Motion Planning under Uncertainty for Medical Needle Steering Using Optimization in Belief Space

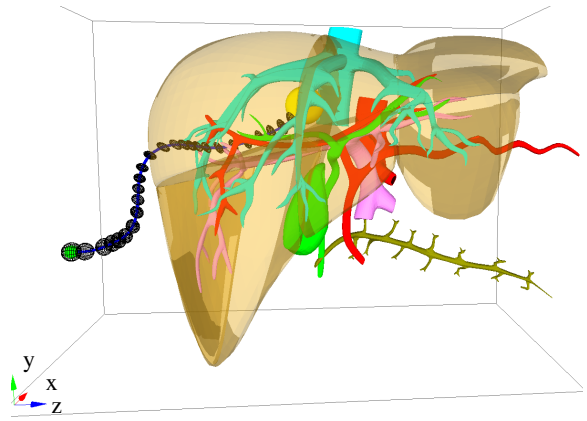
Wen Sun and Ron Alterovitz

***Abstract*— We present an optimization-based motion planner for medical steerable needles that explicitly considers motion and sensing uncertainty while guiding the needle to a target in 3D anatomy. Motion planning for needle steering is challenging because the needle is a nonholonomic and underactuated system, the needle's motion may be perturbed during insertion due to unmodeled needle/tissue interactions, and medical sensing modalities such as ultrasound imaging and x-ray projection imaging typically provide only noisy and partial state information. To account for these uncertainties, we introduce a motion planner that computes a trajectory and corresponding linear controller in the belief space - the space of distributions over the state space. We formulate the needle steering motion planning problem as a partially observable Markov decision process (POMDP) that approximates belief states as Gaussians. We then compute a locally optimal trajectory and corresponding controller that minimize in belief space a cost function that considers avoidance of obstacles, penalties for unsafe control inputs, and target acquisition accuracy. We apply the motion planner to simulated scenarios and show that local optimization in belief space enables us to compute higher quality plans compared to planning solely in the needle's state space.**

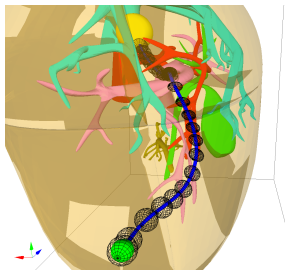
I. Introduction

Many diagnostic and therapeutic medical procedures require physicians to accurately insert a needle through soft tissue to a specific location in the body. Common procedures include biopsies for testing the malignancy of tissues, ablation for locally killing cancer cells, and radioactive seed implantation for brachytherapy cancer treatment. Unlike traditional straight needles, highly flexible bevel-tip needles can be steered along curved trajectories by taking advantage of needle bending and the asymmetric forces applied by the needle tip to the tissue [1]. Steerable needles have the ability to correct for perturbations that occur during insertion, thereby increasing accuracy and precision. Steerable needles also have the ability to maneuver around anatomical obstacles such as bones, blood vessels, and critical nerves to reach targets inaccessible to traditional straight needles.

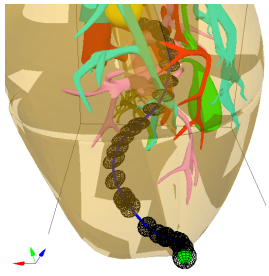
Controlling a steerable needle to reach a target while avoiding obstacles is unintuitive for a human operator, motivating the need for motion planning algorithms. Motion planning for needle steering is challenging because the needle is a nonholonomic system and underactuated, and the challenge is compounded by uncertainty in both motion and sensing. As the needle is inserted into tissue, the motion of the needle is subject to uncertainty due to factors such as inhomogeneous tissue, needle torsion, actuation errors, and tissue deformations [1]. Furthermore, in clinical settings it is typically difficult to precisely sense the pose of the needle tip. Imaging modalities that could provide complete and accurate state information, such as MRI and CT, are either too expensive for many procedures or would emit too much radiation to the patient if used for continuous intra-operative state estimation. Sensing modalities such as ultrasound imaging and x-ray projection imaging are widely



(a) Locally optimal solution computed by our method (front view)



(b) Locally optimal solution (side view)



(c) Initial trajectory (side view)

Fig. 1. We apply our needle steering motion planner to a simulated liver biopsy scenario. The objective is to access the tumor (yellow sphere) while avoiding the hepatic arteries (red), hepatic veins (cyan), portal veins (pink), and bile ducts (green). The sensor is assumed to be mounted at front of the workspace, pointing to the x direction (red arrow), and provides better position estimates of the needle tip when closer to the sensor. (a) The nominal trajectory and the associated beliefs (3 standard deviations) of the locally optimal solution computed using our approach. (b) The side view of the locally optimal solution computed by our method. The locally optimal solution first guides the needle to the right to move near to the sensor to obtain higher accuracy sensing and reduce uncertainty. (c) The initial trajectory computed by an RRT. Steering the needle along the initial trajectory using an LQG controller results in higher expected uncertainty and cost compared to using the locally optimal solution computed by our method.

This research was supported in part by the National Science Foundation (NSF) under award IIS-1149965 and by the National Institutes of Health (NIH) under award R21EB017952.

Wen Sun and Ron Alterovitz are with the Department of Computer Science, The University of North Carolina at Chapel Hill, NC, USA. {wens, ron}@cs.unc.edu

available but provide noisy and/or partial information (e.g., poor resolution or only 2D projections).

To fully consider the impact of uncertainty in motion and sensing, a steerable needle motion planner should not merely compute a static path through the anatomy but rather a policy that defines the motion to perform given any current state information. Although we cannot accurately observe the needle's current state, we can instead estimate a distribution over the set of possible states (i.e., a *belief state*) based on available noisy, partial sensor measurements. In this paper, we introduce a motion planner for steerable needles that computes a trajectory and corresponding linear controller optimized directly in belief space.

Motion planning for steerable needles in belief space introduces multiple challenges. The problem of motion planning over the space of belief states is formally described as a partially observable Markov decision process (POMDP). Computing an optimal solution to a POMDP is known to be computationally complex [2]. In our approach for steerable needles we compute a feasible trajectory using a sampling-based motion planner and then (locally) optimize that trajectory in belief space [3]. To enable local optimization in belief space, we represent the needle dynamics model, defined in the $SE(3)$ group [4], [5], in a vector form that is compatible with optimization in belief space. We also create a cost function compatible with optimization in belief space that explicitly considers avoidance of anatomical obstacles, penalties for unsafe control inputs, target acquisition accuracy, and the diversity of sensor models used in medicine.

We apply the motion planner to simulated scenarios and show that local optimization in belief space enables us to compute motion plans of higher quality compared to plans computed solely in the needle's state space.

II. Related Work

We focus here on bevel-tip needle steering, which is one of a variety of needle steering mechanisms that has been developed (e.g., [1], [6]–[9]). When inserted into soft tissue, bevel-tip steerable needles curve in the direction of the bevel and follow constant curvature arcs in the tissue [1]. Duty-cycled spinning of a bevel-tip steerable needle during insertion enables control of the needle along arcs of different curvatures [10].

Motion planning for steerable needles has been investigated for a variety of scenarios. Motion planners have been developed that consider uncertainty for needle steering through 2D slices of anatomy [11]–[14]. For 3D needle steering under motion uncertainty (but not sensing uncertainty), Park et al. introduced the path-of-probability (POP) algorithm [4]. Van den Berg et al. applied to needle steering the LQG-MP motion planning algorithm [5] that considers both motion and sensing uncertainty.

For motion planning problems with both motion uncertainty and imperfect sensing (i.e., partial and noisy sensing), the planning and control problem is usually modeled as a POMDP problem [15], [16], which has been proved to be PSPACE complete. Point-based algorithms [17]–[23] have been developed for problems with discrete state, action, and/or observation spaces. Methods have been developed [24]–[26] that generate a large number of candidate paths in the state space using sampling-based methods, create controllers for each path, and then evaluate each path and corresponding controller to select the best one. Methods have also been developed [3], [27], [28] that approximate the belief states as Gaussian distributions and compute value functions in parametric form only in local regions of the belief space. For non-Gaussian beliefs, local approaches can be extended by using particle filters [29]. In this paper, we utilize an iLQG-based belief space planner introduced in [3], which we customize and apply to needle steering.

III. Problem Formulation

We consider bevel-tip steerable needles that curve in the direction of the bevel when inserted into tissue. The needle's motion is controlled by two control inputs: the insertion speed $v(\tau)$ and the rotational speed of the needle shaft $\omega(\tau)$ as a function of time τ. The control input vector is hence $\mathbf{u}(\tau) = \begin{bmatrix} v(\tau) & \omega(\tau) \end{bmatrix}^\top \in \mathbb{R}^2$. Since the needle shaft curves and follows the needle tip, the motion of the needle can be modeled by the pose of the needle tip over time. We define the needle tip pose by $\mathbf{x} = \begin{bmatrix} \mathbf{p} & \mathbf{r} \end{bmatrix}^\top$, where $\mathbf{p} \in \mathbb{R}^3$ is the 3D position of the tip and $\mathbf{r} \in \mathbb{R}^3$ is the tip's orientation represented using the axis-angle representation [30].

Uncertainty in the needle's motion can arise due to factors such as tissue inhomogeneity, tissue deformation, actuation errors, and needle torsion. Based on [5], we model motion uncertainty by corrupting the control inputs with an additive noise sampled from a zero-mean Gaussian distribution as described in detail in Sec. IV.

As the needle moves through tissue, we assume that sensing modalities such as ultrasound imaging, X-ray imaging, or electromagnetic tracking can be used to sense the needle tip pose, but this sensing information may be noisy or partial. For example, ultrasound may have substantial noise while X-ray imaging only provides precise position estimates in a plane. We define the stochastic sensing model as

$$\mathbf{z}(\tau) = \mathbf{h}(\mathbf{x}(\tau)) + \mathbf{n}(\tau), \;\; \mathbf{n}(\tau) \sim \mathcal{N}(\mathbf{0}, V(\mathbf{x}(\tau))) \qquad (1)$$

where $\mathbf{z}(\tau)$ is the output of the sensor at time τ, $\mathbf{h}$ is a (possibly non-linear) function modeling the sensor, and $\mathbf{n}(\tau)$ is the sensor noise at time τ assumed to have a Gaussian distribution with state-dependent variance $V(\mathbf{x}(\tau))$. We note that the dimension of $\mathbf{z}$ might be less than $\mathbf{x}$ in the case of partial sensing. We will introduce several detailed sensing models in Sec. VI.

A. The Belief Space Planning Problem for Needle Steering

We assume a motion plan is discretized into time steps of equal, finite time duration $\Delta \in \mathbb{R}^+$. Hence, the corresponding time τ for time step t is $t\Delta$, where $t \in \mathbb{N}$. The stochastic nature of the needle motion and sensing models means that it is typically impossible to know the exact pose of the needle tip. Instead, the robot maintains a belief state, or probability distribution over all possible states. Formally, the belief state

$\mathbf{b}_t \in \mathbb{B}$ of the needle is the distribution of the needle state $\mathbf{x}_t$ at time t given all past control inputs and sensor measurements:

$$\mathbf{b}_t = p[\mathbf{x}_t | \mathbf{u}_0, \ldots, \mathbf{u}_{t-1}, \mathbf{z}_1, \ldots, \mathbf{z}_t]. \quad (2)$$

We approximate the distribution of the needle state $\mathbf{x}_t$ at any time step t using a Gaussian distribution $\mathbf{x}_t \sim \mathcal{N}(\hat{\mathbf{x}}_t, \Sigma_t)$, where $\hat{\mathbf{x}}_t$ is the mean and Σ_t is the covariance matrix. We represent the belief state $\mathbf{b}_t$ by the Gaussian distribution of $\mathbf{x}_t$.

Given a control input $\mathbf{u}_t$ and a measurement $\mathbf{z}_{t+1}$, the belief state is updated using Bayesian filtering:

$$\mathbf{b}_{t+1} = \beta(\mathbf{b}_t, \mathbf{u}_t, \mathbf{z}_{t+1}). \quad (3)$$

The output of the motion planner is a plan that specifies controls that guide the needle from its start pose $\mathbf{x}_0$ to a goal pose $\mathbf{x}_l^*$ while avoiding obstacles. Due to uncertainty in motion and sensing, we compute a plan $\pi_t : \mathbb{B} \to \mathbf{U}$ that is defined as a policy over the belief space, i.e., $\mathbf{u}_t = \pi_t(\mathbf{b_t})$ for $t = 0, \ldots, l-1$ for some finite l.

B. Optimization Objective

Our objective is not only to find a policy that reaches the goal, but to find a policy that minimizes costs. Formally, our objective is to compute a policy π_t minimizing expected costs

$$\mathbb{E}_{\mathbf{z}_1,\ldots,\mathbf{z}_l}\left[c_l(\mathbf{b}_l) + \sum_{t=0}^{l-1} c_t(\mathbf{b}_t, \mathbf{u}_t)\right], \quad (4)$$

subject to Eq. (3) for all $0 \le t \le l-1$. We consider a parameterized cost function specifically designed for medical steerable needles.

First, we encode in the cost function the desire to reach the goal pose with minimal error. To minimize the expected deviation of the needle tip pose from the goal pose, we define the cost function at the final time step as

$$c_l(\mathbf{b}) = (\hat{\mathbf{x}} - \mathbf{x}_l^*)^\top Q_l (\hat{\mathbf{x}} - \mathbf{x}_l^*) + \mathrm{tr}[\sqrt{\Sigma_t} Q_l \sqrt{\Sigma_t}], \quad (5)$$

where Q_l is a pre-defined positive semi-definite matrix.

Second, we encode in the cost function clinically desirable properties for a plan that steers a needle to a goal. We define the local cost function c_t as

$$c_t(\mathbf{b}, \mathbf{u}) = (\mathbf{u} - \mathbf{u}^*)^\top R_t (\mathbf{u} - \mathbf{u}^*) + \mathrm{tr}[\sqrt{\Sigma_t} Q_t \sqrt{\Sigma_t}] + f(\mathbf{b}), \quad (6)$$

where R_t and Q_t are positive semi-definite matrices and $\mathbf{u}^*$ is a user defined nominal control input. As discussed in the experiments, we set $\mathbf{u}^*$ to penalize overly slow or fast insertion speeds and twist speeds. The function $f(\mathbf{b})$ is a cost term that enforces obstacle avoidance during plan optimization. We use the function $f(\mathbf{b})$ introduced by van den Berg et al. [3]. It returns $+\infty$ if the state $\hat{\mathbf{x}}$ centered at $\mathbf{b}$ is in collision with an obstacle. Otherwise, the function $f(\mathbf{b})$ returns a conservative estimate of the probability of collision of the needle tip with obstacles given the belief state $\mathbf{b}$.

IV. Needle Steering Belief Dynamics

To solve the belief space planning problem formulated above, we first must analytically define the belief state update in Eq. 3. We approximate the belief dynamics using an Extended Kalman Filter (EKF), which is applicable to Gaussian beliefs. We first introduce the needle's stochastic dynamics defined in the state space in Sec. IV-A by following the formulation in [5]. We then vectorize the stochastic dynamics in Sec. IV-B. Combining the vectorized stochastic dynamics with the general sensing model defined in Eq. 1, we formulate the stochastic belief dynamics for needle steering using EKF in Sec. IV-C.

A. Needle Stochastic Dynamics

Following the definitions in [5], let us define the state of the needle tip as $X \in SE(3)$, where $X = \begin{bmatrix} R & \mathbf{p} \\ \mathbf{0}^\top & 1 \end{bmatrix}$, $R \in SO(3)$ is the rotation matrix describing the tip's orientation, and $\mathbf{p} = [x, y, z]$ represents the 3D tip position.

The needle's base can be axially rotated, which changes the direction of the bevel tip and hence changes the steering direction. When the needle is inserted into tissue without axial rotation, the needle curves with an arc of curvature κ_0. With duty-cycled high-speed axial spinning of the needle [10], the needle can bend with variable curvature $\kappa \in [0, \kappa_0]$, where κ_0 is the maximum achievable curvature. Continuous high-speed spinning results in a straight needle trajectory, while no spinning achieves maximum curvature. By introducing a high level control input w, which is the angular velocity applied at the base regardless of the angular velocity used for duty-cycling, we define a high-level control input $\mathbf{u}(\tau) = \begin{bmatrix} v(\tau) & w(\tau) & \kappa(\tau) \end{bmatrix}^\top \in \mathbb{R}^3$, where τ is the time, $v(\tau)$ is the insertion speed, $w(\tau)$ is the rotation speed, and $\kappa(\tau)$ is the curvature ($0 \le \kappa(\tau) \le \kappa_0$). We use this definition of $\mathbf{u}$ from this point forward.

The continuous-time dynamics of the motion of the needle tip is given as

$$X'(\tau) = X(\tau) U(\tau). \quad (7)$$

The 4×4 matrix $U \in se(3)$ from Eq. (7) is represented as:

$$U(\tau) = \begin{bmatrix} [\mathbf{w}(\tau)] & \mathbf{v}(\tau) \\ \mathbf{0}^\top & 0 \end{bmatrix}, \quad (8)$$

where

$$\mathbf{w}(\tau) = \begin{bmatrix} \kappa(\tau) v(\tau) & 0 & w(\tau) \end{bmatrix}^\top, \ \mathbf{v}(\tau) = \begin{bmatrix} 0 & 0 & v(\tau) \end{bmatrix}^\top, \quad (9)$$

and the notation $[\mathbf{a}]$ for a vector $\mathbf{a} \in \mathbb{R}^3$ refers to a 3×3 skew-symmetric matrix $[\mathbf{a}] = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$.

To model motion uncertainty, we follow [5] by corrupting the control input U by additive noise $\tilde{U}$ sampled from a zero-mean Gaussian distribution with a covariance matrix M:

$$\tilde{U} = \begin{bmatrix} [\tilde{\mathbf{w}}] & \tilde{\mathbf{v}} \\ \mathbf{0}^\top & 0 \end{bmatrix}, \quad \begin{bmatrix} \tilde{\mathbf{w}} \\ \tilde{\mathbf{v}} \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, M).$$

Hence, the stochastic continuous-time dynamics becomes

$$X' = X(U + \tilde{U}). \quad (10)$$

Assuming the control input U and the additive noise $\tilde{U}$ is constant for a small time duration Δ between time $t\Delta$ to $t\Delta+\Delta$, the stochastic discrete-time dynamics can be computed in closed form as

$$X_{t+1} = X_t \exp(\Delta(U_t + \tilde{U}_t)), \tag{11}$$

where $X_{t+1} = X(t\Delta+\Delta)$ and $X_t = X(t\Delta)$.

B. Vectorization of the Needle Stochastic Dynamics

Our belief space planning algorithm in Sec. V requires that we vectorize the stochastic dynamics (Eq. (11)). We do this by mapping the rotation matrix R in X from the $SO(3)$ group to the $so(3)$ algebra. In $SE(3)$ we represent the state of the needle tip by its 3D position and its orientation represented by a rotation matrix. The $SO(3)$ group and $so(3)$ algebra are related with each other via the matrix exponential and matrix logarithm operation: $\exp : so(3) \to SO(3)$ and $\log : SO(3) \to so(3)$ [30]. We use an operation $^\wedge : \mathbb{R}^6 \to SE(3)$ that maps $\mathbf{x} = \begin{bmatrix}\mathbf{p} & \mathbf{r}\end{bmatrix}^\top$, where $\mathbf{p} \in \mathbb{R}^3$ is the 3D position of the tip and $\mathbf{r} \in \mathbb{R}^3$ is the tip's orientation represented in an axis-angle coordinate, to $X \in SE(3)$ as

$$\mathbf{x}^\wedge = \begin{bmatrix}\mathbf{p}\\ \mathbf{r}\end{bmatrix}^\wedge = \begin{bmatrix}\exp([\mathbf{r}]) & \mathbf{p}\\ \mathbf{0}^\top & 1\end{bmatrix}.$$

We also use an operation $^\vee : SE(3) \to \mathbb{R}^6$ to map X to $\mathbf{x}$:

$$X^\vee = \begin{bmatrix}R & \mathbf{p}\\ \mathbf{0}^\top & 1\end{bmatrix}^\vee = \begin{bmatrix}\mathbf{p}\\ \langle \log(R) \rangle\end{bmatrix},$$

where $\langle S \rangle$ maps a skew-symmetric matrix S to a 3D vector, which is exactly the inverse operation of $[\mathbf{a}]$. Now, we are ready to define a vectorized stochastic dynamics. Given $\mathbf{x}_t = \begin{bmatrix}\mathbf{p}_t & \mathbf{r}_t\end{bmatrix}^\top$ and $\mathbf{u}_t = \begin{bmatrix}v_t, w_t, \kappa_t\end{bmatrix}^\top$, the vectorized stochastic dynamics $\mathbf{f} : \mathbb{R}^6 \times \mathbb{R}^3 \times \mathbb{R}^6 \to \mathbb{R}^6$ is

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{m}_t) = ((\mathbf{x}_t)^\wedge \exp(\Delta(U_t + \tilde{U}_t)))^\vee, \tag{12}$$

where U_t is derived from $\mathbf{u}_t$ following Eqs. (9) and (8), $\mathbf{m}_t = \begin{bmatrix}\tilde{\mathbf{w}}_t & \tilde{\mathbf{v}}_t\end{bmatrix}^\top \sim \mathcal{N}(\mathbf{0}, M)$, and $\tilde{U}_t$ is derived from $\mathbf{m}_t$ following Eq. (8).

C. Needle Belief Dynamics

With the vectorized stochastic dynamics in Eq. (12) and the stochastic sensing model in Sec. III, we derive the stochastic belief dynamics. The general Bayesian filter in Eq. (3) depends on the measurement $\mathbf{z}_{t+1}$. Since the measurement is unknown in advance, the belief dynamics becomes stochastic. Following the derivation in [3], we use the Extended Kalman Filter (EKF) and derive the belief dynamics assuming $\mathbf{z}_{t+1}$ is random:

$$\begin{aligned}\hat{\mathbf{x}}_{t+1} &= \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{0}) + \mathbf{w}_t, \;\; \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, K_t H_t T_t),\\ \Sigma_{t+1} &= T_t - K_t H_t T_t,\end{aligned} \tag{13}$$

where

$$\begin{aligned}T_t &= A_t \Sigma_t A_t^\top + V_t M V_t^\top, \;\; A_t = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}[\hat{\mathbf{x}}_t, \mathbf{u}_t, \mathbf{0}],\\ V_t &= \frac{\partial \mathbf{f}}{\partial \mathbf{m}}[\hat{\mathbf{x}}_t, \mathbf{u}_t, \mathbf{0}], \;\; H_t = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}[\mathbf{f}(\hat{\mathbf{x}}_t, \mathbf{u}_t, \mathbf{0})],\\ K_t &= T_t H_t^\top (H_t T_t H_t^\top + V(\mathbf{f}(\hat{\mathbf{x}}_t, \mathbf{u}_t, \mathbf{0}))).\end{aligned}$$

Since we assume $\mathbf{x}_t \sim \mathcal{N}(\hat{\mathbf{x}}_t, \Sigma_t)$, we can write $\mathbf{b}_t = \begin{bmatrix}\hat{\mathbf{x}}_t\\ \mathrm{vec}(\sqrt{\Sigma_t})\end{bmatrix}$, which is a vector that consists of the mean $\hat{\mathbf{x}}_t$ and the columns of $\sqrt{\Sigma_t}$. In a vector version, the above stochastic belief dynamics can be written as

$$\mathbf{b}_{t+1} = \mathbf{g}(\mathbf{b}_t, \mathbf{u}_t) + W(\mathbf{b}_t, \mathbf{u}_t)\xi_t, \;\; \xi_t \sim \mathcal{N}(\mathbf{0}, I_{6\times 6}), \tag{14}$$

where

$$W(\mathbf{x}_t, \mathbf{u}_t) = \begin{bmatrix}\sqrt{K_t H_t T_t}\\ 0\end{bmatrix}. \tag{15}$$

Eq. 14 is a stochastic version of Eq. 3 since the observation $\mathbf{z}_{t+1}$ is treated as a random variable.

V. Needle Steering Belief Space Planning

We use an iterative optimization-based approach for belief space planning [3]. The belief space planner requires as input the stochastic dynamics of the needle, the stochastic sensing model, the environment scenario (e.g., the needle's initial state, the goal state, and obstacle geometry), and a user-defined cost function as described in Sec. III. The output is a locally optimal control policy that minimizes the cost function. Given the computed control policy, we can compute a collision-free nominal trajectory by shooting the control policy from the initial state. More importantly, we can use the control policy for closed-loop execution with sensor feedback.

A. Computing Costs

A general approach for solving the POMDP problem is *value iteration*. We define the cost-to-go function $v_t : \mathbb{B} \to \mathbb{R}$, which takes the belief state $\mathbf{b}$ at time step t as input and computes the minimum expected future cost that will be accrued between time step t and time step l if the robot starts at $\mathbf{b}$ at time step t. Value iteration starts at time step l by setting $v_l(\mathbf{b}) = c_l(\mathbf{b})$, and iteratively computes the cost-to-go functions and control policy by moving backward in time using

$$v_t(\mathbf{b}_t) = \min_{\mathbf{u}_t}(c_t(\mathbf{b}_t, \mathbf{u}_t) + \mathbb{E}_{\mathbf{z}_t}[v_{t+1}(\beta(\mathbf{b}_t, \mathbf{u}_t, \mathbf{z}_t))]), \tag{16}$$

$$\pi_t(\mathbf{b}_t) = \arg\min_{\mathbf{u}_t}(c_t(\mathbf{b}_t, \mathbf{u}_t) + \mathbb{E}_{\mathbf{z}_t}[v_{t+1}(\beta(\mathbf{b}_t, \mathbf{u}_t, \mathbf{z}_t))]). \tag{17}$$

B. Computing a Locally Optimal Solution to the POMDP

The optimization starts from a feasible (e.g., collision free, dynamically feasible) plan, which can be computed using a sampling-based motion planner (e.g., RRT [31]). We require that the initial plan inserts the needle during rotations; the local optimization cannot directly handle motion plans computed by planners that stop insertion to axially rotate the needle (e.g., [32]). The initial plan is defined in the robot's state space, so we compute the corresponding nominal trajectory in belief space by shooting the controls from the initial plan using Eq. 14 with zero noise.

The local optimization then begins iterating. Given a nominal trajectory defined in the belief space $[\mathbf{b}_0^k, \mathbf{u}_0^k, ..., \mathbf{b}_{l-1}^k, \mathbf{u}_{l-1}^k, \mathbf{b}_l^k]$ at the k'th iteration, the iLQG-based method from [3] first linearizes the belief dynamics and quadratizes the local cost function around the trajectory.

With the linearized dynamics and the quadratized cost function, value iteration start from the last time step l and recursively computes the cost-to-go functions following the Bellman Equation in Eq. (16). The cost-to-go functions v_t for $0 \leq t \leq l$ become quadratic in terms of the belief state and control input. Minimizing the cost-to-go v_t with respect to $\mathbf{u}_t$, we compute a linear control policy in the form of

$$\mathbf{u}_t^{k+1} - \mathbf{u}_t^k = L_t(\mathbf{b}_t^{k+1} - \mathbf{b}_t^k) + I_t, \tag{18}$$

for the $(k+1)$'th iteration.

By shooting the control policy from the initial belief, we can compute a new nominal trajectory. To ensure the iterative procedure converges to a locally optimal solution, we augment the iterative procedure with line search. Specifically, the new computed sequence of controls $\mathbf{u}_t$ will be adjusted by the line search if it leads to a higher expected total cost. If the new computed trajectory at the $(k+1)$'th iteration is in collision with obstacles (c_t returns $+\infty$ in this case), the line search will adjust the controls by pulling the trajectory closer to the nominal trajectory from the k'th iteration until a collision free trajectory is found. Hence, if the initial trajectory is collision free, this approach guarantees a collision free nominal trajectory upon convergence.

Upon convergence, the result is a locally optimal control policy with respect to the objective function. This iLQG-based iterative approach performs in a similar manner to a Newton method. However, we do not achieve a second-order convergence rate since we linearize the dynamics and do not explicitly compute the Hessian matrix of the objective function. We refer readers to [3] for more details.

VI. Evaluation

We demonstrate our approach in simulation for steerable needles navigating in 3D environments with obstacles. We tested our C++ implementation on a 3.7 GHz Intel i7 PC.

We define the cost functions $c_t(\mathbf{b}, \mathbf{u})$ and $c_l(\mathbf{b})$ in a manner that quantifies costs associated with medical needle steering. We set $R_l = 800I$, $R_t = I$, and $Q_t = 10I$. We also set $\mathbf{u}^* = [v^*, 0, \alpha\kappa_0]^\top$, where $v^* = 1$ cm/s, $\kappa_0 = 2.5$ cm^{-1}, and $\alpha = 0.5$ in our experiments. This penalizes insertion speeds that are faster or slower than a user-specified ideal insertion speed v^*. It also penalizes curvatures that are too large (close to the kinematic limits of the device) and too small (requiring high-rate duty cycling, which may cause tissue damage).

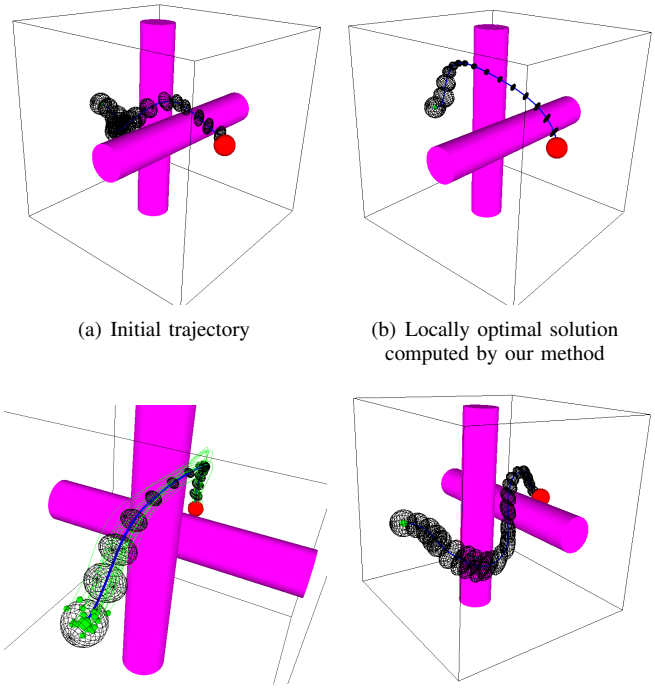
We evaluate plans using several criteria relevant to needle steering. The first is probability of collision with obstacles. The second is target error, which is Euclidean distance between the target and the needle's tip position at the end of the insertion. The third is the average curvature deviation, which is the average deviation from the desired curvature $\alpha\kappa_0$ along a plan ($\sum_{t=0}^{l-1} \|\kappa_t - \alpha\kappa_0\|$). We also consider average rotation speed excluding duty cycling ($\frac{1}{l}\sum_{t=0}^{l-1} |w_t|$) and average insertion speed ($\frac{1}{l}\sum_{t=0}^{l-1} v_t$) over the duration of the needle insertion.

A. Cylindrical Obstacles Scenario

We first consider a cubic environment with two perpendicular cylindrical obstacles as shown in Fig. 2. During execution, we assume that the (x, y, z) position of the needle can be sensed by a sensor (e.g., 3D ultrasound) that is mounted at the top of the cubic workspace. The sensor provides a more accurate measurement when the needle tip is closer to the top of the workspace. This gives us the following sensing model:

$$\mathbf{z}_t = \mathbf{p}_t + \mathbf{n}_t, \;\; \mathbf{n}_t \sim \mathcal{N}(\mathbf{0}, ((y_t - \hat{y})^2 + \gamma)cI_{3\times 3}), \tag{19}$$

where $\mathbf{p}_t$ is the 3D position of the tip at any time step t, $\gamma \in \mathbb{R}^+$, $c \in \mathbb{R}^+$, and $\hat{y}$ is the position of the sensor on the y-axis. When the tip of the needle is closer to the top of the workspace, the variance of the noise becomes smaller.



(a) Initial trajectory (b) Locally optimal solution computed by our method

(c) Simulated closed-loop executions (d) Best solution from LQG-MP

Fig. 2. The cylindrical obstacles scenario has two perpendicular cylindrical obstacles (magenta) in a cubic environment in which the faces of the cube are also obstacles. The red sphere represents the goal state. The green sphere represents the insertion position of the needle. (a) An initial trajectory computed using RRT. (b) The nominal trajectory and the associated beliefs (3 standard deviations shown by wireframe ellipses) of the locally optimal solution computed using our approach. (c) Twenty simulated executions of closed-loop execution using the computed control policy where the needle's state at each stage is shown by a small green sphere. (d) The solution computed by LQG-MP and the associated estimated beliefs.

Fig. 2 shows the results of our method. We computed an initial trajectory, shown in Fig. 2(a), using RRT. The nominal trajectory of the locally optimal solution computed by our motion planner, together with the associated beliefs, are shown in Fig. 2(b). The locally optimal solution first guides the needle toward the top of the workspace to better localize the needle and then approaches the goal state. Compared to the initial trajectory computed by RRT, we see for the solution computed by our method that (1) the three-standard-deviation ellipses do not collide with any obstacles and (2) the uncertainty at the final time step is much smaller, which results in a smaller expected deviation to the goal. Note that the optimized plan is only locally optimal, in

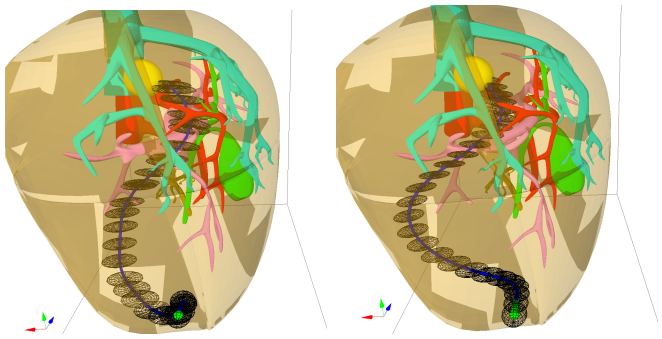
	Our Method	LQG-MP	RRT+LQG
Probability of collision	0%	0%	44%
Target error (cm)	0.08±0.02	0.2±0.004	0.3±0.007
Avg. curvature deviation (cm^{-1})	0.55±6e-4	0.66±8e-5	0.68±7e-5
Avg. rotation speed (rad/s)	0.45±0.1	1.6±7e-5	1.74±2e-4
Avg. insertion speed (cm/s)	0.948±2e-4	0.78±0.002	0.82±0.001
Computation time (s)	26.91	857	0.847

TABLE I

PERFORMANCE OF OUR METHOD, LQG-MP, AND LQG CONTROL ON A RRT SOLUTION ON THE CYLINDRICAL OBSTACLES SCENARIO. MEANS AND STANDARD DEVIATIONS SHOWN FOR 2ND-5TH ROWS.

the sense that it is in the same homotopic class as the initial trajectory. Fig. 2(c) shows 20 simulated closed-loop executions of the control policy computed by our method. Fig. 2(d) shows the best solution found by LQG-MP [5]. The LQG-MP solution is substantially different from the locally optimal solution computed by our approach. The objective of LQG-MP is to minimize a cost function that is correlated with probability of collision. LQG-MP selects a solution in which the needle moves far away from the obstacles to decrease the probability of the collisions. On the other hand, the locally optimal solution from our method achieves a low probability of collision by decreasing the uncertainty along the trajectory, which encourages the needle to move through regions with more accurate sensing. Hence, the locally optimal solution computed by our approach has smaller uncertainty along the entire trajectory and it does not have to steer the needle as far away from the obstacles. Our approach aims to keep uncertainty low along the entire trajectory and simultaneously penalizes control efforts that deviate from a specified quantity.

We also compared the qualities of the plan generated by our approach in Fig. 2(b) and the plan from LQG-MP in Fig. 2(d). For both plans, we simulated 1,000 closed-loop executions of their control policy and show the results in Table I. The simulated executions of both plans never collided with obstacles. However, due to the lack of local optimization, the average deviation to the target is higher for LQG-MP than for our approach. This is because our approach explicitly optimizes the expected deviation to the target. Similarly, compared to LQG-MP, our optimization-based approach yields a lower average curvature deviation and rotation and insertion speeds closer to the ideal specifications. In terms of computation time, our approach requires 26 seconds to achieve the locally optimal solution shown in Fig. 2(b). For LQG-MP, in our current implementation we use a general RRT motion planner to generate 1,000 feasible plans, which takes 847 seconds. We also executed the initial plan in Fig. 2(a) with LQG feedback control, of which the statistics are reported in the third column (RRT+LQG). Although this approach is computationally fast, the probability of collision in 1,000 simulated executions was high (44%). We also executed the initial plan in an open-loop manner without a controller for 1,000 simulated runs. Each of these runs resulted in failure, which illustrates the significant impact of uncertainty in this scenario.



(a) Initial trajectory (b) Locally optimal solution computed by our method

Fig. 3. The objective is to steer the needle from a start pose outside the liver to a clinical target inside the liver while avoiding critical vasculature and ducts. We assume the sensing model is an X-ray imager pointing in the x direction (red arrow), which provides a 2D measurement of the tip's y and z position. (a) An initial trajectory computed using RRT. (b) A locally optimal solution computed using our method.

B. 3D Liver Biopsy Scenario

We next consider in simulation the scenario of steering a needle through liver tissue to reach a target for biopsy for cancer diagnosis while avoiding critical vasculature (see Fig. 1). Similar to the sensor model in Eq. (19), we assume the tip position measurement is more accurate when the needle tip is closer to the sensor, although here the sensor is placed pointing in the x direction (red axis in Fig. 1).

The locally optimal solution computed by our method, shown in Fig. 1(a,b), first steers the needle to the right to obtain more accurate sensor measurements. This reduces uncertainty compared to the initial trajectory, shown in Fig. 1(c). We simulated closed-loop executions of the computed control policy 1,000 times. The probability of collision was 0.6%. The target error was 0.17±0.03 cm. We also computed an LQG controller for the RRT initial trajectory and simulated closed-loop execution of the controller 1,000 times. The resulting probability of collision was 59.9% and the target error averaged 0.26±0.005 cm.

We also evaluated our motion planner using a different sensing model in which the sensor (e.g., an X-ray projection imager) can only sense the y and z position of the needle tip, resulting in the following sensing model:

$$\mathbf{z}_t = \begin{bmatrix} y_t & z_t \end{bmatrix}^T + \mathbf{n}_t, \quad \mathbf{n}_t \sim \mathcal{N}(\mathbf{0}, N), \tag{20}$$

where N is a constant. This sensing model results in smaller uncertainty in the y and z direction but larger uncertainty along the x direction. Fig. 3 shows an initial trajectory computed by RRT as well as the locally optimal solution computed using our method for this sensing model. The plan computed by our method (Fig. 3(b)) steers the needle far away from the hepatic veins (cyan) and then passes above the portal veins (pink). We ran LQG control on the initial trajectory for 1,000 simulated executions, resulting in a 48.7% probability of collision and a 0.29±0.02 cm average target error. We also ran the locally optimal control policy computed using our approach for 1,000 simulations, resulting in a 0.4% probability of collision and a 0.18±0.05 cm average target error.

VII. Conclusion

We introduced an optimization-based motion planner for needle steering that explicitly considers uncertainty in the needle's motion and sensing of system state. Our method formulates the problem of needle steering under uncertainty as a POMDP and computes (locally) optimal trajectories and control policies in belief space. To enable optimization in belief space, we first represent the needle dynamics model defined in the $SE(3)$ group in a vector form and then create cost functions compatible with belief space optimization that explicitly consider the avoidance of obstacles, target acquisition accuracy, penalties for unsafe control inputs, and the diversity of medical sensor modalities.

To the best of our knowledge, this is the first approach that locally optimizes steerable needle motion plans in belief space. In future work we plan to address some of the limitations of our POMDP formulation. We currently assume the beliefs can be reasonably approximated as Gaussian distributions, as is commonly done in many applications. Due to the steerable needle's kinematics, the distribution of the tip position in Cartesian coordinates in many cases would be better modeled using a banana-shaped distribution [33]. In future work we plan to use the matrix exponential map introduced in [33], which will introduce new challenges for estimating probability of collision with obstacles. We also plan to integrate our approach with a physical system and evaluate system performance in biological tissues using ultrasound and X-ray projection images.

References

[1] N. J. Cowan, K. Goldberg, G. S. Chirikjian, G. Fichtinger, R. Alterovitz, K. B. Reed, V. Kallem, W. Park, S. Misra, and A. M. Okamura, "Robotic needle steering: design, modeling, planning, and image guidance," in *Surgical Robotics: System Applications and Visions*, J. Rosen, B. Hannaford, and R. M. Satava, Eds. Springer, 2011, ch. 23, pp. 557–582.

[2] C. Papadimitriou and J. Tsisiklis, "The complexity of Markov decision processes," *Mathematics of Operations Research*, vol. 12, no. 3, pp. 441–450, 1987.

[3] 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, Sept. 2012.

[4] W. Park, Y. Wang, and G. S. Chirikjian, "The path-of-probability algorithm for steering and feedback control of flexible needles," *Int. J. Robotics Research*, vol. 29, no. 7, pp. 813–830, June 2010.

[5] 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.

[6] R. J. Webster III, J. S. Kim, N. J. Cowan, G. S. Chirikjian, and A. M. Okamura, "Nonholonomic modeling of needle steering," *Int. J. Robotics Research*, vol. 25, no. 5-6, pp. 509–525, May 2006.

[7] S. P. DiMaio and S. E. Salcudean, "Needle steering and motion planning in soft tissues," *IEEE Trans. Biomedical Engineering*, vol. 52, no. 6, pp. 965–974, June 2005.

[8] D. Glozman and M. Shoham, "Image-guided robotic flexible needle steering," *IEEE Trans. Robotics*, vol. 23, no. 3, pp. 459–467, June 2007.

[9] S. Ko, L. Frasson, and F. Rodriguez y Baena, "Closed-loop planar motion control of a steerable probe with a "programmable bevel" inspired by nature," *IEEE Trans. Robotics*, vol. 27, no. 5, pp. 970–983, 2011.

[10] 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.

[11] R. Alterovitz, M. Branicky, and K. Goldberg, "Motion planning under uncertainty for image-guided medical needle steering," *Int. J. Robotics Research*, vol. 27, no. 11-12, pp. 1361–1374, Jan. 2008.

[12] 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.

[13] S. Patil, J. van den Berg, and R. Alterovitz, "Motion planning under uncertainty in highly deformable environments," in *Robotics: Science and Systems (RSS)*, June 2011.

[14] M. C. Bernardes, B. V. Adorno, P. Poignet, and G. A. Borges, "Robot-assisted automatic insertion of steerable needles with closed-loop imaging feedback and intraoperative trajectory replanning," *Mechatronics*, vol. 23, pp. 630–645, 2013.

[15] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[16] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.

[17] S. Thrun, "Monte Carlo POMDPs," in *Advances in Neural Information Processing Systems 12*, S. Solla, T. Leen, and K.-R. Müller, Eds. MIT Press, 2000, pp. 1064–1070.

[18] J. Pineau, G. Gordon, and S. Thrun, "Point-based value iteration: An anytime algorithm for POMDPs." *Int. Joint Conf. Artificial Intelligence (IJCAI)*, pp. 1025–1032, 2003.

[19] T. Smith and R. Simmons, "Heuristic search value iteration for POMDPs," in *Proc. UAI*, 2004, pp. 520–527.

[20] J. M. Porta, N. Vlassis, M. T. J. Spaan, and P. Poupart, "Point-based value iteration for continuous POMDPs," *J. Machine Learning Research*, vol. 7, pp. 2329–2367, 2006.

[21] 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.

[22] H. Bai, D. Hsu, W. S. Lee, and V. A. Ngo, "Monte Carlo value iteration for continuous-state POMDPs," in *Proc. Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2010, pp. 175–191.

[23] H. Bai, D. Hsu, and W. S. Lee, "Integrated perception and planning in the continuous space: A POMDP approach," in *Robotics: Science and Systems (RSS)*, 2013.

[24] 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.

[25] 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.

[26] 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, June 2011.

[27] 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.

[28] T. Erez and W. D. Smart, "A scalable method for solving high-dimensional continuous POMDPs using local approximation," in *Proc. Conf. on Uncertainty in Artificial Intelligence (UAI)*, 2010.

[29] R. Platt and L. Kaelbling, "Efficient planning in non-Gaussian belief spaces and its application to robot grasping," in *Int. Symp. Robotics Research (ISRR)*, 2011.

[30] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL: CRC Press, 1994.

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

[32] 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)*, Sept. 2010, pp. 893–899.

[33] A. W. Long, K. C. Wolfe, M. J. Mashner, and G. S. Chirikjian, "The banana distribution is Gaussian: A localization study with exponential coordinates," in *Robotics: Science and Systems (RSS)*, July 2012.