DOI:
10.1109/TRO.2023.3275381

Document Version
Peer reviewed version

Designing Robots for Reachability and Dexterity
Continuum Surgical Robots as a Pretext Application

Konrad Leibrandt, Lyndon da Cruz, Christos Bergeles, *Senior Member, IEEE*

***Abstract*—This paper contributes a novel method to assess robot dexterity. Existing Jacobian-based dexterity metrics, such as the manipulability index or the condition number, do not allow for comparisons between robot architectures, are local in nature, and are affected by robot dimensions (robot size). On the contrary, the introduced metric is global and allows for quantitative comparisons of robot architectures as it explicitly incorporates the orientational and positional coverage of a robot's end-effector. Experiments presented show that the proposed dexterity metric can improve the computational and precision performance of numerical inverse kinematics, and showcase its suitability for use in computational dexterous robot design, and in particular for designing concentric tube robots with high orientational and positional dexterity.**

***Index Terms*—Workspace Analysis, Design Optimization, Concentric Tube Robot, Flexible Manipulator, Medical Robotics.**

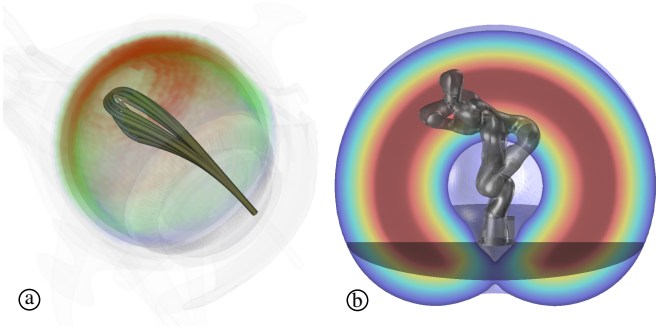

Fig. 1. A continuum and a serial manipulator reaching a defined 3D position with different tip orientations: a.) A concentric tube robot pivoting at a dexterous position within its workspace, in a vitreoretinal surgical scenario. b.) A serial-link manipulator reaching the same position with three different end-effector orientations. Red regions indicate increased robot tip dexterity.

I. Introduction

DESIGNING bespoke surgical robots or evaluating them for fitness to a task is gaining attention due to the increased demand for intervention-specific systems. The research presented in this article is motivated by the desire to design systems that maximise surgical dexterity given task and anatomy considerations. While the developed theoretical framework is applicable across the spectrum of robotics, the particular focus lies in the optimised design of continuum surgical robots such as concentric tube robots (CTRs) [1], [2], recently reviewed in [3], [4].

Purposefully designed CTRs can safely navigate the human anatomy to reach deep-seated pathologies while offering increased tip dexterity at the surgical site. This customisation possibility renders them promising for a variety of surgical scenarios ranging from bronchoscopic procedures [5] to neurosurgery [6], renal surgery [7], transurethral prostate surgery [8], cardiac surgery [9], olfactory cell sampling [10], and vitreoretinal surgery [11]–[13]. While most existing research is evaluated on phantoms, notable *in vivo* demonstrations have also emerged [14], [15] to demonstrate the increased maturity of the technology.

K. Leibrandt was with the Wellcome/EPSRC Centre for Interventional and Surgical Engineering Sciences, University College London, London, United Kingdom, e-mail: research@leibrandt.eu.

L. da Cruz is with Moorfields Eye Hospital, London, United Kingdom. C. Bergeles is with the School of Biomedical Engineering & Imaging Sciences, King's College London, London, United Kingdom.

This work was supported by the National Institute for Health Research (Invention for Innovation, i4i; II-LB-0716-20002, and NIHR202879). The views expressed are those of the author(s) and not necessarily those of the NHS, the National Institute for Health Research or the Department of Health and Social Care. Part of this work was supported by an ERC Starting Grant [714562].

The state-of-the-art in CTR design [6], [7], [16], [17] has mainly considered robot-tip reachability in 3D and conformance to the anatomy as indicators of design suitability. Within this paper, metrics to evaluate and optimise robot tip dexterity are introduced, enabling the use of CTR as nimble intraluminal manipulators, see Fig. 1(a). The described methodology is applicable to any continuum manipulator, *e.g.* surgical continuum robots as reviewed in [18], or serial-link robots, see Fig. 1(b).

In the following paragraphs, metrics considered for robot design optimisation, and the state-of-the-art in computational robot design are summarised.

A. Workspace Analysis

Common metrics that characterise a manipulator's performance are workspace reachability and dexterity considering actuator limits and potentially collisions with the environment. Algorithmic evaluation of these metrics, usually via the robot end-effector Jacobian matrix [19]–[21], allows to assess manipulator suitability to a specific task-space objective, *e.g.* reaching a set of target end-effector poses. Individual point reachability can then be considered in the context of robot configuration evaluation and path planning. While given a robot configuration, the Jacobian can provide information on tip dexterity, it is not readily linked with dexterity metrics given a task-space pose, especially if one considers redundant robots that can reach a pose via configurations that potentially differ on their dexterity. In other words, end-effector poses lying closely in task-space might lie far apart in configuration space resulting in decreased dexterity.

Besides reachability, there are other workspace analysis considerations that use metrics which focus on end-effector

force, speed capabilities, mechanical actuator characteristics, isotropy, or sensitivity to noise and inaccuracies in calibration or motor control.

B. Design Optimisation

Manipulator design optimisation aims to identify parameters that maximise selected metrics for given objectives. According to [22], manipulator design can be separated into:

- Structure design, which describes the number and type of joints and links, and
- Dimensioning, which is the process of finding the optimal dimensions of the structural components, *e.g.* the length of links, and the pose of the joints.

In practice, physical and mechanical constraints place limits on the flexibility of the structural design, which at the extreme has infinite dimensionality. This implies that design optimisation is commonly limited to the dimensioning challenge. The following work focuses in regard to design optimisation on reachability in $\mathbb{R}^6$. Other performance criteria such as force, speed, or susceptibility to mechanical noise, could be incorporated with the presented approach, however is not further elaborated.

C. Related Work

Analysing the workspace dexterity of manipulators is most frequently based on using the Jacobian matrix, while metrics to evaluate dexterity are the condition number and the manipulability measure, see (8) and (9) for their definition. In [21], manipulability is combined with an availability function to take joint limits into consideration. A workspace analysis using an augmented Jacobian is proposed in [20]. Constraints such as joint limits, obstacle collision, self-collision and task objectives are used in [20] to optimise the grasping of an anthropomorphic robot. In [23], column-wise penalisation of the Jacobian was used to account for distance to joint limits, obstacles, and mechanical instabilities of CTR. The resulting dexterity analysis, based on the singular values of that penalised Jacobian, guided an operator to regions of high dexterity during telemanipulation.

The global isotropy index (GII) is a metric that optimizes for global isotropy, [24]. It is computed by searching over the entire manipulator workspace for the minimum quotient of the minimum and maximum singular values based on a scaled end-effector Jacobian. The translational and rotational parts of the Jacobian are individually scaled based on respective performance requirements, *e.g.* force or speed. The noise amplification index (NAI), [25], is a metric that assesses robot design regarding how noise or calibration errors in the joint-space are amplified in the task-space.

Simulation-based design optimisation of continuum manipulators is also an active field of research. In [7], a human-centric approach is described. A virtual reality headset and a haptic device were used to simulate and interactively modify CTR design parameters (curvature, curved length, and actuator angle) for a given patient anatomy. The clinical goal was percutaneous access to the kidney. Alternative methods are purely computer-based and involve a first-level optimisation that alters the parameter set of the robot architecture, and a second-level evaluation, which constitutes the optimisation criteria, and which quantifies the fitness of the robot to the desired task given the provided parameter set. In order to evaluate the fitness, the second-level evaluation often employs optimisation itself.

Design optimisation limited to dimensioning searches the parameter-space (first-level optimisation) and is amenable to approaches such as non-linear optimisation *e.g.* Nelder-Mead used in [6], [16] or *e.g.* gradient-based optimisation [26], or stochastic optimisation *e.g.* adaptive Simulated Annealing (ASA) used in [27]–[29]. Additional optimisation approaches have been employed for serial-link robots, *e.g.* interval analysis [22], grid methods [30], and genetic algorithms [31], [32].

The evaluation of suitability metrics (second-level optimisation) has also seen extensive research. Most published research requires that goals and objectives are explicitly provided to the optimisation algorithm in the way of 3D points or task space poses. Other approaches use coarser goals such as volumetric/reachability objectives, which means that only the $\mathbb{R}^3$ task space is considered. Examples of second-level optimisation approaches include:

- Inverse kinematics to multiple targets considering position, orientation accuracy, and stability [6],
- Volume-based coverage [16],
- Fréchet distance between the continuum robot centre line and a desired path [33],
- Number of collaborative configurations specifically for a dual-arm system [34],
- Reachable workspace, task and orientation dexterity, and insertion port diameter [35],
- Relative frequency of reachable goals based on motion planning results with rapidly exploring random tree (RRT) [27],
- High stability [36], and
- Ability to visualise a target based on motion planning using a modified RRT approach [29].

D. Contributions

The following contributions are presented in the remainder of the paper.

1) A novel algorithm is proposed to characterise the workspace and dexterity capabilities of a manipulator independently of whether it is a serial-link robot or a flexible robot, using uniform voxelisation of the 6 DoF task-space, see Sec. II. The presented approach avoids both local approximations and prescriptive task-space objectives.
2) Based on the presented algorithm, a dexterity measure, which is subsequently used as an objective function for robot design optimisation, is introduced. Instead of using linearisation, the approach uses reachability of the 6 DoF voxels augmented with additional criteria which *i.e.* consider the neighbourhood, see Sec. III.
3) It is demonstrated that the presented principle can be used to improve the performance of inverse kinematics

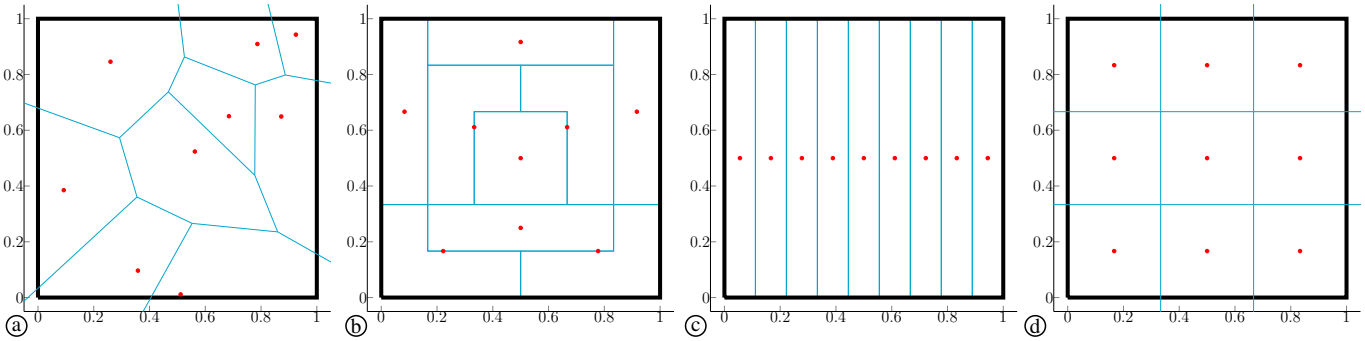

Fig. 2. Different tessellation approaches for a square, without loss of generality, they are also applicable to SO(3): (a) Using random uniform distributed samples, resulting area sizes and area shapes are different, (b) tessellation using different shapes of equal area size, (c) tessellation results in areas of same size and shape but are not compact, (d) tessellation is compact and areas are of same size and shape. Tessellation similar to (d) is preferred.

optimisers that suffer from local minima, particularly in the challenging domain of CTR inverse kinematics.The approach of storing joint-configuration in the 6 DoF lookup table and fostering uniformly dense sampling is presented in Sec. IV.

4) Optimisation of a robot architecture is introduced that considers the overall workspace reachability and robot dexterity without relying on predefined tasks, see Sec. V.

II. Workspace Voxelisation

A uniform subdivision of the workspace is essential for computing the capabilities of a manipulator and characterising its dexterity within the workspace. It is particularly important that the subdivision volumes are equal in surface area/volume and shape to be suitable for comparing different parts of the manipulator's workspace.

This section introduces an approach to divide the 6-dimensional SE(3) workspace into hyper-volumes of equal dimensions. The translational space, $\mathbb{R}^3$, and the rotational space, SO(3), are independently partitioned and can therefore be independently queried with regards to reachability ($\mathbb{R}^3$) and dexterity (SE(3)).

The translation component of SE(3) is subdivided with a common linear axis-aligned division of a cuboid workspace, which leads to a uniform discretisation in all 3-dimensions. With regards to the rotational space, SO(3), the subdivision process is based on recursive division of an initial uniform partitioning. Combining the two discretisation results ultimately creates a voxelised workspace that is suitable for assessing the reachability and dexterity of a manipulator in its different regions.

A. Voxelisation of Translation Space

For comparing different workspace regions, each voxel in $\mathbb{R}^3$ must be of the same dimensions, which is routinely achieved by subdividing 3D space into cubes of identical size. Assuming the minimum workspace to evaluate has an edge length of $\mathcal{E}^0_{\{x,y,z\}}$, origin $\boldsymbol{p}_0 \in \mathbb{R}^3$, and that the edge length of the voxels should be e_v, then the number of discretisations along each axis is:

$$N_{\{x,y,z\}} = \left\lceil \frac{\mathcal{E}^0_{\{x,y,z\}}}{e_v} \right\rceil, \tag{1}$$

which, due to quantisation, adjusts the final size of the workspace under consideration to have an edge length of

$$\mathcal{E}_{\{x,y,z\}} = N_{\{x,y,z\}}\, e_v. \tag{2}$$

Indices $\boldsymbol{I}_T \in \mathbb{N}^3$ associate a 3D point $\boldsymbol{p} \in \mathbb{R}^3$ to a voxel in the partitioned workspace via

$$\boldsymbol{I}_T = \left\lfloor (\boldsymbol{p} - \boldsymbol{p}_0) \oslash \mathcal{E}_{\{x,y,z\}} \right\rfloor, \tag{3}$$

where $\oslash$ is the *Hadamard* division operator, which performs element-wise division of two vectors or matrices of equal dimensionality.

This index associates the translation component ($\boldsymbol{p}$) of an end-effector pose ($\boldsymbol{\mathcal{T}}$) to a voxel. The end-effector orientation ($\boldsymbol{\mathcal{R}}$), however, needs to be separately considered.

B. Voxelisation of Rotation Space

In contrast to the translation space, the rotation space SO(3) is bounded and is represented by the 3D hypersurface of the 4D unit sphere. Points on that hypersphere are represented by the four coordinates of a quaternion. To identify indices that associate an end-effector orientation $\boldsymbol{\mathcal{R}}$ to a quantised rotation volume, SO(3) needs to be uniformly partitioned.

The problem of SO(3) discretisation is recurring and different approaches to address it are possible. One possible approach is to use a uniform random distribution in SO(3), see [37]–[39] in conjunction with *Voronoi* tessellation. This approach allows creation of any number of rotation space voxels. However, the resulting voxels are of different shapes and volumes, which makes it difficult to compare them, see Fig. 2 (a).

Other published endeavours, [40], [41], propose tessellations that deliver elements of equal surface area. However, they do not produce discretisation areas of identical shape or dimensionality, see Fig. 2 (b). Therefore, they are not suitable for workspace and dexterity evaluation in robotics. Also, tessellations as depicted in Fig. 2 (c), which have identical shape and size, are not suitable as they are not compact. The issue is that they may represent areas which can be far apart. To address this, a novel approach for SO(3) discretisation, building on [42] but deviating in terms of indexing of the rotation space, is described in the next sections. The goal is to obtain a tessellation that is compact, and volumes are of

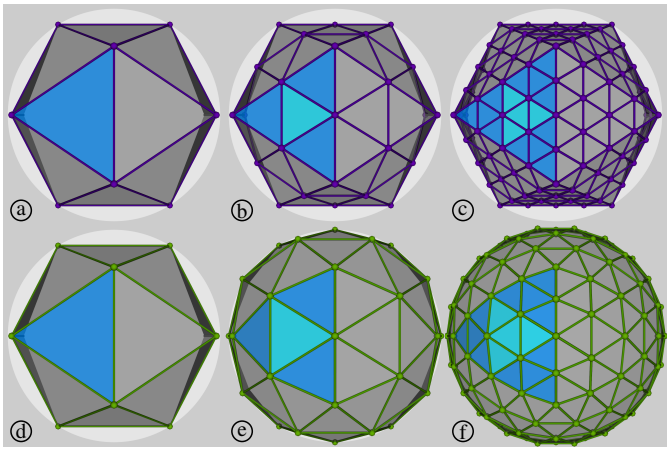

Fig. 3. Recursive subdivision of an icosahedron surface to partition the unit sphere surface. Vertices represent the inclination and azimuth values in SO(2). *Top row:* without projection of new vertices onto the surface of the sphere. *Bottom row:* after projecting the vertices onto the surface of the sphere following each subdivision step.

identical size and shape, as depicted in Fig. 2 (d). To ease the reader into the methodology, the subdivision of SO(2) is explained first.

1) Subdivision of SO(2): A unit sphere in $\mathbb{R}^3$ is considered for subdivision of SO(2). Since every point on the surface of that unit sphere can be described using two variables, the inclination and the azimuth, the surface of the sphere represents SO(2). The process to subdivide the surface of a sphere in equally shaped and sized areas is depicted in Fig. 3.

The process starts with an icosahedron, whose vertices all lie on the unit sphere, see Fig. 3(a). While it is possible to use another of the five platonic solids in $\mathbb{R}^3$, the icosahedron has the highest number of faces, it uses the simplest face polygon (the equilateral triangle), and it is commonly used in computer graphics and astrophysics for sphere discretisation.

First, each of the icosahedron's triangular faces is subdivided. By connecting the triangle's edge centres, new triangles are formed, see Fig. 3(a). This subdivision results into splitting the original triangle into four, see change from Fig. 3(a) to (b). The newly generated vertices, which are the edge centres of the original triangle, however, lie on a secant through the unit sphere. To address this, the newly generated triangles' vertices are projected back onto the sphere's surface via normalisation of their coordinates, see change from Fig. 3(b) to (e). This projection introduces an error as the edges between *new-new* and *new-original* vertices experience different elongation from the projection. The resulting error was analysed in Appendix A comparing a uniform-subdivision with a fixed number of iterative subdivisions. Recursive subdivision and projection produce increasingly finer regular meshes of the sphere's surface, see Fig. 3(f). Note that the alternative of projecting the vertices onto the sphere surface only at the last step is not appropriate; preliminary evaluation demonstrated that it leads to higher error.

To achieve a discretisation of the sphere's surface, an association of surface points to discretised rotations is required. The discretised rotation which represents a set of continuous rotations will be called a discretisation centre. An adequate association is one in which every surface point is associated to its closest discretisation centre. Therefore, the chosen discretisation centres can be *Voronoi* cell centres of equal surface areas. Furthermore, a perfect association must make sure that any rotation $\boldsymbol{r}_{\mathrm{I},i}$ that is associated to a discretisation centre $\boldsymbol{r}_{\mathrm{I}}$ is closer to $\boldsymbol{r}_{\mathrm{I}}$ than the distance of any rotation $\boldsymbol{r}_{\mathrm{II},j}$ to $\boldsymbol{r}_{\mathrm{I}}$, where $\boldsymbol{r}_{\mathrm{II},j}$ is *not* associated to $\boldsymbol{r}_{\mathrm{I}}$. Such an optimal association would only be fulfilled by a circular *Voronoi* cell, which, however, cannot be used to tessellate a surface.

This challenge can be overcome in two ways, illustrated in Fig. 4. The first approach, shown in Fig. 4(a), entails using the vertices (triangle corners) created during the subdivision process as *Voronoi* cell centres (discretisation centres), and leads to hexagonal cells. The second approach, shown in Fig. 4(b), uses the generated triangles' face centres as discretisation centres, which leads to triangular cells. The coloured areas in Fig. 4 represent all the rotations that are associated with the *Voronoi* centre of the same colour. With either approach, any arbitrary continuous rotation can be associated to a discrete unique vertex, *i.e.* the centre of the *Voronoi* cell within which this continuous rotation lies in. Therefore, the *Voronoi* cells can be termed SO(2) rotation voxels.

The most appropriate selection of the *Voronoi* cell shape is

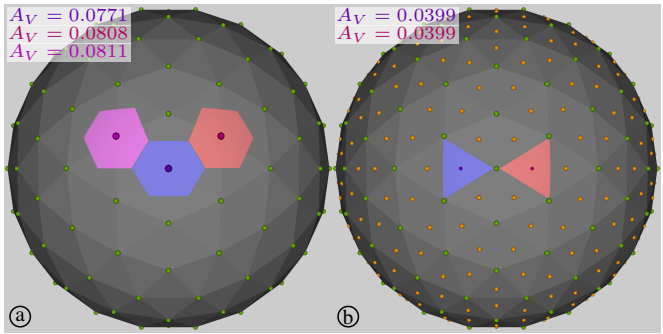

Fig. 4. Association of the surface of SO(2) to discretised rotations. Different *Voronoi* cell centres are used for the same level of subdivision: (a) Triangle vertices (green) are used as *Voronoi* cell centres *i.e.* discretised rotations. (b) Face centres (orange) are used as discretised rotations. There are (a) three and (b) two discretised rotations displayed with their associated areas in the same colour. The sphere depicted is identical between (a) and (b), and was subdivided twice resulting in 162 vertices and 320 faces.

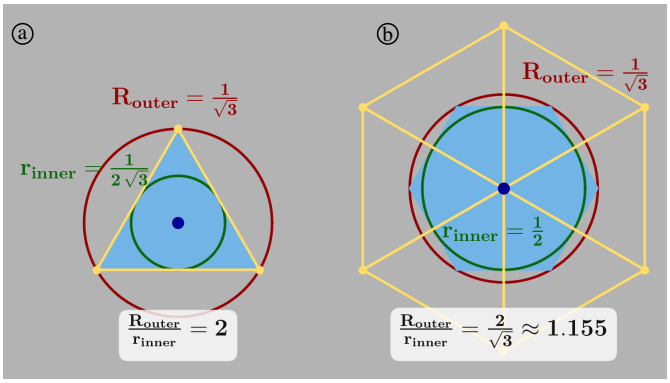

Fig. 5. Alternative *Voronoi* cells for SO(2) subdivision. (a) Triangle face centres are used as centres of the triangular *Voronoi* cells. (b) Vertices (triangle corners) are used as hexagonal *Voronoi* cell centres.

further elaborated in the following. Fig. 5 depicts the geometry of those *Voronoi* cells considering a locally flat surface, and triangles which have a edge-length of 1. The figure depicts two radii:

1) the outer radius, R_{outer}, that represents the distance of a rotation that is furthest from the *Voronoi* cell centre to which it is associated;
2) the inner radius, r_{inner}, that represents the distance of the closest rotation which is *not* associated with the *Voronoi* centre.

In an ideal case, circular *Voronoi* cells would be chosen with a quotient of R_{outer} to r_{inner} equal to 1. When $\frac{R_{\text{outer}}}{r_{\text{inner}}} >> 1$, a discretised rotation captures continuous rotations that are distant from each other. A triangle-shaped *Voronoi* cell, see Fig. 5(a), results in a quotient of 2. On the contrary, using the triangle corners as discretisation centres, see Fig. 5(b), results in a quotient of ≈ 1.155 and an approximately circular shape for the rotation voxel, making this approach the preferred way to discretise the SO(2) rotation space.

To conclude the study on discretisation of SO(2), the subtended angle of neighbouring discretisation centres will be calculated. This subtended angle between two vectors linking the unit sphere's centre and two neighbouring vertices on the original icosahedron (ϕ_{vcv}, vertex-centre-vertex angle) is approximately $63.435°$, see [43]. Each subdivision halves the angle, implying that the discretisation resolution is $63.435° \times 2^{-n_{\text{div}}}$, where n_{div} is the number of applied subdivisions. The resulting angle (ϕ_{vcv}) represents the resolution of the discretised rotation space.

The maximum angle (vertex-centre-vertex) between the rotation voxel centre and the furthest rotation associated with that voxel is computed by multiplying the angle between two voxel centres ϕ_{vcv} with R_{outer} which is based on a triangle edge length of 1, see Fig. 5.

To validate the proposed subdivision process and hexagonal *Voronoi* cells can indeed robustly represent discretised SO(2), random samples from the continuous SO(2) space were uniformly drawn. Then, combinations of different subdivision levels, *Voronoi* cell shapes, and discretisation centres generated the SO(2) partitions to be evaluated. The evaluation criterion was that samples uniformly drawn from SO(2) should uniformly map to the created rotation voxels. It was found that 2 subdivisions and tessellation with hexagonal *Voronoi* cells whose centre is also the centre of the rotation voxel leads to angular resolution of $16°$, making this approach appropriate for the discretisation of SO(2); details are provided in Appendix A.

2) Subdivision of SO(3): The process for subdividing SO(3) is similar and makes use of the 4D unit sphere (a unit hypersphere). It starts with a convex regular 4-polytope, the 600-cell, which is analogous to the icosahedron. The 600-cell constitutes 600 tetrahedra, each connecting 4, out of 120 unique, 4-dimensional vertices that represent unit quaternions.

Building on [44], the 120 quaternions are generated based on the golden ratio $\phi = \frac{1}{2}(1+\sqrt{5})$ as follows:

- 96 from even permutations of $(\pm\phi, \pm 1, \pm\phi^{-1}, 0)$;
- 8 from permutations of $(\pm 2, 0, 0, 0)$;
- 16 from permutations of $(\pm 1, \pm 1, \pm 1, \pm 1)$.

The resulting vectors are scaled to unit length to provide the set of unique unit quaternions.

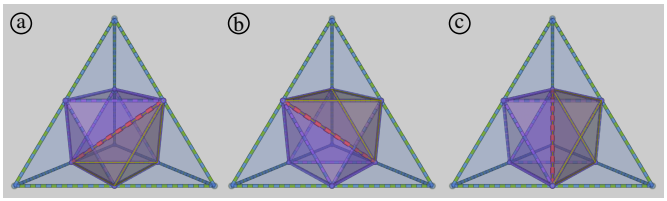

Fig. 6. Subdivision of a tetrahedron into 8 smaller tetrahedra. (a), (b), (c) depict the different options of how the inner octahedron can be subdivided. Green dotted: initial tetrahedron. Blue dotted: 4 new similar tetrahedra having their origin at the corners of the original tetrahedron. Purple dotted: 4 new tetrahedra (from the central purple octahedron) filling the centre. Yellow: highlight of one of the inner tetrahedron. Red dotted: subdivision pattern.

By construction, the smallest angular distance (ϕ_{vcv}, as before), of any two vertices in the 600-cell is $36°$. To group the 600-cell into tetrahedra that correspond to discretisation voxels to be subdivided, first the 4-permutations of 120 quaternions are identified, therefore creating $\binom{120}{4}$ tetrahedra. Then, tetrahedra discretising the 600-cell are identified as those 4-quaternion sets for which the pairwise angles of all their members are equal to $36°$. Since quaternions $\mathcal{Q}$ and $-\mathcal{Q}$ represent the same rotation, only one hemisphere of the unit hyper-sphere needs to be considered. To identify the tetrahedra corresponding to a single hemisphere, first a reference quaternion $\mathcal{Q}_0$ is selected, *e.g.* $\mathcal{Q}_0 = \mathcal{Q}_{\text{N}} = (0, 0, 0, 1)$. Then, every tetrahedron whose vertices correspond to 4-quaternions with an angle of less than $108° = 90° + \frac{36°}{2}$ are considered as part of the hemisphere. The selected angular threshold considers all tetrahedra to the north of the hyper-sphere equator ($90°$) in addition to those lying on the equator $\left(\frac{36°}{2} = 18°\right)$; please note that the term *equator* is defined with respect to the selected reference quaternion. A total of 330 tetrahedra is considered.

These tetrahedra can be subdivided similarly to the triangles in SO(2) (recall Fig. 3), according to [45]. A 3-dimensional visualisation is depicted in Fig. 6. A tetrahedron can be subdivided by introducing 6 vertices, one at the centre of each of its edges. As in the SO(2) case, these vertices must be normalised, *i.e.* projected onto the hypersurface of the 4D hypersphere. Then, four identical sub-tetrahedra are obtained each having an *origin* at a corner of the initial tetrahedron (Fig. 6, blue); an octahedron is obtained in the interior (Fig. 6, purple). Subsequently, the octahedron is split into four sub-tetrahedra by introducing an edge passing through its centre. The three possible splits are illustrated in Fig. 6. Since the goal of the subdivision is to reduce the angle between quaternions, the edge $\boldsymbol{e}_k$ for which the angle between the quaternions corresponding to its end vertices ($\mathcal{Q}_{k,0}$, $\mathcal{Q}_{k,1}$) is minimum is chosen:

$$\begin{aligned} k^\star &= \underset{k\in\{a,b,c\}}{\arg\min} \left\{2 \arccos\left(\|\langle \mathcal{Q}_{k,0}, \mathcal{Q}_{k,1}\rangle\|\right)\right\} \\ &= \underset{k\in\{a,b,c\}}{\arg\max} \left\{\langle \mathcal{Q}_{k,0}, \mathcal{Q}_{k,1}\rangle\right\}, \end{aligned} \tag{4}$$

where a, b, c refer to the three cases depicted in Fig. 6, and the $\langle \cdot, \cdot \rangle$-operator denotes the dot product. Rearranging (4) exploits that all Quaternions are from the Northern hemisphere, and hence computing of the absolute value of the dot product is unnecessary. Additionally, $\arccos$ is monotonically decreasing such that the trigonometric evaluation can be omitted.

Once the subdivision is sufficiently fine, the list of quaternions which represent the vertices of all tetrahedra is consolidated by removing duplicate entries. Duplicates may occur since the subdivision process for every tetrahedron is independently performed. In the remainder of this article, the set of unique quaternions ($\boldsymbol{\mathcal{Q}}^v = \{\mathcal{Q}_0^v, \mathcal{Q}_1^v, \ldots\}$) will also be referred to as rotation voxels, since they will constitute the centre of a rotation volume.

3) Association of a Rotation to a Rotation Voxel: To associate an arbitrary quaternion $\mathcal{Q}_R$ to a rotation voxel, one could iterate over the entire set of $\boldsymbol{\mathcal{Q}}^v$ and identify the closest match in terms of angular difference using:

$$I_R = \underset{i \in \{1 \ldots |\boldsymbol{\mathcal{Q}}^v|\}}{\arg\min} \left\{ 2 \arccos \left(\| \langle \mathcal{Q}_R, \mathcal{Q}_i^v \rangle \| \right) \right\}. \tag{5}$$

This search, however, is of complexity linear to the cardinality of $\boldsymbol{\mathcal{Q}}^v$ ($\mathcal{O}(n)$). Using a 4D-tree prepopulated by $\boldsymbol{\mathcal{Q}}^v$ would reduce computational complexity to $\mathcal{O}(\log(n))$, which would still be prohibitive when billions of lookups are required. Therefore, an alternative approach in constant-time ($\mathcal{O}(1)$) was developed to tackle the association task.

A 3-dimensional cube ($\mathcal{C}^{\text{rot}}$) with edges spanning $[-\pi, \pi]$ is generated and discretised with $N_{\text{cube}}^{\text{rot}}$ equally spaced subdivisions per dimension. The cube in $\mathbb{R}^3$, which will be used as a lookup table, describes all possible rotations, and each of the generated voxels in $\mathbb{R}^3$ describe a subset of rotations in SO(3). All rotations within one voxel are represented by the centre of the voxel which is a rotation vector in SO(3).

The closest quaternion in $\boldsymbol{\mathcal{Q}}^v$ is computed for each of the rotation vectors using (5), and the quaternion's index within $\boldsymbol{\mathcal{Q}}^v$ is stored in the cube's voxel. This 3D lookup table needs to be generated only once per given number of $N_{\text{cube}}^{\text{rot}}$ discretisations and hyper-sphere subdivision levels; recall that quaternions $\boldsymbol{\mathcal{Q}}^v$ are generated per subdivision level. An example cube is depicted in Fig. 7. For visualisation purposes, the cube is cut along the diagonal plane and the colours represent the different indices values for I_R.

The volume of the rotation space that is represented by a single rotation vector varies for different $N_{\text{cube}}^{\text{rot}}$. Larger $N_{\text{cube}}^{\text{rot}}$ imply smaller individual voxel volumes and hence more accurate representations. It is generally advisable to choose $N_{\text{cube}}^{\text{rot}} \gg |\boldsymbol{\mathcal{Q}}^v|$.

Once the lookup table is generated, the index I_R for a given $\mathcal{Q}_R$ can be computed by converting $\mathcal{Q}_R$ to the corresponding rotation vector $\boldsymbol{v}_R^{\text{rot}}$. Then the index:

$$I_R = \mathcal{C}^{\text{rot}} \left(\left\lfloor \frac{\boldsymbol{v}_R^{\text{rot}} + [\pi, \pi, \pi]^T}{2\,\pi / N_{\text{cube}}^{\text{rot}}} \right\rfloor \right), \tag{6}$$

where $\mathcal{C}^{\text{rot}}$ is the rotation cube lookup table.

The angle between neighbouring vertices on the unit sphere in SO(3) is 36° for the initial 600-cell, and every subdivision results into this angle being halved. Then, the angular distance between centres of two adjacent rotations voxels in SO(3) is $\approx 36^\circ \cdot 2^{-n_{\text{div}}}$.

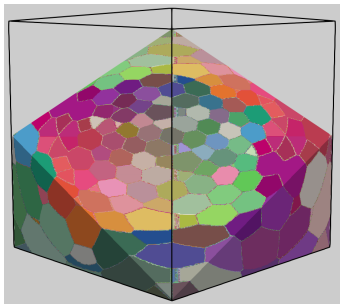

Fig. 7. Lookup table for SO(3) rotation indices; colours represent an associated index.

To validate that the discretisation approach leads to uniform representation of SO(3), frequency-based error analysis similar to the SO(2) case is available in Appendix B.

With the described approach the 6-dimensional workspace, *i.e.* 3D translation and SO(3), can be subdivided into equally shaped and sized voxels. The presented voxelisation enables the development of approaches which analyse robot workspace and provides excellent computational performance.

III. Workspace Analysis

As mentioned in Sec. I-A, common approaches to analyse a robot's workspace are primarily based on linearisation of the mapping between joint-space and task-space. Approaches based on local approximations, however, can lead to erroneous conclusions on the robot's task-space dexterity. To avoid these errors, a global approach which is not based on linearisation is introduced. The approach is global in the sense that it considers the global joint-space, *i.e.* all possible joint-space configurations, to make an assessment of the task-space. The metric can be used to assess the entire manipulator workspace, task-based relevant volumes of the workspace, or a single task-space voxel. Independent of what part of the workspace is evaluated, the global set of joint-configurations is considered.

A. Drawbacks of Linearisation for Workspace Analysis

Recall the end-effector Jacobian matrix:

$$\boldsymbol{\mathcal{J}} = \frac{\partial \dot{\boldsymbol{x}}}{\partial \dot{\boldsymbol{q}}}, \tag{7}$$

where $\boldsymbol{x}$ denotes the end-effector's pose, $\boldsymbol{q}$ denotes the joint values, and the $\dot{}$ operator denotes derivation with respect to time. The Jacobian matrix's singular values $\boldsymbol{\sigma}_J$ can be used to compute the manipulability ($\mathcal{M}$) or inverse condition number ($\bar{\mathcal{C}}$) of a manipulator:

$$\mathcal{M} = \sqrt{\| \boldsymbol{\mathcal{J}} \boldsymbol{\mathcal{J}}^T \|} = \prod_i \sigma_i, \tag{8}$$

$$\bar{\mathcal{C}} = \frac{\sigma_J^{\min}}{\sigma_J^{\max}}, \tag{9}$$

where $\sigma^{\min}$, and $\sigma^{\max}$, denote the minimum, and maximum, singular values, respectively. These two measures are often

used to characterise the dexterity ($\mathcal{D}$) of a manipulator in the neighbourhood of a joint configuration under examination.

While $\mathcal{M}$ or $\bar{\mathcal{C}}$ are useful for comparing different parts of the same manipulator workspace, their numeric values are not geometrically meaningful. As a result, they cannot be directly compared among manipulators, since they heavily depend on kinematic parameters such as link-lengths.

These metrics are generally used to characterise a 3D voxel in translational space disregarding that the Jacobian can significantly change depending on the end-effector orientation. In this case, each 3D voxel $\boldsymbol{v}_3$ is associated with the maximum robot dexterity, $\mathcal{D}^L_{\mathcal{V}_3}$, regardless of end-effector orientation:

$$\mathcal{D}^L_{\mathcal{V}_3}(\boldsymbol{v}_3) = \max_{\boldsymbol{q}_k} \left\{ \mathcal{D}_q(\boldsymbol{q}_k) \mid \boldsymbol{x}(\boldsymbol{q}_k) \in \boldsymbol{v}_3 \right\}, \tag{10}$$

with $\mathcal{D}_q$ being the dexterity of a specific joint configuration and $\mathcal{D}^L_{\mathcal{V}_3}$ the dexterity at a specified task-space voxel. With this approach, each voxel $\boldsymbol{v}_3$ that belongs to the discretised $\mathbb{R}^3$ is linked to the manipulator orientation that reaches only that specific 6 DoF pose, with reduced capabilities of extrapolation to neighbouring voxels. Finally, when used to assess whether a manipulator is dexterous in a specific part of the task-space, $\mathcal{D}^L_{\mathcal{V}_3}$ may provide misleading information, especially when conclusions are drawn for task-space poses which are not in the neighbourhood of the currently examined joint configuration; recall that neighbouring task-space poses do not imply a neighbourhood in joint space. Metric $\mathcal{D}^L_{\mathcal{V}_3}$ is a local one, and a global accurate measure of reachability and dexterity of the 6 DoF task-space is needed.

As previously mentioned, there are metrics such as the GII and the NAI, which scale the Jacobian to overcome the shortcomings of the conditional number $\bar{\mathcal{C}}$ and manipulability $\mathcal{M}$. Nevertheless, as those metrics are based on the end-effector Jacobian, their conclusiveness for the task-space is hindered due to the non-linearity between joint- and task-space as well as the multivalued mapping from task- to joint-space. Those metrics and Jacobian scaling can, however, be combined with the approach presented in the following.

B. Workspace Analysis without Linearisation

As detailed in Sec. II, task-space SE(3) can now be voxelised in six dimensions with (hyper) voxels $\boldsymbol{v}_6$ (belonging to discretised $\mathbb{R}^6$) that have the same shape and size. These voxels can be classified as reachable or non-reachable when sampling the joint-space of a manipulator. The 6 DoF dexterity ($\mathcal{D}_{\mathcal{V}_6}$) of a voxel ($\boldsymbol{v}_6$) can then be defined as *one* if it is reached in position and orientation during sampling and as *zero* if not reached:

$$\mathcal{D}_{\mathcal{V}_6}(\boldsymbol{v}_6) = \bigvee_{\boldsymbol{q}_k} \begin{cases} 1, \text{ if } \boldsymbol{x}(\boldsymbol{q}_k) \in \boldsymbol{v}_6 \\ 0, \text{ else} \end{cases}, \tag{11}$$

where $\boldsymbol{q}_k$ represents a random joint configuration during sampling and $\boldsymbol{x}(\boldsymbol{q}_k)$ the corresponding task-space pose.

To assess the extent in which a manipulator can attain different orientations at a specific 3D voxel $\boldsymbol{v}_3$ (of discretised $\mathbb{R}^3$), the number of attainable discretised rotations relative to the overall number of discretised rotations can be computed:

$$\mathcal{D}^G_{\mathcal{V}_3}(\boldsymbol{v}_3) = \frac{\sum_{\forall \boldsymbol{v}_6 : \boldsymbol{v}_{6|3} \equiv \boldsymbol{v}_3} \mathcal{D}_{\mathcal{V}_6}(\boldsymbol{v}_6)}{|\boldsymbol{\mathcal{Q}}^{\boldsymbol{v}_{6|3}}(\boldsymbol{v}_3)|}, \tag{12}$$

where $\boldsymbol{v}_{6|3}$ corresponds to the translational part of the 6 DoF voxel $\boldsymbol{v}_6$, $\mathcal{D}^G_{\mathcal{V}_3}$ is the dexterity of voxel $\boldsymbol{v}_3$, and $|\boldsymbol{\mathcal{Q}}^{\boldsymbol{v}_{6|3}}|$ is the number of $\boldsymbol{v}_6$ voxels within a $\boldsymbol{v}_3$ voxel or with other words the number of discretised orientations for a given position. The dexterity for the $\boldsymbol{v}_3$ voxel is 1 if all orientations are reachable at that voxel location and 0 if no orientation is reachable. Since this method evaluates the possible positions and orientations considering the entire definition space of feasible joint values, in this paper it is considered a global measure ($\mathcal{D}^G_{\mathcal{V}_3}$).

Equation (12) provides a measure (ratio) of the rotational space that can be reached at a given position. The opposite could be analogously evaluated, so that the measure would indicate how many positions can be reached given an orientation.

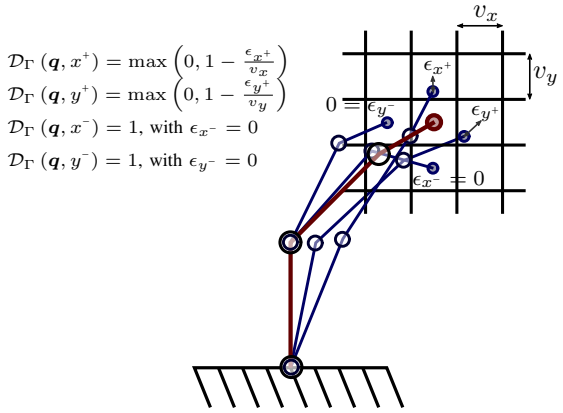

Fig. 8. Dexterity measure based on how well a manipulator in its initial configuration $\boldsymbol{q}$ (red) can reach neighbouring voxels. The deviation from the targeted neighbouring pose is $\epsilon_{x^+}, \epsilon_{x^-}, \epsilon_{y^+}, \epsilon_{y^-}$. No deviations from the targeted pose lead to a partial dexterity ($\mathcal{D}_\Gamma$) of *one*. Small deviations lead to a value $\in [0, 1]$. Deviations of more than v_x or v_y lead to a partial dexterity of *zero*. All partial dexterities are combined to the configuration dexterity ($\mathcal{D}(\boldsymbol{q})$) using (13).

C. Weighting the Reachability Map

When dexterity must be assessed in the context of a specific task, the reachability of certain rotations and certain positions may be weighted. For example, it might be important that one manipulator can retroflex in one application, while in another the same manipulator might require its end-effector to only face forward.

Using the presented voxelisation approach, a 6D weight matrix representing the importance of specific rotations at specific positions can be used by multiplying each voxel value. In other words a specific voxel index ($\boldsymbol{I}_{T,R}$) can be associated with a voxel-specific scaling factor ($\boldsymbol{S}_{T,R}$). Important voxels can be multiplied with a factor larger than one while insignificant voxels are multiplied with a factor smaller than one or, for irrelevant voxels, zero. This approach is sketched in Fig. 9, where a robot is tasked to orthogonally sweep the boundary indicated in grey. Within the example, orientation-scaling

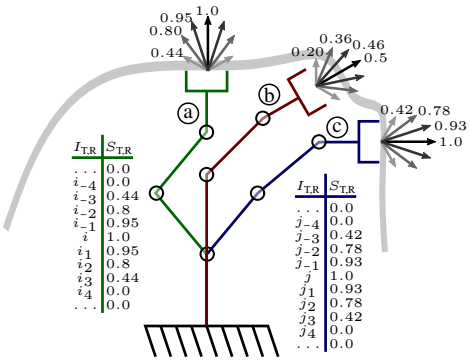

Fig. 9. Regional workspace weighting via association of scaling factors to voxels to regulate emphasis on certain rotations and translations. The grey line represents the target workspace. The three differently coloured configurations show different weights in their proximity. (a) *Green configuration*: Orientations orthogonal to the tangent of the line are highly weighted. (b) *Red configuration*: Positions at a distance from the line are weighted lower than positions close to the line. (c) *Blue configuration*: Weighting at the end of the line of interest are lower than in the middle of the line. *Tables*: the tables represent weights for different rotation indices at the specific end-effector position. The orientation normal to the trajectory (index: i) is weighted the highest while deviations from the normal are considered of lower importance and hence are weighted lower.

differs for each robot configuration. Further, as the middle robot configuration, indicated as (b), is not contacting the boundary, the scaling factors $\boldsymbol{S}_{T,R}$ for the different directions are lower than for poses (a), (c), which are in contact with the boundary. Weighting enables the comparison of different manipulators for specific tasks, rather than broadly across their workspaces.

D. Dexterity Map Considering the Neighbourhood

Reachability and the ability to move to neighbouring poses are important considerations in robot design. Naturally, other considerations such as force, speed requirements, noise amplification may be important in different applications. The reachability map in $\mathbb{R}^6$ weights a reached voxel as 1 if reachable and 0 if non-reachable. To incorporate other performance considerations, every joint configuration associated with a specific voxel can be further evaluated against performance criteria presented in [24], [25] to compute a weight $\in [0, 1]$.

The reachability map provides information on which *Cartesian* poses a manipulator can reach. It does not assess, however, whether the manipulator can move from that specific pose to neighbouring poses. Assessing motion in different directions allows for more comprehensive insights. Combining the reachability map presented in Sec. III-B with techniques presented in Sec. III-A has the potential to provide further insights into the manipulators capabilities. The numerical values provided by both $\mathcal{M}, \bar{\mathcal{C}}$, however, cannot be easily interpreted. Therefore, a new approach is presented that uses inverse kinematics to conclude to what extent the neighbourhood $\Gamma(\boldsymbol{q})$ in the *Cartesian* space can be reached from a given configuration ($\boldsymbol{q}$). Simulating motions into the neighbourhood allows taking motion constraints into account including joint-limits, self-collisions, or collisions with the environment. The neighbourhood can be coarse such that only points along the positive and negative directions of the orthonormal basis ($\pm \boldsymbol{e}_i$) are considered as neighbours and evaluated ($n_\Gamma = 12$ evaluations), or fine by evaluating permutations of diagonal neighbours, which results in $n_\Gamma = 64$ evaluations, with $\Gamma \in \{-1, +1\}^6$. Furthermore, the size of the sphere of influence is an important design choice, meaning the distance within which a neighbourhood is still considered relevant.

To calculate the dexterity for a specific joint configuration, the reachability of neighbouring poses is considered:

$$\begin{aligned} \mathcal{D}_q(\boldsymbol{q}) = \quad & (1 - w_n)\, \mathcal{D}^G_{\mathcal{V}_6}(\boldsymbol{x}(\boldsymbol{q})) + \frac{w_n}{n_\Gamma} \sum_{i=1}^{n_\Gamma} \mathcal{D}^\Gamma_q(\boldsymbol{q}, i), \\ \text{with} \quad & \mathcal{D}^G_{\mathcal{V}_6}, \mathcal{D}^\Gamma_q, w_n \in [0, 1], \end{aligned} \tag{13}$$

where $\mathcal{D}_q(\boldsymbol{q})$ is the dexterity for a specific joint value ($\boldsymbol{q}$), $\mathcal{D}^G_{\mathcal{V}_6}(\boldsymbol{x}(\boldsymbol{q}))$ the dexterity for reaching voxel $\boldsymbol{x}(\boldsymbol{q})$, $\mathcal{D}^\Gamma_q(\boldsymbol{q}, i)$ the dexterity as a measure of ability to reach a pose from $\boldsymbol{x}(\boldsymbol{q})$ to the i^{th} neighbour of Γ, and $w_n \in [0, 1]$ a weight of how much the neighbourhood should influence the dexterity of a specific joint configuration.

To evaluate whether the manipulator has the dexterity to reach a neighbouring pose from the joint configuration under consideration, the motion to that pose is simulated using a single inverse kinematic step. Metric $\mathcal{D}^\Gamma_q(\boldsymbol{q}, i)$ is calculated based on the distance between the target pose and the pose achieved using a single inverse kinematic step employing the Jacobian method. Alternative approaches including path planning could be considered, but Jacobian-based inverse kinematics represent the direct/linear way to move between configurations:

$$\Delta \boldsymbol{x}^0_i = v_i \boldsymbol{e}_i, \tag{14}$$

$$\Delta \boldsymbol{q}_i(\boldsymbol{q}) = \boldsymbol{\mathcal{J}}^\dagger(\boldsymbol{q}) \Delta \boldsymbol{x}^0_i, \tag{15}$$

$$\Delta \boldsymbol{x}_i(\boldsymbol{q}) = \Delta_x(\boldsymbol{\mathcal{T}}(\boldsymbol{q}), \boldsymbol{\mathcal{T}}(\boldsymbol{q} + \Delta \boldsymbol{q}_i)), \tag{16}$$

$$\mathcal{D}_\Gamma(\boldsymbol{q}, i) = \max \left\{ 0, 1 - \frac{\| \Delta \boldsymbol{x}_i - \Delta \boldsymbol{x}^0_i \|}{v_i} \right\}, \tag{17}$$

where v_i is the step length of the i^{th} direction of Γ, $\Delta \boldsymbol{x}^0_i$ is the desired *Cartesian* step, $\boldsymbol{\mathcal{J}}^\dagger$ is the pseudo-inverse used to compute the joint delta $\Delta \boldsymbol{q}_i$, $\boldsymbol{\mathcal{T}}(\cdot)$ is the forward kinematics function to compute the homogeneous transformation matrix of the end-effector, and $\Delta_x(\cdot) \in \mathbb{R}^6$ provides the *Cartesian* step to reach $\boldsymbol{\mathcal{T}}(\boldsymbol{q} + \Delta \boldsymbol{q}_i)$ from $\boldsymbol{\mathcal{T}}(\boldsymbol{q})$. The Jacobian $\boldsymbol{\mathcal{J}}$ can be adjusted column-wise to address motion restrictions in joint-space such as joint limits, or task-space such as collisions. If a motion is restricted, the columns of the Jacobian that represent the restricted joints can be scaled down to characterise those restrictions. Further details can be found in [23] and [46, Ch. 5].

This set of equations provides the measure $\mathcal{D}_\Gamma(\boldsymbol{q}, i) \in [0, 1]$, which quantifies how well a manipulator can reach *Cartesian* poses in its neighbourhood from the configuration under examination, see Fig. 8. Parameter v_i can be chosen based on how large the neighbourhood which is taken into account should be. When this dexterity measure ($\mathcal{D}_\Gamma$) is used together with the voxelisation approach in Sec. II, the lattice distance in each dimension can be used as a parameter for v_i. Different to the Jacobian approach presented in Sec. III-A this measure allows to easily compare different manipulators and their performance

in the *Cartesian* workspace. This method is fundamentally based on the same principals as $\mathcal{D}^{G}_{\mathcal{V}_3}$ however, it is not solely considering whether a $\mathcal{V}_3$ voxel can be reached with different orientations but further evaluates to what extent the neighbourhood can be reached taking joint-limits, collision constraints, or other custom workspace constraints into account. Joint limits can be taken into account by scaling down $\boldsymbol{\Delta}q$ such that $\boldsymbol{q} + \boldsymbol{\Delta}q$ meets the joint limit criteria, which means that the desired pose cannot be reached fully leading to a lower dexterity value. Similarly, collision constraints can be taken into account by rejecting or modifying resulting configurations from the inverse kinematics step if they result into a collision. Since this approach is based on the global approach and takes the local neighbourhood and constraints into account it is considered a locally refined global method ($\mathcal{D}^{G,L}_{\mathcal{V}_3}$).

IV. Workspace Voxelisation for Inverse Kinematic Lookup

In this section the SE(3) voxelisation approach is used to create a lookup table for querying joint configurations ($\boldsymbol{q}$) that can reach a given task-space (*Cartesian*) pose.

A. Limitations of Common Lookup Approaches

In conventional approaches, random samples of joint configurations are drawn from a chosen distribution (*e.g.* uniform), and evaluated with regards to task-space objectives and workspace limitations. Valid configurations may need to fulfil additional criteria such as stability or collision avoidance. *Cartesian* poses and joint values are then associated within a search-tree structure for quick lookup queries. Functional approximations can be further used to interpolate between neighbouring samples. The size of the lookup table, however, quickly increases with the number of random configurations considered. Furthermore, uniform sampling of the joint-space likely results in non-uniform sampling of the task- space, implying that some volumes are over-sampled while others are under-sampled.

B. Associating Joint Configurations to Voxels

Using the scheme proposed here, each random joint configuration is associated to a discretised SE(3) voxel via

$$\boldsymbol{q} \rightarrow \boldsymbol{\mathcal{T}}(\boldsymbol{q}) \rightarrow \boldsymbol{v}_6, \tag{18}$$

where $\boldsymbol{q}$ is the robot configuration, $\boldsymbol{\mathcal{T}}(\boldsymbol{q})$ the corresponding end-effector pose, and $\boldsymbol{v}_6$ the SE(3) voxel.

Robotic manipulators, and in particular redundant manipulators, can reach certain task space poses with different joint configurations. Therefore, an approach is needed wherein such dicretised voxels $\boldsymbol{v}_6$ are associated with a plurality of robot configurations, but without risking the creation of unnecessarily large lookup tables.

In this proposed framework, each lookup voxel ($\mathcal{LU}_{\mathcal{V}_6}$) has a maximum capacity ($N_{\mathcal{LU}}$) of the number of joint configurations it can store, thereby controlling the amount of memory that the lookup table occupies. This way, increasing the number of random joint configuration samples also increases the probability of finding configurations in otherwise sparsely sampled volumes of the workspace while avoiding oversampling other areas. Moreover, this presented approach allows to suppress singular configurations which are characterised by joint-space changes resulting in marginal task-space variation. Singular configurations would otherwise create densely sampled volumes in the task-space without adding useful information on either dexterity or kinematics.

In the proposed approach new *incoming* randomly generated configurations are associated with voxels via a priority algorithm that maximises the variance of stored joint configurations in any voxel $\mathcal{LU}_{\mathcal{V}_6}$. Here, variance is calculated with respect to the configuration's joint values. Given that the number of configurations that each voxel can be associated with is limited to $N_{\mathcal{LU}}$, incoming configurations may replace existing ones only when variance increases.

Mathematically, it is assumed that a set of $\boldsymbol{Q}^{\mathrm{pre}}_{lu}$ configurations are stored in $\mathcal{LU}_{\mathcal{V}_6}$, while $\boldsymbol{q}_r$ is the incoming configuration. Then, all $(N_{\mathcal{LU}}+1)$ candidate configuration sets are generated, $\boldsymbol{Q}^{\mathrm{cand}}$, that result from either disregarding $\boldsymbol{q}_r$ or substituting any $\boldsymbol{q}_i \in \boldsymbol{Q}^{\mathrm{pre}}_{lu}$ with $\boldsymbol{q}_r$:

$$\boldsymbol{Q}^{\mathrm{cand}} = \left\{ \boldsymbol{Q}^{\mathrm{pre}}_{lu}, \bigcup_{\forall \boldsymbol{q}_i \in \boldsymbol{Q}^{\mathrm{pre}}_{lu}} \Big((\boldsymbol{Q}^{\mathrm{pre}}_{lu} \setminus \boldsymbol{q}_i) \cup \boldsymbol{q}_r \Big) \right\} \tag{19}$$

Subsequently, the variance for each configuration set within $\boldsymbol{Q}^{\mathrm{cand}}$ is computed, and the set with the maximum variance is chosen as the updated set to be associated with $\mathcal{LU}_{\mathcal{V}_6}$ according to:

$$\boldsymbol{Q}^{\mathrm{post}}_{lu} = \underset{\forall \boldsymbol{Q}^{\star} \in \boldsymbol{Q}^{\mathrm{cand}}}{\arg\max} \left\{ \boldsymbol{w}_q^T \mathrm{Var}[\boldsymbol{Q}^{\star}] \right\}, \tag{20}$$

where the joint weighting factor $\boldsymbol{w}_q$ can be used to emphasise different joints, *e.g.* distal joints over proximal joints.

It must be noted that computing the mean or distances of revolute joint values (periodic entities) requires scrutiny, [47, Ch. 2.3.8]. Specifically, before computing the mean, the joint value represented as an angle on the unit circle is converted into *Cartesian* coordinates:

$$\bar{\boldsymbol{q}}_i = \mathrm{atan2} \left\{ \frac{1}{N_{\mathcal{LU}}} \sum_{j=1}^{N_{\mathcal{LU}}} \sin\left(q_{i,j}\right), \frac{1}{N_{\mathcal{LU}}} \sum_{j=1}^{N_{\mathcal{LU}}} \cos\left(q_{i,j}\right) \right\}. \tag{21}$$

Additionally, the difference between angles is always wrapped to the interval $[-\pi, \pi]$ to obtain a meaningful variance metric.

C. Inverse Kinematic Solver using Lookup

The generated lookup table can then be used as a maximum-time inverse kinematic solver. The lookup-table is stored using a dense tensor representation. A vertex $\boldsymbol{v}_6$ and associated translation ($\boldsymbol{I}^{\star}_T$) and rotation ($I^{\star}_R$) indices are determined in constant-time given a target pose $\boldsymbol{\mathcal{T}}^{\star}$ in SE(3). To find the most suitable configuration, a radius search is used with a computational complexity linear to the covered volume. The distance of $\boldsymbol{I}^{\star}_T$ and $I^{\star}_R$ to the lookup indices $\boldsymbol{I}_T$ (3) and I_R (6) represent the respective search radii in translation (r_{I_T}) and rotation (r_{I_R}). The maximum capacity of configurations per

voxel ($N_{\mathcal{LU}}$) is also linearly correlated to the computational complexity.

The corresponding lookup voxel $\mathcal{LU}_{\mathcal{V}_6}$ stores joint configurations with end-effector poses that are associated with the same *Voronoi* cell. Sufficiently fine voxelisations imply that the stored configurations correspond to the approximately same target pose. If the corresponding voxel $\mathcal{LU}_{\mathcal{V}_6}$ is empty, neighbouring voxels can be queried. The query might result into multiple possible joint configuration candidates which have an end-effector pose sufficiently close to the target pose.

A secondary selection criteria can be proximity between current and next configuration in joint-space. Certain tasks such as telemanipulation, prefer task-space trajectories which are associated with small joint value increments, *i.e.* small joint velocities.

To ensure that the looked up joint configuration does not require unacceptably high joint velocities from the respective current configuration, a maximum distance criteria is employed:

$$\left|\boldsymbol{q}_{\text{current}} - \boldsymbol{q}_{\text{lu},i}\right| \leq \Delta q_{\max}, \tag{22}$$

where $\boldsymbol{q}_{\text{current}}$ is the current joint configuration or the robot and $\boldsymbol{q}_{\text{lu},i}$ a looked up configuration.

Recall that up to $N_{\mathcal{LU}}$ joint configurations may be associated with vertex $\boldsymbol{v}_6$, while sufficiently close configurations may also be associated with neighbouring task-space voxels. To compare the entire set of relevant configurations, radii were defined in translational (r_{I_T}) and rotational (r_{I_R}) voxel space to provide a search boundary. All configurations, which are associated to voxels within that boundary, are considered, and a cost based on the distance between a lookup configuration and the current configuration in joint space ($\boldsymbol{\Delta}q$) and target configuration in task space ($\boldsymbol{\Delta}x$) is minimised to find the optimal joint configuration for the given target pose:

$$\boldsymbol{\Delta}q_i = \boldsymbol{q}_{\text{current}} - \boldsymbol{q}_{\text{lu},i} \tag{23}$$

$$\boldsymbol{\Delta}x_i = \boldsymbol{x}_{\text{target}} - \boldsymbol{x}\left(\boldsymbol{q}_{\text{lu},i}\right) \tag{24}$$

$$\boldsymbol{q}^{\star}_{\text{lu}} = \arg\min_i \left\{ \left(\boldsymbol{w}_q^T \boldsymbol{\Delta}q_i\right)^2 + \left(\boldsymbol{w}_x^T \boldsymbol{\Delta}x_i\right)^2 \right\}, \tag{25}$$

where $\boldsymbol{q}^{\star}_{\text{lu}}$ represents the optimal joint configuration, and $\boldsymbol{w}_q$ and $\boldsymbol{w}_x$ weights to compute the overall cost of the lookup configuration considering proximity to the previous joint configuration ($\boldsymbol{\Delta}q_i$) and proximity to the target pose ($\boldsymbol{\Delta}x_i$).

The search boundary governed by r_{I_T} and r_{I_R} expands until either a suitable joint configuration is found, or the distance to the target pose exceeds a threshold that signifies lookup failure.

V. Robot Architecture Optimisation

A function which associates a robot architecture ($\mathcal{A}_i$) to a scalar metric that corresponds to its task-space capabilities ($\boldsymbol{\mathcal{C}}_{\mathcal{T}}$) enables algorithmic robot optimisation for a specific task:

$$\mathcal{A}^{\star}_R = \arg\max_{\mathcal{A}_i} \left\{ \boldsymbol{\mathcal{C}}_{\mathcal{T}}\left(\mathcal{A}_i\right) \right\}, \tag{26}$$

where $\mathcal{A}^{\star}_R$ is the optimal robot architecture. It can be found using appropriate optimisation algorithms, keeping in mind the non-convexity of the problem in the general case.

A metric should enable comparisons of significantly different robot architectures and dimensionalities. Global metrics such as the introduced global dexterity are superior to local approximations when the design of robots with improved dexterity is considered. The dexterity computation presented in Sec. III-B fulfils these requirements when used in conjunction with a function which extracts from the global capability measure based on task-specific preferences ($\boldsymbol{\mathcal{C}}_{\mathcal{V}}$) the task-specific capabilities of the robot ($\boldsymbol{\mathcal{C}}_{\mathcal{T}}$).

A composition between the dexterity measure $\mathcal{D}_{\mathcal{V}_6}$ and function $\boldsymbol{\mathcal{C}}_{\mathcal{V}}$, which could be a continuous parameterised trajectory or volumetric function, can be expressed as:

$$\boldsymbol{\mathcal{C}}_{\mathcal{T}}\left(\mathcal{A}_i\right) = \mathcal{D}_{\int_T}\left(\mathcal{A}_i\right) = \int_{v_6} \left(\boldsymbol{\mathcal{C}}_{\mathcal{V}} \circ \mathcal{D}_{\mathcal{V}_6}\left(\mathcal{A}_i\right)\right)\left(v_6\right). \tag{27}$$

This generic representation can for example be implemented in the discretised case as the sum over the dexterity voxels, which were previously scaled with $\boldsymbol{\mathcal{C}}_{\mathcal{V}}$ to compute $\boldsymbol{\mathcal{C}}_{\mathcal{T}}$:

$$\mathcal{D}_{\Sigma_T}\left(\mathcal{A}_i\right) = \sum_{\forall v_6 \in \mathcal{D}_{\mathcal{V}}} \boldsymbol{\mathcal{C}}_{\mathcal{V}}\left(\mathcal{D}_{\mathcal{V}_6}\left(\mathcal{A}_i, v_6\right)\right). \tag{28}$$

This idea was previously presented in Sec. III-C in form of a masking approach. The mask ($\boldsymbol{\mathcal{C}}_{\mathcal{V}}$) represents the task-space specific preferences to promote those *Cartesian* poses, which are most relevant to perform a task and to suppress the weight of those which are not important to achieve a preferred task-space goal.

When the specific task is not known (*e.g.* in telemanipulation) the masking approach depicted in Fig. 9 and formalised in (28) is even more useful. Instead of optimising the robot for a specific trajectory to the task-space target, it allows for a fuzzy approach in which 6 DoF volumes or *Cartesian* poses of high task relevance are described. In other words, an *a priori* knowledge of the exact end-effector path is not required, and instead regions of interest can be used.

VI. Experiments

To demonstrate different use cases and the versatility of the presented 6 DoF voxelisation approach, experiments were conducted with: (i) a serial link robotic arm, (ii) a serial link flexible surgical instrument, and (iii) a flexible concentric tube robot.

A. Voxelisation-based Dexterity

Robot dexterity is evaluated with three methods. The first method uses the common manipulability metric defined in (8). It is denoted as $\mathcal{D}^L_{\mathcal{V}_3}$, and it is based on the end-effector Jacobian, and therefore describes the local neighbourhood of a configuration.

The second method was introduced in Sec. III-B. For each position $\in \mathbb{R}^3$, the algorithm computes what proportion of all discretised orientations can be reached. The metric, denoted as $\mathcal{D}^G_{\mathcal{V}_3}$, is based on a global evaluation but does not explicitly take into account whether the manipulator can directly move between neighbouring voxels.

The third method was introduced in Sec. III-D. Evaluation uses metric $\mathcal{D}^G_{\mathcal{V}_3}$ while also considering whether the manipulator can move from one configuration and its corresponding

end-effector voxel in $\mathbb{R}^6$ towards neighbouring voxels in $\mathbb{R}^6$. The resulting metric, denoted as $\mathcal{D}^{G,L}_{\mathcal{V}_3}$, is a globally evaluated metric that incorporates local behaviour.

All computations use a self-collision model, see Fig. 10. A simple capsule based model was chosen to enable Graphics Processing Unit (GPU) accelerated computation. The model is conservative insofar as it potentially rejects configurations which are collision-free.

1) Serial Link Robot: In the first experiment the dexterity of a 7 DoF serial link robot (KUKA LBR iiwa 7 R800, see specifications: *http://tiny.cc/x9z2vz*) is evaluated.

The resulting dexterity maps are depicted in Fig. 11. The colours represent the respective metrics for each voxel. A high dexterity is represented by a red (warm) colour while a position in which the robot has a low dexterity is represented with a blue (cold) colour. Please note that the values of the metrics differ between the subfigures. The dexterity value for $\mathcal{D}^{L}_{\mathcal{V}_3}$ in Fig. 11(a) is the maximum manipulability $\mathcal{M}$, see (8), achieved at the end-effector position, and is not easily interpreted. This method primarily allows for a relative assessment of dexterity.

On the contrary, for $\mathcal{D}^{G}_{\mathcal{V}_3}$ and $\mathcal{D}^{G,L}_{\mathcal{V}_3}$, depicted in Fig. 11(b) and Fig. 11(c), the dexterity value represents the proportion of orientations which are achieved based on all possible (here: 3240) rotation voxels. Crucially, these metrics allow to conclude that the assessed manipulator can reach all voxels within the inner red volume under any desired orientation. For the metric $\mathcal{D}^{G,L}_{\mathcal{V}_3}$ the 12-voxel neighbourhood with $w_n = 0.8$ was used. The memory usage for the dexterity data structure that differentiates between $125 \times 125 \times 116 \times 3240 = 5872.5 \times 10^6$ voxels, requires 4 bytes per voxel (using single precision floating point numbers). This leads to 21.87 GB of RAM, which was designed to suit the memory of the used GPU.

There are notable differences between method $\mathcal{D}^{L}_{\mathcal{V}_3}$, see Fig. 11(a), and the other two methods (b), (c). $\mathcal{D}^{L}_{\mathcal{V}_3}$ suggests that the dexterity is highest around the ground plate that the robot is standing on, see Fig. 11(a) - circle 1. Furthermore, that approach suggests that the manipulator is less dexterous at the centre at approximately 900 mm height, see Fig. 11(a) - circle 2. The global approaches (b), (c) show a corridor of uniformly high dexterity in an approximate (red) Ω-shape.

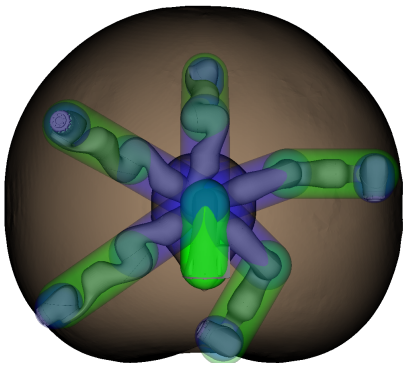

Fig. 10. Collision model for the 'KUKA LBR iiwa 7 R800' used during the dexterity analysis experiment. The green and blue capsules represent the geometric collision primitives attached to the robotic links. The brown background represents the reachable workspace of the manipulator.

The differences between $\mathcal{D}^{G}_{\mathcal{V}_3}$, and $\mathcal{D}^{G,L}_{\mathcal{V}_3}$ are more subtle, with the main difference being under the robot base, see Fig. 11(b) - circle 3. Considering the local neighbourhood, the volume under the base is described with lower dexterity then when only considering 3D reachability.

Both of the introduced metrics lead to similar evaluations. The additional computational burden to include local considerations is therefore in most cases only justified when the boundaries of the workspace are of particular interest.

2) Flexible Surgical Instrument: The second experiment assesses the dexterity of different kinematic architectures of same dimensions for a surgical instrument detailed in [48]. It showcases the advantage of meaningful dexterity metrics and, in particular, the importance of meaningful absolute values that inform a technical reviewer about the ratio of the desired workspace that is reachable.

Classic metric $\mathcal{D}^{L}_{\mathcal{V}_3}$ was used to evaluate the capabilities of two different manipulators, and results are shown in Fig. 12(a), (b). Metric $\mathcal{D}^{G}_{\mathcal{V}_3}$ was also employed for the same manipulators, see Fig. 12(c), (d). Fig. 12(a), (c) refers to a 5 DoF tool with pitch-yaw-yaw-pitch-yaw (PYYPY) configuration, while Fig. 12(b), (d) refers to a 6 DoF tool with a pitch-yaw-yaw-pitch-yaw-roll (PYYPYR) configuration.

First, consider Fig. 12(a) and Fig. 12(b). The classic dexterity metric, $\mathcal{D}^{L}_{\mathcal{V}_3}$, shows fairly uniform regions of dexterity in both cases. As expected, the 5 DoF tool is overall less dexterous, even up to an order of magnitude, when compared to the 6 DoF tool. The classic approach shows homogeneously low dexterity of the under-actuated 5 DoF tool (overdetermined system) since it is based on the eigenvalues of the Jacobian matrix. Finally, fine details on dexterity regions cannot be readily identified.

Next, Fig. 12(c) is compared to Fig. 12(d). The voxelisation approach highlights the different regions of high dexterity within the otherwise identical robot workspaces. In addition, it provides quantitative information on the orientation coverage for the two manipulators. Specifically, even in its most dexterous region, the 5 DoF can reach less than 2.5% of orientations, while the 6 DoF can reach almost 25%.

The metric introduced in this paper allows for a direct comparison with Fig. 11 as well, where almost 100% orientation coverage is observed. Such conclusions cannot be derived through the use of Jacobian-based dexterity values. Notably, $\mathcal{D}^{L}_{\mathcal{V}_3}$ in Fig. 12 is higher in comparison to Fig. 11 since the unit of the link-lengths for the instrument are millimetre (*mm*) while for the robotic manipulator the unit was in meter (*m*).

B. Inverse Kinematics

The third experiment evaluates the inverse kinematic approach presented in Sec. IV. The voxelised workspace is used to store multiple joint configurations in the 6 DoF voxels of SE(3). In this experiment the amount of joint configurations stored per voxel is bound to 5 configurations, and thus reduces the amount of required memory while being able to use large numbers of samples to generate an evenly dense lookup table. Limiting the number of stored configurations can result

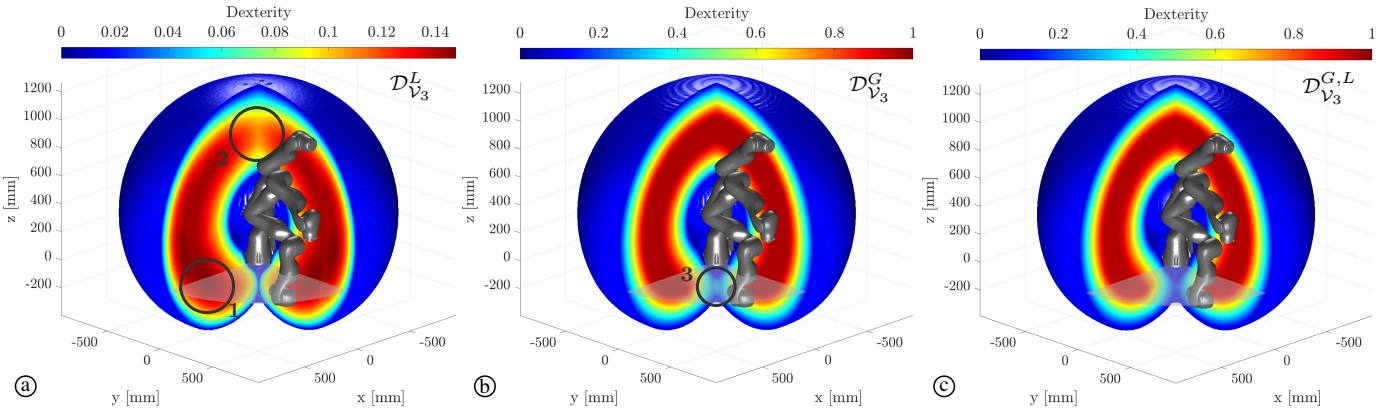

Fig. 11. Dexterity assessment based on kinematic model of an iiwa 7 robot of the KUKA AG. In each analysis the workspace was divided in 125 × 125 × 116 voxels for the position. In (b) and (c) the rotation space was further divided in 3240 rotation voxels. Each dexterity analysis considers 10^{13} random configuration samples. Dexterity measures used: (a) $\mathcal{D}^L_{\mathcal{V}_3}$, (b) $\mathcal{D}^G_{\mathcal{V}_3}$, (c) $\mathcal{D}^{G,L}_{\mathcal{V}_3}$. Gray circles in (a) describe area of significant difference compared to (b) and (c), grey circle in (b) shows subtle difference to (c).

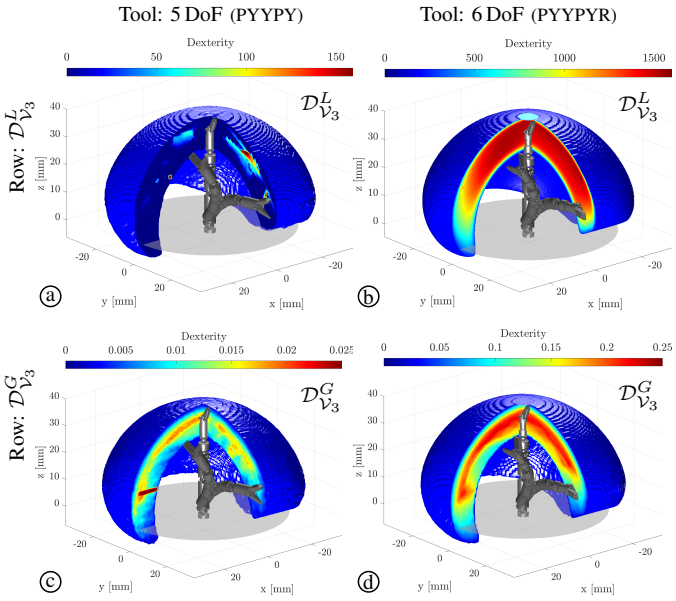

Fig. 12. Dexterity assessment based on kinematic model of custom surgical tool. Each experiment considers 10^{13} configurations. (a) $\mathcal{D}^L_{\mathcal{V}_3}$ of 5 DoF tool, (b) $\mathcal{D}^L_{\mathcal{V}_3}$ of 6 DoF tool, (c) $\mathcal{D}^G_{\mathcal{V}_3}$ of 5 DoF, (d) $\mathcal{D}^G_{\mathcal{V}_3}$ of 6 DoF tool.

into sub-optimal lookup results, while a high number of configurations can lead to uneven sampling of the workspace as discussed in section IV. Given a limited amount of computing memory, both the number of voxels for the evaluated workspace and the number of maximum stored configurations per voxel need to be considered. For dexterous architectures, 1.5 GB of disk-space was used to store the lookup table.

Four different CTR architectures aimed at vitreoretinal surgery were used, with a structural design similar to [12]. In vitreoretinal surgery, the robot enters the eye from an incision on the sclera (the white part of the eye) and extends up to the retina, where surgery takes place. Increased dexterity is required to precisely carry out interventions, and in particular deliver novel treatments, such as regenerative therapies [49]. An in-depth introduction of this surgical domain is out of the scope of this paper; please refer to [50] for a review of robotic retinal surgery. The four CTR architectures were chosen to have varying suitability for the task, and thus assess to what extent an advanced inverse kinematics algorithm can compensate for design imperfections. The architectures are intermediate steps of a design optimisation algorithm, detailed later, and are denoted as $\mathcal{I}_{\mathcal{O},0}$ (sub-optimal), $\mathcal{I}_{\mathcal{O},3}$ (unsuitable), $\mathcal{I}_{\mathcal{O},7}$ (unsuitable), and $\mathcal{I}_{\mathcal{O},11}$ (optimised).

The simulated telemanipulation task entails inserting the robot through the sclera and the vitreous body to reach a point close to the macula (central part of the retina). The target is indicated with coordinate axes in Fig. 16. Once there, the end-effector was pivoted $\pm 60°$ around the x, y-axis (pitch, yaw) and $360°$ around the z-axis (roll), before the CTR was retracted back to the origin. It is important to note that while CTRs have so far been controlled precisely with regards to their tip position, orientation control is more challenging. While different tip orientations lie close in task space, their associated configurations (joint space) are far away. The proposed framework and lookup approach is perfectly suited to address this challenge.

Following the framework introduced by the authors in [51], optimisation algorithms compete in parallel to deliver the best inverse kinematics solution. Two different inverse kinematic approaches were used to solve the inverse kinematics for each simulated telemanipulation task. The first approach employed two standard local non-linear optimisation algorithms (BOBYQA and Nelder-Mead) [52]. The competing second approach additionally employed the lookup table, resulting in utilisation of three optimisers. The idea of combining the local non-linear optimisers with the lookup approach arose from the observation that local optimisers regularly converge to sub-optimal local minima. The global lookup provides a good initial guess for the local optimisers to improve, and helps steer away from local minima. The optimisers and the lookup approach were timed to provide their best result every 100 ms, which resulted into a control rate of 10 Hz. To improve motion smoothness, joint level based trajectory planning is envisioned. Alternatively, previous research [51], has shown using optimisers with different timeouts 10 Hz - 50 Hz improves latency in areas in which optimised results are

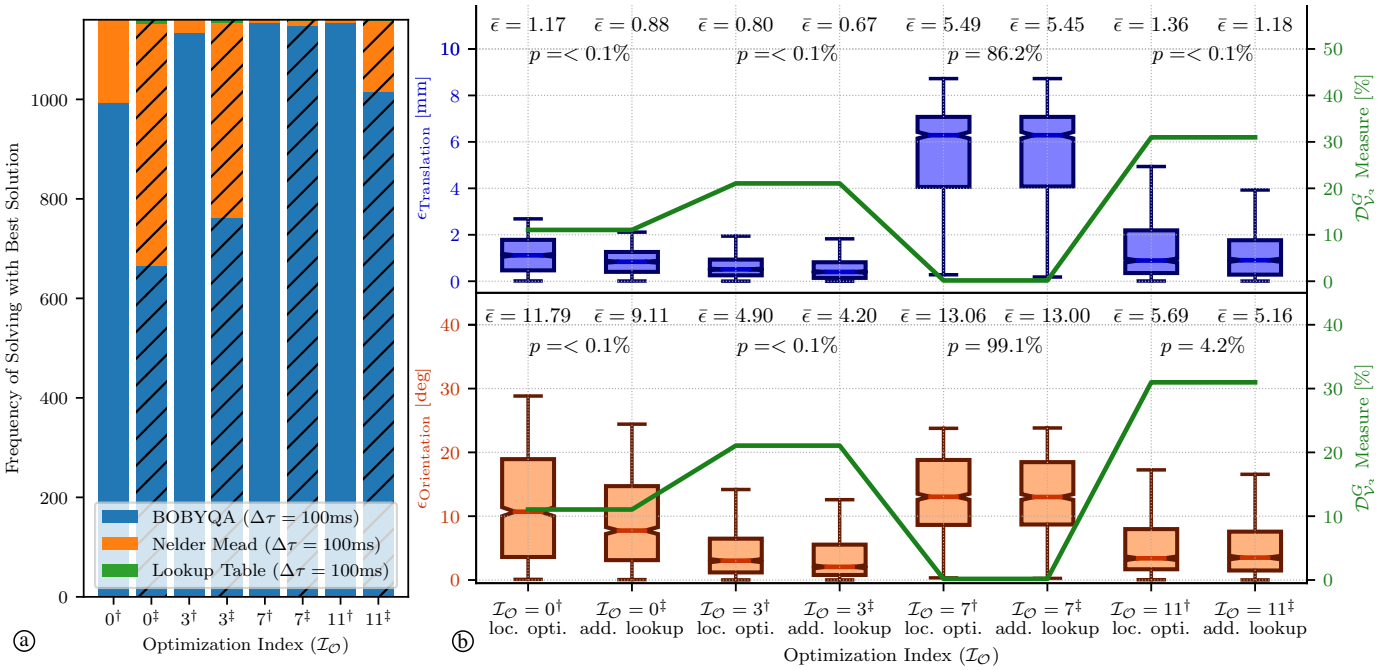

Fig. 13. Voxelisation is used to store multiple joint configurations of a CTR per voxel. This information is then used for inverse kinematic lookup. Comparison of inverse kinematic performance when using purely local optimisers (†) and combining local optimisers with the lookup (‡) (also hatched in (a)) to provide seed configurations. Different CTR architectures ($\mathcal{I}_\mathcal{O} \in \{0, 3, 7, 11\}$) were used. $\mathcal{D}^G_{\mathcal{V}_3}$ is the reachability of the robot architecture relative to a perfect workspace reachability. The lookup was based on 10^8 samples. (a) Statistic of best performing optimiser, over 1160 consecutive inverse kinematic tasks, with 100 ms per task. The box plots in (b) represent the performance over the entire trajectory with 1160 samples of an equally spaced time series.

easily found. The comparative results are presented in Fig. 13.

The results in Fig. 13(b) show that the introduction of the lookup table as a first step in the optimisation leads to a mean (ϵ) that demonstrates up to 25% improvement, both in terms of position and orientation errors. Only for the architecture with the index $\mathcal{I}_{\mathcal{O},7}$ the results show no significant difference. This can be explained by the large error this architecture has in position and orientation. A detailed investigation revealed that the inverse kinematic task was largely outside of the workspace of this specific architecture and hence was not capable of following the target end-effector pose.

The frequency of which optimiser finds the best solution is depicted in Fig. 13(a). The figure shows that the lookup table itself would only very rarely provide the best result. Using it, however, significantly boosts the performance of the Nelder-Mead algorithm, which implies that the latter benefited from the initial guesses provided by the lookup table. Previous experiments using only the Nelder-Mead algorithm showed that it suffers from local minima. Here, this issue was eliminated by combining it with the lookup approach, which also resulted in a better convergence result overall.

C. Manipulator Architecture Optimisation

The final experiment demonstrates how the developed metrics can be used to computationally design robots optimised for dexterous manipulation. The experiment is divided into two parts: (i) robot architecture optimisation, according to (26), based on the robot's workspace dexterity, and (ii) the architecture evaluation, based on the error of a concrete inverse kinematic task, which is used to simulate telemanipulation.

A specific trajectory for architecture optimisation was avoided, as telemanipulation tasks are determined in the moment by the operator and take place across the robot's workspace. Robot architecture optimisation, and robot architecture evaluation, therefore use different metrics. A discussion on how the two metrics compare is provided in the following and depicted in Fig. 14.

Similar to the previous section, the goal is to maximise the dexterity of a CTR to carry out vitreoretinal surgery and ultimately regenerative therapy delivery. Such tasks require extensive workspace coverage on the retina, in addition to high orientation dexterity at every point; this latter property is rarely a feature of CTR.

The voxelisation approach is used to compute a metric that describes the dexterity of a robot architecture for a given workspace. Rather than directly solving the inverse kinematics, the metric predicts the capability of a robot to execute an inverse kinematic task. The hypothesis is that increased dexterity metric corresponds to increased inverse kinematic convergence, especially considering reachability of neighbours as per the developed theory.

The metric $\mathcal{D}_{\Sigma_T}$ formalised in (27) was used, which combines the voxelisation approach together with the masking approach. It is the sum of the masked dexterity values in the workspace. In the following the metric $\mathcal{D}_{\Sigma_T}$ is based on dexterity measure $\mathcal{D}^G_{\mathcal{V}_3}$.

Starting from a guessed CTR, the BOBYQA algorithm, was used to identify the optimised lengths and curvatures of each of the CTR tubes. The BOBYQA algorithm is a local nonlinear optimisation algorithm which performed well in initial

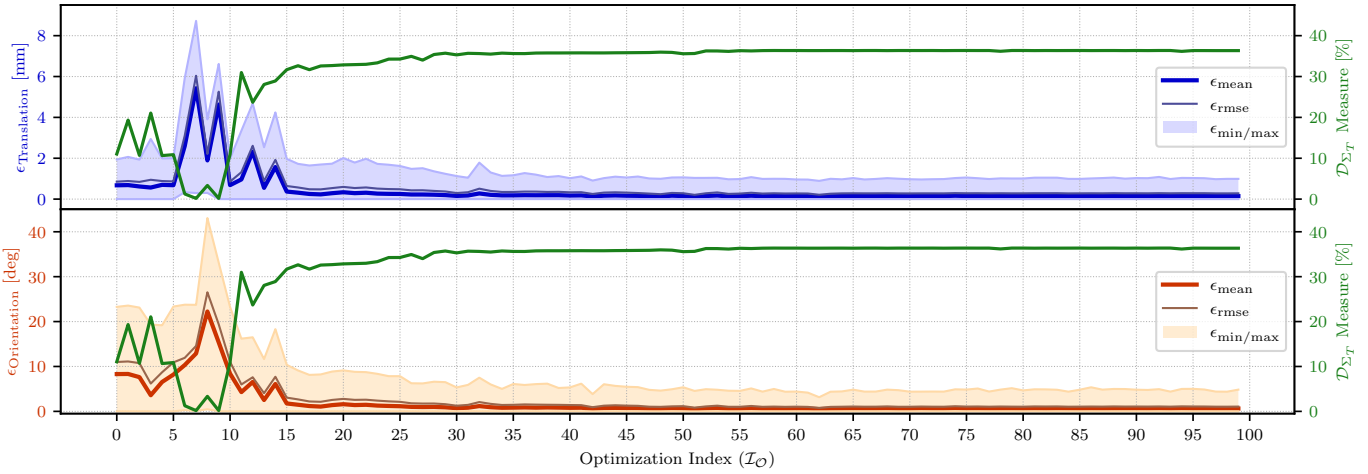

Fig. 14. Optimising CTR architecture: Dexterity ($\mathcal{D}_{\Sigma_T}$) visualisation and inverse kinematics error for each CTR architecture during optimisation. For every index $\mathcal{I}_{\mathcal{O}}$ the error (ϵ) statistics of the full trajectory are depicted. ϵ_{mean}: mean error along the trajectory, ϵ_{rmse}: root-mean-square error, $\epsilon_{\text{min/max}}$: minimum and maximum error. $\mathcal{D}^{G}_{\mathcal{V}_3}$ is the reachability of the robot architecture relative to a perfect workspace reachability.

tests and it is characterised to suffer little from local minima, and additionally has a fast convergence rate.

The considered robot architecture comprises variable curvature (VC) sections, and constant curvature (CC) sections. As first introduced in [1], a VC comprises two tubes of identical stiffness, whose rotations are independent but the translations are coupled, and delivers 3 DoF. A CC comprises a single tube, and delivers 2 DoF. The CTR architecture that was chosen to optimise for consists of a VC, a curved CC, and a straight tool-carrying CC with minimal length (1 DoF), for a total of 6 DoF.

The optimisation output is shown in Table I, while convergence plots are depicted in Fig. 14.

Fig. 14 shows how the dexterity measure ($\mathcal{D}_{\Sigma_T}$) developed over each optimisation step. It is visualised as a relative value based on the ideal case in which all voxels within the desired workspace can be reached. During the optimisation CTR architecture parameterisations were assessed to maximise the CTR dexterity score ($\mathcal{D}_{\Sigma_T}$). After the optimisation concluded the same parameterisations were further evaluated in regards to how well they perform for a simulated telemanipulation task. This second evaluation is used to assess the hypothesis that high dexterity scores correlate to low inverse kinematics errors.

This simulated telemanipulation task was identical to the inverse kinematic experiment described in Sec. VI-B.

The average inverse kinematic error, along the trajectory, between desired end-effector and achieved end-effector pose in translation (mm) and orientation ($^\circ$) for each optimisation step is shown in Fig. 14. The dexterity measure quickly converges and the errors for translation and orientation reduce with increasing dexterity. The tube parameters evaluated first ($\mathcal{I}_{\mathcal{O},\{5-10\}}$) resulted into poor dexterity ($\mathcal{D}_{\Sigma_T}$) values, and poor performance in the telemanipulation task. Fig. 16 depicts CTRs during the telemanipulation task to the same pose while following the trajectory. The previously computed dexterity metric for the architecture's workspace is overlaid. A subset of those architecture's was discussed in Sec. VI-B. Fig. 16(a) depicts the initial architecture, which is not quite able to reach the desired end-effector pose. Intermediate architectures Fig. 16(b)-(c), with very low manipulability scores, cannot reach the desired pose. Finally, Fig. 16(d), which depicts the optimised CTR, reaches the desired pose with overall increase manipulability.

The analysis of how the telemanipulation error, the average translation and rotation error over the entire trajectory, corresponds to dexterity measure is depicted in Fig. 15. Estimation

TABLE I
CTR ARCHITECTURE OPTIMISATION PARAMETERS

Sec (Type)	Tube	Stiffness Ratio	Radius $[mm]$	Curvature* $[mm^{-1}]$	Curved Length* $[mm]$
0 (VC)	0	5 : 1	0.3125	$52.20 \in [0, 200]$	$15 \in [2, 45]$
	1	5 : 1	0.29	$52.20 \in [0, 200]$	$15 \in [2, 45]$
1 (CC)	2	1 : 1	0.25	$52.63 \in [0, 200]$	$15 \in [2, 45]$
2 (CC)	3	0.05 : 1	0.125	0	$1 \in [0.5, 2]$

*Value range for optimisation specified as: [min, max]

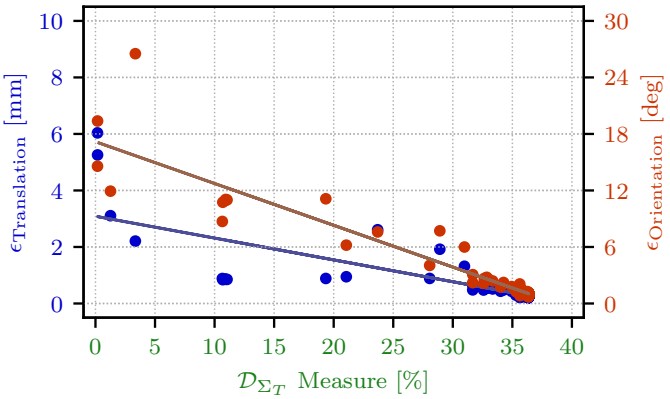

Fig. 15. Optimising CTR architecture: Relation between dexterity $\mathcal{D}_{\Sigma_T}$ and inverse kinematics error. Pearson correlation coefficient (ρ): -0.810 (Translation), -0.946 (Orientation), p-value: $< 0.1\%$ (both). The relation shows that a high dexterity measure ($\mathcal{D}_{\Sigma_T}$) results in small inverse kinematic errors in position and orientation. $\mathcal{D}_{\Sigma_T}$: measures the reachability of the respective robot architecture relative to a perfect reachability of the entire workspace.

of the correlation coefficient shows that the telemanipulation error is negatively correlated to the dexterity for both position and orientation (highly significant). This confirms the hypothesis that increased dexterity as calculated by metric $\mathcal{D}_{\Sigma_T}$ is linked with improved inverse kinematics performance.

The error metrics for the considered CTRs are summarised in Table II. The table shows that initial errors ($\mathcal{I}_{\mathcal{O},0}$) of up to $1.94\,mm$ and $23.27\,^\circ$ were reduced to ($\mathcal{I}_{\mathcal{O},99}$) worst case errors in the sub-millimetre range for the position and less than $5\,^\circ$ in orientation. The average error for the optimised architecture was $0.15\,mm$ and $0.63\,^\circ$.

It is important to note that this is the first paper that considers orientational dexterity in CTR design. A preliminary vitreoretinal surgery CTR developed for reachability in [12] achieved only 0.92 % of the maximum achievable dexterity compared to more than 36 % achieved with the presented approach. The maximum achievable dexterity considers a simple task-space priority model, which factors in a wide range of forward facing orientations and locations in the eye.

D. Computational Performance

For all experiments, the majority of the computing tasks were performed on the GPU using a OpenCL™ language implementation. CTR experiments were performed on an Intel® Core™ i7-6900K CPU with 2× Nvidia Quadro P6000 (2× 12.6 TFLOPS) GPUs. Computing the dexterity for a single architecture considered 10^8 random joint configurations and lasted 108.7 s. The CTR optimisation experiment with 100 iterations lasted approximately 3 h. The generation of the inverse kinematic lookup table completed on average in 686 s.

Computing the forward kinematics for serial-link robots is significantly faster than for CTR due to kinematics simplicity. The following metrics for computing the workspace for the iiwa 7 robot were achieved with an Intel® Core™ i7-8850H CPU with an Nvidia GeForce 1050 Ti (2.1 TFLOPS) GPU. The workspace was divided into $75 \times 75 \times 70 = 393,750$ positional voxels with each further subdivided in 420 unique rotations for the 6 DoF approaches. This resulted into 165.375×10^6 unique voxels. The workspace was assessed with $N_{\text{sample}} = 10^{11}$ random uniformly distributed joint-configurations. Computing $\mathcal{D}^L_{\mathcal{V}_3}$ took 440 min, $\mathcal{D}^G_{\mathcal{V}_3}$ took 210 min, and $\mathcal{D}^{G,L}_{\mathcal{V}_3}$ took 500 min. This renders the presented metric $\mathcal{D}^G_{\mathcal{V}_3}$ significantly faster than the manipulability based approach $\mathcal{D}^L_{\mathcal{V}_3}$. The presented global dexterity measure approach that considers also reachability of the 12 neighbour voxels $\mathcal{D}^{G,L}_{\mathcal{V}_3}$ took only 13.6% longer than the manipulability based approach.

TABLE II
TELEMANIPULATION ERROR OF CTR ARCHITECTURES

$\mathcal{I}_\mathcal{O}$	$\mathcal{D}_{\Sigma_T}$	ϵ_{mean} $[mm/^\circ]$	ϵ_{rmse} $[mm/^\circ]$	ϵ_{min} $[mm/^\circ]$	ϵ_{max} $[mm/^\circ]$
0	11.0%	0.68 / 8.31	0.85 / 11.00	0.00 / 0.01	1.94 / 23.27
3	21.1%	0.57 / 3.59	0.95 / 6.19	0.00 / 0.01	2.94 / 19.35
7	0.2%	5.43 / 12.89	6.04 / 14.57	0.30 / 0.09	8.72 / 23.72
11	31.0%	0.96 / 4.29	1.32 / 6.00	0.00 / 0.02	3.35 / 16.19
25	34.3%	0.25 / 1.15	0.48 / 2.13	0.00 / 0.00	1.62 / 7.84
99	36.3%	0.15 / 0.63	0.28 / 1.07	0.00 / 0.01	0.99 / 4.84

It should be noted that the manipulability based approach does not subdivide the rotational space, which arguably could justify a lower number of samples as there are less unique voxels. In our experience, the number of samples used to assess the workspace should be 100× - 1000× larger than the number of unique voxels; this ensures that the result is not influenced by undersampling. Furthermore, the degrees-of-freedom (N_{DoF}) of the manipulator should be considered. For a total number of samples N_{sample}, the equivalent number of samples per joint is calculated as $N_{\text{sample, joint}} = \sqrt[N_{\text{DoF}}]{N_{\text{sample}}}$. Then, if every joint was subdivided in $N_{\text{sample, joint}} = \sqrt[7]{10^{11}} \approx 37.28$ samples, the overall number of joint-configuration samples would be 10^{11}.

VII. CONCLUSION

Different approaches can be used to assess the dexterity of manipulators. Common approaches assess the dexterity through Jacobian based methods, which linearise the mapping between joint- and task-space. These methods have the disadvantage of providing information only on manipulability in the neighbourhood of the configuration under examination, and do

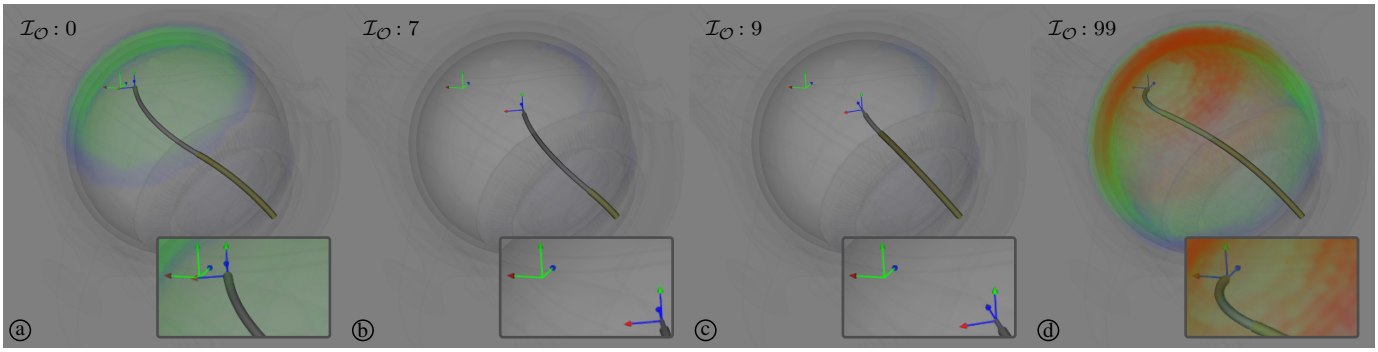

Fig. 16. Optimising CTR architecture: Visualisation of the inverse kinematics error for a specific tip pose during a predefined trajectory. Evaluation of Optimisation Index $\mathcal{I}_\mathcal{O} \in \{0, 7, 9, 99\}$. (a): the architecture is not able to reach the target neither in position nor orientation. (b), (c): the middle two architectures are neither able to reach the target the error is higher than in (a), (d): the architecture is able to reach the target pose in both position and orientation. The dexterity metric $\mathcal{D}^G_{\mathcal{V}_3}$ is overlayed as heat map for 4 different architectures. Dexterity ($\mathcal{D}^G_{\mathcal{V}_3}$) visualisation [■: high dexterity, ■: medium dexterity, ■: low dexterity]. (a): the overlay shows some medium and low dexterity regions, (b), (c): the overlay shows no dexterity or low dexterity regions, (d): the overlay shows high and medium dexterity regions. The overlays show that higher dexterity indicated by warmer colours results into smaller inverse kinematic errors.

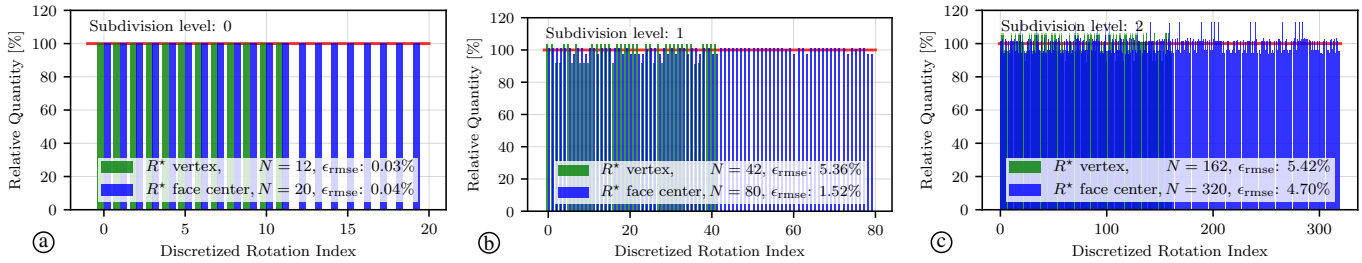

Fig. 17. Subdivision of an icosahedron surface. Statistical analysis of the frequency of associating a random rotation to each discretised rotation relative to the expected frequency. For each level 10^9 uniform random rotations are drawn. Vertices used for discretised rotations (green): (a) $\epsilon_{\text{rmse}} = 0.03\%$, (b) $\epsilon_{\text{rmse}} = 5.36\%$, (c) $\epsilon_{\text{rmse}} = 5.42\%$, Face centres used as discretised rotations (blue): (a) $\epsilon_{\text{rmse}} = 0.04\%$, (b) $\epsilon_{\text{rmse}} = 1.52\%$, (c) $\epsilon_{\text{rmse}} = 4.70\%$. (a) χ^2-test confirms both approaches result in a uniform distribution with a 0.1% significance level. The subdivision errors mean that (b), (c) are not uniformly distributed with a 0.1% significance level.

not provide a global metric on the capabilities of a manipulator within a workspace.

In this article, an approach to subdivide the workspace in equal size volumes was presented as the first step to enable a global assessment of the capabilities of a manipulator. Using a highly parallel implementation, this voxelisation approach was used to assess the workspace of different types of manipulators. Furthermore, the approach was used to generate a lookup table which provide a mapping from task-space volumes to joint-space configurations. Using the presented voxelisation approach the generated lookup table has a limited amount of saved configurations per voxel, which enables performance similar to denser lookup tables but with a reduced memory footprint. Finally, the voxelisation approach was used in conjunction with applying a task-specific mask to the volumes to compute a summative single-value metric that describes the capabilities of a manipulator for a specific task. This enabled the optimisation of a flexible manipulator for a complex telemanipulation task.

The presented diverse set of experiments considered different types of manipulators alongside representative applications to demonstrate the advantage of the proposed approach. The paper demonstrates that the presented approaches have the potential to provide insights and solutions in a variety of robotic research questions around dexterous performance.

Software

Implementation details presented in this paper can be found at *http://code.leibrandt.eu*. In particular, an implementation of the 6 DoF task-space to joint-values lookup is available at *http://rwa.code.leibrandt.eu*.

Appendices

The uniform voxelisation of the rotational space is critical for the described workspace analysis. The following sections evaluate the error which is introduced by the subdivision algorithm introduced in Sec. II.

Appendix A
SO(2) - Subdivision Error

Analysis of the resulting error for the SO(2) case was conducted by drawing uniform random rotations in SO(2) using the approach described in [53], which leads to a random vector ($\boldsymbol{r}_{v_3}$) on the unit sphere, with:

$$u \in [0,1],\ v \in [0,1],$$
$$\boldsymbol{r}_{v_3} = \begin{bmatrix} 2\sqrt{u\,(1-u)}\cos(2\pi v) \\ 2\sqrt{u\,(1-u)}\sin(2\pi v) \\ 2\,u-1 \end{bmatrix} \tag{29}$$

with u, v drawn from uniform random distributions.

For each of the random vectors ($\boldsymbol{r}_{v_3}$), the closest unique vertex on the unit sphere, which represents the centre of a *Voronoi* cell, is found. A uniform discretisation implies that all discretised rotations are equally often associated to a random SO(2) rotation over an infinite larger number of random rotations. If a discretised rotation deviates from the mean number of rotations per discretised rotation, it would indicate that its *Voronoi* cell area is respectively smaller or larger than the average area size of the discretised rotations.

Fig. 17 shows the resulting histograms for the first 2 subdivision levels using one billion (10^9) random rotations per histogram. The green coloured histogram represents the relative quantity of how often a specific discretised rotation was associated to a random rotation using the vertices from the subdivision process. The blue histogram shows the relative quantity when using the face centres as discretised rotations. There are more faces per subdivision level than there are vertices.

The root-mean-square-error from an optimal subdivision is computed and also depicted in Fig. 17. Without a subdivision, both approaches give close to perfectly uniform discretised rotations. After one subdivision, the projection approach results into small errors when using vertices ($\approx 5.36\%$) and face centres ($\approx 1.52\%$) as discretised rotations. Notably, using the face centres result into a more uniform tessellation.

After the second subdivision the errors of both approaches are similar $\approx 5.42\%$ using the vertices and $\approx 4.70\%$ when using the face centres.

Please note that although the area size of the *Voronoi* cell (triangle) using face centres is smaller than *Voronoi* cell (hexagon) using the vertex both areas have the same max distance from the centre to the most distant rotation in the cell, see Fig. 5. For this reason, vertices are employed as *Voronoi* centres and discretised rotations. Using the hexagon *Voronoi* cell has a similar error compared to the triangle *Voronoi* cell

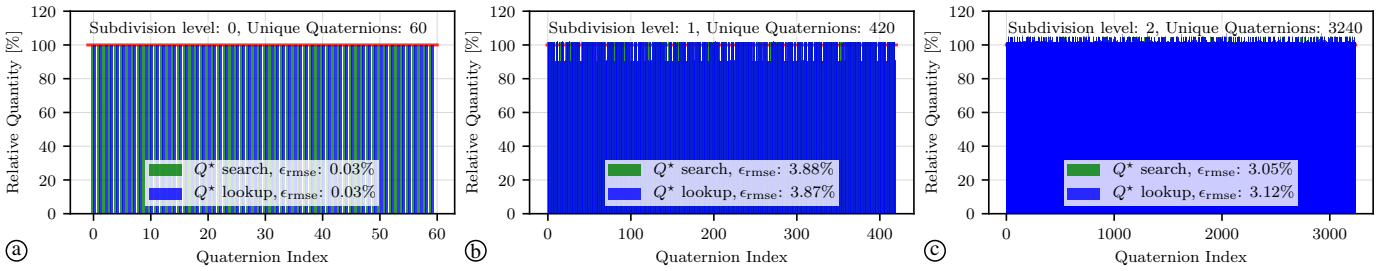

Fig. 18. Subdivision of the 600-cell. Statistical analysis of how often a random rotation is associated to each unique quaternion relative to the expected frequency. For each level 10^9 random rotations are drawn from a uniform distribution. Green: Linear search according to (5). Blue: Constant time lookup according to (6). (a) χ^2-test confirms both approaches result in uniform distribution with a 0.1% significance level. The subdivision errors mean that (b), (c) are not uniformly distributed with a 0.1% significance level.

at the subdivision level 2, which is used in this paper. The circle like shape of those hexagons was therefore the deciding factor in the choice of the *Voronoi* cell shape.

APPENDIX B
SO(3) - SUBDIVISION ERROR

The error analysis of the discretisation error for SO(3) was conducted similarly to the SO(2) analysis.

To show that the discretisation approach also works for SO(3) one billion (10^9) uniform random rotations are generated using the quaternion approach presented in [54], with

$$u \in [0,1],\ v \in [0,1],\ w \in [0,1],$$
$$\mathcal{Q}_r = \begin{bmatrix} \sqrt{1-u}\cos(2\pi v) \\ \sqrt{1-u}\sin(2\pi v) \\ \sqrt{1-u}\cos(2\pi w) \\ \sqrt{1-u}\sin(2\pi w) \end{bmatrix} \tag{30}$$

with u, v, w drawn from uniform random distributions.

For each of the random rotations the closest rotation in $\boldsymbol{\mathcal{Q}}^v$ is computed and the number of the closest matches for each index in $\boldsymbol{\mathcal{Q}}^v$ is counted. The results for the first three subdivision levels are depicted in Fig. 18. Both the linear search (5) and the constant time lookup approach (6) provide similarly good results in terms of frequency relative to the expected frequency. The expected frequency is the quotient of random rotations and the number of unique quaternions for the respective discretisation level. The expected frequency is also the reference for the root-mean-square error (RMSE) calculation in Fig. 18.

The presented approaches to subdivide and project the unit-sphere in $\mathbb{R}^3$ for SO(2) and in $\mathbb{R}^4$ for SO(3), show low discretization errors. Furthermore, the equal shape and equal size of the resulting discretised *Voronoi* areas and volumes make it an ideal approach to compare and assess the reachability in $\mathbb{R}^6$ and therefore enables to compute meaningful dexterity metrics for manipulators.

REFERENCES

[1] P. E. Dupont, J. Lock, B. Itkowitz, and E. Butler, "Design and control of concentric-tube robots," *IEEE Transactions on Robotics (T-RO)*, vol. 26, no. 2, pp. 209–225, 2009.

[2] R. J. Webster, A. M. Okamura, and N. J. Cowan, "Toward active cannulas: Miniature snake-like surgical robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2006, pp. 2857–2863.

[3] H. Alfalahi, F. Renda, and C. Stefanini, "Concentric Tube Robots for Minimally Invasive Surgery: Current Applications and Future Opportunities," *IEEE Transactions on Medical Robotics and Bionics (T-MRB)*, pp. 1–16, 2020.

[4] Z. Mitros, S. H. Sadati, R. Henry, L. Da Cruz, and C. Bergeles, "From theoretical work to clinical translation: Progress in concentric tube robots," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, 2021.

[5] J. B. Gafford, S. Webster, N. Dillon, E. Blum, R. Hendrick, F. Maldonado, E. A. Gillaspie, O. B. Rickman, S. D. Herrell, and R. J. Webster, "A concentric tube robot system for rigid bronchoscopy: A feasibility study on central airway obstruction removal," *Annals of Biomedical Engineering*, Jul 2019.

[6] C. Bergeles, A. H. Gosline, N. V. Vasilyev, P. J. Codd, P. J. del Nido, and P. E. Dupont, "Concentric tube robot design and optimization based on task and anatomical constraints," *IEEE Transactions on Robotics (T-RO)*, vol. 31, no. 1, pp. 67–84, Feb 2015.

[7] T. K. Morimoto, J. D. Greer, E. W. Hawkes, M. H. Hsieh, and A. M. Okamura, "Toward the design of personalized continuum surgical robots," *Annals of Biomedical Engineering*, vol. 46, no. 10, pp. 1522–1533, 2018.

[8] R. J. Hendrick, C. R. Mitchell, S. D. Herrell, and R. J. Webster III, "Hand-held transendoscopic robotic manipulators: A transurethral laser prostate surgery case study," *The International Journal of Robotics Research (IJRR)*, vol. 34, no. 13, pp. 1559–1572, 2015.

[9] A. Gosline, N. V. Vasilyev, E. Butler, C. Folk, A. Cohen, R. Chen, N. Lang, P. J. del Nido, and P. E. Dupont, "Percutaneous intracardiac beating-heart surgery using metal MEMS tissue approximation tools," *The International Journal of Robotics Research (IJRR)*, vol. 31, no. 9, pp. 1081–1093, 2012.

[10] C. Girerd, K. Rabenorosoa, P. Rougeot, and P. Renaud, "Towards optical biopsy of olfactory cells using concentric tube robots with follow-the-leader deployment," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 5661–5887.

[11] M. U. Farooq, B. Xu, and S. Y. Ko, "A concentric tube-based 4-DOF puncturing needle with a novel miniaturized actuation system for vitrectomy," *Biomedical Engineering Online*, vol. 18, no. 1, pp. 1–16, 2019.

[12] F.-Y. Lin, C. Bergeles, and G.-Z. Yang, "Biometry-based concentric tubes robot for vitreoretinal surgery," in *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Aug 2015, pp. 5280–5284.

[13] W. Wei and N. Simaan, "Modeling, Force Sensing, and Control of Flexible Cannulas for Microstent Delivery," *Journal of Dynamic Systems, Measurement, and Control*, vol. 134, no. 4, 04 2012, 041004.

[14] N. V. Vasilyev, A. H. Gosline, E. Butler, N. Lang, P. J. Codd, H. Yamauchi, E. N. Feins, C. R. Folk, A. L. Cohen, R. Chen *et al.*, "Percutaneous steerable robotic tool delivery platform and metal mems device for tissue manipulation and approximation: Closure of patent foramen ovale in an animal model," *Circulation. Cardiovascular interventions*, vol. 6, no. 4, 2013.

[15] G. Fagogenis, M. Mencattelli, Z. Machaidze, B. Rosa, K. Price, F. Wu, V. Weixler, M. Saeed, J. E. Mayer, and P. E. Dupont, "Autonomous robotic intracardiac catheter navigation using haptic vision," *Science Robotics*, vol. 4, no. 29, 2019.

[16] J. Burgner, H. B. Gilbert, and R. J. Webster, "On the computational design of concentric tube robots: Incorporating volume-based objectives,"

in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2013, pp. 1193–1198.

[17] X. Yang, S. Song, L. Liu, T. Yan, and M. Q.-H. Meng, "Design and optimization of concentric tube robots based on surgical tasks, anatomical constraints and follow-the-leader deployment," *IEEE Access*, vol. 7, pp. 173 612–173 625, 2019.

[18] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," *IEEE Transactions on Robotics (T-RO)*, vol. 31, no. 6, pp. 1261–1280, Dec 2015.

[19] K. Leibrandt, P. Wisanuvej, G. Gras, J. Shang, C. A. Seneci, P. Giataganas, V. Vitiello, A. Darzi, and G. Yang, "Effective manipulation in confined spaces of highly articulated robotic instruments for single access surgery," *IEEE Robotics and Automation Letters (RA-L)*, vol. 2, no. 3, pp. 1704–1711, July 2017.

[20] N. Vahrenkamp and T. Asfour, "Representing the Robot's Workspace through Constrained Manipulability Analysis," *Autonomous Robots*, vol. 38, no. 1, pp. 17–30, 2015.

[21] M. J. Tsai and Y. H. Chiou, "Manipulability of manipulators," *Mechanism and Machine Theory*, vol. 25, no. 5, pp. 575 – 585, 1990.

[22] J.-P. Merlet, "Optimal Design of Robots," in *Robotics: Science and systems*, 2005.

[23] K. Leibrandt, C. Bergeles, and G.-Z. Yang, "Implicit active constraints for concentric tube robots based on analysis of the safe and dexterous workspace," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 193–200.

[24] L. Stocco, S. Salcudean, and F. Sassani, "On the use of scaling matrices for task-specific robot design," *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 15, no. 5, pp. 958–965, 1999.

[25] A. Nahvi and J. M. Hollerbach, "The noise amplification index for optimal pose selection in robot calibration," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1. IEEE, 1996, pp. 647–654.

[26] J.-T. Lin, C. Girerd, J. Yan, J. T. Hwang, and T. K. Morimoto, "A generalized framework for concentric tube robot design using gradient-based optimization," *IEEE Transactions on Robotics (T-RO)*, pp. 1–18, 2022.

[27] C. Baykal, L. G. Torres, and R. Alterovitz, "Optimizing design parameters for sets of concentric tube robots using sampling-based motion planning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 4381–4387.

[28] C. Baykal, C. Bowen, and R. Alterovitz, "Asymptotically optimal kinematic design of robots using motion planning," *Autonomous Robots*, vol. 43, no. 2, pp. 345–357, 2019.

[29] A. Kuntz, C. Bowen, C. Baykal, A. W. Mahoney, P. L. Anderson, F. Maldonado, R. J. Webster, and R. Alterovitz, "Kinematic design optimization of a parallel surgical robot to maximize anatomical visibility via motion planning," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 926–933.

[30] J.-Y. Park, P.-H. Chang, and J.-Y. Yang, "Task-oriented design of robot kinematics using the grid method," *Advanced Robotics*, vol. 17, no. 9, pp. 879–907, 2003.

[31] D. Salle, P. Bidaud, and G. Morel, "Optimal design of high dexterity modular mis instrument for coronary artery bypass grafting," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, April 2004, pp. 1276–1281 Vol.2.

[32] O. Chocron, "Evolutionary design of modular robotic arms," *Robotica*, vol. 26, no. 3, p. 323–330, 2008.

[33] S. Niyaz, A. Kuntz, O. Salzman, R. Alterovitz, and S. S. Srinivasa, "Optimizing motion-planning problem setup via bounded evaluation with application to following surgical trajectories," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[34] M. T. Chikhaoui, J. Granna, J. Starke, and J. Burgner-Kahrs, "Toward motion coordination control and design optimization for dual-arm concentric tube continuum robots," *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, no. 3, pp. 1793–1800, 2018.

[35] B. An, W. Lee, S. Kang, and C. Kim, "Design parameter optimization of a novel serial manipulator for microsurgery," in *14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, June 2017, pp. 146–151.

[36] J. Ha, F. C. Park, and P. E. Dupont, "Achieving elastic stability of concentric tube robots through optimization of tube precurvature," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2014, pp. 864–870.

[37] J. Kuffner, "Effective sampling and distance metrics for 3d rigid body path planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 4, 2004, pp. 3993–3998 Vol.4.

[38] X. Perez-Sala, L. Igual, S. Escalera, and C. Angulo, "Uniform sampling of rotations for discrete and continuous learning of 2d shape models," in *Robotic Vision: Technologies for Machine Learning and Vision Applications*. IGI Global, 2013, pp. 23–42.

[39] J. Arvo, "Fast random rotation matrices," in *Graphics Gems III (IBM version)*. Elsevier, 1992, pp. 117–120.

[40] A. Yershova, S. Jain, S. M. Lavalle, and J. C. Mitchell, "Generating uniform incremental grids on so(3) using the hopf fibration," *The International Journal of Robotics Research (IJRR)*, vol. 29, no. 7, pp. 801–812, 2010.

[41] K. M. Gorski, E. Hivon, A. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke, and M. Bartelmann, "Healpix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere," *The Astrophysical Journal*, vol. 622, no. 2, p. 759, 2005.

[42] J. Straub, T. Campbell, J. P. How, and J. W. Fisher, "Efficient global point cloud alignment using bayesian nonparametric mixtures," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 2403–2412.

[43] K. Kuo, "Mackay, anti-mackay, double-mackay, pseudo-mackay, and related icosahedral shell clusters," *Structural chemistry*, vol. 13, no. 3, pp. 221–230, 2002.

[44] H. S. M. Coxeter, "Introduction to Geometry, 2nd ed." p. 404, 1969.

[45] A. Liu and B. Joe, "Quality local refinement of tetrahedral meshes based on 8-subtetrahedron subdivision," *Mathematics of Computation*, vol. 65, no. 215, pp. 1183–1200, 1996.

[46] K. Leibrandt, "Cooperative manipulation of highly articulated and continuum surgical robots in confined spaces," Ph.D. dissertation, Imperial College London, 2017.

[47] C. M. Bishop, *Pattern Recognition and Machine Learning*, ser. Information Science and Statistics. Springer: New York, 2006.

[48] A. Schmitz, S. Treratanakulchai, P. Berthet-Rayne, and G. Yang, "A rolling-tip flexible instrument for minimally invasive surgery," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 379–385.

[49] L. da Cruz, K. Fynes, O. Georgiadis, J. Kerby, Y. H. Luo, A. Ahmado, A. Vernon, J. T. Daniels, B. Nommiste, S. M. Hasan, S. B. Gooljar, A.-J. F. Carr, A. Vugler, C. M. Ramsden, M. Bictash, M. Fenster, J. Steer, T. Harbinson, A. Wilbrey, A. Tufail, G. Feng, M. Whitlock, A. G. Robson, G. E. Holder, M. S. Sagoo, P. T. Loudon, P. Whiting, and P. J. Coffey, "Phase 1 clinical study of an embryonic stem cell-derived retinal pigment epithelium patch in age-related macular degeneration." *Nature Biotechnology*, vol. 36, pp. 328–337, Apr. 2018.

[50] E. Vander Poorten, C. N. Riviere, J. J. Abbott, C. Bergeles, M. A. Nasseri, J. U. Kang, R. Sznitman, K. Faridpooya, and I. Iordachita, "Robotic retinal surgery," in *Handbook of Robotic and Image-Guided Surgery*. Elsevier, 2020, pp. 627–672.

[51] K. Leibrandt, C. Bergeles, and G.-Z. Yang, "On-line Collision-free Inverse Kinematics with Frictional Active Constraints for Effective Control of Unstable Concentric Tube Robots," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3797–3804, 2015.

[52] S. G. Johnson, S. Kucherenko, Y. Sytsko, P. Kaelo, M. M. Ali, W. L. Price, R. Brent, J. A. Nelder, R. Mead, and M. J. D. Powell, "The nlopt nonlinear-optimization package," 2014. [Online]. Available: http://ab-initio.mit.edu/nlopt

[53] G. Marsaglia *et al.*, "Choosing a point from the surface of a sphere," *The Annals of Mathematical Statistics*, vol. 43, no. 2, pp. 645–646, 1972.

[54] K. Shoemake, "Uniform random rotations," in *Graphics Gems III (IBM Version)*. Elsevier, 1992, pp. 124 – 132.

Konrad Leibrandt received the Dipl.-Ing. (M.Sc. equivalent) degree in electrical engineering from Technical University of Munich (TUM), Munich, Germany, in 2012, as well as the M.Res. degree in medical robotics and image guided intervention, and the Ph.D. degree in robotic surgery with the department of computing from Imperial College London (ICL), London, UK, in 2013 and 2018 respectively. He is now contributing to the development of medical robotic systems that received 510(k) clearance from the U.S. Food and Drug Administration (FDA), and he is leading engineering teams in the medical device industry to research algorithms and develop software for novel medical robotic applications. Dr Leibrandt's work in robotic bronchoscopy has lead to successful diagnosis of lung nodules in patients and his contributions in a robotic urology procedure enables minimally invasive treatment of kidney stones in humans.

Lyndon Da Cruz completed his Ph.D. in Retinal Gene therapy at the University of Western Australia on an NHMRC medical Scholarship, his post-doc fellowship on the Royal Society Howard Florey fellowship, and was named the Menzies scholar for Australia in 2000. He is now Consultant Retinal Surgeon and Head of Department of Vitreo-retinal surgery at Moorfields Eye Hospital; Professor of Retinal and Stem Cell Transplantation, UCL; Hon. Professor of Biomedical Engineering at King's College London. He has worked in innovative research for the treatment of blinding retinal conditions since 1990, including implanting the first prosthetic retina (bionic eye) in Britain (2008) and the first-in-man delivery of a stem cell derived retinal pigment epithelial sheet for macular degeneration that he has co-developed (2015). For his research he has been awarded The Ruskell Medal by the Worshipful company of Spectacle Makers (2018); The Alan Alderman award from the Macular disease Society (2018); The Denuncio medal from The Society of Ophthalmic Professors, Italy (2016); The Harold Ridley Medal, The Ridley Foundation (2014); The Gulstrand Medal from the Swedish National Ophthalmic Society (2010). Prof. da Cruz cofounded *The London Project*, to create new stem cell treatments for blinding retinal conditions. The project realises the need for large scale multidisciplinary teams of stem cell scientists, surgeons and engineers to solve the problems of tissue manufacture, understanding of disease treatment points and reliable delivery of the therapeutic element. He has established a robotic engineering group at King's College London with Dr Christos Bergeles to create robotic delivery systems for novel biological treatments in the retina.

Christos Bergeles (M'11, SM'23) received the M.Sc. degree in electrical and computer engineering from the National Technical University of Athens, Athens, Greece, in 2006, and the Ph.D. degree in mechanical engineering from ETH Zurich, Zurich, Switzerland, in 2011. He was a post-doctoral research fellow at Harvard Medical School (Boston Children's Hospital) and Imperial College London. He is now Associate Professor (Reader) at King's College London, where he directs the Robotics and Vision in Medicine Lab. Dr Bergeles has been awarded an ERC Starting Grant and holds i4i NIHR funding for the development of instrumentation that delivers stem cells to diseased retinal layers.