

Optimized 3D path planner for steerable catheters with deductive reasoning

Alice Segato¹, Valentina Corbetta¹, Jessica Zangari², Simona Perri², Francesco Calimeri² and Elena De Momi¹

Abstract—Keyhole neurosurgery is challenging, due to the complex anatomy of the brain and the inherent risk of damaging vital structures while reaching the surgical target. This paper presents a path planner for safe and effective neurosurgical interventions. The strengths of the proposed framework lay in the integration of multiple risk structures combined into a deductive method for fast and intuitive user interaction, and a modular architecture. The tool is intended to support neurosurgeons at quickly determining the most appropriate surgical trajectory through the brain matter with minimized risk; the user interface guides the user through the decision making process and helps save planning time of neurosurgical interventions. Risk structures and trajectories can be visualized in an intuitive way, thanks to a 3D brain surgery simulator developed with Unity. A qualitative evaluation with clinical experts shows the practical relevance, while a quantitative performance and functionality analysis proves the robustness and effectiveness of the system with respect to literature.

I. INTRODUCTION

Minimally invasive procedures are the gold standard in brain surgery due to the brain’s sensitivity and the associated high risk [1]. Straight surgical tools are inserted into intracerebral structures; a carefully planned trajectory is crucial for the fast and safe performance of neurosurgical procedures and for minimizing injuries to nearby tissues inside the brain. Moreover, target can be difficult to reach by traditional rigid needles, without damaging the adjacent tissues. Steerable needles represent a breakthrough in neurosurgery, as they grant access to sensitive destinations [2]. Over the last two decades, different research groups have focused their efforts on the development on needles able to steer inside the tissue [3]. These needles can perform curvilinear trajectories planned in order to maximize the distance from sensitive anatomical structures to be avoided and reach targets that results to be inaccessible via rectilinear insertion paths; different steerable needle designs and approaches have been proposed in literature [4].

Planning a keyhole surgical operation in the brain can be a very difficult task, even for experienced surgeons. The challenge arises from the human brain’s complex morphology and the requirement to avoid various risk structures and eloquent areas in the brain.

In the current standard of care, trajectory planning is manually performed and often relies on the surgeon’s experience. To verify the chosen trajectory and estimate the treatment risk, the surgeon must consider every slice of a Magnetic Resonance Imaging (MRI) or Computed Tomography (CT) scan along the trajectory when planning the operation. This is a complicated and time-consuming task [5]. Procedural complexity is added during the planning of a surgical trajectory by taking into account additional non-linear trajectories and different risk structures. Differently from conventional needles, for which the insertion path can be planned and performed by the clinician on the basis of the target location and the patient anatomy, the complex kinematics of steerable needles make the path planning unbearable requiring the aid of automatic or semi-automatic path planning solutions. As a result, surgeons need adequate automated planning systems for planning brain interventions assisting in the process.

Computing the optimal trajectory in the preoperative phase can be modeled as a path planning problem [6]. Realization of a digital access plan requires to calculate the risk of a trajectory penetrating certain risk structures, which must be correctly segmented prior to planning.

The goal of this work is to develop an optimization-based path planner, able to pre-operatively assist the surgeon and estimate optimal curvilinear trajectories. The proposed method solved the path planning problem with a deductive learning approach applied to the expert manual planning and assists neurosurgeons by providing a tool for efficient and straightforward volumetric risk structure visualization and faster planning of neurosurgical interventions. The proposed system supports the surgeon in choosing the most suitable curvilinear trajectory, as well as avoiding vital structures, and minimizing potential trauma to healthy tissue.

II. RELATED WORK

In the context of path planning, a variety of approaches has been proposed in the literature and can be divided into four main categories:

A. Optimization-Based methods

In Schulman *et al.* [7], the path optimization problem is solved via sequential convex optimization. Although the method does not guarantee to find a solution if it exists, it can provide locally optimal collision-free paths. Duindam *et al.* [8] proposed an explicit solution to inverse kinematics.

*This project has received funding from the European Union Horizon 2020 research and innovation program under grant agreement No 688279.

¹Department of Electronics, Information and Bioengineering, Politecnico di Milano, Italy alice.segato@polimi.it

²Department of Mathematics and Computer Science, University of Calabria, Italy

The method 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 [9], compute a potential field similar to the one generated by electrical charges. This approach has the drawback of generating local minima. To address this problem, Li *et al.* [10] proposed an artificial potential field method. 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.

B. 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.* [11] proposed the incremental A* solution for 2D applications. These methodologies show high computational time as the discretization of the environment becomes finer. Leibrandt *et al.* [12] reported a method that 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. The search is confined in a subset of configurations close the current one.

C. Sampling-based methods

Sampling-based planners are a type of path planner that generate the robot’s path by sampling sequentially different points in robot’s workspace and gradually construct a data structure that represent collision-free paths. They feature a probabilistic completeness, i.e. if a solution to the path planning problem exists, they will eventually find it. A combination of RRT and a reachability-guided sampling heuristic (RG-RRT) was used in the work of Patil *et al.* [13] to compute motion plans. 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 [14]. Patil *et al.* [15] proposed a solution confining the search on the subset of points in the workspace that meet the kinematic constraints of the needle allows a re-planning of the path suitable for online procedures, ensuring a clinically acceptable error. Gammell *et al.* [16] proposed the Batch Informed Tree (BIT*) algorithm. In our previous work [17], a RRT* approach has been implemented for developing a path. Additionally, the path optimization was hindered by the intrinsically limited flexibility of cardinal splines used for the path interpolation.

D. Learning-based methods

Graph-based and sampling-based methods, considered the standard approaches for path planning, are limited in the

context of KN with flexible catheters, by the impossibility, unless using further steps, to directly optimize the trajectory in terms of obstacle clearance and kinematic constraints. In our previous work [18], made use of the DRL approach to a grid path planning problem with promising results on small environments.

E. Reasoning-based methods

Path Planning for robotic steerable needle can be tackled exploiting classical methods, such as graph-based, search-based and learning-based approaches. However these techniques present some limitations: the first two require a trade-off between completeness and efficiency, while the latter needs large datasets to successfully train models.

Interestingly, reasoning-based approaches for path planning have been successfully designed, providing high-level reasoning methods like in [19]. [20] successfully solved the path planning problem.

Reasoning-based approaches have increased capability of explicitly represent domain knowledge; however, a path planning system for complex environments based only on a deductive reasoning-based method or similar approaches might be insufficient, as current implementations cannot handle an excessive increase of the search space and generalise on different environments [21]. In order to overcome these drawbacks, we propose to model the path planning problem for steerable needle in neurosurgery combining surgeon manual planning with deductive optimization method. In particular our system, exploits Answer Set Programming (ASP) semantics to model the brain environment and thus implement an artificial intelligent agent able to move within it, satisfying requirements, which can be customized depending on the specific application and based on the preferences expressed by domain experts, as surgeons and clinicians.

III. MATERIALS AND METHODS

As briefly mentioned in Section I, the goal of the proposed system is to help neurosurgeons to detect an optimal trajectory from an entry point (*EP*) to a target structure (*TS*) inside the human brain using a steerable needle. The criteria that make a trajectory “the best” one are several and their importance depends on the application (e.g. they vary from Deep Brain Stimulation (DBS) and Convection Enhanced Delivery (CED), which are two relevant applications of steerable needles in keyhole neurosurgery). The main criteria to consider a trajectory as viable is that it must not hit an obstacle, meaning that it must avoid the sensitive structures, and that it must hit the target. Another important aspect is that the trajectory must respect the kinematic constraints of the moving agent. These criteria apply to any neurosurgery scenario, while depending on the use case, the clinician might prefer to privilege shortness of path, minimization of the curvature or maximization of the distance from obstacles. A planning tool is implemented in 3D Unity [22] that visualizes the 3D segmented risk structures of the brain of the patient derived from that data. First of all the surgeon is asked to input in the neurosurgical simulator the kinematic limitations

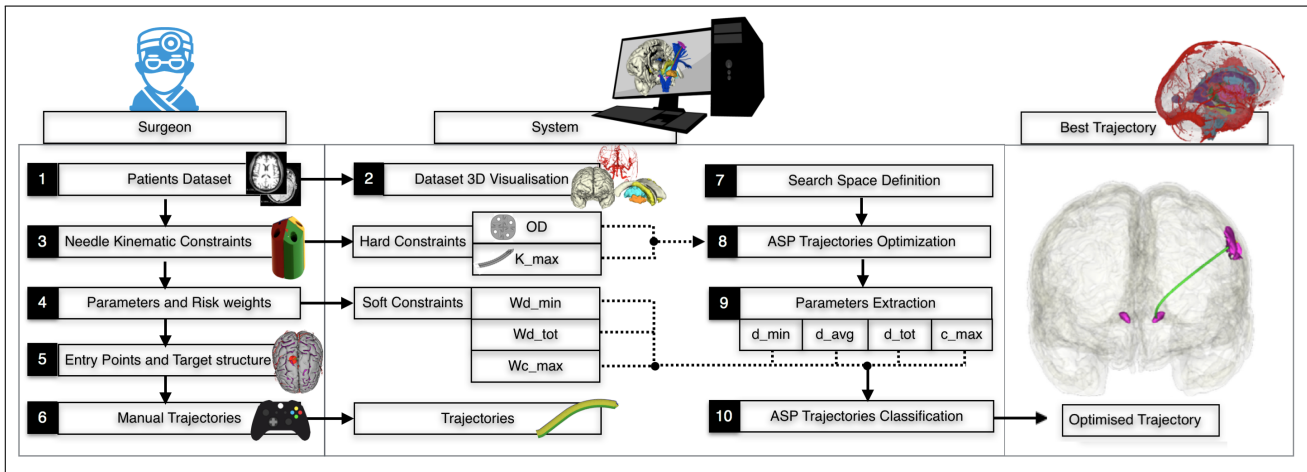


Fig. 1. Architecture of the proposed system, displaying the basic workflow and the user interactions.

of the needle he/she intends to use, which are the curvature, K_{max} , and outer diameter, OD , and the parameters he wants to prioritize in the selection of the best trajectory, $w_{d_{tot}}$, $w_{d_{min}}$ and $w_{c_{max}}$, used either to minimize or to maximize the rule expression. A surgeon can then select a target in the brain (TS), e.g. the tumor and an entry point EP , on the brain cortex.

Then, starting from the EP the surgeon is asked to manually draw a trajectory, defined as:

$$T(T = \{t_0, t_1, \dots, t_{n-1}\}) \quad (1)$$

where $t_0 = EP$ and $t_{n-1} = TS$.

Each of these paths is the input of an optimization procedure followed by a classification procedure, which will provide the optimal trajectory, T^{Opt} .

All system components and the workflow of the surgical trajectory planning tool are depicted in Figure 1, presenting an overview of the proposed system.

A. Answer Set Programming (ASP)

As already mentioned in Section I, the optimization and selection of the most suitable trajectory is achieved via deductive reasoning, implemented through ASP [23], a declarative programming paradigm born in the field of logic programming and non-monotonic reasoning. Thanks to its declarative approach, its high expressivity and the ability to deal with incomplete knowledge, ASP is very popular in AI and represents an effective tool for Knowledge Representation and Reasoning (KRR). With ASP a problem is modeled by means of a logic program composed by a collection of rules; intended models of such a program are called *answer sets*, and correspond one-to-one to a solution of the modeled problem; indeed, in general, an ASP program can feature none, one or multiple answer sets. In case the input program has no answer sets, the encoded problem has no solutions. Thanks to years of work in the scientific community, several efficient and reliable ASP systems are currently available, supporting the ASP language and semantics and hence capable of computing answer sets of an input program, such as DLV/DLV2 [24] and Clingo [25].

With respect to imperative languages, ASP presents several benefits. Rather than focusing on coding algorithms and thus on *how to solve* the computational problem at hand, the declarative nature of ASP allows one to focus on *how to describe* such computational problem, completely avoiding the need for explicitly designing the steps to be executed. In turn, as there is no need for algorithm design or coding, explicit updates in the problem specification can be more easily incorporated and this enables different advantages such as fast prototyping, quick error detection and modularity. Besides, a clean model-theoretic semantics grants correctness; intuitively, an ASP program can be seen as a formal yet executable description of the problem. The basic construct of ASP is a rule, that has form $Head \leftarrow Body$, where the *Body* is a logic conjunction in which negation may appear, and *Head* can be either an atomic formula or a logic disjunction. A rule is interpreted according to common sense principles: roughly, its intuitive semantics corresponds to an implication. Hereafter, we briefly recall the main concepts of ASP syntax, necessary to understand the proposed methodology. For further details on ASP, its features and applications, we refer the reader to the vast literature, starting from [26] and [27].

If t_1, \dots, t_k are terms (either constants or variables) and p is a predicate symbol of arity k , then $p(t_1, \dots, t_k)$ is an atom. A literal l is of the form a or $\text{not } a$, where a is an atom; in the former case l is positive, negative otherwise. A rule is of the following form:

$$a_0 | \dots | a_h : -b_1, \dots, b_n \text{ not } b_{n+1}, \dots, b_m. \quad (2)$$

On the left, the symbol “|” connects atoms that are part of a disjunction in the head, whereas comma separated literals in the right side, i.e., the body, are part of a conjunction. A fact is a rule with one single atom in the head and an empty body and represents a certainly true information. A constraint is a rule with empty head; hard (“strong” or “classical”) and soft (“weak”) constraints can be specified in order to cut out undesired models and express preferences in case of optimization problems, respectively. Weak constraints are

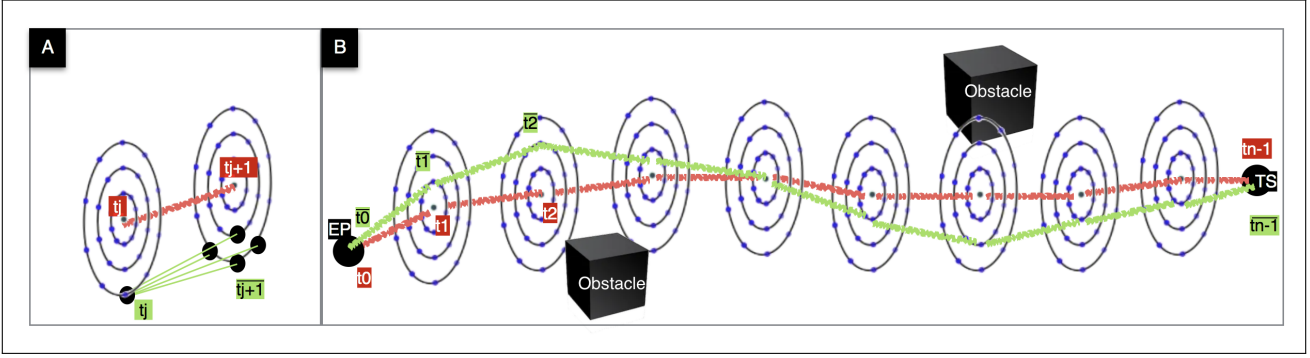


Fig. 2. **A** shows an example of search space built around two points t_j and t_{j+1} of manual trajectory $T(T = \{t_0, t_1, \dots, t_{n-1}\})$. **B** shows the comparison between manual trajectory $T(T = \{t_0, t_1, \dots, t_{n-1}\})$ and optimized trajectory $\bar{T}(\bar{T} = \{\bar{t}_0, \bar{t}_1, \dots, \bar{t}_{n-1}\})$ from $EP = t_0 = \bar{t}_0$ to $TS = t_{n-1} = \bar{t}_{n-1}$.

expressed with the symbol $:\sim$ instead of $:-$ as it is the case for hard constraints. These latter are conditions that must be satisfied, whereas soft constraints represent conditions that should be full-filled; intuitively, when a solution violates a soft constraint it pays a cost; this induces an ordering among solutions that allows one to express minimization or maximization criteria. Such criteria that can equivalently be expressed via `#minimize` and `#maximize` statements. Moreover, ASP features choice rules, that are compact ways for expressing disjunction of atoms that must adhere to some cardinality conditions.

B. Search Space Definition

Starting from the manual trajectory, T , a new search space is built in which ASP is used to look for alternative paths.

First of all, around each point $t_j \in T$, excluding the entry point and target position, we create a set of concentric circles of radius $0 \leq r \leq 5mm$. Then, for each circle, we sample \bar{t}_j points equally spaced on its circumference by applying the formulas in Equation 3. Given the coordinates x_{t_j} , y_{t_j} and z_{t_j} of the center of the circle, θ_{t_j} values of angles uniformly distributed between 0° and 360° and the value of the radius r of the desired circle, the formulas in Equation 3 allow to pass from polar to cartesian coordinates. This way, we obtain the x , y and z coordinates of each point \bar{t}_j on its circumference.

$$\begin{aligned} \bar{x}_{t_j} &= r \times \cos(\theta_{t_j}) + x_{t_j} \\ \bar{y}_{t_j} &= r \times \sin(\theta_{t_j}) + y_{t_j} \\ \bar{z}_{t_j} &= z_{t_j} \end{aligned} \quad (3)$$

From the newly generated set of available points \bar{t}_j , we create a search graph: each point constitutes a node in the graph. Considering point t_j in the input path T , each point \bar{t}_j , in the concentric circles around it, is connected by an edge to the corresponding point \bar{t}_{j+1} , on the concentric circle having the same radius, built around point t_{j+1} ; \bar{t}_j is also connected by an edge with the immediately adjacent points to \bar{t}_{j+1} . The entry point \bar{t}_1 and the target position \bar{t}_{n-1} are the same of the original trajectory T . Figure 2A shows an example of search space built around two points t_j and t_{j+1} of the trajectory T .

C. ASP Trajectory Optimization

The search graph is encoded by facts of form `edge(X, Y)` expressing that there is an edge between points X and Y . Moreover, facts `start(X)` and `finish(X)` encode EP and TS , respectively. Eventually, possible steps are represented as facts of the form `step(X)`. These facts are the input to an ASP program constituted by the logic rules described next. First, we define that at the step 0 the needle must be placed in the start position EP by means of this rule:

$$\text{needle_at}(0, P) :- \text{start}(P). \quad (4)$$

Then, we guess for each step $S > 0$ different from TS the points the needle should visit in order to obtain a possible path. More in detail, the needle can move from point $P1$ at step $S-1$ to point $P2$ at step S only if the needle visited $P1$ at step $S-1$, $P1$ and $P2$ are connected by an edge and the target structure TS has not been reached yet. Moreover, we ensure that the needle cannot visit two or more points at the same time and that it cannot move to positions already visited. This is encoded by the following choice rule:

$$\begin{aligned} 0 \leq \{ & \text{needle_at}(S, P2) : \text{edge}(P1, P2), \\ & \text{needle_at}(S-1, P1), \\ & \text{not finish}(P2) \} \leq 1 :- \text{step}(S). \end{aligned} \quad (5)$$

Lastly, we check that guessed paths reach the target structure TS and discard paths that do not satisfy such condition via an integrity constraint:

$$\begin{aligned} \text{finished} :- & \text{needle_at}(S, P), \\ & \text{step}(S), \text{finish}(P). \end{aligned} \quad (6)$$

$$:- \text{not finished}. \quad (7)$$

Figure 2B shows a comparison between manual trajectory $T(T = \{t_0, t_1, \dots, t_{n-1}\})$ and optimized trajectory $\bar{T}(\bar{T} = \{\bar{t}_0, \bar{t}_1, \dots, \bar{t}_{n-1}\})$ from $EP = t_0 = \bar{t}_0$ to $TS = t_{n-1} = \bar{t}_{n-1}$. As it can be seen from the figure, the optimized

trajectory allows to keep a higher distance from the obstacles, which is the most critical parameter according to the surgeon, as it is explained in the following paragraph.

D. Parameters Extraction

For each path \bar{T} found by the optimization procedure carried out via the above ASP logic program, the following parameters are computed:

- the length of the path (d_{tot}),
- the average distance from the set of obstacles (d_{avg}),
- the minimum distance from the set of obstacles (d_{min}),
- the maximum curvature of the path (c_{max}).

These parameters are used in the final step of our workflow, which is the classification of the trajectories.

E. ASP Classification of trajectories

The classification part of our methodology chooses the best trajectory among the ones created by the clinician and the ones generated in the optimization step. In this phase, we guess a path to choose among all trajectories, encoded by facts of form `trajectory(X)`:

$$1 \leq \text{choose}(X) : \text{trajectory}(X) \leq 1 \quad (8)$$

The guessed path is then checked against some requirements. In particular, a domain expert contributed with the knowledge necessary to define the hard and soft constraints used to select the optimal path, which we have encoded in ASP rules. We now describe more in detail these constraints.

First of all, the main hard constraint guarantees the respect of the kinematic limitations K_{max} and OD . The ASP reasoner must discard a trajectory X if the maximum curvature c_{max} of X (modeled by a fact of form `maxCurve(X, c_{max})`) is bigger than K_{max} :

$$\begin{aligned} & :- \text{choose}(X), \text{curvature}(c_{max}), \\ & \text{maxCurve}(X, K_{max}), K_{max} < c_{max}. \end{aligned} \quad (9)$$

Furthermore, a path X cannot be chosen if it approaches the delicate structures at a minimum distance d_{min} (encoded by a fact of form `distObst(X, d_{min})`), which is smaller than the radius of the needle $r = \frac{OD}{2}$ (expressed by the fact `radius(r)`):

$$\begin{aligned} & :- \text{choose}(X), \text{radius}(r), \\ & \text{distObst}(X, d_{min}), d_{min} < r. \end{aligned} \quad (10)$$

An anticipated, in keyhole neurosurgery the characteristics which make a path optimal can vary among different use cases. The integration of ASP in our tool allows the flexibility and generalizability necessary to implement these desiderata. Indeed, from the expert's knowledge, we selected some soft constraints which can be prioritized by the surgeon: when inserting the inputs in the simulator, he can select all, some or none of these rules and also their priority, by assigning a weight to each of them. We show below the implementation of these preferences in the ASP program.

- 1) Minimization of the path length d_{tot} (expressed by facts of form `length(X, d_{tot})`), as this caution helps to minimize risks in the motion of the needle:

$$\begin{aligned} & \# \text{minimize} \{ d_{tot} @ w_{d_{tot}}, X : \text{choose}(X), \\ & \text{length}(X, d_{tot}) \}. \end{aligned} \quad (11)$$

- 2) Maximization of the distance from obstacles, to mitigate the damage of possible mistakes in the motion of the needle:

$$\begin{aligned} & \# \text{maximize} \{ d_{min} @ w_{d_{min}}, X : \text{choose}(X), \\ & \text{distObst}(X, d_{min}) \}. \end{aligned} \quad (12)$$

- 3) Minimization of the curvature performed by the needle:

$$\begin{aligned} & \# \text{minimize} \{ c_{max} @ w_{c_{max}}, X : \text{choose}(X), \\ & \text{maxCurve}(X, c_{max}) \}. \end{aligned} \quad (13)$$

Essentially, the facts `length(X, d_{tot})`, `distObst(X, d_{min})` and `maxCurve(X, c_{max})` respectively couple d_{tot} , d_{min} and c_{max} to the corresponding trajectory X . $w_{d_{tot}}$, $w_{d_{min}}$ and $w_{c_{max}}$ are the weights for each constraint set by the user; the higher the weight, the higher the priority.

F. Hardware Specification

Experiments were performed on a Linux machine equipped with a 6-core i7 CPU, 16GB of RAM and 1 NVIDIA Titan XP GPU with 12GB of VRAM.

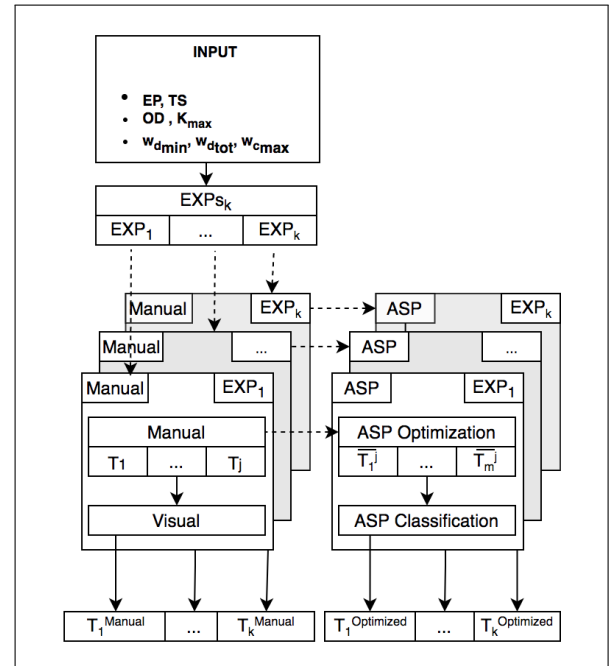


Fig. 3. The system takes in input the Entry points, EP_j (with $1 \leq j \leq 10$), the target structure, TS , the weights, $w_{c_{max}}$, $w_{d_{min}}$, $w_{d_{tot}}$ and the kinematic constraints of the moving agent, OD and K_{max} . k experiments, EXP_k (with $1 \leq k \leq 5$), were conducted for each approach, Manual and ASP, and for each the optimal one, $path_k$, is selected.

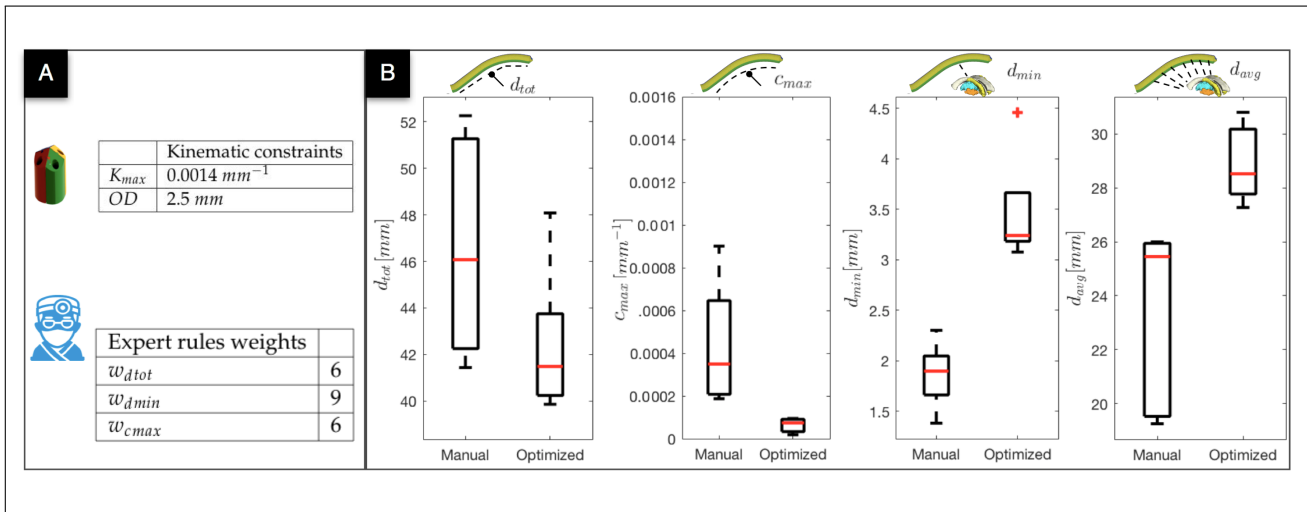


Fig. 4. **A** Shows expert’s rules, and catheter kinematic constraints. **B** Shows a comparison between Manual and Optimized approach in CED environment. The results for both considered scenarios and used approaches are reported in terms of the minimum (d_{min}) and the mean (d_{avg}) distance from the critical obstacles, the total path length (d_{tot}) and the curvature (c_{max}) calculated over the five best paths for each approach. P-values were calculated using Wilcoxon matched-pairs signed-rank test.

G. Statistical analysis

All the performance metrics, extracted from the path, were analysed employing Matlab (The MathWorks, Natick, Massachusetts, R2020a). Lilliefors test has been initially applied for data normality. Due to the non-normality of data distribution, pairwise comparison was performed with the Wilcoxon matched-pairs signed-rank test. Differences were considered statistically significant at p-value < 0.05 .

H. Experimental Protocol

The surgical procedure considered is the CED, the target is the tumor (surrounded by different essential structures where vessels represent the extensive obstacles). 3-D brain structures were identified on two datasets: 1) Time-of-Flight (ToF) Magnetic Resonance (MR) for vessels visualisation, 2) T1 for brain cortex, skull surface, arterial blood vessels, ventricles and all the relevant deep grey matter structures visualisation, segmented through FreeSurfer Software [28] and 3-D Slicer [29]. The catheter considered is the bio-inspired, multi-segment programmable bevel-tip needle (PBN), currently under development as core technology of the EDEN2020(*) project [30].

As shown in Figure 3 to evaluate the optimal path found by the ASP planner, we compared it with the manual trajectories computed by the expert surgeon. The expert surgeon (age: 37, performed surgical procedures: 2440) was asked to select 10 entry points, EP_i , on the brain cortex, a target structure TS and the weights, $w_{c_{max}}$, $w_{d_{min}}$, $w_{d_{tot}}$ for the rules prioritization. k experiments, EXP_k (with $1 \leq k \leq 5$), were conducted. For each EXP_k , the surgeon was asked to generate a pool of trajectories, $\{T_j\}$ (with $1 \leq j \leq 10$), and choose the optimal one, T^{Manual} , based on his expertise. The same pool of surgical paths generated in the manual approach, $\{T_j\}$, was used for the optimization procedure and the optimal one, $\{T^{Opt}\}$, was selected, using rules, weights

and kinematic constraints given in input by the surgeon as reported in Figure 4A.

IV. RESULTS

Figure 4B shows a comparison between the Manual and Optimized approaches in terms of d_{min} , d_{avg} , c_{max} and d_{tot} calculated over the best path of left hemisphere. The Optimized approach keeps lower d_{tot} and significantly lower c_{max} (p-value < 0.01) of the path than the Manual approach. While it keeps a significantly greater d_{min} (p-value < 0.01) and d_{avg} (p-value < 0.01) following the rules dictated by the expert who gives more importance to these two parameters in this case.

V. DISCUSSION AND CONCLUSION

In this work we presented a novel approach to model the problem of path planning in the context of keyhole neurosurgery using deductive reasoning. The presented tool was planned and implemented to supplement the surgical workflow and potentially accelerate the required time for safely planning a trajectory, and thus increasing the outcome for both the neurosurgeon and the patient. ASP allowed us to model in a purely declarative fashion the environment and the steerable needle which moves in it using a set of logic rules, hence creating a flexible methodology which can be customized by surgeons depending on the single patient. Results show the viability of our approach and the effectiveness of the framework that, thanks to the use of ASP, allows one to find the minimum number of steps to reach the target while easily customize priorities.

ACKNOWLEDGMENTS

We thank Eden2020 project consortium partners for precious advice during the project activities. We thank M. Riva for providing the dataset and the manual trajectories for the CED environment simulation.

REFERENCES

- [1] T. Liu, Y. Tai, C. Zhao, L. Wei, J. Zhang, J. Pan, and J. Shi, "Augmented reality in neurosurgical navigation: A survey," *The International Journal of Medical Robotics and Computer Assisted Surgery*, p. e2160, 2020.
- [2] A. Segato, V. Pieri, A. Favaro, M. Riva, A. Falini, E. D. Momi, and A. Castellano, "Automated steerable path planning for deep brain stimulation safeguarding fiber tracts and deep gray matter nuclei," *Front. Robotics and AI*, vol. 2019, 2019.
- [3] M. Scali, T. P. Pusch, P. Breedveld, and D. Dodou, "Needle-like instruments for steering through solid organs: A review of the scientific and patent literature," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 231, no. 3, pp. 250–265, 2017.
- [4] N. J. Cowan, K. Goldberg, G. S. Chirikjian, G. Fichtinger, R. Alterovitz, K. B. Reed, V. Kallem, W. Park, S. Misra, and A. M. Okamura, *Robotic Needle Steering: Design, Modeling, Planning, and Image Guidance*. Boston, MA: Springer US, 2011, pp. 557–582.
- [5] M. Trope, R. R. Shamir, L. Joskowicz, Z. Medress, G. Rosenthal, A. Mayer, N. Levin, A. Bick, and Y. Shoshan, "The role of automatic computer-aided surgical trajectory planning in improving the expected safety of stereotactic neurosurgery," *International journal of computer assisted radiology and surgery*, vol. 10, no. 7, pp. 1127–1140, 2015.
- [6] M. A. Audette, S. P. Bordas, and J. E. Blatt, "Robotically steered needles: A survey of neurosurgical applications and technical innovations," *Robotic Surgery: Research and Reviews*, vol. 7, p. 1, 2020.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] A. Segato, V. Pieri, A. Favaro, M. Riva, A. Falini, E. De Momi, and A. Castellano, "Automated steerable path planning for deep brain stimulation safeguarding fiber tracts and deep grey matter nuclei," *Frontiers in Robotics and AI*, vol. 6, p. 70, 2019.
- [18] A. Segato, L. Sestini, A. Castellano, and E. De Momi, "Ga3c reinforcement learning for surgical steerable catheter path planning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2429–2435.
- [19] V. Lifschitz, "Answer set programming and plan generation," *Artificial Intelligence*, vol. 138, no. 1-2, pp. 39–54, 2002.
- [20] J. J. Portillo, C. L. Garcia-Mata, P. R. Márquez-Gutiérrez, and R. Baray-Arana, "Robot platform motion planning using answer set programming," in *LA-NMR*, 2011, pp. 35–44.
- [21] E. Erdem, E. Aker, and V. Patoglu, "Answer set programming for collaborative housekeeping robotics: representation, reasoning, and execution," *Intelligent Service Robotics*, vol. 5, no. 4, pp. 275–291, 2012.
- [22] W. Goldstone, *Unity game development essentials*. Packt Publishing Ltd, 2009.
- [23] M. Gelfond and V. Lifschitz, "Classical negation in logic programs and disjunctive databases," *New generation computing*, vol. 9, no. 3-4, pp. 365–385, 1991.
- [24] W. T. Adrian, M. Alviano, F. Calimeri, B. Cuteri, C. Dodaro, W. Faber, D. Fuscà, N. Leone, M. Manna, S. Perri *et al.*, "The asp system dlv: advancements and applications," *KI-Künstliche Intelligenz*, vol. 32, no. 2-3, pp. 177–179, 2018.
- [25] M. Gebser, R. Kaminski, B. Kaufmann, and T. Schaub, "Multi-shot asp solving with clingo," *Theory and Practice of Logic Programming*, vol. 19, no. 1, pp. 27–82, 2019.
- [26] V. Lifschitz, *Answer set programming*. Springer, 2019.
- [27] F. Calimeri, W. Faber, M. Gebser, G. Ianni, R. Kaminski, T. Krennwallner, N. Leone, M. Maratea, F. Ricca, and T. Schaub, "Asp-core-2 input language format," *Theory and Practice of Logic Programming*, vol. 20, no. 2, pp. 294–309, 2020.
- [28] B. Fischl, "Freesurfer," *Neuroimage*, vol. 62, no. 2, pp. 774–781, 2012.
- [29] S. Pieper, M. Halle, and R. Kikinis, "3d slicer," in *2004 2nd IEEE international symposium on biomedical imaging: nano to macro (IEEE Cat No. 04EX821)*. IEEE, 2004, pp. 632–635.
- [30] R. Secoli and F. Rodriguez y Baena, "Adaptive path-following control for bio-inspired steerable needles," in *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 6 2016, pp. 87–93. [Online]. Available: <http://ieeexplore.ieee.org/document/7523603/>