

Rationally Inattentive Path-Planning via RRT*

Ali Reza Pedram^{1†}, Jeb Stefan^{2†}, Riku Funada³ and Takashi Tanaka³

Abstract— We consider a path-planning scenario for a mobile robot traveling in a configuration space with obstacles under the presence of stochastic disturbances. A novel path length metric is proposed on the uncertain configuration space and then integrated with the existing RRT* algorithm. The metric is a weighted sum of two terms which capture both the Euclidean distance traveled by the robot and the perception cost, i.e., the amount of information the robot must perceive about the environment to follow the path safely. The continuity of the path length function with respect to the topology of the total variation metric is shown and the optimality of the Rationally Inattentive RRT* algorithm is discussed. Three numerical studies are presented which display the utility of the new algorithm.

I. INTRODUCTION

As robots are designed to be more self-reliant in navigating complex and stochastic environments, it is sensible for the strategic execution of perception/cognition tasks to be included in the theory which governs their path-planning [1]–[3]. Even though the body of work surrounding motion planning techniques has greatly expanded recently, a technological gap remains in the integration of perception concerns into planning tasks [4]. Mitigating this gap is paramount to missions which require robots to autonomously complete tasks when sensing actions carry high costs (battery power, computing constraints, etc).

Path-planning is typically followed by feedback control design, which is executed during the path following phase. In the current practice, path-planning and path-following are usually discussed separately (notable exceptions include [5]–[7]), and the cost of feedback control (perception cost in particular) is not incorporated in the path-planning phase. The first objective of this work is to fill this gap by introducing a novel path cost function which incorporates the expected perception cost accrued during path-following into the planning phase. This cost jointly penalizes the amount of sensing needed to follow a path and the distance traveled. Our approach is closely related to the concept of *rationally inattentive (RI) control* [8] (topic from macroeconomics which has recently been applied in control theory [9], [10]). The aim of rationally inattentive control is to jointly design the control and sensing policies such that the least amount

of information (measured in *bits*) is collected about the environment in order to achieve the desired control.

The second objective of this work is to integrate the proposed path length function with an existing sampling-based algorithm, such as Rapidly-Exploring Random Trees (RRT) [11]. The RRT algorithm is suited for this problem as it has been shown to find feasible paths in motion planning problems quickly. A modified version of this algorithm, RRT* [12], will be utilized as it has the additional property of being asymptotically optimal. We develop an RRT*-like algorithm incorporating the proposed path length function (called the RI-RRT* algorithm) and demonstrate its effectiveness.

While the practical utility of the proposed framework must be thoroughly studied in the future, its expected impact is displayed in Fig. 1. This figure shows the example of a robot moving through the two-dimensional, obstacle-filled environment. Path A (red) represents the path from the origin to target location which minimizes the Euclidean distance. However, this path requires a large number of sensor actuations to keep the robot’s spatial uncertainty small and avoid colliding with obstacles. Alternatively, Path B (blue) allows for the covariance to safely grow more along the path. Although the Path B travels a greater Euclidean distance to reach the target, it is cheaper in the information-theoretic sense as it requires fewer sensing actions. Therefore, if the perception cost is weighed more than the travel cost, Path B is characterized as the shortest path in the proposed path planning framework. We will demonstrate this effect in a numerical simulation in Section V-C.

The proposed concept of rationally inattentive path-planning provides insight into the mathematical modeling of human experts’ skills in path planning [13], especially in terms of an efficiency-simplicity trade-off. Several path-planning algorithms have been proposed in the literature that are capable of enhancing path simplicity; this list includes potential field approaches [14], multi-resolution perception and path-planning [15], [16], and safe path-planning [17], [18]. The information-theoretic distance function we introduce in this paper can be thought of as an alternative measure of path simplicity, which may provide a suitable modeling of the human intuition for simplicity in planning. In our standard, a path which requires less sensor information during the path-following phase is more “simple”; Path B in Fig. 1 is simpler than Path A, and the simplest path is that which is traceable by an open-loop control policy.

The contributions of this paper are summarized as follows:

- A novel path cost (RI cost) is formulated which jointly accounts for travel distance and perception cost.

*This work is supported by Lockheed Martin Corporation, FOA-AFRL-AFOSR-2019-0003 and DARPA Grant D19AP00078.

† AR. Pedram and J. Stefan contributed equally to this work.

¹Walker Department of Mechanical Engineering, University of Texas at Austin. apedram@utexas.edu. ²Odyssey Space Research. jeb.stefan@odysseysr.com. ³Department of Aerospace Engineering and Engineering Mechanics, University of Texas at Austin. riku.funada@austin.utexas.edu and ttanaka@utexas.edu.

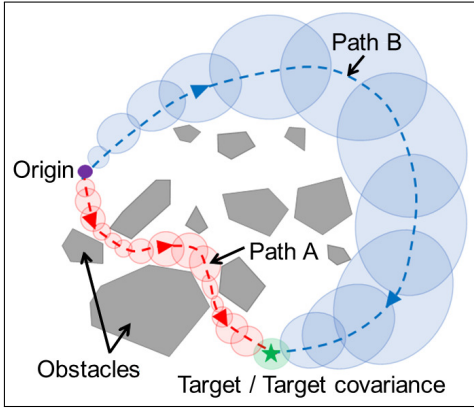


Fig. 1. Example of an autonomous robot navigating a two-dimensional configuration space with obstacles. The goal of the robot is to reach the target location. As it moves, the uncertainty of the robot’s exact location in the environment grows, represented by the varying sized covariance ellipses.

- The continuity of the path cost with respect to the topology of the total variation metric is shown in the single dimensional case, which is a step forward to guaranteeing the asymptotic optimality of sampling-based algorithms.
- An RRT*-like algorithm is produced implementing the RI path-planning concept.

Notation: For the purpose of this work, the following definitions for vectors (lower case) and matrices (upper case) hold: $S^d = \{P \in \mathbb{R}^{d \times d} : P \text{ is symmetric}\}$, $S_{++}^d = \{P \in S^d : P \succ 0\}$, $S_\epsilon^d = \{P \in S^d : P \preceq \epsilon I\}$, and bold symbols such as \mathbf{x} represent random variables. The vector 2-norm is $\|\cdot\|_2$ and $\|\cdot\|_F$ is Frobenius norm. The maximum singular values of a matrix M is denoted by $\sigma(M)$.

II. PRELIMINARY MATERIAL

In this paper, we consider a path-planning problem for a mobile robot with dynamics given by model (1). Let $\mathbf{x}(t)$ be a \mathbb{R}^d -valued random process representing the robot’s position at time t , given by the controlled Ito process:

$$d\mathbf{x}(t) = \mathbf{v}(t)dt + W^{\frac{1}{2}}db(t), \quad (1)$$

with $\mathbf{x}(0) \sim N(x_0, P_0)$ and $t \in [0, T]$. Here, $\mathbf{v}(t)$ is the velocity input command, $\mathbf{b}(t)$ is the d -dimensional standard Brownian motion, and W is a given positive definite matrix used in modeling the process noise intensity. We assume that the robot is commanded to travel at a unit velocity (i.e., $\|\mathbf{v}(t)\|_2 = 1$). Let $P = (0 = t_0 < t_1 < \dots < t_N = T)$ be a partition of $[0, T]$, which must not necessarily be of equal spacing. Time discretization of (1) based on the Euler-Maruyama method [19] yields:

$$\mathbf{x}(t_{k+1}) = \mathbf{x}(t_k) + \mathbf{v}(t_k) \Delta t_k + \mathbf{n}(t_k), \quad (2)$$

where $\Delta t_k = t_{k+1} - t_k$ and $\mathbf{n}(t_k) \sim N(0, \Delta t_k W)$. Introducing a new control input $\mathbf{u}(t_k) := \mathbf{v}(t_k) \Delta t_k$ and applying the constraint $\|\mathbf{v}(t_k)\|_2 = 1$, (2) can be written as:

$$\mathbf{x}(t_{k+1}) = \mathbf{x}(t_k) + \mathbf{u}(t_k) + \mathbf{n}(t_k), \quad (3)$$

with $\mathbf{n}(t_k) \sim N(0, k\mathbf{u}(t_k)kW)$. Due to the unit velocity assumption above, the time intervals $\Delta t_k, k = 0, 1, 2, \dots$ are determined once the command sequence $\mathbf{u}(t_0), \mathbf{u}(t_1), \mathbf{u}(t_2), \dots$ is formalized. Since the physical times t_k do not play significant roles in our theoretical development in the sequel, it is convenient to rewrite (3) as the main dynamics model of this work:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{u}_k + \mathbf{n}_k, \quad \mathbf{n}_k \sim N(0, k\mathbf{u}_k kW). \quad (4)$$

Let the probability distributions of the robot position at a given time step k be parametrized by a Gaussian model $\mathbf{x}_k \sim N(x_k, P_k)$, where $x_k \in \mathbb{R}^d$ is the nominal position and $P_k \in S_{++}^d$ is the associated covariance matrix (with d being the dimension of the configuration space). In this paper, we consider a path-planning framework in which the sequence $f(x_k, P_k)g_{k \in \mathbb{N}}$ is scheduled. Following [20], [21], the product space $\mathbb{R}^d \times S_{++}^d$ is called the *uncertain configuration space*. In what follows, the problem of finding the shortest path in the uncertain configuration space with respect to a novel information-theoretic path length function is formulated.

First, an appropriate directed distance function from a point $(x_k, P_k) \in \mathbb{R}^d \times S_{++}^d$ to another $(x_{k+1}, P_{k+1}) \in \mathbb{R}^d \times S_{++}^d$ is introduced. This function is interpreted as the cost of steering the random state variable $\mathbf{x}_k \sim N(x_k, P_k)$ to $\mathbf{x}_{k+1} \sim N(x_{k+1}, P_{k+1})$ in the next time step under the dynamics provided by (4). In order to implement the rational inattention concept, we formulate this cost as a weighted sum of the control cost $D_{\text{cont}}(k)$ and the information cost $D_{\text{info}}(k)$ in achieving each state transition.

A. Control Cost

The control cost is simply the commanded travel distance in the Euclidean metric:

$$D_{\text{cont}}(k) := \|x_{k+1} - x_k\|. \quad (5)$$

B. Information Cost

Jointly accounting for both the control efficiency and sensing simplicity in planning necessitates the formulation of a metric that captures the information acquisition cost required for path following. We utilize the information gain (entropy reduction) for this purpose.

Assume that the control input $\mathbf{u}_k = x_{k+1} - x_k$ is applied to (4). The propagation of the prior covariance during the movement of the robot, over the time interval $[t_k, t_{k+1})$, is denoted as $\hat{P}_k = P_k + kx_{k+1} - x_k kW$. At time t_{k+1} , the covariance is “reduced” to $P_{k+1}(\hat{P}_k)$ by utilizing a sensor input. The minimum information gain (minimum number of *bits* that must be contained in the sensor data) for this transition is:

$$D_{\text{info}}(k) = \frac{1}{2} \log_2 \det \hat{P}_k - \frac{1}{2} \log_2 \det P_{k+1}. \quad (6)$$

The notion of an “optimal” sensing signal which reduces \hat{P}_k to P_{k+1} has been previously discussed in [22] in the context of optimal sensing in filtering theory. The information cost function $D_{\text{info}}(k)$ in (6) is well-defined for the pairs

(P_k, P_{k+1}) satisfying $P_{k+1} \preceq \hat{P}_k$. For those pairs which do not satisfy $P_{k+1} \preceq \hat{P}_k$, we generalize (6) as:

$$D_{\text{info}}(k) = \min_{Q_{k+1} \succeq 0} \frac{1}{2} \log_2 \det \hat{P}_k - \frac{1}{2} \log_2 \det Q_{k+1} \quad (7)$$

s.t. $Q_{k+1} \preceq P_{k+1}, Q_{k+1} \preceq \hat{P}_k$.

Notice that (7) takes a non-negative value for any given transition from an origin (x_k, P_k) to destination (x_{k+1}, P_{k+1}) . However, (7) is an implicit function involving a convex optimization problem in its expression (more precisely, the max-det problem [23]). To see why (7) is an appropriate generalization of (6), consider a two-step procedure $\hat{P}_k \rightarrow Q_{k+1} \rightarrow P_{k+1}$ to update the prior covariance \hat{P}_k to the posterior covariance P_{k+1} . In the first step, the uncertainty is “reduced” from \hat{P}_k to satisfy both $Q_{k+1} \preceq \hat{P}_k$ and $Q_{k+1} \preceq P_{k+1}$. The associated information gain (the amount of telemetry data) is $\frac{1}{2} \log_2 \det \hat{P}_k - \frac{1}{2} \log_2 \det Q_{k+1}$. In the second step, the covariance Q_{k+1} is “increased” to P_{k+1} ($Q_{k+1} \preceq P_{k+1}$). This step incurs no information cost, since the location uncertainty can be increased simply by “deteriorating” the prior knowledge. The max-det problem (7) can then be interpreted as finding the optimal intermediate step Q_{k+1} which minimizes the information gain in the first step.

C. Total Cost

The cost to steer a random state variable $\mathbf{x}_k \sim N(x_k, P_k)$ to $\mathbf{x}_{k+1} \sim N(x_{k+1}, P_{k+1})$ is a weighted sum of $D_{\text{cont}}(k)$ and $D_{\text{info}}(k)$. Introducing $\alpha > 0$, the total RI cost is:

$$\begin{aligned} D(x_k, x_{k+1}, P_k, P_{k+1}) &:= D_{\text{cont}}(k) + \alpha D_{\text{info}}(k) \\ &= \min_{Q_{k+1} \succ 0} kx_{k+1} - x_k k \\ &\quad + \frac{\alpha}{2} \left[\log_2 \det \hat{P}_k - \log_2 \det Q_{k+1} \right] \quad (8) \\ \text{s.t. } &Q_{k+1} \preceq P_{k+1}, Q_{k+1} \preceq \hat{P}_k. \end{aligned}$$

By increasing α , more weight is placed on the amount of information which must be gained compared to the distance traversed. Note that the information cost D_{info} is an asymmetric function, so that transitioning $(x_1, P_1) \rightarrow (x_2, P_2)$ does not return the same cost as $(x_2, P_2) \rightarrow (x_1, P_1)$.

III. PROBLEM FORMULATION

Having introduced the RI cost function (8), it is now appropriate to introduce the notion of path length. Let $\gamma : [0, T] \rightarrow \mathbb{R}^d \times S_{++}^d, \gamma(t) = (x(t), P(t))$ be a path. The RI length of a path γ is defined as:

$$c(\gamma) := \sup_P \sum_{k=0}^{N-1} D(x(t_k), x(t_{k+1}), P(t_k), P(t_{k+1})),$$

where the supremum is over the space of partitions P of $[0, T]$. If $\gamma(t)$ is differentiable and $W = \frac{d}{dt} P(t) \otimes t \succeq [0, T]$, then it can be shown that:

$$c(\gamma) = \int_0^T \left[\left\| \frac{d}{dt} x(t) \right\|^2 + \frac{\alpha}{2} \text{Tr} \left(W \frac{d}{dt} P(t) \right) P^{-1}(t) \right] dt$$

A. Topology on the path space

In this subsection, we introduce a topology for the space of paths $\gamma : [0, T] \rightarrow \mathbb{R}^d \times S_{++}^d$, which is necessary to discuss continuity of $c(\gamma)$. The space of all paths $\gamma : [0, T] \rightarrow \mathbb{R}^d \times S_{++}^d$ can be thought of as a subset (convex cone) of the space of *generalized paths* $\gamma : [0, T] \rightarrow \mathbb{R}^d \times S^d$. The space of generalized paths is a vector space on which addition and scalar multiplication exist and are defined as $(\gamma_1 + \gamma_2)(t) = (x_1(t) + x_2(t), P_1(t) + P_2(t))$ and $\alpha\gamma(t) = (\alpha x(t), \alpha P(t))$ for $\alpha \geq \mathbb{R}$, respectively. Assuming that a path can be partitioned such that $P = (0 = t_0 < t_1 < \dots < t_N = T)$, the variation $V(\gamma; P)$ of a generalized path γ with respect to the choice of P is given by:

$$V(\gamma; P) := kx(0)k + \sigma(P(0)) + \sum_{k=0}^{N-1} \left[kx_k k + \sigma(P_k) \right]$$

where $x_k = x(t_{k+1}) - x(t_k)$, and $P_k = P(t_{k+1}) - P(t_k)$. Utilizing the above definition for the variation of a path, the total variation of a generalized path γ corresponds to the partition P which results in the supremum of the variation:

$$j\gamma j_{\text{TV}} := \sup_P V(\gamma; P).$$

Notice that $j\gamma j_{\text{TV}}$ defines a norm on the space of generalized paths. The following relationship holds between $j\gamma j_{\text{TV}}$ and

$$k\gamma k_{\infty} := \sup_{t \in [0, T]} kx(t)k + \sigma(P(t)).$$

Lemma 1: [24, Lemma 13.2] For a given path γ with partitioning P the following inequality holds:

$$k\gamma k_{\infty} \leq j\gamma j_{\text{TV}}.$$

Proof: See Appendix I for proof. \blacksquare

In what follows, we assume on the space of generalized paths $\gamma : [0, T] \rightarrow \mathbb{R}^d \times S^d$ the topology of total variation metric $j\gamma_1 j_{\text{TV}} - j\gamma_2 j_{\text{TV}}$, which is then inherited to the space of paths $\gamma : [0, T] \rightarrow \mathbb{R}^d \times S_{++}^d$. We denote by $BV[0, T]$ the space of paths $\gamma : [0, T] \rightarrow \mathbb{R}^d \times S_{++}^d$ such that $j\gamma j_{\text{TV}} < \infty$. In the next subsection, we discuss the continuity of the RI path cost $c(\cdot)$ in the space $BV[0, T]$.

B. Continuity of RI Cost Function

The continuity of RI path cost function plays a critical role in determining the theoretical guarantees we can provide when we use sampling-based algorithms to find the shortest RI path. Specifically, the asymptotic optimality (the convergence to the path with the minimum cost as the number of nodes is increased) of RRT* algorithms [12], the main numerical method we use in this paper, expects the continuity of the path cost function. Showing that the RI cost function (8) is continuous requires additional derivation which is shown via Theorem 1.

Theorem 1: When $d = 1$, the path cost function $c(\cdot)$ is continuous in the sense that for every $\gamma \in BV[0, T]$, $\gamma' : [0, T] \rightarrow \mathbb{R}^1 \times S_{2\epsilon}^1$, and for every $\epsilon_0 > 0$, there exists $\delta > 0$ such that

$$j\gamma' j_{\text{TV}} < \delta \implies |c(\gamma') - c(\gamma)| < \epsilon_0.$$

Proof: See Appendix II for proof. ■

Before discussing the required modifications for implementing RRT* algorithm with RI cost in Section IV, we first characterize the shortest RI path in the obstacle-free space, and then formally define the shortest RI path problem in obstacle-filled spaces in the following subsections.

C. Shortest Path in Obstacle-Free Space

In obstacle-free space, it can be shown that the optimal path cost between $z_1 = (x_1, P_1)$ and $z_2 = (x_2, P_2)$ is equal to $D(z_1, z_2)$. In other words, the triangular inequality $D(z_1, z_2) \leq D(z_1, z_{int}) + D(z_{int}, z_2)$ holds. This means it is optimal for the robot to follow the direct path (i.e. the straight line) from x_1 to x_2 without sensing through which the uncertainty grows to $\hat{P} = P_1 + kx_2 - x_1kW$, and then make a measurement at x_2 to shrink the uncertainty from \hat{P}_1 to P_2 . In what follows, we call such a motion plan the “move-and-sense” strategy. The optimality of the move-and-sense path for one-dimensional geometric space is shown in Appendix III. We confirm this optimality by simulation in Section V-A, where the move-and-sense path is the wedge-shaped path depicted in Fig. 3 (a).

D. Shortest Path Formulation

The utility of path-planning algorithms is made non-trivial by the introduction of obstacles in the path space. Let $X_{obs}^\ell \subset \mathbb{R}^d$ be a closed subset of spatial points representing obstacle $\ell \in \{1, \dots, M\}$. The initial configuration of the robot is defined as $z_{init} = (x_0, P_0) \in \mathbb{R}^d \times S_{++}^d$, while $Z_{target} \subset \mathbb{R}^d \times S_{++}^d$ is a given closed subset representing the target region which the robot desires to attain. Given a confidence level parameter $\chi^2 > 0$, the shortest RI path problem can be formulated as:

$$\begin{aligned} & \min_{\gamma \in \mathcal{BV}[0, T]} c(\gamma) \\ & \text{s.t.} \quad \gamma(0) = z_{init}, \quad \gamma(T) \in Z_{target} \\ & \quad (x(t) \ x_{obs})^\top P^{-1}(t) (x(t) \ x_{obs}) \leq \chi^2, \\ & \quad \forall t \in [0, T], \quad \forall x_{obs} \in X_{obs}^\ell, \quad \forall \ell \in \{1, \dots, M\}. \end{aligned} \quad (9)$$

The χ^2 term in the constraints of (9) is implemented to provide a confidence bound on probability that a robot with position $x(t)$ will not be in contact with an obstacle x_{obs} .

IV. RI-RRT* ALGORITHM

A. RRT*

The RRT algorithm [25] constructs a tree of nodes (state realizations) through random sampling of the feasible state-space and then connects these nodes with edges (tree branches). A user-defined cost is utilized to quantify the length of the edges, which are in turn summed to form path lengths. Each new node is connected via a permanent edge to the existing node which provides the shortest path between the new node and the initial node of the tree. Although the RRT algorithm is known to be probabilistically complete (the algorithm finds a feasible path if one exists), it does not achieve asymptotic optimality (path cost does not converge to the optimal one as the number of nodes is increased) [12].

The RRT* algorithm [12] attains asymptotic optimality by including an additional “re-wiring” step that re-evaluates if the path length for each node can be reduced via a connection to the newly created node. This paper utilizes RRT* as a numerical approach to the shortest path problem (9).

B. Algorithm

Provided below is a Rationally Inattentive RRT* (RI-RRT*) algorithm for finding a solution to (9). Like the original RRT* algorithm, the RI-RRT* algorithm constructs a graph of state nodes and edges $(G = (Z, E))$ in spaces with or without obstacles.

Algorithm 1: RI-RRT* Algorithm

```

1  $(z_1) \leftarrow (z_{init}); E \leftarrow \emptyset; G^0 \leftarrow (z_1; E);$ 
2 for  $i = 2 : N$  do
3    $G \leftarrow G^0;$ 
4    $z_i = (x_i; P_i) \leftarrow \text{GENERATE}(i);$ 
5    $(Z^0; E^0) \leftarrow (Z; E);$ 
6    $z_{near} \leftarrow \text{NEAREST}(Z^0; z_i);$ 
7    $z_{new} \leftarrow \text{SCALE}(z_{near}; z_i; ED_{min});$ 
8   if  $\text{OBSCHECK}(z_{near}; z_{new}) = \text{False}$  then
9      $Z^0 \leftarrow Z^0 \cup z_{new};$ 
10     $Z_{nbors} \leftarrow \text{NEIGHBOR}(Z; z_{new}; ED_{nbors});$ 
11     $Path_{z_{new}} \leftarrow \text{realmax};$ 
12    for  $z_j \in Z_{nbors}$  do
13      if  $\text{OBSCHECK}(z_j; z_{new}) = \text{False}$  then
14         $Path_{z_{new}, j} \leftarrow Path_{z_j} + \mathcal{D}(z_j; z_{new});$ 
15        if  $Path_{z_{new}, j} < Path_{z_{new}}$  then
16           $Path_{z_{new}} \leftarrow Path_{z_{new}, j};$ 
17           $z_{nbor} \leftarrow z_j;$ 
18     $E^0 \leftarrow [z_{nbor}; z_{new}] \cup E^0;$ 
19    for  $z_j \in Z_{nbors} \setminus z_{nbor}$  do
20      if  $\text{OBSCHECK}(z_{new}; z_j) = \text{False}$  then
21         $Path_{z_j, \text{rewire}} = Path_{z_{new}} + \mathcal{D}(z_{new}; z_j);$ 
22        if  $Path_{z_j, \text{rewire}} < Path_{z_j}$  then
23           $E^0 \leftarrow E^0 \cup [z_{new}; z_j] \setminus [z_j, \text{parent}; z_j];$ 
24           $z_{j, \text{parent}} \leftarrow z_{new};$ 
25           $\text{UPDATEDES}(G; z_j);$ 
26    $G^0 \leftarrow (Z^0; E^0)$ 

```

In Algorithm 1, the $\text{GENERATE}(i)$ function creates a new point by randomly sampling a spatial location ($x \in \mathbb{R}^d$) and covariance ($P \in S_{++}^d$). Notice that for a d -dimensional configuration space, the corresponding uncertain configuration space $\mathbb{R}^d \times S_{++}^d$ has $d + \frac{1}{2}d(d+1)$ dimensions from which the samples are generated. The $\text{NEAREST}(Z, z_i)$ function finds the nearest point (z_{near}), in metric $\hat{D}(z, z') := kx - x'k + kP - P'k_F$, between the newly generated state $z_i = (x_i, P_i)$ and an existing state in the set Z . Using the metric $\hat{D}(z, z')$, the $\text{SCALE}(z_{near}, z_i, ED_{min})$ function linearly shifts the generated point (z_i) to a new location as:

$$z_{new} = \begin{cases} z_{near} + \frac{ED_{min}}{\hat{D}(z_i, z_{near})} (z_i - z_{near}) & \text{if } \hat{D}(z, z') > ED_{min}, \\ z_i & \text{otherwise,} \end{cases}$$

where ED_{min} is a user-defined constant. In addition to generating z_{new} , the SCALE function also ensures that its χ^2 covariance region does not interfere with any obstacles.

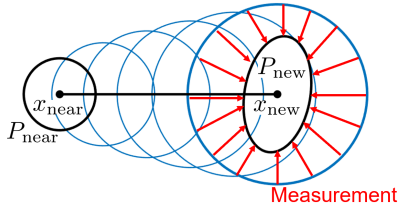


Fig. 2. Transition from state z_{near} to z_{new} following the move-and-sense path plan. The blue ellipses represent the propagation of covariance where the $\text{OBSCHECK}(z_{\text{near}}; z_{\text{new}})$ function in Algorithm 1 checks collisions between these propagated covariances, including z_{near} and z_{new} , and obstacles. The measurement at $z_{\text{new}} = (x_{\text{new}}; P_{\text{new}})$ makes a covariance smaller.

The $\text{OBSCHECK}(z_{\text{near}}, z_{\text{new}})$ function ensures that transition from state z_{near} to z_{new} does not intersect with any obstacle $\ell \in \mathcal{F}_1, \dots, M, g$. More precisely, we assume the transition $z_{\text{near}} \rightarrow z_{\text{new}}$ follows the move-and-sense path, introduced in Section III-C, and the OBSCHECK function returns *False* if all state pairs along the move-and-sense path, shown by blue ellipses in Fig. 2, has χ^2 covariance regions that are non-interfering with obstacles.

The $\text{NEIGHBOR}(Z, z_{\text{new}}, ED_{\text{nbors}})$ function returns the sub-set of nodes described as $Z_{\text{nbors}} = \{z_i = (x_i, P_i) \in Z : \hat{D}(z_i, z_{\text{new}}) \leq ED_{\text{nbors}}\}$. This set is then evaluated for the presence of obstacles via the OBSCHECK function of the previous paragraph. Note that in this instance, the function is evaluating obstacle interference along the continuous path of state-covariance pairs from $z_j \rightarrow z_{\text{new}}$. Lines 14-17 of Algorithm 1 connect the new node to the existing graph in an identical manner to RRT*, where the $Path_z$ denote the cost of the path from the z_{init} to node z through the edges of G . Line 18 creates a new edge between the new node and the existing nodes from the neighbor group Z_{nbors} which results in the minimum $Path_{z_{\text{new}}}$. The calculation of RI path cost in Line 14 utilizes (8).

Lines 19-24 are the tree re-wiring steps of Algorithm 1. In line 20, the OBSCHECK function is called again. This is because the move-and-sense path is direction-dependent, and thus $\text{OBSCHECK}(z_j, z_{\text{new}}) = \text{False}$ does not necessarily imply $\text{OBSCHECK}(z_{\text{new}}, z_j) = \text{False}$. Finally, for each rewired node z_j , its cost (i.e., $Path_{z_j}$) and the cost of its descendants are updated via $\text{UPDATEDES}(G, z_j)$ function in line 25.

To increase the computational efficiency of RI-RRT* algorithm we deploy a branch-and-bound technique as detailed in [26]. For a given tree G , let z_{min} be the node that has the lowest cost along the nodes of G within Z_{target} . As discussed in Section III-C, $D(z, z_{\text{goal}})$ is a lower-bound for the cost of transitioning from z to z_{goal} . The branch-and-bound algorithm periodically deletes the nodes $Z'' = \{z \in Z : Path_z + D(z, z_{\text{goal}}) > Path_{z_{\text{min}}}\}$. This elimination of the non-optimal nodes speeds up the RI-RRT* algorithm.

C. Properties of RI-RRT*

The question regarding the asymptotic optimality of RI-RRT* naturally arises. Recall that the proof of the asymptotic optimality of the RRT* algorithm [27] is founded on four

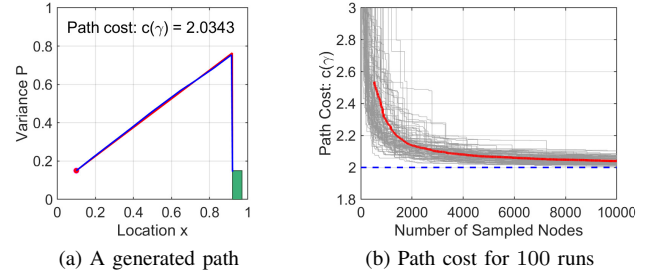


Fig. 3. Results of the RI-RRT* algorithm with $\alpha = 1$ and $W = 0.75$ applied to one-dimensional joint movement-perception problem. (a): The blue line illustrates the path generated with 10,000 nodes, which almost converges to the known optimal path depicted as the red curve. (b): The total path cost for 100 runs of 10,000 nodes. Gray lines plot the path cost for each run, while the average of them is shown in the red line. The path costs of all 100 runs approach the optimal path cost (dashed blue line).

main assumptions:

- 1) additivity of the cost function,
- 2) the cost function is monotonic,
- 3) there exists a finite distance between all points on the optimal path and the obstacle space,
- 4) the cost function is continuous, either in the topology of total variation metric [27] or the supremum norm metric [12].

The proofs of the first three assumptions are trivial for the RI cost (8). However, Theorem 1 does not suffice to guarantee that the RI cost meets the fourth condition for $d = 2$. For this reason, currently the asymptotic optimality of the RI-RRT* algorithm cannot be guaranteed, while the numerical simulations of Section V do show that the proposed algorithm does have merit in rationally inattentive path-planning.

V. SIMULATION RESULTS

A. One-Dimensional Simulation

The first study is the case of a robot which is allowed to travel at a constant velocity in a one-dimensional geometric space from a predetermined initial position and covariance $z_0 = (x_0, P_0)$, specified by the red dot in Fig. 3 (a). The robot has a goal of reaching some final state within the green box representing a goal region which is a sub-set of the reachable space. Note that the goal region contains acceptable bounds on both location and uncertainty. Although, in one-dimensional setting, the strategy which minimizes the control cost is obviously the one that moves directly toward the target region, we utilize the RI-RRT* algorithm to solve the non-trivial measurement scheduling problem.

In Fig. 3 (a), the blue curve represents the path generated by the RI-RRT* algorithm with 10,000 nodes, which is sufficiently close to the shortest path obtainable via the RI-distance depicted as the red curve. These wedge-shaped optimal paths are created by the “move-and-sense” strategy integrated in the RI-cost, where it has a section of covariance propagation followed by an instantaneous reduction of covariance as discussed in Section III-C. For example, if the robot were an autonomous ground vehicle with GPS capabilities, then this path signifies the robot driving the total

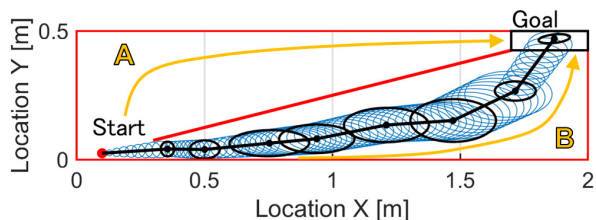


Fig. 4. Results of the RI-RRT* algorithm with 10,000 nodes in the two-dimensional space containing roughly two paths, A and B, separated by a diagonal wall. The black line is the shortest path with the associated covariance ellipses. The blue ellipses illustrate the propagation of covariance between nodes. The simulation was completed with $W = 10^{-3}$ and 2 covariance ellipses representing 90% certainty regions. The boundaries of the plots are considered as obstacles.

distance without any GPS updates, followed by a reduction its spatial uncertainty with a single update once the goal region is reached. The minimum path cost at the end of each iteration of the *for-loop* in Algorithm 1 is depicted in Fig. 3 (b), where the red curve represents the average of 100 independent simulations. The path cost of each simulation approaches to the optimal cost.

B. Two-Dimensional Asymmetric Simulation

The asymmetric characteristic of the RI-cost is demonstrated via a simulation in the two-dimensional configuration space with a diagonal wall, as seen in Fig. 4. The initial position and covariance of the robot is depicted as a red dot, while the target region is illustrated as the black rectangle at the upper-right corner.

The path is generated by the RI-RRT* algorithm by sampling 10,000 nodes. The corresponding sampled covariance ellipses are shown in black where the blue ellipses represent covariance propagation. As shown in Fig. 4, there are two options; path A requires the robot to move into a funnel-shaped corridor, while the path B moves out of a funnel. In this setting, the RI-cost prefers the path B even though both A and B have the same Euclidean distance. This asymmetric behavior results from the fact that as the robot approaches the goal region path B requires a less severe uncertainty reduction compared to path A. Similarly, by exchanging the start and goal positions, the RI-cost prefers path A over path B, thus displaying the directional dependency of our efficient sensing strategy.

C. Two-Dimensional Simulation with Multiple Obstacles

In a final demonstration, the RI-RRT* algorithms with $\alpha = 0, 0.1, \text{ and } 0.3$ are implemented in a two-dimensional configuration space containing multiple obstacles in order to illustrate the effects of varying the information cost. All three paths in panels of Fig. 5 are generated from 4,000 nodes.

As seen in Fig. 5 (a), when $\alpha = 0$ the algorithm simply finds a path that has the shortest Euclidean distance, even if that path requires frequent sensing actions. In contrast, the RI-RRT* algorithm with $\alpha = 0.3$ does not take a constrained pathway, and thus requires fewer sensor actuations in order to avoid obstacle collisions. As a result, the algorithm deviates

from the shortest Euclidean distance path and allows the covariance to propagate safely, as seen in Fig. 5 (c). A moderate path, illustrated in Fig. 5 (b), can be also obtained by choosing $\alpha = 0.1$.

VI. CONCLUSION

In this work, a novel RI cost for utilization in path-planning algorithms is presented. The cost accounts for both the path distance traversed (efficiency) and the amount of information which must be perceived by the robot. Information gained from perception is important in that it allows the robot to safely navigate obstacle-filled environments with confidence that collisions will be avoided. This method, in balancing path distance and perception costs, provides a simplicity-based path which can be tailored to mimic the results potentially generated by an expert human path-planner. Three numerical simulations were provided demonstrating these results.

Currently, the preliminary version of the RI-RRT* algorithm is optimized for computational efficiency by the aid of a *branch-and-bound* technique. The authors note that utilizing other RRT* improvement methods, such as *k-d trees* could further improve the computational speed of Algorithm 1 and should be considered in future work. In the same vein of future work, the authors note the importance of quantifying the impact that the user-defined constants, such as the distances which signify which nodes are neighbors, have on the results of the RI-RRT* algorithm. Also, the topic of path refinement should be further explored as RRT*-like algorithms converge asymptotically.

It should be noted that once the RI-RRT* algorithm finds an initial feasible path, there exist iterative methods for path “smoothing” which do not require additional node sampling. In a theorized hybrid method, RI-RRT* is first utilized to find some initial path and then an iterative method takes the initial path and “smooths” towards the optimal path. Convergence benefits of iterative methods are often improved as the initial path guess is more similar to the optimal path, and a trade-off could be found in computational efficiency which results in the best time to switch between algorithms.

APPENDIX I PROOF OF LEMMA 1

For every $t \in [0, T]$, set a partition $P = (0, t, T)$. Then

$$\begin{aligned}
 kx(t)k + \sigma(P(t)) &= kx(0)k + \sigma(P(0)) \\
 &\quad + kx(t) \quad x(0)k + \sigma(P(t) \quad P(0)) \\
 kx(0)k + \sigma(P(0)) &= kx(0)k + \sigma(P(0)) \\
 &\quad + kx(t) \quad x(0)k + \sigma(P(t) \quad P(0)) \\
 &\quad + kx(T) \quad x(t)k + \sigma(P(T) \quad P(t)) \\
 V(\gamma, P) &= \int_{\gamma} j_{TV}.
 \end{aligned}$$

APPENDIX II PROOF OF THEOREM 1

The proof is based on the following lemma:

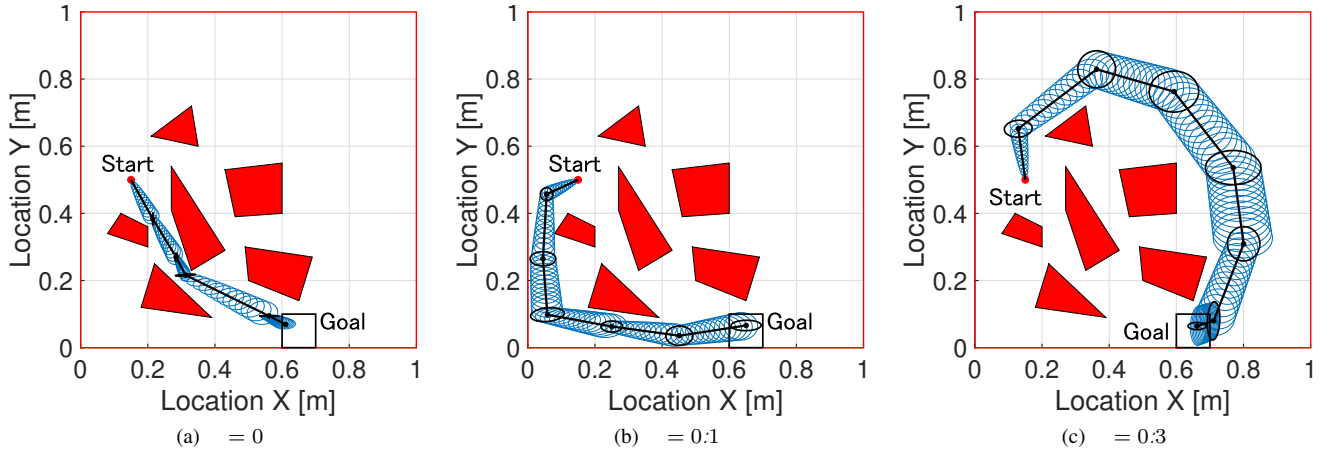


Fig. 5. Results of simulation with $\alpha = 0; 0.1; 0.3$ under the existence of multiple obstacles. The simulation was completed with $W = 10^{-3} J$ and covariance ellipses representing 90% certainty regions. The boundaries of the plots are considered as obstacles.

Lemma 2: Assume $d = 1$. For each (ϵ, δ) satisfying $0 < \delta \leq \frac{\epsilon}{2}$, there exists a constant L_ϵ such that the inequality:

$$jD(x'_k, x'_{k+1}, P'_k, P'_{k+1}) - D(x_k, x_{k+1}, P_k, P_{k+1})j \leq L_\epsilon \left[|(x'_{k+1} \ x_{k+1}) - (x'_k \ x_k)| + |(P'_{k+1} \ P_{k+1}) - (P'_k \ P_k)| + \delta |P_{k+1} - P_k| + \delta |x_{k+1} - x_k| \right]$$

holds for all

$$x'_k, x'_{k+1}, x_k, x_{k+1} \in \mathbb{R} \text{ and } P'_k, P'_{k+1}, P_k, P_{k+1} \in \mathbb{P}$$

such that

$$x'_k := x'_k - x_k \leq \delta, \quad x'_{k+1} := x'_{k+1} - x_{k+1} \leq \delta \\ P'_k := P'_k - P_k \leq \delta, \quad P'_{k+1} := P'_{k+1} - P_{k+1} \leq \delta.$$

Proof: For simplicity, we assume $\alpha = W = 1$, but the extension of the following proof to general cases is straightforward. In what follows, we write

$$D_{\text{info}}(x_k, x_{k+1}, P_k, P_{k+1}) := \max \left\{ 0, \frac{1}{2} \log_2(P_k + |x_{k+1} - x_k|), \frac{1}{2} \log_2(P_{k+1}) \right\}.$$

We consider four different cases depending on the signs of $D_{\text{info}}(x'_k, x'_{k+1}, P'_k, P'_{k+1})$ and $D_{\text{info}}(x_k, x_{k+1}, P_k, P_{k+1})$.

Case 1: First, we consider the case with

$$D_{\text{info}}(x'_k, x'_{k+1}, P'_k, P'_{k+1}) > 0 \text{ and } D_{\text{info}}(x_k, x_{k+1}, P_k, P_{k+1}) > 0.$$

In this case:

$$\begin{aligned} & \left| D(x'_k, x'_{k+1}, P'_k, P'_{k+1}) - D(x_k, x_{k+1}, P_k, P_{k+1}) \right| \\ &= \left| |x_{k+1} + \Delta x_{k+1} - x_k - \Delta x_k| - |x_{k+1} - x_k| \right. \\ & \quad + \frac{1}{2} \log_2(P_k + \Delta P_k + |x_{k+1} + \Delta x_{k+1} - x_k - \Delta x_k|) \\ & \quad - \frac{1}{2} \log_2(P_{k+1} + \Delta P_{k+1}) - \frac{1}{2} \log_2(P_k + |x_{k+1} - x_k|) \\ & \quad \left. + \frac{1}{2} \log_2(P_{k+1}) \right| \\ &\leq \left| |x_{k+1} + \Delta x_{k+1} - x_k - \Delta x_k| - |x_{k+1} - x_k| \right| \\ & \quad + \frac{1}{2} \left| \log_2 \left(1 + \frac{\Delta P_k}{P_k} + \frac{|x_{k+1} + \Delta x_{k+1} - x_k - \Delta x_k|}{P_k} \right) \right. \\ & \quad \left. - \log_2 \left(1 + \frac{|x_{k+1} - x_k|}{P_k} + \frac{P_k + |x_{k+1} - x_k|}{P_k P_{k+1}} \Delta P_{k+1} \right) \right|. \end{aligned} \quad (10)$$

Using the fact that $|ja + bj - ja - bj| \leq |ja| + |bj|$ for $a, b \in \mathbb{R}$, we have:

$$\left| \frac{j|x_{k+1} + \Delta x_{k+1} - x_k - \Delta x_k|}{P_k} - \frac{j|x_{k+1} - x_k|}{P_k} \right| \leq \frac{j|x_{k+1} + \Delta x_{k+1} - x_k - \Delta x_k|}{P_k} + \frac{j|x_{k+1} - x_k|}{P_k}$$

Noticing that the arguments in the logarithmic terms in (10) are $\frac{1}{2}$, and using the fact that $|\log_2(a) - \log_2(b)| \leq \frac{|a-b|}{\min(a,b)}$

$2 \log_2(e)ja \quad bj, \delta a, b \quad \frac{1}{2}$, we have:

$$\begin{aligned}
& \frac{1}{2} \left| \log_2 \left(1 + \frac{P_k}{P_k} + \frac{jx_{k+1} + x_{k+1} \quad x_k \quad x_{kj}}{P_k} \right) \right. \\
& \quad \left. \log_2 \left(1 + \frac{jx_{k+1} \quad x_{kj}}{P_k} + \frac{P_k + jx_{k+1} \quad x_{kj}}{P_k P_{k+1}} \quad P_{k+1} \right) \right| \\
& \log_2(e) \left| \frac{jx_{k+1} + x_{k+1} \quad x_k \quad x_{kj}}{P_k} \quad \frac{jx_{k+1} \quad x_{kj}}{P_k} \right. \\
& \quad \left. + \frac{P_k}{P_k} \quad \frac{P_k + jx_{k+1} \quad x_{kj}}{P_k P_{k+1}} \quad P_{k+1} \right| \\
& \frac{\log_2(e)}{P_k} \left| jx_{k+1} + x_{k+1} \quad x_k \quad x_{kj} \quad jx_{k+1} \quad x_{kj} \right| \\
& + \log_2(e) \left| \frac{P_k \quad P_{k+1}}{P_k} \right. \\
& \quad \left. + \frac{P_{k+1} \quad P_k \quad jx_{k+1} \quad x_{kj}}{P_k P_{k+1}} \quad P_{k+1} \right| \\
& \frac{\log_2(e)}{P_k} j \quad x_{k+1} \quad x_{kj} + \frac{\log_2(e)}{P_k} j \quad P_{k+1} \quad P_{kj} \\
& + \frac{\log_2(e) j \quad P_{k+1} j}{P_k P_{k+1}} |P_{k+1} \quad P_k \quad jx_{k+1} \quad x_{kj}| \\
& \frac{\log_2(e)}{P_k} j \quad x_{k+1} \quad x_{kj} + \frac{\log_2(e)}{P_k} j \quad P_{k+1} \quad P_{kj} \\
& + \frac{\log_2(e) j \quad P_{k+1} j}{P_k P_{k+1}} jx_{k+1} \quad x_{kj} \\
& + \frac{\log_2(e) j \quad P_{k+1} j}{P_k P_{k+1}} jP_{k+1} \quad P_{kj} \\
& \frac{\log_2(e)}{\epsilon} j \quad x_{k+1} \quad x_{kj} + \frac{\log_2(e)}{\epsilon} j \quad P_{k+1} \quad P_{kj} \\
& + \frac{\log_2(e) \delta}{\epsilon^2} jx_{k+1} \quad x_{kj} + \frac{\log_2(e) \delta}{\epsilon^2} jP_{k+1} \quad P_{kj}
\end{aligned}$$

Therefore,

$$\begin{aligned}
& jD(x'_k, x'_{k+1}, P'_k, P'_{k+1}) \quad D(x_k, x_{k+1}, P_k, P_{k+1})j \\
& \left(1 + \frac{\log_2(e)}{\epsilon} \right) j \quad x_{k+1} \quad x_{kj} \\
& + \frac{\log_2(e)}{\epsilon} j \quad P_{k+1} \quad P_{kj} \\
& + \frac{\log_2(e) \delta}{\epsilon^2} jx_{k+1} \quad x_{kj} + \frac{\log_2(e) \delta}{\epsilon^2} jP_{k+1} \quad P_{kj}
\end{aligned} \tag{11}$$

Case 2: Next, we consider the case with

$$D_{\text{info}}(x'_k, x'_{k+1}, P'_k, P'_{k+1}) > 0 \text{ and} \tag{12a}$$

$$D_{\text{info}}(x_k, x_{k+1}, P_k, P_{k+1}) = 0. \tag{12b}$$

Notice that (12b) implies $P_k + jx_{k+1} \quad x_{kj} \quad P_{k+1} \quad 0$. In

this case:

$$\begin{aligned}
& jD(x'_k, x'_{k+1}, P'_k, P'_{k+1}) \quad D(x_k, x_{k+1}, P_k, P_{k+1})j \\
& = \left| jx_{k+1} + x_{k+1} \quad x_k \quad x_{kj} \quad jx_{k+1} \quad x_{kj} \right. \\
& \quad \left. + \frac{1}{2} \log_2(P_k + P_k + jx_{k+1} + x_{k+1} \quad x_k \quad x_{kj}) \right. \\
& \quad \left. \frac{1}{2} \log_2(P_{k+1} + P_{k+1}) \right|
\end{aligned} \tag{13a}$$

$$\begin{aligned}
& j \quad x_{k+1} \quad x_{kj} \quad \frac{1}{2} \log_2(P_{k+1} + P_{k+1}) \\
& + \frac{1}{2} \log_2(P_k + P_k + jx_{k+1} + x_{k+1} \quad x_k \quad x_{kj})
\end{aligned} \tag{13b}$$

$$\begin{aligned}
& j \quad x_{k+1} \quad x_{kj} \\
& + \frac{\log_2(e)}{\epsilon} (P_k + P_k + jx_{k+1} + x_{k+1} \quad x_k \quad x_{kj} \\
& \quad P_{k+1} \quad P_{k+1})
\end{aligned} \tag{13c}$$

$$\begin{aligned}
& = j \quad x_{k+1} \quad x_{kj} \\
& + \frac{\log_2(e)}{\epsilon} (P_k + jx_{k+1} \quad x_{kj} \quad P_{k+1} + P_k \quad P_{k+1} \\
& + jx_{k+1} + x_{k+1} \quad x_k \quad x_{kj} \quad jx_{k+1} \quad x_{kj})
\end{aligned} \tag{13d}$$

$$\begin{aligned}
& j \quad x_{k+1} \quad x_{kj} \\
& + \frac{\log_2(e)}{\epsilon} \left| jx_{k+1} + x_{k+1} \quad x_k \quad x_{kj} \right.
\end{aligned} \tag{13e}$$

$$\begin{aligned}
& \quad \left. jx_{k+1} \quad x_{kj} \right| \\
& + \frac{\log_2(e)}{\epsilon} j \quad P_{k+1} \quad P_{kj}
\end{aligned} \tag{13f}$$

$$\begin{aligned}
& \left(1 + \frac{\log_2(e)}{\epsilon} \right) j \quad x_{k+1} \quad x_{kj} \\
& + \frac{\log_2(e)}{\epsilon} j \quad P_{k+1} \quad P_{kj}
\end{aligned} \tag{13g}$$

In step (13b), we have used the fact that the difference between the two logarithmic terms is positive, because of the hypothesis (12a). In step (13c), we used the fact that $\log_2 a \quad \log_2 b \quad \frac{2 \log_2(e)}{\epsilon} (a \quad b)$, for $a > b \quad \frac{\epsilon}{2}$.

Case 3: Next, we consider the case with

$$D_{\text{info}}(x'_k, x'_{k+1}, P'_k, P'_{k+1}) = 0 \text{ and} \tag{14a}$$

$$D_{\text{info}}(x_k, x_{k+1}, P_k, P_{k+1}) > 0. \tag{14b}$$

The first hypothesis (14a) implies:

$$\begin{aligned}
& P_k + P_k + jx_{k+1} + x_{k+1} \quad x_k \quad x_{kj} \\
& \quad P_{k+1} \quad P_{k+1} \quad 0.
\end{aligned} \tag{15}$$

Using

$$\begin{aligned}
& jx_{k+1} \quad x_{kj} \quad j \quad x_{k+1} \quad x_{kj} \\
& \quad jx_{k+1} + x_{k+1} \quad x_k \quad x_{kj},
\end{aligned}$$

one can deduce from (10) that:

$$\begin{aligned}
& P_k + jx_{k+1} \quad x_{kj} \quad P_{k+1} \\
& \quad P_{k+1} \quad P_k + j \quad x_{k+1} \quad x_{kj} \\
& \quad j \quad P_{k+1} \quad P_{kj} + j \quad x_{k+1} \quad x_{kj}.
\end{aligned} \tag{16}$$

This results in:

$$jD(x'_k, x'_{k+1}, P'_k, P'_{k+1}) \quad D(x_k, x_{k+1}, P_k, P_{k+1})j$$

$$= \left| \begin{array}{cc} jx_{k+1} & x_{k+1} & x_k & x_kj \\ jx_{k+1} & x_{k+1} & x_k & x_kj \end{array} \right| \quad (17a)$$

$$+ \frac{1}{2} \log_2(P_k + jx_{k+1} \quad x_kj) + \frac{1}{2} \log_2(P_{k+1}) \quad (17b)$$

$$+ \frac{\log_2(e)}{2\epsilon} (P_k + jx_{k+1} \quad x_kj) \quad P_{k+1} \quad (17c)$$

$$\left(1 + \frac{\log_2(e)}{2\epsilon}\right) j \quad x_{k+1} \quad x_kj$$

$$+ \frac{\log_2(e)}{2\epsilon} j \quad P_{k+1} \quad P_kj \quad (17d)$$

In (17b) we used the fact that the difference between the two logarithmic terms is positive. In (17c) we used the fact that $\log_2 a - \log_2 b = \frac{\log_2(e)}{\epsilon} (a - b)$, for $a > b$. Finally, the inequality (16) was used in step (17d).

Case 4: Finally, we consider the case with

$$D_{\text{info}}(x'_k, x'_{k+1}, P'_k, P'_{k+1}) = 0 \text{ and}$$

$$D_{\text{info}}(x_k, x_{k+1}, P_k, P_{k+1}) = 0.$$

In this case:

$$jD(x'_k, x'_{k+1}, P'_k, P'_{k+1}) \quad D(x_k, x_{k+1}, P_k, P_{k+1})j$$

$$= \left| \begin{array}{cc} jx_{k+1} & x_{k+1} & x_k & x_kj \\ jx_{k+1} & x_{k+1} & x_k & x_kj \end{array} \right| \quad (18)$$

To summarize, (11), (13g), (17d), and (18) are sufficient to be able to choose $L_\epsilon = \max\{1, \frac{\log_2(e)}{\epsilon}, \frac{\log_2(e)}{\epsilon^2}g\}$ to obtain the desired result. ■

Proof of Theorem 1

Suppose $\gamma(t) = (x(t), P(t))$ and $\gamma'(t) = (x'(t), P'(t))$. In what follows, we consider the choice:

$$\delta = \min \left\{ \frac{\epsilon_0}{2L_\epsilon(1 + j\gamma_{\text{TV}})}, \frac{\epsilon}{2} \right\} \quad (19)$$

Since $j\gamma' - \gamma_{\text{TV}} < \delta$, we have $k\gamma' - \gamma_{\text{TV}} < \delta$. In particular, for each $t \geq [0, T]$, we have $jx'(t) - x(t)j < \delta$ and $jP'(t) - P(t)j < \delta$. Moreover:

$$jP'(t)j = jP(t) + P'(t) - P(t)j$$

$$jP(t)j - jP'(t) - P(t)j > 2\epsilon - \delta > \epsilon.$$

Therefore, $\forall t \geq [0, T]$, we have both $P(t) > \epsilon$ and $P'(t) > \epsilon$. Let $P = (0 = t_0 < t_1 < \dots < t_N = T)$ be a partition and define:

$$c(\gamma; P) := \sum_{k=0}^{N-1} D(x(t_k), x(t_{k+1}), P(t_k), P(t_{k+1})).$$

For any partition P , the following chain of inequalities holds:

$$jc(\gamma'; P) \leq c(\gamma; P)j$$

$$= \left| \sum_{k=0}^{N-1} D(x'(t_k), x'(t_{k+1}), P'(t_k), P'(t_{k+1})) \right.$$

$$\left. D(x(t_k), x(t_{k+1}), P(t_k), P(t_{k+1})) \right| \quad (20a)$$

$$\sum_{k=0}^{N-1} jD(x'(t_k), x'(t_{k+1}), P'(t_k), P'(t_{k+1}))$$

$$D(x(t_k), x(t_{k+1}), P(t_k), P(t_{k+1}))j \quad (20b)$$

$$L_\epsilon \sum_{k=0}^{N-1} \left[j(x'(t_{k+1}) \quad x(t_{k+1})) \quad (x'(t_k) \quad x(t_k))j \right.$$

$$\left. + j(P'(t_{k+1}) \quad P(t_{k+1})) \quad (P'(t_k) \quad P(t_k))j \right.$$

$$\left. + \delta jP(t_{k+1}) \quad P(t_k)j + \delta jx(t_{k+1}) \quad x(t_k)j \right] \quad (20c)$$

$$= L_\epsilon (V(\gamma' - \gamma, P) + \delta V(\gamma, P)) \quad (20d)$$

$$L_\epsilon (j\gamma' - \gamma_{\text{TV}} + \delta j\gamma_{\text{TV}}) \quad (20e)$$

$$< L_\epsilon (\delta + \delta j\gamma_{\text{TV}}) \quad (20f)$$

$$L_\epsilon (1 + j\gamma_{\text{TV}}) \left(\frac{\epsilon_0}{2L_\epsilon(1 + j\gamma_{\text{TV}})} \right) \quad (20g)$$

$$= \frac{\epsilon_0}{2} \quad (20h)$$

The inequality (20c) follows from Lemma 2. Let $fP_i, g_i \in \mathbb{N}$ and $fP'_i, g'_i \in \mathbb{N}$ be sequences of partitions such that:

$$\lim_{i \rightarrow \infty} c(\gamma; P_i) = c(\gamma), \quad \lim_{i \rightarrow \infty} c(\gamma'; P'_i) = c(\gamma'), \quad (21)$$

and let $fP''_i, g''_i \in \mathbb{N}$ be the sequence of partitions such that for each $i \geq 2$, P''_i is a common refinement of P_i and P'_i . Since both

$$c(\gamma; P_i) \leq c(\gamma; P''_i) \leq c(\gamma) \text{ and } c(\gamma'; P'_i) \leq c(\gamma'; P''_i) \leq c(\gamma')$$

hold for each $i \geq 2$, (21) implies

$$\lim_{i \rightarrow \infty} c(\gamma; P''_i) = c(\gamma), \quad \lim_{i \rightarrow \infty} c(\gamma'; P''_i) = c(\gamma'). \quad (22)$$

Now, since the chain of inequalities (20) holds for any partitions,

$$jc(\gamma; P''_i) \leq c(\gamma'; P''_i)j < \frac{\epsilon_0}{2}$$

holds for all $i \geq 2$. This results in:

$$jc(\gamma) \leq c(\gamma')j = \lim_{i \rightarrow \infty} jc(\gamma; P''_i) \leq c(\gamma'; P''_i)j$$

$$\leq \frac{\epsilon_0}{2} < \epsilon_0. \quad (23)$$

where (23) follows from (22).

APPENDIX III

ONE-DIMENSIONAL PROBLEM OPTIMAL PATH

Consider taking the single perception optimal path γ_1 in Fig. 3 (a):

$$(x_0, P_0) \rightarrow (x_T, P_T)$$

and dividing it into the combination of two sub-paths γ_2 :

$$(x_0, P_0) \rightarrow (x_a, P_a) \rightarrow (x_T, P_T)$$

such that $\hat{P}'_0 = P_0 + \beta kx_T \quad x_0kW > P_a$,

$$\hat{P}'_a = P_a + (1 - \beta)kx_T \quad x_0kW > P_T$$

where $\beta \in (0, 1)$ is a constant which denotes where in γ_2 the additional sensing action takes place. The combination of the divided sub-paths have in the same initial (z_0) and ending (z_T) states as the original path, but also achieve an intermediate state (z_a).

Path γ_1 has a total RI cost:

$$D(\gamma_1) = kx_T \quad x_0k + \frac{\alpha}{2} \left[\log_2 \hat{P}_0 \quad \log_2 P_T \right],$$

where, $\hat{P}_0 = P_0 + kx_T \quad x_0kW$. Likewise, the path γ_2 has a length which is the summation of two information gains and while transitioning the same distance as γ_1 .

$$D(\gamma_2) = kx_T \quad x_0k + \frac{\alpha}{2} \left[\log_2 \hat{P}'_0 \quad \log_2 P_a \right]$$

$$+ \frac{\alpha}{2} \left[\log_2 \hat{P}'_a \quad \log_2 P_T \right]$$

By comparing the costs between γ_1 and γ_2 , it is possible to achieve:

$$D(\gamma_1) - D(\gamma_2) = \frac{\alpha}{2} \left[\log_2 \hat{P}_0 \quad \log_2 P_T \right]$$

$$- \frac{\alpha}{2} \left[\log_2 \hat{P}'_0 \quad \log_2 P_a \right] - \frac{\alpha}{2} \left[\log_2 \hat{P}'_a \quad \log_2 P_T \right]$$

$$= \frac{\alpha}{2} \left[\log_2 \hat{P}_0 \quad \log_2 \hat{P}'_0 \right] - \frac{\alpha}{2} \left[\log_2 \hat{P}'_a \quad \log_2 P_a \right]$$

$$= f(\hat{P}'_0) - f(P_a) < 0,$$

where $f(P) = \frac{\alpha}{2} \left[\log_2(P + (1 - \beta)kx_T \quad x_0kW) \quad \log_2 P \right]$. Last inequality follows the facts that $\hat{P}'_0 > P_a$ and $f(P)$ is a decreasing function ($\frac{df}{dP} < 0$).

REFERENCES

- [1] S. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. Eng, D. Rus, and M. Ang, "Perception, planning, control, and coordination for autonomous vehicles," *Machines*, vol. 5, no. 1, p. 6, 2017.
- [2] M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, and C. Cadena, "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 1527–1533.
- [3] L. Carlone and S. Karaman, "Attention and anticipation in fast visual-inertial navigation," *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 1–20, 2019.
- [4] R. Alterovitz, S. Koenig, and M. Likhachev, "Robot planning in the real world: research challenges and opportunities," *AI Magazine*, vol. 37, no. 2, pp. 76–84, 2016.
- [5] Y. Kuwata, J. Teo, S. Karaman, G. Fiore, E. Frazzoli, and J. How, "Motion planning in complex environments using closed-loop prediction," *AIAA Guid. Navig. Control Conf. Exhibit*, 2008.
- [6] 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.
- [7] A.-A. Agha-Mohammadi, S. Chakravorty, and N. M. Amato, "FIRM: Sampling-based feedback motion-planning under motion uncertainty and imperfect measurements," *Int. J. Robot. Res.*, vol. 33, no. 2, pp. 268–304, 2014.
- [8] C. A. Sims, "Implications of rational inattention," *J. Monetary Economics*, vol. 50, no. 3, pp. 665–690, 2003.
- [9] E. Shafieepoorfard, M. Raginsky, and S. P. Meyn, "Rationally inattentive control of markov processes," *SIAM J. Control Optimization*, vol. 54, no. 2, pp. 987–1016, 2016.

- [10] E. Shafieepoorfard and M. Raginsky, "Rational inattention in scalar lqg control," in *Proc. Conf. Decision Control*, 2013, pp. 5733–5739.
- [11] S. M. LaValle, *Planning algorithms*. Cambridge University Press, 2006.
- [12] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," in *Proc. Robot.: Sci. Syst.*, 2010.
- [13] J. J. Marquez and M. L. Cummings, "Design and evaluation of path planning decision support for planetary surface exploration," *J. Aerosp. Comput., Info., Comm.*, vol. 5, no. 3, pp. 57–71, 2008.
- [14] Y. K. Hwang and N. Ahuja, "A potential field approach to path planning," *IEEE Trans. Robot. Autom.*, vol. 8, no. 1, pp. 23–32, 1992.
- [15] S. Kambhampati and L. Davis, "Multiresolution path planning for mobile robots," *IEEE J. Robot. Autom.*, vol. 2, no. 3, pp. 135–145, 1986.
- [16] F. Hauer, A. Kundu, J. M. Rehg, and P. Tsotras, "Multi-scale perception and path planning on probabilistic obstacle maps," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 4210–4215.
- [17] A. Lambert and D. Gruyer, "Safe path planning in an uncertain-configuration space," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 3, 2003, pp. 4185–4190.
- [18] 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.
- [19] P. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*. Berlin: Springer, 1992.
- [20] A. Lambert and D. Gruyer, "Safe path planning in an uncertain-configuration space," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 3, 2003, pp. 4185–4190.
- [21] 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.
- [22] T. Tanaka, K.-K. Kim, P. A. Parrilo, and S. K. Mitter, "Semidefinite programming approach to Gaussian sequential rate-distortion trade-offs," *IEEE Trans. Automat. Control*, vol. 62, no. 4, pp. 1896–1910, 2016.
- [23] L. Vandenberghe, S. Boyd, and S.-P. Wu, "Determinant maximization with linear matrix inequality constraints," *SIAM J. Matrix Analysis and Applications*, vol. 19, no. 2, pp. 499–533, 1998.
- [24] N. L. Carothers, *Real analysis*. Cambridge University Press, 2000.
- [25] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, May 2001.
- [26] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the RRT," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 1478–1483.
- [27] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.