. IEEE, 2013.

[13] K. Hsiao, L. Kaelbling, and T. Lozano-Pérez. Task-driven tactile exploration. In *Robotics: Science and Systems*, 2010.

[14] K. Hsiao, M. Ciocarlie, P. Brook, et al. Bayesian grasp planning. In *ICRA 2011 Workshop on Mobile Manipulation: Integrating Perception and Manipulation*, 2011.

[15] J. Ilonen, J. Bohg, and V. Kyrki. Fusing visual and tactile sensing for 3-d object reconstruction while grasping. In *IEEE Int'l. Conf. on Robotics and Automation*, pages 3547–3554. IEEE, 2013.

[16] P. Kazanzides, J. Zuhars, B. Mittelstadt, and R. H. Taylor. Force sensing and control for a surgical robot. In *IEEE Int'l. Conf. on Robotics and Automation*, pages 612–617. IEEE, 1992.

[17] J. H. Kotecha and P. M. Djuric. Gibbs sampling approach for generation of truncated multivariate gaussian random variables. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, volume 3, pages 1757–1760. IEEE, 1999.

[18] M. C. Koval, N. S. Pollard, and S. S. Srinivasa. Pre- and post-contact policy decomposition for planar contact manipulation under uncertainty. *Int'l. Journal of Robotics Research*, 35(1-3):244–264, 2016. doi: 10.1177/0278364915594474. URL https://doi.org/10.1177/0278364915594474.

[19] M. C. Koval, M. Klingensmith, S. S. Srinivasa, N. Pollard, and M. Kaess. The manifold particle filter for state estimation on high-dimensional implicit manifolds. In *IEEE Int'l. Conf. on Robotics and Automation*, May 2017.

[20] M. Li, K. Hang, D. Kragic, and A. Billard. Dexterous grasping under shape uncertainty. *Robotics and Autonomous Systems*, 75:352–364, 2016.

[21] R. Li, R. Platt, W. Yuan, A. ten Pas, N. Roscup, M. A. Srinivasan, and E. Adelson. Localization and manipulation of small parts using gelsight tactile sensing. In *IEEE/RSJ Int'l. Conf. Intel. Robots and Systems*, pages 3988–3993. IEEE, 2014.

[22] M. Meier, M. Schopfer, R. Haschke, and H. Ritter. A probabilistic approach to tactile shape reconstruction. *IEEE Transactions on Robotics*, 27(3):630–635, June 2011. ISSN 1552-3098. doi: 10.1109/TRO.2011.2120830.

[23] M. P. Murphy, B. Aksak, and M. Sitti. Gecko-inspired directional and controllable adhesion. *Small*, 5(2):170–175, 2009.

[24] A. Pakman and L. Paninski. Exact hamiltonian monte carlo for truncated multivariate gaussians. *Journal of Computational and Graphical Statistics*, 23(2):518–542, 2014.

[25] A. Parness. Anchoring foot mechanisms for sampling and mobility in microgravity. In *IEEE Int'l. Conf. on Robotics and Automation*, pages 6596–6599. IEEE, 2011.

[26] S. Patil, J. Van Den Berg, and R. Alterovitz. Estimating probability of collision for safe motion planning under gaussian motion and sensing uncertainty. In *IEEE Int'l. Conf. on Robotics and Automation*, pages 3238–3244. IEEE, 2012.

[27] A. Petrovskaya and O. Khatib. Global localization of objects via touch. *IEEE Trans. Robotics*, 27(3):569–585, June 2011. ISSN 1552-3098. doi: 10.1109/TRO.2011.

2138450.

[28] C. Phillips-Grafflin and D. Berenson. Planning and resilient execution of policies for manipulation in contact with actuation uncertainty. In *Workshop on the Alg. Found. Robotics*, 2016.

[29] R. Platt, F. Permenter, and J. Pfeiffer. Using bayesian filtering to localize flexible materials during manipulation. *IEEE Trans. on Robotics*, 27(3):586–598, June 2011. ISSN 1552-3098. doi: 10.1109/TRO.2011.2139150.

[30] C. Rosales, A. Ajoudani, M. Gabiccini, and A. Bicchi. Active gathering of frictional properties from objects. In *IEEE/RSJ Int'l. Conf. Intel. Robots and Systems*, pages 3982–3987. IEEE, 2014.

[31] D. Santos, B. Heyneman, S. Kim, N. Esparza, and M. R. Cutkosky. Gecko-inspired climbing behaviors on vertical and overhanging surfaces. In *IEEE Int'l. Conf. on Robotics and Automation*, pages 1125–1131. IEEE, 2008.

[32] A. Shapiro, A. Greenfield, and H. Choset. Frictional compliance model development and experiments for snake robot climbing. In *IEEE Int'l. Conf. on Robotics and Automation*, pages 574–579. IEEE, 2007.

[33] M. Spenko, G. C. Haynes, J. Saunders, M. R. Cutkosky, A. A. Rizzi, R. J. Full, and D. E. Koditschek. Biologically inspired climbing with a hexapedal robot. *Journal of Field Robotics*, 25(4-5):223–242, 2008.

[34] K. Walas. Terrain classification and negotiation with a walking robot. *Journal of Intelligent & Robotic Systems*, 78(3-4):401, 2015.

[35] R. Wein, J. Van Den Berg, and D. Halperin. Planning high-quality paths and corridors amidst obstacles. *Int'l. Journal of Robotics Research*, 27(11-12):1213–1231, 2008.

[36] Z. Yi, R. Calandra, F. Veiga, H. van Hoof, T. Hermans, Y. Zhang, and J. Peters. Active tactile object exploration with gaussian processes. In *IEEE/RSJ Int'l. Conf. Intel. Robots and Systems*, pages 4925–4930. IEEE, 2016.

[37] M. Yim, S. Homans, and K. Roufas. Climbing with snake-like robots. *IFAC Proceedings Volumes*, 34(4):7–12, 2001.