Convergent Planning

Aaron M. Johnson, Jennifer E. King, and Siddhartha Srinivasa

***Abstract*—We propose a number of “divergence metrics” to quantify the robustness of a trajectory to state uncertainty for under-actuated or under-sensed systems. These metrics are inspired by contraction analysis and we demonstrate their use to guide randomized planners toward more convergent trajectories through three extensions to the kinodynamic RRT. The first strictly thresholds action selection based on these metrics, forcing the planner to find a solution that lies within a contraction region over which all initial conditions converge exponentially to a single trajectory. However, finding such a monotonically contracting plan is not always possible. Thus, we propose a second method that relaxes these strict requirements to find “convergent” (i.e., low-divergence) plans. The third algorithm uses these metrics for postplanning path selection. Two examples test the ability of these metrics to lead the planners to more robust trajectories: a mobile robot climbing a hill and a manipulator rearranging objects on a table.**

***Index Terms*—Motion and Path Planning, Manipulation Planning, Field Robots.**

I. Introduction

CONSIDER a mobile robot, such as in Fig. 1, tasked with traversing a hilly terrain to reach a goal configuration using only local gradient sensors (as in [1]). In red, we see a simple trajectory consisting of a closed loop uphill controller run from multiple initial conditions. While the controller has closed the loop on heading angle relative to the hill, it is not stable in the unobserved global frame. Small disturbances in the initial conditions lead to larger disturbances in the final conditions. In contrast, the blue trajectories (which are more complicated but still follow a gradient direction feedback control scheme) naturally converge over the length of the path reducing any state disturbance. The goal of this work is to quantify the difference between these trajectories and use a motion planner to generate solutions like the blue trajectories.

Robustness to this sort of uncertainty has been considered in *contraction analysis*, [2], which provides a proof of global exponential convergence for a controller over a contraction region (a subset of the configuration space where all states will converge to a single trajectory). However while contraction analysis provides conditions for convergence and methods for choosing controller gains, it does not provide a method of finding such regions. The strongest conditions of the planning framework proposed here extends this work with a method of finding a contracting solution, when one exists. However, not every problem will admit such a region, nor is such a strict convergence result always necessary. We relax these requirements and instead draw inspiration from contraction analysis to bias search based planning methods towards more convergent trajectories.

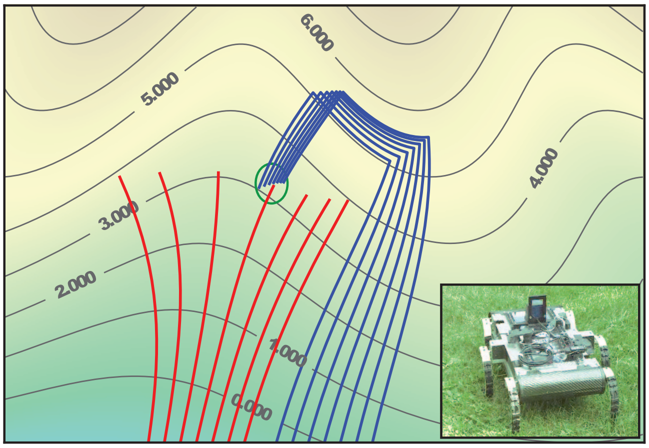

Fig. 1. Example trajectories from a closed-loop controller with different initial conditions for a robot (e.g. [1], *inset*) climbing hilly terrain with only local sensors. The red (shorter) trajectories are diverging by a factor of $E_a = 2.67$. The blue trajectories, while twice as long, are converging by a factor of $E_a = 0.25$. Photo credit: M. Fogelson and D. Koditschek.

As such, this letter may be summarized by the research question: *How can we incorporate and extend contraction analysis into path planning in order to generate trajectories that are more robust to uncertainty?*

We present three *divergence metrics*, partially based on contraction analysis and extending beyond it, as well as motion planning methods designed to find *convergent* (low-divergence) trajectories. These metrics and methods minimize uncertainty and improve the reliability of systems that cannot accurately sense and/or control the full system state.

Specifically, Section II-A introduces metrics that measure the divergence of an action (among them div, the vector field divergence). Section II-B defines numerical approximations to these metrics when a closed form vector field is not available, and shows that in the limit these are equal to the closed-form metrics. These single point metrics are extended to metrics over a path in Section II-C. Then Section III proposes three planning algorithms that incorporate these metrics to either guarantee a contraction region or heuristically bias the solutions towards convergent plans.

We demonstrate the effectiveness of these methods on two problems: a mobile robot climbing a hill (Section IV-A) and a manipulator rearranging an object (Section IV-B). These results show that the planning methods can find strictly contracting paths in some cases, but also that such paths may not always

Manuscript received August 31, 2015; accepted January 15, 2016. Date of publication February 18, 2016; date of current version March 8, 2016. This paper was recommended for publication by Associate Editor S. J. Guy and Editor N. Amato upon evaluation of the reviewers’ comments. This work was supported by the Toyota Motor Corporation.

The authors are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: amj1@andrew.cmu.edu; jeking@andrew.cmu.edu; ss5@andrew.cmu.edu).

Digital Object Identifier 10.1109/LRA.2016.2530864

be possible. In addition, these results show that using a local divergence metric to bias the planning extension step can lead to globally more robust solutions. Finally, we present limited experimental results supporting the claim that these metrics predict success of the actual system.

While these planning methods show promising results, they are simple and based on randomized search. However, the divergence metrics potentially have broad applications for finding robust trajectories with other motion control algorithms. In particular trajectory optimization [3], [4] or optimal anytime planning techniques [5] could use these metrics to find shorter low-divergence trajectories. Here we focus on state uncertainty, but the same concepts could be applied to system uncertainty, e.g. by adding the unknown parameter to the state and searching for convergent plans in the larger space.

A. Related Work

This work is closely related to the control and motion planning literature. From the controls side, we present a method of generating trajectories that in the best cases meet the conditions of [2], i.e. that a contraction region exists over which any state will converge to a single trajectory. This is in some sense a smooth version of the ideas of preimage backchaining [6] or sequential composition [7], wherein each discrete step "prepares" the next over it's entire local domain. While the contraction analysis of [2] provides strong guarantees of convergence to a trajectory, it does not prescribe a method of finding such a trajectory or suggest what to do when a contraction cannot be found.

One method of addressing the more general question of generating plans that are robust to uncertainty is to formulate the planning problem as a partially observable Markov decision process (POMDP) [8]. POMDP solvers can reason about uncertainty and incorporate closed-loop feedback from local or global sensors, but do not easily generalize to continuous spaces and are limited to low-dimensional problems.

An alternative approach is to frame the problem as an instance of probabilistic conformant planning. There, the goal is to find an open-loop trajectory that maximizes the probability of success under uncertainty [6], [9]. These methods are useful for part alignment [10], [11], manipulation in clutter [12] and navigation planning [13]. These ideas are similar to those presented here, but only consider trajectories that are agnostic to sensor feedback.

Our method falls between these two extremes. We aim to select closed-loop plans achievable with execution-time sensor data. Other works have presented similar methods for linear systems and/or Gaussian uncertainty [14]–[16]. We consider problems outside of this domain, such as the problem in Section IV-B, and avoid explicit uncertainty models.

The particle RRT [17] (pRRT) relaxes the linearity assumption, representing uncertainty as a set of state particles. Growth of the search tree on branches of low probability is discouraged by a heuristic method for node selection [18] – the planner presented in Section III-C has some similar properties but removes the reliance on this heuristic. Meanwhile, [19] offers a related RRT-based method for incorporating uncertainty, but explicitly thresholds low probability extensions. Section III-B presents a similar algorithm using our divergence metrics, but we note that this can lead to failures when no paths exist that meet the threshold.

Some approaches track the expected uncertainty and constrain the plan to have an acceptably low chance of failure. These "chance-constrained" methods have been applied in predictive controls [20], [21], receding horizon control [22], and randomized planning [23], [24]. Related predictive controls results bound the uncertainty to a set and show input-to-state stability of the system [25], [26]. Of the planning results, the Safe-RRT [23] and CC-RRT [24] are similar to some of the methods presented here in that they prune possible extension actions based on the uncertainty. These methods are best suited to settings where a running model of the uncertainty is available and the probability of failure may be explicitly checked based on that uncertainty. Here, we do not require any knowledge of the magnitude of uncertainty and instead aim at each time step to reduce the uncertainty, whatever it may be.

Finally, some approaches handle uncertainty in post-processing rather than during planning [16], [27]. In these approaches, a randomized planner generates candidate trajectories and scores them according to some metric in order to select a single trajectory for execution. Our work is complementary to these approaches – we prove some theoretical results for the empirically-motivated numerical approximations in [27], (Section II-B) and also implement a similar post-processing method (Section III-D).

II. Contraction Analysis and Divergence Metrics

In this section we review the main results from contraction analysis [2]. We then define the *divergence metrics*, D_i, partially based on this analysis, as well as corresponding numerical approximations and path metrics. These metrics are key contributions of this letter, and provide a way to quantify the convergence of a path as well as guide the search for a convergent plan.

A. Contraction Analysis

Consider a system with state $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n$, control input $\mathbf{u} \in \mathcal{U} \subseteq \mathbb{R}^m$, and a (possibly time-varying) vector field $\mathbf{f} : \mathcal{X} \times \mathcal{U} \times \mathbb{R} \rightarrow T\mathcal{X}$. Define $\mathbf{F}$ as the symmetric part of the Jacobian of $\mathbf{f}$, i.e.,

$$\mathbf{F}(\mathbf{x}, \mathbf{u}, t) := \frac{1}{2}\left(\frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u}, t)}{\partial \mathbf{x}} + \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u}, t)}{\partial \mathbf{x}}^T\right) \tag{1}$$

The magnitude of a virtual displacement (an infinitesimal displacement at a fixed time t), $\delta\mathbf{x}(t)$, is bounded by the magnitude of the initial displacement, $\delta\mathbf{x}(t_0)$, and the integral of $\lambda_{max}(\mathbf{x}, \mathbf{u}, t)$, the maximum eigenvalue of $\mathbf{F}$ at $\mathbf{x}$ at time t, [2, Eqn. 3],

$$\|\delta\mathbf{x}(t)\| \leq \|\delta\mathbf{x}(t_0)\| e^{\int_{t_0}^{t} \lambda_{max}(\mathbf{x}, \mathbf{u}, \tau) d\tau}. \tag{2}$$

Define the maximal divergence metric $D_m := \lambda_{max}$. In particular if D_m (and therefore also $\mathbf{F}$) is uniformly negative definite everywhere in a region around a nominal trajectory, any differential length at the start of a trajectory will vanish exponentially along its length, [2, Thm. 1], *Theorem 1:* Given a nominal trajectory, $\mathbf{x}_0(t)$, that is the solution to a vector field, $\dot{\mathbf{x}}_0(t) = \mathbf{f}(\mathbf{x}_0(t), \mathbf{u}_0(t), t)$, under control $\mathbf{u}_0(t)$, any other trajectory that begins within region defined by a ball of radius r around the nominal trajectory will converge exponentially to that trajectory so long as $\mathbf{F}$, (1), is uniformly negative definite over that region, i.e. if,

$$\exists \beta > 0, \forall t \geq t_0, \mathbf{x} \in R(t), \quad D_m(\mathbf{x}, \mathbf{u}_0(t), t) \leq -\beta < 0,$$

where $R(t) := \{\mathbf{x} : \|\mathbf{x} - \mathbf{x}_0(t)\| < r\}$. By bounding D_m we conclude that all neighboring trajectories converge to a single trajectory.

Consider now the evolution of a differential volume, δV, around the trajectory,

$$\|\delta V(t)\| = \|\delta V(t_0)\| e^{\int_{t_0}^{t} \operatorname{div} \mathbf{f}(\mathbf{x}, \mathbf{u}, \tau) d\tau}. \tag{3}$$

Define the average divergence metric, $D_a := \operatorname{div} \mathbf{f}$. As a relaxation of Theorem 1, consider [2, Sec. 3.9],

Theorem 2: Given a nominal trajectory, $\mathbf{x}_0(t)$, that is the solution to a vector field, $\dot{\mathbf{x}}_0(t) = \mathbf{f}(\mathbf{x}_0(t), \mathbf{u}_0(t), t)$, under control $\mathbf{u}_0(t)$, any other trajectory that begins within a volume element δV around the nominal trajectory will converge exponentially to a set of measure zero around that trajectory so long as $\operatorname{div} \mathbf{f}$ is uniformly negative definite at every point of the nominal trajectory, i.e. if,

$$\exists \beta > 0, \forall t \geq t_0 \qquad D_a(\mathbf{x}_0(t), \mathbf{u}_0(t), t) \leq -\beta < 0.$$

This theorem says that if the average eigenvalue of $\mathbf{F}$ is negative (since $\operatorname{div} \mathbf{f} = \operatorname{tr} \mathbf{F} = \sum \lambda_{\mathbf{F}}$) then a volume around a given trajectory will collapse on average. There may still be some differential directions which do not collapse down to the nominal trajectory (and, indeed, may diverge), however the differential volume will shrink to zero and the trajectories will lie on some set of measure zero.

Extending beyond the results of contraction analysis, consider the evolution of the expected value of a virtual displacement, $E\left[\|\delta \mathbf{x}(t)\|\right]$, taken over some distribution,

$$E\left[\|\delta \mathbf{x}(t)\|\right] = E\left[\|\delta \mathbf{x}(t_0)\|\right] e^{\int_{t_0}^{t} D_e(\mathbf{x}, \mathbf{u}, \tau) d\tau}, \tag{4}$$

$$D_e(\mathbf{x}, \mathbf{u}, t) := \frac{d}{dt} \ln E\left[\|\delta \mathbf{x}(t)\|\right]. \tag{5}$$

The form of the expected divergence metric, D_e, may not seem particularly useful however we will show in the next section that it is easy to compute numerically.

B. Numerical Approximation

The contraction analysis of [2] assumes a closed form differentiable vector field that may not be available in practice. Instead, to approximate D_m, D_a, and D_e we introduce numerical divergence metrics that approximate the virtual displacement, $\delta \mathbf{x}$, with finite samples.

Given a nominal trajectory, $\mathbf{x}_0(t)$, generated by applying some action $\mathbf{u}_0(t)$ to a system with dynamics $\mathbf{f}$, a perturbed trajectory (or *noisy rollout*), $\mathbf{x}_i(t)$, is the solution to the same system and action as the nominal trajectory, $\dot{\mathbf{x}}_i(t) = \mathbf{f}(\mathbf{x}_i(t), \mathbf{u}_0(t), t)$, but with a different initial condition, $\mathbf{x}_i(t_0) = \mathbf{x}_0(t_0) + \delta \mathbf{x}_i$. Thus (2) may be modified as,

$$\|\mathbf{x}_i(t) - \mathbf{x}_0(t)\| \leq \|\mathbf{x}_i(t_0) - \mathbf{x}_0(t_0)\| e^{\int_{t_0}^{t} D_m(\mathbf{x}_0, \mathbf{u}, \tau) d\tau}, \tag{6}$$

which holds in the limit as $\delta \mathbf{x}_i$ goes to zero. Thus if $D_m < 0$, the ratio, $\|\mathbf{x}_i(t) - \mathbf{x}_0(t)\| / \|\mathbf{x}_i(t_0) - \mathbf{x}_0(t_0)\|$, goes to zero exponentially. To get the closest approximation, consider the largest such ratio, each of which abides by the bound in (6),

$$\max_i \frac{\|\mathbf{x}_i(t) - \mathbf{x}_0(t)\|}{\|\mathbf{x}_i(t_0) - \mathbf{x}_0(t_0)\|} \leq e^{\int_{t_0}^{t} D_m(\mathbf{x}_0, \mathbf{u}_0, \tau) d\tau} \tag{7}$$

For a small time step δt, we have that,

$$\hat{D}_m(\mathbf{x}_0, \mathbf{u}_0, t) := \frac{1}{\delta t} \ln \max_i \frac{\|\mathbf{x}_i(t + \delta t) - \mathbf{x}_0(t + \delta t)\|}{\|\mathbf{x}_i(t) - \mathbf{x}_0(t)\|} \tag{8}$$

and we arrive at the numerical approximation, $\hat{D}_m \approx D_m$.

Similarly, for the average divergence D_a, we will approximate the differential volume by taking the volume spanned by a finite set of points. Let $V(\mathbf{x})$ define such a volume, then $\hat{D}_a \approx D_a$ is a numerical approximation where,

$$\hat{D}_a(\mathbf{x}_0, \mathbf{u}_0, t) := \frac{1}{\delta t} \ln \frac{V(\mathbf{x}(t + \delta t))}{V(\mathbf{x}(t))} \tag{9}$$

Finally, to estimate the divergence of expectation $\hat{D}_e \approx D_e$, consider the ratio of the average displacements,

$$\hat{D}_e(\mathbf{x}_0, \mathbf{u}_0, t) := \frac{1}{\delta t} \ln \frac{\frac{1}{N} \sum_{i=0}^{N} \|\mathbf{x}_i(t + \delta t) - \mathbf{x}_0(t + \delta t)\|}{\frac{1}{N} \sum_{i=0}^{N} \|\mathbf{x}_i(t) - \mathbf{x}_0(t)\|}. \tag{10}$$

The approximation holds exactly in the limit as N goes to infinity and δt goes to zero, as can be derived from (10) or from standard results in Monte Carlo estimation, e.g. [28, Sec. 1.3.1]. Note that the condition in (4) will hold over the length of a path if the set of noisy samples, $\{\delta \mathbf{x}_i\}$, is drawn once at time t_0 and not independently at each time step.

C. Path Metrics

Define for each metric D_i the exponential of the integral of that metric along a trajectory,

$$E_i := e^{\int_{0}^{t} D_i(\mathbf{x}, \mathbf{u}, \tau) d\tau} \tag{11}$$

where note that some of the divergence metrics admit the following simplifications,

$$E_a = \exp\left(\int_{t_0}^{t} \operatorname{div} \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau\right) = \frac{\|\delta V(\mathbf{x}(t))\|}{\|\delta V(\mathbf{x}(t_0))\|}$$

$$\hat{E}_a = \exp\left(\lim_{\delta t \to 0} \sum_{\tau} \ln \frac{V(\mathbf{x}(\tau + \delta t))}{V(\mathbf{x}(\tau))}\right) = \frac{V(\mathbf{x}(t))}{V(\mathbf{x}(t_0))}$$

$$E_e = \exp\left(\int_{t_0}^{t} \frac{d}{d\tau} \ln E\left[\|\delta \mathbf{x}(\tau)\|\right] d\tau\right) = \frac{E\left[\|\delta \mathbf{x}(t)\|\right]}{E\left[\|\delta \mathbf{x}(t_0)\|\right]}$$

$$\hat{E}_e = \exp \lim_{\delta t \to 0} \left(\sum_{\tau = t_0}^{t} \ln \frac{\frac{1}{N} \sum_{i=0}^{N} \|\delta \mathbf{x}_i(\tau + \delta t)\|}{\frac{1}{N} \sum_{i=0}^{N} \|\delta \mathbf{x}_i(\tau)\|}\right)$$

$$= \frac{\frac{1}{N} \sum_{i=0}^{N} \|\delta \mathbf{x}_i(t)\|}{\frac{1}{N} \sum_{i=0}^{N} \|\delta \mathbf{x}_i(t_0)\|}.$$

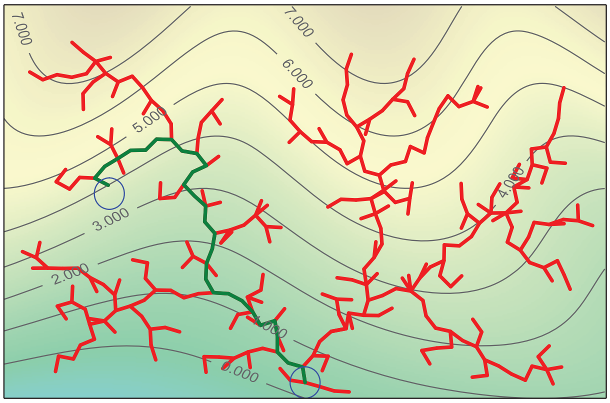

Fig. 2. Topographic map of a hill showing elevation lines, with a KD-RRT overlayed in red and the solution path in green connecting the start and goal.

III. Convergent Planning Methods

This section presents methods of automatically finding paths that respect the dynamics of the system, $\mathbf{f}$, and are robust to state uncertainty. We first define the kinodynamic RRT (KD-RRT) and then propose three extensions that aim to find *convergent* (low-divergence) plans. The CR-RRT sets a threshold on divergence in order to find a monotonically converging trajectory (e.g. to meet the requirements of [2]). The B-RRT uses the divergence as a heuristic bias in order to find more robust trajectories even when a monotonically converging trajectory is impossible. Finally, the AMD-RRT builds off of the other planners in order to find progressively improving solutions if given sufficient time.

A. Kinodynamic RRT (KD-RRT)

Given a known initial state, $\mathbf{x}_s \in \mathcal{X}$, and a goal region $\mathcal{X}_G \subseteq \mathcal{X}$, traditional planning problems search for a trajectory, $(\mathbf{x}, \mathbf{u}) : \mathbb{R} \to \mathcal{X} \times \mathcal{U}$ subject to three constraints,

$$\mathbf{x}(t_0) = \mathbf{x}_s \tag{12}$$

$$\mathbf{x}(t_f) \in \mathcal{X}_G \tag{13}$$

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \text{ for all } t_0 \le t \le t_f. \tag{14}$$

Constraints (12) and (13) ensure the trajectory meets the planning goals, while (14) guarantees a feasible trajectory.

One method of path planning for systems with a constraint like (14) is to use a kinodynamic RRT [29]. At each iteration, sample a random configuration and find the nearest neighbor in the RRT under some cost-to-go metric (typically Euclidean distance). From this configuration generate a candidate action by running a set of controllers. Add a new node to the RRT with the endpoint of the candidate action that is closest to the sampled point, typically under the same metric. The search finishes when a node is within the goal set. Fig. 2 shows an example KD-RRT for the hill climbing problem (Sec. IV-A).

B. Contraction Region RRT (CR-RRT)

To find a motion plan that meets the requirements of Theorem 1 (or Theorem 2), we must modify the extension step of the KD-RRT to only consider actions such that,

$$D_m(\mathbf{x}(t), \mathbf{u}(t), t) < 0 \text{ for all } t_0 \le t \le t_f, \tag{15}$$

(respectively, $D_a < 0$). If no such actions are sampled, the tree is not extended. If one or more such actions are sampled, then add a node in the same way as the KD-RRT extend step.

Not every problem will have a solution that meets the requirement of Theorem 1, and in such cases this algorithm will never terminate with a solution. We can relax the requirement by altering the constraint,

$$D_m(\mathbf{x}(t), \mathbf{u}(t), t) < d_m \text{ for all } t_0 \le t \le t_f, \tag{16}$$

where $d_m \in \mathbb{R}$ is a parameter that corresponds to the maximum admissible divergence value.

C. Biased RRT (B-RRT)

A less strict method of incorporating the divergence metrics is to include them as a cost in an optimization,

$$\mathbf{x}^*, \mathbf{u}^* = \underset{\mathbf{x},\mathbf{u}}{\operatorname{argmin}}\, E_i(\mathbf{x}, \mathbf{u}) \text{ s.t.(12)}-(14). \tag{17}$$

Solving this optimization exactly is difficult, but we can approximate the optimization by incorporating one of the metrics D_i into the selection criteria at each extension. Rather than selecting the best action based on a Euclidean distance metric, this biased RRT (B-RRT) algorithm scales the original distance by a factor of $s = e^{bD_i}$. Here $b \in \mathbb{R}$ is a *bias* and D_i is the chosen divergence metric. With this, actions that perform well with respect to the divergence metric are preferred even if they are not the most direct path. Thus, the B-RRT heuristically tries to reduce the divergence, without enforcing the strict conditions of the CR-RRT. Note that when $b = 0$ this algorithm is identical to the KD-RRT.

D. Anytime Minimal Divergence RRT (AMD-RRT)

The third extension to the KD-RRT takes advantage of the fact that with a randomized planner, identical calls for the same query will generate different solutions. Here we use multiple calls to either the KD-RRT or the B-RRT to generate a set of candidate paths, and then we select the best path with respect to a chosen path metric, E_i. This process can use a fixed number of trials or a termination condition such as a threshold on the metric. Alternatively, it can be an anytime algorithm and simply return the best trajectory found when stopped. This AMD-RRT implicitly assumes that the trajectory generator is capable of finding good candidates with non-zero probability, and if so, with enough time, the AMD-RRT will return one such good trajectory.

IV. Experiments

We demonstrate the algorithms from Section III in two scenarios: a mobile robot traversing hilly terrain, and a manipulator rearranging objects on a table.

A. Hill Climbing Example

1) Problem Specification: Suppose you have a mobile robot that is navigating hilly terrain, such as in the contour plot

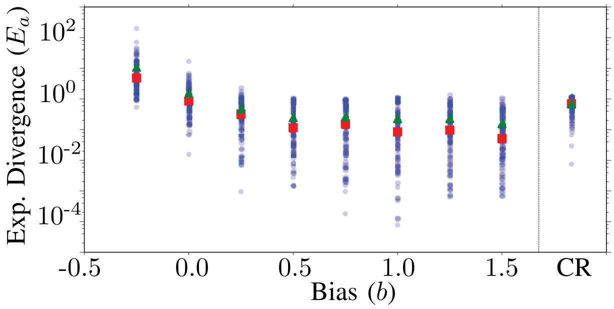

Fig. 3. Average exponential divergence, E_a, for the hill climbing problem, calculated from 100 trials each at different values of the bias terms, b, and for the CR-RRT. The mean (green triangle) and median (red square) are also shown for each b.

TABLE I

Hill Climbing Planning Results for the KD-RRT (*top*), B-RRT (*middle*), and CR-RRT(*bottom*), Including Success Rate (S), Exponential Divergence (E_a and $\hat{E}_a$), and Planning Time. Mean ± One Standard Deviation Listed. Non-Gray Results are Statistically Different From the Baseline With $p < 0.005$.

Test	S	E_a	$\hat{E}_a$	Planning Time
KD	100	1.48 (±2.03)	1.72 (±3.24)	31.2 (±28.0)
$b =$				
-0.25	100	10.86 (±21.72)	74.53 (±399.08)	32.7 (±23.5)
0.25	100	0.47 (±0.43)	0.47 (±0.43)	39.2 (±32.1)
0.50	100	0.25 (±0.29)	0.25 (±0.29)	48.2 (±52.2)
0.75	100	0.26 (±0.29)	0.26 (±0.29)	59.7 (±69.3)
1.00	99	0.23 (±0.31)	0.23 (±0.31)	67.1 (±90.0)
1.25	97	0.23 (±0.28)	0.24 (±0.28)	72.3 (±108.4)
1.50	98	0.16 (±0.22)	0.16 (±0.22)	72.3 (±96.7)
CR	30	0.56 (±0.27)	0.57 (±0.27)	247.1 (±146.0)

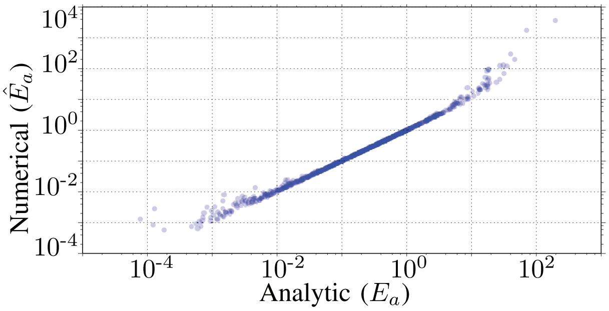

Fig. 4. Correlation between E_a and $\hat{E}_a$ for the hill climbing B-RRT.

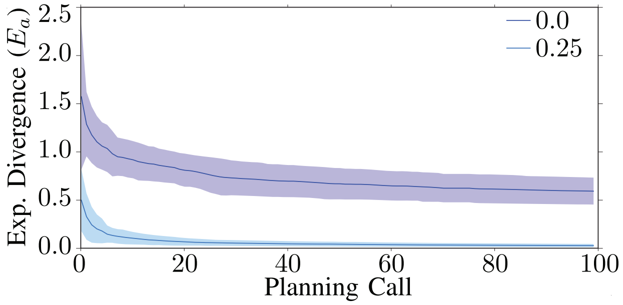

Fig. 5. Minimum divergence found by a given planning call for the AMD-RRT based on the KD-RRT (top) and the B-RRT (bottom). Mean and ± one standard deviation shown over 100 trials (each consisting of 100 calls).

in Fig. 1 or Fig. 2. We model the hill as a height function, $z = h(x, y) := 3y + \sin(x + xy)$ for $(x, y) \in [-2, 2] \times [0, 2.5]$. The robot uses a controller that allows it to follow a constant forward velocity at any arbitrary angle θ relative to the hill gradient ∇h. Thus the closed loop vector field of the system is $\mathbf{f}(x, y, \theta) = \mathbf{R}(\theta)\nabla h(x, y)/\|\nabla h(x, y)\|$, where $\mathbf{R}(\theta)$ is a rotation matrix. Note that such a controller uses only local sensors, as done in [1] for the uphill direction. For a two dimensional problem like this with fixed forward velocity, one eigenvalue of the vector field Jacobian will be zero and therefore either $\lambda_{max} = 0$ or $\lambda_{max} = \text{div} f = \text{tr} F = \lambda_1 + \lambda_2$. As such we will consider D_a and related metrics, since either $D_m = D_a$ or $D_m = 0$.

The task is to find a trajectory from a start location to a goal location using the hill-relative controller. Fig. 2 shows one example trajectory and final tree generated with the KD-RRT. In this section, the planner extension samples 8 fixed-length actions (i.e. it samples a value of θ and applies the hill-relative controller for a fixed amount of time). A planning call is successful if it returns a path to the goal after adding fewer than 10,000 nodes to the tree.

2) B-RRT: We use the B-RRT to test whether using a local metric as a heuristic bias results in solutions that are globally more robust to uncertainty. The overall integral divergence, E_a, scores the solution generated by the planner. To see the effect of the bias term, we ran 100 trials at different values of b with randomized start and goal location.

The results of Fig. 3 and Table I show that increasing the bias term does indeed lead to a lower exponential integral divergence (E_a) on average (all are statistically different from $b = 0.0$ with $p < 0.0001$). This added bias does slightly increase the planning time – see Table I. It appears that $b = 0.5$ achieves most of the improvement for this problem.

3) Numerical Approximations: To confirm the accuracy of the numerical approximations given in Section II-B, we compare E_a, the integral of the analytic divergence, with $\hat{E}_a$, the numerically computed ratio of initial volume to final volume of N perturbed initial conditions, over all of the trials from the previous section. Fig. 4 shows that these metrics are highly correlated, even with a relatively small $N = 4$. The correlation (in log-log) is linear with $R^2 = 0.98$, with most of the deviations coming from very large and very small values of E_a. We suspect that improved numerical methods and smaller displacements would reduce the remaining differences.

4) AMD-RRT: We test the AMD-RRT on both the KD-RRT ($b = 0$) and the B-RRT ($b = 0.25$) with fixed start and goal locations. Fig. 5 reports the best-so-far from up to 100 calls, showing the mean and standard deviation over 100 trials of each AMD-RRT conditions (and therefore represents a total of 100×100 calls to each underlying planner).

First, consider the problem of finding a trajectory with $E_a < 1$ using only the KD-RRT, as done in [27] (for a different objective function). While only about 11% $(1053/10000)$ of individual KD-RRT runs results in a candidate trajectory with $E_a < 1$, if we allow the AMD-RRT to pick from 7 candidates more than half $(51\%, 51/100)$ of the AMD-RRT trials succeed

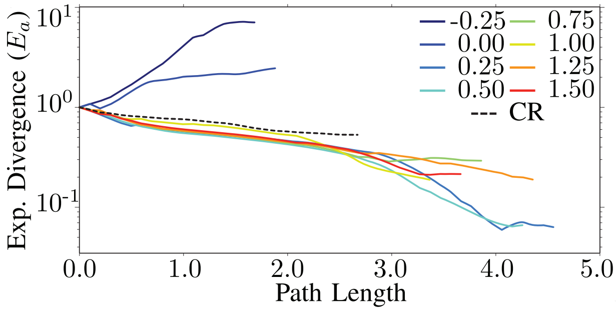

Fig. 6. Exponential divergence, E_a, over the length of sample solution paths for the B-RRT with different bias terms, b, as well as one solution from a CR-RRT, the only example that is monotonically decreasing in E_a.

with $E_a < 1$. If we allow 32 runs, then all 100 trials succeeded. Thus the AMD-RRT can find low divergence paths without modifying the motion planner at all.

Biasing the search by using a B-RRT can find a good trajectory much faster, with 92% $(9194/10000)$ of individual runs succeeding with $E_a < 1$. In this case all 100 trials of the AMD-RRT succeeded by the 3rd run of the B-RRT. These results show that even though the unbiased search (KD-RRT) can find low-divergence paths, using a local bias (B-RRT) will find better paths faster.

5) CR-RRT: The above methods were able to find trajectories that had low divergence, however they did not guarantee monotonic contraction as required for the strongest of the theoretical results in Section II-A. For example, the solid lines of Fig. 6 show the evolution of E_a over the length of the trajectory for eight different B-RRT solutions – note that none are monotonically decreasing (i.e. they all have at least one point with $D_a > 0$). One way to generate trajectories that meet the requirements of, e.g., Theorem 2 is to use the CR-RRT. As shown in Table I, a successful contraction region (monotonically contracting) path was found $S = 30\%$ of the time (30/100). The dashed line in Fig. 6 shows an example of a successful monotonically decreasing trajectory.

B. Rearrangement Planning

1) Problem Specification: Next we consider a rearrangement planning problem [12], [30]. We task a manipulator to push an object to a goal region, as seen in Fig. 7. The robot is not endowed with sensors that can detect the object's state. However, prior work has shown that some pushing actions are inherently uncertainty reducing [31], and we would like to guide our planners to select such actions.

The vector field, $\mathbf{f}$, of the system describes the motion of the manipulator and the object and is defined by the physics of the contact between the manipulator and the object. This vector field is not smooth – contact is inherently discontinuous – and lacks an analytic representation (although for simple problems this is theoretically possible [32]). Prior work has shown that $\mathbf{f}$ can be effectively approximated by a physics simulator [33], [34], however analytic divergence measures are unavailable.

The state space is the joint configuration of the manipulator and the pushed object, each in SE(2). Computing the volume of a set of points in this high-dimensional space is challenging, and so we will use only $\hat{D}_m$ and $\hat{D}_e$. For a potential motion, $\mathbf{u}_0(t)$, applied to a state, $\mathbf{x}_0(t)$, these metrics are calculated by rolling out a set of N noisy samples using the physics simulator. The samples are drawn from a Gaussian distribution around $\mathbf{x}_0(t)$ with a standard deviation of 4 cm for both the object and the manipulator pose.

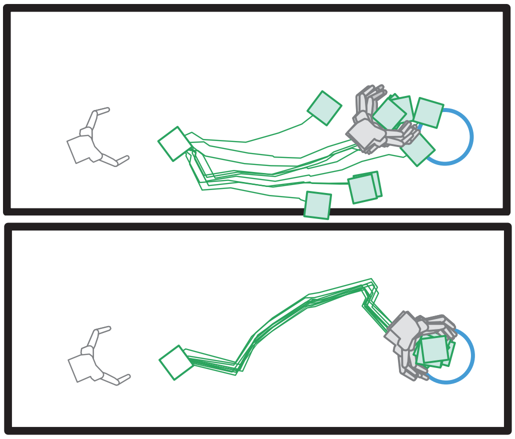

Fig. 7. Two pushing examples. *Top:* Exponential divergence $E_e = 11.14$, $N_G = 5\%$ of rollouts reaching the goal. *Bottom:* $E_e = 1.84$, $N_G = 100\%$.

We plan for a hand pushing a 5 cm × 5 cm box in the plane. The planner samples 8 actions and uses separate threads to evaluate each in parallel. An action is a linear and angular velocity of the hand and a duration. Each component of a sampled action is drawn uniformly from a bounded interval. A planning call is successful if it returns a path within 480 seconds that moves the box to a 10 cm radius goal region. Plans are evaluated using $\hat{E}_e$ calculated using $N = 100$ and sampling the initial object and manipulator poses from a Gaussian distribution with standard deviation of 1cm.

2) B-RRT: For this rearrangement planning problem, we implemented the B-RRT using $\hat{D}_e$ as the divergence metric, i.e. the numerical approximation to the expected value divergence. As with the hill climbing problem, we ran 100 trials of the B-RRT for different values of the bias b and the results are in Table II and Fig. 8, including success rate (S) from 100 trials, exponential divergence $\hat{E}_e$, percent of trials better than the KD-RRT (K), percent of rollouts that reach the goal (N_G), and planning time. These trials used $N = 4$ samples at each extension step to calculate $\hat{D}_e$, which appears to be a very small number of samples for this high dimensional problem, but is sufficient to bias the results towards lower values of $\hat{E}_e$. Results from an additional set of trials with $N = 10$ were not statistically different from those of Table II, with the exception of the planning time which was, unsurprisingly, about twice as long.

Overall the B-RRT was able to find more reliable trajectories, with lower values of $\hat{E}_e$ on average. As with the results from Section IV-A2, $b = 0.5$ achieves most of the improvement in $\hat{E}_e$. In addition for each condition we list the percent of the noisy samples used to calculate $\hat{E}_e$ that reach the goal region, N_G, which [27] anecdotally shows correlates with the success of actual experiments. Similar to the $\hat{E}_e$ metric, N_G is higher for the B-RRT, but further increasing the value of b had little effect. These improvements do come at the cost of higher planning times.

TABLE II

REARRANGEMENT PLANNING RESULTS FOR THE KD-RRT (*top*), B-RRT (*middle*), AND CR-RRT(*bottom*), INCLUDING SUCCESS RATE (S), EXPONENTIAL DIVERGENCE ($\hat{E}_e$), % OF RUNS BETTER THAN KD (K), NUMBER OF SAMPLES THAT REACH THE GOAL (N_G), AND PLANNING TIME. MEAN ± ONE STANDARD DEVIATION LISTED. ALL RESULTS STATISTICALLY DIFFERENT FROM THE BASELINE WITH $p < 0.005$.

Test	S	$\hat{E}_e$	K	N_G	Planning Time
KD	100	11.39 (±8.45)	-	44.2 (±33.0)	4.8 (±8.3)
$b =$					
0.25	100	8.33 (±6.07)	4.00	59.0 (±31.9)	31.3 (±42.7)
0.50	97	8.06 (±7.81)	7.22	58.8 (±29.2)	55.4 (±83.3)
0.75	97	7.63 (±7.40)	14.43	63.3 (±29.2)	62.3 (±88.9)
1.00	97	7.38 (±6.60)	10.31	64.2 (±26.3)	51.0 (±67.9)
1.25	95	8.07 (±7.25)	9.47	59.9 (±27.5)	65.1 (±83.0)
1.50	100	8.27 (±7.69)	11.00	58.6 (±29.7)	78.7 (±106.9)
1.75	99	6.87 (±6.50)	11.11	68.7 (±25.5)	54.8 (±78.3)
2.00	99	6.79 (±7.39)	12.12	64.6 (±26.2)	62.6 (±89.3)
$d_m =$					
2.00	94	6.95 (±6.05)	10.64	63.0 (±27.7)	85.8 (±106.8)
1.50	78	3.39 (±2.64)	20.51	69.5 (±23.0)	116.9 (±93.6)
1.00	70	4.24 (±4.07)	17.14	70.0 (±21.5)	129.2 (±113.9)
0.75	66	4.38 (±3.38)	15.15	74.9 (±19.9)	159.0 (±119.1)
0.50	21	3.06 (±1.89)	28.57	66.4 (±27.5)	303.7 (±105.8)
0.25	0	-	-	-	-
0.00	0	-	-	-	-

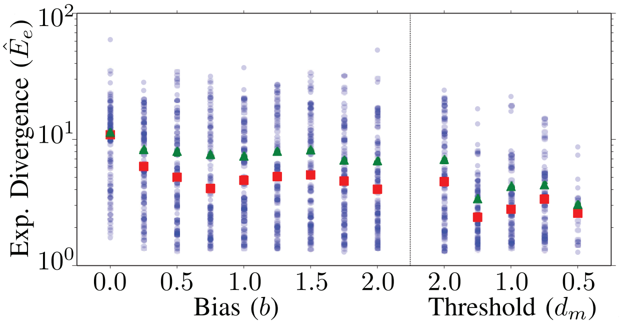

Fig. 8. Expected exponential divergence, $\hat{E}_e$, for the rearrangement problem under the test conditions from Table II. The mean (green triangle) and median (red square) are also shown for each b.

3) CR-RRT: In this more complex problem with non-smooth dynamics, finding a solution trajectory that lies in a contraction region is quite difficult. To test this, we used the CR-RRT with $\hat{D}_m$ computed with $N = 10$ samples. After 100 trials with a timeout of 480s each, we found no successful contracting solutions (with $\hat{D}_m < 0$ everywhere). To relax this requirement, we consider higher thresholds for $\hat{D}_m < d_m$; these results are also in Table II and Fig. 8.

The results from these trials have lower $\hat{E}_e$ on average, and are more likely to reach the goal region (N_G). However this higher performance does come at the cost of a lower planning success rate S (i.e. more trials reached the timeout of 480s), and the successful trials took much longer.

4) AMD-RRT: Individual planning calls from the B-RRT and CR-RRT provided better results than the KD-RRT, but took much longer to do so (due to the time needed to compute $\hat{D}_e$ or $\hat{D}_m$). To level the playing field, consider applying the AMD-RRT to find the best $\hat{E}_e$ possible in a given amount of time. Running a single trial of this problem takes much longer than the hill climbing problem, and so evaluating the AMD-RRT with 20000 total trials is infeasible. Instead, we estimate the AMD-RRT performance based only on Table II.

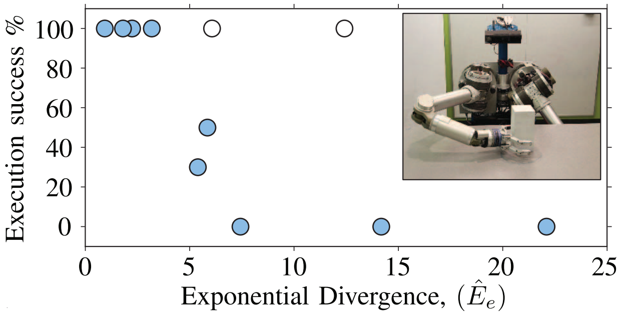

Fig. 9. Measured success rate over 10 executions on HERB [35] (*inset*) of trajectories as a function of exponential divergence, $\hat{E}_e$. Most trials (shaded) demonstrate a negative correlation between success rate and exponential divergence. Outliers (open) are likely due to errors in the physics model.

For each condition we list the percent, K, of the successful trials that finished with an $\hat{E}_e$ less than the best $\hat{E}_e$ over all 100 KD-RRT trials. These trials of the KD-RRT took a total of 1542s to plan and evaluate $\hat{E}_e$, and in that time found a $\min \hat{E}_e = 1.66$. Compare that to the $b = 0.25$ for example, which took 4193s but found a better $\hat{E}_e$ in 4% of trials (K), for an average of once every 1048s. Therefore we would expect to find a better $\hat{E}_e$ with the B-RRT in less time than all 100 trials of the KD-RRT would take. This result is encouraging although not as conclusive as the result from the hill climbing example and more experiments are needed.

5) Real Robot Experiments: In this section, we demonstrate that trajectories with lower $\hat{E}_e$ succeed more often in execution than trajectories with high $\hat{E}_e$ values. We use the KD-RRT to find solutions to the rearrangement planning problem, however here we plan for the full 7-DOF right arm of the HERB robot [35] rather than only a hand. We use AprilTags [36] to detect the pose of the box. Plans are evaluated using $\hat{E}_e$ calculated with $N = 100$ and sampling the object poses from a Gaussian distribution with standard deviation of 2 cm (computed from AprilTag measurements).

Figure 9 shows the results of 11 plans with varying $\hat{E}_e$ each executed 10 times on HERB. An execution is a success if the final pose of the box is within 15 cm of the planned final pose. The trials shaded in blue demonstrate a negative correlation between success rate and $\hat{E}_e$, i.e. trajectories with lower $\hat{E}_e$, like those generated by the convergent planners, are more likely to succeed. Two trials, not shaded, do not follow the expected trend. A closer look at these paths attributes the discrepancy to unrealistic behavior in the physics simulator used to calculate $\hat{E}_e$. As the focus of this letter is not the realism of the physical model, we have left these outliers.

V. CONCLUSION

The problem of uncertainty is pervasive in robotics, and must be carefully considered in order to have reliable systems. The most common way to reduce uncertainty is to use a closed-loop feedback controller to measure and correct errors that may accumulate. However there are many settings where a closed-loop trajectory is not enough, as there is insufficient sensor information or control authority to correct all of the uncertainty. We propose new convergent path planning methods that can search for closed-loop trajectories that are inherently robust to state uncertainty despite sensor or actuation limitations. We introduce analytic and numerical divergence metrics that the convergent planners seek to minimize. Using the strongest of these planners and metrics, we show the first planning based method to find contraction regions where all states converge to a single trajectory.

The convergent planning methods presented here are relatively simple, but there are many ways to use the divergence metrics to guide planning. In the future we plan to apply these metrics to trajectory optimization techniques, e.g. [3], [4], that can locally search for the lowest-divergence path. The examples in this letter were all quasi-static but these ideas can also be applied to dynamic problems [34]. Furthermore, the divergence metrics provide a nice compliment to the ideas of geometric mechanics, e.g. [37], which aims to extract feasible trajectories from the geometry of the system's vector field.

The divergence metrics are fundamental properties of the underlying vector field, and motion planning will be most effective when it considers these properties. Convergent motion planners, like those presented here, provide a new way to generate behaviors that are robust to the uncertainty that is always present when running robots in the real world.

References

[1] A. M. Johnson, M. T. Hale, G. C. Haynes, and D. E. Koditschek, "Autonomous legged hill and stairwell ascent," in *Proc. IEEE Int. Workshop Saf. Secur. Rescue Robot.*, Nov. 2011, pp. 134–142.

[2] W. Lohmiller and J.-J. E. Slotine, "On contraction analysis for non-linear systems," *Automatica*, vol. 34, no. 6, pp. 683–696, 1998.

[3] M. Zucker *et al.*, "CHOMP: Covariant hamiltonian optimization for motion planning," *Int. J. Robot. Res.*, vol. 32, nos. 9–10, pp. 1164–1193, 2013.

[4] M. Kalakrishnan *et al.*, "STOMP: Stochastic trajectory optimization for motion planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 4569–4574.

[5] S. Karaman *et al.*, "Anytime motion planning using the RRT*," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 1478–1483.

[6] T. Lozano-Perez, M. T. Mason, and R. H. Taylor, "Automatic synthesis of fine-motion strategies for robots," *Int. J. Robot. Res.*, vol. 3, no. 1, pp. 3–24, 1984.

[7] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek, "Sequential composition of dynamically dexterous robot behaviors," *Int. J. Robot. Res.*, vol. 18, no. 6, pp. 534–555, 1999.

[8] L. Kaelbling, M. Littman, and A. Cassandra, "Planning and acting in partially observable stochastic domains," *Artif. Intell.*, vol. 101, nos. 1–2, pp. 99–134, 1998.

[9] N. Hyafil and F. Bacchus, "Conformant probabilistic planning via CSPs," in *Proc. AAAI Int. Conf. Autom. Plann. Schedul.*, 2003, pp. 205–214.

[10] M. Brokowski, M. Peshkin, and K. Goldberg, "Curved fences for part alignment," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1993, pp. 467–473.

[11] M. Erdmann and M. Mason, "An exploration of sensorless manipulation," *IEEE J. Robot. Autom.*, vol. RA-4, no. 4, pp. 369–379, Aug. 1988.

[12] M. Dogar and S. Srinivasa, "A planning framework for non-prehensile manipulation under clutter and uncertainty," *Auton. Robots*, vol. 33, no. 3, pp. 217–236, 2012.

[13] M. Levihn, J. Scholz, and M. Stilman, "Planning with movable obstacles in continuous environments with uncertain dynamics," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 3832–3838.

[14] S. Prentice and N. Roy, "The belief roadmap, efficient planning in linear POMDPs by factoring the covariance," in *Proc. Int. Symp. Robot. Res.*, vol. 66, 2008, pp. 293–305.

[15] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "LQR-Trees: Feedback motion planning via sums-of-squares verification," *Int. J. Robot. Res.*, vol. 29, no. 8, pp. 1038–1052, 2010.

[16] J. Van Den Berg, P. Abbeel, and K. Goldberg, "LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 895–913, 2011.

[17] N. A. Melchior and R. Simmons, "Particle RRT for path planning with uncertainty," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 1617–1624.

[18] C. Urmson and R. Simmons, "Approaches for heuristically biasing RRT growth," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2003, pp. 1178–1183.

[19] G. Kewlani, G. Ishigami, and K. Iagnemma, "Stochastic mobility-based path planning in uncertain environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 1183–1189.

[20] A. T. Schwarm and M. Nikolaou, "Chance-constrained model predictive control," *AIChE J.*, vol. 45, no. 8, pp. 1743–1752, 1999.

[21] L. Blackmore, "A probabilistic particle control approach to optimal, robust predictive control," in *Proc. AIAA Guidance Navig. Control Conf.*, 2006, pp. 2006–6240.

[22] N. Du Toit and J. Burdick, "Robot motion planning in dynamic, uncertain environments," *IEEE Trans. Robot. Autom.*, vol. 28, no. 1, pp. 101–115, Feb. 2012.

[23] R. Pepy and A. Lambert, "Safe path planning in an uncertain-configuration space using RRT," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 5376–5381.

[24] B. Luders, M. Kothari, and J. P. How, "Chance constrained RRT for probabilistic robustness to environmental uncertainty," in *Proc. AIAA Guidance Navig. Control Conf.*, 2010, pp. 2010–8160.

[25] D. L. Marruedo, T. Alamo, and E. Camacho, "Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties," in *Proc. IEEE Conf. Decis. Control*, 2002, vol. 4, pp. 4619–4624.

[26] Z.-P. Jiang and Y. Wang, "Input-to-state stability for discrete-time nonlinear systems," *Automatica*, vol. 37, no. 6, pp. 857–869, 2001.

[27] M. C. Koval, J. E. King, N. S. Pollard, and S. S. Srinivasa, "Robust trajectory selection for rearrangement planning as a multi-armed bandit problem," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 2678–2685.

[28] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. New York, NY, USA: Springer, 2001.

[29] S. LaValle and J. Kuffner, "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, 2001.

[30] M. Stilman, J. Schamburek, J. Kuffner, and T. Asfour, "Manipulation planning among movable obstacles," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 3327–3332.

[31] M. Dogar and S. Srinivasa, "Push-grasping with dexterous hands: Mechanics and a method," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 2123–2130.

[32] A. M. Johnson, S. E. Burden, and D. E. Koditschek, "A hybrid systems model for simple manipulation and self-manipulation systems," *arXiv preprint arXiv:1502.01538 [cs.RO]*, 2015.

[33] J. King, J. Haustein, S. Srinivasa, and T. Asfour, "Nonprehensile whole arm rearrangement planning on physics manifolds," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2015, pp. 2508–2515.

[34] J. Haustein, J. King, S. Srinivasa, and T. Asfour, "Kinodynamic randomized rearrangement planning via dynamic transitions between statically stable states," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2015, pp. 3075–3082.

[35] S. S. Srinivasa *et al.*, "HERB 2.0: Lessons learned from developing a mobile manipulator for the home," *Proc. IEEE*, vol. 100, no. 8, pp. 2410–2428, Aug. 2012.

[36] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 3400–3407.

[37] R. L. Hatton and H. Choset, "Geometric motion planning: The local connection, Stokes' theorem, and the importance of coordinate choice," *Int. J. Robot. Res.*, vol. 30, no. 8, pp. 988–1014, 2011.