Implicit Active Constraints for a Compliant Surgical Manipulator

Konrad Leibrandt[1], *Member, IEEE*, Hani J. Marcus[1,2],
Ka-Wai Kwok[3], *Member, IEEE*, and Guang-Zhong Yang[1], *Fellow, IEEE*

***Abstract*— Active constraints are high-level control algorithms providing software-generated force feedback from virtual environments. When applied to surgery, they can assist surgeons in performing complex tasks by guiding their navigation pathways along narrow, possibly convoluted, surgical trajectories. This paper presents a method to generate concave tubular constraints implicitly from pre- or intra-operative data. Patient-specific constraints may be generated efficiently with the proposed scheme and readily deployed in various surgical scenarios. Furthermore, a five degree-of-freedom active constraint framework is proposed, which accounts for the entire tool shaft rather than just the end-effector, and is applicable to both static and dynamic active constraint scenarios. Experimental results on simulated surgical tasks show that this framework can improve safety and accuracy as well as reduce the perceived workload during complex surgical tasks.**

I. Introduction

Advances in medical imaging have enabled real-time registration of robotic devices within the surgical environment for safe and effective surgical navigation [1]. Moreover, they have enabled human-robot collaborative control schemes by incorporating Virtual Fixtures [2] or Active Constraints, in which imaging models are used to provide intra-operative haptic force feedback [3]. Since its use for tele-operated robotic platforms [4]–[6], the concept of Virtual Fixtures has also been adopted for hand-held robotic systems [7], [8], in which both the operator and the robotic manipulator hold the tool and share one end-effector. The manipulation guidance can be generated *in situ* towards desired targets by imposing haptic feedback on the tool. The concept of Active Constraints has attracted increasing attention as a method for intra-operative surgical guidance, reducing the risk of inadvertent damage to surrounding anatomical structures [1]. It is particularly useful for procedures carried out in confined working spaces.

Contemporary systems incorporating Active Constraints generally require that imaging-derived models be imposed in a relatively stationary fashion onto the live surgical workspace; inevitably restricting the application of such systems to cases in which minimal intra-operative shift occurs. Park *et al.* recently proposed a constraint model capable of rapidly compensating for cardiac motion [9], allowing a catheter to be navigated along the centerline of a deforming vessel to the target site. Simple constraints in the form of curved lines are unstable under the closed loop force control [10]. To access the intra-mural motion of the myocardium, Ren *et al.* [11] proposed guidance constraints with explicit spatial details computed by a potential field. However, only a single-point end-effector, rather than the instrument body, was constrained, limiting its clinical value in the context of restricted surgical workspaces. Li *et al.* [6] first devised static and spatial constraints with the aim to limit the motion of both the tool shaft and its tip accessed deeply inside a small sinus cavity but their methodology relied heavily on accurate registration between the pre-operative imaging model and the cavity, and had very low tolerance for operator interaction with the constraints boundary.

In this paper, we develop a control interface that enables the operator to define a safe constraint tunnel leading to the surgical site of interest, while avoiding prohibitive regions. The pathway comprises a series of contours outlined by various number of vertices so that it can be modeled with complex morphology. Such vertices can be registered rapidly with the deforming tissue or moving anatomy. To generate haptic interaction in real time, we also exploit a fast Proximity Query (PQ) process [3], which computes the relative placement between the constraint model and the tool at a rate above 1 kHz. Furthermore, distribution function defining the force vector aggregation among the vertices is also provided, which represent the constraint boundary. Therefore, the operator will sense seamless, steady force feedback throughout the procedure. Additional constraints, such as remote-center of motion (RCM), can also be superimposed to the prescribed active constraints.

The contributions of this paper are listed as follows:

- Concave tunnel algorithm for generalized anatomical constraints.
- Control framework for five degree-of-freedom dynamic active constraints and tool shaft consideration.
- Force aggregation function for seamless, steady force feedback.

While minimally invasive surgery (MIS) has been widely adopted within general surgery, the use of such techniques in certain surgical fields has been hampered by significant technical challenges. Keyhole transcranial approaches, for example, utilize a single entry point, a narrow surgical corridor, and a small working space [12]. Moreover, the morbidity associated with deviating from the safe surgical trajectory and injuring unaffected eloquent brain tissue is often severe and usually permanent. The application of tunnel-shaped

[1]Konrad Leibrandt, Hani J. Marcus, and Guang-Zhong Yang are with the Hamlyn Centre for Robotic Surgery, Imperial College London, United Kingdom k.leibrandt12 at imperial.ac.uk.

[2]Hani J. Marcus is furthermore with the Department of Neurosurgery, Imperial College Healthcare NHS Trust, United Kingdom.

[3]Ka-Wai Kwok is with the Department of Mechanical Engineering, The University of Hong Kong, Hong Kong.

constraints therefore holds the promise of increasing patient safety and improving the effectiveness of such approaches.

The remainder of this paper is organized as follows: Sec. II presents the algorithm to generate implicit concave constraint tunnels; Sec. III introduces the dynamic active constraint framework; Sec. IV describes the validation studies conducted and their findings; and Sec. V presents our conclusions.

II. Implicit Concave Constraint Tunnels

The tubular constraints are generated from cross-sections described by three-dimensional (3D) point sets. The algorithm allows point sets which fill out the entire permitted area (Fig. 1 (a) [red]) or point sets which represent the contour of the permitted area (Fig. 1 (a) [green]). We anticipate that the point sets are obtained from pre-operative segmented images or from *in situ* point picking with an optical tracker or articulated arm. In the first stage the concave contour of each cross-section is calculated. In the second stage these contours are connected to form a tunnel.

A. Calculating Cross-Section Contour

The input point set $\mathbf{P}^{\text{in}}$, with $\mathbf{c}^{\text{in}}$ the centroid of $\mathbf{P}^{\text{in}}$, is first projected on the best fit plane in 3D. The eigenvectors of matrix $\mathbf{M}$, where

$$\mathbf{M} = \begin{bmatrix} \mathbf{P}_1^{\text{in}T} - \mathbf{c}^{\text{in}T} \\ \vdots \\ \mathbf{P}_{\text{n}}^{\text{in}T} - \mathbf{c}^{\text{in}T} \end{bmatrix}, \tag{1}$$

are determined with a singular value decomposition (SVD),

$$\mathbf{M} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^{*}. \tag{2}$$

The two eigenvectors $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ in $\mathbf{V}^{*}$ correspond to the two highest eigenvalues and are used to define a plane as:

$$\langle \boldsymbol{n}\,, (\boldsymbol{r} - \boldsymbol{r}_0) \rangle \;=\; 0\;;\; \boldsymbol{n} = \boldsymbol{v}_1 \times \boldsymbol{v}_2,\; \boldsymbol{r}_0 = \mathbf{c}^{\text{in}}, \tag{3}$$

where $\boldsymbol{n}$ the normal vector of the plane; $\boldsymbol{r}$ is defined as an arbitrary point on the plane

$$\boldsymbol{r} = \boldsymbol{r}_0 \;+\; s\,\boldsymbol{v}_1 \;+\; t\,\boldsymbol{v}_2 \tag{4}$$

is the alternative parameter representation. The point set $\mathbf{P}^{\text{in}}$ is then projected on the plane:

$$\boldsymbol{r}_i' = \mathbf{P}_i^{\text{in}} - \left(\boldsymbol{n}^T\,\left(\mathbf{P}_i^{\text{in}} - \boldsymbol{r}_0\right)\right)\boldsymbol{n}, \tag{5}$$

where $\boldsymbol{r}_i'$ is the projection of $\mathbf{P}_i^{\text{in}}$ on the plane. $\boldsymbol{r}_i'$ is then transformed into the two-dimensional (2D); s, t-space using,

$$\begin{bmatrix} s & t & 0 \end{bmatrix}^T = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \boldsymbol{e}_z \end{bmatrix}^{-1} (\boldsymbol{r}_i' - \boldsymbol{r}_0), \tag{6}$$

where $\boldsymbol{e}_z = [0, 0, 1]^T$. Fig. 1 (b) shows the 2D points in the domain of (s, t). The convex hull (**CVH**) of this 2D point set $\mathbf{S}$ is then calculated using the Graham Scan Algorithm [13]. To precisely represent the contour of $\mathbf{S}$, a further process is applied which transforms the **CVH** into a concave hull (**CCH**). A modified version of the concave hull approach presented in [14] is used. **CCH** is first initialized with the points of **CVH**. In a recursive process points

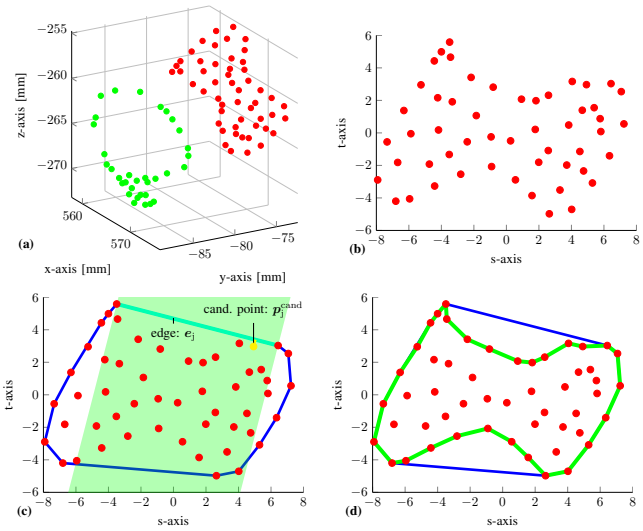

Fig. 1. Process of implicit tunnel generation. (a): example cross-section user input, (b): user input transformed in plane, (c): transforming convex hull into concave hull, (d): resulting concave hull for $k = 3$.

$(\boldsymbol{p}_i \in \mathbf{S}) \wedge (\boldsymbol{p}_i \notin \textbf{CVH})$ are included into **CCH**. Edge $\boldsymbol{e}_j$ links up two adjacent points $(\boldsymbol{x}_j, \boldsymbol{x}_{j+1})$, of which the linear interpolation is denoted as:

$$\boldsymbol{e}_j = \boldsymbol{x}_j + \lambda\,(\boldsymbol{x}_{j+1} - \boldsymbol{x}_j) = \boldsymbol{x}_j + \lambda\;\boldsymbol{u}. \tag{7}$$

The point-edge distance is defined as,

$$D\left(\boldsymbol{p}_i, \boldsymbol{e}_j\right) = \begin{cases} \frac{|\boldsymbol{u} \times (\boldsymbol{p}_i - \boldsymbol{x}_j)|}{|\boldsymbol{u}|}, \text{if } 0 < \frac{\langle \boldsymbol{u}, (\boldsymbol{x}_j - \boldsymbol{p}_i) \rangle}{|\boldsymbol{u}|^2} < 1 \\ \infty \qquad\quad , \text{else} \end{cases}. \tag{8}$$

The closet point $\boldsymbol{p}_j^{\text{cand}}$ is called candidate point for inclusion and is defined as:

$$\boldsymbol{p}_j^{\text{cand}} = \min_{\boldsymbol{p}_i} \left(D\left(\boldsymbol{p}_i, \boldsymbol{e}_j\right)\right) \tag{9}$$

The use of infinity in the case differentiation of (8) ensures that only those points are considered for inclusion which lie in the light green corridor in Fig. 1 (c). Thus, this ensures that the point on $\boldsymbol{e}_j$ closest to $\boldsymbol{p}_i$ lies between $\boldsymbol{x}_j$ and $\boldsymbol{x}_{j+1}$. The candidate point is included if:

$$D(\boldsymbol{p}_j^{\text{cand}}, \boldsymbol{e}_j) < \frac{|\boldsymbol{u}|}{k}, \tag{10}$$

where k is the parameter which determines how deep the algorithm "digs" the boundary into the convex hull. The smaller k, the more detailed the hull geometry can be deduced by the algorithm. Values of $k \leq 1$ are not reasonable since both new edges would be longer than the considered one. The parameter k can be adjusted during the procedure, based on visualization of the resultant concave hull. This adjustment depends on clinical requirements, for instance, how detailed the constraint morphology needs to be prescribed for safe tool guidance. The resulting concave hull contains the points representing the contour of $\mathbf{S}$, they are stored in order and can be transferred into 3D space using (4). An exemplary result is depicted in Fig. 1 (d). The next step is to generate an implicit tunnel or segment by connecting two adjacent $\textbf{CCH}_{\text{I}}$s originating from two different point sets.

B. Calculating a Tunnel from Concave Contours

$\mathbf{CCH}_{\mathrm{I}}$ and $\mathbf{CCH}_{\mathrm{II}}$ denote two concave hulls obtained by the previous algorithm. The two piecewise linear contours are first interpolated with the aim to obtain the same number of sample points $N_\circ$ on both contours while satisfying a maximum mesh lattice distance of $s^{\mathrm{lat}}_{\max}$. The number of points ($\boldsymbol{x}_i$) in the concave hulls is N_{I} and N_{II} respectively. The lengths of the contours are:

$$L_{\{\mathrm{I,II}\}} = \sum_{i=0}^{N_{\{\mathrm{I,II}\}}-1} \left| \boldsymbol{x}_i^{\{\mathrm{I,II}\}} - \boldsymbol{x}_{i+1}^{\{\mathrm{I,II}\}} \right|. \tag{11}$$

And the number of sample points for both contours can be expressed as:

$$N_\circ = \left\lceil \max\left((L_{\mathrm{I}}, L_{\mathrm{II}}) / s^{\mathrm{lat}}_{\max} \right) \right\rceil, \tag{12}$$

which yields to the corresponding mesh lattices:

$$s^{\mathrm{lat}}_{\{\mathrm{I,II}\}} = L_{\{\mathrm{I,II}\}} / N_\circ. \tag{13}$$

Each of the hulls is then linearly interpolated with the respective mesh lattice calculated in (13). To connect the two interpolated concave hulls $\mathbf{CCH}^{\mathrm{ip}}_{\mathrm{I}}$, $\mathbf{CCH}^{\mathrm{ip}}_{\mathrm{II}}$, the corresponding points on the hulls have to be found. Since the points in $\mathbf{CCH}^{\mathrm{ip}}_{\{\mathrm{I,II}\}}$ are in order, the task is simplified to finding only one correspondence. Subsequent correspondences are established by iterating simultaneously through both interpolated hulls. To find a good corresponding point pair ($\boldsymbol{z}_{\mathrm{I}}, \boldsymbol{z}_{\mathrm{II}}$), the vector connecting both centroids $\mathbf{c}^{\mathrm{ip}}_{\{\mathrm{I,II}\}}$ of $\mathbf{CCH}^{\mathrm{ip}}_{\{\mathrm{I,II}\}}$ is calculated:

$$\tilde{\boldsymbol{n}}^{\mathrm{cor.}} = \mathbf{c}^{\mathrm{ip}}_{\mathrm{II}} - \mathbf{c}^{\mathrm{ip}}_{\mathrm{I}}. \tag{14}$$

The center is located at the middle of the two centroids:

$$\boldsymbol{r}_0^{\mathrm{cor.}} = \left(\mathbf{c}^{\mathrm{ip}}_{\mathrm{II}} + \mathbf{c}^{\mathrm{ip}}_{\mathrm{I}} \right) / 2. \tag{15}$$

A new intermediate plane is then defined as:

$$\langle \boldsymbol{n}^{\mathrm{cor.}}, (\boldsymbol{r} - \boldsymbol{r}_0^{\mathrm{cor.}}) \rangle = 0; \; \boldsymbol{n}^{\mathrm{cor.}} = \tilde{\boldsymbol{n}}^{\mathrm{cor.}} / \left| \tilde{\boldsymbol{n}}^{\mathrm{cor.}} \right|. \tag{16}$$

As done in Sec. II-A, both interpolated contours are projected onto the plane defined in (16) using (5). To transform the projected points into 2D, the two vectors spanning the plane determined in (16) are required. They can be calculated as:

$$\boldsymbol{v}_1^{\mathrm{cor.}} = \sqrt{\left(n_x^{\mathrm{corr}}\right)^2 + \left(n_y^{\mathrm{cor.}}\right)^2}^{-1} \left[-n_y^{\mathrm{cor.}}, n_x^{\mathrm{cor.}}, 0 \right]^T, \tag{17}$$

$$\boldsymbol{v}_2^{\mathrm{cor.}} = \boldsymbol{n}^{\mathrm{cor.}} \times \boldsymbol{v}_1^{\mathrm{cor.}}, \tag{18}$$

where $\boldsymbol{n}^{\mathrm{cor.}}$, $\boldsymbol{v}_1^{\mathrm{cor.}}$, $\boldsymbol{v}_2^{\mathrm{cor.}}$ represent an orthonormal basis. The points of $\mathbf{CCH}^{\mathrm{ip}}_{\{\mathrm{I,II}\}}$, which were projected onto the intermediate plane, are then further transformed into the two dimensional s, t-space applying (6) based on the calculated unit vectors $\boldsymbol{v}_1^{\mathrm{cor.}}$, $\boldsymbol{v}_2^{\mathrm{cor.}}$. The first corresponding point pair ($\boldsymbol{z}^{\mathrm{st}}_{\mathrm{I}}, \boldsymbol{z}^{\mathrm{st}}_{\mathrm{II}}$) therefore, can be found in s, t-space. The two points ($\boldsymbol{z}^{\mathrm{st}}_{\mathrm{I}}, \boldsymbol{z}^{\mathrm{st}}_{\mathrm{II}}$) within the two point sets ($\boldsymbol{x}_i^{\mathrm{st,I}}$, $\boldsymbol{x}_i^{\mathrm{st,II}}$) are determined as:

$$\boldsymbol{z}^{\mathrm{st}}_{\{\mathrm{I,II}\}} = \max_i \left(\left\langle \boldsymbol{x}_i^{\mathrm{st},\{\mathrm{I,II}\}}, \boldsymbol{e}_{\{s,t\}} \right\rangle \right), \tag{19}$$

where $\boldsymbol{e}_s = [1, 0]^T$ and $\boldsymbol{e}_t = [0, 1]^T$. Although the maximization in (19) can be conducted in either the s- or t-coordinate, we suggest the one in which the points have the greater expansion. This can be determined by performing a SVD and using the coordinate in which the eigenvector corresponding to the highest eigenvalue has the higher value.

Thus, the corresponding points in 3D $\boldsymbol{z}_{\{\mathrm{I,II}\}}$ are calculated through the indirect route over $\boldsymbol{z}^{\mathrm{st}}_{\{\mathrm{I,II}\}}$. Before the iterative process for searching correspondence point pairs is started, both $\mathbf{CCH}^{\mathrm{ip}}_{\mathrm{I}}$ and $\mathbf{CCH}^{\mathrm{ip}}_{\mathrm{II}}$ have to be ensured not to turn in anti-parallel directions whilst iterating through the lists. Let $\boldsymbol{z}^{+}_{\mathrm{I}}$ and $\boldsymbol{z}^{+}_{\mathrm{II}}$ succeed $\boldsymbol{z}_{\mathrm{I}}$ and $\boldsymbol{z}_{\mathrm{II}}$ in the respective point lists $\mathbf{CCH}^{\mathrm{ip}}_{\mathrm{I}}$ or $\mathbf{CCH}^{\mathrm{ip}}_{\mathrm{II}}$, and let $\boldsymbol{z}^{-}_{\mathrm{I}}$ and $\boldsymbol{z}^{-}_{\mathrm{II}}$ be the respective predecessors. Then the turn normal when iterating through the concave hulls is defined as,

$$\boldsymbol{n}^{\mathrm{turn}}_{\{\mathrm{I,II}\}} = (\boldsymbol{z}_{\{\mathrm{I,II}\}} - \boldsymbol{z}^{+}_{\{\mathrm{I,II}\}}) \times (\boldsymbol{z}_{\{\mathrm{I,II}\}} - \boldsymbol{z}^{-}_{\{\mathrm{I,II}\}}). \tag{20}$$

The concave hulls will be turned in the same direction, if

$$\left\langle \boldsymbol{n}^{\mathrm{turn}}_{\mathrm{I}}, \boldsymbol{n}^{\mathrm{turn}}_{\mathrm{II}} \right\rangle > 0. \tag{21}$$

If (21) is true the correspondences are determined by incrementing the point list of both concave hulls, if not, one point list is incremented while the other point list is decremented.

Once all correspondences are established, a linear interpolation between each point pair is performed to create the tunnel. The number of interpolation points is calculated for each point pair separately such that a maximum lattice distance of $s^{\mathrm{lat}}_{\max}$ is ensured. Fig. 2 shows an example constraint tunnel constructed based on the two point sets which are depicted in Fig. 1 (a).

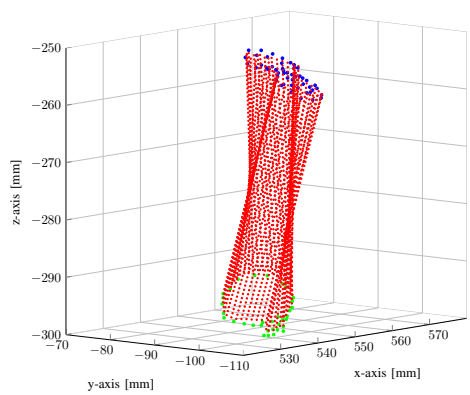

Fig. 2. Example Implicit Concave Tunnel.

With multiple point sets describing multiple cross-sections, the connection of two concave contours, as well as the distance between points on the contour itself, can be interpolated with splines, instead of lines. An example of such a spline tunnel is depicted in Fig. 3.

The algorithm presented in this section allows to generate implicit concave constraint tunnels, which can be used to define patient specific tubular constraints, which is particularly useful for flexible access surgery involving curved access pathways. The next section will propose a control framework for both static and dynamic active constraints, which can make use of tunnels generated with this algorithm.

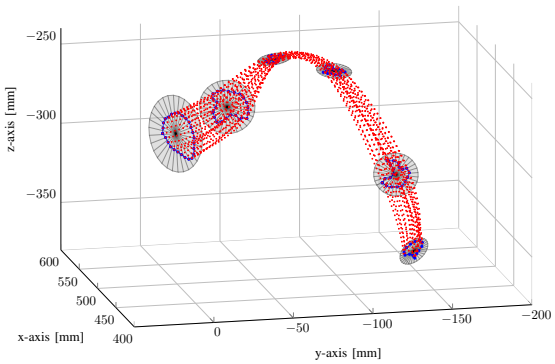

Fig. 3. Example Implicit Concave Tunnel outlined by the spline interpolation of the contour sets.

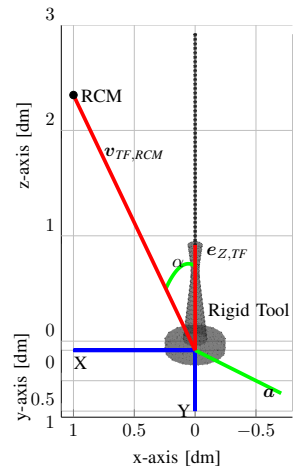

Fig. 4. Definition of the reference frames for the RCM constraint.

III. Dynamic Active Constraint Framework

In MIS, surgical tools are inserted through the incision port into the surgical workspace. This poses challenges for the surgeon since the tool tip motion is inverted to the hand motion due to the fulcrum effect. Lateral motion at the incision port has to be minimized to avoid inadvertent tissue damage. Mechanically constrained systems were developed to ensure a remote center of motion (RCM) [15]. Software based RCM solutions were also proposed for telemanipulation [16] as well as for Hands-on manipulation [17]. Besides RCM constraints, which represent guidance constraints in regards to the tool orientation, position constraints ensure compliance with forbidden-region or path following. We take up again the approach of [17] regarding constraining the RCM and extend the framework in regards to entire shaft consideration in the position constraint, which introduces some control issues. Our framework is deduced under the impedance control where the torque ($\boldsymbol{\tau}$) introduced by the articulated robotic manipulator is:

$$\boldsymbol{\tau} = \boldsymbol{J}(q)^T \left(\boldsymbol{K}_m(\boldsymbol{x}_d - \boldsymbol{x}(q)) + \boldsymbol{D}_m \dot{\boldsymbol{x}}(q) + \boldsymbol{F}_a\right) + \boldsymbol{f}^*(q, \dot{q}, \ddot{q}). \tag{22}$$

In (22), $\boldsymbol{J}$: denotes the Jacobi matrix, $\boldsymbol{K}_m$, $\boldsymbol{D}_m$: represent the stiffness and damping matrices, $\boldsymbol{x}_d$ and $\boldsymbol{x}$: denote the 6 DoF desired and current tool pose respectively in Cartesian space, $\boldsymbol{F}_a$: describes additional introduced forces, and $\boldsymbol{f}^*$ represents the force introduced by gravity and due to the dynamic properties of the manipulator.

A. Remote Center of Motion Constraint

The basic principle of guidance RCM constraint, as presented below, is to define a tool orientation (roll, pitch, yaw) in $\boldsymbol{x}_d$ which ensures correct alignment with the incision point according to the current position (x,y,z) in $\boldsymbol{x}$. Let ${}^0\boldsymbol{T}_{TF}$ be the homogeneous transformation between world frame and tool frame (*TF*). The tool frame is defined such that its origin lies where the tool shaft starts and that shaft and z-direction are parallel. If there is a bendable tool tip, the x-direction of *TF* is chosen to be perpendicular to the bending plane of the tool.

A unit vector $\boldsymbol{v}_{TF,RCM}$ is defined. Its direction points from *TF* towards the fulcrum point (RCM). Along with the z-direction of the tool, $\boldsymbol{e}_{Z,TF}$, the rotation matrix ${}^{TF}\boldsymbol{R}_{TF,d}$ can be deduced. This matrix allows to find the desired tool orientation ${}^0\boldsymbol{R}_{TF,d}$ such that the tool shaft and the RCM are perfectly aligned. ${}^{TF}\boldsymbol{R}_{TF,d}$ is calculated using the angle-axis representation, where,

$$\boldsymbol{a} = \frac{\boldsymbol{e}_{Z,TF} \times \boldsymbol{v}_{TF,RCM}}{|\boldsymbol{e}_{Z,TF} \times \boldsymbol{v}_{TF,RCM}|} \tag{23}$$

and,

$$\alpha = \arctan\left(\frac{|\boldsymbol{e}_{Z,TF} \times \boldsymbol{v}_{TF,RCM}|}{\langle \boldsymbol{e}_{Z,TF}, \boldsymbol{v}_{TF,RCM}\rangle}\right) \tag{24}$$

denote the rotation axis and rotation angle respectively, see Fig. 4. The required ${}^0\boldsymbol{R}_{TF,d}$ is then calculated as,

$${}^0\boldsymbol{R}_{TF,d} = {}^0\boldsymbol{R}_{TF}\ {}^{TF}\boldsymbol{R}_{TF,d}. \tag{25}$$

The desired tool pose $\boldsymbol{x}_d$ is calculated in (25), keeping the position in $\boldsymbol{x}_d$ and $\boldsymbol{x}$ equal. With the control objective in (22) the corresponding constraint torque is evaluated to realign the end-effector with the RCM.

B. Constraint Evaluation for the Entire Tool Shaft

To evaluate the tool constraints, we use the proximity query algorithm presented in [3]. The algorithm calculates the shortest distance between constraint points and a generalized cylinder which represents the tool shaft. For each point in a cloud ($\boldsymbol{p}_i^{\mathrm{c}}$), representing the constraint surface the algorithm determines the closest point on the tool shaft ($\boldsymbol{p}_i^{\mathrm{t}}$). The tool shaft is represented by multiple generalized cylindrical segments. Each segment is enclosed by two adjacent contours. The contour center is located at $\boldsymbol{p}_i$ with the normal direction $\boldsymbol{n}_i$. Scalar r_i is the contour radius, as shown in Fig. 5. The positions of $\boldsymbol{p}^{\mathrm{c}}$ and $\boldsymbol{p}^{\mathrm{t}}$ are sufficient to enforce the constraint for the entire tool shaft.

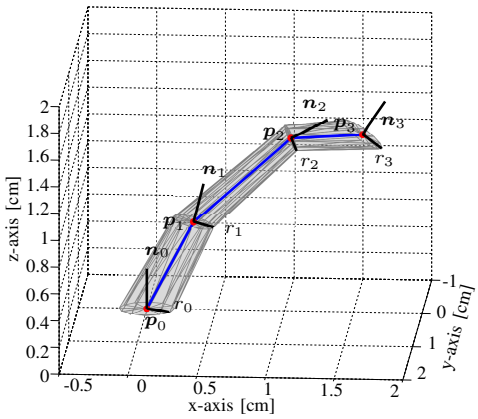

Fig. 5. Geometric parameters representing tool configuration and boundary for the PQ computation proposed in [3].

C. Constraint Enforcement for the Entire Tool Shaft

In the existing literature, point clouds become a common representation of object boundaries, to constraint the motion of the tool tip, due to their geometric simplicity, [17], [18]. Using the in Sec. III-B described constraint evaluation method we are able to adapt *in situ* to flexible constraints. Furthermore, we address an issue regarding discrete constraints which becomes important when constraining the tool shaft.

The basic approach to calculate constraint forces is to introduce a transition zone of thickness ϵ so that a force on the tool is introduced when the tool enters the transition zone and increases until the tool reaches the constraint. Using a point cloud as constraint, the discrete force ($\boldsymbol{f}_i^\delta$) for each point in the cloud ($\boldsymbol{p}_i^{\mathrm{c}}$) is calculated as:

$$\boldsymbol{f}_i^\delta = \begin{cases} k\ \left(\epsilon - |\boldsymbol{p}_i^{\mathrm{t}} - \boldsymbol{p}_i^{\mathrm{c}}|\right) \frac{\boldsymbol{p}_i^{\mathrm{t}} - \boldsymbol{p}_i^{\mathrm{c}}}{|\boldsymbol{p}_i^{\mathrm{t}} - \boldsymbol{p}_i^{\mathrm{c}}|}, \text{if } |\boldsymbol{p}_i^{\mathrm{t}} - \boldsymbol{p}_i^{\mathrm{c}}| < \epsilon \\ \mathbf{0} \qquad\qquad\qquad\qquad , \text{else} \end{cases} \tag{26}$$

where $\boldsymbol{p}_i^{\mathrm{t}}$ is the corresponding closest point on the tool and k is the stiffness parameter.

Varying numbers of points (N_f) will take effect to prescribe the resistant force in the transition zone. $\boldsymbol{F}_a$ in (22) cannot be just summarized as $\boldsymbol{F}_a = \sum_i \boldsymbol{f}_i^\delta$, since this aggregation highly depends on the point cloud density which is likely not uniform. Therefore, an option would be to calculate the resulting force taking the number $N_f = \left|\left\{\boldsymbol{f}_i^\delta > 0\right\}\right|$ into account. An averaging:

$$\boldsymbol{F}_a = \frac{\sum_i \boldsymbol{f}_i^\delta}{N_f}, \tag{27}$$

seems natural and was done in [18]. However, when considering the entire tool shaft an issue depicted in Fig. 6 should be considered. In situations when most of the tool shaft has a distance to the constraint of approximately ϵ, but particular small parts of the tool shaft are closely proximal to the constraint, the force, which would be appropriate for these closely proximal parts, is weakened by the further distant cloud points. An example of such a situation depicted

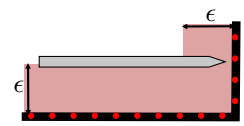

Fig. 6. Constraint-tool distance is approximately ϵ for most of the shaft length and very small for the tool tip.

in Fig. 6 is likely to occur, particularly when the tool is constrained by a convoluted tunnel.

Thus, to overcome this problem a maximization could be used,

$$\boldsymbol{F}_a = \max_{|\boldsymbol{f}_i^\delta|} \left(\boldsymbol{f}_i^\delta\right). \tag{28}$$

The problem with this approach, which relies on the calculation of the closest point pair, is that C^0 continuity of $\boldsymbol{F}_a$ would not be fulfilled. Without loss of generality we represent here the force in 2D polar coordinates such that,

$$\boldsymbol{F}_a = \begin{bmatrix} \mathrm{Re}\left(|\boldsymbol{F}_a|\, e^{i\ \phi(\boldsymbol{F}_a)}\right) \\ \mathrm{Im}\left(|\boldsymbol{F}_a|\, e^{i\ \phi(\boldsymbol{F}_a)}\right) \end{bmatrix}. \tag{29}$$

To illustrate this loss of continuity in Fig. 7 (a) a proxy is moved from left to right approximately parallel to a constraint. We can calculate the force with (28) in the polar representation. The result is depicted in Fig. 7 (b). As shown

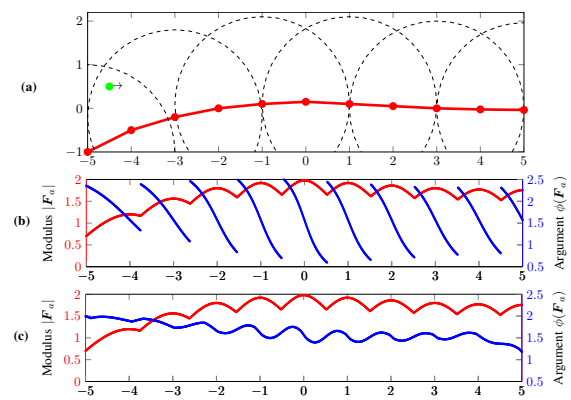

Fig. 7. Illustration of constraint force aggregation problem. (a): dashed line: representing ϵ, (green) proxy which is moved from left to right, (red) constraint. (b)/(c): (red) resulting force modulus $|\boldsymbol{F}_a|$, (blue) resulting force argument $\phi(\boldsymbol{F}_a)$. (b) calculated according to (28). (c) calculated according to (30).

in Fig. 7 (b) the argument of the force is not continuous. To overcome this issue we propose to calculate the force as:

$$\boldsymbol{F}_a = \max_i \left(\left|\boldsymbol{f}_i^\delta\right|\right) \frac{\sum_i \boldsymbol{f}_i^\delta}{\left|\sum_i \boldsymbol{f}_i^\delta\right|}, \tag{30}$$

which results in force continuity in modulus ($|\boldsymbol{F}_a|$) and argument ($\phi(\boldsymbol{F}_a)$) as shown in Fig. 7 (c).

The advantage of having (30) is that the discrete forces can be aggregated without having to attain a discontinuous force profile. Furthermore it reduces the risk of weakening the force in critical situations. This is particularly important when considering the entire tool shaft and by avoiding dense point clouds, which require much computing resources [18].

Two experiments were carried out to demonstrate how the proposed active constraint framework potentially improves a surgical task and thus outperforms the constraint- and guidance-free manipulation.

IV. Experiments

A. Neurosurgical Simulation

The first experiment simulates a keyhole transcranial endoscope-assisted microsurgical approach, the supraorbital subfrontal approach. This approach utilises a short cosmetically favorable incision through the eyebrow, a keyhole craniotomy approximately 20-30 mm in diameter and, most importantly, reduces the exposure and manipulation of unaffected brain tissue, which can in turn reduce the risk of post-operative complications such as stroke [19], [20]. The difficulty with such approaches is that they utilize a single entry point, a very narrow surgical corridor, and a small working space. At present the technical challenges associated with keyhole transcranial approaches provide a significant barrier to their use, particularly for neurosurgeons unfamiliar with such techniques. Here, we apply active constraints in a preclinical model, using the approach presented in Sec. II. The objective of this experiment is to show that active constraints can significantly increase the patient safety while not worsening other performance criteria.

1) Participants, Setup, Results: Ten healthy volunteers (one female) took part in the experiment. All participants were novices. The mean age was 27 years (range 22 - 36 years). Using a CT scan of a brain phantom the cross sections for the tunnel were defined preoperatively. The experimental setup is depicted in Fig. 8. A rigid shaft mounted to a KUKA LWR 4+ compliant manipulator (KUKA Roboter GmbH, Germany) was used. Three targets were placed in the brain to which the tunnel was leading. The task was to touch all targets centrally (in a random order). Furthermore the participants were asked to consider a 'soft' RCM (radius: 1 cm) at the craniotomy site. The intra-operative image was provided using a 3D endoscope (Visionsense Ltd., Israel) and a 3D display. The participants were randomized into two groups (active constraints and free manipulation), and then at the end crossed over. All participants had time to get familiar with the robotic system and both experimental modes. Seven performance metrics were used to quantify the task performance: I. time to complete the task (duration), II. travel distance of the tool tip, III. targeting precision, defined as closest distance between target and tool tip, IV. maximum deviation distance of RCM and tool shaft, V. cumulative time penetrating the constraint, VI. maximum distance of constraint penetration, and VII. penetration distance over time. The resulting mean values of these metrics are summarized in Table I.

In the first column of Table I the performance metrics and their respective units are listed. The second column shows the mean values of the respective performance metrics when the task was performed with active constraints. The third column shows the mean when the task was performed unconstrained. The p-value was calculated using the Wilcoxon rank sum test, since the performance metrics were not normally distributed according to the Kolomogrov-Smirnov test.

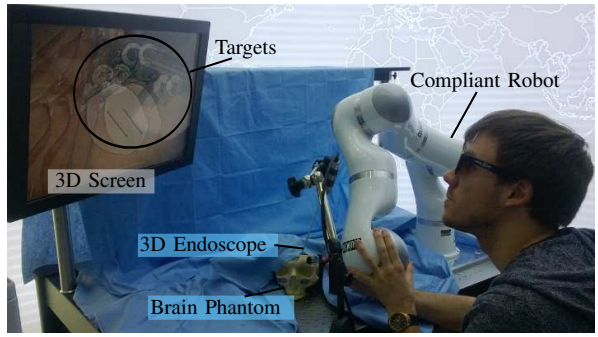

Fig. 8. Simulated neurosurgical procedure visualized by a stereoscope.

TABLE I

Mean values and p-values of performance metrics of the neurosurgical simulation.

Metric	Const.	Free	p-value
Task duration [s]	39.83	9.14	0.91
Tool tip travel distance [m]	0.51	0.48	0.68
Targeting precision [mm]	0.99	0.95	0.68
Max. deviation from RCM [mm]	8.23	12.06	<0.01*
Duration of constraint penetration [s]	0.0	1.43	<0.01*
Max. constraint penetration [mm]	0.0	3.16	<0.01*
Penetration over time [mm s]	0.0	2.19	<0.01*

2) Discussion: The objective of the experiment was to increase the safety in a keyhole transcranial endoscope-assisted microsurgery while not worsening other performance criteria. The first three performance metrics do not show a significant difference, thus the performance can be considered as equal. However, the safety metrics do demonstrate a significant improvement using the presented active constraint framework. Using active constraints none of the participants penetrated the safety zone, in comparison to an average max. penetration of 3 mm in the control group. Furthermore the participants deviated significantly less from the RCM using active constraints.

B. Arthroscopic Simulation

In the second experiment, the dynamic component of the presented framework was evaluated. In arthroscopy, the surgeon has to examine the entire knee cavity, and does so by flexing and extending the knee, thus providing an example of a dynamic surgical environment. The objective of this experiment is to show that dynamic active constraints increase safety and furthermore increase precision due to the haptic guidance in the moving environment.

1) Participants, Setup, Results: Eleven healthy volunteer (two female) participated in the experiment. All participants were novices regarding this procedure. The mean age was 27 years (range 22 - 36 years).

To simulate this the tibia and femur of a knee phantom CT scan were segmented to represent the constraint surface. Optical markers (Polaris Vicra, Northern Digital Inc., Canada)

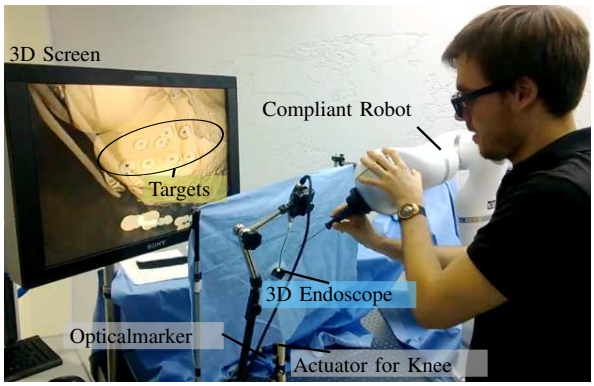

Fig. 9. Setup of a simulated arthroscopic procedure with a non-stationary knee model.

were attached to tibia and femur and registered in the robot coordinate frame. Four targets were placed on the tibia and also registered in the robot coordinate as well as in the virtual model reference frame. The point clouds representing tibia and femur were rigidly-transformed using the information of the optical tracking system (update rate: 20 Hz). When pressing a pedal the knee was flexed by activating a linear motor (Faulhaber GmbH & Co. KG, Germany) which pushed the tibia from an angled configuration upwards. Another pedal activation lowered the knee. A cloth with two holes was placed in front of the knee blocking the direct view on the knee. A 3D endoscope was inserted through the first hole of the cloth to facilitate the intra-operative view on the knee. The rigid shaft mounted to the compliant manipulator was inserted through the other hole to enable touching of the targets. The task was to place the tool tip 3 mm above the first target and to activate the rising of the knee. While the tibia rose the participants were asked to keep the distance to it constant. The participants had to follow first target one and three in an upwards movement and target two and four in a downwards movement, and in a second round *vice versa*. Thus, all four targets had to been followed in an up- and downwards movement. The participants were randomized into two groups (active constraints and free manipulation), and then at the end crossed over. The experimental setup is depicted in Fig. 9. Seven performance metrics were used to quantify the task performance. I. time to complete the task (duration), II. travel distance of the tool tip, III. target following precision, representing the mean distance between target and tool tip during the movement, IV. tibia following precision, representing the mean distance between tibia and tool tip, V. cumulative time penetrating the constraint, VI. maximum distance of constraint penetration, and VII. penetration distance over time. The resulting mean values of these metrics are summarized in Table II.

The first column of Table II states the respective performance metrics, and the second and third column the mean values of the metrics of constrained and unconstrained mode respectively. The p-values are calculated with the Wilcoxon rank sum test, since the performance metrics were not normal distributed according to the Kolomogrov-Smirnov test.

TABLE II

MEAN VALUES AND P-VALUES OF PERFORMANCE METRICS OF THE ARTHROSCOPIC SIMULATION.

Metric	Const.	Free	p-value
Task duration [s]	57.06	54.44	0.970
Tool tip travel distance [m]	19.73	19.84	0.623
Target following precision [mm]	3.34	6.71	0.026*
Tibia following precision [mm]	1.29	3.98	0.038*
Duration of constraint penetration [s]	3.79	13.43	0.044*
Max. constraint penetration [mm]	1.28	4.06	0.018*
Penetration over time [mm s]	7.68	26.41	0.044*

2) Discussion: The objective of this experiment was to increase safety and precision when using dynamic active constraints in comparison to free-hand manipulation. To this end, we found significantly reduced duration and distance of penetration into the no-go area in the active constraint arm, suggesting improved safety. Moreover, following the target or the tibia was also significantly improved (p=0.026 and p=0.038 respectively), suggesting improved precision.

V. CONCLUSION

The concept of MIS has been widely adopted within general surgery and is associated with reduced post-operative pain, fewer complications, and a shorter length-of-stay in hospital. In fields such as keyhole neurosurgery, however, the consequences of veering away from a narrow and defined surgical trajectory, and injuring unaffected brain tissue, can result in disability or death. The corollary is that technological interventions that provide feedback and reduce the risk of breaching a safe surgical corridor, can significantly impact on patient outcome. To this end, this paper presents a concept to implicitly generate active constraint tunnels, which can be adapted to patient specific anatomy and thus help to increase the safety when operating in those narrow corridors. The conducted neurosurgical experiment showed that active constraints can assist to avoid penetrating forbidden regions. Furthermore, a framework for dynamic active constraints is presented which can be used in dynamic environments, and has been validated in a preclinical arthroscopy study. In an arthroscopic scenario the benefits of the guiding character of dynamic active constraints were shown. Safety as well as precision were improved. The two concepts of implicit tunnels and dynamic active constraints can be combined to, for example, guide catheters through the dynamic cardiac vascular tree.

ACKNOWLEDGMENT

This work was supported in part by an Institute of Global Health Innovation - Imperial College London studentship and by Medical Engineering Solutions in Osteoarthritis Centre of Excellence [088844/Z/09/Z] from the Wellcome Trust and the EPSRC.

REFERENCES

[1] S. A. Bowyer, B. L. Davies, and F. Rodriguez y Baena, "Active constraints/virtual fixtures: A survey," *IEEE Transactions on Robotics*, 2013, *to be published*.

[2] L. B. Rosenberg, "Virtual fixtures: Perceptual tools for telerobotic manipulation," in *IEEE Virtual Reality Annual International Symposium, 1993*, pp. 76–82.

[3] K. W. Kwok, K. H. Tsoi, V. Vitiello, *et al.*, "Dimensionality reduction in controlling articulated snake robot for endoscopy under dynamic active constraints," *IEEE Transactions on Robotics*, vol. PP, no. 99, pp. 1–17, 2012.

[4] C. Passenberg, A. Peer, and M. Buss, "A survey of environment-, operator-, and task-adapted controllers for teleoperation systems," *Mechatronics*, vol. 20, no. 7, pp. 787 – 801, 2010.

[5] B. L. Davies, S. J. Harris, W. J. Lin, *et al.*, "Active compliance in robotic surgery - the use of force control as a dynamic constraint," *Proceedings of the Institution of Mechanical Engineers, Part H*, vol. 211, no. 4, pp. 285–292, 1997.

[6] M. Li, M. Ishii, and R. H. Taylor, "Spatial motion constraints using virtual fixtures generated by anatomy," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 4–19, 2007.

[7] B. Becker, R. MacLachlan, G. Hager, *et al.*, "Handheld micromanipulation with vision-based virtual fixtures," in *IEEE International Conference on Robotics and Automation (ICRA), 2011* , pp. 4127–4132.

[8] C. J. Payne, K.-W. Kwok, and G.-Z. Yang, "An ungrounded hand-held surgical device incorporating active constraints with force-feedback," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2013* , *to be published*.

[9] J. W. Park, J. Choi, Y. Park, *et al.*, "Haptic virtual fixture for robotic cardiac catheter navigation," *Artificial Organs*, vol. 35, no. 11, pp. 1127–1131, 2011.

[10] A. Bettini, P. Marayong, S. Lang, *et al.*, "Vision-assisted control for manipulation using virtual fixtures," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 953–966, 2004.

[11] R. Jing, R. V. Patel, K. A. McIsaac, *et al.*, "Dynamic 3-d virtual fixtures for minimally invasive beating heart procedures," *IEEE Transactions on Medical Imaging*, vol. 27, no. 8, pp. 1061–1070, 2008.

[12] H. J. Marcus, C. A. Seneci, C. J. Payne, *et al.*, "Robotics in keyhole transcranial endoscope-assisted microsurgery: A critical review of existing systems and proposed specifications for new robotic platforms." *Neurosurgery*, 2013.

[13] R. L. Graham, "An efficient algorith for determining the convex hull of a finite planar set," *Information processing letters*, vol. 1, no. 4, pp. 132–133, 1972.

[14] J.-S. Park and S.-J. Oh, "A new concave hull algorithm and concaveness measure for n-dimensional datasets," *Journal of information science and engineering*, vol. 28, no. 3, pp. 587–600, 2012.

[15] A. Madhani, G. Niemeyer, and J. Salisbury, "The black falcon: a teleoperated surgical instrument for minimally invasive surgery," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1998*, vol. 2, pp. 936–944.

[16] U. Hagn, R. Konietschke, A. Tobergte, *et al.*, "DLR mirosurge: a versatile system for research in endoscopic telesurgery," *International journal of computer assisted radiology and surgery*, vol. 5, no. 2, pp. 183–193, 2010.

[17] E. Lopez, K.-W. Kowk, C. J. Payne, *et al.*, "Implicit active constraints for robot-assisted arthroscopy," in *IEEE International Conference on Robotics and Automation, 2013*, pp. 5370–5375.

[18] F. Rydén and H. Chizeck, "Forbidden-region virtual fixtures from streaming point clouds: Remotely touching and protecting a beating heart," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012*, pp. 3308–3313.

[19] A. Perneczky and R. Reisch, *Keyhole Approaches in Neurosurgery: Concept and surgical technique*. Springer, 2008, vol. 1.

[20] R. Reisch, H. Marcus, N. O. Koechlin, *et al.*, "Transcranial endoscope-assisted keyhole surgery: anterior fossa," *Innovative Neurosurgery*, vol. 1, no. 2, pp. 77–89, 2013.