Optimally Convergent Trajectories for Navigation

Nathan J. Kong and Aaron M. Johnson

Carnegie Mellon University, Pittsburgh PA 15289, USA,
njkong@andrew.cmu.edu

Abstract. This paper investigates optimization-based planning methods for generating trajectories which are robust to state uncertainty in undersensed and underactuated systems. Specifically, these methods are applied to an undersensed robotic hill climbing system. In previous work, divergence metrics based on contraction analysis were used to quantify robustness of a trajectory to state uncertainty in conjunction with a kinodynamic RRT planner to guide the planner towards more convergent directions. Resulting trajectories were sub-optimal or needed to be smoothed prior to implementation. This work proposes an optimization framework to plan optimally robust and smooth trajectories which can also be readily implemented on the robotic hill climbing problem. A new hill climbing controller is also presented which can guarantee for the first time the strongest result of contraction analysis, global asymptotic convergence, where possible. Trajectories created using the new trajectory optimization framework and hill controller are shown to be smoother and more robust than previous methods as well as an asymptotically optimal versions of previous methods.

Keywords: Path Planning, Field Robotics, Optimization

1 Introduction

Consider a mobile robot on hilly terrain that has precise heading control and an accurate map. The robot is tasked to traverse from its current state to a goal region, as in [5, 6]. However, the robot may not be well localized, and has only a rough understanding of where it is currently located on the hill, e.g. in GPS-denied settings. If a robot path planner only considers the shortest path to the goal region, the resulting trajectory may diverge due to continuous growth of uncertainty, as in the left side of Fig. 1. However, in some regions the shortest path will lead to convergent behaviors, as in the right side of Fig. 1, where the uncertainty shrinks over time.

In many robotic applications, state uncertainty growth is dealt with by using closed-loop feedback. A controller can be used to sense and reduce errors in the state. However, robotic systems are often undersensed or underactuated and cannot close the loop on certain state uncertainties. In these cases they must rely on planning robust behaviors through leveraging the underlying geometry of their dynamics. This issue is especially prevalent for the hill climbing robot

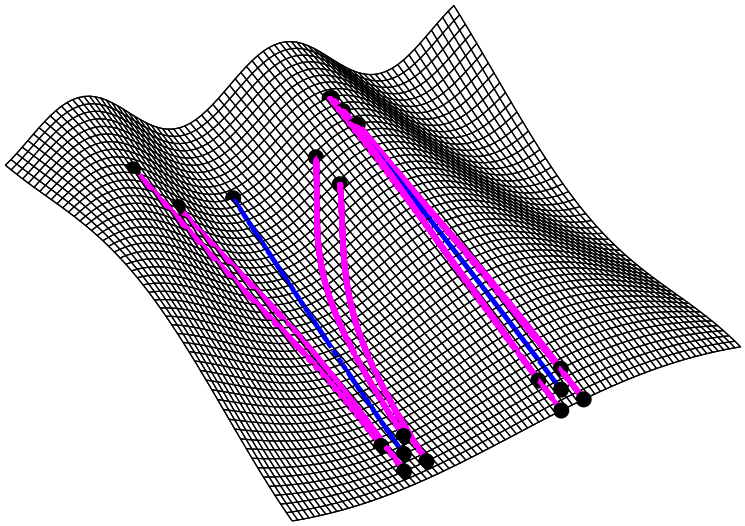

Fig. 1. Diverging (left) versus converging (right) trajectories. The nominal trajectory is shown in (blue), and 4 points of state uncertainty (black) are simulated forwards (magenta).

example in [6] and planning for robust manipulator pushing motions [11], but arises in many more general settings.

Contraction analysis [15] considers robustness to state uncertainty through geometric analysis, and provides a proof for global asymptotic convergence for dynamical systems with a vector field with a strictly negative Jacobian, a contraction region. This states that if all the eigenvalues of the Jacobian are strictly negative in a region, all trajectories converge to a single trajectory and the resulting behavior is robust to state uncertainty because there is global asymptotic convergence.

To plan for robust trajectories, *convergent planning* [7] utilizes a kinodynamic rapidly-exploring random tree (RRT) [12] which is biased towards behaviors that are on average convergent. Average convergence is defined in [7] to be trajectories where the average eigenvalue of the vector field's Jacobian to be negative instead of the maximum eigenvalue being negative in the case of strict contraction. This corresponds to contraction down to a set of zero volume, instead of a single point [15]. In [7] these trajectories are created for both robotic hill climbing and manipulator pushing motions. Another RRT method which may be modified to include directions of divergence to reduce state uncertainty is the Vector Field RRT (VF-RRT) [10] algorithm which uses the vector field of the system to minimize upstream cost. However, since these trajectories are created through an RRT, they are not optimal. Asymptotically near optimal trajectories are created in [14] by modifying the average divergence metric used in [7] to have optimality guarantees, and more optimal RRT methods include multiple restarts RRT (C-MRRT) and stable sparse RRT (C-SSRT) [13].

One issue with trajectories produced by RRT is that they are often not smooth. In [14] it is noted that the near optimal trajectories produced by C-MRRT and C-SRRT can be post-processed to feasibly work on a robot, because optimal and smooth RRT methods such as RRT* [9] are not suitable in these contexts. This is because a steering function may not always be available for kinodynamic planning, which is required for RRT* to rewire the tree to improve solution quality [14]. However, if the near optimal trajectories are post-processed, they will no longer be optimal and may no longer satisfy contraction constraints. In this paper, RRT* is implemented and compared against previous methods because there exists a steering function for this specific problem. Although these trajectories are asymptotically optimal and more directly usable than RRT methods, optimality is only guaranteed as time approaches infinity.

This paper presents a new trajectory optimization (T-OPT) framework based on [3] that can produce optimally convergent and smooth trajectories which can readily be implemented on a robot and converge to an optimal solution in finite time. The framework uses analytical calculations of divergence and also includes options to create trajectories which guarantee asymptotic convergence or average convergence. In addition to independently creating optimal trajectories which can satisfy constraints, the proposed T-OPT framework can post-process trajectories generated using RRT methods while maintaining optimality and specified constraints.

In the context of the hill climbing example, both [7] and [14] do not consider the case of strict contraction, and only considers an estimate of average convergence. In fact, the hill climbing controller used can never guarantee strict convergence and at best only guarantee average convergence to a set of zero volume. To resolve this issue, a new hill climbing controller that varies speed as a function of local hill steepness is proposed in this paper. This change allows the guarantee of strict convergence, and ultimately increases the robustness of the controller to state uncertainty.

1.1 Related Work

This work focuses on planning trajectories where state uncertainty is expected. One popular method of planning under state uncertainty is to formulate the planning problem as a partially observable Markov decision process (POMDP) [8]. However, POMDP solvers do not perform well when the action space is continuous and do not scale well in higher dimensions.

Probabilistic conformant planning is another method to plan for state uncertainty. The goal of the planner is to maximize the likelihood of success given an expectation of state uncertainty [4,16]. Although conformant planning is useful to plan robust trajectories in some contexts, it requires an accurate model of state uncertainty and doesn't utilize sensor feedback to increase robustness. The methods proposed here do not require a model of the state uncertainty, and instead try to continually reduce whatever uncertainty there is.

The hill climbing problem used in this paper is also used in [5–7,14], where a robot is tasked to navigate from a start position to a goal region using the gra-

dient of the hill as feedback. The hill gradient controller in these examples were inspired by [1]. This paper analyzes contraction properties of this controller and proposes modifications to create a contraction region, satisfying the strongest results from contraction analysis [15].

2 Methods

2.1 Contraction Analysis and Divergence Metrics

Converging behaviors of a dynamical system can be analyzed through contraction analysis [15]. Consider a dynamical system with state $x \in \mathcal{X} \subseteq \mathbb{R}^n$, control input $u \in \mathcal{U} \subseteq \mathbb{R}^m$, and a continuously differentiable vector field $f(x, u, t)$ such that $f : \mathcal{X} \times \mathcal{U} \times \mathbb{R} \to T\mathcal{X}$. Define F as the symmetric part of the Jacobian of f [15, Definition 1],

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

Now consider two neighboring trajectories in the vector field f. Define δx to be the virtual displacement between the neighboring trajectories, and the squared distance to be $\delta x^T \delta x$. The rate of change of $\delta x^T \delta x$ is bounded by the max eigenvalue $\lambda_{max}(x, u, t)$ of the symmetric part of the Jacobian F,

$$\frac{\delta}{\delta t}(\delta x^T \delta x) \leq 2\lambda_{max}(x, u, t)\delta x^T \delta x \tag{2}$$

This implies that the magnitude of the virtual displacement δx can also be bounded by the max eigenvalue $\lambda_{max}(x, u, t)$ [15, Eqn. 3],

$$\|\delta x\| \leq \|\delta x_0\| e^{\int_0^t \lambda_{max}(x,u,t)dt} \tag{3}$$

If λ_{max} is uniformly negative definite ($\exists \beta > 0, \forall t \geq 0, \lambda_{max}(x, u, t) \leq -\beta < 0$), then any infinitesimal length $\|\delta x\|$ converges exponentially to zero [15, Thm. 1]. Similarly, consider a differential volume δV around the trajectory, and the evolution of δV is defined as [15, Sec. 3.9],

$$\|\delta V\| = \|\delta V(t_0)\| e^{\int_0^t \mathrm{div} f(x,u,t)dt} \tag{4}$$

As a relaxation to the maximum eigenvalue λ_{max} being uniformly negative definite, [15] considers the case where the average eigenvalue λ_{avg} is uniformly negative definite. Since the divergence of f is just the sum of the eigenvalues in F [15], then if the average eigenvalue λ_{avg} is uniformly negative definite then so is the divergence of f. This implies that the magnitude of a differential volume δV converges exponentially to zero if λ_{avg} is uniformly negative definite.

These results motivated the creation of two divergence metrics in [7]: the maximal divergence metric $D_m := \lambda_{max}$ and the average divergence metric $D_a := \lambda_{avg}$. If D_m is uniformly negative definite for an entire trajectory, then all neighboring trajectories converge to a single trajectory [15, Thm. 1]. If D_a is uniformly negative definite for an entire trajectory, then on average all neighboring trajectories converge to a set of zero volume [15, Sec. 3.9].

2.2 Hill Climbing Problem

The well established hill climbing problem [5–7, 14] considers a mobile robot navigating hilly terrain as shown in Fig. 1. Since contraction analysis [15] requires a smooth system, the hill is modeled as a smooth height function, $h(x, y)$.

The hill gradient controller from [7] follows a constant speed α while choosing any arbitrary angle θ relative to the hill gradient ∇h,

$$f(x, y, \theta) = \alpha R(\theta) \frac{\nabla h(x, y)}{\|\nabla h(x, y)\|}, \tag{5}$$

where $R(\theta)$ is a rotation matrix. In [7], it is stated that one eigenvalue of the vector field Jacobian is zero. Therefore, the maximum divergence is equal to the average divergence, $D_m = D_a$, or equal to 0, $D_m = 0$. However, this is not entirely correct. While it is true that there is always some direction that is neither converging or diverging, it is possible that one eigenvalue is positive and one eigenvalue is negative. The controller can never have eigenvalues with the same sign, and thus the main conclusion that the controller can never have a negative max eigenvalue is still valid.

Lemma 1. *Given a vector field f defined by* (5) *and obtaining F through* (1)*, the maximum eigenvalue of F, λ_{max}, is non-negative, $\lambda_{max} \geq 0$.*

The proof is included in Lemma 2.

2.3 Power Controller

Define a new controller where the gradient is scaled by its norm to the p^{th}-power,

$$f(x, y, \theta, p) = \alpha R(\theta) \frac{\nabla h}{\|\nabla h\|^p}, \quad p \in \mathbb{R} \tag{6}$$

This new controller changes speed with the local steepness of the hill, and following it there can exist a negative maximum eigenvalue for certain choices of p with respect to the local curvature of the hill.

Lemma 2. *Given a vector field f defined by* (6) *and obtaining F through* (1)*, there exists a heading θ such that the maximum eigenvalue of F is negative, $\lambda_{max} < 0$, if one of the following condition is true: $p < 1$ for regions with positive hill curvature or $p > 1$ for regions with negative hill curvature.*

Proof. Define G to be the un-rotated part of the controller in (6),

$$G = \frac{\nabla h}{\|\nabla h\|^p} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} \tag{7}$$

Set θ to be piecewise constant, and thus there is no gradient of theta with respect to the states (x, y). Furthermore, because the speed scaling term α does not affect

the analysis, it is set to be unity. Therefore, the symmetric part of the Jacobian F of the vector field f is,

$$
\begin{aligned}
F &= \begin{bmatrix} a & c \\ c & b \end{bmatrix} := \frac{1}{2}(D_f + D_f^T) \\
a &= \frac{\partial G_x}{\partial x}\cos\theta - \frac{\partial G_y}{\partial x}\sin\theta, \qquad b = \frac{\partial G_y}{\partial y}\cos\theta + \frac{\partial G_x}{\partial y}\sin\theta \\
c &= \frac{1}{2}\left((\frac{\partial G_x}{\partial y} + \frac{\partial G_y}{\partial x})\cos\theta + (\frac{\partial G_x}{\partial x} - \frac{\partial G_y}{\partial y})\sin\theta \right)
\end{aligned} \tag{8}
$$

The characteristic equation of F is given by

$$
\lambda_{1,2} = \frac{a + b \pm \sqrt{a^2 - 2\,a\,b + b^2 + 4\,c^2}}{2} \tag{9}
$$

By (9), the maximum eigenvalue of F is defined as,

$$
\lambda_{max} = \frac{e\cos\theta + g\sin\theta + d}{2} \tag{10}
$$

where d, e and g are defined as,

$$
d = \sqrt{\frac{\partial G_x}{\partial x}^2 + \frac{\partial G_y}{\partial y}^2 + \left[\frac{\partial G_y}{\partial x}^2 + 2\frac{\partial G_y}{\partial x}\frac{\partial G_x}{\partial y} + \frac{\partial G_x}{\partial y}^2 \right] - 2\frac{\partial G_x}{\partial x}\frac{\partial G_y}{\partial y}} \tag{11}
$$

$$
e = \frac{\partial G_x}{\partial x} + \frac{\partial G_y}{\partial y}, \qquad g = \frac{\partial G_x}{\partial y} - \frac{\partial G_y}{\partial x} \tag{12}
$$

Then solve for θ when the maximum eigenvalue is zero and define $\phi = \tan^{-1}\left(\frac{g}{e}\right)$,

$$
0 = \tan^{-1}\left(\frac{g}{e}\right) \pm \cos^{-1}\left(-\frac{d}{\sqrt{e^2 + g^2}} \right) \tag{13}
$$

In (13), the $\pm$ term describes the range of diverging directions, and the first component ϕ is in the direction of $\max(\lambda_{max})$. A negative maximum eigenvalue exists when the term in the inverse cosine is within the open set from -1 to 1 and so,

$$
d^2 - (e^2 + g^2) < 0 \tag{14}
$$

By substituting values of d, e, and g and plugging in the derivatives of G (7) for the power controller (6) into the constraint (14), the existence of a negative λ_{max} can be evaluated using constraint,

$$
\left(\frac{\partial^2 h}{\partial x^2}\frac{\partial^2 h}{\partial y^2} - \frac{\partial^2 h}{\partial x \partial y}^2 \right)(p - 1) < 0 \tag{15}
$$

The original controller (5) has a p value of 1, and never satisfies (15). Therefore, there does not exist a θ which results in a negative λ_{max} for any position (x, y) on any hill $h(x, y)$ with $p = 1$.

The Gaussian curvature for a hill function $h(x, y)$ is,

$$K = \frac{\frac{\partial^2 h}{\partial x^2}\frac{\partial^2 h}{\partial y^2} - \frac{\partial^2 h}{\partial x \partial y}^2}{\left(1 + \frac{\partial h}{\partial x}^2 + \frac{\partial h}{\partial y}^2\right)^2} \tag{16}$$

The numerator in (16) determines the sign of the Gaussian curvature at a point (x, y) on a hill. By using the constraint in (15), it is clear that there exists a negative λ_{max} for regions in positive curvature with a p value less than 1, and for regions of negative curvature with a p value greater than 1, that is,

$$\begin{aligned} (\text{sign}(K) > 0 \ \wedge \ p < 1) \ &\rightarrow \ \exists\, \theta \ \text{s.t} \ \lambda_{max} < 0 \\ (\text{sign}(K) < 0 \ \wedge \ p > 1) \ &\rightarrow \ \exists\, \theta \ \text{s.t} \ \lambda_{max} < 0 \end{aligned} \tag{17}$$

□

2.4 Trajectory Optimization Framework

This section presents a method of automatically finding optimally convergent and smooth trajectories through trajectory optimization (T-OPT) using direct collocation [3]. The resulting trajectories approximate a smooth trajectory with N piecewise-smooth trajectories. The trajectory optimization framework uses a cost function which seeks to maximize convergence and smoothness of a trajectory. The first order dynamics of the mobile robot are enforced through linear collocation constraints. The trajectory's start and end positions are bounded to match the problems specifications. Nonlinear constraints are used to enforce uniformly negative definite divergence metrics $D_m < 0$ or $D_a < 0$.

The trajectories consist of N waypoints and 5 decision variables per waypoint i: the position $(x^{(i)}, y^{(i)})$, velocity $(\dot{x}^{(i)}, \dot{y}^{(i)})$, and controller power $p^{(i)}$. The heading angle $\theta^{(i)}$ and forward speed are encoded as a velocity vector $(\dot{x}, \dot{y})$ to avoid using nonlinear constraints for collocation. The velocity is calculated in a global frame and converted to a hill-relative angle θ as a post processing step. In this paper MathWorks MATLAB's nonlinear programming solver `fmincon` [17] is used with the sequential quadratic programming (SQP) algorithm. The trajectory optimization framework is initialized with a straight line trajectory from the starting point to the center of the goal region.

Cost Function The cost is defined as,

$$J(x, y, \dot{x}, \dot{y}, p) = \frac{1}{N-1} \sum_{i=1}^{N-1} \left(J_{D_a}^{(i)} + J_{D_m}^{(i)} + J_{accel}^{(i)} + J_{path}^{(i)} \right) \tag{18}$$

where

$$\begin{aligned} J_{D_a}^{(i)} &= \sigma D_a(x^{(i)}, y^{(i)}, \dot{x}^{(i)}, \dot{y}^{(i)}, p^{(i)}) \\ J_{D_m}^{(i)} &= \upsilon \max(D_m(x^{(i)}, y^{(i)}, \dot{x}^{(i)}, \dot{y}^{(i)}, p^{(i)}), -\epsilon) \\ J_{accel}^{(i)} &= \gamma[(\dot{x}^{(i+1)} - \dot{x}^{(i)})^2 + (\dot{y}^{(i+1)} - \dot{y}^{(i)})^2] \\ J_{path}^{(i)} &= \rho[(\dot{x}^{(i)})^2 + (\dot{y}^{(i)})^2] \end{aligned} \tag{19}$$

The cost function is designed to minimize average divergence J_{D_a}, encourage negative maximum eigenvalues J_{D_m}, smooth the trajectory by adding cost to big changes in velocity J_{accel}, and to minimize path length J_{path}, based on some user desired weighting $(\sigma, \upsilon, \gamma, \rho, \epsilon)$. The sum is evaluated up to $N-1$, because the cost associated to the Nth waypoint does not affect the trajectory.

Linear Constraints and Bounds Linear equality constraints are used to ensure first order dynamics.

$$\begin{aligned} x^{(i+1)} - x^{(i)} - \dot{x}^{(i)} dt = 0 \\ y^{(i+1)} - y^{(i)} - \dot{y}^{(i)} dt = 0 \end{aligned} \tag{20}$$

In addition, upper and lower bounds are defined to be:

$$\begin{aligned} x_{initial} \leq x^{(0)} \leq x_{initial} \\ y_{initial} \leq y^{(0)} \leq y_{initial} \\ x_{end} - \eta \leq x^{(n)} \leq x_{end} + \eta \\ y_{end} - \eta \leq x^{(n)} \leq y_{end} + \eta \\ x_{min} \leq x \leq x_{max} \\ y_{min} \leq y \leq y_{max} \\ p_{min} \leq p \leq p_{max} \end{aligned} \tag{21}$$

Initial and ending positions are bounded instead of using linear equality constraints to help the solver find solutions. The initial position is bounded to be the exact desired starting position, and the end position is bounded to be within a square with side length 2η around the desired end location. All positions are bounded to be within the limits of the defined map. Bounds on the power variable p are placed to ensure stable and realistic values.

Nonlinear constraints Nonlinear constraints are used to enforce the strict convergence results of [15, Thm. 1] and [15, Sec. 3.9]. The three different methods considered are:

1. Applying [15, Thm. 1] to ensure strict convergence at each waypoint:

$$D_m^{(i)} < 0 \quad \forall \quad 1 \leq i \leq N-1 \tag{22}$$

2. Applying [15, Sec 3.9] to ensure average convergence at each waypoint:

$$D_a^{(i)} < 0 \quad \forall \quad 1 \leq i \leq N-1 \tag{23}$$

3. No nonlinear constraints (unconstrained)

Trajectories where D_m are uniformly negative definite are resilient to uncertainty in all directions at all times and are guaranteed to exponentially converge to the nominal trajectory. In the case that D_a is uniformly negative definite, the

trajectories are on average robust to uncertainties, but may suffer from uncertainty in the direction of maximum divergence. Lastly, the unconstrained case is considered because there will more often be a feasible solution and the average divergence is minimized by the cost function, while in the constrained cases, feasible solution may often not exist due to the harshness of the constraints. The analytical solutions of D_m and D_a were calculated using the eigenvalues from (9) as well as their associated gradients with respect to the states. Due to the strictness and complexity of satisfying these constraints, the gradient played a crucial role in finding solutions and naively calling `eig` [17] did not find any feasible solutions for the ($D_m < 0$) constrained case.

2.5 RRT methods

The proposed trajectory optimization (T-OPT) methods are compared against the rapidly-exploring random tree (RRT) methods as shown in [7]. Specifically, biased RRT (B-RRT) and contraction region RRT (CR-RRT) are implemented. The T-OPT methods were also compared against asymptotic optimal variants of the RRT methods by utilizing (RRT*) [9]. Since both B-RRT and CR-RRT are extensions of Kinodynamic RRT (KD-RRT) [12], an RRT* version of each method was developed: B-RRT* and CR-RRT*.

Kinodynamic RRT (KD-RRT) KD-RRT for the hill climbing problem follows the same collocation constraints (20) and bounds on start position and goal position set (21) as for trajectory optimization. The planner builds a tree which starts at the desired starting position. At each iteration the planner samples a random position and finds the nearest neighbor in the RRT using the Euclidean distance metric. Random directions and powers p are sampled to generate a set of candidate actions. Each action is evaluated by finding the end point of a trajectory that follows that direction for a fixed distance. An action a is selected and a new node is added to the RRT based on which action reaches closest to the sampled point, using the same Euclidian distance metric. This is repeated until a node is within the goal position set (21).

Biased RRT (B-RRT) B-RRT attempts to minimize divergence by scaling the Euclidean distance metric used for action selection by e^{bD_i}, where $b \in \mathbb{R}$ is a bias term and D_i is the specified divergence metric to minimize. In our implementation, average divergence D_a is used.

Contraction Region RRT (CR-RRT) CR-RRT applies the same constraint as (22) at the action selection step to ensure a trajectory that meets the contraction region requirements at every step. However, in [7] there can never exist a true contraction region because a constant velocity controller was used (as proven in Lemma 2). Here, both D_m (22) and D_a (23) are used.

Biased RRT* (B-RRT*) and Contraction Region RRT* (CR-RRT*) RRT* follows the same algorithm as KD-RRT except that the goal was ran-

domly sampled 5% of the time and before adding the node to the RRT, there is a rewiring step which optimizes which node in the tree should connect to the candidate node. Nodes that are within a distance of r away from the candidate node are considered for rewiring. Branches are grown from all neighboring nodes to the candidate node, and the entire path length for each branch is calculated. The power variable p for each branch is chosen from the set $\{0, 1, 2\}$ to minimize the cost of the branch. The branch that leads to the minimum path length from the candidate node to the beginning node is added to the tree. Once a node is within the goal position set (21), the algorithm is iterated k more times to continue rewiring. B-RRT* uses the same biased distance metric as B-RRT in the rewiring step and CR-RRT* can apply either constraint (22) or (23) during the rewire step.

2.6 Simulation

To evaluate convergence of the generated trajectories (from either the trajectory optimization or RRT methods), a circle of 40 points of uncertainty with radius $r = \eta$ (half the side length of the desired goal region) was added to the trajectories' starting positions and simulated forwards by integrating the power controller's (6) dynamics using MathWorks MATLAB's `ode45` [17]. To ensure heading and velocity were nominally constant between each waypoint, heading angles θ and velocity scaling variable α were calculated in between each waypoint. The starting area of the circle and radius of the circle is compared against the convex hull of the particles at the end of the trajectory and the particle furthest from the nominal trajectories end position. Define two metrics of path convergence to be the ratio between the end particle area and the starting circle area E_a and the ratio between the maximum end particle and the starting circle radius E_m [7]. The area ratio E_a corresponds to average divergence D_a and the maximum distance ratio corresponds to the maximum divergence D_m. An example output of the simulation for a trajectory optimized using the unconstrained trajectory optimization method is shown in Fig. 2.

3 Experiments

This section demonstrates the effectiveness of the proposed trajectory optimization framework and the new power controller (6) for a mobile robot traversing hilly terrain. The hill function is chosen to be consistent with [7, 14],

$$h(x, y) := 3y + \sin(x + xy) \quad \text{s.t.} \quad (x, y) \in [-2, 2] \times [0, 2.5] \tag{24}$$

This hill contains a range of curvature content. Therefore, sampling random starting and ending positions evaluates the efficacy of each planner and controller on a collection of diverse landscapes.

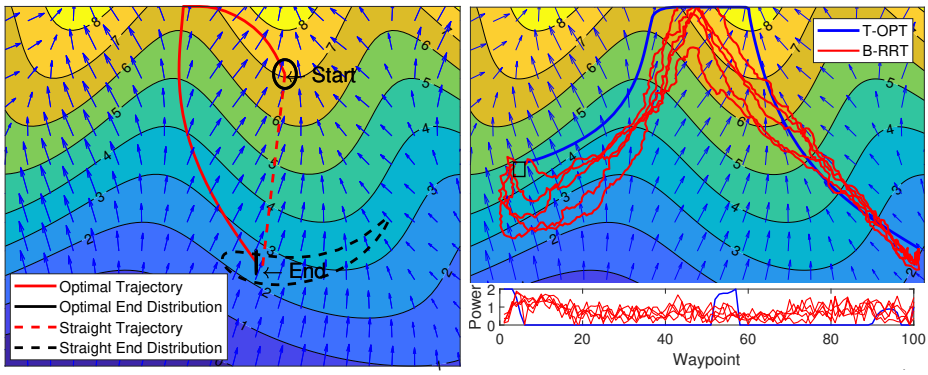

Fig. 2. Topographic map of a hill showing elevation lines and hill gradient (blue arrows). Left: Simulating forward a circle of uncertainty (black) around the nominal trajectory (red). An optimal trajectory (red solid) converged to a line (black solid) with an area ratio of $E_a = 0.02$ and an max of distance ratio $E_m = 0.88$ and a straight line method (red dashed) diverged to an end distribution (black dashed) with $E_a = 11.32$ and $E_m = 11.66$. Top Right: T-OPT (blue line) trajectory, 5 B-RRT (red line) trajectories, and the goal region (black square). Bottom Right: Hill power controller's (6) power value p at each waypoint for T-OPT (blue) and the 5 B-RRT (red).

3.1 Experiment Methods

To evaluate the convergence and smoothness of the trajectories generated using T-OPT, B-RRT, CR-RRT, B-RRT*, and CR-RRT*, 200 random start and end points were used to calculate an average area ratio E_a, max distance ratio E_m, and acceleration cost J_{accel}. Eleven different trajectory planning methods with different algorithms, power controller bounds, constraints on divergence metrics (22)-(23), and cost functions were tested:

1. (T-OPT), $p = 1$, unconstrained, minimize path length
2. (T-OPT), $p \in [0, 2]$, unconstrained, minimize cost function (19)
3. (B-RRT), $p \in [0, 2]$, unconstrained, bias $b = 1.5$ action size $a = 0.025$
4. (B-RRT*), $p \in [0, 2]$, bias $b = 1.5$ action size $a = 0.005$ rewire size $r = 0.025$
5. (T-OPT), $p \in [0, 2]$, $D_m < 0$, minimize cost function (19)
6. (CR-RRT), $p \in [0, 2]$, $D_m < 0$, $b = 1.5$, $a = 0.025$
7. (CR-RRT*), $p \in [0, 2]$, $D_m < 0$, $b = 1.5$, $a = 0.005$, $r = 0.025$
8. (T-OPT), $p \in [0, 2]$, $D_a < 0$, minimize cost function (19)
9. (T-OPT), $p = 1$, $D_a < 0$, minimize cost function (19)
10. (CR-RRT), $p \in [0, 2]$, $D_a < 0$, $b = 1.5$, $a = 0.025$
11. (CR-RRT*), $p \in [0, 2]$, $D_a < 0$, $b = 1.5$, $a = 0.005$, $r = 0.025$

The constants in the cost function (19) were heuristically tuned to be $\sigma = 30$, $\upsilon = 2$, $\gamma = 10^5$, $\rho = 10^4$, and $\epsilon = 1$. In the upper and lower bounds, the square goal region's side length $\eta = 0.05$ was set to be half of the acceptable goal region in [7]. Powers were bounded to allow a change of ± 1 from the nominal $p = 1$ since higher powers lead to instability, $p_{min} = 0$ and $p_{max} = 2$.

Table 1. Results for trajectory optimization (T-OPT), biased RRT (B-RRT), and contraction region RRT (CR-RRT) path planning methods over 200 random trials where the power of the controller was either $p = 1$ or in the closed set $p \in [0, 2]$. Results include success rate (S), log area ratio (E_a), log distance ratio (E_m), acceleration cost (J_{accel}), and planning time (T) in seconds. Mean ± one standard deviation are listed.

Method	p	S	$\log(E_a)$	$\log(E_m)$	J_{accel}	T
shortest						
1 T-OPT	1	100	0.09 ± 1.12	0.685 ± 0.686	0.406 ± 0.325	0.741 ± 0.478
uncon						
2 T-OPT	$[0, 2]$	100	-3.26 ± 2.66	0.425 ± 0.736	5.89 ± 2.66	219 ± 37.8
3 B-RRT	$[0, 2]$	96.5	-3.03 ± 1.67	-0.00146 ± 0.170	31.8 ± 7.62	73 ± 108
4 B-RRT*	$[0, 2]$	94	-3.08 ± 1.55	0.133 ± 0.243	7.96 ± 1.79	181 ± 259
$Dm < 0$						
5 T-OPT	$[0, 2]$	4.5	-5.17 ± 1.52	-0.221 ± 0.167	76.3 ± 95.8	244 ± 81.3
6 CR-RRT	$[0, 2]$	7.5	-1.78 ± 1.89	-0.124 ± 0.116	20 ± 10.4	32.3 ± 65
7 CR-RRT*	$[0, 2]$	1	-0.20 ± 0.13	-0.005 ± 0.033	5.55 ± 2.63	2160 ± 3060
$Da < 0$						
8 T-OPT	$[0, 2]$	92.5	-4.2 ± 1.67	0.0633 ± 0.168	6.52 ± 3.28	714 ± 440
9 T-OPT	1	84	-3.06 ± 1.74	0.0921 ± 0.156	6.37 ± 12.7	121 ± 35.5
10 CR-RRT	$[0, 2]$	93	-3.17 ± 1.69	-0.0176 ± 0.174	29.2 ± 8.28	96.6 ± 151
11 CR-RRT*	$[0, 2]$	83.5	-3.01 ± 1.5	0.028 ± 0.278	9.2 ± 2.27	708 ± 939

3.2 Results

The success rate S, average log area ratio E_a, average log max distance ratio E_m, and planning time for each tested method is shown in Table 1. As expected, planning for the shortest path led to an average positive log area ratio E_a, and using converging planning methods led to negative log area ratio E_a averages. A paired difference test is used for mean comparisons. All comparisons discussed are significant ($p < 0.05$) unless noted.

Unconstrained Problems Method 2 (unconstrained T-OPT) on average created trajectories with lower log area ratios E_a than method 3 (unconstrained B-RRT) and method 4 (unconstrained B-RRT*) as shown in Table 2. However, methods 3 and 4 had a lower mean log area ratio E_a than method 2 when using paired differences. This resulted from the few divergent paths created from method 2 dominating the comparisons. Divergent paths occurred because in some rare cases, T-OPT picked an early divergent step in a trajectory to gain access to more convergent steps later in the trajectory. B-RRT and B-RRT* are not affected by this issue because it they are inherently greedy search algorithms, and are biased towards the most convergent direction at each waypoint. To avoid this issue for T-OPT, a saturation limit can be placed on the average divergence cost J_{D_a} or the $D_a < 0$ constraint can be applied. Method 2 was able to find a feasible solution for all 200 random trials while methods 3 and 4 could not. Methods 3 and 4 also had a lower mean log E_m than method 2, likely because

Table 2. Comparing the percentage of trials that the end area ratio E_a is smaller using the method on the vertical axis compared to the method on the horizontal axis.

Method \ Compared Method Number	8	2	4	9	11	10	3
8 T-OPT, $p \in [0,2]$, $D_a < 0$	0	0.66	0.67	0.9	0.73	0.67	0.72
2 T-OPT, $p \in [0,2]$, uncon	0.34	0	0.6	0.72	0.63	0.6	0.63
4 B-RRT*, $p \in [0,2]$, uncon	0.33	0.4	0	0.5	0.66	0.64	0.64
9 T-OPT, $p = 1$, $D_a < 0$	0.099	0.28	0.5	0	0.59	0.57	0.57
11 CR-RRT*, $p \in [0,2]$, $D_a < 0$	0.27	0.37	0.34	0.41	0	0.51	0.53
10 CR-RRT, $p \in [0,2]$, $D_a < 0$	0.33	0.4	0.36	0.43	0.49	0	0.57
3 B-RRT, $p \in [0,2]$, uncon	0.28	0.37	0.36	0.42	0.47	0.43	0

RRT inherently randomizes the direction of max divergence, while the optimal solution tends to keep the worst direction aligned in a similar direction. Because this hill is smooth, and the hill gradient does not rapidly change, resulting T-OPT solutions tend to grow uncertainty in the same direction. Method 4 is smoother than method 3 when comparing acceleration cost J_{accel} which in turn likely made method 4 have a greater mean log E_m than method 3.

Constrained Maximum Convergence For this hill climbing problem, these are the first results where trajectories can be produced with a contraction region. Method 6 (CR-RRT, $D_m < 0$) found 15 feasible solutions while method 5 (T-OPT, $D_m < 0$) only found 9 and method 7 (CR-RRT*, $D_m < 0$) found 2. However, all but one trajectory has a negative log maximum distance ratio E_m for method 5 and all trajectories for method 7 have a negative log maximum distance ratio E_m, while 3 of the 15 trajectories produced by method 6 have a positive log maximum distance ratio E_m. The resulting positive log maximum distance ratio E_m is due to the action size a being too large in comparison to the geometry of the hill. Since planning time grew exponentially with increasing tree size, the action size could not be further reduced. In the single case that method 4 had a positive log distance ratio E_m, the maximum action size was also large ($a = 0.4518$ on average).

Constrained Average Convergence In the case of constraining the average divergence to be negative ($D_a < 0$), method 8 (T-OPT, $p \in [0,2]$, $D_m < 0$) was able to find 9 % more feasible solutions than method 9 (T-OPT, $p = 1$, $D_m < 0$), 9.5 % more feasible solutions than method 11 (CR-RRT*, $D_m < 0$), and 0.5 % fewer feasible solutions than method 10 (CR-RRT, $D_m < 0$). Method 8 has a lower mean log area ratio E_a than methods 9, 10, and 11. However, methods 8 and 11 took about 7 times longer to compute than methods 9 and 10, and like the unconstrained problems, the RRT methods have a lower maximum distance ratio E_m, Table 1.

Log Area Ratio E_a Direct comparisons For a more fair comparison between methods, direct trial comparisons for log area ratio E_a are made between

the unconstrained and the average divergence constrained ($D_a < 0$) methods as shown in Table 2. Only runs where both methods had a feasible solution were compared. Because constraining ($D_m < 0$) led to few solutions, the strict convergence case was not directly compared against the other methods. For all problems, method 8 (T-OPT, $p \in [0, 2]$, $D_a < 0$) has on average a lower log area ratio E_a for each direct comparison. In second place, method 2 (T-OPT, $p \in [0, 2]$, unconstrained) has a lower log area ratio E_a for the majority of problems except when compared to method 8. Method 9 (T-OPT, $p = 1$, $D_a < 0$) is worse than all other T-OPT cases, but is slightly better than or equal to the RRT and RRT* cases methods (3,4,10,11). When directly comparing the power controller (6), method 8, versus the constant speed controller (5), method 9, the power controller is more convergent 90.1% of the time.

Trajectory Smoothness Trajectories generated by trajectory optimization were smoother than both RRT* and RRT methods in both position choices and power choices. An example run shown in the top side of Fig. 2 compares trajectories generated using T-OPT and B-RRT. The bottom side of Fig. 2 shows that the powers chosen by T-OPT are smoother and more consistent than the powers chosen by the B-RRT runs. The acceleration cost J_{accel} was imposed to the trajectory optimization framework to reduce sudden changes in linear and rotational velocity. When comparing the acceleration cost over the 200 trials between the unconstrained (methods 2, 3, and 4), and constrained ($D_a < 0$, methods 8, 9, 10, and 11), T-OPT had a smaller mean acceleration cost J_{accel} than B-RRT and B-RRT*. There were not enough successful trials for the ($D_m < 0$) constrained cases to draw meaningful conclusions on smoothness.

4 Conclusion

Planning trajectories which are robust to uncertainty is critical for creating reliable robotic systems. Typically, uncertainty is reduced by using closed-loop feedback which can sense and reduce errors. However, in situations like the hill climbing problem, the robot is under-sensed and cannot reduce error by using closed-loop feedback. Instead it must rely on reducing error by leveraging the geometry of the underlying vector field.

Prior work [7, 14] to create robust trajectories using convergent rapidly-exploring random trees. However, the resulting trajectories are not smooth and at best could only enforce average convergence. The work in this paper creates a new convergent optimization framework which generates optimally smooth and convergent trajectories.

This work also introduces a new hill navigation controller (6) which enables the possibility for strictly convergent trajectories while a constant speed controller (5) can only produce average convergent trajectories. The new power controller was also more convergent 90% of the time and found 9% more feasible solutions than the constant speed controller when solving for average convergent trajectories using T-OPT. Although a low number of trials emitted a feasible

solution when applying the strict convergence constraint, the work in this paper is the first example of finding contraction regions for this hill climbing problem. The low number of solutions is due to the specifics of the problem and how strict this requirement is. Problems that more readily admit such solutions would see higher success rate.

Trajectories generated through the trajectory optimization framework on average are more convergent and smoother than the ones produced using RRT and RRT* methods. However, RRT and RRT* methods produced trajectories with smaller max distance ratios E_m than the T-OPT methods when not constraining maximum divergence to be negative. We believe the randomness when picking heading directions from RRT methods help shrink the maximum distance ratio, because the maximum divergence direction is rapidly changing at each waypoint, while the optimal solution keeps the direction of maximum divergence aligned in a similar direction throughout the trajectory. However, solutions produced by RRT methods would need to first be smoothed out before implementation, and may lose the benefits of randomized alignment. Post-processing these trajectories may lead to less optimal solutions or violation of the convergence constraints. These 2 drawbacks are apparent in the RRT* trajectories. The trajectories produced by RRT* are smoother than the ones produced by RRT, but in the process, it is likely that the trajectories aligned the maximum divergence direction in one direction more and worsened the max distance ratio E_m. In the constrained RRT* cases, RRT* could not exploit creating jagged paths to satisfy the constraints, and ultimately emitted few or poor convergence trajectories. On the other hand, trajectories generated from T-OPT can readily be implemented on a robot with simple unicycle dynamics while guaranteeing optimality and constraints on divergence.

In general, optimizing for convergence is a useful tool for tasks where robots are undersensed or underactuated. The presented trajectory optimization framework and convergent controller analysis can also be expanded to more dynamic path planning problems. The framework will directly translate to other 2 dimensional state space problems. Modifications to eigenvalue analytical calculations must be made for higher dimensions because they are trivial in the 2D case. In the future, we plan to analyze the uncertainty growth in non-smooth systems such as hybrid dynamical systems [2], and to utilize a similar optimization framework to plan robust walking behaviors for legged robots.

Acknowledgment

Research was sponsored by the Army Research Office and was accomplished under Grant Number W911NF-19-1-0080. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

1. Ronald Arkin. Behavior-based robot navigation for extended domains. *Adaptive Behavior*, 1(2):201–225, 1992.
2. Samuel A. Burden, Thomas Libby, and Samuel D. Coogan. On contraction analysis for hybrid systems. *CoRR*, abs/1811.03956, 2018.
3. Charles R Hargraves and Stephen W Paris. Direct trajectory optimization using nonlinear programming and collocation. *Journal of Guidance, Control, and Dynamics*, 10(4):338–342, 1987.
4. Nathanael Hyafil and Fahiem Bacchus. Conformant probabilistic planning via csps. In *ICAPS*, volume 98, pages 205–214, 2003.
5. B. Deniz Ilhan, Aaron M. Johnson, and D. E. Koditschek. Autonomous legged hill ascent. *Journal of Field Robotics*, 35(5):802–832, August 2018.
6. Aaron M Johnson, Matthew T Hale, Galen C Haynes, and Daniel E Koditschek. Autonomous legged hill and stairwell ascent. In *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, pages 134–142. IEEE, 2011.
7. Aaron M Johnson, Jennifer E King, and Siddhartha Srinivasa. Convergent planning. *IEEE Robotics and Automation Letters*, 1(2):1044–1051, 2016.
8. Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.
9. Sertac Karaman and Emilio Frazzoli. Incremental sampling-based algorithms for optimal motion planning. *arXiv preprint arXiv:1005.0416*, 2010.
10. Inyoung Ko, Beobkyoon Kim, and Frank Chongwoo Park. Vf-rrt: Introducing optimization into randomized motion planning. In *2013 9th Asian Control Conference (ASCC)*, pages 1–5. IEEE, 2013.
11. Michael C Koval, Jennifer E King, Nancy S Pollard, and Siddhartha S Srinivasa. Robust trajectory selection for rearrangement planning as a multi-armed bandit problem. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2678–2685. IEEE, 2015.
12. Steven M LaValle and James J Kuffner Jr. Randomized kinodynamic planning. *The international journal of robotics research*, 20(5):378–400, 2001.
13. Yanbo Li, Zakary Littlefield, and Kostas E Bekris. Asymptotically optimal sampling-based kinodynamic planning. *The International Journal of Robotics Research*, 35(5):528–564, 2016.
14. Kaiwen Liu, Yang Zhang, Andrew Dobson, and Dmitry Berenson. Asymptotically near-optimal methods for kinodynamic planning with initial state uncertainty. *IEEE Robotics and Automation Letters*, 2019.
15. Winfried Lohmiller and Jean-Jacques E Slotine. On contraction analysis for nonlinear systems. *Automatica*, 34(6):683–696, 1998.
16. Tomas Lozano-Perez, Matthew Mason, and Russell H. Taylor. Automatic synthesis of fine-motion strategies for robots. *The International Journal of Robotics Research*, 3, 03 1984.
17. Matlab, 2018. The MathWorks, Natick, MA, USA.