“© 20XX IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

DOI: 10.1109/TRO.2020.3043692

An evolutionary-optimized surgical path planner for a programmable bevel-tip needle

Alberto Favaro*[a], *Member, IEEE*, Alice Segato[a], *Student Member, IEEE*, Federico Muretti[a] and Elena De Momi[a], *Senior Member, IEEE*

***Abstract*—Path planning algorithms for steerable needles in medical applications must guarantee the anatomical obstacle avoidance, reduce the insertion length and ensure the compliance with the needle kinematics. The majority of the solutions from the literature focus either on fast computation or on path optimality, the former at the expense of sub-optimal paths, the latter by making unbearable the computation in case of a high dimensional workspace. We implemented a 3D path planner for neurosurgical applications which keeps the computational cost consistent with standard pre-operative planning algorithms and fine-tunes the estimated pathways in accordance to multiple optimization objectives. From a user-defined entry point, our method confines a sample-based path search within a subsection of the original workspace considering the degree of curvature admitted by the needle. An evolutionary optimization procedure is used to maximize the obstacle avoidance and reduce the insertion length. The pool of optimized solutions is examined through a cost function to determine the best path. Simulations on one dataset showed the ability of the planner to save time and overcome the state of the art in terms of obstacle avoidance, insertion length and probability of failure, proving this algorithm as a valid planning method for complex environments.**

***Index Terms*—Surgical Robotics: Planning, Surgical Robotics: Steerable Catheters/Needles, Flexible Robots, Motion and Path Planning.**

I. INTRODUCTION

Flexible, small-scale catheters allow reaching deep regions inside the human body. Continuous robots represent a category of robotic tools that provides the required level of dexterity and reliability to perform delicate surgical procedures [1]. As they are flexible and can be built in small-scale, they can be useful in neurosurgery for situations where the access to anatomical structures is particularly challenging due to a cluttered anatomical workspace.

In [2–4], the steering of a flexible robotic probe is achieved using a bevel-tip needle with a fixed shape which rotates according to a duty-cycle that determines the bending of the needle toward the desired direction. Glozman and Shoham in [5] used an external base to which the needle was anchored to drive the insertion and to follow a desired trajectory. Pre-bent concentric elastic tubes [6] use an axial rotation and translation at the base to make the entire needle shape varies. A bio-inspired, multi-segment programmable bevel-tip needle (PBN) is currently under development as core technology of the EDEN2020* project [7]. The PBN consists of a continuous robot manufactured with four axially-interlocked sections. These segments are robotically actuated so that, sliding over each other, they can generate an offset on the needle tip and determine a curvature in the 3D space. The kinematics of the probe is still under investigation but a PBN control strategy is proposed in [8].

The benefits in maneuverability exhibited by flexible manipulators give rise to an increased number of possible needle pathways that can be correctly handled only by an automatic planner. Such a planner can be designed to fine-tune the path according to specific optimality criteria. Essert *et al.* [9] and Patil *et al.* [10] formalized the implicit and explicit principles used by neurosurgeons for the definition of optimal trajectories. The minimization of the surgical path length and the maximization of the clearance from anatomical obstacles can be considered as general axioms aiming at limiting the tissue damage while providing a proper safety margin in case deviations from the planned path occur.

In our previous work [11], we presented a neurosurgical planner for the PBN. The algorithm solves a single-query planning task, i.e. it connects an entry point to a target, guaranteeing the clearance from anatomical obstacles and complying with the PBN's kinematic limits.

The present work aims at describing a novel approach that improves the performance of our previous method on different aspects. Tailored to the PBN, the algorithm focuses the research using a smart redefinition of the search space so that the desired target point can be reached guaranteeing the compliance with the maximum degree of curvature admitted by the needle. This approach permits to save time by examining only the part of the workspace where feasible solutions can lie, reducing the computational cost correlated to the initial search phase. The method uses an innovative evolutionary procedure to perform a path optimization. Optimality criteria provide for the path length minimization, the reduction of the magnitude and variability of the path curvature and the maximization of the obstacle avoidance. A cost function is designed to provide an optimality index for each solution and ultimately define the best one.

The paper is structured as follows. In Section II, an overview of the path planning approaches proposed in the literature is presented. Section III outlines our planning method, focusing on the redefinition of the search space and on the evolutionary optimization. Section IV describes the experimental protocol. Section V presents the comparison between the presented solution and other methods from the literature. Discussion and Conclusions are reported in Sections VI and VII, respectively.

*This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No. 688279

[a]NearLab, Department of Electronics, Information and Bioengineering Department (DEIB), Politecnico di Milano, 20133, Milan, Italy alberto.favaro@polimi.it, elena.demomi@polimi.it

II. RELATED WORKS

In the context of path planning, a variety of approaches has been proposed in the literature.

In Schulman *et at.* [12], the non-convex path optimization problem is subdivided in convex sub-problems, solved via sequential convex optimization. The method is feasible for underactuated non-holonomic systems as flexible medical needles. Although the method does not guarantee to find a solution if it exists, it can provide locally optimal collision-free paths.

Duindam *et al.* [13] proposed a solution for estimating catheter pathways totally described in geometric terms and inspired by the Paden-Kahan subproblem, an explicit solution to inverse kinematics used as alternative to the implicit expression provided by the Denavit-Hartenberg parameters. The method was tested in a simplified environment and showed a high speed in the path computation but with limited obstacle avoidance capabilities.

Potential field methods, based on the idea originally introduced by Khatib [14], compute a potential field similar to the one generated by electrical charges. The potential field results from the interaction between the attractive effect of the target contrasted by the repulsive action of the obstacles. This approach has the drawback of generating local minima. To address this problem, Li *et al.* [15] proposed an artificial potential field method for applications in brachytherapy where a conjugate gradient algorithm was exploited. The clearance from anatomical obstacles was achieved but, as such, the method does not comply with other requirements as the optimization of the total path length or the compliance with catheter's kinematics.

Other approaches fall back into two main categories:

A. Graph-based methods

Graph-search methods are based on the discrete approximation of the planning problem. They are "resolution complete" as they can determine in a finite time whether a solution exists, and "resolution optimal" as they can estimate the best path given a specific resolution.

Likhachev *et al.* [16] proposed the incremental A* solution for 2D applications, which extended the original A* planning method so that to reuse previous information and drive the path towards optimality.

These methodologies show high computational time as the discretization of the environment becomes finer. For this reason, they are not suitable for neurosurgical applications as high-resolution 3D datasets ($\sim 10^7$ points) are normally used.

Leibrandt *et al.* [17] reported a multinode computational framework for fast path planning for concentric tube robots. The method builds an undirected graph of possible transition of needle configurations and queries the graph using A* search to extract the shortest path between the current and the desired tip pose. To address the high computational effort required by the graph search and achieve an interactive rate, a parallel computation is used and the search is confined in a subset of needle configurations close the current one.

B. Sampling-based methods

Sampling-based methods are built on the random sampling of the workspace.

Rapidly-exploring Random Trees (RRT) and RRT-Connect [18] are able to cope better with dense workspace than graph-based approaches. RRT* [19] and bidirectional-RRT [20] are "probabilistically complete": as the number of samples tends to infinity, the probability of finding a solution (provided that it exists) tends to one. Moreover, they are also "asymptotically optimal" as they can refine the initially-estimated raw path when new points are sampled and compute the shortest pathway to connect the start and target points as the number of iterations tends to infinity.

A combination of RRT and a reachability-guided sampling heuristic (RG-RRT) was used in the work of Patil *et al.* [21] to compute motion plans for steerable needles in complex 3D environments by constructing the tree via a sequential connection of arcs with bounded curvature. These solutions can be used in real-time applications, but performance have been assessed only in simplified workspaces. A neurosurgical 2D implementation of RG-RRT was proposed by Caborni *et al.* in [22].

Patil *et al.* [10] proposed a solution for duty-cycling steerable needles able to plan and control the needle motion in a closed loop fashion, guaranteeing obstacles avoidance and uncertainties compensation. Confining the search on the subset of points in the workspace that meet the kinematic constraints of the needle allows a replanning of the path suitable for online procedures, ensuring a clinically acceptable error.

The planning approach proposed in [10] has been used also by Kuntz *et al.* [23] for motion planning of a multi-lumen system for lung biopsy. The system consisted in a concentric tube robot with a beveled tip steerable needle. Simulations demonstrated the ability to quickly computes plans with high clearance from obstacles.

Parallel path computation is used in the Adaptive Fractal Tree (AFT) method [24], which exploits fractal theory and parallelization to separately process the subspaces and build a tree composed of arcs with bounded curvature. This method was designed for real time, computer-assisted minimally invasive surgical applications and requires a high performance Graphics Processing Unit to cope with the domain discretization. The AFT concept was the basis for the Adaptive Hermite Fractal Tree proposed in [25], where the fractal structure was combined with geometric Hermite curves to optimize the path for its smoothness and account for the target heading (i.e. the orientation) at both the start and the target points.

Gammell at al. [26] proposed the Batch Informed Tree (BIT*) algorithm. BIT* confines the search within an ellipsoidal region whose size relates to the cost of the current solution in a way that, any time a shorter path is found, the search occurs within a subspace progressively smaller in size. In our previous work [11], a similar approach has been implemented for developing a path planner for steerable needles where the criterion used to shape the search region accounts only for the length of the current best path and includes parts of the workspace that cannot be reached by the catheter due to its kinematic limits, representing a waste in term of computational

time. Additionally, the path optimization was hindered by the intrinsically limited flexibility of cardinal splines used for the path interpolation.

Evolutionary algorithms [27] are a popular approach well known for their simplicity, flexibility and robustness, that simulate the natural evolution to solve complex optimization problems. This optimization is performed according to specific parameters, which can be fixed or can vary during the optimization process [28]. A review of the different evolutionary approaches is reported in [29].

In the present work, we propose a new path planning solution where the heuristic path search is confined within the region of the workspace that can be actually reached by the needle, exploiting the capability of a sampling-based planner in dealing with dense workspaces. Additionally, an evolutionary procedure based on a different path interpolation method with respect to [11] and supported by a bespoke cost function is proposed to perform a multi-objective path optimization.

III. Methods

Our 3D path planner method consists in three main steps: the path planning (Section III-B, where a set of piece-wise linear paths is computed for the planning query), the path approximation and optimization (Section III-C, where the Evolutionary Optimization Procedure generates smooth paths, reduces their length and maximizes the obstacle avoidance) and the exhaustive search for the best path (Section III-D, where an exhaustive search is performed over the set of paths for determining the best planning solution). The workflow of our solution is reported in Fig. 1.

A. Surgeon's input and patient's data elaboration

In this section, the user input and how it is interpreted by the algorithm is described.

As first step, the surgeon is asked to select the desired target point (TP) within the brain and the entry point (EP_0) on the brain cortex. The system delineates an entry area around EP_0 excluding the sulci (due to the presence of blood vessels, as in [30]). A mesh decimation is performed on the entry area and a pool of feasible entry points EP_i, $i \in 1, ..., N$ is defined, as in [31]. The total number entry points (N) depends upon the radius of the entry area and the decimation level.

The anatomical obstacles are segmented in the patient's dataset images and a distance map is computed [32].

B. Path planning

In the following, we describe our approach of minimizing the search space to achieve a fast path planning. Geometrical properties derived from the needle kinematics are considered to design the search space.

The original workspace consists in the volume represented by the patient's brain. This volume is sampled with a resolution of $1\,\mathrm{mm}^3$ so that an uniformly-distributed set of 3D points is obtained.

For each EP_i, $i \in 1, ..., N$, a Kinematics Search Volume ($\mathrm{KSV}_i \in \mathbb{R}^3$) is defined, representing the "smart redefinition

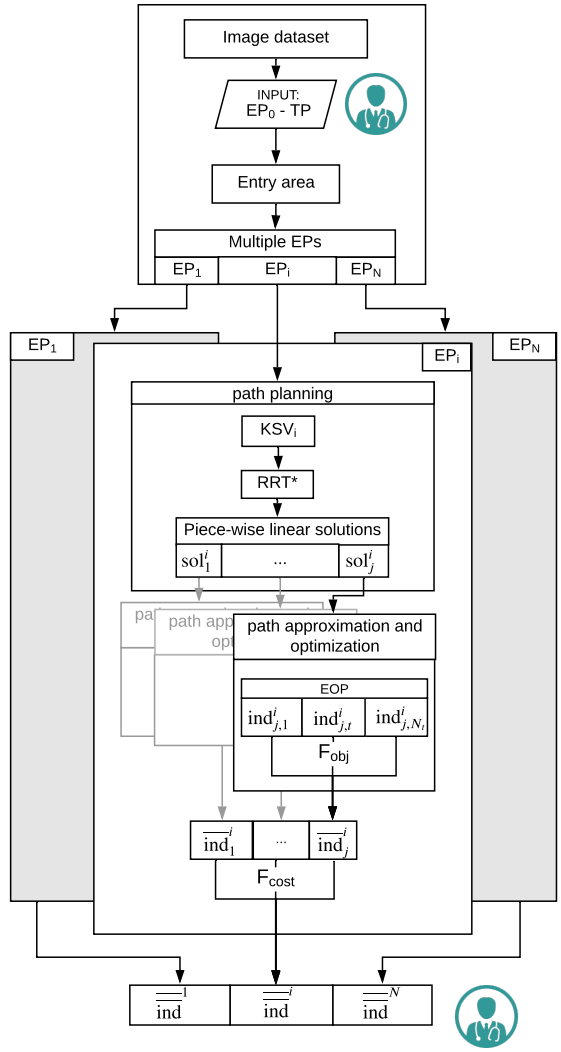


Fig. 1: Schematic representation of the workflow. The surgeon is asked to define the entry point (EP_0) and the target point (TP) on the patient's dataset. The entry area is then computed and the set of possible entry points EP_i, $i \in 1, ..., N$ is defined (Section III-A). Subsequently, for each EP_i, the algorithm performs the path planning (Section III-B) and the Evolutionary Optimization Procedure, EOP (Section III-C). A number of feasible solutions $\overline{\mathrm{ind}}^i_j$ are generated. For each EP_i, the best path $\overline{\overline{\mathrm{ind}}}^i$ is computed and eventually provided to the surgeon by running a cost function over the set of solutions generated by the EOP (Section III-D).

of the search space" for the path planning. The representation in 2D of the KSV as a projection on the *xy*-plane is reported in Fig. 2a, b while a 3D rendering of the KSV is provided in Fig. 2c.

The reason for defining the KSV is to identify a space where, considering the maximum curvature of the PBN (k_{PBN}), the needle tip can be oriented at any $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ without preventing the possibility to reach the TP (see Fig. 2a, b). The remaining workspace would imply a curvature greater than k_{PBN} to connect the EP_i to the TP. The case of $\theta \notin [-\frac{\pi}{2}, \frac{\pi}{2}]$ is not considered as far-fetched for the intended application. The KSV is defined considering the EP_i, the TP and the maximum curvature of the PBN (k_{PBN}). To reduce

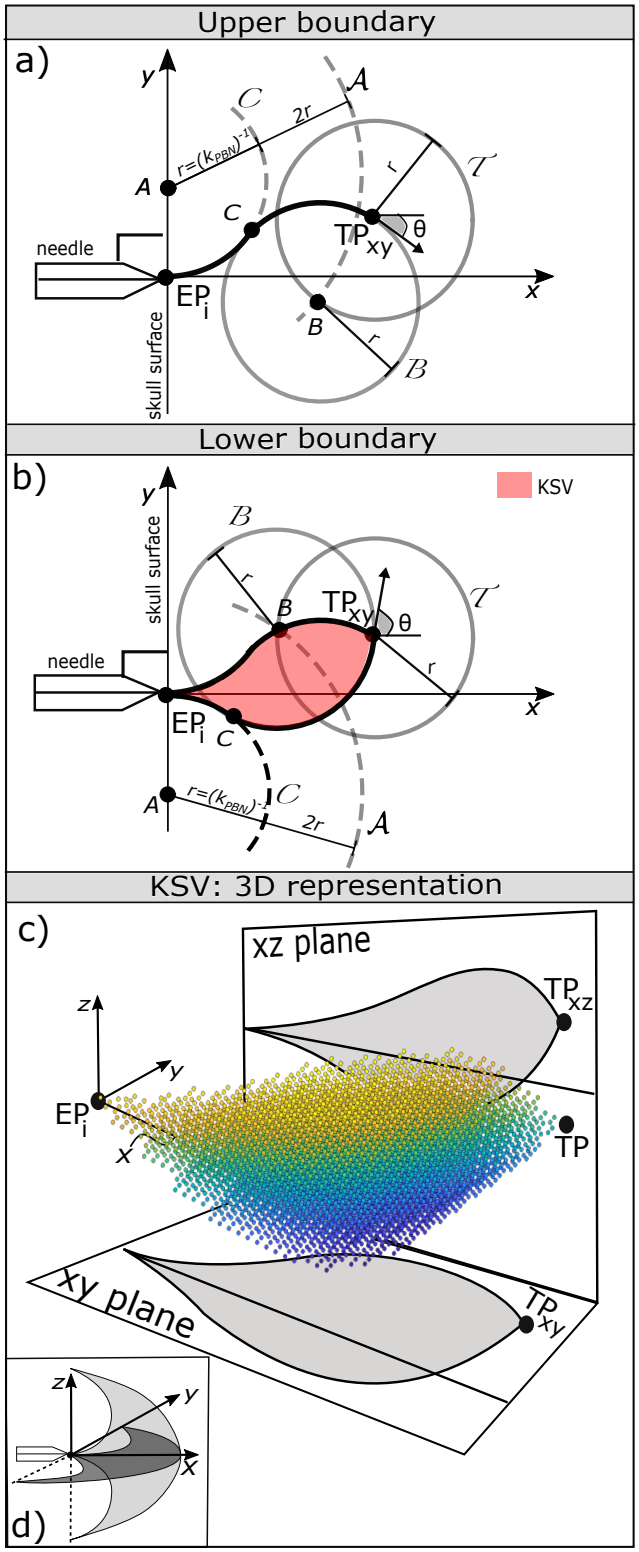


Fig. 2: A 2D projection of the Kinematics Search Volume (KSV) over the *xy*-plane. The insertion happens at the entry point (EP_i) along a direction perpendicular to the skull surface. In a) and b) the geometric relationships that allow the definition of the upper and lower boundary are represented, respectively. The KSV projection is highlighted in light red. In c), the 3D representation of the KSV is reported as a point cloud of samples. The projections of the KSV on the *xz* and *xy* planes are also shown. A color-code, function of the z coordinate, is used to depict the samples to help the understanding of the KSV as a 3D object. In d), the areas where the projection of TP on the *xy* and *xz* planes should lie in order for the method to be applicable are reported, depicted respectively in dark and light gray.

the risk of slipping during the skull drilling, an entry direction orthogonal to the skull surface is employed. This represents the direction of the *x*-axis for the local frame having as origin EP_i [33]. In the local frame, by projecting TP on the xy-plane (TP_{xy}), a 2D representation of the KSV is obtained (see Fig. 2a, b) where the *y* axis symbolizes the skull surface. The geometrical definition of the upper boundary of the KSV projection is shown in Fig. 2a and defined hereinafter.

A circle $\mathcal{A}$ centered in $\mathbf{A}(x_A = 0,\ y_A = r)$, where $r = \frac{1}{k_{PBN}}$ is defined as:

$$\mathcal{A}:\ x^2 + (y - y_A)^2 = (2r)^2$$

the point **B** is defined as the intersection point between $\mathcal{A}$ and $\mathcal{T}$ with the lower *y* value, being $\mathcal{T}$ a circle centered in TP_{xy}:

$$\mathcal{T}:\ (x - x_{TP_{xy}})^2 + (y - y_{TP_{xy}})^2 = r^2$$

A point **C** is defined as the tangent point between a circle $\mathcal{C}$ defined as:

$$\mathcal{C}:\ x^2 + (y - y_A)^2 = r^2$$

and the circle $\mathcal{B}$ centered in **B** :

$$\mathcal{B}:\ (x - x_B)^2 + (y - y_B)^2 = r^2$$

The upper boundary of the KSV projection is defined as a third order polynomial passing though the points EP_i, **C** and TP_{xy}.

As reported in Fig. 2b , the lower boundary of the KSV projection can be defined by considering a point **A** ($x_A = 0$, $y_A = -r$) and as point **B** the intersection point between $\mathcal{A}$ and $\mathcal{T}$ having the higher *y* value.

With similar considerations it is possible to define the boundaries of the KSV projection in the *xz*-plane.

In 3D, starting from the local frame at EP_i, the KSV can be thus defined as the volume enclosing all the uniformly-sampled points of the workspace confined within the lower and the upper boundaries of the KSV projections in the *xy* and *xz* planes (see Fig. 2c). These points are considered as feasible for computing kinematically acceptable solutions for the planning problem. This method is applicable when the projections of TP on the *xy* and *xz* planes are within the areas depicted in Fig. 2d. This implies a maximum insertion length that, considering the radius of curvature of the PBN, is equal to $r(1 + \sqrt{3}) = 19,5$ cm for a pure straight insertion and $2r = 14,2$ cm for a pure curvilinear insertion which, in the present case, is able to cover the entire working volume.

The planning is performed similarly to [11]: in a random fashion, the samples included in the KSV_i are sequentially provided to the planner. This represents a deviation from the way sampling-based planners are normally used, which entails a sampling of the workspace which do not results from any uniform distribution. Nonetheless, the fine resolution of $1\,\mathrm{mm}^3$ of the uniformly discretized workspace matches the resolution of the obstacles and, in thus, is considered as a reasonable approximation of a pure random sampling.

The planner builds a tree, composed by vertexes (**P**) corresponding to samples in the free space of KSV_i. Linear edges are used to connect adjacent vertexes. When a new sample is probed, the vertex that features the shortest path to EP_i

is identified and, as the obstacle clearance of the new edge is verified, the connection is made. The collision check is carried out considering a minimum distance from obstacles equal to half the PBN diameter for guaranteeing the practicability of the edge. The tree keeps evolving following an RRT* approach until a piece-wise linear pathway able to connect the EP_i to the TP is found. A first raw path, sol_1^i, is thus defined as a sequence of vertexes:

$$\{\mathrm{sol}_1^i\} = \{\mathbf{P}_{1,k}^i \in \mathbb{R}^3,\ k = 1, ..., N_{1,\mathbf{P}}^i\}$$

where $\mathbf{P}_{1,1}^i = \mathrm{EP}_i$ and $\mathbf{P}_{1,N_{1,\mathbf{P}}^i}^i = \mathrm{TP}$. The length ($l_{pw}^{sol_j^i}$) is considered as an index of goodness for a generic piece-wise linear solution sol_j^i:

$$l_{pw}^{sol_j^i} = \sum_{k=1}^{N_{1,\mathbf{P}}^i - 1} ||\mathbf{P}_{j,k+1}^i - \mathbf{P}_{j,k}^i||$$

It represents the sum of the distances between consecutive $\mathbf{P}_{j,k}^i$ in the sequence from the EP_i and the TP.

Every time a new sample is provided to the planning algorithm, the planner verifies whether the new sample allows to define $\mathrm{sol}_{\mathrm{f}\neq\mathrm{j}}^{\mathrm{i}}$ featuring a shorter length. If so, the minimum path length is updated with $l_{pw}^{sol_f^i}$ and, although featuring a longer path, solution sol_j^i is not discarded but stored for the next steps.

This process ends when a pre-set maximum number of iterations is reached (N_{RRT*}^{max}) or the number of computed $\mathrm{sol}_{\mathrm{j}}^{\mathrm{i}}$ reaches a predefined threshold (N_{sol}^{max}).

C. Path approximation and optimization

The objective of this step is to generate smooth paths based on the previously computed waypoints. We employed an Evolutionary Optimization Procedure (EOP) to fine-tune the weights of Non-Uniform Rational Beta Splines (NURBS), used to represent smooth trajectories.

Each $\mathrm{sol}_{\mathrm{j}}^{\mathrm{i}}$ defined in the previous section needs to:

1) be smoothed, to comply with the $\mathcal{C}^2$ continuity required by the PBN;
2) be checked for the obstacle clearance;
3) have minimum length.

Additionally, they have to comply with the maximum curvature achievable by the needle (k_{PBN}). The EOP, run for each $\mathrm{sol}_{\mathrm{j}}^{\mathrm{i}}$ and used for tuning the NURBS parameters, has demonstrated to be able to efficiently generate a smooth, obstacle avoiding and curvature-constrained path for non-holonomic robot, featuring minimal length and minimal variations of curvature [34].

A scheme of the EOP is reported in Fig. 3.

1) Population initialization: a general p^{th}-degree NURBS for $\mathrm{sol}_{\mathrm{j}}^{\mathrm{i}}$ is defined in parametric form as:

$$C(u)_j^i = \frac{\sum_{k=1}^{N_{j,\mathbf{P}}^i} B_{k,p}(u) w_{j,k}^i \mathbf{P}_{j,k}^i}{\sum_{k=1}^{N_{j,\mathbf{P}}^i} B_{k,p}(u) w_{j,k}^i}$$

where $\mathbf{P}_{j,k}^i$ are the control points, $w_{j,k}^i$ are the weights linked to each $\mathbf{P}_{j,k}^i$ and $B_{k,p}(u)$ are the pth-degree B-spline basis

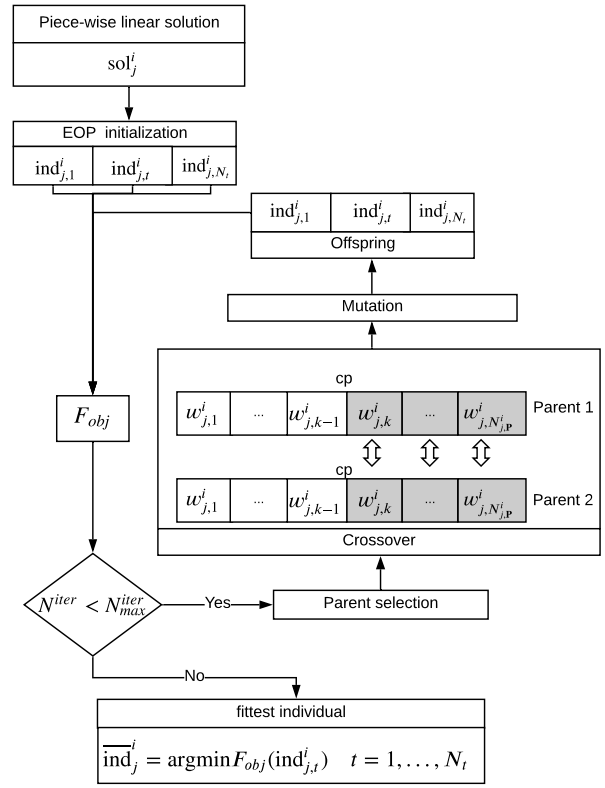


Fig. 3: In the diagram, the steps composing the EOP are depicted. The initial piece-wise linear solution (sol_j^i) is provided to the EOP, and the initial population of individuals is generated by providing random values to $w_{j,k,t}^i$. F_{obj} is then run and the parent selection is carried out es explained in Sec. III-C3. Then the crossover (Sec. III-C4) and the mutation (Sec. III-C5) happen, according to each specific probability. The population of individuals is then updated. The process continues until a predefined number of iteration (N_{max}^{iter}) is achieved and the fittest individual ($\overline{\mathrm{ind}}_j^i$) is returned as output.

functions defined on $u \in [0, 1]$. If the weight $w_{j,k}^i$ coupled to a control point $\mathbf{P}_{j,k}^i$ is moved, this affects only a portion of $C(u)_j^i$, allowing the local shape control: increasing (decreasing) the magnitude of $w_{j,k}^i$ pulls (pushes) the curve closer to (away from) $\mathbf{P}_{j,k}^i$. For further details about NURBS, the reader is referred to [35].

The EOP generates a primitive population of NURBS from $\mathrm{sol}_{\mathrm{j}}^{\mathrm{i}}$. Each NURBS is referred as "individual" (ind) and has the vertexes $\mathbf{P}_{j,k}^i \in \mathrm{sol}_j^i$ as control points. By randomly initializing the weights $w_{j,k}^i$ associated to $\mathbf{P}_k \in \mathrm{sol}_j$, the individuals in the population of NURBS are obtained:

$$\{\mathrm{ind}_{j,t}^i\} = \{\mathbf{P}_{j,k}^i \in \mathbb{R}^3, w_{j,k,t}^i \in \mathbb{R}\}$$

with $k = 1, ..., N_{j,\mathbf{P}}^i$ and $t = 1, ..., N_t$, where N_t represents a constant, pre-set number of individuals in the population.

2) Objective function: the variable $u \in [0, 1]$ used to define each $\mathrm{ind}_{j,t}^i$ in parametric form undergoes a discretization, as explained in Appendix A.

An objective function F_{obj} is defined, which is used by the EOP to rank the performance of each $\mathrm{ind}_{j,t}^i$, as in [34]:

$$F_{obj}(\mathrm{ind}_{j,t}^i) = \alpha \cdot l(\mathrm{ind}_{j,t}^i) + \beta \cdot \#P_o + \gamma \cdot \#P_c + \delta \cdot SD$$

which minimizes:

- the length l: the integral of the derivative of $\mathrm{ind}^i_{j,t}$ over its length, calculated as in [35]:

$$l(\mathrm{ind}^i_{j,t}) = \int_{EP_i}^{TP} \|\mathrm{ind'}^i_{j,t}(u)\| du$$

- $\#P_o$: the number of points $\in$ $\mathrm{ind}^i_{j,t}$ intersecting an obstacle:

$$\{P_o\} : \{\mathrm{ind}^i_{j,t} \cap \Omega_{obs}\}$$

  where $\{\Omega_{obs}\} \subset \mathrm{KSV_i}$ is the set of 3D points representing the obstacle space.
- $\#P_c$: the number of points $\in \mathrm{ind}^i_{j,t}$ to which a curvature (as the second derivative of $\mathrm{ind}^i_{j,t}$) larger than k_{PNB} is associated:

$$\{P_c\} = \{\mathrm{ind''}^i_{j,t} > k_{PBN}\}$$

- SD: the standard deviation of the curvature of $\mathrm{ind}^i_{j,t}$:

$$SD = \sqrt{\frac{1}{N_{samp}} \sum^{N_{samp}} (\mathrm{ind''}^i_{j,t} - \mu_{\mathrm{ind''}^i_{j,t}})^2}$$

  where $\mu_{\mathrm{ind''}^i_{j,t}}$ and N_{samp} are the mean value and the number of samples of $\mathrm{ind''}^i_{j,t}$ that depend upon the discretization of $u \in [0, 1]$. This parameter is considered as a quantity to minimize in order to reduce the variation of the curvature and, consequently, the control effort, as in [34].

α, β, γ and δ values are reported in Table I.

3) Parent selection method: the linear Rank-based Roulette Wheel method [36] is used for parent selection. This selection strategy prevents from the risk to fall into a local minimum during the EOP. At any new generation, it assigns to each $\mathrm{ind}^i_{j,t}$ in the population a probability p to be selected as a parent:

$$\mathrm{p}(\mathrm{ind}^i_{j,t}) = \frac{rank(\mathrm{ind}^i_{j,t})}{\sum_{t=1}^{N_t} rank(\mathrm{ind}^i_{j,t})}$$

where *rank* is defined as:

$$rank(\mathrm{ind}^i_{j,t}) = 2 - SP + \frac{2(SP-1)(\mathrm{pos}^i_{j,t} - 1)}{N_t - 1}$$

where $\mathrm{pos}^i_{j,t}$ is the hierarchical position of $\mathrm{ind}^i_{j,t}$ in the population according to F_{obj} and SP represents the "selective pressure", a favorable bias given to individuals having low F_{obj}. With $SP = 1$, all the individuals of the population have the same rank and thus the same probability to be chosen. If $SP = 2$, high-performing parents (the ones with low F_{obj}) have a high rank and thus a higher probability to be selected with respect to less-performing ones.

4) Crossover: crossover consists in switching part of the weights between two parent individuals. In this work, a single-point crossover is used, with a cutting point (*cp*) randomly selected at each iteration. The crossover can happen according to a predefined probability p_{cross}. Due to its random nature, crossover can lead to offsprings that do not meet the condition $\{P_o\} \wedge \{P_c\} = \emptyset$. This possibility is envisaged by the algorithm, and regardless if such a situation happens or not, the offspring are stored.

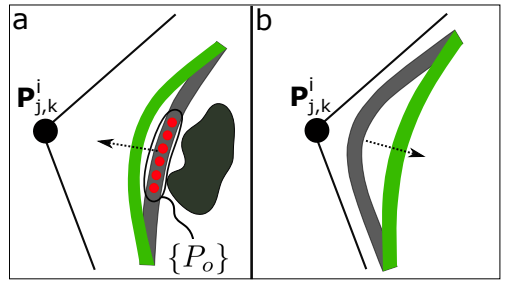


Fig. 4: Mutation algorithm. In *a*), part of the curve results too close to an obstacle (red points), i.e $\{P_o\} \neq \emptyset$, so the weight of $\mathbf{P}^i_{j,k}$ controlling that part of the curve is decreased until the obstacle avoidance is obtained (green line). In *b*), the weight $\mathbf{P}^i_{j,k}$ is decreased for smoothing the curve.

5) Mutation: mutation can happen at each iteration of the EOP according to a predefined probability p_{mut}. To avoid the definition of an unfeasible path, we implemented a controlled mutation method similar to [34], using a supervised adjustment of the weights in the new individual of the offspring as presented in Appendix B and shown in Fig. 4. If part of the curve results too close to an obstacle, i.e $\{P_o\} \neq \emptyset$, the weight of $\mathbf{P}^i_{j,k}$ that controls that part of $\mathrm{ind}^i_{j,t}$ is increased until the condition $\{P_o\} = \emptyset$ is achieved (Fig. 4a). If part of the curve shows $\{P_c\} \neq \emptyset$, the weight is decreased until all the curve points meet the curvature limit. Additionally, whenever possible, the weights of $\mathbf{P}^i_{j,k}$ that controls that part of $\mathrm{ind}^i_{j,t}$ is decreased in order to smooth the curve but without voiding the condition $\{P_o\} \neq \emptyset$ (Fig. 4b). When, as the result of the mutation, the new individual shows a higher value of objective function than the original one, the new individual is rejected and the original one restored.

The EOP stops when the number of generations (N^{iter}) reaches a predefined threshold (N^{iter}_{max}). For each sol^i_j, the individual of the NURBS population that features the lower F_{obj} while guaranteeing the condition $\{P_o\} \wedge \{P_c\} = \emptyset$ is stored as $\overline{\mathrm{ind}}^i_j$:

$$\overline{\mathrm{ind}}^i_j = \mathrm{argmin}\, F_{obj}(\mathrm{ind}^i_{j,t}) \quad t = 1, ..., N_t$$

If no individual meets the condition $\{P_o\} \wedge \{P_c\} = \emptyset$, the algorithm returns no solution for the specific sol^i_j.

D. Exhaustive search for the best path

In this section, the cost function used to perform an exhaustive study of the paths discovered in the previous section is presented.

In order to define the best path for the specific EP_i, the algorithm performs an evaluation over $\overline{\mathrm{ind}}^i_j, \forall j$ through a cost function F_{cost} defined as:

$$F_{cost}(\overline{\mathrm{ind}}^i_j) = a\frac{1}{d_{min} + \bar{d}} + b\frac{l - l_{min}}{l_{min}} + c\frac{k_{max}}{k_{PBN}}$$

where d_{min} represents the closest distance from an obstacle calculated over the whole length l of $\overline{\mathrm{ind}}^i_j$, $\bar{d}$ is the mean value of the distance from the obstacles and l_{min} the Euclidean distance between the EP_i and the TP. The maximum curvature achieved along the path is $k_{max} = \max(\overline{\mathrm{ind''}}^i_j(u))$. This value

is taken into consideration in computing the overall cost for the path $\overline{\text{ind}}^i_j$ as it has been shown experimentally that larger control errors are associated to a high degree of curvature [37].

With $\overline{\overline{\text{ind}}}^i$ we refer to the best curvilinear path corresponding to the one among all the $\overline{\text{ind}}^i_j$ that has the lower value of F_{cost}. A representation of the EOP and the exhaustive search for the best path is reported in Fig. 5.

As the EOP and the exhaustive search is performed for all the EP_i belonging to the entry area, a number of best pathways (that, if the planning does not fail, correspond to the number of EP_i, *N*) are provided to the surgeon as output of the algorithm.

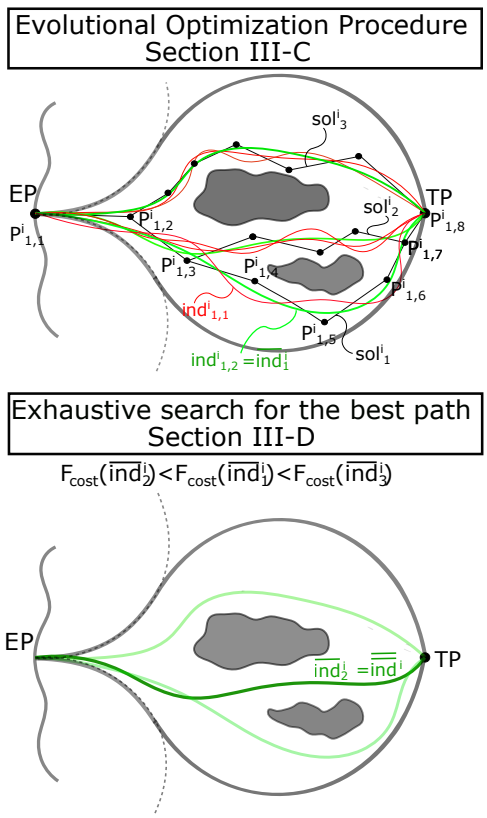


Fig. 5: In the uppermost figure, the EOP is presented. Three piece-wise linear solutions are depicted in black ($\text{sol}^i_{1,\ldots,3}$), along with multiple curvilinear approximations ($\text{ind}^i_{j,t}$) in red. The best solution for each sol_j is reported in green as $\overline{\text{ind}}^i_j$. In the picture underneath, referred as "Exhaustive search for the best path", the cost function described in Section III-D is implemented to define, among the three $\overline{\text{ind}}^i_{1,\ldots,3}$, the best one: $\overline{\overline{\text{ind}}}^i$.

IV. Experimental protocol

A. Input dataset

The dataset used for the experimental trials consists in one brain reconstructed from Magnetic Resonance Imaging acquisitions performed at the Excellence Centre for High Field MR (CERMAC), Vita-Salute San Raffaele University, Milano, Italy.

The anatomical obstacles were identified and segmented in the patient's dataset using 3D Slicer© (www.slicer.org) [38] and consist in blood vessels, ventricles, thalamus and globus pallidus. The curvature of the brain cortex was computed using Freesurfer [39].

Ten entry areas of 10mm-radius (a dimension consistent with the one employed in [40]), were defined on the cortical surface resembling possible entry areas in clinical practice, 5 on the left and 5 on the right hemisphere (Fig. 6a). Ten TPs were set (each one associated to one entry area) in different locations within the brain volume. For each entry area, the EPs are defined according to Section III-A, and this resulted in a total of 172 EPs (17.2 for each entry area, on average). The mean Euclidean distance between EP and TP is calculated as equal to 78.02mm.

The entire workflow described in Fig. 1 was run over all the entry areas. All the relevant parameters used in the simulations for the presented method are reported in Table I. Path planner parameters (N^{max}_{RRT*}, $N^{max}_{\mathbf{sol}}$) were empirically defined as the ones that, on average, allow to discover the shortest path in bounded computational time (see Section III-B). For the EOP, the number of individuals (N_t), the number of generations (N^{iter}_{max}) as well as the crossover and mutation probabilities (p_{cross}, p_{mut}) were adapted to values close to the ones used in [34].

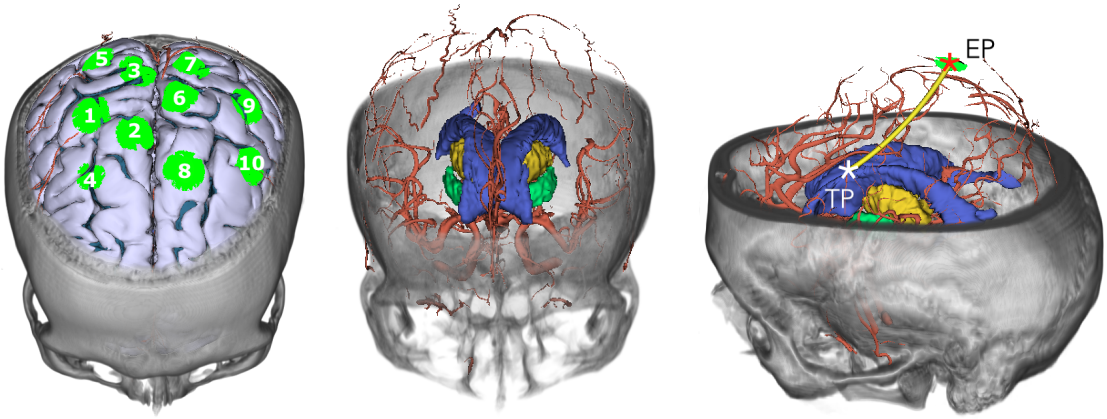
B. Simulations and data analysis

A first test involved the comparison of the proposed solution with two other sample-based planning methods: the previous version of the planner [11] referred as "ellipsoidal", and the RG-RRT proposed by Patil *at al.* [21]. The latter consists in a combination of the RRT-based search and a reachability-guided sampling heuristic.

The comparison between the different algorithms was carried out in terms of mean distance from anatomical obstacles ($\bar{d}$), the normalized path length ($\hat{l}$) as the percentage the path length exceeds the Euclidean distance between the EP_i and the TP, the minimum distance along the entire path from the closest anatomical obstacle (d_{min}) and the value of F_{cost}. The maximum curvature (k_{max}) was also checked as a critical parameter which relates to a higher risk of inaccuracies in the control of the PBN during the trajectory tracking. Additionally, the failure rate (*FR*) was evaluated, defined considering the number of best solution computed by the planning algorithm ($\#\{\overline{\overline{\text{ind}}}\}$) and the number of original EPs defined over the entry area ($\#\{\text{EP}_i\}$), such that:

$$FR = \frac{\#\{\overline{\overline{\text{ind}}}\}}{\#\{\text{EP}_i\}}$$

As maximum curvature value, k_{PBN} was used in the simulation and the nominal catheter thickness was considered for assessing obstacles avoidance, as reported in Table I. For all the three algorithms, the research was biased by providing the TP as new sample with a probability of 10%. When analyzing a new sample, the maximum Euclidean distance between the sample and the closest vertex in the tree was 40mm for the KSV and the ellipsoidal method. No maximum Euclidean distance was considered for arc reachability in the RG-RRT. The three methods used the same original workspace with the same density of samples. The KSV approach and the ellipsoidal method had the same pre-set maximum number of iterations (N^{max}_{RRT*}) and solutions $N^{max}_{\mathbf{sol}}$, the same number of



(a) (b) (c)

Fig. 6: In a) a picture of the 10 entry areas defined on the brain cortex for the scope of the test (5 on the left and 5 on the right hemisphere) is presented. In b), the anatomical obstacles considered in the planning phase are depicted in different colors: ventricles (blue), thalamus (yellow), globus pallidus (green) and blood vessels (red). In c), an example of a planned curvilinear path is proposed (sharp yellow). The entry and target points are labelled respectively as EP and TP. The entry area is also shown in sharp green around the EP.

TABLE I: Parameters used in the experimental setup. From the left to the right, the PBN diameter and maximum degree of curvature are reported, followed by a the pre-set maximum number of raw path (N_{sol}^{max}) allowed at each iteration of Section III-B. The EOP is defined by the initial population size N_t. The number of offspring (N_{max}^{iter}), the cross-over and mutation probabilities (P_{cross} and P_{mut}) as well as the value assigned to the selective pressure (SP) are also reported. Lastly, the values of the weight used in the Objective and the Cost functions are shown.

PBN Parameters		**Path planning**		**EOP settings**					**Objective Function**				**Cost Function**		
Diameter [mm]	$k_{PBN}[mm^{-1}]$	N_{RRT*}^{max}	N_{sol}^{max}	N_t	N_{max}^{iter}	p_{cross}	p_{mut}	SP	α	β	γ	δ	a [mm]	b	c
2.5	0.014	200	5	20	50	0.5	0.1	1.8	100	1	1	100	0.01	0.5	0.5

maximum iteration N_{RRT*}^{max} was used also for the RG-RRT. In case the algorithms reached the maximum number of iterations without finding a solution for the planning query, the search stopped and an error was returned. As foci for the ellipsoidal method, the EP and TP were considered. Variation in the ellipsoidal shape happend every N_v new vertexes were added to the tree without reaching the TP, where $N_v = 5{*}q{*}N_{reshape}$ with $N_{reshape}$ representing the current number of reshape iterations and $q = 1.5$ an increasing factor. Ellipsoidal shape variation is carried out by enlarging the minor axis by a factor q, starting from a value of 10mm. An upper limit of 1000 samples is considered for all algorithms before quitting the search. The results were analysed through a Friedman non-parametric statistical test ($p < 0.05$), followed by a post-hoc Wilcoxon matched pairs test ($p < 0.016$, Bonferroni correction).

The performance of the three algorithms were compared also in terms of computational time. Beside the overall time required for the entire path computation, also the one required to find the initial set of piece-wise linear raw paths (sol_j^i) was measured for the KSV and the ellipsoidal method. This additional analysis aimed at assessing the time improvements associated to confining the path planning within a subset of the original workspace where kinematically-feasible solutions can lie, and was not performed for the RG-RRT as this method does not require an initial piece-wise linear raw paths computation. A pairwise comparison was performed on the obtained time results via Wilcoxon matched pairs tests ($p < 0.05$).

Fourthly, a test was conducted to compare the performance of the proposed method with respect to an optimal planner. For this scope, an A* optimal planner was implemented, having the following heuristic function:

$$h(n) = w_1^h \left(\frac{l_{TP}(n)}{l_{min}} \right) + w_2^h \left(\frac{d_{max} - d(n)}{d_{max}} \right)$$

where $l_{TP}(n)$ and l_{min} are the Euclidean distances between the node n and TP, and between EP and TP, respectively. The parameter d_{max} represents the larger value measured in the distance map and $d(n)$ the distance between n and the closer obstacle. The two coefficients $w_{1,2}^h$ were defined empirically as equal to 1 and 0.2, respectively. This test was performed on one EP, corresponding to the centre of entry area n.1 in Fig. 6 and five repetitions of the KSV-based planning solution were run. As minimum distance from the obstacles, the nominal catheter thickness is considered. As terms of comparison between the A* and the KSV-based planner, the normalized path length, the minimum and mean distance from obstacles and the computational time were considered. The maximum curvature was overlooked in the comparison as A* provides piece-wise linear paths.

All tests were performed using MATLAB® R2019a, on a MacBook Pro (MacOS 10.14.6, 2,7 GHz Intel Core i5, 8 GB of RAM) and the same parameterization was used in all the tests.

V. Results

Results from the comparison between the method herein presented with the RG-RRT and the ellipsoidal solution are

presented in Fig. 7 and Table II. In Fig. 7a, the highest value of curvature reached along the path (k_{max}) shows respectively a median value of $0.01\,\mathrm{mm}^{-1}$ (radius of curvature *r*=100 mm), $6.2 \times 10^{-3}\,\mathrm{mm}^{-1}$ (*r*= 161.3 mm) and $6 \times 10^{-4}\,\mathrm{mm}^{-1}$ (*r*= 1667 mm) for the RG-RRT, ellipsoidal and KSV algorithm, respectively ($p < 0.01$).

The minimum distance from anatomical obstacles (d_{min}) resulted in a median value of 0.63 mm for the RG-RRT, 0.34 mm for the ellipsoidal method and 1.9 mm for the KSV (Fig. 7b) while the mean distance ($\bar{d}$) showed a median value of 6.06 mm for the RG-RRT, 7.26 mm for the ellipsoidal method and 9.1 mm for the KSV ($p < 0.01$) (Fig. 7c).

With regards to the normalized path length ($\hat{\mathrm{l}}$), its median values were equal to 2.92 %, 2.35 % and 1.19 % for the RG-RRT, ellipsoidal method and KSV, respectively ($p < 0.01$) (Fig. 7d).

For the RG-RRT, F_{cost} demonstrates a median value of 0.35 , while a value of 0.215 was found for the ellipsoidal method and of 0.017 for the KSV ($p < 0.01$) (Fig. 7e).

For the failure rate *FR*, its median value for the RG-RRT was 33.7 % while for the ellipsoidal method was 21.1 % and for the KSV was 5.2 % ($p < 0.05$) (Fig. 7f).

Results in terms of computational time are shown in Table II, where the difference between the KSV and the other two algorithms (RG-RRT and ellipsoidal) is reported. In the estimation of the initial raw paths, the ellipsoidal approach showed a median value of 96.75 sec while the KSV resulted in a median computational time of 4.3 sec ($p < 0.01$). The entire planning process results faster for the RG-RRT, which was able to compute a path in less than 30 sec, on average. The less performing one was the ellipsoidal method (up to 243 sec), while the time required by the KSV was lower than a minute.

Results from the comparison of the presented solutions with the A* are presented in Table III, where the difference in terms of normalized path length, minimum and mean distance from obstacles and computational time are reported.

TABLE II: Results in terms of computational time are shown for the proposed solution, the ellipsoidal method and the RG-RRT as 25th, 50th and 75th percentiles. The computational cost of the initial piece-wise linear path computation for the KSV and the ellipsoidal method is reported (not applicable for the RG-RRT). The total computational cost for the estimation of the best path is also reported. Statistical significance effect of the type of search space has been found through a Wilcoxon matched pairs test ($p < 0.05$).

Computational time						
	Piece-wise planning			**Total computation time**		
	25th	50th	75th	25th	50th	75th
RG-RRT [sec]	*na*	*na*	*na*	7.1	8.4	24.3
ellipsoidal [sec]	89.6	96.7	107.9	102.4	140.4	243.0
KSV [sec]	4.2	4.3	5.3	16.5	17.6	44.7

VI. Discussion

This work presents a planning solution for computing pathways suitable for being performed by a flexible catheter in neurosurgical applications and used in combination with bespoke control and actuating systems so that they can be used as surgical trajectories.

TABLE III: Performance test between A* approach and the presented solution in term of normalized path length, minimum and mean distance from obstacles and computational time as 25th, 50th and 75th percentiles.

A* - KSV performance comparison				
	A*	KSV		
		25th	50th	75th
$\hat{\mathrm{l}}$ [%]	10.6	0.1	0.2	0.3
d_{min} [mm]	1.32	0.03	0.25	0.39
$\bar{d}$ [mm]	7.92	5.43	5.94	6.17
comp. time [sec]	270.6	34.0	45.2	55.5

A smart redefinition of the search space, i.e. the Kinematic Search Volume (KSV), considers the curvature limit of the PBN and ignores those part of the working domain that will give rise to unfeasible paths. This allows the algorithm to save time. By looking at Table II, where the KSV is compared to the ellipsoidal search used in [11], a significant decrease of the computational time ($\sim$ 20 times) can be noticed in the computation of the initial piece-wise linear solutions. Although slower than the RG-RRT, our solution results faster than the ellipsoidal method also in the overall path computation. Furthermore, a new approximation method based on the use of NURBS provides the $\mathcal{C}^2$ continuity required by the PBN and the local control of the path. This, through a bespoke Evolutionary Optimization Procedure (EOP), allows to fine-tune each solution according to multiple optimization objectives. The combination of the KSV and the EOP provides a high level of flexibility allowing the variation of the weights associated to the control points in a way to privilege some aspects (e.g. the distance from obstacles) more than others and at a level that was not achievable by other approaches like those based on inverse kinematics [13] or potential fields methods [15] that have, on the other hand, the benefit of allowing a quasi-real-time path computation. The sampling-based planning performed within the KSV, followed by the path optimization performed by the EOP, even if it can not be expected assuredly to find the global optimum to the planning problem, can generate an excellent quasi-optimal solution [36].

When compared with two other sample-based planning solutions [11], [21], our method was able to outperform them in two relevant parameters: the distance from anatomical obstacles and the normalized path length, as shown in Fig. 7. The obtained path was also smoother, which represents a good starting point in the view of reducing the possible control error in the trajectory tracking. Additionally, the obtained overall cost resulted consistently lower in our solution with respect to the other two algorithms with a lower risk of failure in the path search: the KSV algorithm failed respectively 4 and 6 times less frequently than [11] and [21].

On the other hand, the method herein proposed is limited to pre-operative applications as its computational time does not permit re-computing a path in real-time in case errors occur during the needle insertion. Two possible approaches can be implemented for increasing the responsiveness of the algorithm: at first, the introduction of a bidirectional-RRT search [20] would speed up the search for the raw path (sol_j^i) by letting the tree evolving from the EP_i and the TP at the

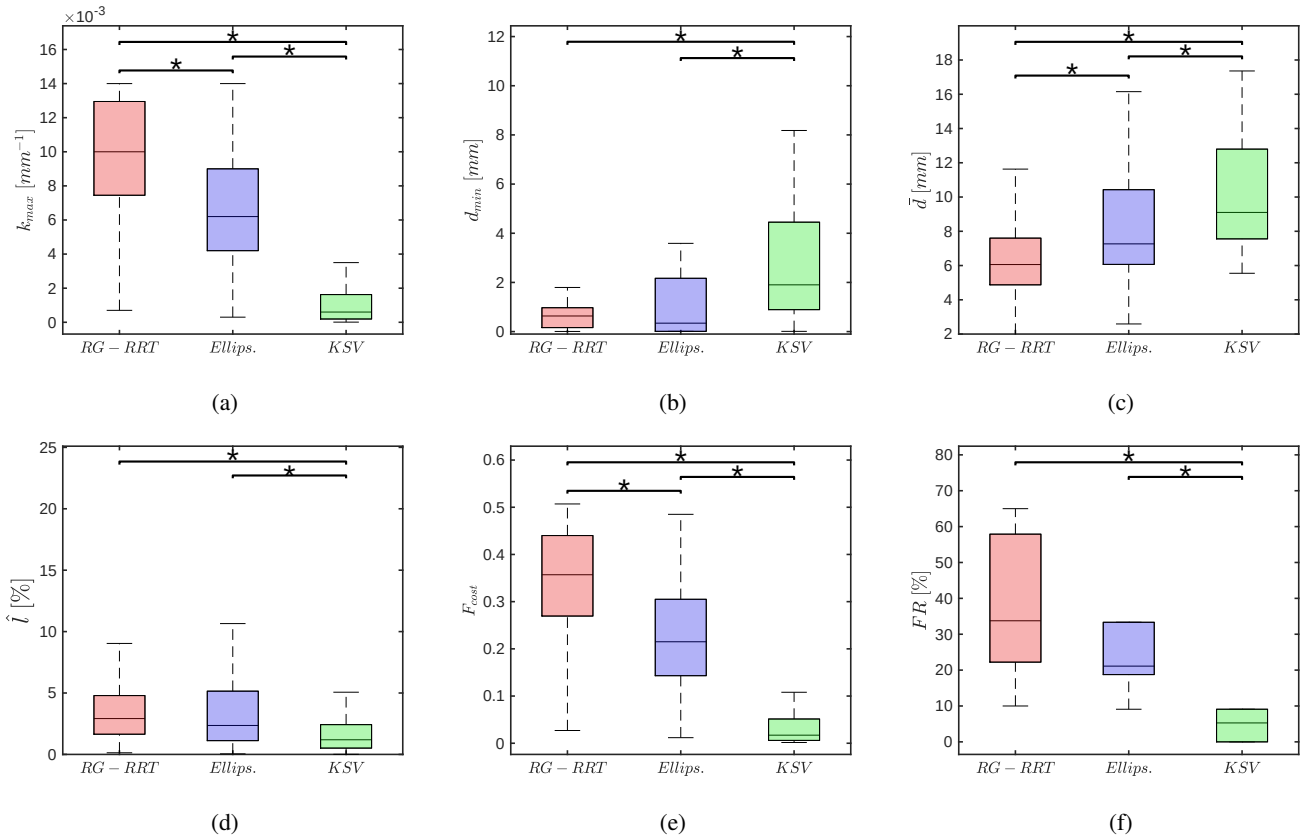


Fig. 7: Results from the comparison between the presented solution (KSV) and other two methods from the literature: the ellipsoidal planner ("*Ellips.*") and the RG-RRT. The maximum curvature (k_{max}), the minimum and mean distance from anatomical obstacles (d_{min} and $\bar{d}$), the normalized path length ($\hat{l}$), the overall cost (F_{cost}) and the failure rate (*FR*) are reported respectively in *a,b,c,d,e* and *f*. Statistical significance between different algorithms is highlighted ($*$, $p < 0.05$).

same time, furthermore the EOP may be further boosted by reducing the computational cost due to the mutation of the individuals. We aim at exploring also the use of reinforcement learning for the computation of the raw path.

From the comparison with an optimal planner, it was evidenced that the A* algorithm was able to provide paths that are safer in terms of minimum and mean distance from obstacles at the expense of an increased path length and a longer computational time, as expected from a graph-based planning method in 3D scenarios. However, it has to be noted that, differently from the planner herein presented, the A* algorithm computes piece-wise linear paths. These paths are less complex to elaborate as they do not meet the needle curvature constraint nor the $\mathcal{C}^2$ continuity and this aspect makes the comparison less meaningful.

The quasi-optimal path estimated by the proposed solution assumes a kinematic model of motion for the PBN during insertion and a proper control system exist. In [8], an adaptive controller combined with a kinematic model of the PBN needle in Parallel Transport Frame are proposed. Test performed in-vitro demonstrated the ability of the controller method to perform curvilinear paths, provided that they meet the PBN curvature limit. The tests were performed over multiple curvilinear paths comparable to the one estimated in this work in terms of insertion length and curvature. This implies that the present solution can be considered as a feasible path planner for such a control method. Further in-vitro tests will be conducted using paths estimated by the present solution in order to evaluate the system tracking error.

VII. Conclusions

The paper describes a solution for the definition of feasible pathways for robotically-actuated flexible needles in a neurosurgical scenario, which represents a typical example of a dense environment characterized by narrow spaces.

A smart redefinition of the workspace based on the maximum curvature of a neurosurgical needle (in the present study, the Programmable Bevel-Tip Needle [7]) is used to limit the sample-based path search within a confined region, the so-called Kinematic Search Volume, where feasible solutions lie. This, combined to a path optimization based on a bespoke evolutionary optimization procedure, results in providing a significant improvement in the performance guaranteeing a higher obstacle avoidance and a reduced path length, also keeping the computational time consistent with standard pre-operative planing algorithms. Our planning method can also reduce the failure rate in finding a path for a specific query, augmenting the pool of possible solutions where to look for the best path. This lies the foundation for further enhancements in the quality of the estimated paths and for a real-time path computation through the reduction of the computational cost.

APPENDIX

A. Discretization of the independent variable u

The independent variable $u \in [0, 1]$ used to define a NURBS curve in parametric form is discretized in order to compute the parameters of Section III-C2. This is done according to a discretization parameter ρ determined empirically and linked to the number $N^i_{j,\mathbf{P}}$ of control points $\mathbf{P}^i_{j,k} \in \mathrm{sol}^i_j$, such that:

$$\rho = \begin{cases} 0.1 & \text{if} \quad N^i_{j,\mathbf{P}} = 2 \\ 0.01 & \text{if} \quad N^i_{j,\mathbf{P}} = 3 \\ 0.001 & \text{if} \quad N^i_{j,\mathbf{P}} \geq 4 \end{cases}$$

B. Mutation algorithm

If $\{P_o\} \neq \emptyset$ (Line 2, Figure 4b), the set of control points $\{\mathbf{P}^{critic}_o\}$ that control parts of $\mathrm{ind}^i_{j,t}$ where the condition $\{P_o\} \neq \emptyset$ happens is defined (Line 3). For each $\mathbf{P}^i_{j,k} \in \{\mathbf{P}^{critic}_o\}$ (Line 4), the weight $w^i_{j,k,t}$ is increased (Line 8) to push $\mathrm{ind}^i_{j,t}$ away from the obstacle until the condition $\{P_{o,k}\} = \emptyset$ is reached (i.e. obstacle avoidance is achieved within the part of $\mathrm{ind}^i_{j,t}$ controlled by $\mathbf{P}^i_{j,k}$, Line 7). If the weight modification leads to exceeding k_{PBN} or to approaching another obstacle located in the same section controlled by $\mathbf{P}^i_{j,k}$ ($\#\{P_{c,k}\}^{new} > \#\{P_{c,k}\}$, Line 11), the weight variation (Δ) is reduced according to a "divide and conquer" approach, as in as [11], provided that the stop condition $\Delta/2 < \Delta_{thr}$ is maintained (Lines 12-15). If $\mathrm{ind}^i_{j,t}$ is feasible (i.e. $\{P_o\} = \emptyset$ and $\{P_c\} = \emptyset$, Line 16, Figure 4a), all $w^i_{j,k,t}$ with $k = 2, ..., N^i_{j,\mathbf{P}} - 1$ (i.e. all weights except the ones associated to EP_i and TP, Line 17) are decreased (Line 21) to smooth the curve and reduce its total length provided that the weight reduction does not generate obstacle collision and complies with a limit of flatness (k_{thr}), which correspond to stop conditions (Lines 20). In case of obstacle collision, a "divide and conquer" approach is used (Lines 25-28).

Algorithm 1 Mutation($\mathrm{ind}^i_{j,t}$)

1: $\{P_c\}, \{P_o\}$ = verify($\mathrm{ind}^i_{j,t}$)
2: **if** $\{P_o\} \neq \emptyset$ **then**
3: $\{\mathbf{P}^{critic}_o\}$ =controlPointsObs($\mathrm{ind}^i_{j,t}$,$\mathbf{P}^i_{j,k}$)
4: **for all**$\{\mathbf{P}^i_{j,k}\} \in \{\mathbf{P}^{critic}_o\}$ **do**
5: $\{P_{c,k}\}, \{P_{o,k}\}$ = verify($\mathrm{ind}^i_{j,t}, \mathbf{P}^i_{j,k}$)
6: reset(Δ)
7: **while** $\{P_{o,k}\} \neq \emptyset$ **do**
8: $w^i_{j,k,t}$= $w^i_{j,k,t} + \Delta$
9: update($\mathrm{ind}^i_{j,t}$)
10: $\{P_{c,k}\}^{new}\{P_{o,k}\}^{new}$= verify($\mathrm{ind}^i_{j,t}$)
11: **if** $\#\{P_{c,k}\}^{new} > \#\{P_{c,k}\}$ **or** $\#\{P_{o,k}\}^{new} > \#\{P_{o,k}\}$ **then**
12: $w^i_{j,k,t}$= $w^i_{j,k,t} - \Delta$
13: update($\mathrm{ind}^i_{j,t}$)
14: **if** $\Delta/2 < \Delta_{thr}$ **then** break
15: **else** $\Delta = \Delta/2$
16: **if** $\{P_c\} = \emptyset$ **and** $\{P_o\} = \emptyset$ **then**
17: **for all**$\{w^i_{j,k,t}, k = 2, ..., N^i_{j,\mathbf{P}} - 1$ **do**
18: $\{P_{o,k}\}$ = verify($\mathrm{ind}^i_{j,t}, \mathbf{P}^i_{j,k}$)
19: reset(Δ)
20: **while** $\{P_{o,k}\} = \emptyset$ **and** $\mathrm{ind}''^i_{j,t} > k_{thr}$ **do**
21: $w^i_{j,k,t}$= $w^i_{j,k,t} - \Delta$
22: update($\mathrm{ind}^i_{j,t}$)
23: $\{P_{o,k}\}^{new}$= verify($\mathrm{ind}^i_{j,t}$)
24: **if** $\{P_{o,k}\}^{new} \neq \emptyset$ **then**
25: $w^i_{j,k,t}$= $w^i_{j,k,t} + \Delta$
26: update($\mathrm{ind}^i_{j,t}$)
27: **if** $\Delta/2 < \Delta_{thr}$ **then** break
28: **else** $\Delta = \Delta/2$
return $\mathrm{ind}^i_{j,t}$

REFERENCES

[1] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum Robots for Medical Applications: A Survey," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1261–1280, 12 2015.

[2] 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. 1117–1122, 10 2010.

[3] V. Kallem and N. J. Cowan, "Image guidance of flexible tip-steerable needles," *IEEE Transactions on Robotics*, vol. 25, no. 1, pp. 191–196, 2 2009.

[4] K. B. Reed, A. Majewicz, V. Kallem, R. Alterovitz, K. Goldberg, N. J. Cowan, and A. M. Okamura, "Robot-assisted needle steering," *IEEE Robotics and Automation Magazine*, vol. 18, no. 4, pp. 35–46, 12 2011.

[5] D. Glozman and M. Shoham, "Image-guided robotic flexible needle steering," *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 459–467, 6 2007. [Online]. Available: http://ieeexplore.ieee.org/document/4252165/

[6] D. C. Rucker, B. A. Jones, and R. J. Webster, "A model for concentric tube continuum robots under applied wrenches," in *Proceedings - IEEE International Conference on Robotics and Automation*. IEEE, 5 2010, pp. 1047–1052.

[7] R. Secoli and F. R. Y. Baena, "Adaptive path-following control for bio-inspired steerable needles," in *Proceedings of the IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics*, vol. 2016-July. IEEE, 6 2016, pp. 87–93.

[8] R. Secoli, F. Rodriguez, and Baena, "Experimental validation of curvature tracking with a programmable bevel-tip steerable needle," in *2018 International Symposium on Medical Robotics, ISMR 2018*, vol. 2018-Janua. IEEE, 2018, pp. 1–6.

[9] 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," *International Journal of Computer Assisted Radiology and Surgery*, vol. 7, no. 4, pp. 517–532, 2012.

[10] S. Patil, J. Burgner, R. J. Webster, and R. Alterovitz, "Needle steering in 3-D Via rapid replanning," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 853–864, 2014.

[11] A. Favaro, L. Cerri, S. Galvan, F. R. Y. Baena, and E. De Momi, "Automatic Optimized 3D Path Planner for Steerable Catheters with Heuristic Search and Uncertainty Tolerance," in *Proceedings - IEEE International Conference on Robotics and Automation*, IEEE. IEEE, 5 2018, pp. 9–16.

[12] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.

[13] V. Duindam, J. Xu, R. Alterovitz, S. Sastry, and K. Goldberg, "Three-dimensional motion planning algorithms for steerable needles using inverse kinematics," *International Journal of Robotics Research*, vol. 29, no. 7, pp. 789–800, 6 2010.

[14] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," in *Autonomous Robot Vehicles*, I. J. Cox and G. T. Wilfong, Eds. New York, NY: Springer New York, 1986, pp. 396–404.

[15] P. Li, S. Jiang, J. Yang, and Z. Yang, "A combination method of artificial potential field and improved conjugate gradient for trajectory planning for needle insertion into soft tissue," *Journal of Medical and Biological Engineering*, vol. 34, no. 6, pp. 568–573, 2014.

[16] M. Likhachev, D. I. Ferguson, G. J. Gordon, A. Stentz, and S. Thrun, "Anytime Dynamic A*: An Anytime, Replanning Algorithm." in *ICAPS*,

2005, pp. 262–271.
[17] K. Leibrandt, C. Bergeles, and G.-Z. Yang, "Concentric Tube Robots: Rapid, Stable Path-Planning and Guidance for Surgical Use," *IEEE Robotics & Automation Magazine*, vol. 24, no. 2, pp. 42–53, 6 2017.
[18] J. J. Kuffner and S. M. La Valle, "RRT-connect: an efficient approach to single-query path planning," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2. IEEE, 2000, pp. 995–1001.
[19] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 6 2011.
[20] M. Jordan and A. Perez, "Optimal Bidirectional Rapidly-Exploring Random Trees," *Computer Science and Artificial Intelligence Laboratory*, 2013.
[21] S. Patil and R. Alterovitz, "Interactive motion planning for steerable needles in 3D environments with obstacles," *2010 3rd IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics, BioRob 2010*, pp. 893–899, 9 2010.
[22] C. Caborni, S. Y. Ko, E. De Momi, G. Ferrigno, and F. R. y Baena, "Risk-based path planning for a steerable flexible probe for neurosurgical intervention," in *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 6 2012, pp. 866–871.
[23] A. Kuntz, L. G. Torres, R. H. Feins, R. J. Webster, and R. Alterovitz, "Motion planning for a three-stage multilumen transoral lung access system," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem. IEEE, 9 2015, pp. 3255–3261.
[24] F. Liu, A. Garriga-Casanovas, R. Secoli, and F. Rodriguez y Baena, "Fast and Adaptive Fractal Tree-Based Path Planning for Programmable Bevel Tip Steerable Needles," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 601–608, 7 2016.
[25] M. Pinzi, S. Galvan, and F. RodriguezyBaena, "The Adaptive Hermite Fractal Tree (AHFT): a novel surgical 3D path planning approach with curvature and heading constraints," pp. 659–670, 2019.
[26] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Batch Informed Trees (BIT): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, pp. 3067–3074, 5 2015.
[27] A. E. Eiben and J. Smith, "From evolutionary computation to the evolution of things," 2015.
[28] J. Liu and J. Lampinen, "A fuzzy adaptive differential evolution algorithm," *Soft Computing*, 2005.
[29] P. A. Vikhar, "Evolutionary algorithms: A critical review and its future prospects," in *Proceedings - International Conference on Global Trends in Signal Processing, Information Computing and Communication, ICGTSPICC 2016*, 2017.
[30] E. De Momi, C. Caborni, F. Cardinale, G. Casaceli, L. Castana, M. Cossu, R. Mai, F. Gozzo, S. Francione, L. Tassi, G. Lo Russo, L. Antiga, and G. Ferrigno, "Multi-trajectories automatic planner for StereoElectroEncephaloGraphy (SEEG)," *International Journal of Computer Assisted Radiology and Surgery*, vol. 9, no. 6, pp. 1087–1097, 11 2014.
[31] A. Favaro, L. Cerri, D. Scorza, and E. De Momi, "Automatic multi-trajectory planning solution for steerable catheters," in *2018 International Symposium on Medical Robotics, ISMR 2018*, vol. 2018-Janua. IEEE, 3 2018, pp. 1–6.
[32] P. E. Danielsson, "Euclidean distance mapping," *Computer Graphics and Image Processing*, vol. 14, no. 3, pp. 227–248, 11 1980.
[33] K. Toyoda, E. Urasaki, T. Umeno, W. Sakai, A. Nagaishi, S. Nakane, T. Fukudome, and Y. Yamakawa, "The effectiveness of the stereotactic burr hole technique for deep brain stimulation," *Neurologia Medico-Chirurgica*, vol. 55, no. 9, pp. 766–772, 2015.
[34] S. Jalel, P. Marthon, and A. Hamouda, "Optimized NURBS Curves Modelling Using Genetic Algorithm for Mobile Robot Navigation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, pp. 534–545.
[35] L. Piegl and W. Tiller, "The NURBS Book," *Computer-Aided Design*, vol. 28, no. 8, pp. 665–666, 8 1996.
[36] N. M. Razali and J. Geraghty, "Genetic algorithm performance with different selection strategiesin solving TSP," in *Proceedings of the World Congress on Engineering 2011*, 2011, pp. 1134–1139.
[37] S. Y. Ko, L. Frasson, and F. Rodriguez Y Baena, "Closed-loop planar motion control of a steerable probe with a programmable bevel inspired by nature," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 970–983, 10 2011.
[38] A. Fedorov, R. Beichel, J. Kalpathy-Cramer, J. Finet, J. C. Fillion-Robin, S. Pujol, C. Bauer, D. Jennings, F. Fennessy, M. Sonka, J. Buatti, S. Aylward, J. V. Miller, S. Pieper, and R. Kikinis, "3D Slicer as an image computing platform for the Quantitative Imaging Network," *Magnetic Resonance Imaging*, 2012.
[39] B. Fischl, "FreeSurfer," *NeuroImage*, vol. 62, no. 2, pp. 774–781, 8 2012.
[40] D. Scorza, E. De Momi, L. Plaino, G. Amoroso, G. Arnulfo, M. Narizzano, L. Kabongo, and F. Cardinale, "Retrospective evaluation and SEEG trajectory analysis for interactive multi-trajectory planner assistant," *International Journal of Computer Assisted Radiology and Surgery*, vol. 12, no. 10, pp. 1727–1738, 10 2017.

Alberto Favaro was born in 1989. He achieved his MSc degree in Biomedical Engineering at Politecnico di Milano in 2016 and his PhD in Bioengineering at the Department of Electronics, Information and Bioengineering (DEIB) of Politecnico di Milano in 2020. His research regards path planning and tracking solutions for steerable catheters in neurosurgery. He spent a period of research in the Control System Group at the Technical University of Berlin in 2015 and in the Mechatronics in Medicine Lab at Imperial College of London in 2017.

Alice Segato was born in 1994, in Monza (MB), Italy. She obtained the Bachelor and Master of Science in Computer Science and Engineering at Politecnico di Milano. Since November 2018, she is a PhD student at the Department of Electronics, Information and Bioengineering (DEIB) of Politecnico di Milano. Her main research interests include Robotics, Artificial Intelligence and Machine Learning, in the field of Minimal Invasive Neurosurgery.

Federico Muretti was born in Sassari, Italy, in 1993. He obtained his BSc in Biomedical Engineering in 2015 at the University of Pavia and his MSc in Biomedical Engineering in 2018 at Politecnico di Milano with a thesis concerning a novel steerable needle 3D path planning for neurosurgical application. His topics of interest regard Robotics Process Automation and Artificial Intelligence.

Elena De Momi was born in Legnago, Verona, Italy, in 1978. MSc in Biomedical Engineering in 2002, PhD in Bioengineering in 2006, currently Assistant Professor in Electronic Information and Bioengineering Department (DEIB) of Politecnico di Milano. IEEE Senior Member, from 2016 she has been an Associated Editor of IEEE ICRA, IROS and BioRob and she was Publication Co-Chair of ICRA 2019. She is PI for POLIMI of the EDEN2020 project and of ATLAS MSCA-ITN-2018-EJD. She has been evaluator and reviewer for the European Commission in FP6, FP7 and H2020.