High-Frequency Replanning Under Uncertainty Using Parallel Sampling-Based Motion Planning

Wen Sun, Sachin Patil, *Student Member, IEEE,* and Ron Alterovitz, *Member, IEEE*

***Abstract*—As sampling-based motion planners become faster, they can be re-executed more frequently by a robot during task execution to react to uncertainty in robot motion, obstacle motion, sensing noise, and uncertainty in the robot's kinematic model. We investigate and analyze high-frequency replanning (HFR), where during each period fast sampling-based motion planners are executed in parallel as the robot simultaneously executes the first action of the best motion plan from the previous period. We consider systems with linear and nonlinear (but linearizable) dynamics in which controls are executed at discrete time points, uncertainty can be modeled using Gaussian distributions, and the objective is to maximize the probability of success or minimize path length subject to a chance constraint. We show that, as parallel computation power increases, HFR offers asymptotic optimality during each period for goal-oriented problems for uncertainty-aware cost metrics. We then demonstrate the effectiveness of HFR for holonomic and nonholonomic robots including car-like vehicles and steerable medical needles.**

***Index Terms*—motion and path planning, nonholonomic motion planning, sampling-based methods.**

I. Introduction

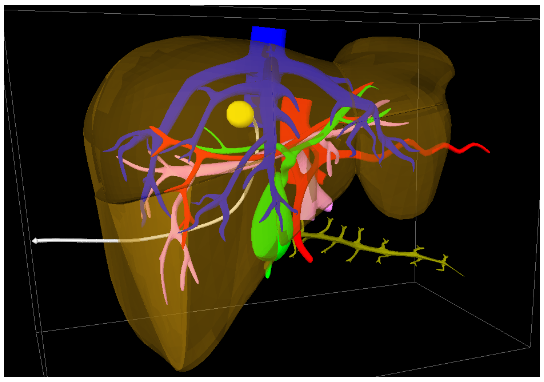
As a robot moves through an environment to accomplish a task, uncertainty may arise from a variety of real-world sources such as unpredictable robot actuation, sensing errors, errors in modeling robot motion, and unpredictable movement of obstacles in the environment. The cumulative effect of all these sources of uncertainty can be difficult to model and account for in the planning phase before task execution. We leverage the increasingly fast performance of sampling-based planners available for certain robots, combined with stochastic modeling, to enable these robots to quickly respond to uncertainty during task execution in an asymptotically globally optimal manner.

In this paper we consider tasks in which the objective is to maximize the probability of success (i.e., avoid collision with obstacles and reach the goal), or to minimize path length subject a constraint on the probability of success. We consider holonomic and nonholonomic robots with linear or nonlinear (but linearizable) discrete-time dynamics and with motion and sensing uncertainty that can be modeled as Gaussian distributions. We also assume obstacle locations are known or can be sensed. Our approach is applicable to robots for which fast, subsecond sampling-based motion planners are available, including medical steerable needles [28, 29] (see Fig. 1).

We investigate and analyze the approach of high frequency replanning (HFR), where the robot executes a motion plan that is updated periodically with high frequency by executing fast sampling-based motion planners in parallel during each period. The replanning computation occurs as the robot is executing the current best plan. If any of the computed plans is better than the plan currently being executed, then the current plan is updated. To select the best plan, we focus on uncertainty-aware cost metrics that explicitly consider the probability of success. We note that for these cost metrics, previous optimal motion planners such as RRT* and related variants [15] cannot be used since the optimal substructure property does not hold. This is because, as shown in Fig. 2, the optimal plan from any configuration to a goal is dependent on the plan chosen to reach that configuration [38]. This substantially complicates the motion planning challenge for the uncertainty-aware cost

Wen Sun and Ron Alterovitz are with the Department of Computer Science, University of North Carolina at Chapel Hill, North Carolina, USA. e-mail: wens, ron@cs.unc.edu

Sachin Patil is with the University of California, Berkeley, USA. e-mail: sachinpatil@berkeley.edu


(a) Needle steering in the liver using HFR (front view)

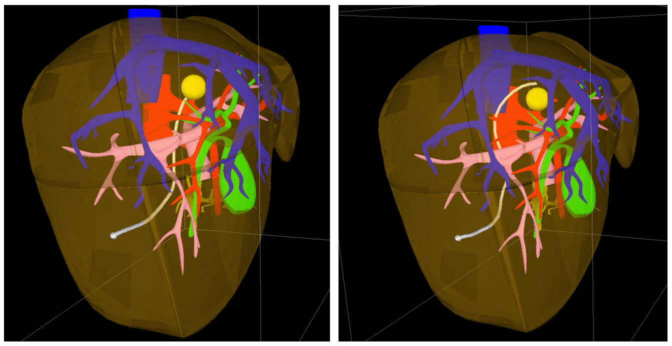

(b) HFR (side view) (c) LQG-MP (side view)

Fig. 1. We apply high frequency replanning (HFR) to medical needle steering in the liver for a biopsy procedure. The objective is to access the tumor (yellow) while avoiding the hepatic arteries (red), hepatic veins (blue), portal veins (pink), and bile ducts (green). To reach the tumor, the needle (white) must pass through two types of tissue: muscle/fat tissue and then liver tissue. The maximum curvature of the needle in the liver is not known *a priori* and must be estimated during task execution. Using HFR (a, b), in simulation we successfully guided the needle to the tumor by estimating online the needle's maximum curvature and replanning with high frequency. Using an approach based on preplanning and LQG control (LQG-MP) (c), the needle veered away from the goal and the LQG controller was unable to correct the trajectory.

Preprint submitted to IEEE Transactions on Robotics. Received: May 29, 2014 09:47:12 PST

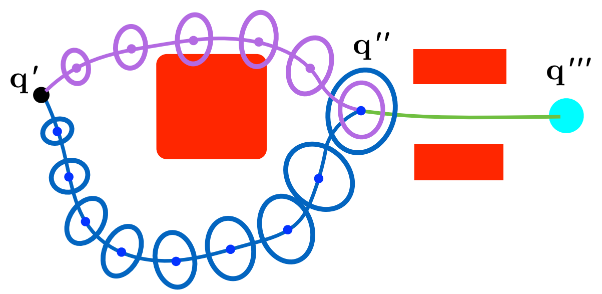


Fig. 2. Consider two paths from configuration $\mathbf{q}'$ to a configuration $\mathbf{q}'''$, with ellipsoids representing uncertainty at stages along the paths. RRT* and related asymptotically optimal motion planners assume that the optimal substructure property holds, i.e., if $\mathbf{q}''$ is along the optimal path from $\mathbf{q}'$ to $\mathbf{q}'''$, the subpath from $\mathbf{q}'$ to $\mathbf{q}''$ is itself an optimal path from $\mathbf{q}'$ to $\mathbf{q}''$. While this is true for shortest path problems, optimal substructure does not hold for uncertainty-aware cost metrics such as maximizing probability of success. Between $\mathbf{q}'$ and $\mathbf{q}''$, the probability of the collision of the purple path is larger than the probability of collision of the blue path due to greater clearance from the obstacles. But the purple plan has lesser estimated uncertainty in state estimation at $\mathbf{q}''$, which will lead to a smaller probability of collision when the robot passes through the narrow passage between $\mathbf{q}''$ and $\mathbf{q}'''$. The lack of optimal substructure when planning in configuration space makes achieving asymptotic optimality more difficult.

metrics considered in this paper.

HFR combines the benefits of global motion planning with the responsiveness of a controller. By executing multiple sampling-based rapidly-exploring random tree (RRT) [19] motion planners in parallel, under certain reasonable assumptions HFR will compute at each a period a motion plan that asymptotically approaches the globally optimal plan as computation power (i.e., processor speed and number of cores) increases. For practical problems with finite computational resources, we add a heuristic to HFR to consider newly generated plans as well as the best plan from the prior period adjusted based on the latest sensor measurements and a linear feedback controller. Asymptotic optimality in each period of HFR comes not from generating more configuration samples in a single RRT but rather from generating many independent RRTs (where each RRT terminates when it finds plan and launches a new RRT to begin construction). Another benefit of HFR is that it automatically handles control input bounds since the sampling-based motion planner enforces kinematic and dynamic constraints during plan generation. HFR only applies to goal-oriented problems that do not require returning to previously explored regions of the state space for information gathering, and hence does not address the general POMDP problem [18].

This paper makes three main contributions. First, we show that generating multiple RRTs and selecting the one that optimizes the chosen metric is asymptotically optimal under reasonable assumptions. Asymptotic optimality is important for many applications that require high quality solutions, and gives the user confidence that increasing the computational hardware applied to the problem will result in better and better solutions that improve toward optimality. Second, we present an approach for handling uncertainty by using high-frequency replanning that considers the impact of future uncertainty and uses a heuristic to efficiently adjust the prior best plan based on the latest sensor measurements. Third, we experimentally show that HFR is effective for many problems with uncertainty in motion, sensing, and kinematic model parameters. Evaluation scenarios include a holonomic disc robot navigating in a dynamic environment with moving obstacles whose motion are not known *a priori*, a nonholonomic car-like robot maneuvering with uncertainty in motion and sensing, and a nonholonomic steerable medical needle [36] maneuvering through tissues around anatomical obstacles to clinical targets under uncertainty in motion, sensing, and kinematic modeling.

II. Related Work

Motion planning under uncertainty has received considerable attention in recent years. Approaches that blend planning and control by defining a global control policy over the entire environment have been developed using Markov decision processes (MDPs) [2] and partially-observable MDPs (POMDPs) [18]. These approaches are difficult to scale, and computational costs may prohibit their application to robots navigating using non-discrete controls in uncertain, dynamic environments. Sampling-based approaches that consider uncertainty [1, 4, 12, 24, 34] or approaches that compute a locally-optimal trajectory and an associated control policy [33, 39, 42] are effective for a variety of scenarios but are currently not suitable for real-time planning in uncertain, dynamic, and changing environments. Approaches have been proposed to efficiently estimate the probability of collision of plans for evaluating the quality of motion plans under Gaussian models of uncertainty [8, 30, 38]. Our approach combines ideas from efficient sampling-based motion planning and planning under uncertainty to effectively plan in a manner that considers uncertainty and changing environments.

An alternative to feedback control is to continuously replan to account for changes in the robot state or the environment. Several approaches have been suggested for planning in dynamic environments [13, 17, 31, 41, 44, 45] but they do not explicitly account for robot motion and sensing uncertainty. Majumdar et al. [23] use a pre-defined library of motion primitives, which depending on the application can be effective or restrictive. Our work is also related to model predictive control (MPC) approaches [35], which account for state and control input constraints in an optimal control formulation. HFR can be viewed as a variant of MPC control where the goal is always within the horizon and global optimality is achieved (in an asymptotic sense) at each time step using a sampling-based motion planner. In this regard, Du Toit [8] introduced a new method for planning under uncertainty in dynamic environments by solving a nonconvex optimization problem, although this method for some problems is computationally expensive and suffers from local minima. In contrast, HFR is based on an efficient motion planner that takes uncertainty into account, resulting in higher probabilities of successful plan execution.

Asymptotically optimal motion planners such as RRT* and related variants [15, 21] return plans that converge toward optimality for cost functions that satisfy certain criteria. In this paper we focus on uncertainty-aware cost metrics (e.g., maximizing probability of success), for which the optimality substructure does not hold, e.g., the optimal solution starting from any robot configuration is not independent of the prior history of the plan (see Fig. 2). Consequently, RRT* and related variants cannot guarantee asymptotic optimality for our problem without modification. A recently developed approach CC-RRT* [22] utilizes RRT* to generate asymptotic optimal motion plan that satisfies user defined chance constraints. Each node of the CC-RRT* tree explicitly stores all the information (a sequence of states and uncertainty distributions) of the prior path that leads to the state in the node. Hence CC-RRT* obtains the optimal substructure. Unlike our approach, CC-RRT* does not consider sensing uncertainty and hence grows the uncertainty distributions in an open-loop fashion by simply evolving the stochastic dynamics forward in time. In HFR, we explicitly consider the motion and sensing uncertainty and grow the uncertainty distributions in a closed-loop fashion assuming that the LQG control and the Extended Kalman Filter are used.

HFR takes advantage of the fact that the speed and effectiveness of sampling-based planning algorithms [6, 20] has improved substantially over the past decade due to algorithmic improvements and innovations in hardware. One source of speedup has been parallelism, including the use of multiple processors (e.g., [5, 7, 14, 32]) and GPUs (e.g., [3, 27]) to achieve speedups for single-query sampling-based motion planners (e.g., RRT and RRT*). The idea of constructing multiple sampling-based search trees across multiple CPUs has been used for testing hybrid systems [10] and for speeding up shortest path motion planning [26] by building search trees on multiple CPUs and then allowing the CPUs to exchange information. Wedge and Branicky showed that periodically restarting sampling-based tree construction can improve the mean and the variability of the runtime of RRT [43]. Our method extends the OR RRT approach [5, 7] to solve a class of problems with asymptotic optimality rather than solely feasibility. Also, unlike prior approaches that leverage parallelism, we explicitly focus on motion planning problems that involve uncertainty in robot motion and sensing and moving obstacles.

III. Approach

A. Problem Statement

Let $\mathbf{q} \in \mathcal{Q} \subset \mathbb{R}^l$ denote the robot state which consists of parameters over which the robot has control, such as the robot's position, orientation, and velocity. Let $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n,\ n \geq l$, denote the system state which includes the robot state and also relevant parameters that can be sensed, such as obstacle positions and velocities, as well as robot-specific parameters over which the robot does not have direct control, such as robot kinematic parameters (e.g., a steerable needle's maximum curvature in a particular tissue). Note that the robot's state $\mathbf{q}$ is a subset of the whole system's state $\mathbf{x}$. We define the initial state of the system as $\mathbf{x}_0$.

We assume that continuous time τ is discretized into periods of equal duration Δ such that the t'th period begins at time $\tau = t\Delta$. A motion plan π is then defined by a sequence of discrete control inputs $\pi = \{\mathbf{u}_t : t = 0, \ldots, T\}$, where $\mathbf{u}_t \in \mathcal{U} \subset \mathbb{R}^m$ is a control drawn from the space of permissible control inputs. Starting from $\mathbf{x}_0$ and applying a control sequence π with number of steps T, the state of the robot at time τ, where $0 \leq \tau \leq T\Delta$, is expressed as $\mathbf{x}(\tau, \pi)$. For simplicity, we define $\mathbf{x}_t = \mathbf{x}(\tau, \pi)$, when $\tau = t\Delta$.

The whole system evolves according to a stochastic dynamics model:

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{m}_t), \qquad (1)$$

where $\mathbf{m}_t$ models the cumulative uncertainty, including uncertainty in robot and obstacle motion. We assume $\mathbf{m}_t$ is from a Gaussian distribution $\mathbf{m}_t \sim N(\mathbf{0}, \mathbf{M}_t)$ with variance $\mathbf{M}_t$.

During task execution, the robot obtains sensor measurements according to the stochastic observation model:

$$\mathbf{z}_t = h(\mathbf{x}_t, \mathbf{n}_t) \qquad (2)$$

where $\mathbf{z}_t$ is the vector of measurements and $\mathbf{n}_t$ is noise which is from a Gaussian distribution $\mathbf{n}_t \sim N(\mathbf{0}, \mathbf{N}_t)$ with variance $\mathbf{N}_t$.

In addition, our method also requires as input an estimate of the initial system state $\hat{\mathbf{x}}_0$ with corresponding variance $\hat{\mathbf{\Sigma}}_0$, a cost function $c(\pi, \mathbf{x})$ that defines the expected cost of a motion plan π from an initial system state $\mathbf{x}$, the period duration Δ, a set of obstacles $\mathcal{O}$, and a goal region $\mathcal{G}$ such that $\mathbf{x}_t \in \mathcal{G}$ signifies success at time step t. We focus on two optimization objectives:

1) Maximize the probability of success, i.e., the probability of avoiding collision with obstacles and reaching the goal. We define the cost function $c(\pi, \mathbf{x})$ as the negative of the probability of success estimated for plan π.
2) Minimize path length subject to a chance constraint (in our case, a lower bound on the probability of success). We define $c(\pi, \mathbf{x})$ as the length of the nominal plan resulting from executing π if the path satisfies the chance constraint and as ∞ if the plan violates the constraint.

Our objective is to compute and execute a control input $\mathbf{u}_t$ at each time step to optimize an uncertainty-aware cost metric.

B. High-Frequency Replanning (HFR) Algorithm Overview

We outline HFR in Algorithm 1. The outer loop of the HFR algorithm (lines 3-17) runs at a high frequency with a period of Δ to address uncertainty in robot motion, obstacle motion, and the robot's kinematic model. In each period, the bulk of the computation is computing a motion plan. In each period, the robot estimates the system state using a filter, updates the current motion plan with a better motion plan if a better plan is found, and then executes the first control input of the current best plan. We assume that except for the inner loop at line 7, other steps are sufficiently fast for real-time computing.

C. Parallel Sampling-Based Replanning

For motion planning, we use a rapidly-exploring random tree (RRT), a well-established sampling-based motion planner that, when properly implemented, provides probabilistic completeness guarantees that a plan will be found if one exists [20]. In HFR, at time step t the RRT is rooted at the system state $\tilde{\mathbf{x}}_{t+1}$, the estimated state of the robot at the next time step (as discussed in Section III-D). At each iteration of the RRT algorithm, we sample a robot state

Algorithm 1: HFR Algorithm

```
Input: $(\hat{\mathbf{x}}_0, \hat{\mathbf{\Sigma}}_0)$: initial estimate of system state; $f(\cdot)$:
       system dynamics; $c(\cdot)$: cost function; $\Delta$: planning
       interval; $\mathcal{O}$: obstacles; $\mathcal{G}$: goal region;
1  $t := 0$
2  $\pi^*$ := initial plan from $\hat{\mathbf{x}}_0$ (found by multiple parallel
   RRTs or some other motion planner)
3  repeat
4      $(\tilde{\mathbf{x}}_{t+1}, \tilde{\mathbf{\Sigma}}_{t+1})$ := predict the system state at time $t+1$
5      $\Pi := \emptyset$
6      execute $\pi^*$ up to $\Delta$ time while simultaneously
       computing the following loop for $\Delta$ time:
7          do in parallel on any available processor core
8              $\pi'$ := first plan found by RRT from $\tilde{\mathbf{x}}_{t+1}$
9              $\Pi := \Pi \cup \pi'$
10         loop
11     end
12     $(\hat{\mathbf{x}}_{t+1}, \hat{\mathbf{\Sigma}}_{t+1})$ := filter update at time $t+1$
13     $\pi'$ := adjust plan $\pi^*$ to start from $\hat{\mathbf{x}}_{t+1}$ (optional)
14     $\Pi' := \Pi' \cup \pi'$ (optional)
15     $\pi^* := \mathrm{argmin}_{\pi \in \Pi'} c(\pi, \hat{\mathbf{x}}_{t+1})$
16     $t := t+1$
17 until $\mathbf{x}_t \in \mathcal{G}$ or robot cannot avoid collision;
```

$\mathbf{q}_{\text{sample}} \in \mathcal{Q}$, find its nearest neighbor $\mathbf{q}_{\text{near}}$ in the tree, and compute a control $\mathbf{u} \in \mathcal{U}$ that grows the tree from $\mathbf{q}_{\text{near}}$ toward $\mathbf{q}_{\text{sample}}$ using the RRT's "extend" function, i.e., $(\mathbf{q}_{\text{new}}, \mathbf{u}) = \text{extend}(\mathbf{q}_{\text{near}}, \mathbf{q}_{\text{sample}})$ where $\mathbf{q}_{\text{new}}$ requires no more than Δ time to reach from $\mathbf{q}_{\text{near}}$ [19]. Although motion planning is conducted in the robot's state space $\mathcal{Q}$, collision detection may depend on the system state $\mathbf{x}_t \in \mathcal{X}$, especially for cases with moving obstacles. For any $\mathbf{q}_t \in \mathcal{Q}$, we can estimate the corresponding $\mathbf{x}_t \in \mathcal{X}$ by augmenting the robot state vector at time t to incorporate missing entries from the prior system state and then evolving the vector through the stochastic dynamics function with zero noise. The output of the RRT is a motion plan, $\pi = [\mathbf{u}_{t+1}, \mathbf{u}_{t+2}, \ldots, \mathbf{u}_T]$, that specifies a sequence of control inputs to reach the goal.

During each period, our objective is to compute a motion plan that minimizes the cost function c. Prior asymptotically optimal sampling-based methods (e.g., [15]) are not suitable for cost metrics such as maximizing probability of success for which the optimal substructure property does not hold in configuration space, as shown in Fig. 2.

Our approach to finding an optimal plan is to simultaneously execute a large number of independent RRTs and then select the RRT with the lowest cost plan. By evaluating the cost metric over an entire plan we avoid requiring the optimal substructure property. We refer to lines 7–9 and line 15 as Multiple Parallel RRTs (MPRRT). Specifically, during each period of duration Δ, we execute independent RRTs in parallel on each available processor core. As soon as an RRT finds its first solution, we add the plan to a set Π (line 9) and start executing a new independent RRT on that processor core. As parallel computation power increases, the number of RRT solutions obtained in time Δ rises. We note that any individual RRT, with probability 1, will not compute an optimal plan for a general motion planning problem regardless of how many robot state samples are generated [15]. We show in Sec. IV that, for goal-oriented problems with a finite (although not necessarily known) number of periods, the plan computed by MPRRT will asymptotically approach the optimal plan as computational power increases.

D. Predicting System State for Motion Planning

Since the robot is in motion during replanning and we desire a full period of time Δ for replanning, the computation of the motion plan for period $t+1$ must occur during period t. Hence, at the beginning of period t we must predict the system state at the next time step $\tilde{\mathbf{x}}_{t+1}$, which is used to seed the RRTs. Due to uncertainty, the exact state of the system at time $(t+1)\Delta$ cannot be perfectly predicted at time $t\Delta$. To estimate $\tilde{\mathbf{x}}_{t+1}$, we evolve the estimated system state distribution using the system dynamics function (Eq. 1). More formally, we compute $\tilde{\mathbf{x}}_{t+1} = f(\hat{\mathbf{x}}_t, \mathbf{u})$, where $\mathbf{u}$ is the first control input of the current plan π^* that is being executed. To represent the uncertainty distribution, we truncate the portion of the Gaussian distribution of system state in collision with obstacles [30].

Near the end of each time step t (line 12), we obtain sensor measurement $\mathbf{z}_{t+1}$ and estimate the system state $\hat{\mathbf{x}}_{t+1}$ for the next time step. In our implementation, we use an Extended Kalman Filter (EKF). Since the estimated state $\hat{\mathbf{x}}_{t+1}$ may not equal the state $\tilde{\mathbf{x}}_{t+1}$ that was predicted at the beginning of time step t (line 4), the RRT plans may be rooted at a state that is slightly incorrect. Since HFR plans at high frequency and Δ is small, the deviation in the initial system state in each RRT is expected to be small. We analyze this difference in Sec. IV.

E. Optional Heuristic for Adjusting the Prior Best Plan

To improve HFR performance when limited finite computation power is available, we include the prior best motion plan in the set of motion plans considered by the optimization in line 15. However, the prior best plan may have been computed multiple periods earlier, and hence did not consider any of the sensed information from recent periods. For example, the system may be at a significantly different state from the state implied by the prior best motion plan without any adjustments. To incorporate the recently sensed information into the prior best plan in an effective manner, in line 13 of Algorithm 1 we adjust the prior best motion plan to start from $\tilde{\mathbf{x}}_{t+1}$ and adjust the control inputs to the values that would most likely have been performed by a linear quadratic Gaussian (LQG) controller used in conjunction with a Kalman filter.

Specifically, given a prior motion plan $\pi = [\tilde{\mathbf{u}}_{t+1}, \ldots, \tilde{\mathbf{u}}_T]$ from the predicted system state $\tilde{\mathbf{x}}_{t+1}$, the corresponding nominal trajectory of the RRT plan based on the system dynamics with zero noise is given by $\Gamma = [\tilde{\mathbf{x}}_{t+1}, \tilde{\mathbf{u}}_{t+1}, \ldots, \tilde{\mathbf{x}}_T, \tilde{\mathbf{u}}_T]$. We linearize the dynamics and sensing functions around the trajectory Γ and compute an LQG control policy and a Kalman filter [4, 38]. This computation provides a sequence of feedback matrices $\{\mathbf{L}_t\}$ and a sequence of Kalman gains $\{\mathbf{K}_t\}$. We then compute the maximum likelihood simulated trajectory (MLST) as

$$\mathbf{x}'_{t+2} = f(\mathbf{x}'_{t+1}, \mathbf{L}_{t+1}(\hat{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_{t+1}) + \tilde{\mathbf{u}}_{t+1}, \mathbf{0}) \quad (3)$$

where $\hat{\mathbf{x}}_{t+1}$ is updated by the Kalman Filter by assuming each future observation is obtained without sensor noise (maximum likelihood observation [33]),

$$\mathbf{z}'_{t+2} = h(\mathbf{x}'_{t+2}, \mathbf{0}). \quad (4)$$

Starting from $\mathbf{x}'_{t+1} = \hat{\mathbf{x}}_{t+1}$, we compute an adjusted motion plan $\pi' = [\mathbf{L}_{t+1}(\hat{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_{t+1}) + \tilde{\mathbf{u}}_{t+1}, \ldots, \mathbf{L}_{T-1}(\hat{\mathbf{x}}_{T-1} - \tilde{\mathbf{x}}_{T-1}) + \tilde{\mathbf{u}}_{T-1}, \mathbf{0}]$ and corresponding MLST $[\mathbf{x}'_{t+1}, \mathbf{x}'_{t+2}, \ldots, \mathbf{x}'_T]$. In Algorithm 1, π' is generated in line 13 and is added into the plan set in line 14 for consideration as the best plan for the next period.

F. *Estimating the* A Priori *Probability of Success of a Plan*

Both cost functions introduced in Sec. III-A require *a priori* estimation of the probability of success of a motion plan. Under the assumption of Gaussian uncertainty, the *a priori* distributions of the robot state at each time step over the course of executing a plan can be estimated quickly [4, 30, 38, 42]. These methods typically assume that the robot will execute the plan with an optimal linear feedback controller and use a Kalman Filter variant for state estimation [37], as we assume in our implementation of HFR. Formally, given a plan π, a nominal trajectory $[\tilde{\mathbf{x}}_t, \tilde{\mathbf{x}}_{t+1}, \ldots, \tilde{\mathbf{x}}_T]$, and an initial belief $N(\hat{\mathbf{x}}_t, \hat{\mathbf{\Sigma}}_t)$, the above methods can be used to estimate a sequence of Gaussian distributions $\{N(\tilde{\mathbf{x}}_{t'}, \tilde{\mathbf{\Sigma}}_{t'})\}$, for $t \le t' \le T$. For each time step t', the mean of the Gaussian distribution is the original nominal system state $\tilde{\mathbf{x}}_{t'}$, while covariance $\tilde{\mathbf{\Sigma}}_{t'}$ captures the possible deviation from $\tilde{\mathbf{x}}_{t'}$ when the robot is at time step t' during execution. This sequence of Gaussian distributions captures the distributions of deviations from nominal trajectory during the execution of the plan π with the LQG feedback controller. In our implementation, we truncate the Gaussian distributions at each time step to remove portions of the distribution for which collisions would occur and then only propagate the truncated Gaussian distribution to future time steps [30] for computing an accurate, yet conservative, estimate of the probability of success (Fig. 3). More formally, at any time step t', let us assume the state of the system $\mathbf{x}_{t'} \sim \mathcal{N}(\tilde{\mathbf{x}}_{t'}, \tilde{\Sigma}_{t'})$. Before propagating the distribution to time step $t'+1$, we truncate the Gaussian distribution against obstacles to remove the parts of the distribution that collide with obstacles. The truncated distribution captures the possible states of the system that are collision-free at time step t'. Before propagating the uncertainty distribution to the next time step, we fit a Gaussian distribution to the truncated distribution. By propagating the new Gaussian distribution to time step $t'+1$, we only propagate to the next time step states that are collision-free. Hence, we properly consider the dependency of uncertainty on previous time steps. Namely, the possible states of the system at time step $t'+1$ should be conditioned on the fact that the system is collision-free at time step k, where $0 \le k \le t'$.

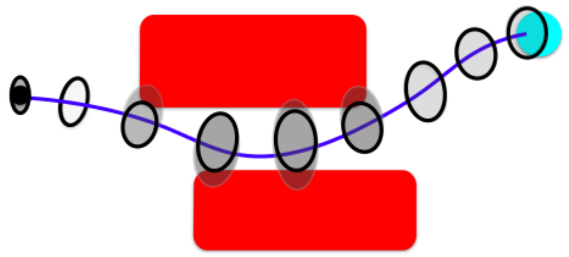


Fig. 3. Consider a path from the initial position (black dot) to the goal region (cyan circle). To evaluate the probability of the success of this motion plan, we estimate uncertainty distributions along the motion plan. The probability of collision at each time step is conditioned on the previous stages being collision free. We truncate a Gaussian distribution (gray ellipses) propagated from the previous step against obstacles and then use a Gaussian distribution (black ellipse) to approximate the truncated distribution. We then propagate the new Gaussian distribution (black ellipse) to the next time step.

Using the approach above for estimating the *a priori* probability of success of a candidate plan, we have the tools necessary to evaluate both of the cost functions c in Sec. III-A.

IV. Analysis

Our analysis of HFR applies to *goal-oriented problems*: we require the optimal plan π^* to have the property that state $\mathbf{x}^*_t$ along the plan π^* is closer (based on the distance metric used by the RRT) to the state $\mathbf{x}^*_{t-1}$ than to any state $\mathbf{x}^*_{t'}$ where $t' < t-1$. For goal-oriented problems, the robot will never return close (i.e., reachable in Δ time) to a previously explored state.

We first analyze MPRRT, our approach for executing multiple RRTs in each planning interval Δ as described in Sec. III-C. Our analysis applies to holonomic and nonholonomic robots. We show that for a goal-oriented problem in which a plan is represented by control inputs over a finite number of periods, the plan returned by MPRRT asymptotically converges toward the optimal plan π^* with probability 1.

We then analyze HFR in comparison to following a precomputed plan with an associated closed-form linear control policy (e.g., an LQG controller). We show that, as computational power increases, at each time step t the method will select a control input $\mathbf{u}_t$ that is equal or better than a control input generated by a linear feedback controller with respect to a cost metric involving probability of success.

A. *Asymptotic Optimality of MPRRT*

We first show this MPRRT is asymptotically optimal. To simplify notation we present the analysis assuming $\mathcal{X} = \mathcal{Q}$, although the results can be extended to the case where these spaces are distinct by applying the analysis below solely to the elements in $\mathcal{Q}$. We call a plan *α-collision free* if all states along the plan are a distance of at least α from any obstacle. We consider cost functions for which α for an optimal solution is non-zero, which is reasonable for safe motion planning under uncertainty and is satisfied by the cost functions considered in this paper.

Given starting state $\mathbf{x}_0$ and two feasible plans π_1 and π_2, we define the distance between π_1 and π_2 as $\|\pi_1 - \pi_2\| =$

$\max_{\zeta\in[0,1]}\|\mathbf{x}(\zeta T_1\Delta,\pi_1)-\mathbf{x}(\zeta T_2\Delta,\pi_2)\|$, where T_1 is the number of periods of π_1 and T_2 is the number of periods of π_2.

Assumption A *The cost function c is Lipschitz continuous, i.e., there exists some constant K such that starting from $\mathbf{x}_0$, for any two feasible sequence of controls π_1 and π_2, $\|c(\pi_1,\mathbf{x}_0)-c(\pi_2,\mathbf{x}_0)\|\le K\|\pi_1-\pi_2\|$.*

Assumption A expresses that two plans that are close to each other have similar cost.

Assumption B *The discrete-time dynamics function $\mathbf{x}_{t+1}=f(\mathbf{x}_t,\mathbf{u}_t,\mathbf{0})$:*

- *is locally Lipschitz continuous, and*
- *can be approximated as $\mathbf{x}_{t+1}=\mathbf{x}_t+\Delta\dot{\mathbf{x}}_t$ for a sufficiently small duration Δ.*

The second point of assumption B assumes that the system dynamics can be locally approximated as linear, which is also assumed by many previous methods (e.g., [30, 38]) for motion planning under uncertainty.

Given a plan $\pi=[\mathbf{u}_0,\mathbf{u}_1,\ldots,\mathbf{u}_T]$ and $\mathbf{x}_0$, we define $\Gamma=[\mathbf{x}_0,\mathbf{u}_0,\mathbf{x}_1,\mathbf{u}_1,\ldots,\mathbf{x}_T,\mathbf{u}_T]$ as the nominal trajectory when executing π from $\mathbf{x}_0$ with zero process noise.

Assumption C *There exists an optimal plan such that the corresponding optimal nominal trajectory can be expressed in a discrete form as $\Gamma^*=[\mathbf{x}_0^*,\mathbf{u}_0^*,\ldots,\mathbf{x}_T^*,\mathbf{u}_T^*]$ for some finite T, where $\mathbf{u}_T^*=0$, and there exists a constant $\alpha\in R^+$ such that Γ^* is α-collision free.*

Assumption C requires that the optimal trajectory Γ^* has a finite (but possibly unknown) number of steps and is some nonzero distance away from obstacles. Assumption C is satisfied for common uncertainty-based cost functions such as maximizing probability of success or minimizing path length subject to a chance constraint.

Using Assumption B, we can show that:

Lemma IV.1 *$\forall(\mathbf{x}_t^*,\mathbf{u}_t^*,\mathbf{x}_{t+1}^*)$, where $\mathbf{x}_{t+1}^*=f(\mathbf{x}_t^*,\mathbf{u}_t^*,\mathbf{0})$, there exists constants δ and L such that, for any $(\mathbf{x}_t,\mathbf{u}_t)$, if $\|\mathbf{x}_t-\mathbf{x}_t^*\|\le\delta$ and $\|\mathbf{u}_t-\mathbf{u}_t^*\|\le\delta$, then:*

- *$\|f(\mathbf{x}_t,\mathbf{u}_t,\mathbf{0})-f(\mathbf{x}_t^*,\mathbf{u}_t^*,\mathbf{0})\|\le 2L\delta$, i.e., $\|\mathbf{x}_{t+1}-\mathbf{x}_{t+1}^*\|\le 2L\delta$,*
- *$\|\mathbf{x}_{t'}-\mathbf{x}_{t'}^*\|\le 2L\delta$ for any time $t'\in(t,t+1)$.*

The first point of lemma IV.1 is derived from the first part of assumption B. The second part of assumption B implies the second point of lemma IV.1.

Motivated by the ideas in [16] and based on lemma IV.1, we can build "balls" along the optimal trajectory Γ^*. Given any $\epsilon\in(0,\min(\alpha,\delta))$, for any $(\mathbf{x}_t^*,\mathbf{u}_t^*)$, we define ϵ,t-ball $B_{\mathbf{x}_t^*}$ as $\|\mathbf{x}_t-\mathbf{x}_t^*\|\le\frac{\epsilon}{(2L)^{T-t}}$ centered at $\mathbf{x}_t^*$ and we define ϵ,t-ball $B_{\mathbf{u}_t^*}$ as $\|\mathbf{u}_t-\mathbf{u}_t^*\|\le\frac{\epsilon}{(2L)^{T-t}}$ centered at $\mathbf{u}_t^*$. At time step t, if the robot is at a state within $B_{\mathbf{x}_t^*}$ and executes a control input within $B_{\mathbf{u}_t^*}$, then based on lemma IV.1 the robot will reach a state within $B_{\mathbf{x}_{t+1}^*}$. Based on the latter result of lemma IV.1 and the choice of ϵ, any state between time t and $t+1$ belongs to a bounded space centered at the corresponding state on the segment $(\mathbf{x}_t^*,\mathbf{x}_{t+1}^*)$, and hence is collision free as well. Given a path π that has the same number of periods as the optimal path π^*, we call path π an ϵ,t-ball path if and only if for any $(\mathbf{x}_t,\mathbf{u}_t)$ on π, $\mathbf{x}_t\in B_{\mathbf{x}_t^*}$ and $\mathbf{u}_t\in B_{\mathbf{u}_t^*}$.

We consider the RRT algorithm as described in Sec. III-C, where the RRT terminates as soon as the first path is found. We require that the sampling strategy cover the state space and that the RRT node expansion/steering function adequately covers the control input space as described below.

Assumption D *Consider a robot at state $\mathbf{x}$ for which there exists a ball of controls B_U of nonzero volume that can feasibly move the robot in Δ time to a state inside a ball $B_{\mathbf{x}'}$ with nonzero volume centered at $\mathbf{x}'$. The extend($\mathbf{x}$, $\mathbf{x}''$) function in RRT is implemented such that there is a non-zero probability that, for a sample $\mathbf{x}''$ sampled from $B_{\mathbf{x}'}$, the extend function returns a control $\mathbf{u}\in B_U$ to a next state $\mathbf{x}'''\in B_{\mathbf{x}'}$.*

We note that many commonly used implementations of the RRT extend function satisfy the assumption above, such as the simple strategy of randomly generating several controls and selecting the control that moves the robot closest to the sampled state.

Theorem IV.2 *(MPRRT is asymptotically optimal) Let π_i denote the best plan found after i RRTs have returned solutions. Given the assumptions above and assuming the problem is goal-oriented and admits a feasible solution, as the number of independent RRT plans generated in MPRRT increases, the best plan almost surely converges to the optimal plan π^*, i.e., $P(\lim_{i\to\infty}\|c(\pi_i,\mathbf{x}_0)-c(\pi^*,\mathbf{x}_0)\|=0)=1$.*

Proof Without loss of generality, we assume ϵ is sufficiently small ($\epsilon<\min(\alpha,\delta)$). We build ϵ,t-balls along Γ^* for any ϵ. Consider a sequence of events that can generate a ϵ,t-ball path. In a single RRT tree, we start from the initial state $\mathbf{x}_0^*$. From Assumption D, we know there is non-zero probability that RRT can generate a control in $B_{\mathbf{u}_0^*}$ to extend to a new state in $B_{\mathbf{x}_1^*}$. The probability of the second sample ending in $B_{\mathbf{x}_1^*}$ is non-zero since it has non-zero volume. Because of the goal-oriented assumption, the second sample is closest to the newly generated state in $B_{\mathbf{x}_1^*}$ and the steer function extends the tree from the state in $B_{\mathbf{x}_1^*}$ to the second sample. Again, based on Assumption D, there is non-zero probability that RRT can generate a control in $B_{\mathbf{u}_1^*}$ to extend to a new state in $B_{\mathbf{x}_2^*}$. The process keeps going until the final state ends up in $B_{\mathbf{x}_T^*}$. Since we assume the optimal plan has finite steps and in each step we have non-zero probability to generate a control that leads to a new state ending inside the next ϵ,t ball centered around the corresponding state of the optimal plan. Hence, the probability of generating an ϵ,t-ball path by one execution of RRT is nonzero. Note that an ϵ,t-ball path must be an ϵ-close path, i.e., for any $(\mathbf{x}_t,\mathbf{u}_t)$ on π, $\|\mathbf{x}_t-\mathbf{x}_t^*\|\le\epsilon$ and $\|\mathbf{u}_t-\mathbf{u}_t^*\|\le\epsilon$. Thus the probability of generating an ϵ-close path is non-zero as well, which we express as $P_\epsilon\in R^+$. Hence we have $P(\|\pi_i-\pi^*\|>\epsilon)=(1-P_\epsilon)^i=\bar{P}_\epsilon^i$. Thus, $\sum_i P(\|\pi_i-\pi^*\|>\epsilon)=\sum_i\bar{P}_\epsilon^i\le\frac{1}{1-\bar{P}_\epsilon}<\infty$. Based on a Borel-Cantelli argument [11], we have $P(\lim_{i\to\infty}\|\pi_i-\pi^*\|=0)=1$. Since the cost function is Lipschitz continuous based on assumption A, $P(\lim_{i\to\infty}\|c(\pi_i,\mathbf{x}_0)-c(\pi^*,\mathbf{x}_0)\|=0)=1$.

The above theorem shows that as computational power increases, the plan returned by MPRRT will almost surely converge to the optimal plan. However, as in RRT* and related methods, MPRRT will never exactly generate the optimal plan since optimal plans have zero-measure volume [15]. Although each independent RRT is not asymptotically optimal as the number of configuration samples rises [15], we have shown that as long as the above assumptions hold, as the number of feasible plans from independent RRTs approaches infinity, the best plan will almost surely converge to an optimal plan.

B. Analysis of HFR

We analyze HFR as a pure sampling-based replanning strategy (i.e., without using the MLST heuristic). Continuing with the results of Sec. IV-A, we assume computation power approaches infinity in lines 7–9 in HFR and assume Δ is small such that the difference between $\tilde{\mathbf{x}}_{t+1}$ and $\hat{\mathbf{x}}_{t+1}$ is small.

We define $\pi^*_{\hat{\mathbf{x}}_t}$ as the optimal plan that starts from $\hat{\mathbf{x}}_t$. The plan $\pi^*_{\hat{\mathbf{x}}_t}$ has a finite number of periods T and is α_t-collision free where $\alpha_t \in R^+$. Let η_t be the shortest distance between the system state at the last period and the goal region boundary $\mathcal{G}$. Let $\epsilon_t = \min(\frac{\alpha_t}{2}, \delta, \eta_t)$ and build ϵ_t, t-balls along $\pi^*_{\hat{\mathbf{x}}_t}$, resulting in a ball centered at $\hat{\mathbf{x}}_t$ with radius $r_t = \frac{\epsilon_t}{(2L)^T}$. We define $\pi_{\tilde{\mathbf{x}}_t}$ as a plan starting at $\tilde{\mathbf{x}}_t$ with the sequence of controls in $\pi^*_{\hat{\mathbf{x}}_t}$.

We first generalize theorem IV.2 as follows.

Theorem IV.3 *For any plan π', if it has a finite number of periods and there exists an $\alpha \in R^+$ such that π' is an α-collision free plan, then MPRRT will generate a plan that almost surely converges to π'.*

The proof for theorem IV.3 is similar to that of theorem IV.2 and requires defining a sequence $\{\pi_i\}_{i \in N^+}$ of plans where π_i is a plan that is the closest to π' after i RRT plans are generated. Using theorem IV.3, properly accounting for collisions, and assuming the cost of the paths are computed after estimating the state (line 15 of HFR), we can show that if $\|\hat{\mathbf{x}}_t - \tilde{\mathbf{x}}_t\| \le r_t$, asymptotic optimality is retained in each period. We note that the threshold r_t may be small but does not need to be zero.

Theorem IV.4 *Assume Δ is small such that at any time step t in HFR, $\|\hat{\mathbf{x}}_t - \tilde{\mathbf{x}}_t\| \le r_t$ holds. Let π^* be the plan with least cost returned by HFR and let π_c be any possible plan (a sequence of controls) that would have been executed by a linear feedback controller. Then, at time step t, executing the first control of π^* is better than or at least equal to executing the first control of π_c with respect to the cost metric.*

Theorem IV.4 holds only as computation power increases. For practical problems where computation power is limited, we optionally add the MLST of the prior best plan (lines 13 and 14) to the set of considered plans. Based on the construction of the MLST, the robot will execute a control that is the same as the control that would have been executed if an LQG controller was applied.

We can prove theorem IV.4 by computing the control input $\mathbf{u}_t$ that would be executed at time step t using the closed form formula for a linear controller. We define π_c as the best plan that could be executed with the first control input being $\mathbf{u}_t$. Since the optimal plan π^* computed by HFR satisfies $c(\pi^*, \hat{\mathbf{x}}_t) \le c(\pi_c, \hat{\mathbf{x}}_t)$, HFR is at least as good as a linear controller with respect to the cost metric at each time step t.

V. Experiments

We apply HFR to three robots: (1) a holonomic robot, (2) a nonholonomic car-like robot, and (3) a nonholonomic steerable medical needle. We tested our C++ implementation on a PC with two 2.00 GHz Intel (R) Xeon (R) processors (12 processor cores total).

A. Holonomic Robot with Two Moving Obstacles

We consider a scenario, based on Du Toit et al. [8, 9], where a holonomic robot moves to a goal state while two dynamic moving obstacles cross the space between the robot and the goal (see Fig. 4). Following the setup of Du Toit [8] (Sec. 4.1.4.3), we define the robot state by its position and velocity, $\mathbf{q}_t = (x, y, v_x, v_y)$. The heading of the robot θ can be computed from v_x and v_y. The control input $\mathbf{u}$ is a 2-D vector that encodes change in velocity. We define the obstacle's state as $\mathbf{o} = (o_x, o_y, o_{v_x}, o_{v_y})$, which consists of the 2-D position and the 2-D velocity. Then, the system state at any time step t is defined as $\mathbf{x} = (\mathbf{q}, \mathbf{o}^1, \mathbf{o}^2)$ where $\mathbf{q}$ is the state of the robot, and $\mathbf{o}^i$ is the state of the i'th obstacle ($i \in \{1, 2\}$). The robot does not have direct control on the parameters $\mathbf{o}^i$. The dynamics of the robot is governed by

$$\mathbf{q}_t = \mathbf{A}\mathbf{q}_{t-1} + \mathbf{B}\mathbf{u}_{t-1} + \mathbf{F}\mathbf{w}_{t-1} \tag{5}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \Delta & 0 \\ 0 & 1 & 0 & \Delta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \text{ and } \mathbf{F} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The 4-D noise term $\mathbf{w}_{t-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{W})$, where $\mathbf{W} = 0.01 \times \mathbf{I}$. We set Δ to 0.5 seconds. We assume the system can receive observations of the robot's position and obstacles' positions, which gives the following sensing model:

$$h(\mathbf{x}, \mathbf{n}) = [x, y, o^1_x, o^1_y, o^2_x, o^2_y]^\top + \mathbf{n}, \tag{6}$$

where the 6-D sensing vector consists of robot's position and two obstacles' positions , corrupted by a 6-D noise vector $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{N})$, where $\mathbf{N} = 0.01 \times \mathbf{I}$.

Following Du Toit et al. [8, 9], we set the robot's initial pose $\mathbf{q}_0 = (x, y, v\cos\theta, v\sin\theta)$ by setting $x = 0$, sampling y uniformly from $[-2, 2]$, sampling θ uniformly from $[-22.5^o, 22.5^o]$ (where $\theta = 0$ represents the direction of horizontal to the right), and setting the initial velocity v to 1.2. The goal $\mathcal{G}$ is at $(12, 0)$, so the robot moves roughly to the right. For both obstacles, the initial x position is uniformly sampled from $[4, 6]$. The initial y positions for the two obstacles are fixed to 6 and -6. The initial heading of the top obstacle is uniformly sampled from $[-120^o, -75^o]$ while the initial heading of the bottom obstacle is uniformly sampled from $[75^o, 120^o]$. The initial velocity for both obstacles is set

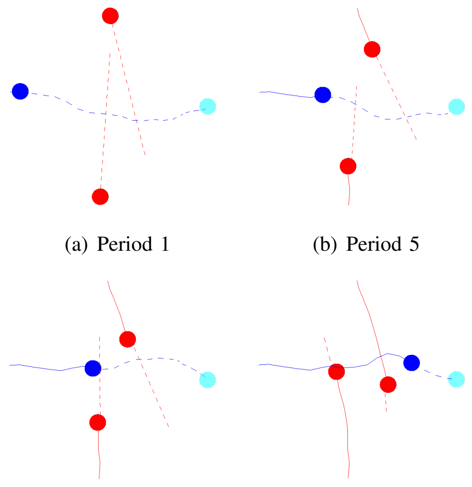


Fig. 4. HFR applied to a holonomic robot with two moving obstacles. The robot (dark blue) is moving to a goal (sky blue) and must avoid two moving obstacles (red) that travel roughly up-down and cross the robot's intended path. In each period of execution shown, the solid lines represent past motions and the dashed lines represent planned motions. Before period 5, the robot was planning to react to the upper moving obstacle by passing from below. Between periods 5 and 7, the robot found a new, better plan passing the upper moving obstacle from above. The robot avoided local minima by successfully finding a new, better plan in a different homotopic class.

to 1. Hence, the crossing obstacles start from random locations near the top and bottom of the environment and move at random headings roughly up/down. The obstacles move based on the dynamics model in equation 5 with the control inputs set to $\mathbf{0}$:

$$\mathbf{o}_t = \mathbf{A}\mathbf{o}_{t-1} + \mathbf{F}\mathbf{w}_{t-1}. \tag{7}$$

Using this scenario, we compare HFR to Partially Closed-Loop Receding Horizon Control (PCLRHC) [9], which is a receding horizon control approach specifically designed for dynamic, uncertain environments. As in [8], for given motion plans of the robot and the moving obstacles, we use the EKF with the maximum likelihood observation assumption to propagate the uncertainty modelled as a Gaussian distribution from the first step of the plan to the end of the plan. Identical to the setup in [8], we use a cost function related to shortest path subject to a chance constraint of less than 1% probability of failure at each time step. Control magnitude is constrained to be less than 1 at each step. At each time step, each velocity component is subject to chance constraints of $P(v_x > 2) \leq 1\%$ and $P(v_x < -2) \leq 1\%$, and similarly for v_y.

We randomly simulated 200 executions using HFR, and in 176 cases the robot reached the goal while satisfying the constraints, implying an 88% success rate. We verified that the chance constraint of a 1% maximum failure rate at each period was achieved. We show a simulated run of HFR in Fig. 4. We compare the path lengths achieved by HFR to the path lengths achieved by PCLRHC [8] in Fig. 5. HFR more regularly achieves lower path lengths, even though PCLRHC in some cases stops short of the goal when the local optimization criteria are satisfied. HFR achieves a lower average path length because, unlike PCLRHC, HFR automatically explores multiple homotopic classes and is not sensitive to local minima.

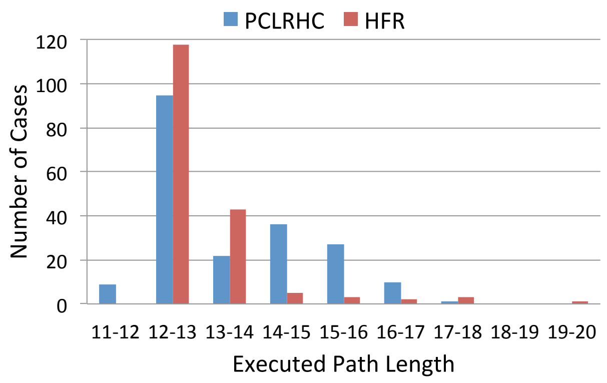


Fig. 5. Histogram of executed path lengths for HFR and for PCLRHC [8]. Note that the length of the shortest path that connects the start position and the goal position is at least 12. PCLRHC occasionally returns values that are less than 12, which is likely due to the robot not exactly reaching the goal. HFR achieves a lower average path length because, unlike PCLRHC, HFR is not sensitive to local minima.

B. Nonholonomic Car-like Robot

We also apply HFR to a nonholonomic car-like robot with 2nd-order dynamics in a 2-D environment with obstacles. The state of the robot $\mathbf{q} = (x, y, \theta, v)$ consists of the robot's 2-D position (x, y), its orientation θ, and its speed v. Here the system state $\mathbf{x}$ and the robot state $\mathbf{q}$ are the same. Its control input $\mathbf{u} = (\alpha, \phi)$ is a 2-D vector consisting of an acceleration α and the steering wheel angle ϕ, corrupted with noise $\mathbf{m} = (\tilde{\alpha}, \tilde{\phi}) \sim \mathcal{N}(\mathbf{0}, \mathbf{M})$. This gives the following nonlinear dynamics model:

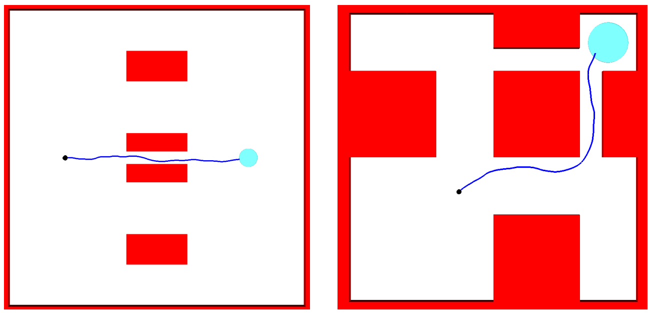
$$f[\mathbf{x}, \mathbf{u}, \mathbf{m}] = \begin{bmatrix} x + \tau v \cos\theta \\ y + \tau v \sin\theta \\ \theta + \tau v \tan(\phi + \tilde{\phi})/d \\ v + \tau(\alpha + \tilde{\alpha}) \end{bmatrix}, \tag{8}$$

where τ is the time step and d is the length of the car. The robot can sense its position and speed, giving us the following stochastic sensing model:

$$h(\mathbf{x}, \mathbf{n}) = \begin{bmatrix} x \\ y \\ v \end{bmatrix} + \mathbf{n} \tag{9}$$

where the 3-D observation vector consists of the robot's (x, y) position and its velocity v, corrupted by a 3-D sensing noise vector $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{N})$. We evaluate HFR for this car-like nonholonomic robot in two environments shown in Fig. 6 to demonstrate HFR's convergence to optimality and performance with respect to the planning interval Δ.

1) Environment with Known Optimal Plan: We applied HFR to the car-like robot in the environment shown in Fig. 6(a) for which the the optimal plan is known. The optimization objective for this scenario is to minimize path length subject to a chance constraint that $P(\text{collision free}) \geq 90\%$. We set $\mathbf{M} = 0.001 \times \mathbf{I}$ and $\mathbf{N} = 0.0001 \times \mathbf{I}$. With this cost function and this low uncertainty, the optimal plan for the environment



(a) Car-like Robot Environment 1 (b) Car-like Robot Environment 2

Fig. 6. We evaluate HFR for a car-like robot in two environments: (a) an environment with a known optimal motion plan, and (b) a more complex planar environment previously used by Patil et al. [30]. The environments include the robot's initial position (black dots), the goal region (sky blue discs), and obstacles (red rectangles) that divide the environment into several different homotopic classes.

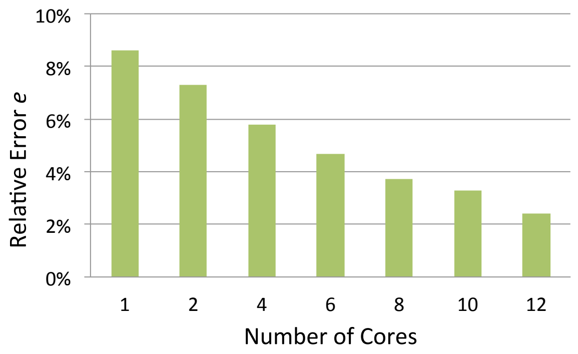


Fig. 7. Convergence to the optimal solution for the car-like robot in environment 1 as the number of processor cores increases.

in Fig. 6(a) is to move in a straight line from the initial position (black dot) to the goal region while passing through the middle narrow passage. For this environment, when executed a single time to find only a feasible solution, the RRT method is more likely to find a suboptimal solution moving through a wider passage than to find the optimal solution moving through the narrow passage.

We evaluate the performance of HFR as parallel computation power increases, i.e., as the number of CPU cores increases. For each given number of available cores, we executed HFR 100 times and computed the average path length of all successful executions. We provide HFR 2 seconds to compute an initial plan and we set $\Delta = 0.5$ seconds. We computed the relative error e between the average path length $\bar{L}$ and the optimal path length L^* as $e = \frac{\bar{L}-L^*}{L^*}$. As shown in Fig. 7, the robot using HFR moves along a path that is closer to the globally optimal solution as the number of processor cores increases. The relative error drops to below 3% when the number of CPU cores rises to 12.

2) Performance with Respect to Δ: We also applied HFR to a more complex scenario for which the optimal solution cannot be computed analytically. We consider the environment shown in Fig. 6(b) and use the optimization objective of maximizing the probability of success (i.e., reaching the goal while not colliding with obstacles). To model a more realistic car-like robot operating at higher speeds, we also set the process

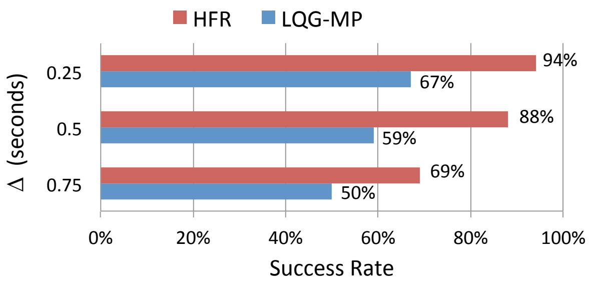


Fig. 8. The performance of HFR and LQG-MP for different values of Δ for the car-like robot in environment 2. HFR performs replanning with period Δ, and for LQG-MP Δ is used to determine the time step size of the computed LQG controller.

uncertainty to be linearly dependent on the robot's velocity. Specifically, we set $\mathbf{N} = 0.005 \times \mathbf{I}$ and $\mathbf{M} = v \times 0.001 \times \mathbf{I}$.

Using this scenario, we ran HFR for different values of Δ and measured the probability of success over 100 simulated runs. For comparison, we also ran LQG-MP [38] for the same scenario and computed the probability of success over 100 simulated runs. Both HFR and LQG-MP were allowed 2 seconds for pre-computing plans and selecting the best plan as the initial plan. As shown in Fig. 8, decreasing the value of Δ improves the probability of success for HFR since the predicted start of the robot for planning is more accurate and the robot can respond more quickly to uncertainty. This result is consistent with Theorem IV.4. Fig. 8 also shows that HFR consistently has a higher probability of success than LQG-MP. We also show an example execution of HFR with $\Delta = 0.25$ in Fig. 6(b).

C. Nonholonomic Steerable Needle

We also apply HFR to a flexible bevel-tip steerable medical needle [36], a type of steerable needle that bends in the direction of the bevel when inserted into soft tissue. By axially rotating the needle, the direction of the bevel can be adjusted, enabling the needle to follow curved trajectories in 3D tissue environments with obstacles. The motion of the steerable needle tip has non-linear dynamics and can be modeled as a nonholonomic robot.

Following the derivation in [40], we describe the state of the needle $\mathbf{q}$ by the 4×4 matrix $\mathbf{X} = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0} & 1 \end{bmatrix}$ where $\mathbf{R} \in SO(3)$ is the rotation matrix and $\mathbf{p} \in \mathbb{R}^3$ is the position of the needle tip. The system state $\mathbf{x}$ consists of the state $\mathbf{q}$ of the robot and a kinematics parameter $\kappa_{\max}$, which represents the maximum curvature of the needle. The curvature of the needle can be varied by applying duty cycled spinning during the needle's insertion [25]. The control inputs $\mathbf{u} = [v, w, k]^T \in \mathbb{R}^3$, consist of the insertion speed v along the z axis, the twist angular velocity w, and the curvature κ. We describe the dynamics of the needle tip in terms of the instantaneous twist $\mathbf{U} \in se(3)$ expressed in a local coordinate frame attached to the needle tip: $\mathbf{U} = \begin{bmatrix} [\mathbf{w}] & \mathbf{v} \\ \mathbf{0} & 0 \end{bmatrix}$, where $\mathbf{w} = [v\kappa, 0, w]^T$, $\mathbf{v} = [0, 0, v]^T$, and the notation $[\mathbf{s}]$ for a vector $\mathbf{s} \in \mathbb{R}^3$ refers to the 3×3 skew symmetric cross product matrix. The instantaneous

	Highest probability of success (success rate)			Shortest path with chance constraint (Average path length ± STD)	
Method:	LQG-MP	Preplan+LQG	HFR	Preplan+LQG	HFR
Performance:	90%	94%	98%	11.64±0.32	11.13±0.59

TABLE I
COMPARISON OF HFR WITH LQG-MP AND PREPLAN+LQG FOR THE NONHONOMIC STEERABLE NEEDLE

twist $\mathbf{U}'$ encodes the additive motion noise $\mathbf{m} = [\mathbf{v}', \mathbf{w}']^T \sim \mathcal{N}(\mathbf{0}, \mathbf{M})$ in a similar way.

Given time step duration Δ, the stochastic discrete-time dynamics of the needle tip is given by the following model:

$$f(\mathbf{x}, \mathbf{u}, \mathbf{m}) = \begin{bmatrix} \mathbf{X} \exp(\Delta(\mathbf{U} + \mathbf{U}')) \\ \kappa_{\text{max}} \end{bmatrix}, \tag{10}$$

where the control input $\mathbf{u}$ cannot directly control κ_{max}, which is fully determined by the physical properties of the given needle and the tissue. Below, we consider two different scenarios: (1) a scenario where κ_{max} is *a priori* known, and (2) a scenario where κ_{max} is not known and needs to be estimated online.

We assume that noisy observations of pose are obtained. The noise in the sensing measurement is modeled as $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$. This gives the stochastic measurement model:

$$h(\mathbf{x}, \mathbf{n}) = \mathbf{q} + \mathbf{n}. \tag{11}$$

The steerable needle has non-linear dynamics, and van den Berg et al. provide details about linearizing the needle dynamics and sensing models for LQG control [40].

1) Artificial Environment Scenario: We apply HFR under two optimization objectives: maximizing probability of success and minimizing path length with a chance constraint of P(Collision free) $\geq$ 80%. We set the needle motion and sensing model parameters to $v_{\text{max}} = 1.5$ cm/s, $w_{\text{max}} = 2\pi$ rad/s, $\Delta = 0.5$ s, $\kappa_{\text{max}} = 2$ (cm)$^{-1}$, $\mathbf{Q} = 0.1 \times \mathbf{I}$, and $\mathbf{M} = 0.005 \times \mathbf{I}$.

In Fig. 9 we show the environment and a simulated successful run under each of the cost functions. A plan through the passage between red obstacles (i.e., the slightly wider passageway) offers the highest probability of success. A plan next to the yellow obstacle minimizes path length under the given chance constraint.

Table I shows that our method outperforms other methods based on LQG control. For each run with LQG-MP [38], we generate 1000 RRT plans, select the best plan using the LQG-MP metric, and then execute an LQG controller along the selected plan. For each run with Preplan+LQG, we generate 1000 RRT plans, select the best plan using a truncated Gaussian approach [30], and then execute an LQG controller along the selected plan. Approximately 5 seconds are required to generate 1000 feasible plans. We ran each method for 100 runs. HFR performed best on both metrics due to its ability to refine plans during execution.

2) Liver Biopsy Scenario: We consider steering a needle through liver tissue while avoiding critical vasculature. We use the same needle model as before except we assume the maximum curvature of the needle varies by tissue type and is *a priori* unknown to the robot. For muscle/fat tissue outside the liver, $\kappa^{\text{out}}_{\text{max}}$ = 5 cm^{-1}, and for the liver tissue, $\kappa^{\text{in}}_{\text{max}}$ = 4 cm^{-1}. Although the curvature $\kappa^{\text{in}}_{\text{max}}$ is *a priori* unknown to the robot, we assume the system can accurately sense the needle tip pose, $\mathbf{Q} = 0.0001 \times \mathbf{I}$, to assist in curvature estimation. HFR can easily be integrated with curvature estimation by replanning with the latest estimated maximum curvature at each time step.

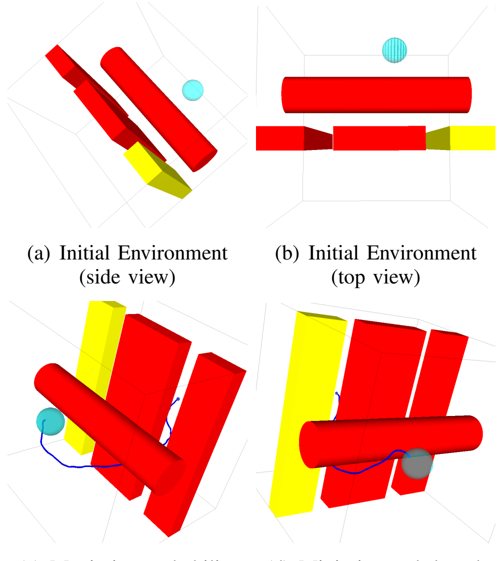

(a) Initial Environment (side view) (b) Initial Environment (top view) (c) Maximize probability of success (d) Minimize path length with chance constraint

Fig. 9. (a) The artificial environment for the steerable needle has two narrow passages in a 10 cm × 10 cm × 10 cm box. The goal region (sky blue sphere) is defined in $\mathbb{R}^3$ with radius 0.5 cm. (b) The passage between the two red boxes (1.2 cm) is wider than the passage between the yellow box and the middle red box (1.0 cm). The goal is closer to the narrower passage. (c) For highest probability of success, HFR guided the needle to pass through the wider passage in order to acheive a higher probability of success. (d) For minimizing path length with the defined chance constraint, HFR guided the needle to pass through the passage between the yellow box and the middle red box to acheive a shorter path while still satisfying the chance constraint.

Since the curvature κ of the needle is linearly dependent on the proportion β of time spent in duty cycled spinning in one insertion duration [25], we model the relationship between β and κ as $\beta = 1 - \frac{\kappa}{\kappa_{\text{max}}}$ where κ_{max} is the maximum curvature feasible in the tissue. The high level control $\mathbf{u} = [v, w, \kappa]^T$ thus can be transformed to a low level control $\mathbf{u}(\kappa_{\text{max}}) = [v, w, (1-\beta)\kappa_{\text{max}}]^T$ parameterized by κ_{max}.

Since κ_{max} is unknown to the robot, in each time step the robot estimates κ_{max} on the fly using an optimization-based method that fits the nominal needle curve to the most recent sensor measurements. We then use the estimated κ_{max} for motion planning at that time step.

We evaluated the performance of HFR in the liver scenario (see Fig. 1) using the optimization objective of maximizing probability of success. Fig. 10 shows results for a different needle insertion location. In Fig. 11, we compare HFR to LQG-MP and Preplan+LQG. Because the maximum curvature in this scenario's tissue is *a priori* unknown, LQG-MP and Preplan+LQG perform poorly. HFR has a 98% success rate by estimating the maximum curvature on the fly and using the latest estimated maximum curvature at each replanning time step.

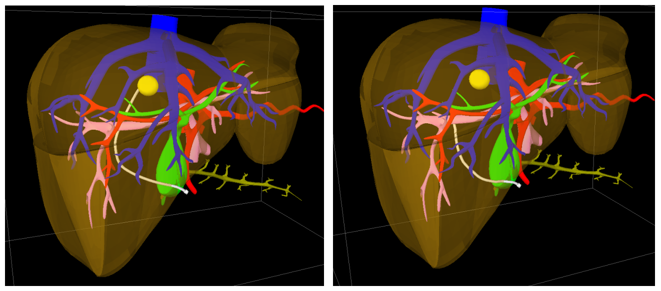

(a) HFR (b) LQG-MP

Fig. 10. HFR applied in simulation to medical needle steering in the liver for a biopsy procedure at a different site from Fig.1. The needle is inserted in front of the liver. (a) HFR successfully guides the needle (white) to the tumor (yellow). Using LQG-MP (b), the needle collides with portal veins (pink).

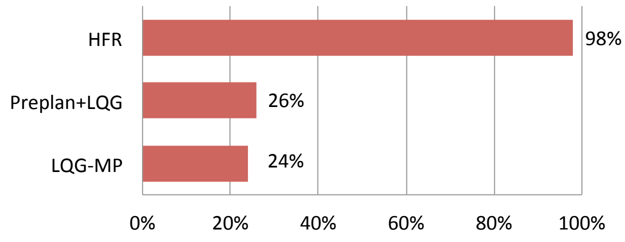


Fig. 11. The percent of successful executions (i.e., reaching the goal and no collisions with obstacles) for the HFR, Preplan+LQG, and LQG-MP for the steerable needle liver biopsy scenario.

VI. Conclusion

As sampling-based motion planners become faster, they can be re-executed more frequently by a robot during task execution to react to uncertainty in robot motion, obstacle motion, sensing noise, and uncertainty in the robot's kinematic model. We investigated and analyzed high-frequency replanning (HFR), where during each period, RRT-based motion planners are executed in parallel as the robot simultaneously executes the first action of the best motion plan from the previous period.

We showed that, as parallel computation power increases, HFR for holonomic and nonholonomic robots will perform equal or better than the more traditional approach of pre-computing a plan and corresponding linear feedback controller. We applied HFR to problems in which uncertainty can be modeled using Gaussian distributions and considered two metrics: maximizing the probability of successfully reaching a goal and minimizing path length subject to a chance constraint. We demonstrated the effectiveness of HFR for three scenarios: (1) maneuvering a holonomic robot to a goal while avoiding moving obstacles, (2) steering a nonholonomic car-like robot with motion and sensing uncertainty, and (3) autonomously guiding a nonholonomic steerable medical needle whose curvature in different tissues types is not known *a priori*.

In ongoing and future work we would like to integrate new uncertainty-based cost functions with the HFR framework. One limitation of our current cost functions for considering uncertainty is that they assume Gaussian distributions. Although Gaussian distributions are commonly used for modeling uncertainty in a broad class of problems, they are not an acceptable approximation in some domains, such as problems in which uncertainty has a multi-modal distribution. For non-Gaussian problems, we will consider integrating particle filter approaches with the HFR framework. We also plan to formally investigate the asymptotic rate of convergence of MPRRT, which is a challenging problem for general robotics scenarios. Our method also assumes the dynamics and sensing model of the system are linearizable, and we would like to investigate integrating into HFR methods for estimating probability of collision that do not require this property. We also plan to investigate cost functions that efficiently estimate the probability of collision for geometrically complex robots, which will enable HFR to be used effectively for articulated manipulators. With the above future work, we hope to extend the applicability of the HFR approach to a broader class of robots.

Acknowledgment

This research was supported in part by the National Science Foundation (NSF) through awards #IIS-0905344, #IIS-1117127, and #IIS-1149965 and by the National Institutes of Health (NIH) under award #R21EB011628.

References

[1] A.-a. Agha-mohammadi, S. Chakravorty, and N. M. Amato, "Sampling-based nonholonomic motion planning in belief space via dynamic feedback linearization-based FIRM," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Oct. 2012, pp. 4433–4440.
[2] R. Alterovitz, T. Siméon, and K. Goldberg, "The Stochastic Motion Roadmap: A sampling framework for planning with Markov motion uncertainty," in *Robotics: Science and Systems (RSS)*, Jun. 2007, pp. 1–8.
[3] J. J. Bialkowski, S. Karaman, and E. Frazzoli, "Massively parallelizing the RRT and the RRT*," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, San Francisco, CA, Sep. 2011, pp. 3513–3518.
[4] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2011, pp. 723–730.
[5] S. Carpin and E. Pagello, "On parallel RRTs for multi-robot systems," *Proc. 8th Conf. Italian Association for Artificial Intelligence*, 2002.
[6] H. Choset, K. M. Lynch, S. A. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, 2005.
[7] D. Devaurs, T. Siméon, and J. Cortés, "Parallelizing RRT on large-scale distributed-memory architectures," *IEEE Trans. Robotics*, vol. 29, no. 2, pp. 767–770, Apr. 2013.
[8] N. E. Du Toit, "Robot Motion Planning in Dynamic, Cluttered, and Uncertain Environments: the Partially Closed-Loop Receding Horizon Control Approach," Ph.D. dissertation, California Institute of Technology, 2010.

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

[10] J. Esposito, J. Kim, and V. Kumar, "Adaptive RRTs for validating hybrid robotic control systems," in *Algorithmic Foundations of Robotics VI*, 2005.

[11] G. Grimmett and D. Stirzaker, *Probability and Random Processes*, 3rd ed. New York: Oxford University Press, 2001.

[12] K. Hauser, "Randomized belief-space replanning in partially-observable continuous spaces," *Algorithmic Foundations of Robotics IX*, pp. 193–209, 2011.

[13] ——, "On responsiveness, safety, and completeness in real-time motion planning," *Autonomous Robots*, vol. 32, no. 1, pp. 35–48, Sep. 2011.

[14] J. Ichnowski and R. Alterovitz, "Parallel sampling-based motion planning with superlinear speedup," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Oct. 2012, pp. 1206–1212.

[15] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robotics Research*, vol. 30, no. 7, pp. 846–894, Jun. 2011.

[16] L. E. Kavraki, M. N. Kolountzakis, and J.-C. Latombe, "Analysis of probabilistic roadmaps for path planning," *IEEE Trans. Robotics and Automation*, vol. 14, no. 1, pp. 166–171, 1998.

[17] S. Koenig and M. Likhachev, "Fast replanning for navigation in unknown terrain," *IEEE Trans. Robotics*, vol. 21, no. 3, pp. 354–363, 2005.

[18] H. Kurniawati, D. Hsu, and W. Lee, "SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces," in *Robotics: Science and Systems (RSS)*, 2008.

[19] S. M. LaValle and J. J. Kuffner, "Rapidly-exploring random trees: Progress and prospects," in *Algorithmic and Computational Robotics: New Directions*, B. R. Donald and Others, Eds. Natick, MA: AK Peters, 2001, pp. 293–308.

[20] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006.

[21] Z. Littlefield, Y. Li, and K. E. Bekris, "Efficient Sampling-based Motion Planning with Asymptotic Near-Optimality Guarantees for Systems with Dynamics," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Nov. 2013, pp. 1779 – 1785.

[22] B. D. Luders, S. Karaman, and J. P. How, "Robust Sampling-based Motion Planning with Asymptotic Optimality Guarantees," *AIAA Guidance, Navigation, and Control (GNC) Conference*, pp. 1–25, Aug. 2013. [Online]. Available: http://arc.aiaa.org/doi/abs/10.2514/6.2013-5097

[23] A. Majumdar and R. Tedrake, "Robust online motion planning with regions of finite time invariance," in *Proc. Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2012.

[24] N. Melchior and R. Simmons, "Particle RRT for Path Planning with Uncertainty," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2007, pp. 1617–1624.

[25] D. Minhas, J. A. Engh, M. M. Fenske, and C. Riviere, "Modeling of needle steering via duty-cycled spinning," in *Proc. Int. Conf. IEEE Engineering in Medicine and Biology Society (EMBS)*, Aug. 2007, pp. 2756–2759.

[26] M. Otte and N. Correll, "C-FOREST: Parallel Shortest Path Planning With Superlinear Speedup," *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 798–806, Jun. 2013.

[27] C. Park, J. Pan, and D. Manocha, "Real-time optimization-based planning in dynamic environments using GPUs," *Robotics and Automation (ICRA), . . .*, pp. 4090–4097, May 2013.

[28] S. Patil and R. Alterovitz, "Interactive motion planning for steerable needles in 3D environments with obstacles," in *Proc. IEEE RAS/EMBS Int. Conf. Biomedical Robotics and Biomechatronics (BioRob)*, Sep. 2010, pp. 893–899.

[29] S. Patil, J. Burgner, R. J. Webster III, and R. Alterovitz, "Needle steering in 3-D via rapid replanning," *IEEE Trans. Robotics (in press)*, 2014.

[30] S. Patil, J. van den Berg, and R. Alterovitz, "Estimating probability of collision for safe motion planning under Gaussian motion and sensing uncertainty," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2012, pp. 3238–3244.

[31] S. Petti and T. Fraichard, "Safe motion planning in dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Aug. 2005, pp. 2210–2215.

[32] E. Plaku and L. Kavraki, "Distributed sampling-based roadmap of trees for large-scale motion planning," *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, no. April, pp. 3868–3873, 2005.

[33] R. Platt Jr., R. Tedrake, L. Kaelbling, and T. Lozano-Perez, "Belief space planning assuming maximum likelihood observations," in *Robotics: Science and Systems (RSS)*, 2010.

[34] S. Prentice and N. Roy, "The belief roadmap: Efficient planning in belief space by factoring the covariance," *Int. J. Robotics Research*, vol. 28, no. 11, pp. 1448–1465, Nov. 2009.

[35] J. Rawlings, "Tutorial overview of model predictive control," *IEEE Control Systems*, vol. 20, no. 3, pp. 38–52, 2000.

[36] K. B. Reed, A. Majewicz, V. Kallem, R. Alterovitz, K. Goldberg, N. J. Cowan, and A. M. Okamura, "Robot-assisted needle steering," *IEEE Robotics and Automation Magazine*, vol. 18, no. 4, pp. 35–46, Dec. 2011.

[37] R. F. Stengel, *Optimal Control and Estimation*. Toronto, Canada: General Publishing Company, Ltd., 1994.

[38] 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. Robotics Research*, vol. 30, no. 7, pp. 895–913, Jun. 2011.

[39] J. van den Berg, S. Patil, and R. Alterovitz, "Motion planning under uncertainty using iterative local optimization in belief space," *Int. J. Robotics Research*, vol. 31, no. 11, pp. 1263–1278, Sep. 2012.

[40] J. van den Berg, S. Patil, R. Alterovitz, P. Abbeel,

and K. Goldberg, “LQG-based planning, sensing, and control of steerable needles,” in *Algorithmic Foundations of Robotics IX (Proc. WAFR 2010)*, ser. Springer Tracts in Advanced Robotics (STAR), D. Hsu and Others, Eds., vol. 68. Springer, Dec. 2010, pp. 373–389.

[41] J. Vannoy and J. Xiao, “Real-time adaptive motion planning (RAMP) of mobile manipulators in dynamic environments With unforeseen changes,” *IEEE Trans. Robotics*, vol. 24, no. 5, pp. 1199–1212, Oct. 2008.

[42] M. P. Vitus and C. J. Tomlin, “Closed-loop belief space planning for linear, Gaussian systems,” in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2011, pp. 2152–2159.

[43] N. Wedge and M. Branicky, “On heavy-tailed runtimes and restarts in rapidly-exploring random trees,” in *AAAI Conference Workshop on Search in Artificial Intelligence and Robotics*, Jul. 2008, pp. 127–133.

[44] E. Yoshida and F. Kanehiro, “Reactive robot motion using path replanning and deformation,” in *IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2011, pp. 5456–5462.

[45] M. Zucker, J. Kuffner, and M. Branicky, “Multipartite RRTs for rapid replanning in dynamic environments,” in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Roma, Italy, Apr. 2007, pp. 1603–1609.

Wen Sun received his B.Tech degree in Computer Science and Technology from Zhejiang University, China, in 2012 and a B.S. degree with Distinction in the School of Computing Science from Simon Fraser University, Canada, in 2012. He is currently a graduate student in the Department of Computer Science at the University of North Carolina at Chapel Hill, NC. His research interests include motion planning under uncertainty and medical robotics.

Sachin Patil received his B.Tech degree in Computer Science and Engineering from the Indian Institute of Technology, Bombay in 2006 and the Ph.D. degree in Computer Science from the University of North Carolina at Chapel Hill in 2013. He is currently a postdoctoral researcher at UC Berkeley. His research interests include motion and path planning in virtual environments, physically-based simulation, and medical robotics.

Ron Alterovitz received his B.S. degree with Honors from Caltech, Pasadena, CA in 2001 and the Ph.D. degree in Industrial Engineering and Operations Research at the University of California, Berkeley, CA in 2006.

In 2009, he joined the faculty of the Department of Computer Science at the University of North Carolina at Chapel Hill, NC where he leads the Computational Robotics Research Group. His research focuses on robot motion planning and physically-based simulation for medical and assistive robotics applications, including surgical assistance, treatment planning, medical image registration, physician training, and personal assistance. Prof. Alterovitz has co-authored a book on Motion Planning in Medicine, was awarded a patent for a medical device, has received multiple best paper finalist awards at IEEE robotics conferences, and is the recipient of an NIH Ruth L. Kirschstein National Research Service Award and an NSF CAREER award.