A Localization Aware Sampling Strategy for Motion Planning under Uncertainty

Vinay Pilania and Kamal Gupta

Abstract— **We present a localization aware efficient sampling strategy for sampling-based motion planning under uncertainty that uses a new notion of localization ability of a sample. It puts more samples in regions where sensor data is able to achieve higher uncertainty reduction while maintaining adequate samples in regions where uncertainty reduction is poor. This leads to a less dense roadmap and hence results in significant time savings in the path search phase. We provide simulation results that show stochastic planners with our sampling strategy place less samples and find a well-localized path in shorter time with little compromise on the quality of path as compared to existing sampling techniques. We also show that a stochastic planner that uses our sampling strategy is probabilistically complete under some reasonable conditions on parameters.**

I. Introduction

Safe execution of motion plans and accurate information of robot state are of critical importance for many robotic tasks. As a result of uncertainty associated with a robot's motion and its sensory readings, the true robot state is not available. Therefore, it is important that a planning method must account for these uncertainties for safe and collision-free execution of motion plans. Partially observable Markov decision process (POMDP) [1] is a general framework to deal with motion and sensing uncertainty, however due to its significant complexity, solving realistic problems with large state spaces remains a challenge, even though progress has been made on the efficiency issues of these approaches [2], [3], [4]. A class of planners [5], [6], [7] assume the presence of landmark regions in the environment where accumulated motion uncertainty can be "reset". Another class of planners [8]–[14] uses sampling-based methods (graph-based and tree-based) where uncertainty is propagated from start to goal. We call this type as stochastic motion planners.

In this paper, we address the sampling-based stochastic motion planners. These planners are computationally demanding as compared to their counterparts that do not consider uncertainty. This is because they do not follow the "optimal substructure" property [14] of paths, i.e., the incurred costs on different edges depend on each other. To compute the cost of an edge emanating from a node, the full knowledge of belief (robot pose and associated uncertainty) at the node is required, this in turn requires full knowledge of the history of observations and actions leading up to the node [1]. For real time applications, for instance to facilitate anytime planning [15], it is important to reduce this run time. We propose to do achieve at least part of this run time reduction via judicious placing of samples.

It is important to note that sampling techniques used in current stochastic motion planners do not distinguish between well-localized and poorly-localized regions and given that large open spaces generally tend to be poorly-localized, a uniform distribution tends to place a large number of samples in these poorly-localized regions. This leads to a dense roadmap which in turn increases the computational cost especially if collision checks are carried out in 3D (for example, for mobile manipulators) and furthermore, due to the uncertainty propagation during search phase, the cost of additional "unnecessary" samples is even more.

We propose a localization aware sampling strategy that avoids putting large number of samples by considering the "localization ability" of a new sample relative to its neighbouring nodes in the roadmap. It puts more samples in regions where sensor data is able to achieve higher uncertainty reduction while maintaining an adequate number of samples in regions where uncertainty reduction is poor. This leads to a less dense roadmap that results in significant time savings in the path search phase. Note that localization of a robot at a point depends on 1) the path taken to reach the point and 2) on the update based on sensor model. However, at the sampling stage the path taken to a node is not available. We develop a new measure of "localization ability of a sample" that "extracts" how well a sensor observation at a sample point reduces uncertainty without explicitly knowing the path leading to it and use this measure to design a localization aware sampling strategy.

A key reason we use reduction in uncertainty as a measure is that higher uncertainty is more detrimental and hence has higher cost for many tasks. Nevertheless, one possible consequence of our sampling technique is that path quality (we use true localization uncertainty along the path as a quality metric) may suffer, if the path passes through regions where uncertainty reduction is poor. Via simulation results, we show that, at least empirically, there is little compromise in path quality. Furthermore, note that since at the sampling stage, true localization uncertainty is not available, a cost function metric using it can not be computed, hence can not be used. The best one can do is to use the uncertainty reduction ability of the sensor at the sample point, as we

Vinay Pilania and Kamal Gupta are with Robotic Algorithms & Motion Planning (RAMP) Lab, School of Engineering Science, Simon Fraser University, Burnaby, BC V5A1S6, Canada. Email: `vpilania@sfu.ca`, `kamal@sfu.ca`

[1] [14] is an exception in the sense that the incurred costs on different edges do not depend on each other. This comes at the cost of some assumptions like holonomic and Gaussian systems with trivial dynamics.

do. Note that in the search phase (where edges are added and uncertainty is propagate along the path), appropriate cost function is still minimized.

We provide simulation results that show that stochastic planners (we used RRBT [13]) with our localization aware sampling strategy place less samples and find a well-localized path in shorter time with little compromise on the quality of path as compared to existing sampling techniques used in these planners. Furthermore, we also show that our sampling strategy, with suitable restrictions on its parameters, is also probabilistically complete.

II. Related Work

A large number of sampling schemes have been used with the standard (without uncertainty) sampling based planners (RRT or PRM) such as, sample around and near the obstacles, or in narrow corridors, medial axis sampling to sample far away from the obstacles, use visibility to reduce the number of samples, adaptive strategies such as restrict sampling to size-varying balls around nodes, entropy guided approaches, etc. [16] and [17] provide a survey of recent work in non-uniform sampling for PRMs. Above mentioned sampling approaches do not consider the uncertainty associated with robot and its sensors.

Missiuro and Roy [18] proposed an approach where the sampling strategy incorporates mapping uncertainty (they do not consider localization uncertainty that we consider in this paper) in which the decision to accept or reject a sample is based on its collision probability (computed using each of the possible world model). However, the issue of "how good a sample would be in localizing the robot?", which we explicitly consider does not arise in their problem context. As mentioned earlier, computing the collision probability in the presence of localization uncertainty of a sample right at the sampling stage, i.e., before connecting it to the roadmap is not possible. Note that at sampling stage we consider only sensing uncertainty while for path search (where uncertainty is propagated from start) we consider both motion and sensing uncertainty. To the best of our knowledge, we are not aware of any other sampling approach that considers uncertainty. All sampling-based stochastic motion planners [8]–[14] use one of the sampling techniques from deterministic motion planners and address the motion, sensing, and mapping uncertainty at query phase where a search algorithm searches the roadmap by propagating uncertainty from start to goal.

Although not directly related to motion planning (or sampling techniques), the notion of uncertainty has been used in the past to select the best sample (the next best goal of robot) for search and exploration. For example: [19] first plans for each of the possible goal candidates and select the one (as next best goal) which in addition to information maximization (unknown region), also has good localization along the path.

III. Localization Ability Of A Sample

In this section, we describe how to compute the localization ability of a sample. For this, we first briefly explain the extended Kalman filter (EKF) [11] and then develop an expression for the localization ability of a sample.

Applying a control input u_t at time t brings the robot from state x_t at time t to state x_{t+1} at time $t+1$ according to a given stochastic dynamics model:

$$x_t = f(x_{t-1}, u_{t-1}, w_t), \qquad w_t \sim \mathcal{N}(0, W_t) \tag{1}$$

where w_t is the process noise at time t drawn from a zero-mean Gaussian distribution with variance W_t that models the motion uncertainty. After each motion, the robot receives noisy sensor readings z_t at time t that provide us with partial information about the state according to a given stochastic observation model:

$$z_t = h(x_t, q_t), \qquad q_t \sim \mathcal{N}(0, Q_t) \tag{2}$$

where q_t is the measurement noise drawn from a zero-mean Gaussian distribution with variance Q_t that models the sensor uncertainty.

We assume that the robot state is represented by Gaussian (μ, Σ) - μ being the mean and Σ being the covariance. The systems in our case are generally non-linear, therefore, the EKF linearizes f and h functions at each step. The EKF estimates the state at time t from the estimate at time $t-1$ in two separate steps: process step to propagate the applied control input u_{t-1}, and a measurement step to incorporate the obtained measurements z_t. The process step follows as:

$$\overline{\mu}_t = f(\mu_{t-1}, u_{t-1}) \tag{3}$$

$$\overline{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + V_t W_t V_t^T \tag{4}$$

where G_t and V_t are the Jacobian matrices of f with respect to x and w. Similarly, the measurement step follows as:

$$\mu_t = \overline{\mu}_t + K_t(h(\overline{\mu}_t) - z_t) \tag{5}$$

$$\Sigma_t = \overline{\Sigma}_t - K_t H_t \overline{\Sigma}_t \tag{6}$$

where H_t is the Jacobian of h with respect to x and K_t is known as the Kalman gain,

$$K_t = \overline{\Sigma}_t H_t^T (H_t \overline{\Sigma}_t H_t^T + Q_t)^{-1} \tag{7}$$

Equation 4 propagates the uncertainty in robot state at time $t-1$ from Σ_{t-1} to $\overline{\Sigma}_t$ after incorporating control input and associated motion noise. Equation 6 further propagates it from $\overline{\Sigma}_t$ to Σ_t after incorporating sensor measurements and the associated sensor noise. It is Equation 6 that reduces the uncertainty with the help of meaningful measurements and it is this reduction in uncertainty, from $\overline{\Sigma}_t$ to Σ_t, that we are interested in capturing.

This can be achieved if we assume an a priori uncertainty at each sample point, say a covariance matrix M. Therefore Equations 6 and 7 will change to:

$$\Sigma_n = M - K_n H_n M \tag{8}$$

$$K_n = MH_n^T(H_nMH_n^T + Q_n)^{-1} \tag{9}$$

where subscript n stands for newly sampled robot pose. The localization ability of a sample n is then given by $L_n = \frac{tr(M) - tr(\Sigma_n) \times 100}{tr(M)}$. Since we are using trace[2], therefore, we use a diagonal matrix for M, in fact an identity matrix.

Clearly, L_n in general depends on M. However, Figure 1 empirically shows that the L_n monotonically reduces irrespective of specific M (we chose three arbitrary M's) as the distance of samples from beacons increases, i.e., ability of sensor data to reduce uncertainty is reduced for all three different M. It is this trend that is important.

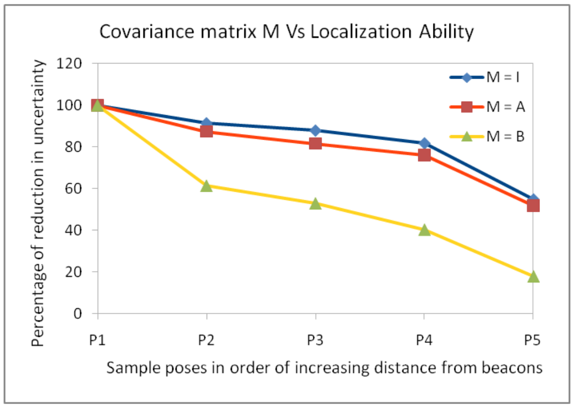

Fig. 1. Covariance matrix (M) Vs Localization ability (L_n) for five sample points (P1 to P5). I = [1, 1, 1], A = [1, 0.15, 1], B = [0.15, 0.15, 1].

As mentioned in the introduction, L_n reflects just the sensor ability to gather accurate information and not the actual localization uncertainty at the sample. The latter also depends on the path chosen and the accuracy of process model and can not be computed at sampling stage. Once the sample is connected to the roadmap, the true belief will be computed by search mechanism (uncertainty propagation from start). L_n is just a measure to accept or reject a sample.

IV. Rapidly-Exploring Random Belief Tree With Localization Aware Sampling Strategy

In this section, we provide a Rapidly-exploring Random Belief Tree with Localization Aware Sampling Strategy (RRBT-LAS) algorithm where we replace uniform sampling of RRBT [13] with our localization aware sampling strategy. Please note that we did not consider linear-quadratic Gaussian (LQG) controller in RRBT-LAS simply because our focus is on showing efficiency of our sampling scheme and if RRBT-LAS is efficient (without LQG) then it will be more efficient after incorporating LQG that requires additional computation along the edges of the roadmap. We construct a roadmap in C-space but the belief paths resulted from the belief propagation do not form a graph rather they form a random tree in belief space, therefore, termed as RRBT.

The algorithm operates on a set of nodes V and edges E, that define a roadmap in state space. Each node $v \in V$ has a state $v.x$, state estimate covariance $v.\Sigma$, a parent node $v.parent$, and localization ability v.loc. The state covariance prediction and chance-constraint checking [13] is implemented by a Propagate(e, v_{start}) routine that takes as arguments an edge and a starting node for that edge, and returns a covariance matrix at the ending node for that edge. If the chance-constraint is violated by the uncertainty at ending node, the function returns no covariance matrix. The comparison of partial paths at a node v is implemented by UpdateBelief(v, Σ) routine that updates the covariance matrix and parent node at v if the new path is less uncertain.

We also require the following routines: Sample() returns i.i.d. uniform samples, Nearest(V, v_{new}) takes the current set of nodes as an argument and returns the node in V that minimizes euclidean distance to v_{new}, and Near(V, v_{new}) returns every node within some ball centered at v_{new} of radius $\rho \propto (\log(n)/n)^{1/d}$ where n is the number of nodes and d is the state dimension (See [20]).

A. Algorithm Description

The RRBT-LAS algorithm is described in Algorithm 1. The roadmap is initialized with a single node with state x_{start}, covariance Σ_0 and its localization ability $tr(M)$ (trace of matrix M) from lines 1-3. At each iteration of the while loop, the roadmap is updated by sampling a new state using our localization aware sampling strategy (line 5), described in Section IV-B, and then adding edges to the nearest and near nodes as in the RRG algorithm [20]. Whenever an edge is added from an existing node to the new node, the existing node is added to the queue (lines 12 and 16). It should be noted that the new node is only added to the roadmap (along with the appropriate edges) if the chance-constraint can be satisfied by propagating an existing belief at the nearest node to the new sampled node as shown on line 8. After all the edges have been added, the queue is exhaustively searched from lines 18-26 using UpdateBelief() routine. Note that the true belief of a node is computed during search mechanism (uncertainty propagation) from lines 20-25 which is different from its localization ability (line 5).

B. Localization Aware Sampling Strategy

Our localization aware sampling strategy puts more samples in regions where sensor data is able to achieve higher uncertainty reduction while maintaining adequate samples in regions where uncertainty reduction is poor. The regions are decided based on a threshold "LocAbilityTH", i.e., if the localization ability of a new uniformly sampled point is above this threshold, then the sample lies in regions with high uncertainty reduction and is simply added as a node. If the localization ability of a sample is below the threshold, the sample lies in regions with low uncertainty reduction, and the decision to accept or reject is governed by the localization ability of neighbouring nodes. If any neighbouring node within a ball of radius "DistTH" centered at the new sample has a localization ability above that of the new sample, the new sample is simply rejected, otherwise it is accepted as

[2] Note that other works have commonly used trace as a measure, however, there are other options as well, for example, one can use determinant.

Algorithm 1: RRBT-LAS Algorithm

```
$v.x := x_{start}$; $v.\Sigma := \Sigma_0$; $v$.parent := NULL
$v$.loc := $tr(M)$
$V := \{v\}$; $E := \{\}$
while $i < P$ do
    $(x_{rand}$, loc_ability$)$ := LocalizationBiasedSample()
    $v_{nearest}$ := Nearest$(V, x_{rand})$
    $e_{nearest}$ := Connect$(v_{nearest}.x, x_{rand})$
    if Propagate$(e_{nearest}, v_{nearest}.\Sigma)$ then
        $v_{rand}$.loc := loc_ability; $v_{rand}.x := x_{rand}$
        $V := V \cup v_{rand}$
        $E := E \cup e_{nearest}$
        $Q := Q \cup v_{nearest}$
        $V_{near}$ := Near$(V, v_{rand})$
        for all $v_{near} \in V_{near}$ do
            $E := E \cup$ Connect$(v_{near}.x, x_{rand})$
            $Q := Q \cup v_{near}$
        end
        while $Q \neq \emptyset$ do
            $u$ := Pop$(Q)$
            for all $v_{neighbor}$ of $u$ do
                $\Sigma'$ :=Propagate$(e_{neighbor}, u.\Sigma)$
                if UpdateBelief$(v_{neighbor}, \Sigma')$ then
                    $Q := Q \cup v_{neighbor}$
                end
            end
        end
    end
    $i := i + 1$
end
```

Algorithm 2: LocalizationBiasedSample()

```
1  $x_{rand}$ := Sample()
2  $L_{x_{rand}}$ :=ComputeLocalizationAbility$(x_{rand})$
3  if $L_{x_{rand}}$ < LocAbilityTH then
4      $V_{neighbor}$ := Neighbor$(V, x_{rand}$, DistTH$)$
5      for all $v_{neighbor} \in V_{neighbor}$ do
6          if $v_{neighbor}$.loc $> L_{x_{rand}}$ then
7              reject sample, go to step 1
8          end
9      end
10 end
11 return $(x_{rand}, L_{x_{rand}})$
```

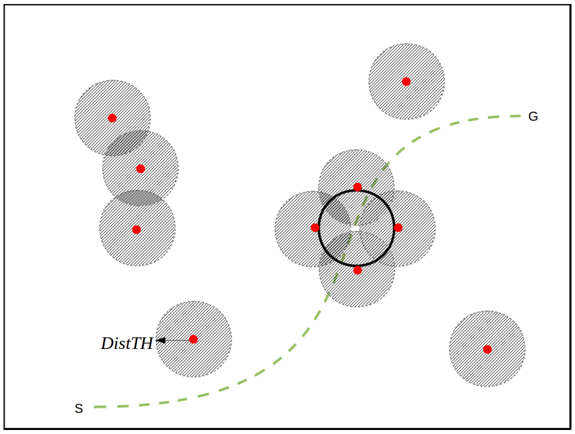

Fig. 2. Black color (bold) circle denotes one of the balls of radius r that is used to tile a path, red color dots represent randomly placed samples, and hatched region (of radius $DistTH = r$) around each sample denotes the restricted region where samples can not be placed according to heuristic used in localization aware sampling. This figure shows the situation where none of the sample has yet been placed inside the black ball, but some samples are placed at a distance d from its center such that $r < d < r + \epsilon$. Even in this worst case scenario, the probability of generating a sample, given by the ratio of volume of white region inside the black ball (after excluding the hatched region) and total volume of white region, is greater than 0. This ratio approaches one as more and more samples are placed outside the black ball.

a node. This localization aware sampling strategy is implemented by routine LocalizationBiasedSample() as described in Algorithm 2.

A main motive behind accepting all the samples which lie within regions with high uncertainty reduction is to favour paths through such regions because they will likely result in high path quality. In worst case, i.e., the only way for a robot is to pass through regions of low uncertainty reduction (as explained in Figure 9 in Section V), path quality will be compromised (as validated in simulations, compromise is small), but at the same time we gain significant savings in planning time. Do note that if we decrease "DistTH" or "LocAbilityTH", our localization aware sampling strategy will converge to uniform sampling. Correspondingly, a higher DistTH results in faster run time, however, to retain probabilistic completeness, there is an upper bound (see Section IV-C).

Furthermore, as we have mentioned before, localization uncertainty also depends on process noise at the sample point, information that is not available at sampling stage since it requires knowledge of path to the sample point. Assuming that the process noise is similar within the neighbourhood of a sample, accepting or rejecting a sample based on L_n (even though we note that localization depends on both steps of EKF) is defensible. It is indeed possible that regions with high uncertainty reduction ability of sensor may also have high process noise, hence the overall uncertainty may still be high. This is mitigated by the search phase of the planner where actual uncertainty is computed and the cost function (based on uncertainty) is minimized. See `http://www.sfu.ca/~vpilania/research.html` for such an example (for lack of space, we have not included it here).

C. Probabilistic Completeness Proof

The probabilistic completeness is along the lines of [21] and is essentially proved by assuming that a collision free path with clearance $\rho \geq 2r$ exists, and then tiling it with a set of carefully chosen balls of radius r such that generating a sample in each ball ensures that these samples can be connected with collision-free edges and therefore, a collision-free path will be found. As shown in Figure 2, we can show that for $DistTH \leq r$, where $2r$ is the radius of the largest inscribed circle within the robot, the probability of generating such samples approaches 1 as the number of

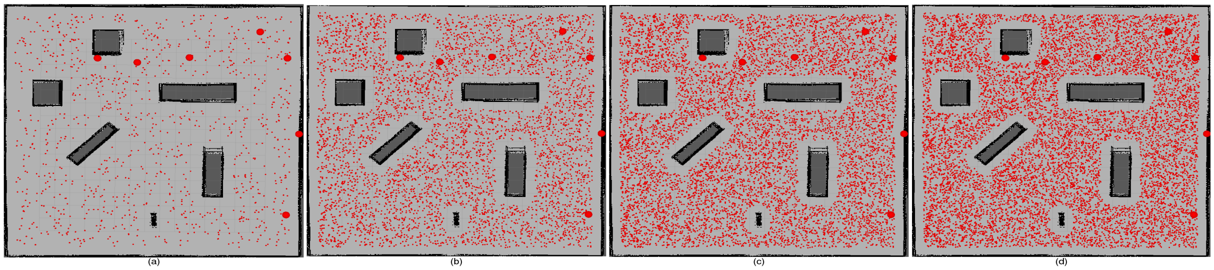

Fig. 3. RRBT [uniform sampling]. # input samples from a seed [# actual nodes in the roadmap] : (a) 1000 [1000], (b) 5000 [5000], (c) 8000 [8000], (d) 10000 [10000]

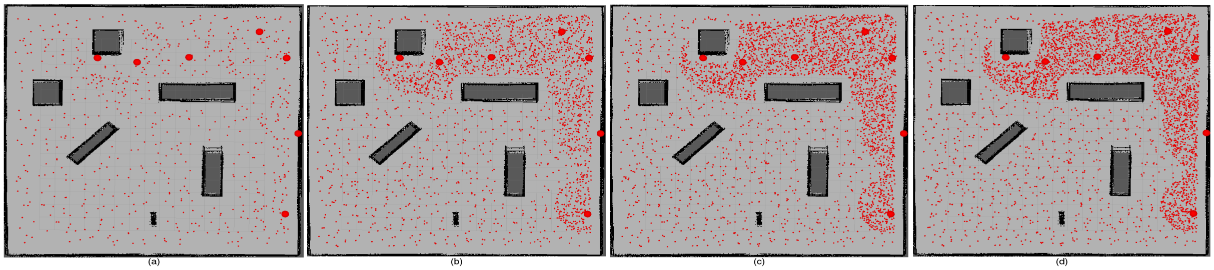

Fig. 4. RRBT-LAS [localization aware sampling] using RangeModel 2. # input samples from a seed [# actual nodes in the roadmap] : (a) 1000 [680], (b) 5000 [2331], (c) 8000 [3347], (d) 10000 [4060] . Here we used DistTH = 30 cm and LocAbilityTH = 90% (reduction in uncertainty).

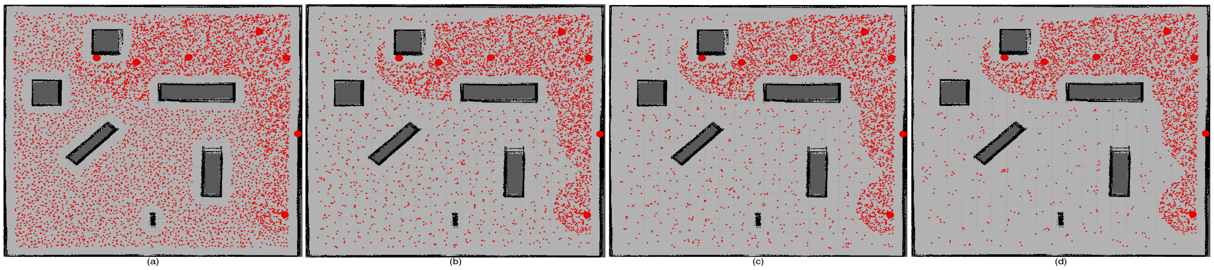

Fig. 5. Effect of varying DistTH (while keeping LocAbilityTH = 90% and the number of input samples from a seed as 10000) in RRBT-LAS using RangeModel 2. DistTH in (a) 10 cm, (b) 30 cm, (c) 40 cm, (d) 60 cm.

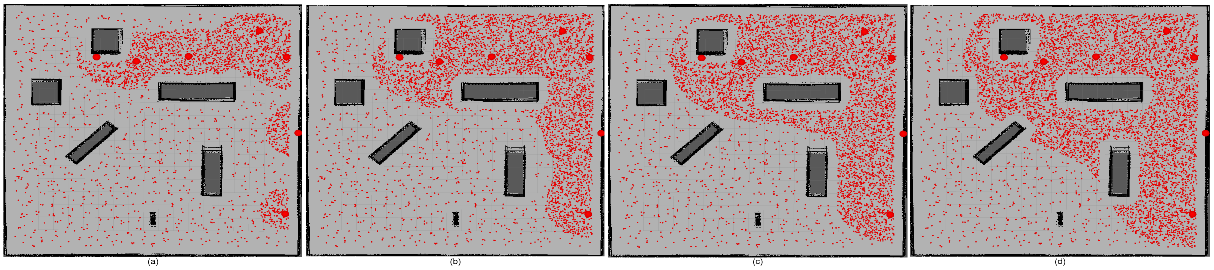

Fig. 6. Effect of varying LocAbilityTH (while keeping DistTH = 30 cm and the number of input samples from a seed as 10000) in RRBT-LAS using RangeModel 2. LocAbilityTH in (a) 93.3%, (b) 86.6%, (c) 83.3%, (d) 76.6%.

samples increases. We show that RRBT-LAS is probabilistically complete under this reasonable restriction of DistTH. The complete proof can be accessed online at `http://www.sfu.ca/~vpilania/research.html`.

V. Experimental Results

We compared RRBT-LAS algorithm with RRBT. We used the motion model and sensor model from [11] for both the planners. For sensor model, the beacons have range that spans the entire map with a distance varying Gaussian noise in the sensor data. We denote this sensor model as RangeModel 2. Please note that our localization aware sampling strategy also holds for complex measurement models, for example: range sensors.

We used 30 different seeds, each seed generating a set of 10000 pseudo random collision-free input samples. We ran both the planners on each set by varying the number of input samples from 100 to 10000 in incremental manner and provided our simulation results by averaging the outcome over 30 sets. Our implementation is in C++ under linux and

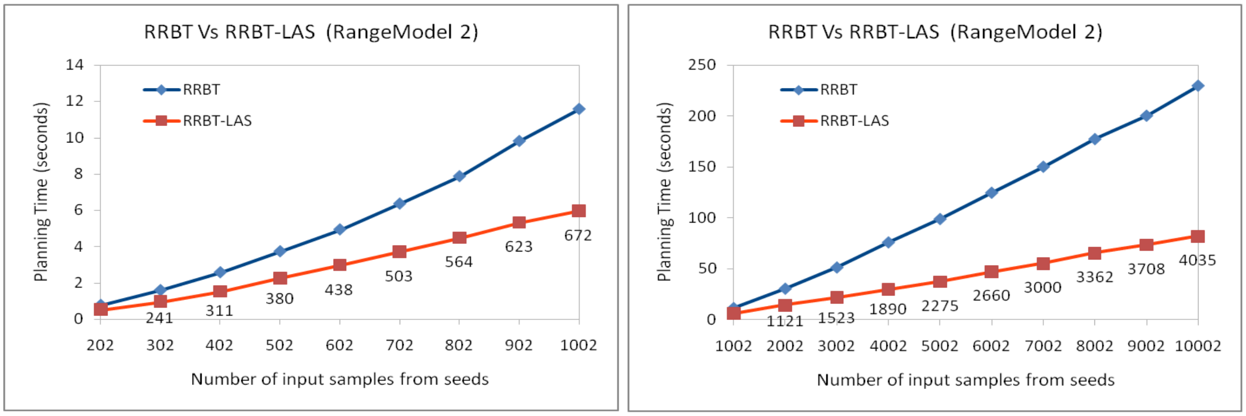

Fig. 7. Plots show the comparison of RRBT Vs RRBT-LAS for incremental motion planning. Data labels for each data point along red curves in RRBT-LAS show the actual number of nodes in the roadmap. For RRBT, actual number of nodes and number of input samples from seeds are the same, therefore, data labels are not shown along their corresponding curves. For the plots we used DistTH = 30 cm and LocAbilityTH = 86.6%. Also note that the saving in planning time is for $DistTH \leq r$ (where $2r$ is the inscribed radius of robot).

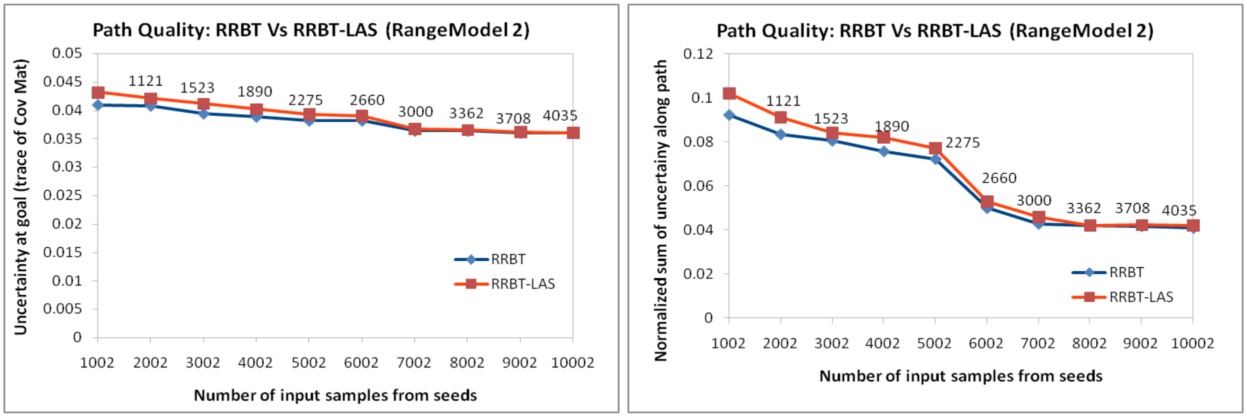

Fig. 8. Comparison of path quality between RRBT and RRBT-LAS for a scenario where the only path to goal passes through regions with low uncertainty reduction, essentially a worst case scenario for path quality for our sampling scheme (see Figure 9). Y-axis denotes the trace of covariance matrix at goal in left plot and normalized sum of trace of covariance matrices along path in right plot.

runs on a Pentium dual core 2.5 Ghz computer with 4GB memory.

We demonstrate our approach in two ways: (i) through visualization in Figures 3 - 6, we show the efficacy of our localization aware sampling strategy in judiciously placing the samples, and (ii) we use plots in Figures 7 and 8 to show that our localization aware sampling leads to saving in planning time with little compromise on the quality of path.

Figure 3 shows the placement of nodes in the roadmap (edges are not shown) for uniform sampling as we increase the number of input samples from a seed. The uniform sampling strategy is not aware of sensor model, therefore, the actual number of nodes in the roadmap are equal to the number of input samples added from a seed. Big red color balls (7 of them) in the snapshots denote the beacons which were used for localization. Compared to uniform sampling (Figure 3), the actual number of nodes in our localization aware sampling strategy are significantly reduced as shown in Figure 4. From the figures, we can observe that our localization aware sampling strategy places more samples in regions with high uncertainty reduction (near beacons) and eliminates unnecessary samples from regions with low uncertainty reduction. We also show the effect of varying two thresholds (DistTH and LocAbilityTH) in our localization aware sampling strategy. In Figure 5, we varied only DistTH and observed that the sparsity of nodes in regions with low uncertainty reduction increases with the increase of DistTH. However, the nodes in regions with high uncertainty reduction remain unchanged with the variation of DistTH. Similarly, in Figure 6, we varied only LocAbilityTH and observed that the area under regions with high uncertainty reduction increases with the decrease of LocAbilityTH. In Figures 5 and 6, we kept the number of input samples from a seed as 10000, therefore, the comparison should be done with Figure 3(d).

In Figure 7, from the comparison of RRBT with RRBT-LAS, we observed that RRBT-LAS reduces the planning time significantly as a result of our localization aware sampling strategy. This can be seen from the graphs as we move from 200 input samples to 10000 input samples (along the x axis). We also observed that the run time savings increase supra linearly with the number of input samples. Please note that the planning time difference at 1002 samples in right plot can be observed in left plot.

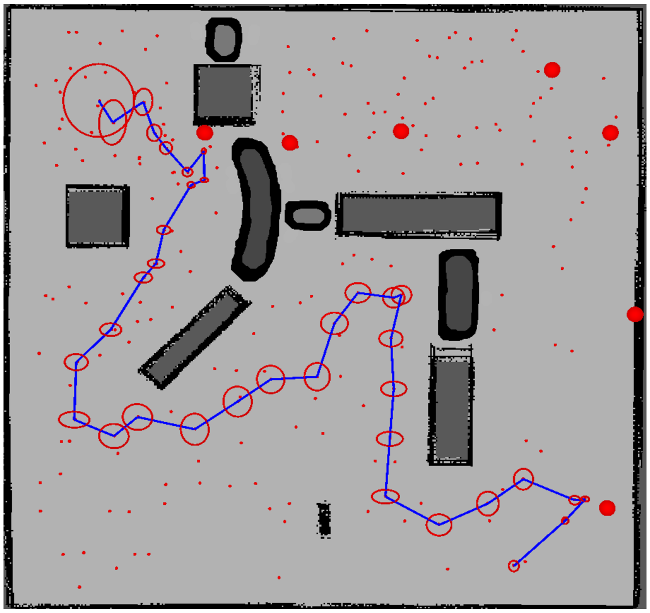

Fig. 9. It demonstrates the importance of maintaining an adequate number of samples in regions with low uncertainty reduction. In the figure, the regions with high uncertainty reduction are obstructed by obstacles, therefore a path was found which most of the time passes through regions with low uncertainty reduction (away from beacons). The red color ellipses show the uncertainty at waypoints along the path.

Furthermore, we compared the quality of paths generated by RRBT and RRBT-LAS. We used two comparison metrics: (a) trace of covariance matrix at goal, (b) normalized sum of trace of covariance matrices along path. For worst case scenario, where the only way for the robot is to pass through regions with low uncertainty reduction (as shown in Figure 9), we observed little compromise on the quality of paths generated by RRBT-LAS as compared to uniform sampling of RRBT. Plots in Figure 8 show the comparison. At 1000 input samples, RRBT-LAS has 10% more uncertainty along path and 5% more uncertainty at goal. This is the highest degradation in path quality that we observed as we move from 200 input samples to 10000 input samples. Note that the path quality saturates at about 7000 samples for the considered case. The small compromise on path quality comes from the fact that, for this scenario, the path passes through regions with low uncertainty reduction where we reduce the number of samples. For a large majority of scenario, only portions of a path pass through regions with low uncertainty reduction, the path quality compromise would be much smaller. This is the key reason that our localization aware sampling strategy simply accepts all the samples within regions with high uncertainty reduction (as shown in Figures 4 to 6). Due to space limitation, we provide other plots (from 200 input samples to 1000 input samples) at `http://www.sfu.ca/~vpilania/research.html`.

Additionally, we also evaluated all the planners with a different sensor model (we call it RangeModel 1) where beacons have a limited range (we used 2 meters). We observed similar behaviour as with RangeModel 2 and the corresponding plots can be accessed online at `http://www.sfu.ca/~vpilania/research.html`.

VI. Conclusion

We presented a novel sampling strategy for sampling-based stochastic motion planners that judiciously places the samples using a new notion of localization ability of a sample, i.e., it puts more samples in regions where sensor data is able to achieve higher uncertainty reduction while maintaining adequate samples in regions where uncertainty reduction is poor. An important aspect of our work is a new measure, the "Localization Ability of a sample" that captures the ability of sensor data in reducing the uncertainty at the sample point without actually knowing the path leading to it. We show that a stochastic planner that uses our sampling strategy is probabilistically complete for DistTH less than or equal to half of the inscribed radius of the robot. Robots of larger size will benefit more from the sampling scheme while maintaining probabilistic completeness.

We implemented incremental (RRBT and RRBT-LAS) stochastic motion planners and demonstrated our localization aware sampling strategy using two type of sensor models: RangeModel 1 and RangeModel 2. We empirically show that the planners with our sampling strategy take less time to find a path with little compromise in path quality.

References

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

[2] J. Pineau, G. Gordon, and S. Thrunn, "Point-based value iteration: An anytime algorithm for POMDPs," in *International Joint Conferences on Artificial Intelligence*, 2003, pp. 1025–1032.

[3] H. Kurniawati, Y. Du, D. Hsu, and W. S. Lee, "Motion planning under uncertainty for robotic tasks with long time horizons," in *Proc. of the International Symposium on Robotics Research*, 2009.

[4] H. Kurniawati, T. Bandyopadhyay, and N. Patrikalakis, "Global motion planning under uncertain motion, sensing, and environment map," *Autonomous Robots*, vol. 33, no. 3, pp. 255–272, 2012.

[5] B. Bouilly, T. Simeon, and R. Alami, "A numerical technique for planning motion strategies of a mobile robot in presence of uncertainty," in *Proc.of the IEEE International Conference on Robotics and Automation (ICRA)*, May 1995, pp. 1327–1332.

[6] A. Lazanas and J.-C. Latombe, "Motion planning with uncertainty: a landmark approach," *Artificial Intelligence*, vol. 76, no. 1-2, 1995.

[7] T. Fraichard and R. Mermond, "Path planning with uncertainty for car-like robots," in *Proc.of the IEEE International Conference on Robotics and Automation (ICRA)*, May 1998, pp. 27–32.

[8] A. Lambert and D. Gruyer, "Safe path planning in an uncertain-configuration space," in *Proc.of the IEEE International Conference on Robotics and Automation (ICRA)*, Roma, Italy, 2003, pp. 4185–4190.

[9] N. A. Melchior and R. Simmons, "Particle RRT for path planning with uncertainty," in *Proc.of the IEEE International Conference on Robotics and Automation (ICRA)*, Roma, Italy, 2007, pp. 1617–1624.

[10] Y. Huang and K. Gupta, "RRT-SLAM for motion planning with motion and map uncertainty for robot exploration," in *Proc.of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2008, pp. 22–26.

[11] S. Prentice and N. Roy, "The belief roadmap: Efficient planning in belief space by factoring the covariance," *The International Journal of Robotics Research*, vol. 28, no. 11-12, pp. 1448–1465, July 2009.

[12] J. van den Berg, P. Abbeel, and K. Goldberg, "LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 895–913, 2011.

[13] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *Proc.of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2011, pp. 723–730.

[14] A. Agha-mohammadi, S. Chakravorty, and N. Amato, “FIRM: Sampling-based feedback motion planning under motion uncertainty and imperfect measurements,” *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 268–304, 2014.

[15] S. Karaman, M. Walter, A. Perez, E. Frazzoli, and S. Teller, “Anytime motion planning using the RRT*,” in *Proc.of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[16] D. Hsu, J. C. Latombe, and H. Kurniawati, “On the probabilistic foundations of probabilistic roadmap planning,” *International Journal of Robotics Research (IJRR)*, vol. 25, no. 7, pp. 627–643, 2006.

[17] R. A. Knepper and M. T. Mason, “Real-time informed path sampling for motion planning search,” *International Journal of Robotics Research (IJRR)*, vol. 31, no. 11, pp. 1231–1250, 2012.

[18] P. E. Missiuro and N. Roy, “Adapting probabilistic roadmaps to handle uncertain maps,” in *Proc.of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2006, pp. 1261–1267.

[19] C. Stachniss, G. Grisetti, and W. Burgard, “Information gain-based exploration using rao-blackwellized particle filters,” in *Proc.of the Robotics: Science and Systems (RSS)*, 2005, pp. 65–72.

[20] S. Karaman and E. Frazzoli, “Incremental sampling-based optimal motion planning,” in *Proc.of the Robotics: Science and Systems (RSS)*, 2010.

[21] 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*. Cambridge, MA: MIT Press, 2005, ch. 7, pp. 242–246.