
3D Motion Planning Algorithms for Steerable Needles Using Inverse Kinematics

Vincent Duindam¹, Jijie Xu², Ron Alterovitz^{1,3}, Shankar Sastry¹, and Ken Goldberg^{1,2}

¹ Department of EECS, University of California, Berkeley,
{vincentd,sastry}@eecs.berkeley.edu

² Department of IEOE, University of California, Berkeley,
{jijie.xu,goldberg}@berkeley.edu

³ Comprehensive Cancer Center, University of California, San Francisco,
ronalt@berkeley.edu

Abstract: Steerable needles can be used in medical applications to reach targets behind sensitive or impenetrable areas. The kinematics of a steerable needle are non-holonomic and, in 2D, equivalent to a Dubins car with constant radius of curvature. In 3D, the needle can be interpreted as an airplane with constant speed and pitch rate, zero yaw, and controllable roll angle.

We present a constant-time motion planning algorithm for steerable needles based on explicit geometric inverse kinematics similar to the classic Paden-Kahan subproblems. Reachability and path competitiveness are analyzed using analytic comparisons with shortest path solutions for the Dubins car (for 2D) and numerical simulations (for 3D). We also present an algorithm for local path adaptation using null-space results from redundant manipulator theory. The inverse kinematics algorithm can be used as a fast local planner for global motion planning in environments with obstacles, either fully autonomously or in a computer-assisted setting.

1 Introduction

Steerable needles [18] form a subclass of flexible needles that provide steerability due to asymmetric forces acting at the needle tip, for example due to a beveled surface [18] or a kink near the end of the needle [7]. By rotating the needle at the base, the orientation of the tip can be changed and hence the trajectory of the needle can be controlled. Steerable needles differ in this respect from symmetric flexible needles, which can only be controlled by applying asymmetric forces at the base [4], not at the tip. The extra mobility of steerable needles over rigid needles can be harnessed in medical applications such as brachytherapy [2] and brain surgery [7] to reach difficult targets behind sensitive or impenetrable areas.

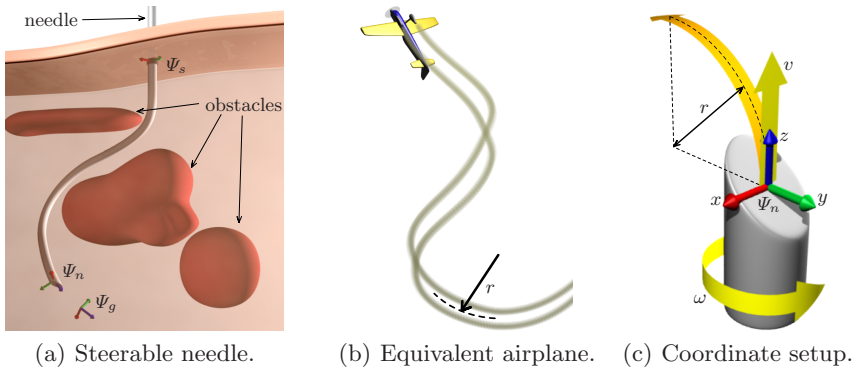


Fig. 1. Model setup of a steerable needle and a kinematically equivalent airplane with fixed speed and pitch rate, zero yaw, and controllable roll rate.

Experimental studies [17] show that the motion of steerable needles can be approximated as having a constant radius of curvature that is independent of insertion speed. The control inputs for the needle are the insertion speed and rotation (roll) angle, although for motion planning (the topic of this paper) insertion speed is often not important. The rotation angle is then the only real control input and trajectories can be parameterized by insertion depth. A steerable needle is thus kinematically equivalent to an airplane with fixed speed and pitch rate, zero yaw, and controllable roll rate (Fig. 1b).

Motion planning for steerable needles is an important problem and has been studied in several ways in literature. Most studies focus on planar motion, for which the control input reduces to switching between curve-left and curve-right. Alterovitz et al. [2, 1, 3] present a roadmap-based motion planning framework that explicitly incorporates motion uncertainty and computes the path that is most likely to succeed. Minhas et al. [12] show planning based on fast duty cycle spinning of the needle, effectively removing the limitation of a fixed-radius path but requiring continuous angular control input. Kalleem et al. [10] introduce a controller that stabilizes the needle motion to a plane, allowing practical implementation of planar motion planning methods.

The first 3D motion planning algorithm was introduced by Park et al. [14, 15] and used diffusion of a stochastic differential equation to generate a family of solution paths. The authors also describe several extensions to avoid obstacles. Duindam et al. [6] presented a second 3D motion planning algorithm that uses fast numerical optimization of a cost-function to compute feasible needle paths in 3D environments with obstacles.

This paper presents a different solution to the 3D motion planning problem for steerable needles, based on inverse kinematics. We propose a new geometry-based algorithm inspired by the Paden-Kahan subproblems in traditional inverse kinematics algorithms [13]. Just as the Paden-Kahan subproblems, our algorithm (Section 3) can be fully described in geometric terms

of intersecting lines, planes, and circles, and computing the solution simply requires a few trigonometric functions. We analyze reachability and competitiveness of the solution (Section 4) using analytic comparison to the Dubins car solution and numerical simulations. We also present a method to locally adapt needle paths using null-motions (Section 5).

The proposed motion planning algorithms find feasible needle paths very quickly and provide some capability for obstacle avoidance. As the current accuracy of medical data is limited, these algorithms may be sufficient by themselves, but we envision them being part of more general global motion planning systems, as discussed in Section 6.

2 Problem statement and modeling assumptions

2.1 Model parameters and assumptions

Throughout this paper, we only consider the idealized kinematics of the needle in a static environment. We assume that the motion of the needle is fully determined by the motion of the needle tip, that the motion of the needle tip is instantaneously along a perfect circle of constant radius r , and that rotations of the base are instantly transmitted to rotations of the tip. Experimental results [17] show that needle materials can be chosen such that the needle indeed moves along an arc of approximately constant radius, but the effects of tissue inhomogeneity, friction, and needle torsion can be significant and will require compensation [10] in practical applications.

Under these assumptions, the motion of the needle is determined kinematically by two control inputs: the insertion velocity, denoted v , and the tip rotation velocity, denoted ω . We present the kinematics model for general $v(t)$ but remove this redundant degree of freedom in the next section.

Fig. 1c illustrates the model setup. We rigidly attach a coordinate frame Ψ_n to the tip of the needle, with axes aligned as in the figure, such that the z -axis is the direction of forward motion v and needle orientation ω , and the beveled tip causes the needle to rotate instantaneously around the line parallel to the x -axis and passing through the point $(0, -r, 0)$.

Following standard robotics literature [13], the position and orientation of the needle tip relative to a reference frame Ψ_s can be described compactly by a 4×4 matrix $g_{sn}(t) \in SE(3)$ of the form

$$g_{sn}(t) = \begin{bmatrix} R_{sn}(t) & p_{sn}(t) \\ 0 & 1 \end{bmatrix} \quad (1)$$

with $R_{sn} \in SO(3)$ the rotation matrix describing the relative orientation, and $p_{sn} \in T(3)$ the vector describing the relative position of frames Ψ_s and Ψ_n .

The instantaneous linear and angular velocities of the needle are described by a twist $V_{sn} \in se(3)$ which in body coordinates Ψ_n takes the convenient form

$$V_{sn}^n(t) = \begin{bmatrix} 0 \\ 0 \\ v(t) \\ v(t)/r \\ 0 \\ \omega(t) \end{bmatrix} \quad \hat{V}_{sn}^n(t) = \begin{bmatrix} 0 & -\omega(t) & 0 & 0 \\ \omega(t) & 0 & -v(t)/r & 0 \\ 0 & v(t)/r & 0 & v(t) \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2)$$

in equivalent vector and matrix notation. The twist relates to g_{sn} as

$$\dot{g}_{sn}(t) = g_{sn}(t)\hat{V}_{sn}^n(t) \quad (3)$$

This kinematic model is the same as the unicycle model by Webster et al. [16]. When the twist is constant, (3) becomes a linear ordinary differential equation (ODE) that can be integrated as

$$g_{sn}(t) = g_{sn}(0) \exp(t\hat{V}_{sn}^n) \quad (4)$$

for which a relatively simple analytic expression exists [13]. In the path planning algorithms, we construct paths for which \hat{V}_{sn}^n is piecewise constant and compute the resulting transformation using this efficient analytic expression.

2.2 Problem statement

The objective of the motion planning algorithms in this paper is to find feasible paths between given start and goal configurations in the absence of obstacles¹. More precisely, the inputs to the algorithm are an initial needle pose $g_{start} \in SE(3)$ and a desired needle pose $g_{goal} \in SE(3)$. The outputs of the algorithm are control functions $v(\cdot)$ and $\omega(\cdot)$ and a finite end time $0 \leq T < \infty$, such that the solution $g_{sn}(\cdot)$ of the differential equation (3) with $g_{sn}(0) = g_{start}$ satisfies $g_{sn}(T) = g_{goal}$. If no feasible path can be found, the algorithm returns failure.

The kinematics equations (2) and (3) are invariant to time scaling, in the sense that the path traced out by the needle does not change if the control inputs $v(t)$ and $\omega(t)$ are scaled by the same (possibly time-varying) factor. Therefore, we can simplify the motion planning problem by assuming without loss of generality that $v(t) \equiv 1$, which is equivalent to parameterization by insertion depth [10, 6]. The insertion time T thus represents the total path length since $\int_0^T |v(t)| dt = \int_0^T dt = T$.

Although motion planning is based on connecting general 3D poses (full position and orientation), solving a difference in initial or final roll angle is trivial as this degree of freedom is directly controlled through $\omega(\cdot)$. So although the inputs to the algorithm are general elements of $SE(3)$, we often mainly focus on path planning between given start and goal position and *direction* of the needle, i.e. only considering the z -axis of Ψ_n . Given a solution from the start position and direction to the goal position and direction, the full motion plan from a start pose to a goal pose follows directly by adding the required roll-rotations to the beginning and end of this solution.

¹ Section 5 discusses a way to avoid obstacles using path adaptation.

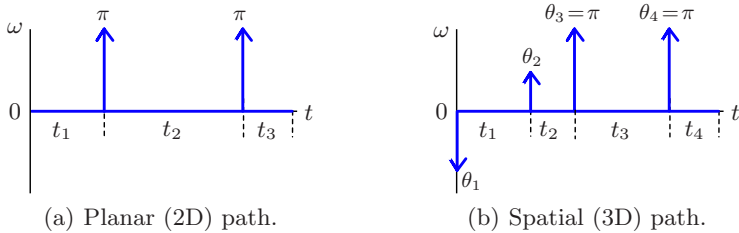


Fig. 2. Structure of the solution $\omega(\cdot)$ for the 2D and 3D motion planning problems.

3 Path planning using inverse kinematics

We present two motion planning solutions based on inverse kinematics, one for the planar (2D) case and one for the general spatial (3D) case. The motion planning problem is considered planar if the start position p_s , start direction z_s , goal position p_g , and goal direction z_g are all in the same plane.

In both inverse kinematics solutions, we look for a control input function $\omega(\cdot)$ of a very specific form, namely a function that is zero everywhere except for a fixed number of Dirac impulses (two in the planar case, four in the spatial case, see Fig. 2). Geometrically, this means we look for trajectories that are concatenations of a fixed number of circular segments with radius r : the needle moves along a circle when $\omega = 0$, and instantaneously changes direction at the time instants that ω is a Dirac impulse. Furthermore, the magnitudes of the Dirac impulses in the planar case are constrained to be exactly π , corresponding to a change in direction between curve-left and curve-right. We also choose a spatial solution for which the last two impulsive rotations are π , making the last three segments of the spatial path co-planar as well.

The specific choices in the structure of $\omega(\cdot)$ result in geometrically intuitive solutions that are relatively straight-forward to compute. The simplicity of the proposed solutions comes at the cost of not necessarily being optimal in terms of path length or required control effort. Practical implementations should clearly not use impulsive rotational control and constant insertion speed, but alternate between pure insertion until the desired depth t_i is reached, and pure rotation to the desired angle θ_i .

3.1 Inverse kinematics in 2D

We first consider the planar path planning problem with $\omega(\cdot)$ as in Fig. 2a. The relative position of the start and goal are described by two displacements x and y , their relative orientation by a single angle θ . The purpose of the path planning algorithm is to find the three insertion depths t_1, t_2, t_3 describing a feasible needle path from start to goal. Note that the needle travels a distance $t_i = r\alpha_i$ when moving along a circle of radius r for α_i radians, and hence we can equivalently look for the three angles α_i in Fig. 3. These should be

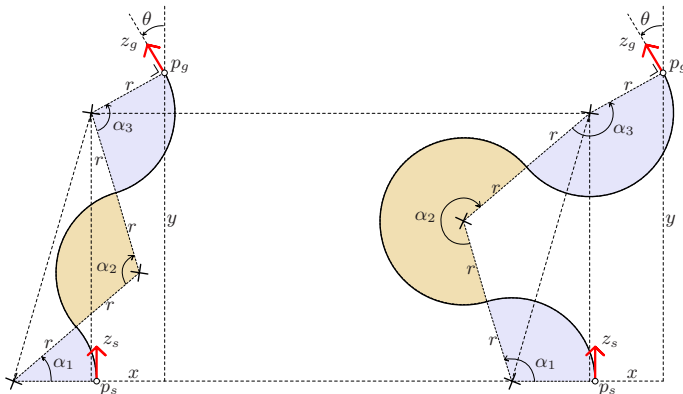


Fig. 3. Two geometric solutions for the same planar inverse kinematics problem, both using sequential bevel-left, bevel-right, bevel-left motions.

such that if we start at p_s in the direction z_s heading left, move $r\alpha_1$ forward, turn π , move $r\alpha_2$ forward, turn π , and move $r\alpha_3$ forward, we arrive exactly at the desired goal pose. The mirrored case starting with a right turn can be computed similarly.

We can solve for the angles α_i by looking at the setup in Fig. 3 and realizing that the three centers of rotation (marked by \times in the figure) form a triangle with known edge lengths. Using the cosine rule for this triangle, we can write

$$\cos(\alpha_2) = 1 - \frac{(x + r - r \cos(\theta))^2 + (y - r \sin(\theta))^2}{8r^2} \quad (5)$$

This equation has two solutions for α_2 , which correspond to the two paths shown in Fig. 3. With α_2 known, the other two angles follow uniquely as

$$\alpha_1 = \text{atan2}(y - r \sin(\theta), x + r - r \cos(\theta)) - \frac{1}{2}(\pi - \alpha_2) \quad (6)$$

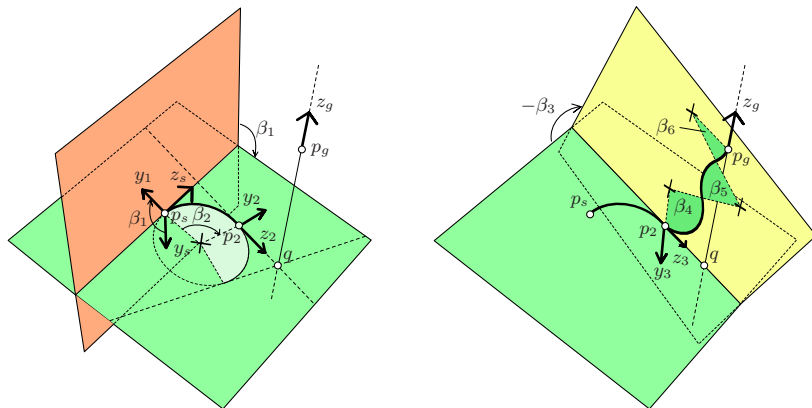
$$\alpha_3 = \theta - \alpha_1 + \alpha_2 \quad (7)$$

with atan2 the inverse tangent function solved over all quadrants. Since the needle can only move forward, angles must be chosen as $\alpha_i \in [0, 2\pi)$. The required insertion distances t_i follow immediately as $t_i = r\alpha_i$.

3.2 Inverse kinematics in 3D

Now consider the 3D inverse kinematics problem of connecting two general poses in $SE(3)$ by a valid needle path. We propose one solution using eight consecutive insert and turn motions as shown in Fig. 2b; an explicit geometric solution using fewer motions is still an open problem.

The geometry of this solution is illustrated in Fig. 4. The problem is split into two parts: first, the needle is turned and inserted such that its instantaneous line of motion (the instantaneous direction of the needle) intersects the



(a) Rotate β_1 around z_s until the needle's (y, z) -plane contains q . Insert $r\beta_2$ until q is on the line through z_2 .

(b) Rotate β_3 around z_2 until the needle's (y, z) -plane contains p_g . Solve the remaining planar problem.

Fig. 4. Geometric derivation of an inverse kinematic solution on $SE(3)$.

line describing the goal position and direction. Second, the remaining planar problem is solved using the solution from Section 3.1.

More precisely, we first choose any point q on the line defined by p_g and z_g . This point q will be the intersection point of the two lines defining the remaining planar problem. With q defined, the needle is first rotated by $\theta_1 = \beta_1$ until its (y, z) -plane contains q . The required angle β_1 satisfies

$$\tan(\beta_1) = -\frac{x_s^T(q - p_s)}{y_s^T(q - p_s)} \quad (8)$$

which has two solutions β_1 that differ by π .

Second, the insertion distance $t_1 = r\beta_2$ is solved such that the line through the needle tip in the direction z_2 contains the point q . If q is outside the circle describing the needle motion along β_2 , two solutions exist, with z_2 either pointing towards (as in the figure) or away from q . These solutions are

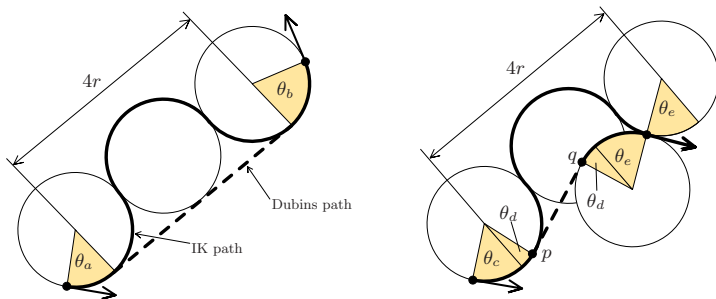
$$\beta_2 = \text{atan2}(z_s^T q_v, y_1^T q_v) \pm \arccos\left(\frac{r}{|q_v|}\right) \quad (9)$$

with $q_v := q - p_s + ry_1$. No solution exists if q is inside the circle ($|q_v| < r$).

Third, the needle is rotated by $\theta_2 = \beta_3$ until p_g (and hence the whole line through q and p_g) is contained in the needle's (y, z) -plane:

$$\tan(\beta_3) = -\frac{x_2^T(p_g - p_2)}{y_2^T(p_g - p_2)} \quad (10)$$

which again gives two solutions that differ by π . The remaining angles $\beta_4, \beta_5, \beta_6$ (corresponding to the time segments t_2, t_3, t_4) can then be solved using the 2D planner from Section 3.1.



(a) Left-straight-left Dubins path. (b) Left-straight-right Dubins path.

Fig. 5. Paths generated by the 2D IK algorithm vs. optimal paths for a Dubins car.

In this algorithm, we have the following degrees of freedom to choose a solution. First is the choice of the point q : this can be anywhere on the line containing p_g and z_g , which means varying q generates a one-dimensional subspace of possible inverse kinematics solutions. Second, we can choose one of two possible solutions β_i for each of the four angles in equations (8–10) and (5), resulting in 2^4 possible combinations. Not all of these choices may give feasible paths for a given start and end pose, and it is also not directly obvious which choice will result in the ‘best’ path between the two poses. Nevertheless, since the inverse kinematics equations can be computed very quickly, one can simply compute all combinations for a number of choices of q and pick the best solution, with ‘best’ defined for example using a cost function [6].

4 Reachability and competitiveness

To evaluate the quality of the paths generated by the presented inverse kinematics (IK) solutions, we study the set of reachable needle poses and competitiveness [9, 8] of the computed solutions. Competitiveness in this case refers to the path length of the computed solution; it has no relation to competitiveness in the sense of computational speed (the IK algorithm runs in constant time).

4.1 Reachability and competitiveness in 2D

Consider first the solution to the planar problem as described in Section 3.1. The algorithm will clearly only find a solution if the right-hand side of (5) has norm less than or equal to one, or geometrically, if the centers of the circles tangent to the start and goal poses are no farther than $4r$ apart. This condition defines the set of reachable relative needle poses.

To describe competitiveness of the algorithm for these reachable poses, we compare the IK solutions to the trajectories generated by allowing an infinite number of direction changes. In that case, a trajectory can be generated with

an arbitrary radius of curvature larger than or equal to r , by asymmetrically cycling between heading-left and heading-right and taking the limit of this cycling frequency to infinity. This means that in the limit, the needle can behave like a Dubins car [5] with minimum radius of curvature r .

The optimal path for a Dubins car is known to consist of two circular arcs with radius r connected by another circular arc or a straight line [5]. For a given start and end pose, the IK solution only differs from the Dubins path if the connecting segment for the Dubins path is a straight line; the IK solution will still contain a (sub-optimal) circular arc. Furthermore, the Dubins path may start and end with circular arcs in the same direction or in opposite directions (Fig. 5), whereas the IK solution always starts and ends with a turn in the same direction. It is intuitively clear that the largest difference in path length occurs at the border of the reachable space where the three circles of the IK solution are aligned. In the first case (Fig. 5a), the maximum ratio of the path lengths is

$$\sup \frac{\|\text{IK path}\|}{\|\text{Dubins path}\|} = \sup_{\theta_i \geq 0} \frac{r\theta_a + r\theta_b + 2\pi r}{r\theta_a + r\theta_b + 4r} = \frac{\pi}{2} \quad (11)$$

For the second case (Fig. 5b), we can compute the length of the straight-line segment $q - p$ as

$$\|q - p\|^2 = 4r^2 (2 - \sin(\theta_d) - \sin(\theta_e))^2 + 4r^2 (\cos(\theta_d) - \cos(\theta_e))^2 \quad (12)$$

from which we find that the maximum path length ratio equals

$$\sup \frac{\|\text{IK path}\|}{\|\text{Dubins path}\|} = \sup_{\theta_i \geq 0} \frac{r\theta_c + 2\pi r - r\theta_e}{r\theta_c + 2r\theta_d + r\theta_e + \|q - p\|} \approx 1.63 > \frac{\pi}{2} \quad (13)$$

The degree of competitiveness of the 2D IK solution is hence approximately 1.63. Note that this is a bound on the competitiveness that does not take into account the number of direction changes; for medical applications, this number should be kept small to avoid excessive tissue damage.

4.2 Reachability and competitiveness in 3D

Continuing with the 3D IK solution from Section 3.2, we present a reachability and competitiveness analysis based on numerical simulation. Formal geometric proofs and bounds of the algorithm are subject of future research; at this point we do not have a good approximation for the optimal path and simply compare the IK solutions to the Euclidean distance between the start and goal positions. Future work could compare the presented solution to the paths generated by the method of Park et al. [15]. We take $q = p_g$ throughout the analysis; different choices give qualitatively similar results.

First consider Fig. 6. This illustrates the lengths of the IK trajectories starting at the center of the figure and ending at goal positions in the plane

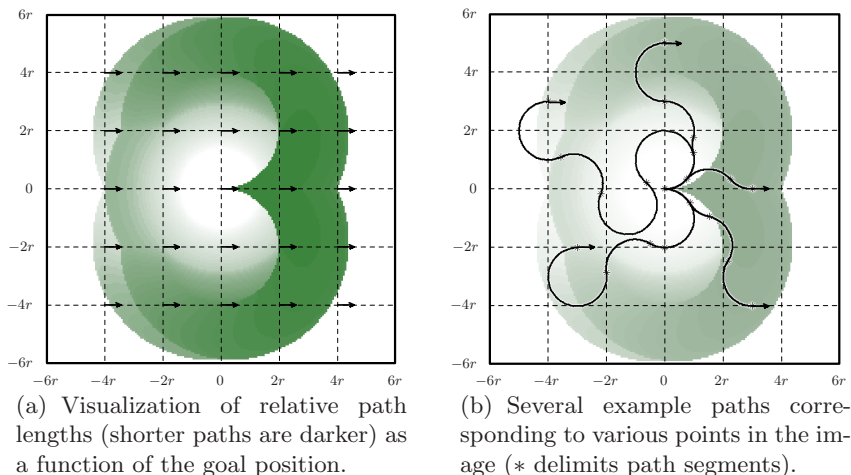


Fig. 6. Reachability and relative path lengths obtained using the inverse kinematics algorithm. The algorithm tries to find a path from the center of the image to each pixel in the image, with both start and goal direction aimed to the right.

of the figure, with start and goal directions equal and pointing to the right. The brightness of each pixel indicates the length of the IK path divided by the Euclidean distance between the start and goal locations: dark colors represent small ratios (good paths) while light colors represent large ratios (bad paths). Fig. 6b illustrates several examples of solution trajectories.

A set of reachable states shows up in the figure as the ‘figure-eight’ around the start location. Points outside this shape cannot be reached for the given goal direction. The figure also shows a distinction between poses that can be reached with reasonably short curves (darker region) and poses that require significantly longer paths (lighter region). This distinction is sharp in the area in front of the needle (right side of the figure) but is more diffuse for poses on the sides of the needle (top and bottom of the figure). If we consider competitiveness in an informal way, meaning whether the algorithm can generate paths of reasonable lengths, we can say that the algorithm is competitive in the darker region of the figure; relative poses that are in the lighter region may be reachable, but the paths are so unwieldy that they are of little practical use in our brachytherapy application.

Fig. 7 shows additional plots generated by varying the two remaining degrees of freedom in placing the goal pose: the in-plane yaw angle and out-of-plane pitch angle (the inverse kinematics solution is invariant to roll about the initial and final needle directions).

The figure shows that as the goal direction is turned away from the straight-ahead case (change in yaw), the set of reachable and competitive paths rotates in the same direction while maintaining a roughly similar shape. As the goal direction is rotated out of plane (change in pitch), the compet-

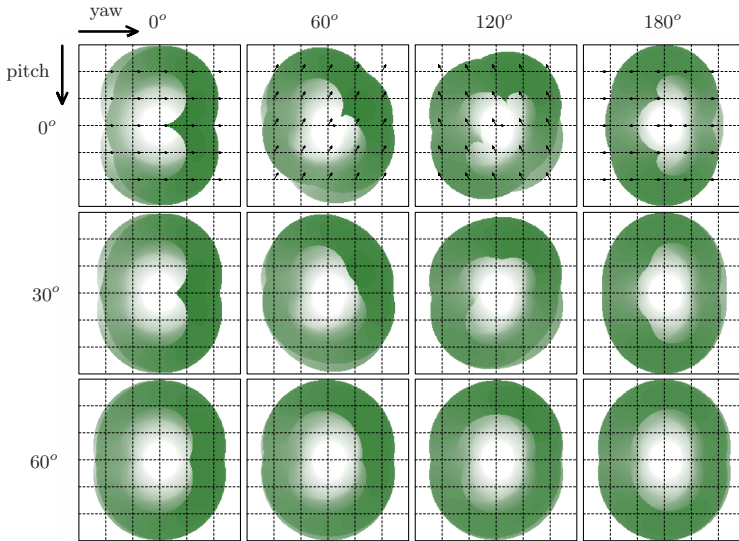


Fig. 7. Reachability and relative path lengths as in Fig. 6 for various relative yaw and pitch angles for the goal pose. Grid lines are $2r$ units apart.

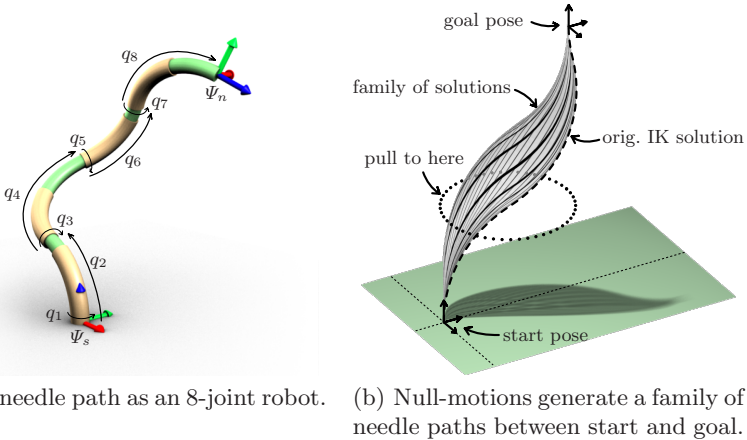
itive paths near the starting pose disappear until only poses at a significant distance (three to five times the radius of curvature) are competitive.

In the three-dimensional case, it remains a difficult task to precisely characterize the set of poses that can be reached with a reasonably short path. Comparing the solutions to the Euclidean distance provides some insight but no global bound on the solutions: competitiveness measures are unbounded when comparing to the Euclidean distance, since for infinitely close but non-collinear needle orientations the path length of the IK solution remains finite.

To use the inverse kinematics planner as a local planner in global roadmap-based planning methods such as the Probabilistic Roadmap method [11], it is important to characterize the set of goal poses that are reachable from a given needle pose. Numerical simulations such as those shown in Fig. 7 can be used to construct an approximation of the set of goal poses that can be reached with a competitive path (with ‘competitive’ in the informal meaning of ‘reasonably short path’). This set can be used as the definition of ‘neighborhood’, i.e. those needle poses that are likely to be connectible and can become edges in the roadmap if they do not intersect obstacles.

5 Path adaptation using null-motions

The presented inverse kinematics (IK) solution can be computed very quickly but is generally not the optimal solution in the sense of avoiding obstacles or minimizing path length or required control effort. Earlier work considered



(a) A needle path as an 8-joint robot. (b) Null-motions generate a family of needle paths between start and goal.

Fig. 8. Representation of a needle path as a robot with eight joints, and use of its null motions to generate a family of solution curves by pulling in different directions.

path planning as a pure numerical optimization problem [6], but in this section we show how sub-optimal paths such as those generated by the IK planner can be locally optimized and adjusted using null-motions.

Consider an IK solution between two general 3D needle poses. By construction, this solution describes a path from start to goal consisting of eight consecutive turning and insertion control actions. We can think of this path as a redundant serial robot manipulator arm with eight joints (Fig. 8a). Since the relative pose of the goal is given by six parameters, standard robotics theory [13] tells us that the robot has a two-dimensional space of null-motions, provided it is not at a singularity. If the joints are moved in this null-motion space, the shape of the robot (i.e. the shape of the needle path) will change without changing the pose of the end effector (i.e. the needle tip).

The set of null motions is described by the null space of the geometric Jacobian $J(q) \in \mathbb{R}^{6 \times 8}$ of the robot, which relates the spatial twist V_{sn}^s to the joint velocities \dot{q} as $V_{sn}^s = J(q)\dot{q}$ [13]. Given the Jacobian, we can change the shape of the path by changing the joint angles in such a way that $\dot{q} \in \text{Null}(J(q))$ at all times.

Fig. 8b shows an example of how one inverse kinematics solution can be locally transformed in this way into a family of solution curves. Intuitively, this set of curves was generated by starting from the IK solution indicated in the figure (the dashed line), and ‘pulling’ on the curve from several points laid out in a circle, while holding the start and end pose of the needle fixed. More precisely, we model the robot as a viscous system with damping in each joint that counteracts applied forces, and write the governing equations as

$$\dot{q} = B(q)J_i^T(q)F_i + B(q)J^T(q)\lambda \quad (14)$$

$$0 = J(q)\dot{q} \quad (15)$$

with F_i the wrench [13] corresponding to the (known) externally applied pulling force, $J_i(q)$ the Jacobian of link i at which F_i is applied, λ the required constraint wrench acting at the tip to constrain its motion, and $B(q) > 0$ the symmetric positive-definite inverse damping matrix that relates the joint torques τ to the joint velocities as $\dot{q} = B(q)\tau$. The first equation relates the total torque (due to external forces F_i and λ) to the change in the joint angles, the second equation describes the end point constraint that should be satisfied. Note that these equations do not relate to any actual physical needle motions and only represent a mathematical procedure.

Substituting (14) into (15), solving for λ , and substituting back into (14) results in an unconstrained equation for \dot{q} that no longer contains λ :

$$\dot{q} = (I - BJ^T(JBJ^T)^{-1}J) BJ_i^T F_i \quad (16)$$

This ODE describes the evolution of q under the influence of an external wrench F_i and tip constraint $J\dot{q} = 0$. It has a unique solution if B is invertible and J has full rank (no singularity). Equation (16) projects the velocity $BJ_i^T F_i$ due to the wrench F_i along the columns of BJ^T onto the null space of J . The matrix $B(q)$ defines a metric on the space of torques that can be chosen in any appropriate way, e.g. as a function that drives the system away from configurations that are singular or contain negative-length path segments.

For the example of Fig. 8b, we chose B diagonal with $B_{jj}(q) \rightarrow 0$ as $q_j \rightarrow 0$ for all joints j describing insertion path segments, thus avoiding negative-length path segments. We applied a linear force in the middle of the kinematic chain (link 5), directed toward one of the dots, and integrated (16) over time to obtain the pod-shaped family of needle paths shown in the figure.

This method of path adaptation can be used in fully automated motion planning (e.g. to perform gradient descent on some cost function with penalty costs for obstacle penetration) with changes in q constrained to be null motions. More directly, for computer-assisted motion planning as described in Section 6, it can provide the user with an intuitive path adjustment tool similar to the control points on a spline curve that can be moved to change its local shape. Although there is no guarantee this approach will always work, it provides the user with an additional tool to construct suitable needle paths.

6 Application in computer-assisted motion planning

The inverse-kinematics based motion planning algorithm quickly computes feasible needle paths and allows the user to focus on specifying higher-level objectives in terms of start and goal needle poses. Indeed, the main motivation and reason for computer-assistance in this motion planning problem is the degree of under-actuation and nonholonomicity, which can be dealt with using the presented approach.

Nevertheless, the algorithms provide no guarantees for a solution or a structured incremental way to search for other trajectories in case of failure

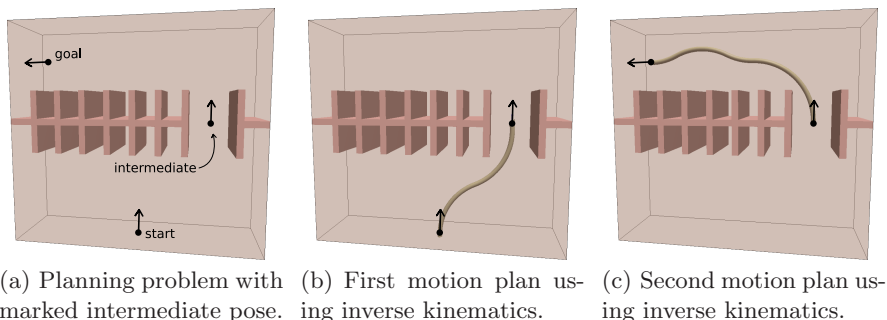


Fig. 9. Difficult motion planning problem solved using semi-automated planning.

due to obstacles or other complications. The only possibilities are to choose different parameter settings or, for the null-motion based method, to try applying external forces at a different point or in a different direction.

When it comes to global motion planning, computers are severely limited in cognitive abilities and can require large amounts of computation power and time to solve problems that are easy for humans (experiments using Rapidly Exploring Random Trees required up to half an hour of computation time [19]). Consider for instance the motion planning problem of Fig. 9a: for the reader it is instantly clear that any feasible path should pass near the intermediate point marked in the figure, but computer-based planners such as those described in the previous sections may not be able to find a feasible solution.

One way to solve this problem is to combine human cognitive abilities for global planning with computer power for local planning. If a human path planner indicates the desired intermediate point as in Fig. 9a, the automatic motion planning algorithms can be applied to solve the resulting two subpaths. Finding a path from the start location to the intermediate location is trivial, as is finding a path from the intermediate location to the goal. Figs. 9b and 9c show the resulting subpaths obtained using the inverse kinematics planner; comparable results are obtained when using direct numerical optimization [6]. For this example, the algorithms are not sensitive to the exact position and orientation of the intermediate pose, but the approach could be extended to iterate over several intermediate poses near the one indicated.

7 Conclusions and future work

This paper presents constant-time geometrically motivated motion planning algorithms for steerable needles and airplanes with constant speed and pitch rate, zero yaw, and controllable roll. The first algorithm uses inverse kinematics (IK) to explicitly compute feasible paths in 3D, the second uses null-motions to adapt paths to avoid obstacles or achieve other objectives.

As briefly discussed, these algorithms can be used as components in larger computer-assisted motion planning schemes that use limited user-input to guide automatic local planning. In future work, we also plan to use the IK algorithm as a local planner in (autonomous) roadmap-based algorithms such as PRM [11]. Recent results using Rapidly-Exploring Random Trees [19] are encouraging, although computation requirements are several orders of magnitude larger than with direct optimization-based algorithms [6].

Another main future direction of our research is to find a systematic way to include uncertainty during motion planning. Our application of steerable needles contains several sources of uncertainty, including needle motion uncertainty, tissue flexibility and friction, and sensing inaccuracies. These uncertainties should be taken into account in the motion planning stage, as discussed and implemented for the 2D case in previous work [3]. The presented fast local motion planning algorithm can be used to quickly test connectivity and iteratively study the effect of perturbations.

Finally, reachability and competitiveness analyses were presented for the 2D and 3D inverse kinematics algorithms. In future work, we plan to extend the analysis of the 3D algorithm to provide bounds on the competitiveness compared to the optimal shortest-path solution. A promising direction is the comparison with paths generated by the approach of Park et al. [15].

Acknowledgments

This work is supported in part by the National Institutes of Health under grants R01 EB006435 and F32 CA124138 and by the Netherlands Organization for Scientific Research. We thank Professors Okamura, Chirikjian, and Cowan from Johns Hopkins University for the helpful comments and suggestions, and Prof. Cowan for the idea of using null-motions for path adaptation.

References

1. R. Alterovitz, M. Branicky, and K. Goldberg. Constant-curvature motion planning under uncertainty with applications in image-guided medical needle steering. In *Proceedings of Workshop on the Algorithmic Foundations of Robotics*, July 2006.
2. R. Alterovitz, K. Goldberg, and A. Okamura. Planning for steerable bevel-tip needle insertion through 2D soft tissue with obstacles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1640–1645, April 2005.
3. R. Alterovitz, T. Siméon, and K. Goldberg. The stochastic motion roadmap: A sampling framework for planning with Markov motion uncertainty. In *Proceedings of Robotics: Science and Systems*, June 2007.
4. S. P. DiMaio and S. E. Salcudean. Needle steering and motion planning in soft tissues. *IEEE Transactions on Biomedical Engineering*, 52(6):965–974, June 2005.

5. 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*, 79(3):497–516, July 1957.
6. V. Duindam, R. Alterovitz, S. Sastry, and K. Goldberg. Screw-based motion planning for bevel-tip flexible needles in 3D environments with obstacles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2483–2488, May 2008.
7. J. A. Engh, G. Podnar, D. Kondziolka, and C. N. Riviere. Toward effective needle steering in brain tissue. In *Proc. 28th Annu. Intl. Conf. IEEE Eng. Med. Biol. Soc.*, pages 559–562, 2006.
8. Y. Gabriely and E. Rimon. *Algorithmic Foundations of Robotics VI*, volume 17 of *STAR*, chapter Competitive Complexity of Mobile Robot On Line Motion Planning Problems, pages 155–170. Springer-Verlag, 2005.
9. C. Icking and R. Klein. Competitive strategies for autonomous systems. In *Modelling and Planning for Sensor Based Intelligent Robot Systems*, pages 23–40, 1995.
10. V. Kallem and N. J. Cowan. Image-guided control of flexible bevel-tip needles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3015–3020, April 2007.
11. L. E. Kavvaki, P. Švestka, J. C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, August 1996.
12. D. S. Minhas, J. A. Engh, M. M. Fenske, and C. N. Riviere. Modeling of needle steering via duty-cycled spinning. In *Proceedings of the International Conference of the IEEE EMBS Cité Internationale*, pages 2756–2759, August 2007.
13. R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.
14. W. Park, J. S. Kim, Y. Zhou, N. J. Cowan, A. M. Okamura, and G. S. Chirikjian. Diffusion-based motion planning for a nonholonomic flexible needle model. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4611–4616, April 2005.
15. W. Park, Y. Liu, Y. Zhou, M. Moses, and G. S. Chirikjian. Kinematic state estimation and motion planning for stochastic nonholonomic systems using the exponential map. *Robotica*, 26:419–434, July 2008.
16. R. J. Webster III, J. S. Kim, N. J. Cowan, G. S. Chirikjian, and A. M. Okamura. Nonholonomic modeling of needle steering. *International Journal of Robotics Research*, 5/6:509–525, May 2006.
17. R. J. Webster III, J. Memisevic, and A. M. Okamura. Design considerations for robotic needle steering. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3588–3594, April 2005.
18. R. J. Webster III, A. M. Okamura, N. J. Cowan, G. S. Chirikjian, K. Goldberg, and R. Alterovitz. Distal bevel-tip needle control device and algorithm. US patent pending 11/436,995, May 2006.
19. 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 *Proceedings of the IEEE Conference on Automation Science and Engineering*, pages 41–46, August 2008.