Asymptotically optimal inspection planning using systems with differential constraints

Citation: Papadopoulos, Georgios, Hanna Kurniawati, and Nicholas M. Patrikalakis. "Asymptotically Optimal Inspection Planning Using Systems with Differential Constraints." 2013 IEEE International Conference on Robotics and Automation (May 2013).

As Published: http://dx.doi.org/10.1109/ICRA.2013.6631159

Publisher: Institute of Electrical and Electronics Engineers (IEEE)

Persistent URL: http://hdl.handle.net/1721.1/97701

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Asymptotically Optimal Inspection Planning using Systems with Differential Constraints

Georgios Papadopoulos*, Hanna Kurniawati†, and Nicholas M. Patrikalakis*
* Department of Mechanical Engineering, Massachusetts Institute of Technology
† School of Information Technology and Electrical Engineering, University of Queensland
Email: gpapado@mit.edu, hannakur@uq.edu.au, nmp@mit.edu

***Abstract*— This paper proposes a new inspection planning algorithm, called Random Inspection Tree Algorithm (RITA). Given a perfect model of a structure, sensor specifications, robot's dynamics, and an initial configuration of a robot, RITA computes the optimal inspection trajectory that observes all points on the structure. Many inspection planning algorithms have been proposed, most of them consist of two sequential steps. In the first step, they compute a small set of observation points such that each point on the structure is visible. In the second step, they compute the shortest trajectory to visit all observation points at least once. The robot's kinematic and dynamic constraints are taken into account only in the second step. Thus, when the robot has differential constraints and operates in cluttered environments, the observation points may be difficult or even infeasible to reach. To alleviate this difficulty, RITA computes both observation points and the trajectory to visit the observation points simultaneously. RITA uses sampling-based techniques to find admissible trajectories with decreasing cost. Simulation results for 2-D environments are promising. Furthermore, we present analysis on the probabilistic completeness and asymptotic optimality of our algorithm.**

I. Introduction

Efficient methods for structural inspection have become increasingly important to ensure safety. Higher shipping traffic and increased terrorist threats demand faster and more frequent ship and harbor inspection. Likewise, greater demands on land infrastructure create the need for faster bridge and road network inspection. The need for faster and more frequent structural inspection has increased the demand for structural inspection using autonomous robots.

An efficient inspection planning algorithm is critical for developing such autonomous robots. Inspection planning is the problem of finding an inspection trajectory, i.e., a trajectory for the robot to scan all points on the structures. In this paper, we focus on the inspection planning problem where the model of the structure and the environment are perfectly known prior to planning, and the robot has no control nor sensing error.

Although many inspection planning methods have been proposed, most are not suitable for robots with differential constraints. Most methods [6], [7], [8], [9] separate the inspection planning problem into two NP-hard problems and then solve each sub-problem sequentially. The first sub-problem is the art gallery problem, which finds the smallest set of observation points that the robot needs to visit to guarantee 100% coverage. The second sub-problem is the traveling salesman problem (TSP), which finds the shortest trajectory to visit all observation points that have been generated by the art gallery solver. This separation approach is difficult to apply when the robot is non-holonomic and operates in cluttered environments, as the observation points generated by the art gallery problem may not be reachable from other observation points in the set. Furthermore, even if we generate the smallest set of observation points and even if we find the shortest trajectory to visit all points in the set of observation points, the generated inspection trajectory may not be the shortest trajectory for scanning all points on the boundary of the structure. In short, most existing inspection planning methods are often not suitable to find a short inspection trajectory for robots with non-holonomic constraints.

This paper proposes a new inspection planning algorithm, called Random Inspection Tree Algorithm (RITA), that *does not* separate the problem into the art-gallery problem and the TSP. RITA uses a sampling-based motion planner to directly find the shortest valid inspection trajectory, where all points on the structure are seen by at least one point in the trajectory. To easily account for differential constraints, RITA performs its sampling in the control space. It finds inspection trajectories with decreasing cost, and in the limit, converges to the optimal inspection trajectory with probability one.

This paper is organized as follows: In the next section, we discuss related work. In Section III, we formally define the inspection planning problem. In Section IV we describe RITA. We discuss its probabilistic completeness and its convergence to the optimal inspection trajectory in Section V. In Section VI, we provide simulation results and we close with conclusions and future work.

II. Related Work

The Problem of Inspection planning has attracted the interest of several researchers over the last 15 years. Choset's group proposed planners for coverage and exploration using cell decomposition methods [1], [3], [5] and various heuristics [2]. Their approaches yield good results but are restricted to low dimensional state spaces.

Most of today's inspection planning methods can be divided into two approaches. One approach [6], [7], [8],

This work is supported by the Singapore-MIT Alliance for Research and Technology (SMART) Center for Environmental Sensing and Modeling (CENSAM).

[9] separates the problem into two sub-problems, the art-gallery problem and the TSP. Methods based on this separation approach often have difficulties to plan trajectories for robots with non-holonomic constraints operating in cluttered environments, because some observation points generated by art-gallery solvers may be difficult or even impossible to reach from other observation points. Furthermore, these methods do not provide optimality guarantees, even if each sub-problem is solved optimally. Although [9] provides a probabilistic completeness analysis of the algorithm and proposes a smoothing algorithm that shortens the inspection paths by taking advantage of the asymptotically optimal RRT* algorithm; however, this method cannot guarantee global optimality. By addressing the inspection planning problem as a whole, RITA guarantees that, in the limit, it converges to the global optimal solution with probability one.

Another approach (e.g., [11]) relies on a sub-modular objective function to guarantee that greedy algorithms generate near-optimal solutions. However, when the objective is to minimize the length of the inspection trajectory, as in our case, the problem is not sub-modular, and a greedy method may have difficulties as described in Figure (1).

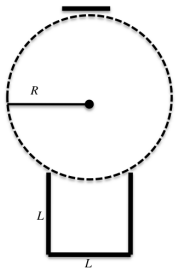


Fig. 1. An illustration of greedy strategy that always choose to move to a configuration within R radius from its current position, that can see the largest unseen structure. The environment contains two structures, i.e., the "U" shaped at the bottom and the line at the top. In this example, greedy may fluctuate moving down and up, while the shortest trajectory would move to cover the structure on the top first, before going down.

III. Problem Definition

Let us first discuss the structures to be inspected and the workspace. The workspace contains the structure to be inspected and obstacles. Obstacles will not be inspected, however they effect the complexity of the problem by changing the properties of the inspection problem and the free space as defined below. We model each structure and obstacle as a simple polygon for 2D workspaces and as simple polyhedra for 3D workspaces.

To inspect the structures, we use a robot with first-order differential constraints equipped with a visibility sensor. The proposed approach can be used for both discrete and continuous sensing actions. Suppose the set of all possible robot configurations is C and the set of all possible control inputs is U. The motion constraints of the robot are described by

$$\dot{q} = g(q, u) \tag{1}$$

where $q \in C$, $u \in U$, $\dot{q}$ is the first derivative of q with respect to time, and g is a smooth function. Assuming that C and U are manifolds of dimensions n and m where $m \leq n$, using appropriate charts we can treat C as a subset of $\mathbb{R}^n$ and U as a subset of $\mathbb{R}^m$. Although we only discuss robots with first-order differential constraints, our algorithm can be extended to robots with higher order-differential constraints in a straightforward manner.

Suppose S is the union of all points that lie on the boundary of the structures in the workspace. Our goal is to generate the shortest collision-free trajectory γ^* from a given initial configuration $q_0 \in C$, such that when the robot follows γ^*, it can see all points in S from at least one point in γ^*. Let us define this objective more formally. Suppose C_{free} is the collision-free subset of C. Let trajectory $\gamma : [0, T] \rightarrow C_{free}$ be a time-parameterized path induced by the function $u : [0, T] \rightarrow U$ through Equation (1). Suppose $V(\gamma(t)) \subseteq S$ is the set of points on the boundary of the structures that are visible to the robot at configuration $\gamma(t)$ for $t \in [0, T]$. An admissible trajectory is a collision-free trajectory that covers S, in the sense that $\bigcup_{t\in[0,T]} V(\gamma(t)) = S$. The proposed framework is flexible enough to define any objective function that spans any subset of the state space and visibility space. In this particular paper, we define the optimal trajectory based on its workspace length. Suppose $D(\gamma)$ is the workspace length of γ. Then, the optimal trajectory is defined as $\gamma^* = \arg\min_{\gamma\in\Gamma} D(\gamma)$, where Γ is the set of all admissible trajectories. Now, since the changes in configurations depend on the control input through the equation of motion (Equation (1)), our goal is to find the control function $u^*(t)$ that induces γ^*.

IV. Random Inspection Tree Algorithm (RITA)

Contrary to many inspection planning methods, RITA *does not* separate the inspection planning problem into the art gallery and the TSP problems. Instead, it approximates the optimal inspection trajectory γ^* by addressing the control and visibility aspects of the in inspection planning problem simultaneously.

RITA approximates the optimal trajectory (if one exists) incrementally. It uses sampling-based motion planning to search the control space U to first find an admissible trajectory, and then improve it subsequently by continuing to search for shorter admissible trajectories.

RITA constructs a tree $\mathbb{T} = \{\mathbb{M}, \mathbb{E}\}$, where the root of the tree is the initial configuration q_0. Each node $q \in \mathbb{M}$ in the tree corresponds to a collision-free configuration while each edge $\overline{qq'} \in \mathbb{E}$ corresponds to a control input $u \in U$ that drives the robot from q to q' in a unit time, without colliding with any of the structures and obstacles while satisfying the equation of motion, Equation (1). Each node $q \in \mathbb{M}$ is annotated with the cost of reaching q from q_0, and with a set of visible points, i.e., the set of points $S' \subseteq S$ that the robot sees if it moves according to the path from q_0 to q in $\mathbb{T}$. The proposed framework is flexible enough to use both continuous and discrete (on the nodes) scanning actions; results in this paper are taken using discrete scanning actions. A path from q_0 to a node q in $\mathbb{T}$ is an admissible trajectory whenever q is annotated with S as its set of visible points.

To construct the tree, any sampling-based motion planning algorithm [4] can be used. In this paper, we use the planner in [12], as it samples the control space directly, and hence does not require inverse-dynamic computation, which may

be difficult to compute for systems with complex dynamics.

Algorithm 1 Path Planning for inspection

```
1: Initialize: Set Initial configuration, Set Structure and Obstacle data
   structure, BestCost=+∞
2: while (PlanningTime=TRUE) do
3:    [Child, Parent, TrajFromParent, u]=ExpandTree(𝕋);
4:    if (CollisionFree(TrajFromParent) ∧ PossiblyOptimal(Child) then
5:       NodeCost=Cost(Parent)+Cost(TrajFromParent)
6:       if (NodeCost<BestCost) then
7:          NodeList=NodeList ∪ Child
8:          EdgeList=EdgeList ∪ ParentChild
9:          NewVisibility=Visib(Child)
10:         Child.GlobalVisib=Parent.GlobalVisib∪NewVisibility
11:         Child.Unseen=S \ Child.GlobalVisib
12:         if (Child.Unseen=∅) then
13:            BestCost=NodeCost
14:            BestNode=Child
15:         end if
16:      end if
17:   end if
18: end while
```

Algorithm 2 ExpandTree($\mathbb{T}$)

1: Compute the number of nodes in the neighborhood: $D_n(q)$, $\forall q \in \mathbb{M}$
2: Probability to choose a node: $P_n(q) \propto 1/(D_n(q))$, $\forall q \in \mathbb{M}$
3: Sample a node: $q' \leftarrow rand(P_n(q))$
4: Sample control input: $u\prime \leftarrow rand()$
5: Propagate: $[q_{new}, TraFromParent] \leftarrow \int_0^{\Delta t} g(q', u\prime)dt$
6: **return** $q_{new}, q\prime, TraFromParent, u\prime$

The tree $\mathbb{T}$ is constructed incrementally (Algorithm 1). At each iteration of the algorithm, we try to expand the tree using the "ExpandTree()", which expands the tree by randomly choosing a node to expand and a control input to apply. The node to expand q' is chosen with probability proportional to the inverse of the number of nodes within its neighborhood (a small ball with q' as its center, Figure (2)). ExpandTree() function takes as its input the current Tree and returns a candidate new node q_{new}, the new node's parent q', the node's trajectory from q' to q_{new}, and the control input used. If q_{new} and the trajectory from q' to q_{new} are obstacle free, then we compute the cost to reach q_{new} through q'. The cost for the best admissible solution found up to the current time is recorded and used to reject node candidates if they result in trajectories longer than the best one found up to that point. If the new node q_{new} is not rejected, then RITA inserts q_{new} as a child of q' in $\mathbb{T}$, annotates the edge $\overline{q_{new}q}$ with u, and computes the set of points $S' \subseteq S$ that is visible when the robot moves from q' to q_{new} (in the case we use continuous scanning actions) or it computes the visibility of q_{new} (in the case we use discrete scanning actions, for this paper we used discrete scanning actions).

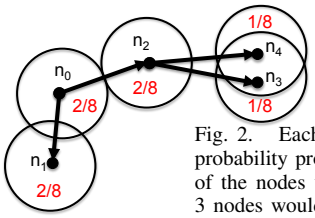


Fig. 2. Each node to be expanded is chosen with probability proportional to the inverse of the density of the nodes within its neighborhood, here the first 3 nodes would be sampled with probability 2/8 and the last 2 with probability 1/8

To speed up the search for the optimal trajectory, RITA rejects nodes that could not be part of the optimal trajectory (PossiblyOptimal function in line-4 of Algorithm 1). RITA excludes controls that drive the search to areas in which there is no probability that parts of the optimal trajectory will be present. To do so, RITA defines the Minkowski sum of the convex hull of the structure of interest and the obstacles within the the workspace with a sphere radius $(R_{sensor} + R_{turning})$, where R_{sensor} denotes the maximum range of the sensor and $R_{turning}$ is the minimum turning radius of the agent (derived from dynamics). The optimal trajectory must lie within the Minkowski sum defined above. Therefore, if a control u drives the agent to node q, whose projection in the workspace is located outside of the Minkowski sum described above, then q is rejected.

V. Analysis

In this section, we prove two fundamental properties of RITA, probabilistic completeness and convergence to the optimal inspection trajectory. To simplify the analysis, we use a simplified version of RITA, called ideal-RITA. Ideal-RITA is the same as RITA, but it replaces the sampling strategy in Algorithm 2 with Ideal-Expand, as described in Algorithm 3. The notation $R_l(q)$ is the set of configurations reachable in l units of time from q.

Algorithm 3 Ideal-Expand($\mathbb{T} = (\mathbb{M}, \mathbb{E})$)

1: Sample a milestone $m\prime$ uniformly at random from $R_l(\mathbb{M})$.
2: Sample a milestone m that can reach $m\prime$.
3: Compute a control input: u to move from m to $m\prime$.
4: **return** $m\prime, m, Trajectory(m, m\prime, u), u$

A. Probabilistic Completeness

Probabilistic completeness in our case means that if an admissible trajectory, i.e., one that covers all points in S, exists, then given enough time, ideal-RITA will find it with high probability. This notion of completeness is slightly different from that which is commonly used in motion planning, i.e. covering the entire C_{free}. To show the probabilistic completeness of ideal-RITA, we first need to define the following property of the inspection planning problem.

Definition 1: Let S be the set of all points on the boundary of the structure to be inspected. We define a v-partitioning of S as the set $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$, where $n \in \mathbb{N}^+$ and for each $i \in [1, n]$, $S_i \subseteq S$ and there exists a non-empty and connected set $A(S_i) \subseteq C_{free}$ whose volume is $Vol(A(S_i)) \geq v \cdot Vol(C_{free})$ and each configuration in $A(S_i)$ can see all points in S_i. An inspection planning problem is (k, v)-inspectionProblem whenever the smallest v-partitioning of S has k elements.
Figure (3) illustrates the above definition. Intuitively, the problem is easy to solve when k is small and v is large.

To analyze the probabilistic completeness of RITA, we need C_{free} to be (α, β)-expansive [12]. The (α, β)-expansive property for robots with differential constraints

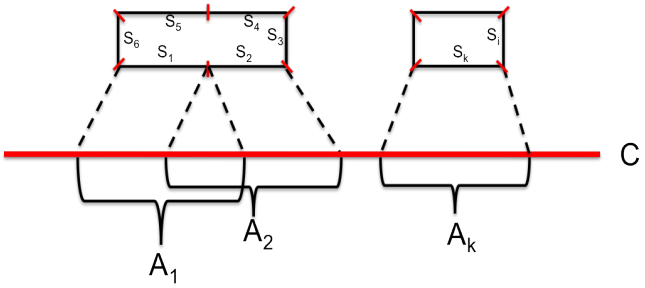


Fig. 3. A cartoon illustration of v-partitioning.

have been defined in [12]; we rewrite this definition here for completeness.

Definition 2: Let $\alpha, \beta \in (0, 1]$ be constants, $q \in C_{free}$, $R(q) \subseteq C_{free}$ be the set of configurations reachable from q, and $R_l(q) \subseteq R(q)$ be the set of configurations reachable from q within l time-steps. Then, for any $q \in C_{free}$, $R(q)$ is (α, β)-expansive if for every connected subset $C' \subset R(q)$, $Vol(\beta\text{-}lookout(C')) \geq \alpha \cdot Vol(C')$, where $\beta\text{-}lookout(C') = \{q \in C' \mid Vol(R_l(q) \backslash C') \geq \beta \cdot Vol(R(C') \backslash C')\}$. The set C_{free} is (α, β)-expansive if for every $q \in C_{free}$, $R(q)$ is (α, β)-expansive.

Using the above definitions and the results in [12], we can compute a lower bound on the number of sampled milestones, such that the structure S is covered with high probability. More precisely, we show that:

Theorem 1: Given a (k, v)-inspectionProblem and an (α, β)-expansive collision-free space as defined in [12], a sequence M of n milestones generated by the ideal-RITA completely covers the structure S (i.e., $S \subseteq V(M)$) with probability at least 1 - $2kf(\beta, v)e^{-\frac{r\alpha}{f(\beta,v)}}$ where $f(\beta, v) = \frac{\ln(2/v)}{\beta}$ and $r = \lfloor n/k \rfloor$.

Proof: Let L be the event that $S \subseteq V(M)$ and let L' be the event that M contains at least one configuration that lies in each $A(S_i)$ for $i \in [1, k]$. Based on the definition of (k, v)-inspectionProblem, it is clear that $L' \subseteq L$ and hence $P(L) \geq P(L')$.

Suppose M is divided into k subsequences where each subsequence contains r consecutive milestones, and L'_i is the event that the i^{th} subsequence contains a configuration in $A(S_i)$. Then,

$$\begin{aligned} P(L') &\geq P(L'_1 \cap L'_2 \cap \ldots \cap L'_k) \\ &= 1 - P(\overline{L'_1} \cup \overline{L'_2} \cup \ldots \cup \overline{L'_k}) \\ &\geq 1 - \sum_{i=1}^{i=k} P(\overline{L'_i}) \end{aligned} \quad (2)$$

Using the results in [12], $P(\overline{L'_i}) \leq 2f(\beta, v)e^{-\frac{r\alpha}{f(\beta,v)}}$ and we can rewrite Equation (2) as $P(L') \geq 1 - 2kf(\beta, v)e^{-\frac{r\alpha}{f(\beta,v)}}$. Since $P(L) \geq P(L')$, we get the results we want. ■

Notice from Theorem 1 that with the same number of milestones, ideal-RITA can cover S with higher probability when we have smaller k and larger v, which fits our intuition.

B. Asymptotic Convergence of RITA

Recall that the optimal trajectory γ^* is an admissible trajectory with the shortest workspace length. To show convergence of ideal-RITA to γ^*, intuitively, we show that the ideal-RITA is equivalent to Rapidly-exploring Random Graphs (RRG) [14], in the sense that any set of trajectories that has non-zero probability to be generated by RRG will also have non-zero probability to be generated by the ideal-RITA. Since RRG has been shown to converge to the optimal trajectory almost surely, so is the ideal-RITA.

Algorithm 4 Inspection-RRG(N)

1: $G.V = \{q_0\}$, $G.E = \emptyset$, $i = 0$.
2: **while** $i < N$ **do**
3: $q_{rand} =$ SampleAConfiguration(G).
4: $q_{nearest} =$ FindNearest(G, q_{rand}).
5: $q_{new} =$ Steer($q_{nearest}, q_{rand}$).
{Steer function finds a configuration within a pre-specified distance d from $q_{nearest}$.}
6: **if** CollisionFree($q_{nearest}, q_{new}$) **then**
7: $G.V = G.V \cup \{q_{new}\}$, $G.E = G.E \cup \{(q_{nearest}, q_{new})\}$.
8: **end if**
9: **if** $R_l(q_{new}, q_{nearest}) \bigwedge$ CollisionFree($q_{new}, q_{nearest}$) **then**
10: $G.E = G.E \cup \{(q_{new}, q_{nearest})\}$.
11: **end if**
12: $Q_{near} =$ FindNearby($G, q_{new}, |V|$).
{FindNearby finds the vertices of G that are within a certain distance from q_{new}. The exact distance depends on $|V|$.}
13: **for all** $q_{near} \in Q_{near}$ **do**
14: **if** $R_l(q_{new}, q_{near}) \bigwedge$ CollisionFree(q_{new}, q_{near}) **then**
15: $G.E = G.E \cup \{(q_{new}, q_{near})\}$.
16: **end if**
17: **if** $R_l(q_{near}, q_{new}) \bigwedge$ CollisionFree(q_{near}, q_{new}) **then**
18: $G.E = G.E \cup \{(q_{near}, q_{new})\}$
19: **end if**
20: **end for**
21: **end while**
22: $\Gamma =$ All admissible trajectories in G.

Note however, that our problem is not the same as the problem in [14], and hence RRG needs to be adapted to fit our problem. We call this adapted RRG "inspection-RRG" (Algorithm 4). Two main differences need to be handled. The first is the definition of admissible trajectory. In our case, an admissible trajectory is a collision free trajectory that covers all points in S and starts from a given initial configuration, while in [14], an admissible trajectory is a collision-free trajectory between a given pair of initial and goal configurations. To adapt RRG to our problem, we add a post-processing step (line-22 of Algorithm 4) that eliminates all trajectories that do not satisfy our admissible definition. The second difference is that RRG does not handle differential constraints, while our system has differential constraints. We handle this difference by adapting RRG similarly to the adaptation in [13]. Notice that none of our modifications eliminates admissible trajectories, and therefore, the asymptotic optimality property of RRG as shown in [14] remains valid in inspection-RRG.

Before we state the equivalent relation and the asymptotic convergence results more formally, we first need to state the required assumptions. The first two assumptions below are similar to the assumptions required for the convergence of RRT* [13], while the last assumption is related to the visibility requirement. The assumptions are:

1) The robotic system is Weakened Local Controllable. A

system is Weakened Local Controllable[1] if there exist constants η, $\bar{\epsilon} \in \mathbb{R}^+$, $p \in \mathbb{N}$, such that for any $\epsilon \in (0, \bar{\epsilon})$ and any configuration q the set $R_\epsilon(q)$ contains a ball of radius $\eta\epsilon^p$, where $R_\epsilon(q)$ refers to the set of all configurations reachable from q with a trajectory that lies entirely inside the ball with center q and radius ϵ.

2) The optimal admissible trajectory $\gamma^* : [0, T_{\gamma^*}] \rightarrow C_{free}$ has ϵ-clearance. This means that for any $t \in [0, T_{\gamma^*}]$, there exist a closed ball $B_\epsilon(\gamma^*(t)) \subset C_{free}$ with radius ϵ and centered at $\gamma^*(t)$. And for any $t_1, t_2 \in [0, T_{\gamma^*}], t1 < t2$, the ball of radius $\eta\|\gamma(t_1) - \gamma(t_2)\|^p$ centered at $\gamma(t_2)$ is ϵ-reachable from $\gamma(t_1)$.

3) The optimal trajectory $\gamma^* : [0, T_{\gamma^*}] \rightarrow C_{free}$ is δ-elastic, where $\delta = \eta\epsilon^p$. Suppose $\mathbb{B}_\delta(\gamma^*) = \bigcup_{t\in[0,T_{\gamma^*}]} B_\delta(\gamma^*(t))$, where $B_\delta(\gamma^*(t))$ is the ball with radius δ and center at $\gamma^*(t)$. γ^* is δ-elastic when any trajectory $\gamma : [0, T_{\gamma^*}] \rightarrow C_{free}$ that lies entirely in $\mathbb{B}_\delta(\gamma^*) \cap C_{free}$ covers the entire S. This elastic trajectory is an extension of the elastic solution in the art gallery problem in [10].

The first assumption is exactly the same as the first assumption in [13], which we restate here for completeness. The second assumption is adapted from the second assumption in [13]. This adaptation is needed because our admissible trajectory starts from q_0 and covers the entire structure boundary S, while an admissible trajectory in [13] starts from q_0 and ends at a given goal configuration. The last assumption is related to the visibility requirement. Again, this assumption is required because, unlike the problem in [13], in our case, an admissible trajectory must cover the entire structure boundary S.

Now, we formally state the equivalent relation between the ideal-RITA and inspection-RRG,

Lemma 1: Suppose G is the set of all admissible trajectories defined from $[0, T]$ to C_{free} that starts from a given initial configuration q_0, for any $T \in \mathbb{R}^+$. For any subset $g \subseteq G$, if P(inspection-RRG samples g) > 0, then P(ideal-RITA samples g) > 0.

Proof: Let γ be an admissible trajectory with ϵ-clearance property and is formed by a sequence of k different milestones, $M = \{m_1, m_2, \ldots, m_k\}$ where $m_1 = q_0$. Let $B_\epsilon(\gamma) \subseteq G$ be a set of admissible trajectories where each trajectory is formed by k milestones $M' = \{m_1, m'_2, \ldots, m'_k\}$, where $m'_i \in B_\epsilon(m_i)$ for $i \in [2, k]$, and $B_\epsilon(m_i) \subset C_{free}$ is a closed ball with radius ϵ and center m_i. Notice that if P(inspection-RRG samples a trajectory in $B_\epsilon(\gamma)) > 0$, then $Vol(B_\epsilon(m_2) \cap R_l(m_1)) > 0$ and inspection-RRG samples $B_\epsilon(m_2) \cap R_l(m_1)$ with non-zero probability. Since the tree $\mathcal{T}$ built by the ideal-RITA contains $q_0 = m_1$, there is a non-zero probability that the ideal-RITA samples a point in $R_l(m_1)$. And since $Vol(B_\epsilon(m_2) \cap R_l(m_1)) > 0$, ideal-RITA will sample $B_\epsilon(m_2) \cap R_l(m_1)$ with non-zero probability too. Let's assume that both inspection-RRG and ideal-RITA samples the same configuration in $B_\epsilon(m_2) \cap R_l(m_1)$. By repeating the same argument until $B_\epsilon(m_k)$, ideal-RITA must have non-zero probability of sampling $B_\epsilon(\gamma)$ too.
When the trajectory generated by inspection-RRG contains loops, ideal-RITA can approximate it arbitrarily close but would require additional computation time. ■

Now, we can state the convergence theorem more formally.

Theorem 2: Suppose all the above assumptions hold. Suppose further that γ_i is the best path in the tree built by ideal-RITA after the i^{th} iteration and γ^* is the optimal trajectory. The cost $D(\gamma_i)$ satisfies $Limit_{i\rightarrow\infty} P(D(\gamma_i) = D(\gamma^*)) = 1$.

Proof: This asymptotic convergence has been proven to hold for RRG [14]. Since inspection-RRG only eliminates non-admissible solutions from the set of all trajectories generated by RRG, asymptotic convergence to the optimal trajectory holds for inspection-RRG. Based on Lemma 1, the ideal-RITA can generate all trajectories generated by inspection-RRG. This means that as i goes to infinity, the set of all trajectories generated by inspection-RRG and ideal-RITA will be the same. Therefore, if the trajectories generated by inspection-RRG contain the optimal trajectory with probability one, then the same holds for ideal-RITA. ■

VI. Simulation Results

We implemented RITA in C++ and run a set of simulations (the code is not optimized, optimized code further reduces the running time). We consider a 2-D workspace and the following robot's dynamics.

$$\begin{aligned} \dot{x} &= u_s \sin\theta \\ \dot{y} &= u_s \cos\theta \\ \dot{\theta} &= \Omega \end{aligned} \tag{3}$$

The configuration space is 3D and the control space is 2D. The sensor model is such that it observes everything within a maximum radius R_{range} that is not occluded by the structure. The visibility for a given trajectory is computed by projecting the footprint of the sensor on the workspace and using linear algebra methods to identify parts that are occluded by the structure and the obstacles in the workspace.

We consider 3 different case studies. In the first case, the range of the sensor is 20 units, u_s is the speed of the agent with values between 0.5 and 3 units per second, and Ω is the angular velocity with values between -0.25 and 0.25 radians per second. These parameters result in a minimum turning radius of 2 units. The initial configuration for the first case study is $q_0 = [0, 0, 0]^T$.

The second case is more challenging with a small visibility range and several narrow passages. Because the maximum range of the sensor is small compared to the environment, the agent has to navigate inside several of the narrow passages. The range of the sensor is 15 units, u_s is the speed of the agent with values between 0.5 and 4 units per second, and Ω is the angular velocity with values between -0.25 and 0.25 radians per second. These parameters result in a minimum turning radius of 2 units. The initial configuration for the second case study is $q_0 = [45, 0, \pi/2]^T$. For both case studies we also consider additional obstacles that are located close to the structure of interest.

[1]not to be confused with the Weak Local Controllability, Dubins car which is a non-holonomic system satisfies Weakened Local Controllability.

For each of the first 2 test cases, we run the algorithm 20 times to get running time and performance statistics. The computer used to generate the results runs on a 3GHz processor and Ubuntu 10.04 as the Operating System. The third case study shows an example where RITA computes the optimal inspection trajectory and at the same time others approaches fail.

1) First Case Study: On average, RITA finds the first admissible solution within 0.24 seconds with standard deviation 0.34 seconds. It generates a solution close to the optimal (below 67 units) after 11 minutes, with standard deviation 4.6 minutes. Figure (6, left) shows the average cost for the 20 runs as a function of the actual running time.

Figure (4) shows the generated admissible trajectory across different time of one of the simulation runs. In this particular run, the algorithm finds the first admissible trajectory (cost 120.6 units) after running for 0.7 seconds. It is important to note that the first admissible solution found does not belong to the same homotopy class as the optimal trajectory. We keep running the algorithm, and at 71.5 seconds it switches to a different homotopy class and finds another admissible trajectory with cost 114.9 units. The algorithm improves the current solution within the same homotopy class (at time 117.3 seconds and cost 100.85 units). At time 171.4 seconds it finds another admissible trajectory with cost 92.5 units. After 400.3 seconds, it finds a better trajectory (this particular one of cost 68.7 units), which belongs to the same homotopy class as the optimal trajectory, and eventually this trajectory converges to a near-optimal one (63.8 units after 951.1 seconds).

Figure (5, above left) shows a comparison between the trajectory that the algorithm converges to (here we show another run; at this particular run RITA converges to a trajectory with cost 64.82 units after 74.73 seconds) and the optimal holonomic trajectory (59 units). The results indicate that the length of the trajectory RITA converges to, is close to the lower bound (optimal holonomic trajectory).

The proposed algorithm can also handle additional obstacles in the workspace. In Figure (5, upper right and below) we show the results RITA gives if we include obstacles close to the structure of interest. In this particular run, the cost decreases from 124.4 to 73.3 units for 0.07 seconds to 1hour running time.

2) Second Case Study: On average, RITA finds the first admissible solution within 12.87 seconds (with standard deviation 25.4 seconds) and gets a solution close to the optimal (below 190 and 185 units) after 44 and 73 minutes respectively (with standard deviation 53 and 85 minutes). As Figure (6) shows, by running it longer, the length of the generated admissible trajectory becomes shorter. Figure (6, right) shows the average cost for the 20 runs as a function of the actual running time.

Figure (7,a-d) shows the generated admissible trajectory across different time of one of the simulation runs. In this run, the algorithm finds the first admissible solution after running for 43.5 seconds (cost 219 units). It improves the current solution within the same homotopy class, as shown in Figure (7,b) (113.7 seconds for cost 200 units). Then the algorithm explores few other homotopy classes and finally it converges to a trajectory with cost close to the optimal one (177.6 units in 51 minutes). For this case study, due to symmetry, there are different homotopy classes of trajectories with cost close to the optimal one (or the same one). In Figure (7,e-f) we can see 2 different runs that resulted in 2 different trajectories with cost close to the optimal one (177.26 units in 67 seconds and 177.66 units in 100 minutes).[2] These runs show that the algorithm indeed explores all homotopy classes that need to be explored, until reaching a trajectory sufficiently close to the optimal one. Figure (7,g-h) shows some of the results RITA gives if we include obstacles close to the structure of interest. In this particular run, the cost decreases from 213.9 to 191.5 units for 65 seconds to 2 hours running time.

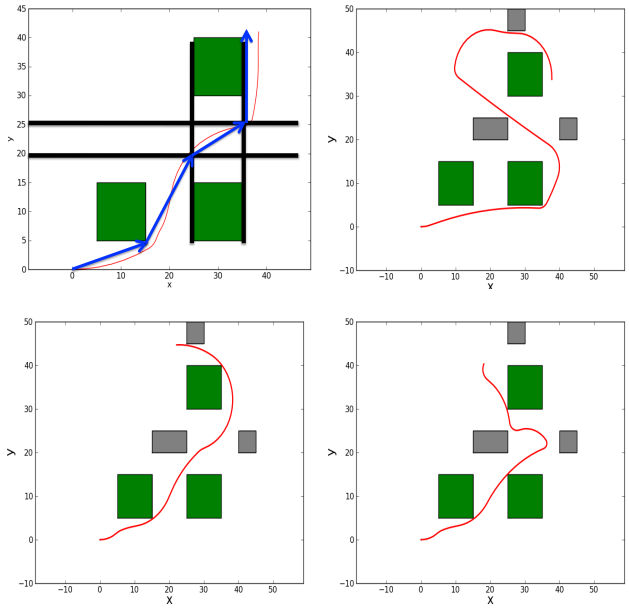
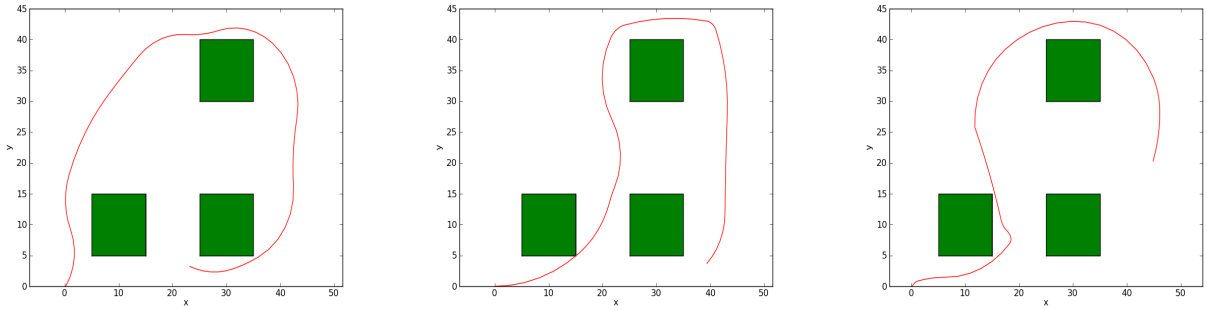


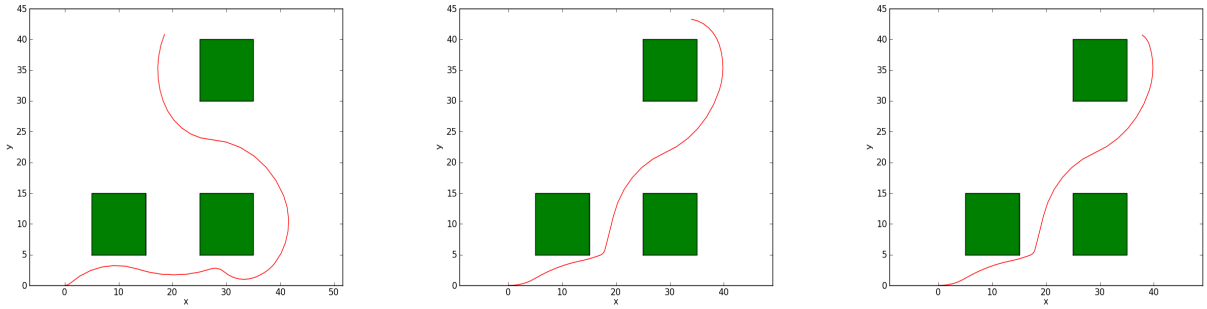
Fig. 5. First case study: Above part) Comparison between the trajectory that the algorithm converges to and the optimal holonomic trajectory. We see that the optimal holonomic trajectory is close to the one the algorithm generates. Upper right and below part) The effect of additional obstacles (gray color) close to the structure of interest.

3) Third case study: In the third case study we consider the corridor example. We are interested in observing the inside part of the corridor. The agent's initial configuration is $q_0 = [0, 20, 0]^T$ and its visibility rage is 23 units. For this particular example, it is easy to identify the optimal inspection trajectory which is a straight line of length 68.64 units. Figure (8) shows the generated admissible trajectory across different time. In this run, the algorithm finds the first admissible solution after running for 0.2 seconds (cost 109 units). It improves the current solution until converging close to the optimal inspection trajectory (cost 70 units). In Figure (8), in the right, we can see the optimal inspection trajectory

[2]we stopped all runs within 2 hours, if we keep running the algorithm for more time it switches back and forth between homotopy classes, continuing to improve the cost.

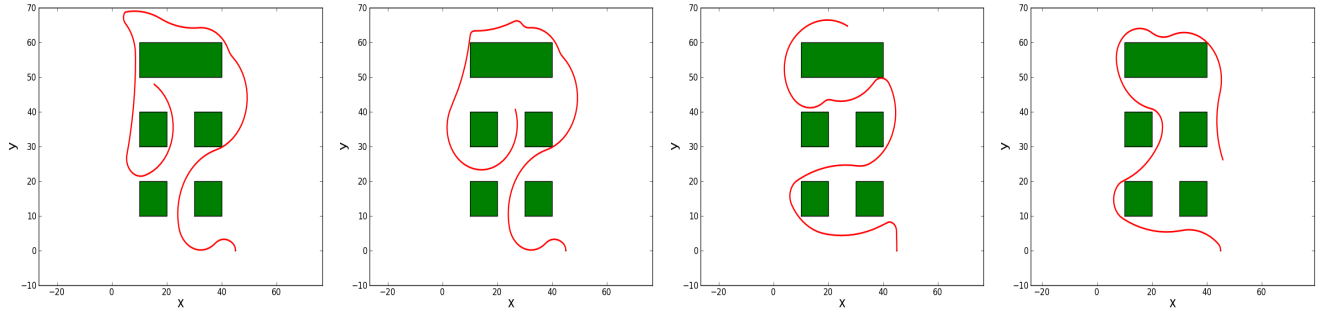


$t = 0.7$ *seconds* $t = 71.5$ *seconds* $t = 117.3$ *seconds*

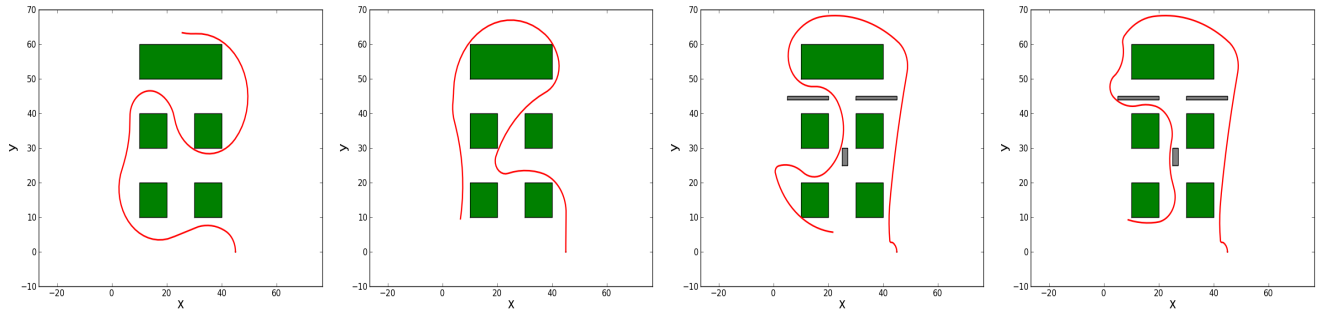


$t = 171.4$ *seconds* $t = 400.3$ *seconds* $t = 951.1$ *seconds*

Fig. 4. First case study, one of the 20 runs: The algorithm finds the first admissible trajectory (cost 120.6 units) after running for 0.7 seconds. It is important to note that the first admissible solution found does not belong to the same homotopy class as the optimal trajectory. We keep running the algorithm, and at 71.5 seconds it switches to a different homotopy class and finds another admissible trajectory with cost 114.9 units. The algorithm improves the current solution within the same homotopy class (at time 117.3 seconds and cost 100.85 units). At time 171.4 seconds it finds another admissible trajectory with cost 92.5 units. After 400.3 seconds, it finds a better trajectory (this particular one of cost 68.7 units), which belongs to the same homotopy class as the optimal trajectory, and eventually this trajectory converges to a near-optimal one (63.8 units after 951.1 seconds).



(a) $t = 43.5$ *seconds* (b) $t = 113.7$ *seconds* (c) $t = 376.8$ *seconds* (d) $t = 51$ *minutes*



(e) $t = 67.1$ *seconds* (f) $t = 100$ *minutes* (g) $t = 39$ *minutes* (h) $t = 2$ *hours*

Fig. 7. Second case study, (a-d) the algorithm finds the first admissible solution after running for 43.5 seconds (cost 219 units). It improves the current solution within the same homotopy class (113.7 seconds for cost 200 units). Then the algorithm explores few other homotopy classes and finally it converges to a trajectory with cost close to the optimal one (177.6 units in 51 minutes). (e-f) Different homotopy classes of trajectories with cost close to the optimal one. (i-j) Results with additional obstacles (obstacles in gray color).

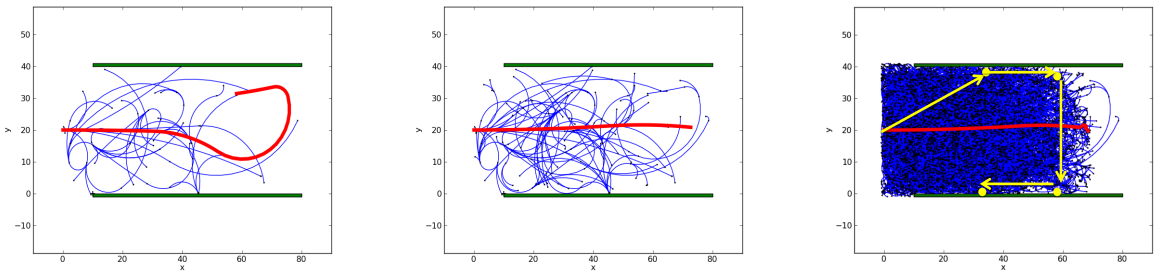


Fig. 8. Third case study: the algorithm converges close to the optimal inspection trajectory, at the same time the TSP and art-gallery approaches fail to give a trajectory close to the optimal one (cost 126.6 units, yellow dots show the optimal art-gallery observation points connected with straight lines, the visiting order is given by solving the TSP).

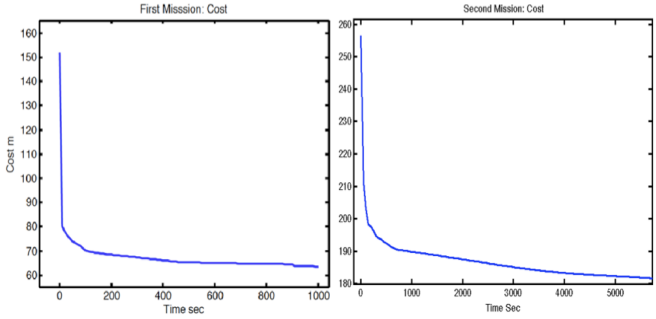


Fig. 6. The average cost for all runs as a function of the actual running time. Left (first scenario): The algorithm, in average, finds the first admissible solution within 0.24 seconds (with standard deviation 0.34 seconds) and get a solution close to the optimal (below 67 m) after 11 minutes (with standard deviation 4.6 minutes). Right (second scenario): The algorithm, in average, finds the first admissible solution within 12.87 seconds (with standard deviation 25.4 seconds) and gets a solution close to the optimal (below 190 and185 units) after 44 and 73 minutes respectively (with standard deviation 53 and 85 minutes).

RITA gives and the trajectory we will get if we use TSP and the art-gallery approaches. We can see that the TSP and the art-gallery approach results cost of 126.6 units.

VII. Conclusions

In this paper, we are interested in deterministic inspection planning. We propose a new sample based inspection planning algorithm, called RITA. Given a perfect model of a structure, sensor specifications, robot's dynamics, and an initial configuration of a robot, RITA computes the optimal inspection trajectory that observes all points on the structure. Instead of separating the problem into the art-gallery problem and the TSP as in most inspection planning algorithms, RITA addresses the control and visibility aspects of the inspection problem simultaneously. To easily account for differential constraints, RITA performs its sampling in the control space. It finds inspection trajectories with decreasing cost, and in the limit, converges to the optimal inspection trajectory with probability one. Furthermore, we present a formal analysis on the probabilistic completeness and asymptotic optimality of our algorithm.

Future work will include improving the implementation to speed up current computation and implementing the algorithm to inspect 3-D structures.

Acknowledgements: We would like to thank the anonymous reviewers for their valuable feedback.

References

[1] E. U. Acar and H. Choset. Sensor-based coverage of unknown environments: Incremental construction of morse decompositions. *International Journal of Robotics Research*, 21(4):345–366, April 2002.

[2] P. N. Atkar, A. Greenfield, D. C. Conner, H. Choset, and A. A. Rizzi. Uniform Coverage of Automotive Surface Patches. *International Journal of Robotic Research*, 24:883–898, 2005.

[3] H. Choset. Coverage of known spaces: The boustrophedon cellupdar decomposition. *Autonomous Robots*, 9:247 – 253, 2000.

[4] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun. Principles of Robot Motion: Theory, Algorithms, and Implementations. *The MIT Press*, 2005.

[5] H. Choset and P. Pignon. Coverage path planning: The boustrophedon cellular decomposition. In *Proceedings of the International Conference on Field and Service Robotics*, 1997.

[6] T. Danner and L. E. Kavraki. Randomized planning for short inspection paths. In *Proceedings of the International Conference on Robotics and Automation*, volume 2, pages 971–976, San Fransisco, CA, April 2000.

[7] B. Englot and F. Hover. Inspection planning for sensor coverage of 3-D marine structures. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 2010.

[8] B. Englot and F.S. Hover. Sampling-based coverage path planning for inspection of complex structures. In *Proceedings of the International Conference on Automated Planning and Scheduling*, pages 29–37, Sao Paulo, Brazil, 2012.

[9] B. Englot and F.S. Hover. Sampling-based sweep planning to exploit local planarity in the inspection of complex 3D structures. In *Proceedings of the International Conference on Intelligent Robots and Systems*, Vilamoura, Portugal, 2012.

[10] H. Gonzalez-Baños and J.-C. Latombe. A randomized art gallery algorithm for sensor placement. In *Proceedings of the 17th Annual ACM Symposium on Computational Geometry*, pages 232–240, 2001.

[11] G.A. Hollinger, B. Englot, F. Hover, U. Mitra, and G.S. Sukhatme. Uncertainty-driven view planning for underwater inspection. In *Proceedings of the International Conference on Robotics and Automation*, pages 4884–4891, St. Paul, MN, 2012.

[12] D. Hsu, R. Kindel, J.C. Latombe, and S. Rock. Randomized kinodynamic motion planning with moving obstacles. *International Journal of Robotics Research*, 21(3):233–255, 2002.

[13] S. Karaman and E. Frazzoli. Optimal kinodynamic motion planning using incremental sampling-based methods. In *Proceedings of the 49th IEEE Conference on Decision and Control*, 2010.

[14] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *International Journal of Robotics Research*, 30(7):846–894, June 2011.