# Planning Locally Optimal, Curvature-Constrained Trajectories in 3D using Sequential Convex Optimization
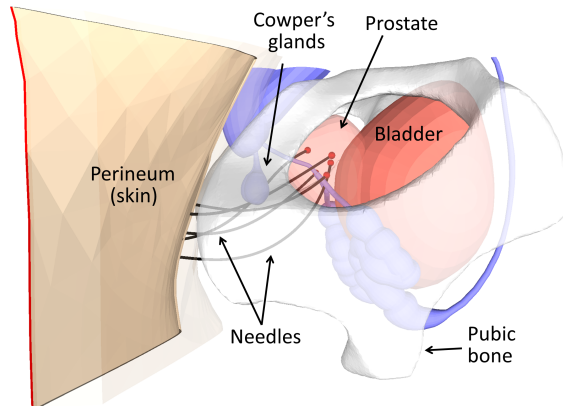
Yan Duan, Sachin Patil, John Schulman, Ken Goldberg, Pieter Abbeel

*Abstract*— 3D curvature-constrained motion planning finds applications in a wide variety of domains, including motion planning for flexible, bevel-tip medical needles, planning curvature-constrained channels in 3D printed implants for targeted brachytherapy dose delivery or channels for cooling turbine blades, and path planning for unmanned aerial vehicles (UAVs). In this work, we present a motion planning technique using sequential convex optimization for computing locally optimal, curvature-constrained trajectories to desired targets while avoiding obstacles in 3D environments. We report two main contributions in this work: (i) curvature-constrained trajectory optimization in 6D pose (position and orientation) space, and (ii) planning multiple trajectories that are mutually collision-free. We demonstrate the performance of our approach on two clinically motivated applications. Our experiments indicate that our approach can compute high-quality plans for medical needle steering in $1.6$ seconds on a commodity PC, enabling re-planning during execution to correct for perturbations. Our approach can also be used for designing optimized channel layouts within 3D printed implants for intracavitary brachytherapy.

## I. INTRODUCTION

The need for curvature-constrained motion planning arises in a wide variety of domains. For instance, a new class of highly flexible, bevel-tip needles are being developed that enable the needle to move along constant curvature trajectories within tissue when a forward pushing force is applied and the direction of motion can be changed by reorienting the bevel tip through twisting of the needle at its base [29]. These steerable needles offer improved maneuverability within tissue during insertion and greater targeting accuracy, offering improved procedure outcomes. They also facilitate access to previously inaccessible clinical targets while avoiding obstacles such as sensitive anatomical tissues (e.g., vital organs and vessels) and impenetrable structures (e.g., bones), as shown in Fig. 1(a). However, guiding a steerable needle around obstacles under image guidance by manipulating the needle at its base requires reasoning in 6D pose space and is not intuitive for a physician. These problems can be addressed with the assistance of motion planning for computing safe and feasible needle motions that avoid anatomical obstacles and reach clinical targets.

Another important application is the design of multiple bounded curvature channels in intracavitary 3D printed implants through which a radioactive source is guided for delivering radiation doses for high dose rate brachytherapy (HDR-BT) (Fig. 1(b)) [12]. The need for designing such channels also arises in applications such as turbine blade

Authors are with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, CA, USA. {dementrock, sachinpatil, joschu, goldberg, pabbeel}@berkeley.edu

(a) Multiple bevel-tip flexible needles inserted through the perineum to reach targets deep within the prostate.
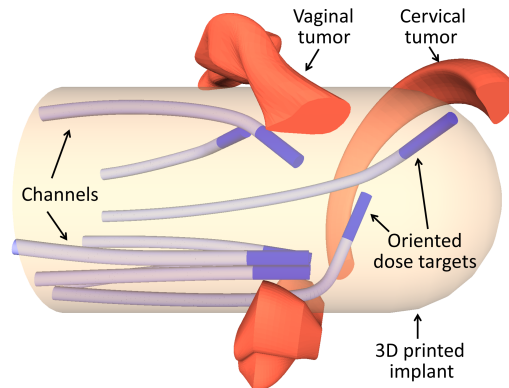


(b) Bounded curvature channels within 3D printed vaginal implants for delivering radiation to OB/GYN tumors.

Fig. 1. High-quality, curvature-constrained trajectories computed using our optimization-based approach for (a) medical needle steering [29], and (b) designing channel layouts for intracavitary brachytherapy [12].

design for delivering coolant through the blades to cool them during operation [13], and planning bounded curvature trajectories for unmanned aerial vehicles (UAVs) [32].

Computing collision-free, curvature-constrained trajectories in 3D environments with obstacles is challenging because it requires planning in the $SE(3)$ configuration space consisting of the 6D pose (position and orientation). Our approach builds on recent advances in trajectory optimization using sequential convex optimization [26]. In this work, we report two main contributions. First, we extend prior work by considering trajectory optimization problems over manifolds such as the $SE(3)$ Lie group, instead of just vector spaces of the form $\mathbb{R}^n$. We accomplish this by iteratively optimizing over increments to the trajectory, defined in terms of the corresponding Lie algebra ($\mathfrak{se}(3)$ in our case) [25]. Second,

we consider the problem of planning multiple trajectories that are mutually collision-free, which arises in planning trajectories for multiple needles for medical procedures [30], multiple channels in intracavitary implants for HDR-BT [12], or simultaneously planning for multiple UAVs [27].

We apply our motion planning technique to two real-world clinical applications. First, we consider the problem of medical needle steering, where the objective is to compute collision-free, constant curvature needle trajectories that avoid obstacles in the environment and optimize clinically relevant metrics such as minimizing the length of the plan or minimizing the total rotations applied at the needle base. We show that our approach can compute locally optimal plans in 1.6 seconds in complex environments on a commodity PC. Our second application considers the problem of channel layout design within intracavitary 3D printed implants, where the objective is to plan multiple curvature-constrained channels that are mutually collision-free. Our optimization-based approach can compute high-quality channel layouts with a higher planning success rate, while offering comparable computation times as compared to a highly optimized sampling-based RRT planner [12].

## II. RELATED WORK

Planning a curvature-constrained shortest path in a 2D plane between two configurations for a Dubins car robot has been extensively studied [7], [24]. Webster et al. [29] experimentally showed that bevel-tipped steerable needles follow paths of constant curvature when inserted into tissue. Planning constant curvature trajectories for steerable needles in a plane has received considerable attention [2], [5].

Computing collision-free, curvature-constrained trajectories in 3D environments requires planning in the 6D configuration space consisting of both position and orientation. Existing optimal motion planning approaches that rely on discretizing the configuration space [23] or sampling-based planners like RRT* [15] require solving a two-point boundary value problem (BVP) for connecting two states in $SE(3)$, closed-form solutions for which are not known [3]. Duindam et al. proposed a fast, optimal planner based on inverse kinematics [9], but this approach does not consider obstacle avoidance. Xu et al. [31], [30] used rapidly-exploring random trees (RRT) [16] which offers a probabilistically-complete, but computationally intensive, algorithm to search for collision-free trajectories. Duindam et al. [8] formulated planning for steerable needles as a non-convex optimization problem, which computes collision-free solutions in a few seconds but collision avoidance is treated as a cost and not as a hard constraint. Patil et al. [21], [22] proposed a RRT planner which plans bounded curvature trajectories for a needle by relying on duty-cycled spinning of the needle during insertion [18], [17]. However, this can cause excessive tissue damage [10]. This approach was also used for designing bounded curvature channels within implants [12] but the issue of optimality of channel layout was not addressed. In recent years, extensions to planning curvature-constrained trajectories in 3D have been proposed for unmanned aerial

vehicles (UAVs) in environments without obstacles [27], and with obstacles [14], [32]. These methods do not consider the problem of planning constant curvature trajectories in 3D.

Prior work on planning and trajectory optimization on Lie groups has proposed Newton-like optimization methods [1], direct (collocation) methods for trajectory optimization for continuous time optimal control problems [25], and primitive-based motion planning [11]. However, these approaches do not address the issue of avoiding collisions with obstacles in the environment.

## III. PROBLEM DEFINITION AND FORMULATION

We assume that a trajectory is discretized into time intervals $\mathcal{T} = \{0, 1, \ldots, T\}$. At each time step $t \in \mathcal{T}$, a trajectory waypoint is parameterized by a pose $X_t = \begin{bmatrix} R_t & \mathbf{p}_t \\ \mathbf{0}_3^T & 1 \end{bmatrix} \in SE(3)$, where $\mathbf{p}_t \in \mathbb{R}^3$ is the position and $R_t \in SO(3)$ is the rotation matrix that encodes the orientation of the waypoint frame relative to a world coordinate frame (Fig 2).

The planning objective can then be stated as:

**Input**: Set of obstacles $\mathcal{O}$, an entry zone $\mathcal{P}_{\text{entry}}$, a target zone $\mathcal{P}_{\text{target}}$, the maximum curvature $\kappa_{\max}$, and the discretization parameter $T$.

**Output**: Given an entry zone $\mathcal{P}_{\text{entry}}$ and a target zone $\mathcal{P}_{\text{target}}$, determine a locally optimal, collision-free, and curvature-constrained trajectory $\{X_t : t \in \mathcal{T}\}$ with $X_0 \in \mathcal{P}_{\text{entry}}$ and $X_T \in \mathcal{P}_{\text{target}}$, or report that no feasible trajectory can be found.
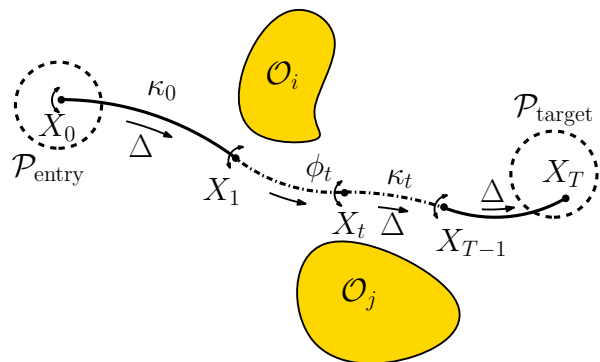


Fig. 2. A discretized curvature-constrained trajectory is parameterized as $\{X_0, \ldots, X_t, \ldots, X_T\}$, where $X_t \in SE(3)$ is the pose of the waypoint frame relative to a world coordinate frame at each time step $t$.

We first describe the curvature-constrained kinematic model used in this work and then present a formulation of the planning objective as a constrained, non-convex optimization problem. Although the formulation is specific to needle steering and channel planning, it can be easily generalized to other curvature-constrained planning problems.

### A. Curvature-Constrained Kinematic Model

In this work, we assume that the trajectory is composed of a sequence of $(T-1)$ circular arcs, each connecting a pose $X_t$ to the subsequent pose $X_{t+1}$ and of curvature $\kappa_t$. Depending on the application, the trajectory may be required to have a constant curvature $\kappa_t = \kappa_{\max}$ for all time steps, or a bounded curvature $0 \leq \kappa_t \leq \kappa_{\max}$ at each time step.

We make two design choices in formulating the curvature-constrained kinematics. First, we constrain the length of each circular arc $\Delta$ to be the same for all time steps. One can just as easily have a separate length parameter $\Delta_t$ for each time step. However, in our experiments, we observed that some of these $\Delta_t$ values shrink to 0 as a result of the optimization, producing large gaps between time steps which is not suitable for collision checking with obstacles in the environment.

Second, we use a "stop-and-turn" strategy for the kinematics, i.e., at each time step $t : 0 \leq t \leq T - 1$, we apply a rotation $\phi_t$ to the pose $X_t$ and then propagate the frame by a distance $\Delta$ to arrive at $X_{t+1}$. This is a natural choice for needle steering, since it corresponds to first twisting the base of the needle, and then pushing it forward, which induces less damage than constantly twisting the needle tip while pushing it. This strategy also results in channels that are easier for catheters to go through. See Fig. 2 for an illustration. Without loss of generality, we assume that the object (either the needle tip or a small trajectory segment for the channels) is oriented along the positive $z$-axis. Hence, the poses at adjacent time steps $X_t$ and $X_{t+1}$ are related as:

$$X_{t+1} = \exp(\mathbf{v}_t^\wedge) \cdot \exp(\mathbf{w}_t^\wedge) \cdot X_t, \qquad (1)$$

where $\mathbf{w}_t = \begin{bmatrix} 0\,0\,0\,0\,0\,\phi_t \end{bmatrix}^T$ and $\mathbf{v}_t = \begin{bmatrix} 0\,0\,\Delta\,\Delta\kappa_t\,0\,0 \end{bmatrix}^T$ are the twist vectors corresponding to the rotation $\phi_t$ and propagating the frame by distance $\Delta$, respectively. We refer the reader to the appendix and to the excellent treatise on the $SE(3)$ Lie group [19] for details on the $^\wedge : \mathbb{R}^6 \to \mathfrak{se}(3)$ and $\exp : \mathfrak{se}(3) \to SE(3)$ operators.

### B. Optimization Formulation

For notational convenience, we concatenate the states from all time steps as $\mathcal{X} = \{X_t : t \in \mathcal{T}\}$ and the control variables as $\mathcal{U} = \{\phi_t, \kappa_t : t \in \mathcal{T}, \Delta\}$. The planning objective is transcribed as a constrained, non-convex trajectory optimization problem as given below:

$$\min_{\mathcal{X}, \mathcal{U}} \alpha_\Delta \text{Cost}_\Delta + \alpha_\phi \text{Cost}_\phi + \alpha_\mathcal{O} \text{Cost}_\mathcal{O}, \qquad (2a)$$

subject to $\qquad\qquad\qquad\qquad\qquad\qquad\qquad (2b)$

$$\log(X_{t+1} \cdot (\exp(\mathbf{v}_t^\wedge) \cdot \exp(\mathbf{w}_t^\wedge) \cdot X_t)^{-1})^\vee = \mathbf{0}_6, \quad (2c)$$

$$X_0 \in \mathcal{P}_\text{entry}, \quad X_T \in \mathcal{P}_\text{target}, \qquad (2d)$$

$$\text{sd}(X_t, X_{t+1}, \mathcal{O}) \geq d_\text{safe} + d_\text{arc}, \qquad (2e)$$

$$-\pi \leq \phi_t \leq \pi, \qquad (2f)$$

$$\kappa_t = \kappa_\text{max} \quad \text{or} \quad 0 \leq \kappa_t \leq \kappa_\text{max}, \qquad (2g)$$

$$\Delta \sum_{t=0}^{T-1} \kappa_t \leq c_\text{max} \quad \text{for channel planning.} \qquad (2h)$$

The constraints and costs are described in detail below. We use sequential convex optimization to solve this trajectory optimization problem (Sec. IV).

*1) Kinematics Constraint (Eq. (2c)):* We transform the kinematic constraint from Eq. (1) to a standard non-convex equality constraint form by using the $\log$ map and relying on the identity $\log(\mathbf{I}_{4\times4}) = \mathbf{0}_6$. We refer the reader to the appendix for more details.

*2) Collision Constraint (Eq. (2e)):* We impose constraints to ensure that the trajectory avoids collisions, where $\text{sd}(X_t, X_{t+1}, \mathcal{O})$ is the signed distance between the trajectory segment in time interval $[t, t+1]$ and the set of obstacles $\mathcal{O}$. The signed distance corresponds to the minimum translation distance required to either put two geometric shapes in contact or separate them if they are overlapping. Two objects are non-colliding if the signed distance is positive, and we want to ensure that the trajectory has a user-defined safety margin $d_\text{safe}$. The distance between two convex shapes can be calculated by the Gilbert-Johnson-Keerthi (GJK) algorithm and the penetration depth is calculated by the Expanding Polytope Algorithm (EPA) [4]. We approximate the segment by the convex hull of the object (the needle tip or a small segment on the channel) between time $t$ and $t + 1$, and we account for the approximation error in rotation by adding an error correction term $d_\text{arc}$. Instead of numerically computing the gradient, we linearize the signed distance using the contact normal $\hat{\mathbf{n}}$. We refer the reader to Schulman et al. [26] for details on how to include the continuous-time non-convex collision avoidance constraint as an $\ell_1$ penalty in the optimization and how to compute the term $d_\text{arc}$.

*3) Total Curvature Constraint (Eq. (2h)):* For channel planning, we constrain the total curvature of the trajectory to ensure that catheters carrying the radioactive source can be pushed through the channels without buckling [12].

*4) Costs (Eq. (2a)):* To penalize tissue damage for needle steering and to optimize channel lengths for minimum radiation exposure, the objective imposes costs on the total length of the trajectory and the twists at each time step:

$$\text{Cost}_\Delta = T\Delta \quad \text{and} \quad \text{Cost}_\phi = \sum_{t=0}^{T-1} \phi_t^2. \qquad (3)$$

For needle steering, we add an extra term to favor large minimum clearance from obstacles to deal with expected needle deflections during execution:

$$\text{Cost}_\mathcal{O} = -\min_{\substack{0 \leq t \leq T-1 \\ \mathcal{O}_i \in \mathcal{O}}} \text{sd}(X_t, X_{t+1}, \mathcal{O}_i). \qquad (4)$$

Instead of directly including the non-convex cost term $\text{Cost}_\mathcal{O}$ in the objective, we include an auxiliary variable $d_\text{min}$ in the optimization and reformulate the cost as

$$\text{Cost}_\mathcal{O} = -d_\text{min}, \qquad d_\text{min} \leq \text{sd}(X_t, X_{t+1}, \mathcal{O}_i). \quad (5)$$

The objective (2a) is a weighted sum of the costs above, where $\alpha_\Delta, \alpha_\phi, \alpha_\mathcal{O} \geq 0$ are user-defined, nonnegative coefficients to leverage different costs. A relatively large $\alpha_\mathcal{O}$, for instance, may result in trajectory with larger clearance from obstacles, at the expense of a longer trajectory.

### IV. TRAJECTORY OPTIMIZATION ON $SE(3)$

Sequential convex optimization is a generic approach for solving constrained, non-convex optimization problems of the form $\min_{\boldsymbol{\theta}} f(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the set of optimization variables. One repeatedly constructs a convex optimization problem that locally approximates the original problem around the current solution $\boldsymbol{\theta}^{(i)}$. Then one solves the subproblem

to compute a step $\Delta\boldsymbol{\theta}^{(i)}$ that make progress on the original problem. In sequential quadratic programming (SQP), the convex subproblems are quadratic programs (QP), which are particularly efficient to solve. Two necessary ingredients in sequential convex optimization algorithms are trust regions and merit functions. A trust region constrains $\boldsymbol{\theta}^{(i)}$ in each subproblem to the region where the approximation is valid. A merit function has the form $f_\mu(\boldsymbol{\theta}) = f(\boldsymbol{\theta}) + \mu \cdot$ ConstraintViolation$(\boldsymbol{\theta})$, and it ensures that the steps taken by the algorithm make progress on both the cost function and the constraints. The optimization algorithm has an outer loop that solves a series of unconstrained minimization problems $\min_{\boldsymbol{\theta}} f_{\mu_0}(\boldsymbol{\theta}), \min_{\boldsymbol{\theta}} f_{\mu_1}(\boldsymbol{\theta}), \ldots, \min_{\boldsymbol{\theta}} f_{\mu_n}(\boldsymbol{\theta})$ for $\mu_0 < \mu_1 < \cdots < \mu_n$. See [20] for general reference on non-convex optimization, and [6] for the application to optimal control.

**Local Coordinate Parameterization**: The optimization problem outlined in Eq. (2) is, however, described directly over the set of poses $\mathcal{X}$. One could use a global parameterization of the rotation group, such as axis-angle coordinates or Euler angles. The drawback of those parameterizations is that they distort the geometry—for example, consider how a map of the world is distorted around the poles. This distortion can severely slow down an optimization algorithm, by reducing the neighborhood where local (first and second-order) approximations are good.

In this work, we generalize sequential convex optimization to the case where the domain is a differentiable manifold such as the $SE(3)$ Lie group rather than $\mathbb{R}^n$ by considering a local coordinate parameterization of the manifold [25]. This parameterization is given by the Lie algebra $\mathfrak{se}(3)$, which is defined as the tangent vector space at the identity of $SE(3)$. We refer the reader to the appendix for additional details.

In this work, we construct and solve each convex subproblem in terms of the increments to the previous solution. At the $i^{\text{th}}$ iteration of SQP, let $\bar{\mathcal{X}}^{(i)} = \{\bar{\mathbf{x}}_0^{(i)}, \ldots, \bar{\mathbf{x}}_T^{(i)}\}$ be the sequence of incremental twists (step) computed by solving the convex subproblem. Given a trajectory consisting of a sequence of nominal poses $\hat{\mathcal{X}}^{(i)} = \{\hat{X}_0^{(i)}, \ldots, \hat{X}_T^{(i)}\}$, the subsequent sequence of poses is obtained by applying $\bar{\mathcal{X}}^{(i)}$ as $\hat{\mathcal{X}}^{(i+1)} = \{\exp(\bar{\mathbf{x}}_0^{(i)\wedge}) \cdot \hat{X}_0^{(i)}, \ldots, \exp(\bar{\mathbf{x}}_T^{(i)\wedge}) \cdot \hat{X}_T^{(i)}\}$.

**Convexification**: For trajectory optimization problems, there are two ways to construct locally convex approximations of the costs and constraints for setting up the convex subproblem. One can either convexify the costs and constraints directly around the current solution $\hat{\mathcal{X}}^{(i)}$, which might correspond to an infeasible trajectory that does not satisfy the kinematic constraints (Eq. (1)). Alternatively, we can forward integrate the computed controls and then construct the convex approximation around the integrated trajectory, which is guaranteed to satisfy all kinematic constraints, but the trajectory might violate the constraints on the entry zone and target zone. It is easier to satisfy constraints on the start and target zones without forward integration but the differential curvature constraint is difficult to satisfy. In Sec. V, we present a detailed comparison of these two methods.

**Multi-Trajectory Optimization**: In this work, we also consider the problem of computing multiple curvature-

---

**Algorithm 1** Multi-Trajectory Optimization

**Parameters:**
    $n_{\text{traj}}$: number of trajectories
    $\mathcal{Z}_i$: problem instance for $i^{\text{th}}$ trajectory
    $\mathcal{X}_i^{(0)}$: initialization for $i^{\text{th}}$ trajectory
    $\epsilon$: perturbation parameter
    $\mathbf{w}$: unit Gaussian samples $\mathbf{w} \sim \mathcal{N}(0, \mathbf{I})$

**Variables:**
    $\mathcal{X}_i^{(k)}$: solution for $i^{\text{th}}$ trajectory in $k^{\text{th}}$ SQP iteration

1: **for** $k = 1, 2, \ldots$ **do**
2:     **for** $i = 1, 2, \ldots, n_{\text{traj}}$ **do**
3:         **if** $k = 1$ or $\mathcal{X}_i^{(k-1)}$ is a feasible solution of $i^{\text{th}}$ trajectory in $(k-1)^{\text{th}}$ iteration **then**
4:             $\mathcal{X}_i^{(k)} \leftarrow$ solve $\mathcal{Z}_i$ with initialization $\mathcal{X}_i^{(k-1)}$
5:         **else**
6:             $\mathcal{X}_i^{(k)} \leftarrow$ solve $\mathcal{Z}_i$ with initialization $\mathcal{X}_i^{(k-1)} + \epsilon\mathbf{w}$
7:         **end if**
8:     **end for**
9:     **if** all $\mathcal{X}_i^{(k)}$ are converged solutions in iteration $k$ **then**
10:         **break**
11:     **end if**
12: **end for**

---

constrained trajectories that are mutually collision-free. The complexity of solving $n_{\text{traj}}$ trajectories simultaneously while avoiding collisions between trajectories increases rapidly. Although the size of the optimization vector grows linearly, the number of collision constraints between trajectories grows quadratically in $n_{\text{traj}}$. In addition, the chances of getting stuck in an infeasible local optima becomes much higher as $n_{\text{traj}}$ increases. A natural extension is to solve for each trajectory sequentially in a predefined order while avoiding collisions with previously computed trajectories. However this approach may result in conflicts where trajectories that are computed first may collide with the target zone of trajectories that need to be solved for later.

We propose a novel multi-trajectory optimization procedure (Alg. 1), which is built on top of the single trajectory optimization described above. In this procedure, we repeatedly compute each trajectory individually, where the optimization is initialized by a perturbed version of the previous solution. The previously computed trajectories are added as static obstacles to the environment since the objective is to compute trajectories that are mutually collision-free. Randomly perturbing the solution from previous optimization runs also has the desirable side effect of perturbing the optimization to potentially finding better local optima.

## V. EXPERIMENTAL RESULTS

We experimentally evaluated our approach in two real-world applications involving medical needle steering and designing channel layouts for intracavitary brachytherapy. We implemented our algorithm in C++ and ran all the experiments on a machine with a Intel i7 3.5 GHz CPU.

## A. Medical Needle Steering

We used an anatomical model of the human male pelvic region to simulate needle insertion in tissue for delivering radioactive doses to targets within the prostate. We considered randomly sampled targets within the prostate for our experiments. We set the entry zone to be a $0.1\,\text{cm}$ $\times$ $5\,\text{cm}$ $\times$ $2.5\,\text{cm}$ region on the perineum (skin) through which needles are typically inserted for needle-based prostate procedures. The target zones were modeled as spheres around the target points with radius $0.25\,\text{cm}$, within the range of average placement errors ($\approx 0.63\,\text{cm}$) encountered during procedures performed by experienced clinicians [28]. The average distance between the entry zone and the target zone is $10\,\text{cm}$ and and we set $\kappa_{\max} = 0.125\text{cm}^{-1}$. We used $T = 10$ time steps for our experiments, such that the step length was roughly $1\text{cm}$. For the objective function, we used $\alpha_\Delta = \alpha_\phi = 1$, and we compared the planned trajectory with different choices of the clearance coefficient $\alpha_\mathcal{O}$.

We compared the effect forward integration on the entire trajectory for constructing the underlying convex subproblems. We also compared the performance of our optimization-based approach with a sampling-based rapidly-exploring random tree (RRT) planner proposed by Xu et al. [31] for computing constant curvature trajectories for the needle. The planner was modified to plan backwards starting from target zones because it is easier to compute feasible constant curvature trajectories.

**Planning for a Single needle**: We first analyzed the planned trajectory for single needle insertion using 400 sampled points in the prostate. In addition to the setup above, we require that the needle insertion axis is at a deviation of at most $5°$ from the horizontal, which is a restriction usually imposed by needle steering hardware that constrains the needle to be horizontal. We do not constrain the orientation of the needle tip at the target. We enforced a safety distance $d_{\text{safe}} = 0.25\,\text{cm}$ between the trajectory and obstacles. The error correction term for rotations (Sec. III-B) is computed to be $d_{\text{arc}} = 0.001\text{cm}$, which is ignored considering the scale of the environment we are planning in (of the order of cm). We compared the planned trajectory with $\alpha_\mathcal{O} = 1$ or $\alpha_\mathcal{O} = 10$, examples of which are shown in Figs. 3(a) and 3(b). Using a larger clearance coefficient results in trajectories farther away from obstacles, at the expense of slightly longer paths.

For each task, we repeatedly ran the optimization initialized by a perturbed solution of the previous run, and we allowed up to 5 reruns. We evaluated the performance of no forward integration versus forward integration in terms of the average running time and percentage of solved problems, as well as the metrics described in Sec. III for the converged solutions. From the statistics listed in Fig. 4, we can see that forward integration outperforms no forward integration in terms of percentage of solved problems and running times. It is worth noting that the optimization solves a larger percentage of problems with $\alpha_\mathcal{O} = 10$ as compared to using $\alpha_\mathcal{O} = 1$ because in the latter case, the optimization finds it difficult to simultaneously satisfy both the kinematics
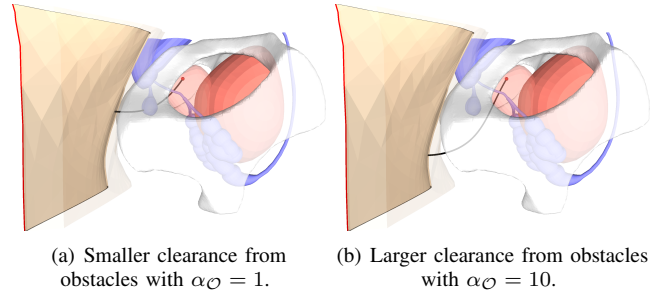


(a) Smaller clearance from obstacles with $\alpha_\mathcal{O} = 1$.

(b) Larger clearance from obstacles with $\alpha_\mathcal{O} = 10$.

Fig. 3. Changing the value of the parameter $\alpha_\mathcal{O}$ influences the clearance of the trajectory from obstacles (Cowper's glands) in the environment. The smaller the value, the lesser the clearance. This could potentially result in unsafe trajectories, as shown in (a).

constraint (Eq. (2c)) and the collision avoidance constraint (Eq. (2e)) when the trajectory is closer to obstacles and has less free space in the environment for improvement.

Our approach outperforms the RRT planner in terms of the number of problems solved. Here, the RRT planner was allotted 10 seconds to find a solution, pending which it reported that a solution could not be found. The trajectories computed using the RRT planner also have a very high twist cost, which is a result of the randomized nature of the planning algorithm. Since the twist cost is directly correlated with tissue damage, the trajectories computed using our approach are preferable over those computed by a randomized planner.

| | RRT [31] | No forward integration $\alpha_\mathcal{O} = 1$ | Forward integration $\alpha_\mathcal{O} = 1$ | No forward integration $\alpha_\mathcal{O} = 10$ | Forward integration $\alpha_\mathcal{O} = 10$ |
|---|---|---|---|---|---|
| solved% | 67.3% | 76.0% | 80.3% | 79.0% | 89.5% |
| time (s) | $9.8 \pm 8.1$ | $1.8 \pm 1.2$ | $1.6 \pm 1.7$ | $1.9 \pm 1.3$ | $1.8 \pm 1.7$ |
| path length | $11.1 \pm 1.5$ | $11.3 \pm 1.4$ | $11.6 \pm 1.7$ | $11.9 \pm 1.7$ | $13.1 \pm 2.3$ |
| twist cost | $34.9 \pm 10.0$ | $1.4 \pm 1.4$ | $1.0 \pm 1.0$ | $1.6 \pm 1.6$ | $1.0 \pm 1.0$ |
| clearance | $0.5 \pm 0.4$ | $0.7 \pm 0.5$ | $0.5 \pm 0.3$ | $1.3 \pm 0.4$ | $1.2 \pm 0.5$ |

Fig. 4. Performance of our approach on the single needle planning case.

**Simulation**: To evaluate the applicability of our method under noise, we ran 100 simulations for a specific target with increasing noise levels. System uncertainty was modeled by perturbing the incremental twists with additive Gaussian noise. We assume access to the pose of the needle, which in practice can be achieved through, for example, an extended Kalman filter with access to sensor readings from a sensing modality and our kinematic motion model. We re-plan after every time step based on the estimated state. We consider a simulation to be successful if it is both collision free and if it reaches the target zone. To ensure path safety, we choose $\alpha_\mathcal{O} = 10$ for all tasks. We examined the effect of increasing noise level on the success rate, as shown in Fig. 5.

Re-planning after each time step greatly increases the probability of success. No forward integration consistently outperforms forward integration in terms of success rate for all noise levels. Since forward integration integrates the trajectory after each iteration of the optimization, the state at the last time step deviates from the target region, making it difficult to find a feasible solution. We also observed a significant difference in the total running time for the
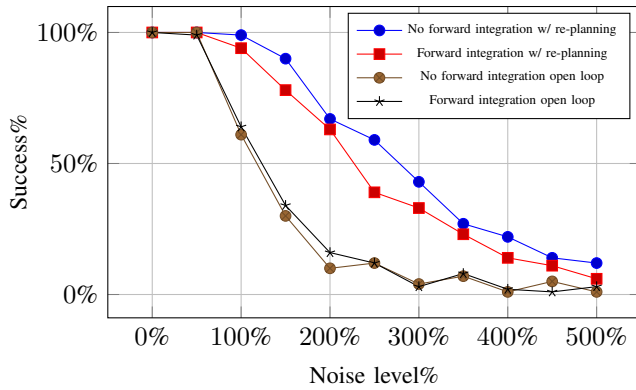
Fig. 5. Effect of noise level on the success rate. The base noise level has standard deviation 0.025cm for the position, and 0.003 units for the rotational components.

simulation, where no forward integration takes 2.3 seconds on average, and forward integration takes 6.7 seconds on average. This suggests that it might be easier to perform local error corrections to an existing trajectory using no forward integration as compared to using forward integration.

**Planning for Multiple needles**: We analyzed the performance of our algorithm planning for 5 needle trajectories using 1000 sampled points within the prostate (200 trials). We compared the result of no forward integration vs forward integration, applying our proposed multi-trajectory planning algorithm, as outlined in Alg. 1. Using forward integration offers an advantage over not using it in terms of computational time required to compute a feasible solution and the quality of trajectories computed. Fig. 1(a) shows planned trajectories for a single trial. Fig. 6 summarizes our result, which shows the advantage of our proposed approach. Our approach outperforms the RRT planner in terms of the number of problems solved. The trajectories computed using the RRT planner have a very high twist cost, which is also undesirable. We also tested planning for multiple trajectories simultaneously, but the running time was too long and the algorithm failed to find a solution for 3 needles or more.

| | RRT | No forward integration | Forward integration |
|---|---|---|---|
| solved% | 48.3% | 74.5% | 78.5% |
| time (s) | $50.0 \pm 19.0$ | $18.0 \pm 9.0$ | $15.3 \pm 15.2$ |
| path length | $54.6 \pm 3.1$ | $53.9 \pm 2.5$ | $56.5 \pm 3.4$ |
| twist cost | $168.3 \pm 28.4$ | $3.8 \pm 1.5$ | $2.5 \pm 1.8$ |
| clearance | $0.1 \pm 0.08$ | $0.1 \pm 0.03$ | $0.1 \pm 0.06$ |

Fig. 6. Performance of our approach for multiple needle planning

### B. Channel Layout Design

We set up a simplified scene for designing the channel layout. We consider a scenario where a 3D printed implant is prepared for treatment of OB/GYN tumors (both vaginal and cervical), as shown in Fig. 1(b).

The implant was modeled as a cylinder of height $7\,cm$ and radius $2.5\,cm$, with a hemisphere on top with radius $2.5cm$. The dimensions of the implant was designed based on dimensions reported by Garg et al. [12]. We placed 3 tumors and picked 8 (oriented) target poses inside the implant. We set the entry region to be the base of the implant, with a deviation angle at most $10°$ to the perpendicular direction.

We require that the curvature along the path is at most $1cm^{-1}$ and that the total curvature on the trajectory (Eq. (2h)) is at most $1.57$. This constraint is important to ensure that catheters carrying the radioactive seed can be pushed through the channels. Instead of planning forward from the entry to the target, we planned backwards from the target to the entry zone using collocation with backward integration, since the entry constraint is much easier to satisfy than the target constraint. Fig. 1(b) shows a channel layout computed using our method.

We compared the performance of our approach with a highly-optimized RRT-based planner [12] proposed for this specific application (Fig. 7). Both the RRT-based approach and our approach have a randomization aspect associated with them – while the RRT uses random sampling, our multi-trajectory planning procedure uses random perturbations to initialize the optimization. We solved the same problem 100 times to investigate the randomized aspect of both approaches. Our approach is able to compute a feasible solution in almost all cases, whereas the RRT algorithm fails more often to find a feasible solution. The RRT planner also computed plans that have a higher cumulative path length and twist cost as compared to the solution computed using our approach, which is undesirable.

| | RRT | Backward integration |
|---|---|---|
| solved% | 74.0% | 98.0% |
| time (s) | $30.8 \pm 17.9$ | $27.7 \pm 9.8$ |
| path length | $41.3 \pm 0.3$ | $38.9 \pm 0.1$ |
| twist cost | $65.5 \pm 8.4$ | $4.1 \pm 1.1$ |

Fig. 7. Performance of our approach for planning channel layouts

## VI. CONCLUSION AND FUTURE WORK

In this work, we presented a method for computing locally optimal, curvature-constrained trajectories in 3D environments using state of the art trajectory optimization methods. We considered two real-world applications involving medical needle steering and designing channel layouts and we demonstrated that our approach offers considerable advantages in computational efficiency and computes high-quality plans as compared to sampling-based approaches. However, our optimization-based method does not guarantee probabilistic completeness offered by sampling-based planners, since the algorithm can only guarantee locally optimal solutions. The adjustment scheme proposed in Alg. 1 partially overcomes this problem, and we defer investigation of theoretical guarantees of this approach to future work.

In future work, we plan to evaluate our approach by conducting needle steering experiments in tissue phantoms and real tissues. The fast computation makes it suitable for online control of bevel-tip steerable needles based on 3D imaging feedback. We also plan to use our approach for designing 3D printed implants for intracavitary brachytherapy.

## REFERENCES

[1] P. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[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)*, 2007.

[3] C. Belta and V. Kumar, "Euclidean metrics for motion generation on SE(3)," *Journal of Mechanical Engineering Science*, vol. 216, no. 1, pp. 47–60, 2002.

[4] G. V. D. Bergen, "Proximity Queries and Penetration Depth Computation on 3D Game Objects," in *Game Developers Conference (GDC)*, 2001.

[5] 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*, pp. 1–39, 2013 (to appear).

[6] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of guidance, control, and dynamics*, vol. 21, no. 2, pp. 193–207, 1998.

[7] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of mathematics*, vol. 79, no. 3, pp. 497–516, 1957.

[8] V. Duindam, R. Alterovitz, S. Sastry, and K. Goldberg, "Screw-based motion planning for bevel-tip flexible needles in 3D environments with obstacles," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2008, pp. 2483–2488.

[9] V. Duindam, J. Xu, R. Alterovitz, S. SastrySastry, and K. Goldberg, "Three-dimensional motion planning algorithms for steerable needles using inverse kinematics," *Int. Journal of Robotics Research*, vol. 29, no. 7, pp. 789–800, 2010.

[10] J. Engh, D. Minhas, D. Kondziolka, and C. Riviere, "Percutaneous intracerebral navigation by duty-cycled spinning of flexible bevel-tipped needles," *Neurosurgery*, vol. 67, no. 4, pp. 1117–1122, 2010.

[11] E. Frazzoli, M. A. Dahleh, and E. Feron, "Maneuver-based motion planning for nonlinear systems with symmetries," *IEEE Trans. on Robotics*, vol. 21, no. 6, pp. 1077–1091, 2005.

[12] A. Garg, S. Patil, T. Siauw, J. A. M. Cunha, I.-C. Hsu, P. Abbeel, J. Pouliot, and K. Goldberg, "An algorithm for computing customized 3d printed implants with curvature constrained channels for enhancing intracavitary brachytherapy radiation delivery," in *IEEE Int. Conf. on Automation Science and Engg. (CASE)*, vol. 4, 2013, pp. 3306–3312.

[13] J.-C. Han, S. Datta, and S. Ekkad, *Gas turbine heat transfer and cooling technology*. CRC Press, 2013.

[14] M. Hwangbo, J. Kuffner, and T. Kanade, "Efficient two-phase 3D motion planning for small fixed-wing UAVs," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2007, pp. 1035–1041.

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

[16] S. M. LaValle, *Planning algorithms*. Cambridge University Press, 2006.

[17] A. Majewicz, J. Siegel, and A. Okamura, "Design and evaluation of duty-cycling steering algorithms for robotically-driven steerable needles," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2014, p. to appear.

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

[19] R. M. Murray and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[20] J. Nocedal and S. Wright, *Numerical Optimization*. Springer Verlag, 2006.

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

[22] S. Patil, J. Burgner, R. J. Webster III, and R. Alterovitz, "Needle steering in 3D via rapid replanning," *IEEE Trans. on Robotics*, p. to appear, 2014.

[23] M. Pivtoraiko, "Differentially constrained motion planning with state lattice motion primitives," Robotics Institute, Carnegie Mellon University, Tech. Rep. CMU-RI-TR-12-07, 2012.

[24] J. Reeds and L. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific Journal of Mathematics*, vol. 145, no. 2, pp. 367–393, 1990.

[25] A. Saccon, J. Hauser, and A. P. Aguiar, "Optimal control on Lie groups: The projection operator approach," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2230–2245, 2013.

[26] J. Schulman, J. Ho, A. Lee, H. Bradlow, I. Awwal, and P. Abbeel, "Finding locally optimal, collision-free trajectories with sequential convex optimization," in *Robotics: Science and Systems (RSS)*, 2013.

[27] M. Shanmugavel, A. Tsourdos, R. Zbikowski, and B. A. White, "3D path planning for multiple UAVs using Pythagorean hodograph curves," in *AIAA Guidance, Navigation, and Control Conference*, 2007, pp. 20–23.

[28] R. Taschereau, J. Pouliot, J. Roy, and D. Tremblay, "Seed misplacement and stabilizing needles in transperineal permanent prostate implants," *Radiotherapy and Oncology*, vol. 55, no. 1, pp. 59–63, 2000.

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

[30] J. Xu, V. Duindam, R. Alterovitz, J. Pouliot, J. A. Cunha, I. Hsu, and K. Goldberg, "Planning fireworks trajectories for steerablemedical needles to reduce patient trauma," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009, pp. 4517–4522.

[31] J. Xu, V. Duindam, R. Alterovitz, and K. Goldberg, "Motion planning for steerable needles in 3D environments with obstacles using rapidly-exploring random trees and backchaining," in *IEEE Int. Conf. on Automation Science and Engg. (CASE)*, 2008, pp. 41–46.

[32] K. Yang and S. Sukkarieh, "An analytical continuous-curvature path-smoothing algorithm," *IEEE Trans. on Robotics*, vol. 26, no. 3, pp. 561–568, 2010.

## VII. APPENDIX

The special Euclidean group $SE(3)$ is a 6D configuration space consisting of the pose (3D position and 3D orientation). The Lie algebra $\mathfrak{se}(3)$ is defined as the tangent vector space at the identity of $SE(3)$. The $SE(3)$ group and $\mathfrak{se}(3)$ algebra are related via the exponential and log maps, $\exp : \mathfrak{se}(3) \to SE(3)$ and $\log : SE(3) \to \mathfrak{se}(3)$, where $\exp$ and $\log$ correspond to the matrix exponential and log operations. In this particular case, closed-form expressions exist for the $\exp$ and $\log$ operators (Appendix A in [19]).

Given a vector $\bar{\mathbf{x}} = \left[\begin{smallmatrix} \bar{\mathbf{p}} \\ \bar{\mathbf{r}} \end{smallmatrix}\right] \in \mathbb{R}^6$ that represents the incremental twist, the corresponding Lie algebra element is given by the mapping $^\wedge : \mathbb{R}^6 \to \mathfrak{se}(3)$ as

$$\bar{\mathbf{x}}^\wedge = \begin{bmatrix} [\bar{\mathbf{r}}] & \bar{\mathbf{p}} \\ \mathbf{0}_3^T & 0 \end{bmatrix},$$

where the notation $[\bar{\mathbf{r}}]$ for the vector $\bar{\mathbf{r}} = [\bar{\mathbf{r}}_x \ \bar{\mathbf{r}}_y \ \bar{\mathbf{r}}_z]^T \in \mathbb{R}^3$ is the $3 \times 3$ skew-symmetric matrix given by

$$[\bar{\mathbf{r}}] = \begin{bmatrix} 0 & -\bar{\mathbf{r}}_z & \bar{\mathbf{r}}_y \\ \bar{\mathbf{r}}_z & 0 & -\bar{\mathbf{r}}_x \\ -\bar{\mathbf{r}}_y & \bar{\mathbf{r}}_x & 0 \end{bmatrix}.$$

Intuitively, $\bar{\mathbf{r}}$ represents the incremental rotation and $\bar{\mathbf{p}}$ represents the incremental translation to be applied to a nominal pose. The inverse is defined by the operator $^\vee : \mathfrak{se}(3) \to \mathbb{R}^6$ to recover $\bar{\mathbf{x}}$ given a Lie algebra element, i.e., $\begin{bmatrix} [\bar{\mathbf{r}}] & \bar{\mathbf{p}} \\ \mathbf{0}_3^T & 0 \end{bmatrix}^\vee = \bar{\mathbf{x}}$. The local neighborhood $X$ of a nominal pose $\hat{X} \in SE(3)$ is defined in terms of $\bar{\mathbf{x}} \in \mathbb{R}^6$ as

$$X = \exp(\bar{\mathbf{x}}^\wedge) \cdot \hat{X}.$$

Conversely, given a nominal pose $\hat{X}$ and a pose $X$, the corresponding twist $\bar{\mathbf{x}} \in \mathbb{R}^6$ can be recovered as:

$$\bar{\mathbf{x}} = \log(X \cdot \hat{X}^{-1})^\vee.$$