3D path planning for flexible needle steering in neurosurgery

Originally published in:
The International Journal of Medical Robotics and Computer Assisted Surgery 15(4), https://doi.org/10.1002/rcs.1998

Funding acknowledgement:
743217 - Soft Micro Robotics (EC)
165564 - Soft Magnetic Robots: Modeling, Design and Control of Magnetically Guided Continuum Manipulators (SNF)

3D Path Planning for Flexible Needle Steering in Neurosurgery

Ayoung Hong, Quentin Boehler, Roman Moser, Ajmal Zemmar, Lennart Stieglitz, Bradley J. Nelson

***Abstract*—Background: We propose a 3D path planning method to steer flexible needles along curved paths in the context of Deep Brain Stimulation (DBS) procedures. Methods: Our approach is based on a rapidly-exploring random tree strategy and it takes into account constraints coming from anatomical obstacles and physical constraints dictated by flexible needle kinematics. The strategy is evaluated in simulation on a realistic 3D CAD model of the brain. Results: The subthalamic nucleus (STN) and the fornix can be reached along several curved paths from various entry points. As compared to the usual straight line path, these curved paths avoid tissue damage to important neural structures while allowing for a much greater selection of entry points. Conclusions: This path planning method offers alternative curved paths to reach DBS targets with flexible needles. The method potentially leads to safer paths and additional entry points capable of reaching the desired stimulation targets.**

***Index Terms*—surgical planning, deep brain stimulation, neurosurgery, flexible needle**

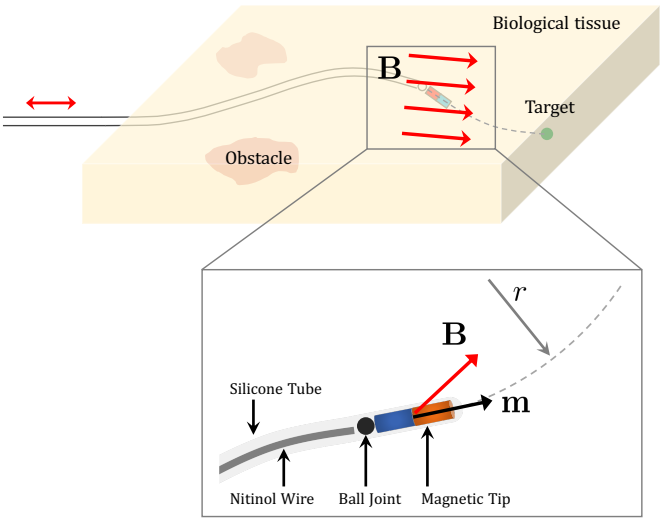


Fig. 1: Scheme of a magnetically guided flexible needle.

I. Introduction

Deep Brain Stimulation (DBS) is an invasive neurosurgical technique used to treat a variety of neurological disorders by electrically stimulating a brain region associated with a specific disease. These disorders include Parkinson's disease (PD), dystonia, and some psychiatric disorders, such as depression and obsessive-compulsive disorder (OCD) [1]–[3]. Stimulation of the subthalamic nucleus (STN), which is one of the most common targets for PD, reduces motor fluctuation in patients by more than 50% [4]. The stimulation of the ventral capsule/ventral striatum relieves the symptoms of patients with highly-resistant OCD [5], [6]. Increased cerebral metabolism after the stimulation of the fornix has been reported in patients with Alzheimer's disease (AD) [7], [8]. However, some adverse effects have been reported using current techniques [9], [10]. For example, patients have experienced a decline in verbal fluency, verbal long-term memory, or mental status after surgery, especially when the surgical trajectories intersect the brain ventricles, or when the final electrode is located far from the midpoint of the anterior to posterior commissure (AC-PC) line [11].

Currently, only rigid straight needles are used to reach deep locations in the brain, which confines the feasible path between the entry point and the target to a straight line. As an alternative, the use of a flexible needle allows for curved paths in neurosurgery. Bevel-tipped needles demonstrated the ability to reach several targets in cadaveric models [12]. The active control of a tip angle has demonstrated the feasibility of accessing a specific target region in a brain phantom employing a magnetic tip [13] and a programmable bevel-tip [14]. However, to precisely guide the flexible needle, surgical path planning is mandatory.

Computer-assisted planning methods have been suggested to support neurosurgeons by saving planning time and improving insertion safety [15], [16]. Suggested paths from automatic planning algorithms are readily accepted by neurosurgeons [17], [18]. However, these clinically available planning methods are only focused on straight paths. For curved paths, planning becomes less intuitive, and different surgical and physical constraints must be considered. Furthermore, the motion interaction of flexible needles with soft tissue, and the control efforts to guide them, must be considered.

The motion planning problem for flexible needles has been well-studied in the literature. A Markov decision process has been adapted to compute discretized inputs to steer a flexible bevel-tip needle [19] and sampling-based algorithms such as rapidly-exploring random trees (RRT) combined with the reachability-guided strategy has been presented in three-dimensional environments [20]. By using a distance metric considering the nonholonomic constraints of flexible needles, an RRT based algorithm was suggested [21]. A continuous curvature model has been proposed to determine the optimal path for flexible needles in neurosurgical applications [22],

A. Hong, Q. Boehler, R. Moser and B. J. Nelson are with Multiscale Robotics Laboratory, ETH Zürich, 8092 Zürich, Switzerland. (e-mail: bnelson@ethz.ch)

A. Zemmar is with Juha Hernesniemi International Neurosurgery Center, Henan Provincial People's Hospital, Zhengzhou University, China

L. Stieglitz is with Department of Neurosurgery, University Hospital Zurich, Zürich, Switzerland

however, this has only been demonstrated in 2D. Full 3D neurosurgical planning for flexible needles has yet to be reported.

This paper introduces a path planning method for flexible needles considering specific surgical and anatomical constraints in neurosurgery. Our approach adopts an RRT* sampling-based approach [23] and incorporates the physical constraints of flexible needles. The feasibility of our approach is assessed on a realistic anatomical model of the brain structures with two scenarios, targeting the STN for PD treatment, and the fornix for mild AD treatment. We consider the use of a magnetically guided flexible needle. The needle's physical constraints, previously introduced in [13], are incorporated into our generic path planning method. A set of realistic entry points are considered, and the best feasible curved path is selected based on path length and safety criteria. Our strategy results in a larger variety of feasible paths and entry points than those considered with straight needles, and safer paths toward the surgical target.

II. Materials and Methods

A. Magnetically Guided Flexible Needle

Magnetically guided flexible needles can be used to aid electrode placement, as depicted in Fig. 1. The needle consists of a rigid magnetic tip and a nonmagnetic flexible shaft connected with a ball joint [13]. Under an external magnetic field controlled by a surgeon, magnetic torque τ is exerted on the magnetic needle tip as $\tau = \mathbf{m} \times \mathbf{B}$, where $\mathbf{m}$ is the magnetic moment, and $\mathbf{B}$ is the applied magnetic field. This torque deflects the tip towards the field direction, subject to soft tissue and ball joint constraints. Due to the asymmetric stress surrounding the tip, the needle penetrates soft tissue in a preferential direction while the needle is linearly inserted, and retracted if necessary, at its distal end. Depending on the deflected tip angle, the needle follows a circular curve with curvature $\kappa = 1/r$. NB: the path planning method introduced here is generically formulated and can be applied to any type of flexible needle.

B. Anatomical Brain Model

To simulate a realistic environment for planning DBS procedures, a 3D model of the brain is utilized, which includes ventricles, arteries, and veins. The models of the STN and fornix are also included as target regions. The 3D model is voxelized with a resolution of $2 \times 2 \times 2$ mm^3 that simulates a data set acquired from MRI. These voxels are used for collision detection in the algorithm with a radius of 2 mm ($r_c = 2$ mm). The resolution of the voxel strongly affects the speed of the planning algorithm, because collision detection and its computational cost are iteratively required. We assume that we have access to preoperative magnetic resonance imaging (MRI), which specifies the geometry of the brain, including the target region and obstacles to be avoided. The automatic segmentation method to extract the geometry of anatomical constraints from medical images [24], [25] are not discussed in this paper.

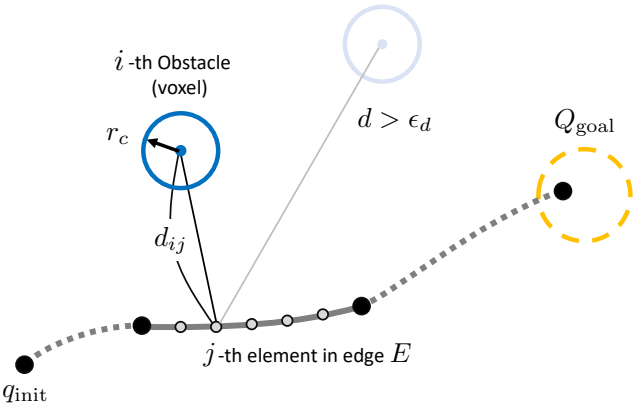


Fig. 2: A segment (i.e., edge E) in the path satisfying the collision-free condition; $d_{ij} > r_c$ for all obstacles. Obstacles located farther than ϵ_d to the edge E do not affect the cost function.

C. Surgical Constraints

To attain a successful surgical outcome, the target regions should be reached within the planned paths without penetrating critical structures in the brain. This introduces several anatomical constraints. First, the entry point is selected so that the midline is not penetrated. This confines the entry point to the ipsilateral region of the target. The entry point should be located posterior to the hairline due to cosmetic reasons. The sulci are avoided as entry points, because there are often small blood vessels at its base. Lastly, ventricles and large blood vessels should be avoided to prevent leakage of cerebrospinal fluid and hemorrhage.

We also consider physical constraints related to our flexible needle. First, the needle can only be steered along a continuous and smooth path. This restricts our choice of available paths to a set of straight lines or constant curvature curves that share a common tangent direction at their meeting point. Second, the needle has a minimum bend radius based on its mechanical properties and interaction with brain tissue. Therefore, the final determined paths should not include segments having a larger curvature than $1/r_{\text{min}}$.

These anatomical and physical constraints are used as hard constraints in the planning algorithm and must be satisfied [16]. Any path that does not satisfy these rules is rejected. The optimal solutions are selected using the cost function within the planning algorithm, which is further described in section II-D.

D. Cost Function

The goal of the planning algorithm is to find the safest path requiring the least effort to steer the flexible needle to the target region. This criterion can be considered in the planning algorithm using a cost function which incorporates two cost terms: the path length and the distance from obstacles. For each segment (i.e., an edge E) in the path, the cost function is defined as

$$c(E) = a_1 f_L + a_2 f_D \tag{1}$$

where f_L and f_D are the cost terms penalizing the path length and distance from obstacles, respectively, and a_1 and a_2 are the positive weights regulating the relative importance of each cost term. Here, the sum of weights is equal to one.

The cost term f_L is defined as

$$f_L = \frac{L}{L_{min}} \tag{2}$$

where L is the length of the edge and L_{min} is the possible minimum distance. For a straight path, f_L is minimum. As the path becomes more circuitous, a larger cost results. This represents the efficiency of the path, as it relates to the insertion duration. This also indicates the effort that is necessary to steer the needle.

The cost term f_D incorporates the safety of the path by considering the distance from the path to the obstacles as

$$\begin{aligned} f_D &= \frac{1}{N_o N_e} \sum_{i}^{N_o} \sum_{j}^{N_e} \frac{\epsilon_d - r_c}{d_{ij} - r_c} \\ d_{ij} &= \begin{cases} d_{ij} & \text{if } r_c < d_{ij} \leq \epsilon_d \\ \epsilon_d & \text{if } \epsilon_d < d_{ij} \end{cases} \end{aligned} \tag{3}$$

where N_o and N_e are the number of obstacles and elements in the edge, r_c is the minimum admissible distance to obstacles, ϵ_d is the maximum distance above which the cost function is not affected, d_{ij} is the distance from the i-th obstacle to the j-th element in the edge in Fig. 2.

E. Planning Approach

The planning algorithm computes feasible paths for a given initial point and a target point by optimizing the cost function discussed in section II-D and respecting the surgical constraints in section II-C. Before discussing the algorithm in detail, the primitive variables and functions are introduced as follow.

- A point q (i.e., a vertex), encodes the three-dimensional position and heading direction, its parent vertex, and the cost as q_{pos}, q_{head}, q_{parent}, and q_{cost}, respectively.
- An edge connects two vertices using `Line` or `Curve`.
- $\texttt{Line}(q_a, q_b)$ connects two vertices through a straight-line path.
- $\texttt{Curve}(q_a, q_b)$ connects two vertices with a constant curvature path by considering the heading direction of q_a.
- The set of vertices V and their corresponding edges E structure the tree $T = (V, E)$.
- $\texttt{Parent}(q)$ returns the parent vertex. A parent of an initial vertex is itself as $\texttt{Parent}(q_{\text{init}}) = q_{\text{init}}$.
- $\texttt{Cost}(q)$ calculates the accumulated cost of the edges connecting q from q_{init} as $\texttt{Cost}(q) = \texttt{Cost}(\texttt{Parent}(q)) + c(E(\texttt{Parent}(q), q))$ where $c(E)$ determines the cost of the edge E using (1). The cost value of an initial vertex is zero as $\texttt{Cost}(q_{\text{init}}) = 0$.
- $\kappa(q_a, q_b)$ computes the curvature of the edge connecting two vertices q_a and q_b as

$$\kappa(q_a, q_b) = \frac{2\|(q_{\text{a.pos}} - q_{\text{b.pos}}) \times q_{\text{b.head}}\|}{\|q_{\text{a.pos}} - q_{\text{b.pos}}\|^2}. \tag{4}$$

Algorithm 1 3D path planning for flexible needles

Input : $q_{\text{init}}, Q_{\text{goal}}$

$P \leftarrow \texttt{LinearPath}(q_{\text{init}}, Q_{\text{goal}}) \cup \texttt{CurvePath}(q_{\text{init}}, Q_{\text{goal}})$
$P \leftarrow P \cup \texttt{RRTStarPath}(q_{\text{init}}, Q_{\text{goal}}, N)$
$P_{\text{opt}} \leftarrow \texttt{MinimumCost}(P)$

Output : P_{opt}

$P \leftarrow \texttt{LinearPath}(q_{\text{init}}, Q_{\text{goal}})$
$E \leftarrow \texttt{Line}(q_{\text{init}}, Q_{\text{goal}})$
if $\texttt{CollisionFree}(E)$ **then**
  $P \leftarrow E$
else
  $P \leftarrow \emptyset$

$P \leftarrow \texttt{CurvePath}(q_{\text{init}}, Q_{\text{goal}})$
$E \leftarrow \texttt{Curve}(q_{\text{init}}, Q_{\text{goal}})$
if $\texttt{CollisionFree}(E)$ **and** $\kappa(q_{\text{init}}, Q_{\text{goal}}) < \frac{1}{r_{\min}}$ **then**
  $P \leftarrow E$
else
  $P \leftarrow \emptyset$

$P \leftarrow \texttt{RRTStarPath}(q_{\text{init}}, Q_{\text{goal}}, N)$
$P \leftarrow \emptyset, V \leftarrow \{q_{\text{init}}\}, E \leftarrow \emptyset, c_{\min} \leftarrow 0$
while $T \cap Q_{\text{goal}} = \emptyset$ **and** $i < N$ **do**
  $q_{\text{rand}} \leftarrow \texttt{SampleFree}$
  $q_{\text{nearest}} \leftarrow \texttt{NearestReachable}(T, q_{\text{rand}})$
  $q_{\text{new}} \leftarrow \texttt{Steer}(q_{\text{nearest}}, q_{\text{rand}})$
  if $\texttt{CollisionFree}(\texttt{Curve}(q_{\text{nearest}}, q_{\text{new}}))$ **then**
    $Q_{\text{near}} \leftarrow \texttt{Near}(T, q_{\text{new}}, r)$, $V \leftarrow V \cup \{q_{\text{new}}\}$
    $q_{\min} \leftarrow q_{\text{nearest}}$, $c_{\min} \leftarrow \texttt{Cost}(q_{\text{nearest}}) + c(\texttt{Curve}(q_{\text{nearest}}, q_{\text{new}}))$
    for each $q_{\text{near}} \in Q_{\text{near}}$ **do**
      if $\texttt{CollisionFree}(\texttt{Curve}(q_{\text{near}}, q_{\text{new}}))$ **and**
        $\texttt{Cost}(q_{\text{near}}) + c(\texttt{Curve}(q_{\text{near}}, q_{\text{new}})) < c_{\min}$ **and**
        $\kappa(q_{\text{near}}, q_{\text{new}}) < \frac{1}{r_{\min}}$ **then**
          $q_{\min} \leftarrow q_{\text{near}}$, $c_{\min} \leftarrow \texttt{Cost}(q_{\text{near}}) + c(\texttt{Curve}(q_{\text{near}}, q_{\text{new}}))$
    $E \leftarrow E \cup \{q_{\min}, q_{\text{new}}\}$
    for each $q_{\text{near}} \in Q_{\text{near}}$ **do**
      if $\texttt{CollisionFree}(\texttt{Curve}(q_{\text{new}}, q_{\text{near}}))$ **and**
        $\texttt{Cost}(q_{\text{new}}) + c(\texttt{Curve}(q_{\text{new}}, q_{\text{near}})) < \texttt{Cost}(q_{\text{near}})$ **and**
        $\kappa(q_{\text{near}}, q_{\text{new}}) < \frac{1}{r_{\min}}$ **then**
          $q_{\text{parent}} \leftarrow \texttt{Parent}(q_{\text{near}})$
          $E \leftarrow (E \backslash \{(q_{\text{parent}}, q_{\text{near}})\}) \cup \{(q_{\text{new}}, q_{\text{near}})\}$
  $i \leftarrow i + 1$
$P \leftarrow \texttt{Path}(T)$

- $\texttt{SampleFree}$ randomly selects a random vertex q_{rand} in the obstacle free space.

- `NearestReachable`$(T = (V, E), q)$ determines the vertex in V that is closest to q and reachable with an edge having a smaller curvature than $\frac{1}{r_{\min}}$. We use the distance measure as [26]

$$\rho = w_d d^2 + w_\theta (1 - |\cos\theta|)^2 \tag{5}$$

  where d and $1-|\cos\theta|$ are measures in Euclidean distance and orientation and w_d and w_θ are the weights for each measure satisfying $w_d + w_\theta = 1$. Here, the distance measure is normalized by the maximum distance between two vertices available in the space. By incorporating the position and orientation in the distance measure, we can penalize the vertex that is near but that requires a large effort to curve.
- `Near`$(T = (V, E), q, r)$ returns the vertices in V that are contained in a sphere of radius r centered at q[a].
- `Steer`(q_a, q_b) determines a new vertex by steering from q_a to q_b with a predefined distance $\eta > 0$.
- `CollisionFree`(q_a, q_b) examines whether the edge connecting two vertices q_a and q_b avoids collisions with obstacles. It returns true if all elements in E satisfy the condition: $d_{ij} > r_c$ for all obstacles in Fig. 2, and false otherwise.
- `Path`(T) determines the path from the goal region to the initial point by searching the tree backward.
- `MinimumCost`(P) selects the optimal one among the set of paths P to minimize the cost.

The overall planning algorithm is described in Alg. 1. For a given entry point q_{init} and target region Q_{goal}, the algorithm first finds a linear path and a constant curvature path that directly connects them using the function `LinearPath` and `CurvePath`, respectively. We also adopt an RRT* based planning algorithm using `RRTStarPath` to determine the tortuous path. These functions only return paths satisfying the surgical constraints by checking the curvature of the path and collision to obstacles. Among the valid paths, the optimal path is the one with minimum cost.

In `RRTStarPath`, we explore the tree iteratively until it reaches the goal region Q_{goal}, or when the maximum number of iteration N is exceeded. At each iteration, the nearest vertex q_{nearest} to q_{rand} is selected among the reachable set with the minimum distance measure. A new vertex q_{new} is determined as in Fig. 3 (a). The new vertex is connected through the one with minimum cost, and if there exists any vertex that gives a lower cost through the new one than its parent, it is rerouted through the new one as in Fig. 3 (b). These procedures provide solutions converging to an optimal solution with reasonable computational complexity. Once the new vertex reaches the goal region, the path is determined by searching backwards to its parent vertex.

The choice of the path planner parameters, i.e., the number of iterations N, the cost coefficients a_1 and a_2 in (1), and

[a]The radius is calculated as $r = \min\{\gamma(\log(\text{card}(V))/\text{card}(V))^{(1/3)}, \eta\}$ where $\text{card}(V)$ is the number of vertices in the tree. The asymptotic optimality of RRT* requires the constant value $\gamma > (2(1+1/3)^{1/3})(\mu(X_{free})/\zeta_d)^{1/3}$ where $\mu(X_{free})$ denotes the Lebesgue measure of the obstacle-free space, and ζ_d denotes the volume of the unit sphere [23].

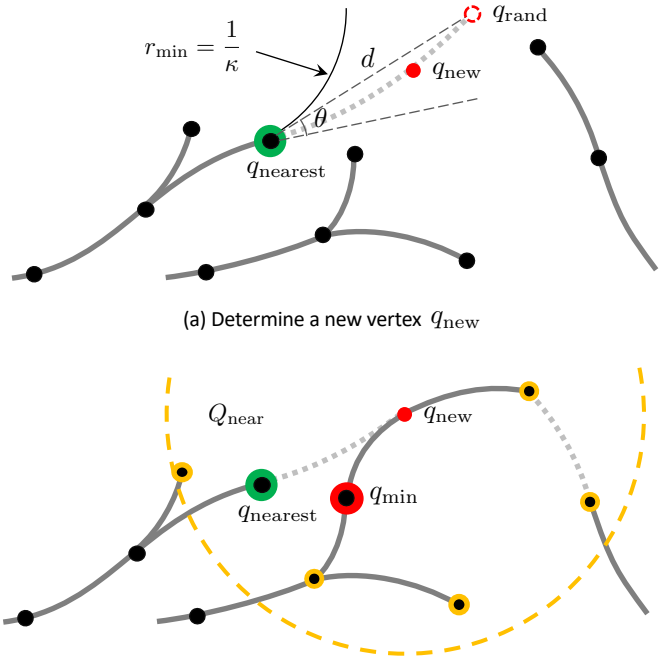
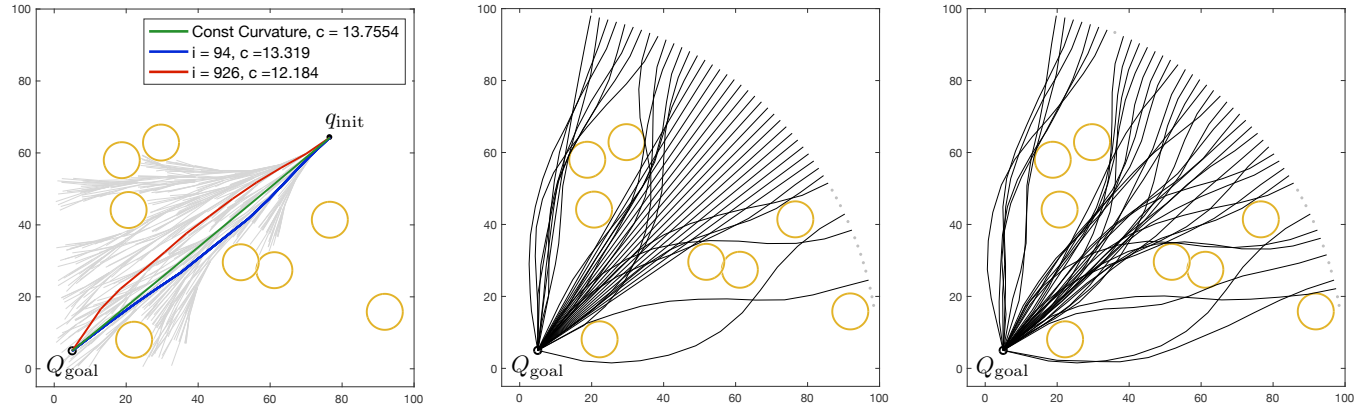


Fig. 3: Schematic of RRT* based path planning. Black spheres represent the existing vertices in the tree and the lines indicate the corresponding edges. (a) Determine q_{nearest} to q_{rand} among the existing vertices using the distance measure. Steer to q_{new} with a reachability test using the minimum bend radius of the flexible needle. (b) Connect q_{new} through q_{min}, which gives the minimum cost among the vertices near q_{new}. Reroute q_{near} through q_{new} if it gives a better cost than through its parent. Yellow spheres show the vertices in the set Q_{near}.

the weights for the distance measure w_d and w_θ in (5), have an influence on the results of `RRTStarPath`. The maximum number of iterations, for example, affects the optimality of the path and the computing time, as in Fig. 4 (a). The tree structure may not reach the goal region when the iteration number is too small and, thus, fails to determine the feasible path. With a large number of iterations, the algorithm may find a path with a lower cost despite an increased computing time.

The cost and the distance measure also affect the final paths. The tree structure grows depending on how we choose the nearest vertex, which is mainly determined by the distance measure. When the Euclidean distance is only considered, the algorithm may result in a circuitous tree structure. The algorithm produces straighter and shorter maps when the orientation is considered. In the process of connecting and rerouting edges, the cost affects the growth of the tree structure. It is also utilized in the last step of the algorithm `MinimumCost`. The effect of cost coefficients on the planning results are shown in Fig. 4 (b,c). When the cost coefficient for path length is set to one $(a_1 = 1, a_2 = 0)$, the shortest path is selected as long as the path avoids collision with obstacles. In this case, the paths determined from `LinearPath` and `CurvePath` are usually selected as the final paths. The safest path is chosen when the cost coefficient for obstacle avoidance



(a) (b) $a_1 = 1, a_2 = 0$ (c) $a_1 = 0, a_2 = 1$

Fig. 4: Path planning results in two-dimensions. (a) Planning results as a function of the iteration number. For a given entry point and a target region, the planning algorithm was ran for $N = 1000$. The path determined with a constant curvature is shown in green. `RRTStarPath` finds the first feasible path after 94 iterations (blue) and the optimal one is determined after 926 iterations (red). The grey lines show the tree structure from `RRTStarPath`. (b,c) Path planning results using different cost coefficients. Each black line shows the determined best path for each initial configuration. The grey dots indicate the initial configurations that the algorithm could not find the feasible path within the iterations ($N = 500$). The parameters used in simulations are $w_d = 0.5$, $w_\theta = 0.5$, $r_d = 5\,\text{mm}$, $\epsilon_d = 20\,\text{mm}$, and $\eta = 8\,\text{mm}$.

is set to one ($a_1 = 0, a_2 = 1$). In this case, the iterative method `RRTStarPath` yields the optimal path. These parameters affect the final path determined by the planning algorithm, however, they do not violate obstacle avoidance constraints and physical properties of flexible needles.

III. Results

A. Path planning for targeting STN

The proposed path planning algorithm was used to determine the feasible path by considering the anatomical constraints and the cost regulating its efficiency and safety. The maximum curvature allowed for the flexible needle was $0.025\,\text{mm}^{-1}$. The distance ϵ_d was set to 20 mm, implying that obstacles farther from the path than this distance are not reflected in the cost. We incorporated the blood vessels and the ventricles as the obstacles to be avoided with a minimum distance of 2 mm. The parameters used in the simulations were $w_d = 0.9$, $w_\theta = 0.1, \eta = 8\,\text{mm}$, $N = 1000$, and the cost coefficients are arbitrarily chosen as $a_1 = 0.1$, $a_2 = 0.9$.

The planning method was first applied to find the best feasible path from the predefined single entry point to the STN, which is the most common target region for PD. Figure 5 shows one path planning result in 3D and in the projected image. The algorithm determined the feasible path from `RRTStarPath` with a lower cost ($c = 9.8656$) than the one from `CurvePath` ($c = 10.0149$). For the same initial configuration, the algorithm was repeated for one hundred trials. On average, `RRTStarPath` determined the first feasible path after 85.14 ± 134.88 iterations with the cost of 10.0003 ± 0.1064. Figure 6 also provides plots of the average cost of the determined path from `RRTStarPath` and its success rate versus the number of iterations run. Success rate is defined as the rate with which `RRTStarPath` can find at least one feasible path within the maximum number of iterations. As the number of iterations increase, we could further improve the resultant path with a lower cost. When the number of iteration is small, such as $N = 50$, the algorithm failed 31 out of 100 trials in finding the feasible path. This success rate was improved to 100% as the number of iteration was raised to 750.

The planning algorithm to optimize the entry point is tested in the same environment. To avoid an insertion point near the sulci, feasible entry points were determined using the structure of the cerebrum, as in Fig. 7 (a). The initial configuration is determined with an exhaustive search within a given insertion area. The color-coded maps in Fig. 7 (b,c) show the minimal distance of the determined path to vessels and ventricles, which range from 2 mm to 8.1 mm and 5.6 mm, respectively. This confirms that the planning algorithm followed the rule for obstacle avoidance with a minimum distance of 2 mm. The entry points which could provide feasible paths are only represented in the map. Although most of the selected entry points result in a good distance to the vessels, the minimal distance becomes larger toward the center of gyri. As the entry point gets closer to the midline, it becomes harder to reach the target nuclei with a simple path because of the ventricles in between. The determined path has a safer trajectory with respect to the ventricles when it gets farther away from midline. The total cost map demonstrated in Fig. 7 (d) is obtained using the cost function discussed in section II-D, concerning the distance from obstacles and its associated path length. Overall, the algorithm prefers an entry point toward the center of gyri and lateral to the midline.

The algorithm takes 7 ms to find a linear path and a constant curvature path and 1.4 s to determine the tortuous path, on average, for 921 different entry points.

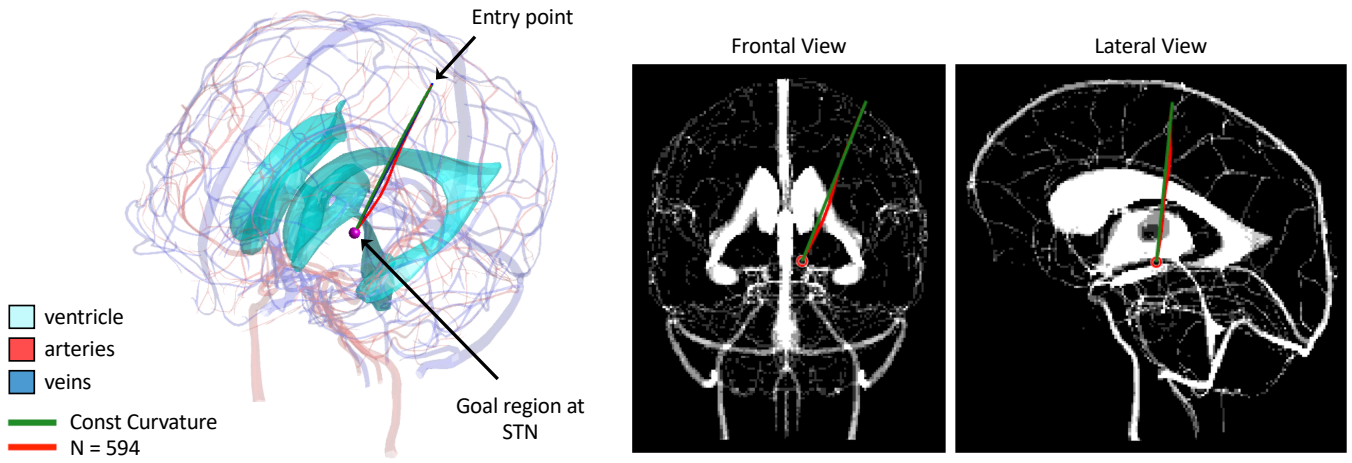


Fig. 5: (a) Path planning results for targeting STN in 3D. (b) A constant curvature (green line) and an RRT*-based (red line) path overlaid on the projected image of human brain CAD model that includes ventricles and brain vessels. The model was projected in the frontal and lateral directions, respectively.

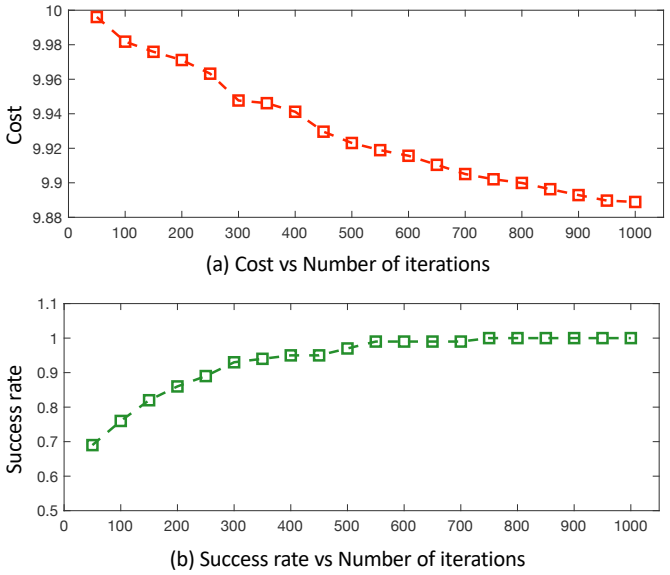


Fig. 6: Performance of the planning approach. (a) The cost of the best paths as a function of iterations averaged over 100 trials. (b) The success rate of the algorithm as a function of iterations.

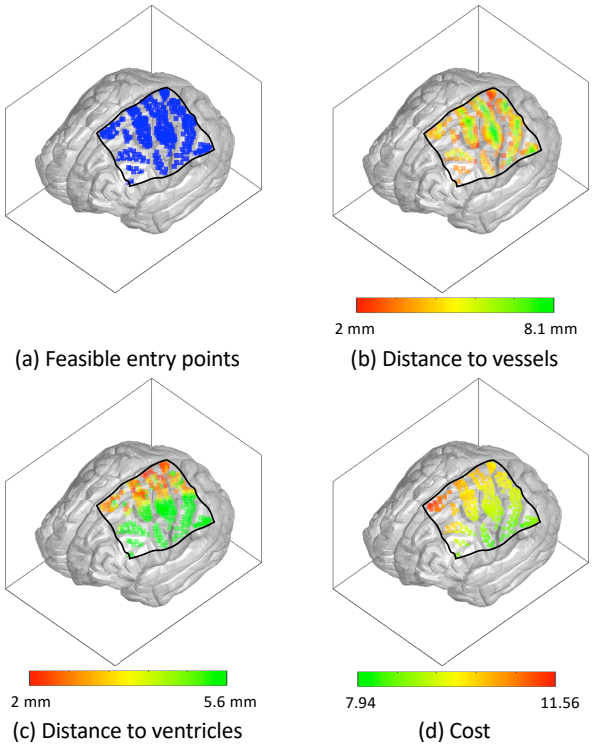


Fig. 7: Path planning result with the entry point optimization. (a) Feasible entry points considering the distance to sulci. Color-coded maps showing the minimal distance to (b) vessels and (c) ventricles from the best path determined and its cost (d) for each initial configuration.

B. Path planning for targeting fornix

The proposed planning approach was also tested for a different target. In treating AD, the goal is to bring the electrodes sufficiently close to the fornix, usually its descending limbs, but still avoiding it. Reaching the fornix is more demanding since it is surrounded by lateral ventricles. Here, the results of the proposed planning algorithm is compared to the straight path. It was possible to reach the target through 57% of the given entry points, which is considerably larger than the straight path which was only available for 23% of them. Figure 8 shows this difference by projecting the feasible entry points on the surface of cerebrum. Most of the straight paths started from the pre-central gyrus, while the curved paths provided entry points in the superior and medial frontal gyrus. Ultimately, this rejects most of the straight paths targeting the fornix. Not only does the proposed planning approach

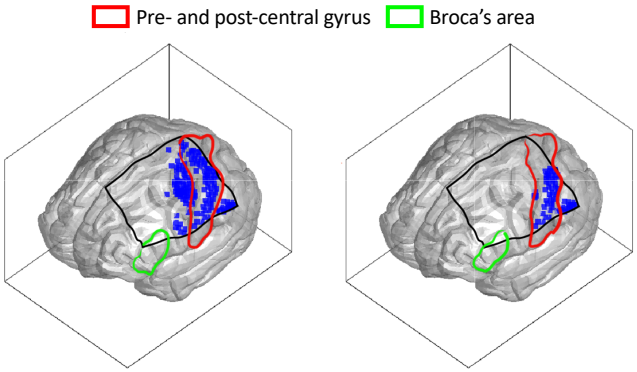


Fig. 8: Feasible entry points from (a) the proposed approach compared to (b) the straight path.

provide a greater choice of entry points, but it also suggest a safer surgical trajectory. The minimal distance to ventricle from the straight path ranged from 2 mm to 3.38 mm, while the path determined from our approach could avoid ventricles with a distance of 2 mm to 4.06 mm. The determined paths were reviewed by two neurosurgeons, an experienced stereotactic neurosurgeon (Lennart Stieglitz) and a resident neurosurgeon (Ajmal Zemmar), and found theoretically superior compared with straight paths.

IV. Discussion

A new 3D path planning method is introduced to guide flexible needles along curved paths for DBS applications. We adopt an RRT* based planning algorithm accounting for anatomical and flexible needle constraints. The path length and distance to obstacles are both considered in a dedicated cost function used in the exploration of the tree, and in the selection of the best path among a set of feasible paths and entry points. Our method is evaluated in simulation using realistic anatomical obstacles from a human brain CAD model and shows the ability to find multiple curved paths from a set of predetermined feasible entry points to the target region.

As compared to current DBS techniques, steering flexible needles along curved paths using the proposed path planner presents several advantages. First, many more entry points can be considered, and second safer paths are selected (i.e., farther from obstacles). To this extent, our method makes it possible to reach the target nuclei with a low-risk path by avoiding ventricles and brain vessels from a safe distance, which otherwise may cause a decline in verbal fluency, memory performance, and hemorrhage.

The frontal and parietal lobe is considered as the entry region in our approach. Entry was constrained to be in the ipsilateral region to the target and posterior to the hairline. These anatomical constraints for the entry points can be further extended by accommodating the functional anatomy of the brain. Some highly important cortical regions, such as the central lobe and Broca's area, should be avoided [27]. The precentral gyrus, the site of the primary motor cortex, is also to be avoided [28]. In planning for a target near the fornix, most straight paths started from the precentral gyrus, while feasible curved paths provided entry points in the superior and medial frontal gyrus. Thus, most straight paths for targeting the fornix are rejected, making our approach of interest for providing safe paths to this region. Additional obstacles can be added by neurosurgeons pre-operatively, which may increase procedure time. However, pre-processing and planning constitute one of the main steps in stereotactic surgeries. The neurosurgeon invests a considerable amount of time for this part. Therefore, even if there is added workload for the neurosurgeon during trajectory planning, it is worth the extra time if it aids in making the surgery safer and decreases operative time and unknown intraoperative variables for the neurosurgeon.

The cost function coefficients a_1 and a_2, and the number of iterations N, have an influence on the selected final path. However, these parameters do not affect the feasibility of the generated paths, because all surgical constraints, including anatomical obstacles and physical properties of the flexible needles, are primarily considered in the algorithm. Although neurosurgeons may choose different paths and their associated cost coefficients on their own preference, the optimal set of cost coefficients can be further investigated by performing qualitative evaluations that include user-experience questionnaires [15] and by measuring quantitative values, for example, distances of the determined path to critical areas, such as sulci, vessels, and ventricles [29]. The choice of the number of iterations is a trade-off between the optimality of the final path and computing time. For example, this number can be chosen sufficiently high so that at least one feasible path is found for every considered entry point. However, this may increase computing time. As any computer-assisted path planning method, our strategy requires a segmentation process of the brain anatomy that is currently not performed in the clinical workflow. Such a process can be time-consuming when done manually, however, automatic segmentation methods can be further considered for our application without necessarily leading to an increase in the procedure time.

Acknowledgments

This research was supported by the Swiss National Science Foundation through grant numbers 200021_165564 and the European Research Council Advanced Grant SOMBOT(743217). The authors would like to thank Nancy M. Amato and Carlos Alcantara for their helpful discussion.

References

[1] K. Ashkan, P. Rogers, H. Bergman, and I. Ughratdar, "Insights into the mechanisms of deep brain stimulation," *Nature Reviews Neurology*, vol. 13, no. 9, p. 548, 2017.

[2] J. Jankovic, "Treatment of hyperkinetic movement disorders," *Lancet Neurol*, vol. 8, no. 9, pp. 844–856, 2009.

[3] D. R. Cleary, A. Ozpinar, A. M. Raslan, and A. L. Ko, "Deep brain stimulation for psychiatric disorders: where we are now," *Neurosurg Focus*, vol. 38, no. 6, p. E2, 2015.

[4] P. Krack, A. Batir, N. Van Blercom, S. Chabardes, V. Fraix, C. Ardouin, A. Koudsie, P. D. Limousin, A. Benazzouz, J. F. LeBas, *et al.*, "Five-year follow-up of bilateral stimulation of the subthalamic nucleus in advanced parkinson's disease," *N Engl J Med*, vol. 349, no. 20, pp. 1925–1934, 2003.

[5] B. D. Greenberg, D. A. Malone, G. M. Friehs, A. R. Rezai, C. S. Kubu, P. F. Malloy, S. P. Salloway, M. S. Okun, W. K. Goodman, and S. A. Rasmussen, "Three-year outcomes in deep brain stimulation for highly resistant obsessive-compulsive disorder," *Neuropsychopharmacology*, vol. 31, no. 11, p. 2384, 2006.

[6] D. D. Dougherty, A. R. Rezai, L. L. Carpenter, R. H. Howland, M. T. Bhati, J. P. OŠReardon, E. N. Eskandar, G. H. Baltuch, A. D. Machado, D. Kondziolka, *et al.*, "A randomized sham-controlled trial of deep brain stimulation of the ventral capsule/ventral striatum for chronic treatment-resistant depression," *Biol Psychiatry*, vol. 78, no. 4, pp. 240–248, 2015.

[7] G. S. Smith, A. W. Laxton, D. F. Tang-Wai, M. P. McAndrews, A. O. Diaconescu, C. I. Workman, and A. M. Lozano, "Increased cerebral metabolism after 1 year of deep brain stimulation in alzheimer disease," *Arch Neurol*, vol. 69, no. 9, pp. 1141–1148, 2012.

[8] F. A. Ponce, W. F. Asaad, K. D. Foote, W. S. Anderson, G. Rees Cosgrove, G. H. Baltuch, K. Beasley, D. E. Reymers, E. S. Oh, S. D. Targum, *et al.*, "Bilateral deep brain stimulation of the fornix for alzheimer's disease: surgical safety in the advance trial," *Journal of neurosurgery*, vol. 125, no. 1, pp. 75–84, 2016.

[9] V. Voon, C. Kubu, P. Krack, J.-L. Houeto, and A. I. Tröster, "Deep brain stimulation: neuropsychological and neuropsychiatric issues," *Mov Disord*, vol. 21, no. S14, 2006.

[10] H. L. Combs, B. S. Folley, D. T. Berry, S. C. Segerstrom, D. Y. Han, A. J. Anderson-Mooney, B. D. Walls, and C. van Horne, "Cognition and depression following deep brain stimulation of the subthalamic nucleus and globus pallidus pars internus in parkinson's disease: a meta-analysis," *Neuropsychol Rev*, vol. 25, no. 4, pp. 439–454, 2015.

[11] M. K. York, E. A. Wilde, R. Simpson, and J. Jankovic, "Relationship between neuropsychological outcome and dbs surgical trajectory and electrode location," *J Neurol Sci*, vol. 287, no. 1, pp. 159–171, 2009.

[12] J. A. Engh, D. S. Minhas, D. Kondziolka, and C. N. Riviere, "Percutaneous intracerebral navigation by duty-cycled spinning of flexible bevel-tipped needles," *Neurosurgery*, vol. 67, no. 4, pp. 1117–1123, 2010.

[13] A. J. Petruska, F. Ruetz, A. Hong, L. Regli, O. Sürücü, A. Zemmar, and B. J. Nelson, "Magnetic needle guidance for neurosurgery: Initial design and proof of concept," in *IEEE Int Conf Robot Autom*, pp. 4392–4397, May 2016.

[14] S. Y. Ko and F. R. y Baena, "Toward a miniaturized needle steering system with path planning for obstacle avoidance," *IEEE Trans Biomed Eng*, vol. 60, no. 4, pp. 910–917, 2013.

[15] E. J. Brunenberg, A. Vilanova, V. Visser-Vandewalle, Y. Temel, L. Ackermans, B. Platel, and B. M. ter Haar Romeny, "Automatic trajectory planning for deep brain stimulation: a feasibility study," in *Med Image Comput Comput Assist Interv*, pp. 584–592, 2007.

[16] C. Essert, C. Haegelen, F. Lalys, A. Abadie, and P. Jannin, "Automatic computation of electrode trajectories for deep brain stimulation: a hybrid symbolic and numerical approach," *Int J Comput Assist Radiol Surg*, vol. 7, no. 4, pp. 517–532, 2012.

[17] S. Bériault, F. Al Subaie, D. L. Collins, A. F. Sadikot, and G. B. Pike, "A multi-modal approach to computer-assisted deep brain stimulation trajectory planning," *Int J Comput Assist Radiol Surg*, vol. 7, no. 5, pp. 687–704, 2012.

[18] Y. Liu, P. E. Konrad, J. S. Neimat, S. B. Tatter, H. Yu, R. D. Datteri, B. A. Landman, J. H. Noble, S. Pallavaram, B. M. Dawant, *et al.*, "Multisurgeon, multisite validation of a trajectory planning algorithm for deep brain stimulation procedures," *IEEE Trans Biomed Eng*, vol. 61, no. 9, pp. 2479–2487, 2014.

[19] R. Alterovitz, A. Lim, K. Goldberg, G. S. Chirikjian, and A. M. Okamura, "Steering flexible needles under markov motion uncertainty," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 1570–1575, IEEE, 2005.

[20] S. Patil and R. Alterovitz, "Interactive motion planning for steerable needles in 3d environments with obstacles," in *Proc IEEE RAS EMBS Int Conf Biomed Robot Biomechatron*, pp. 893–899, 2010.

[21] S. Patil, J. Burgner, R. J. Webster, and R. Alterovitz, "Needle steering in 3-d via rapid replanning," *IEEE Trans Rob*, vol. 30, no. 4, pp. 853–864, 2014.

[22] S. Bano, S. Y. Ko, and F. R. y Baena, "Smooth path planning for a biologically-inspired neurosurgical probe," in *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pp. 920–923, IEEE, 2012.

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

[24] D. L. Collins, C. J. Holmes, T. M. Peters, and A. C. Evans, "Automatic 3-d model-based neuroanatomical segmentation," *Hum Brain Mapp*, vol. 3, no. 3, pp. 190–208, 1995.

[25] P. Coupé, J. V. Manjón, V. Fonov, J. Pruessner, M. Robles, and D. L. Collins, "Patch-based segmentation using expert priors: Application to hippocampus and ventricle segmentation," *NeuroImage*, vol. 54, no. 2, pp. 940–954, 2011.

[26] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *Int J Rob Res*, vol. 20, no. 5, pp. 378–400, 2001.

[27] E. F. Chang, K. P. Raygor, and M. S. Berger, "Contemporary model of language organization: an overview for neurosurgeons," *J Neurosurg*, vol. 122, no. 2, pp. 250–261, 2015.

[28] T. Frigeri, E. Paglioli, E. de Oliveira, and A. L. Rhoton Jr, "Microsurgical anatomy of the central lobe," *J Neurosurg*, vol. 122, no. 3, pp. 483–498, 2015.

[29] N. Hamzé, J. Voirin, P. Collet, P. Jannin, C. Haegelen, and C. Essert, "Pareto front vs. weighted sum for automatic trajectory planning of deep brain stimulation," in *Med Image Comput Comput Assist Interv*, pp. 534–541, 2016.