Bayesian Tactile Exploration for Compliant Docking with Uncertain Shapes

Kris Hauser *Senior Member, IEEE*

***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.**

***Index Terms*—Probability and statistical methods, Force and tactile sensing, Motion and path planning, Climbing robots.**

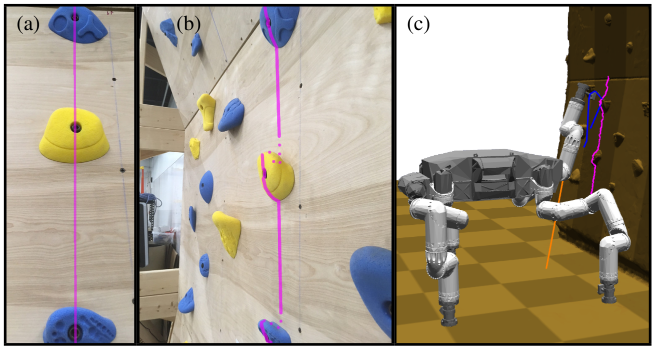


Fig. 1. (a) A climber's view of potential holds from underneath does not reveal the size and shape of pockets. (b) Shown from a side view, occluded regions from the original view are indicated by dotted lines. (c) Tactile exploration can be used by a climbing robot to probe the hidden terrain shape.

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.

This paper studies the problem of using tactile sensing to securely "dock" an end effector against an environment at a location such that a given load is resisted. This problem is encountered in several contexts, such as industrial assembly, spacecraft docking, walking on terrain of unknown friction or shape, and pulling or pushing objects during manipulation. However, this paper specifically addresses the context of enabling a rock climbing robot, equipped with a force/torque sensor, to acquire secure hand holds by feeling the terrain, like a human climber would. Because the margin of error in climbing is so small, we have observed that naïve climbing strategies suffer from even small amounts of localization error, sensor noise, and occlusion: the difference between a successful climb and a catastrophic fall can be mere millimeters.

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 is suitable for 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.

To dock efficiently, robots need a deeper understanding of how to unify probabilistic reasoning, computational geometry, compliance, and contact mechanics. To simplify the problem, this paper addresses the setting of planar movement. 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

K. Hauser is with the Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA, e-mail: kkhauser@illinois.edu.

Manuscript received April 19, 2005; revised August 26, 2015.

low docking probability) in only a few attempts. The method is applied to the Robosimian quadruped docking a hook end effector in 3D terrain, both in a realistic physics simulation and on the physical robot.

II. Related Work

Prior gecko, insect, and snake-like climbing robots [1], [2], [3], [4], [5], [6] 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 [7], 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 [8] and terrain property estimation [9]. 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 [10]. 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 [11] or unknown terrain shape observed by sensors [8]. Tactile sensors have been used to classify terrain friction and local shapes of contact points [9]. 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 [13]. In manipulation, tactile feedback has been used for localization [14], texture identification [15], and shape classification [16], [17] of familiar objects. It has also been used for estimating the location of distinctive features [18], like buttons in textiles [19] and localizing flat objects using texture and high-resolution tactile sensors [20]. 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 [10] and Gaussian processes [21], [22], [23] 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 [24]. 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 [25]. 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 [26] and 3D grasps [27]. Each of these techniques uses sampling for success probability estimation, which is advantageous for parametric object models [28] and deformable object shape [26] 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 [27].

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 [29]. Information gain has been used as a metric for choosing localization actions before manipulating an object [28] and for addressing the exploration vs exploitation tradeoff in goal-directed grasping [30].

Partially-observable Markov decision processes (POMDP) are a principled approach to optimize active goal-directed manipulation [31], [32], 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 [33]. 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 [34]. Novel contributions include the simulation of compliant motion with friction, and an improved probability estimate using truncated bivariate Gaussians rather than univariate ones.

This paper is an extended version of a conference publication [35] that includes 1) more details on the assumptions, Monte Carlo estimation, and planning procedure, 2) computational complexity analysis, and 3) experiments conducted on the physical robot.

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 $Z \sim N(0, I_m)$.

Each column of the basis matrix A encodes a deviation from the mean shape that is statistically independent from

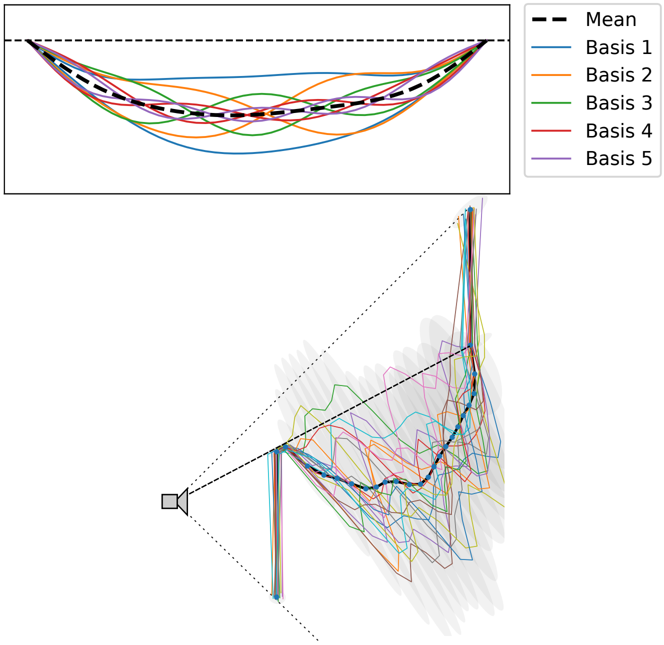


Fig. 2. Top: The mean and first 5 basis vectors for the hidden shape behind an occluding edge, as determined by sampling a Brownian bridge under the constraint $y \leq 0$ and performing PCA on the samples. Bottom: prior distribution of a sensed terrain under localization, depth estimation, and occlusion uncertainty. Several samples from the prior are drawn.

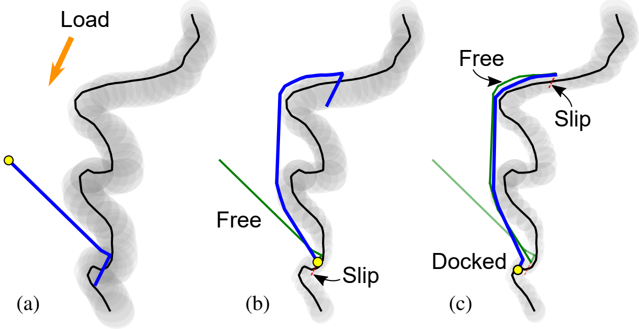


Fig. 3. (a) Bayesian tactile exploration to achieve a diagonally pulling load, with no compliance. The estimated shape distribution is shown in grey and the ground truth is drawn in black. Upon executing the initial plan to the lower ledge, the robot makes contact but slips, and the shape distribution 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.

the remaining columns. Hence, it must be customized for the given sensor and should include localization accuracy, sensor noise characteristics, and assumptions about occluded shapes. For example, a sensor's localization error in translation, with standard deviations (σ_x, σ_y) in the x–y axes, can be encoded as two columns $\begin{bmatrix}\sigma_x & 0 & \sigma_x & 0 & \cdots & \sigma_x & 0\end{bmatrix}^T$ and $\begin{bmatrix}0 & \sigma_y & 0 & \sigma_y & \cdots & 0 & \sigma_y\end{bmatrix}^T$, because each vertex in the shape is affected equally by a given localization error. Using the small-angle approximation, a sensor's localization error in rotation, with standard deviation σ_θ, is encoded in the column $\sigma_\theta \begin{bmatrix}-v_{1y} & v_{1x} & \cdots & -v_{ny} & v_{nx}\end{bmatrix}^T$, assuming the sensor is at (0,0). For uncertainty caused by occlusion, we assume the intervening geometry between two sensed points is distributed according to a Brownian bridge under the constraint that the geometry lies on the hidden side of the occluding line. Specifically, we use the HMC constraint bouncing technique described in Sec. IV-D to sample the constrained Brownian bridge, and then perform principal components analysis (PCA) to obtain a mean and fixed number of basis vectors for the hidden geometry. An example of this procedure is shown in Fig. 2.

2) *Robot motion model:* For simplicity, the robot is assumed to be a point and the shape is assumed static. It may be possible to relax the point robot assumption to handle a translating polygon, since the C-space obstacle has a polygonal shape that can be calculated via a Minkowski sum. However, because if the terrain shape distribution is Gaussian the Minkowski sum operation may introduce statistical dependence and even topological changes between C-obstacle shape samples. We proceed assuming a Gaussian shape model is appropriate, and leave the issue of other distributions to future work.

The robot is assumed to have known position relative to the reference frame. To handle localization uncertainty, this reference frame is taken as the egocentric frame, while $P(X)$ captures the localization error. The robot moves along a 2D path using guarded moves [36], 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.

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 [37]. 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. We assume that all segments are exact and the collision status is inferred precisely, which is reasonable for the precise encoders and force/torque sensors available on our system. Our method can accommodate some minor errors as described in Sec. IV-D, but it is not appropriate for highly noisy sensors.

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 ℓ new information segments $I_{k+1}, ..., I_{k+\ell}$ into the history: $H \leftarrow H \cup \{I_{k+1}, ..., I_{k+\ell}\}$, 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. 3.

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 $H = \{I_i = (a_i, b_i, s_i) | i = 1, \dots, k\}$ 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:

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

where x_u, x_v are the endpoints of an edge (u, v) specified by state x, and g_k, $k = 1, \dots, 4$ are segment-segment collision constraints to be defined in Sec. IV-B (Fig. 5, left). For each colliding segment, we require that

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

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 two additional constraints g_5 and g_6 to (2), for a total of 6 constraints per edge (Fig. 5, right).

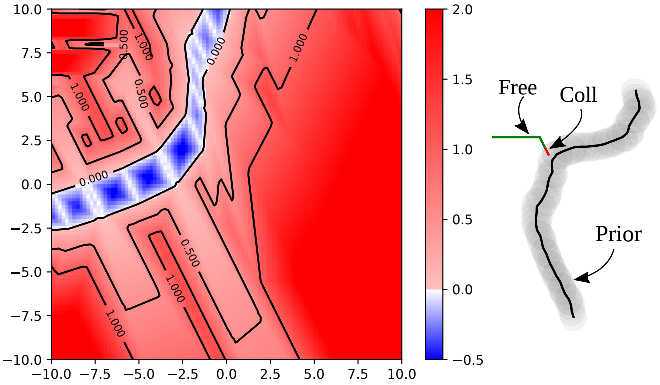


Fig. 4. Contour plot of the history-consistency constraint as illustrated on the Sloper problem (right) with two free constraints and one stick constraint. Eq. (3) is plotted as a function of basis variables z_1 and z_2, which translate the shape in the x and y directions, respectively. The function is non-smooth and the feasible set (region containing negative values) is non-convex.

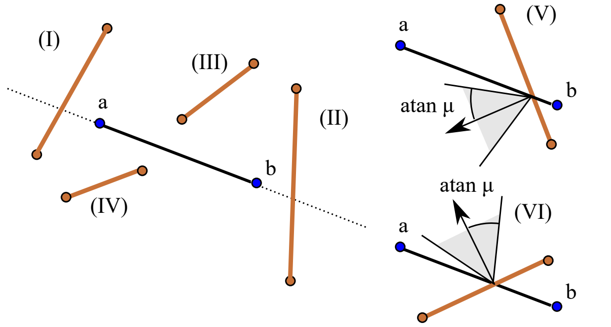


Fig. 5. Left: segment collision violations in cases (I)–(IV) correspond to violations of quadratic and linear inequalities (7)–(10), respectively. Right: stick constraint violations in cases (V) and (VI) correspond to violations of the respective elements of (12).

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

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

An example of a slice through this function is shown in Fig. 4.

B. *Segment-segment collision constraints*

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

$$a + s \cdot \overrightarrow{ab} = c + t \cdot \overrightarrow{cd}, \quad \text{with } 0 \le s, t \le 1. \qquad (4)$$

Solving for (s, t) via 2x2 matrix inversion we get

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

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

$$0 \le \begin{bmatrix} c_2 - d_2 & d_1 - c_1 \\ a_2 - b_2 & b_1 - a_1 \end{bmatrix} \begin{bmatrix} c_1 - a_1 \\ c_2 - a_2 \end{bmatrix} \le \alpha \qquad (6)$$

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

$$-y^T Qy + \left[\begin{array}{cccc} -a_2 & a_1 & a_2 & -a_1 \end{array}\right] y \leq 0 \quad (7)$$
$$y^T Qy + \left[\begin{array}{cccc} b_2 & -b_1 & -b_2 & b_1 \end{array}\right] y \leq 0 \quad (8)$$
$$\left[\begin{array}{cccc} b_2 - a_2 & a_1 - b_1 & 0 & 0 \end{array}\right] y + (a_2 b_1 - a_1 b_2) \leq 0 \quad (9)$$
$$\left[\begin{array}{cccc} 0 & 0 & a_2 - b_2 & b_1 - a_1 \end{array}\right] y + (a_1 b_2 - a_2 b_1) \leq 0. \quad (10)$$

with Q a constant 4x4 matrix:

$$Q = \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}. \quad (11)$$

(It can also be shown that $\alpha \geq 0$ must hold if these equations are simultaneously satisfied.) Eqs. (7–10) comprise the final form of the constraints $g_1, \ldots, g_4$.

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

$$\overrightarrow{ba} \in Cone(\overrightarrow{dc}^\perp + \mu \cdot \overrightarrow{dc}, \quad \overrightarrow{dc}^\perp - \mu \cdot \overrightarrow{dc}), \quad (12)$$

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 (12), so (12) can be rewritten as inequalities g_5 and g_6:

$$\begin{bmatrix} \mu \cdot \overrightarrow{b_2 a_2} - \overrightarrow{b_1 a_1} & -\overrightarrow{b_2 a_2} - \mu \cdot \overrightarrow{b_1 a_1} \\ \mu \cdot \overrightarrow{b_2 a_2} + \overrightarrow{b_1 a_1} & \overrightarrow{b_2 a_2} - \mu \cdot \overrightarrow{b_1 a_1} \end{bmatrix} \overrightarrow{cd} \leq 0, \quad (13)$$

which 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.

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. 6.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). *Metropolis-Hastings* (MH) is a common MCMC technique that takes small perturbations and accepts moves with a given acceptance probability (Fig. 6.b). We compare against MH as a baseline, but our experiments suggest that it performs poorly in constrained sampling due to strong autocorrelation. In other words, an excessive number of samples is needed for MH to move large distances in the state space.

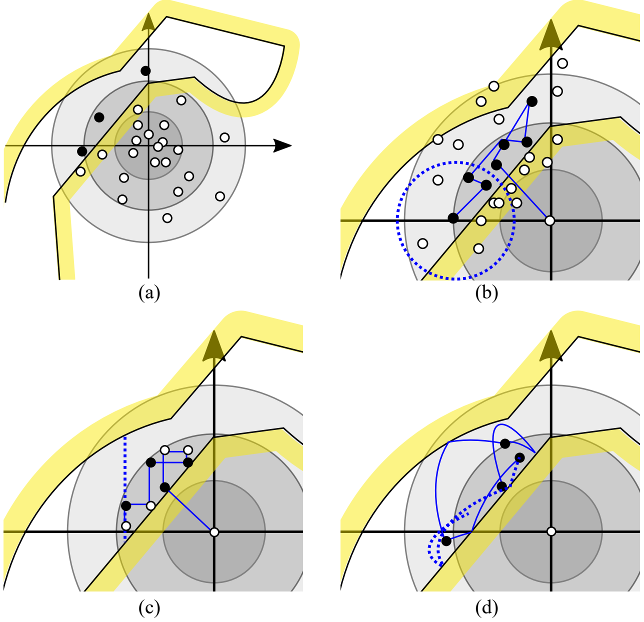


Fig. 6. Monte Carlo methods for constrained shape inference: (a) rejection sampling, (b) Metropolis-Hastings, (c) Gibbs sampling, and (d) Hamiltonian Monte Carlo. Black dots are accepted samples, white dots are rejected samples. The outlined shape illustrates the feasible set, blue paths illustrate a MCMC trajectory, and the dotted lines illustrate a sampling range.

We also consider the *Gibbs sampling* technique (Fig. 6.c), which has been applied to Gaussian distributions truncated by linear inequalities [38]. 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 valid range is a set of disjoint intervals, from which t can be sampled directly (Appendix A). The new z_i is then set to t.

Finally, we present a constraint bouncing *Hamiltonian Monte Carlo* (HMC) method (Fig. 6.d), which has been applied to Gaussian distributions under linear and quadratic inequalities [39]. 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:

$$\frac{d}{dt}z = \frac{\partial H}{\partial p}, \quad \frac{d}{dt}p = -\frac{\partial H}{\partial z}. \tag{14}$$

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.

Although for general probability distributions the dynamic equations may need to be integrated using numerical methods, in the Gaussian case the integration greatly simplifies [39]. 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

$$z(t) = z_0 \cos t + p_0 \sin t. \tag{15}$$

Moreover, the MH acceptance probability is always 1, so a step $z^{(j+1)} = z(T)$ can always be taken for any step size T. A recommended step size is $T = \pi/2$ because it tends to produce low autocorrelation [39].

For constrained sampling, 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. This maintains the reversibility of the dynamical system because reflection is symmetric. The equations of motion are then integrated forward again until another constraint is hit, or the desired total step size is reached. Again, feasible range determination is used here to determine when and whether to bounce (Sec. IV-E). Alg. 1 summarizes the overall HMC procedure for advancing the current sample $z^{(i)}$.

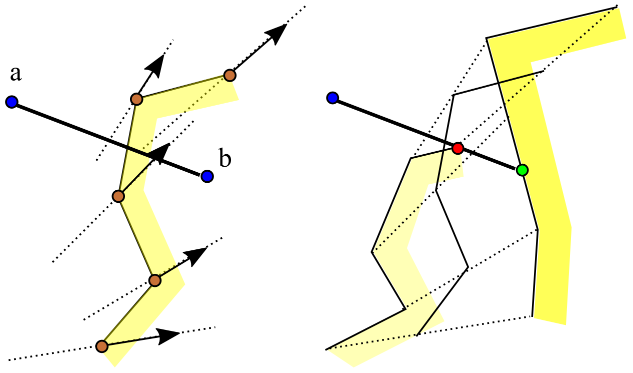
Note that MCMC methods must begin from a feasible initial seed. We find the seed by random descent of f_H from an initial state sampled from $N(0, I_m)$. If this fails after a given number of iterations, we sample another initial state and repeat. If this continues to fail, this would suggest inconsistencies in the history H, possibly caused by sensor error. After some number of random descents, we simply examine the state that maximizes the number of satisfied constraints, and delete any unsatisfied constraints from H. (In our tests we have not observed the need for this fallback procedure except when sensor data was accidentally processed improperly.)

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 (13). Fig. 7 illustrates the process for a linear path and a single collision constraint.

Algorithm 1: HMC with Constraint Bouncing

```
Input : Current feasible sample $z^{(i)}$, step size $T$ (usually $T = \pi/2$)
Output: Next sample $z^{(i+1)}$.
$z_0 \leftarrow z^{(i)}$
Sample momentum $p_0 \sim N(0, I_m)$
$t \leftarrow 0$
while $t < T$ do
    Find the feasible range $S$ of (15) under constraints (3)
    if $T - t \in S$ then
        Set $\Delta t = T - t$
    else
        Set $\Delta t$ to be the end of the interval $I$ in $S$ also containing 0
    end
    Set $z_0 \leftarrow z_0 \cos \Delta t + p_0 \sin \Delta t$
    Set $p_0 \leftarrow -z_0 \sin \Delta t + p_0 \cos \Delta t$
    Set $t \leftarrow t + \Delta t$
    if $t < T$ then                                  ▷ bounced
        Let $g(z) \leq 0$ be the constraint defining the end of interval $I$ (such that $g(z_0) = 0$)
        Let $d = \nabla g(z_0)/\|\nabla g(z_0)\|$
        Reflect $p_0 \leftarrow p_0 - 2d(d^T p_0)$.
    end
end
return $z_0$
```



(a) (b)

Fig. 7. (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.

Let us consider a general primitive constraint $g_k(y) = y^TQy + y^Tp + r \leq 0$, with $y = (c_1, c_2, d_1, d_2)$ the coordinates 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 $p^Ty_0 + tp^Tv + r = 0$. For a quadratic constraint and linear trajectory, the roots of $y_0^TQy_0 + 2tv^TQy_0 + t^2v^TQv + p^Ty_0 + tp^Tv + r \leq 0$ are determined by the quadratic equation.

For elliptical trajectories, we solve for roots of $y(t) = y_0\cos(t) + v\sin(t) + r$ by introducing variables $c = \cos(t)$, $s = \sin(t)$, with $c^2 + s^2 = 1$. Then, a linear equality satisfies $cp^Ty_0 + sp^Tv + r = 0$. Shifting and squaring, we obtain $(p^Ty_0)^2c^2 = (-sp^Tv - r)^2 = (p^Ty_0)^2(1 - s^2)$, which can be rewritten to yield a quadratic equation in s. Quadratic equalities 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. Performance

The computational cost of each MC sample is governed by the cost of feasibility checking (for rejection sampling and MH) and range determination (for Gibbs or HMC). Each feasibility check with k trajectory segments and n shape edges requires kn segment-segment collision checks, each of which is O(1).

For each Gibbs and HMC sample, kn segment-segment range determinations are required. In Gibbs sampling, since only every n'th sample is kept, each kept sample requires solving up to $4kn^2$ linear and $2kn^2$ quadratic equations. In HMC, each bounce requires solving up to $4kn$ quadratic and $2kn$ quartic equations. It is difficult to estimate the number of bounces analytically, but empirical testing suggests the number of bounces per sample is less than 10 on typical problems.

To accelerate the procedure, we can eliminate constraints that are unlikely to be relevant to the range determination procedure, such as areas of the shape far away from the robot's past trajectory. Specifically, we estimate the prior probability that each trajectory segment (a, b) and shape edge (u, v) intersect. If the probability is below a low threshold (10^{-4} in our implementation) then we do not add the pair to $f_H(x)$. Letting $C \subseteq H \times E$ be the subset of possibly-colliding pairs, we reduce the maximization domain of (1) to

$$\tilde{f}_{free,a_i,b_i}(x) = \max_{(a_i,b_i,u,v)\in C}(\ldots) \tag{16}$$

and the minimization domain of (2) to

$$\tilde{f}_{coll,a_i,b_i}(x) = \min_{(a_i,b_i,u,v)\in C}(\ldots). \tag{17}$$

With this definition each bounce requires $O(|C|)$ operations rather than $O(kn)$, and usually $|C| \ll kn$. However, there is still a chance that a sample violates one of the eliminated constraints, so to be ensured of the feasibility of a sample all kn pairs must still be ultimately tested.

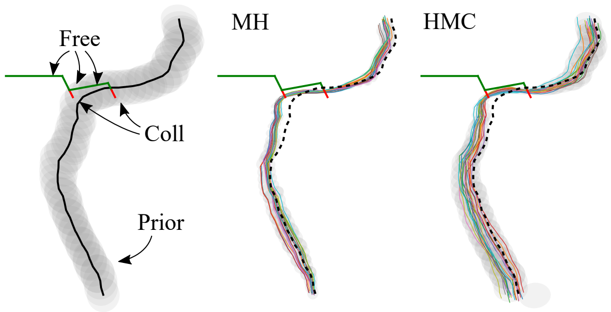


Fig. 8. Illustrating the Sloper problem with five constraint segments. Metropolis-Hastings (MH) samples exhibit strong autocorrelation and bias in estimating the mean (dotted line) on the upper and lower portions of the terrain, while the HMC method is far less autocorrelated and biased. Each plot shows 20 samples drawn at random from sets of 1,000 and 100 samples for MH and HMC, respectively.

A batch collision detection approach gathers a set of N samples, and calculates a bounding volume for samples of each shape segment (u, v). If the bounding volume does not intersect a movement segment (a, b), we can eliminate all the samples of (u, v) from further checking against (a, b). The bounding volume calculation and collision checking takes time $O(nN)$ and $O(kn)$, respectively, and so if few samples are in collision, which is typical, this reduces the amortized collision checking time per sample to $O(n + kn/N)$. If we determine that some fraction of the batch is feasible, we can sample another batch.

Beyond per-sample costs, a more complete picture of sampling performance requires accounting for the autocorrelation of the sequence (particularly in the MH algorithm, as illustrated in Fig. 8). Autocorrelation must be determined empirically. 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. 9 reports performance for all four sampling techniques on three problems. 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. 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.

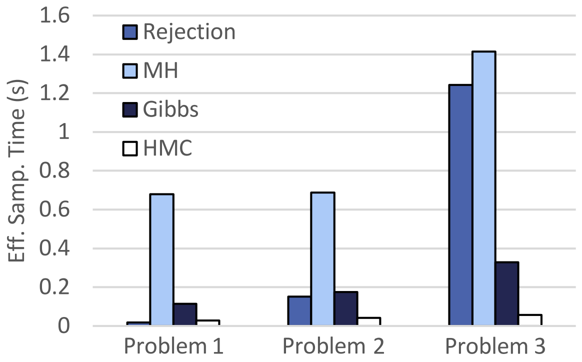


Fig. 9. Effective Sample Time for four sampling techniques (lower is better) over three problems whose constraints are increasingly restrictive. The performance of rejection sampling rapidly degrades when highly constrained, while Gibbs and HMC are more tolerant. Problem 3 is illustrated in Fig. 8.

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:

$$P(E_d|p(t), H) = \int_X P(E_d|p(t), x)P(x|H)dx. \tag{18}$$

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 (18) and an exploration bonus.

A. Probabilistic Simulation

Minimizing the speed of evaluating (18) is essential because the planner will need to evaluate many potential docking paths. Given the Monte Carlo samples $x^{(1)}, \ldots, x^{(N)}$, Eq. (18) could be immediately approximated as:

$$P(E_d|p(t), H) \approx \frac{1}{N}\sum_{i=1}^{N} E_d(p(t), x^{(i)}) \tag{19}$$

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 ℓ segments and a shape with n edges, evaluating (19) is an $O(\ell n N)$ 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 $O(\ell n)$ time per path.

Probabilistic simulation evaluates the probability that a compliant execution of a path stops at any vertex or edge of the shape. 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 $\overrightarrow{ab}$ of the path.

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 during a compliant motion, the robot remains in contact with the shape and if contact is broken, the motion stops. If desired, probabilistic simulation could be extended to handle the case where the robot separates from the shape and then comes back into contact with it, but additional probabilistic collision checking is needed for every possible separation.

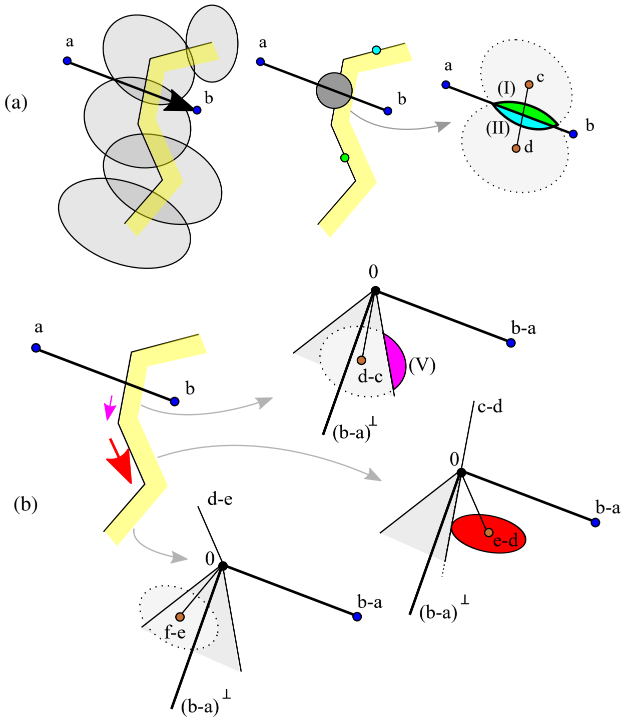


Fig. 10. Illustrating probabilistic simulation of a compliant motion from a to b. (a) To determine the edge collision likelihoods, the segment collision conditions are checked against the joint distribution over endpoints c and d. In condition (I), the motion hits the edge to the right of d, in condition (II) it hits the edge to the right of c, and otherwise $\overline{cd}$ is hit. (b) Determining slide-right probabilities for three edges under compliance. The first slip occurs under condition (V) with moderately low probability. The second slide, conditional on the first slip, occurs under the same condition with high probability. The probability of sliding a third time is nearly 0, since the outgoing edge is far more likely to induce a slip left.

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: execution of $\overline{ab}$ collides with F,
- K_F: execution of $\overline{ab}$ sticks on F,
- SL_F: execution of $\overline{ab}$ slides left on F, and
- SR_F: execution of $\overline{ab}$ slides right on F.

From these primitives, we can derive various compound events. 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 get 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

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

Applying conditioning, we obtain recursive linear equations

$$\begin{aligned} P(S_F) =& P(C_F, K_F) + P(L_{r(F)})P(K_F|L_{r(F)}) \\ &+ P(R_{l(F)})P(K_F|R_{l(F)}) \end{aligned} \tag{20}$$

$$P(L_F) = P(C_F, SL_F) + P(L_{r(F)})P(SL_F|L_{r(F)}) \tag{21}$$

$$P(R_F) = P(C_F, SR_F) + P(R_{l(F)})P(SR_F|R_{l(F)}). \tag{22}$$

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. 10.a). For each edge we produce the 4-D Gaussian approximation $P(c, d|H) \approx N(\mu_{cd}, \Sigma_{cd})$, and the integrated density can be estimated quickly using the method described in Appendix B.

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 $l(v) = \overline{uv}$:

$$\overrightarrow{ba} \in Cone(\overrightarrow{wv}^{\perp} + \mu\overrightarrow{wv}, \overrightarrow{vu}^{\perp} - \mu\overrightarrow{vu}). \tag{23}$$

SL_v is equivalent to $\overrightarrow{ba} \in Cone(\overrightarrow{vu}^{\perp} - \mu\overrightarrow{vu}, -\overrightarrow{vu})$, and SR_v is equivalent to $\overrightarrow{ba} \in Cone(\overrightarrow{wv}, \overrightarrow{wv}^{\perp} + \mu\overrightarrow{wv})$. As before, cones are transformed to inequalities in u, v, and w (Fig. 10.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 $\overrightarrow{ba} \notin Cone(\overrightarrow{wv}^{\perp} + \mu\overrightarrow{wv}, \overrightarrow{wv}^{\perp} - \mu\overrightarrow{wv})$. Hence, for K_v to occur after a left slip, the more restrictive condition $\overrightarrow{ba} \in Cone(\overrightarrow{wv}^{\perp} - \mu\overrightarrow{wv}, \overrightarrow{vu}^{\perp} + \mu\overrightarrow{vu}^{\perp})$ must be satisfied. Similarly, for K_v to occur after $R_{l(v)}$, $\overrightarrow{ba} \in Cone(\overrightarrow{wv}^{\perp} + \mu\overrightarrow{wv}, \overrightarrow{vu}^{\perp} + \mu\overrightarrow{vu})$ must be satisfied. Ignoring long-range dependencies, we approximate $P(K_v|L_{r(v)}) \approx P(K_v|SL_{r(v)}) = P(K_v, SL_{r(v)})/P(SL_{r(v)})$.

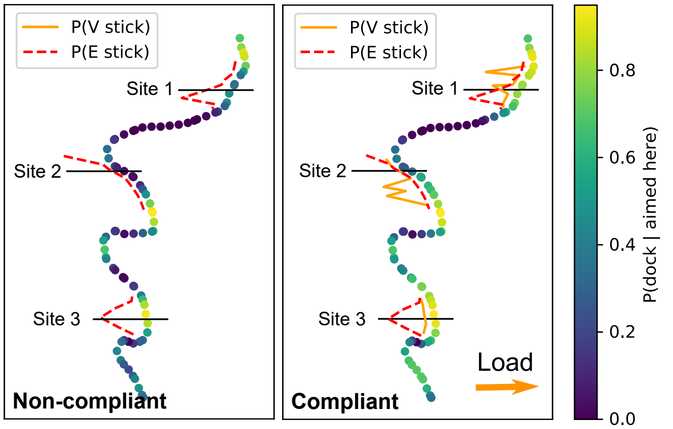


Fig. 11. Probabilistic simulation with non-compliant and linearly compliant motion. The shape distribution is the same as in Fig. 8, but only the mean shape is shown. Three docking sites are chosen, and probability of sticking on nearby features is drawn as horizontally offset lines. Circles are shaded by the probability of docking in a horizontal loading direction, if aimed through the center of circle.

We apply the same conditional dependency to rightward sliding and continued sliding. $P(SL_v, SL_{r(v)})$ is evaluated using the constraints $\overrightarrow{ba}^{\perp T}\overrightarrow{wv}^{\perp} + \mu\overrightarrow{wv} \geq 0$ 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. 11 plots the simulated distributions for both non-compliant and compliant docking on the 3 Ledges example. The terrain contains 82 vertices and 81 edges, and the entire process of constructing and solving (20–22) takes 43 ms per docking site on average. 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

Unlike a standard path planning problem, in the docking problem only the start configuration q_0 is fixed, and we need to find a path $p(t)$ that maximizes probability of docking. We make the assumption that resistance is only desired along the final segment of the path. Hence, our path planner maximizes the probability that the final path segment docks successfully, while minimizing the probability that previous segments of the path collide with the shape.

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 a prefix path connects q_0 to the terminal segment with negligible probability of collision.

To discourage this procedure from repeatedly performing low-information actions, the overall objective function adds an *exploration bonus* to the docking probability as follows:

$$J(p) = P(E_d|p, H)B\left(\frac{1}{w}\min_{p' \in \mathcal{P}} d(p, p')\right) \tag{24}$$

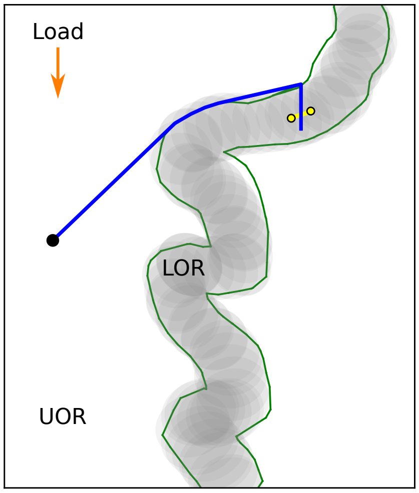


Fig. 12. The planner seeks a terminal shape edge (segment ending in open circles) to be crossed in the loading direction via the final path segment. The plan proceeds from the start (closed circle) optimally through the unlikely obstacle region (UOR) and penetrates into the likely obstacle region (LOR) to the terminal edge.

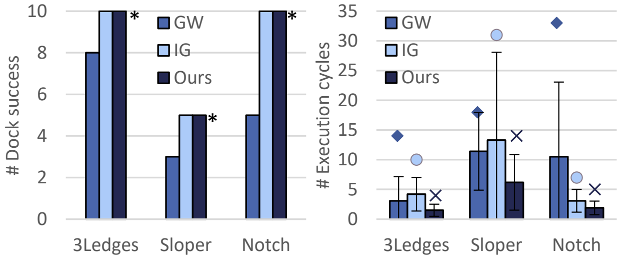
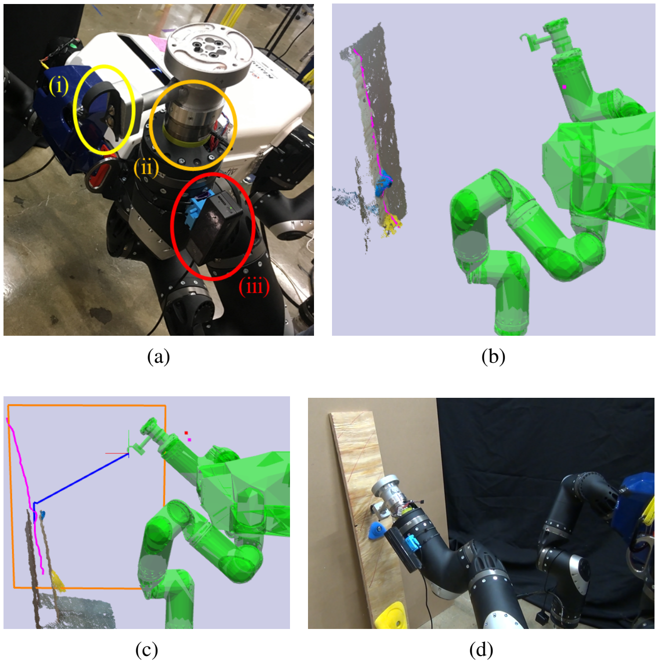


Fig. 13. Docking success rate and number of execution cycles in three problems, over 10 randomly sampled ground truth shapes. The growing-window (GW) heuristic fails in many instances. The information-gain strategy (IG) and our technique (Ours), both using our shape estimator, are more successful. Ours uses fewer execution cycles than both GW and IG. On the left plot, the asterisk * denotes ground truth success rate, and error bars denote cycle count standard deviation. In the right, the dots denote the maximum cycle count over all runs.

where $\mathcal{P}$ is the set of previously executed paths, 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 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.

Our planning algorithm starts by establishing a *likely obstacle region* (LOR), a free-space region in which collision risk exceeds a specified threshold. 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 maintains an optimal path $p^\star$, docking probability $P^\star_{dock}$, and objective function value $J^\star$, initially empty, and then iterates through candidate terminal segments and plans prefixes that are likely to be collision-free (Fig. 12). The procedure is as follows:



(a)

(b)

(c)

(d)

Fig. 14. (a) The Robosimian end effector is equipped with (i) a hook, (ii) a force/torque sensor, and an RGB-D camera. (b) The visualization shows the a 3D terrain model, which is converted to a 2D slice for planning. (c) The camera transform has a deliberate calibration error, as illustrated by the the prior shape (magenta) becoming misaligned with the RGB-D data once the robot begins moving. (d) The plan misses the targeted hold. Supplemental videos illustrate the robot recovering. (Best viewed in color)

1) Calculate LOR from shape samples.
2) Initialize $p^\star \leftarrow nil$, $P^\star_{dock} \leftarrow 0$, and $J^\star \leftarrow 0$.
3) For each edge in order of increasing $P(C_e, K_e)$ for e's optimal terminal segment, repeat:
4) 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.
5) Plan a collision-free path p ending in a segment crossing e. If successful, store $p^\star \leftarrow p$, $P^\star_{dock} \leftarrow P(S_e)$, and $J^\star \leftarrow J(p)$.

In Step 5, 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 [40]. 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. We use an incremental, lazy visibility graph method that only calculates paths to the requested goal vertices. We also note that if the start is in the LOR, then the increasing clearance method is used to determine a free start point.

VI. Experimental results

A. Simulated problems

Fig. 13 shows the docking success rate and cycle count on three problems, where 10 ground truth shapes are sampled at random. A compliant robot motion model is assumed. 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.

B. Robosimian experiments

We also demonstrate the technique on the Robosimian robot, both in simulation and on the physical robot. Given a noisy vertical depth scan, our technique generates docking trajectories for a hook end effector along a 2D plane. For simulation, a 3D climbing wall scan is generated via photogrammetry, and a noisy simulated laser sensor is used to provide the scan. Execution of the trajectory is performed in a realistic physics simulation (Fig. 1). In both cases, an operational space controller performs guarded moves, which are not tangentially compliant, using a force sensor to detect collision. The tactile exploration method attempts to dock at multiple sites in response to failed docking moves. Supplemental videos at http://motion.pratt.duke.edu/locomotion/tactile.html show additional simulation examples.

In the physical robot experiments, shown in the supplemental videos, the Robosimian is equipped with an RGB-D sensor (Intel Euclid) on its forearm and an ATI Mini-45 force/torque sensor on its wrist (Fig. 14.a). The RGB-D sensor has a minimum range of 0.55 m, which means it must be held relatively far away from the terrain for sensing (Fig. 14.b). Moreover, the transform of the camera relative to the link is deliberately not calibrated very well to illustrate the capability of our method to respond to such errors. In the example of Fig. 14.c and d, calibration error causes the robot to bump prematurely into the wall, requiring an additional estimation and planning cycle. The docking example shown in Fig. 15 requires 3 execution cycles. In all physical experiments, the security of the final hold is verified by manually pulling on the board.

Finally, reliability experiments were performed on the setup of Fig. 16, where the two holds are difficult to distinguish from camera data alone. The “top” hold (furthest from the robot) is more tenuous than the “bottom” hold, since only a thin ledge is available for docking. The calibration of the camera was improved, but camera shaking, sensor noise, and slight movements of the terrain board caused approximately 1–2 cm errors estimating the terrain shape. This error, in turn, generated significant variability in terms of which hold appears optimal. In these tests, 12 out of 13 attempted docking trials were successful. 12 were originally planned, but an extra trial was run because one trial terminated incorrectly. The reasons for failure are described in the paragraph below. Of the successful trials, the average number of cycles per trial was 1.75, with a maximum of 4 cycles. In 5/13 trials (44%), the robot attempted docking with the top hold first based on the camera data alone, but in 3/5 of those trials (60%) the robot ended up docking with the bottom hold after failing to dock on the first attempt. Overall, only 25% of trials ended on the top hold.

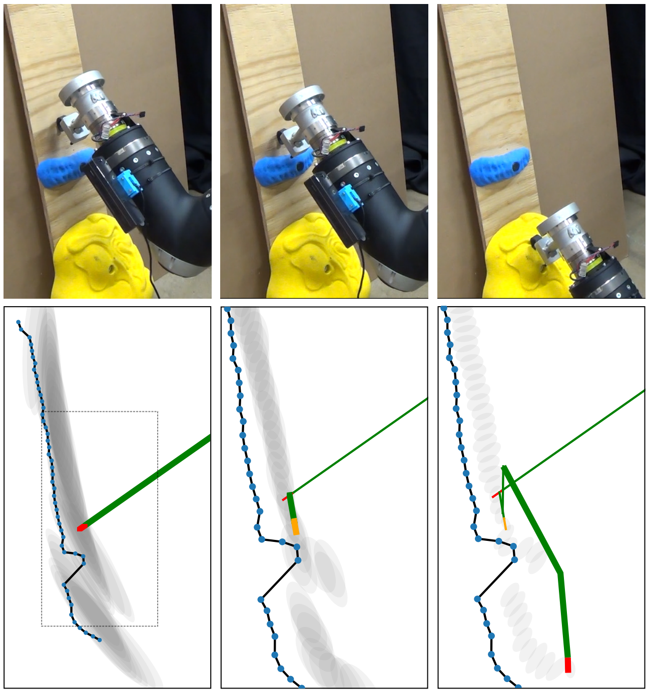


Fig. 15. Snapshots from a 3-cycle execution on the Robosimian, with camera calibration error introduced deliberately. Cycle 1 touches prematurely, a slip occurs in cycle 2, and docking is successful in cycle 3. The bottom row shows the sensed terrain shape, execution history, and inferred shape distribution (ellipses) after each cycle. The most recently executed paths are drawn in bold. Note: Images 2 and 3 show a magnification of the region marked in image 1.

The one incorrect termination was caused by a portion of the hook touching the terrain, which violated our point robot assumption and hence created a problematic sensing history, which led to planner failure. Note that an inadvertent touch also occurred in one of the successful trials, but our method was able to recover from the inconsistencies. Nevertheless, handling end effector geometry in a more principled fashion would be an important direction for future work.

VII. 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 evaluating this technique on the physical Robosimian robot.

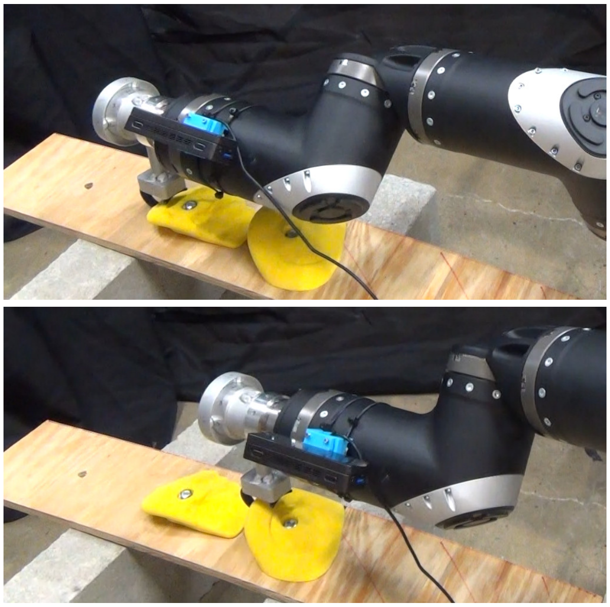


Fig. 16. Experimental setup for reliability testing. The top hold has a tiny (≈4 mm) ledge while the the bottom hold contains a large pocket. The docking direction points to the right. Due to occlusion caused by the initial vantage point of the camera, the two holds visually look quite similar.

Future work may consider generalizing our approach to other geometric representations, such as point clouds, occupancy grids, and 3D meshes. It may also be possible to extend our approach to handle a translating and rotating non-point end effector, and other manipulation tasks, like grasping and placing objects under uncertainty, that involve multiple points of contact.

Acknowledgment

This research was supported by NSF NRI grant #1527826 and performed while the author was employed by Duke University.

Appendix A
Sampling from a truncated univariate Gaussian

Let $N_{[a,b]}(\mu,\sigma^2)$ denote the univariate Gaussian with mean μ and standard deviation σ truncated to the range $[a,b]$. Many scientific computing libraries contain an inverse function transform-based subroutine to sample from a truncated zero-mean, unit variance Gaussian $x \sim N_{[a,b]}(0,1)$. (It should be noted that many implementations suffer from numerical difficulties when a is large and positive or b is large and negative; alternative approximations should be used in such cases.) The subroutine can be used to sample $x \sim N_{[a,b]}(\mu,\sigma^2)$ by first sampling $y \sim N_{[(a-\mu)/\sigma,(b-\mu)/\sigma]}(0,1)$ then performing the linear transform $x = \sigma y + \mu$.

It is straightforward to sample the restriction of a Gaussian to a disjoint union of intervals, $x \sim N_S(\mu,\sigma)$, where $S = [a_1,b_1] \cup \cdots \cup [a_n,b_n]$. First, for all $i = 1,\ldots,n$, the probability p_i of $x \sim N(\mu,\sigma)$ falling within $[a_i,b_i]$ is computed with the Gaussian cumulative distribution function (CDF). Then, an interval $[a_i,b_i]$ is chosen randomly with weight proportional to p_i, and then $x \sim N_{[a_i,b_i]}(\mu,\sigma)$ is sampled as outlined above.

Appendix B
Segment collision probability estimation

We approximate the probability that a segment $\overline{ab}$ collides with an uncertain segment $\overline{cd}$, whose four parameters $y = (c_1, c_2, d_1, d_2)$ are jointly normally distributed according to $y \sim N(\mu_{cd}, \Sigma_{cd})$. We linearize the quadratic term of (7,8) about $y = \mu_{cd}$ and combine them with (9) and (10) to produce 4 linear inequalities of the form $Ey \leq f$. The estimation problem is then approximated by the probability integral

$$P(\overline{ab} \cap \overline{cd} \neq \emptyset) \approx \int_y I[Ey \leq f] dP(y). \tag{25}$$

It is possible to compute a lower bound on this probability via the product of repeated 1D truncations [34]:

$$P(\overline{ab} \cap \overline{cd} \neq \emptyset) \geq \prod_{i=1}^{4} \left(\int_y I[e_i^T y \leq f_i] dP(y) \right) \tag{26}$$

where e_i and f_i are the i'th row of E and f, respectively. Each of these 1D truncations can be estimated via the CDF of a 1D Gaussian

$$\int_y I[e_i^T y \leq f_i] dP(y) = \int_{-\infty}^{f_i} N(z; e_i^T \mu_{cd}, e_i^T \Sigma_{cd} e_i) dz. \tag{27}$$

However, the bound is not tight when the constraints are dependent.

Our method achieves a tighter bound by assuming independence of two pairs of inequalities, but allowing dependence in each pair. Specifically, we decompose the 4 constraints into the forward/left pair (cases (I) and (III) in Fig. 5) and the backward/right pair (cases (II) and (IV).) To calculate that the probability that a pair of inequalities $e_i^T y \leq f_i$ and $e_j^T y \leq f_j$ are mutually satisfied, we define $E_{ij} = \begin{bmatrix} e_i^T \\ e_j^T \end{bmatrix}$ and $f_{ij} = \begin{bmatrix} f_i \\ f_j \end{bmatrix}$ and transform the endpoint distribution to a bivariate Gaussian

$$N(E_{ij}\mu_{cd}, E_{ij}\Sigma_{cd}E_{ij}^T) \tag{28}$$

and evaluate the probability integral over the quadrant $(-\infty, f_i] \times (-\infty, f_j]$:

$$\begin{aligned} &\int_y I[e_i^T y \leq f_i \wedge e_j^T y \leq f_j] dP(y) = \\ &\int_{-\infty}^{f_i} \int_{-\infty}^{f_j} N\left(\begin{bmatrix} z_i \\ z_j \end{bmatrix}; E_{ij}\mu_{cd}, E_{ij}\Sigma_{cd}E_{ij}^T \right) dz_j dz_i. \end{aligned} \tag{29}$$

This evaluation can be done accurately using a low degree quadrature [41]. The improved bound is then

$$\begin{aligned} P(\overline{ab} \cap \overline{cd} \neq \emptyset) \geq & \left(\int_y I[E_{13} y \leq f_{13}] dP(y) \right) \\ & \cdot \left(\int_y I[E_{24} y \leq f_{24}] dP(y) \right). \end{aligned} \tag{30}$$

Numerical experiments were conducted over 1,000 randomly sampled uncertain segments. Each side of (25) is estimated using Monte-Carlo sampling with $N = 100,000$, and the approximation has total RMSE of about 0.04. Compared to the univariate approach, the error of the bivariate approach in estimating the r.h.s. of (25) is better by 30–40%, and the error in estimating the l.h.s. is better by 5–10%.

References

[1] 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*, vol. 25, no. 4-5, pp. 223–242, 2008.

[2] M. P. Murphy, B. Aksak, and M. Sitti, "Gecko-inspired directional and controllable adhesion," *Small*, vol. 5, no. 2, pp. 170–175, 2009.

[3] 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*. IEEE, 2008, pp. 1125–1131.

[4] 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*. IEEE, 2007, pp. 574–579.

[5] M. Yim, S. Homans, and K. Roufas, "Climbing with snake-like robots," *IFAC Proceedings Volumes*, vol. 34, no. 4, pp. 7–12, 2001.

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

[7] T. Bretl, "Motion planning of multi-limbed robots subject to equilibrium constraints: The free-climbing robot problem," *Int'l. Journal of Robotics Research*, vol. 25, no. 4, pp. 317–342, 2006.

[8] 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)*. IEEE, 2014, pp. 112–119.

[9] 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*. IEEE, 2010, pp. 2828–2833.

[10] 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*. IEEE, 2013, pp. 3547–3554.

[11] 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*. IEEE, 2007, pp. 4582–4587.

[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*. IEEE, 2013, pp. 3293–3298.

[13] K. Walas, "Terrain classification and negotiation with a walking robot," *Journal of Intelligent & Robotic Systems*, vol. 78, no. 3-4, p. 401, 2015.

[14] A. Petrovskaya and O. Khatib, "Global localization of objects via touch," *IEEE Trans. Robotics*, vol. 27, no. 3, pp. 569–585, June 2011.

[15] V. Chu, I. McMahon, L. Riano, C. G. McDonald, Q. He, J. M. Perez-Tejada, M. Arrigo, N. Fitter, J. C. Nappo, T. Darrell *et al.*, "Using robotic exploratory procedures to learn the meaning of haptic adjectives," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3048–3055.

[16] M. Meier, M. Schopfer, R. Haschke, and H. Ritter, "A probabilistic approach to tactile shape reconstruction," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 630–635, June 2011.

[17] D. Xu, G. E. Loeb, and J. A. Fishel, "Tactile identification of objects using bayesian exploration," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3056–3061.

[18] A. M. Okamura and M. R. Cutkosky, "Feature detection for haptic exploration with robotic fingers," *The International Journal of Robotics Research*, vol. 20, no. 12, pp. 925–938, 2001. [Online]. Available: https://doi.org/10.1177/02783640122068191

[19] R. Platt, F. Permenter, and J. Pfeiffer, "Using bayesian filtering to localize flexible materials during manipulation," *IEEE Trans. on Robotics*, vol. 27, no. 3, pp. 586–598, June 2011.

[20] 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*. IEEE, 2014, pp. 3988–3993.

[21] 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*. IEEE, 2011, pp. 2845–2850.

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

[23] 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*. IEEE, 2016, pp. 4925–4930.

[24] 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*, vol. 3, no. 3, pp. 3–35, 1984.

[25] 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.

[26] 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*, Oct 2007, pp. 1557–1563.

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

[28] 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*, 05 2013, pp. 99–106.

[29] 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*. IEEE, 2014, pp. 3982–3987.

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

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

[32] 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*, vol. 35, no. 1-3, pp. 244–264, 2016. [Online]. Available: https://doi.org/10.1177/0278364915594474

[33] 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.

[34] 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*. IEEE, 2012, pp. 3238–3244.

[35] K. Hauser, "Bayesian tactile exploration for compliant docking with uncertain shapes," in *Robotics: Science and Systems*, 2018.

[36] 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*. IEEE, 1992, pp. 612–617.

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

[38] 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*, vol. 3. IEEE, 1999, pp. 1757–1760.

[39] A. Pakman and L. Paninski, "Exact hamiltonian monte carlo for truncated multivariate gaussians," *Journal of Computational and Graphical Statistics*, vol. 23, no. 2, pp. 518–542, 2014.

[40] R. Wein, J. Van Den Berg, and D. Halperin, "Planning high-quality paths and corridors amidst obstacles," *Int'l. Journal of Robotics Research*, vol. 27, no. 11-12, pp. 1213–1231, 2008.

[41] Z. Drezner and G. O. Wesolowsky, "On the computation of the bivariate normal integral," *Journal of Statistical Computation and Simulation*, vol. 35, no. 1-2, pp. 101–107, 1990.

Kris Hauser is Associate Professor at University of Illinois at Urbana-Champaign in the Department of Computer Science and the Department of Electrical and Computer Engineering. He received his PhD in Computer Science from Stanford University in 2008, bachelor's degrees in Computer Science and Mathematics from UC Berkeley in 2003, and was a postdoc at UC Berkeley. He has held faculty positions at Indiana University from 2009–2014 and Duke University from 2014–2019, and began his current position at UIUC in 2019. He is a recipient of a Stanford Graduate Fellowship, Siebel Scholar Fellowship, Best Paper Award at IEEE Humanoids 2015, and an NSF CAREER award.