IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Enhanced Dexterity Maps (EDM): A New Map for Manipulator Capability Analysis

Haowen Yao, Riddhiman Laha, Luis F.C. Figueredo, and Sami Haddadin

***Abstract*— The ability of a manipulator to compute a geometry-aware quality index for general tasks with different joint configurations is essential. Such workspace assessment is a well-known and studied field in existing robotics literature, often deployed through embodied structures such as voxelized maps. Notwithstanding, existing literature solely focuses on the assessment of a single pose (end-effector), neglecting the whole-body structure and its dexterity, which allows for secondary task optimization, nullspace motion, body placement, and improved manipulability. The proposed Enhanced Dexterity Maps (EDM) aims to close these gaps using an augmented data structure. It offers a systematic analysis of disjoint flip solutions and accommodates additional performance metrics. Further analysis of EDMs through case studies highlight the limitations of existing methods and supports the need for a whole-body analysis.**

I. Introduction

As robotics research shifts towards human-centric environments, which are often cluttered and dynamic, an embodied structure representation of the robot's whole-body workspace becomes critical for feasibility and performance assessment. Indeed, understanding and assessing feasible robot configurations can significantly improve decision-making and performance, for instance, through reachability, dexterity, manipulability, or other task-specific map. This can be used for different tasks, such as grasping, path planning, and co-design, as shown in the comprehensive literature [1]–[8].

Nonetheless, the existing literature still mostly disregards whole-body dexterity and reduces all feasible posture configurations into a single encoded task-space point (for instance, configurations $\boldsymbol{q}_{a_i}, \boldsymbol{q}_{b_j}$ into same $\boldsymbol{x}_e$ in Fig. 1). These points or voxels band together feasible configurations with a quality index by a heuristic function like min, max, or average. These functions often fail to capture the number and quality of individual nullspace solutions and neglects the existence of disjoint configurations that cannot be reachable through a continuous nullspace motion, i.e., without disturbing the main task. Take for instance, Fig. 1 as example: Existing maps band together $\boldsymbol{q}_{a_i}$ and $\boldsymbol{q}_{b_j}$ metrics (e.g. averaging the manipulability) even though configurations in $\boldsymbol{q}_{a_i}$ and $\boldsymbol{q}_{b_j}$ are disconnected and cannot be reached without disturbing the task. Neglecting such disconnected spaces is even more critical when planning to reach a feasible pose only available on specific flip spaces, which is rather common, as in Fig 1.

The aforementioned challenges call for a novel embodied delineation explicitly covering the nullspace capabilities and disjoint regions of the robot. To this end, we introduce and formalize the concept of Enhanced Dexterity Maps. EDMs augment the robot's original workspace, incorporating two new task-space dimensions concerning the robot's nullspace along feasible swivel angles [9], [10] and its flip nullspace. We show how this structure can be used to analyze feasible configurations, manipulability, and redundancy resolution similar in a more anthropomorphic manner [11], [12].

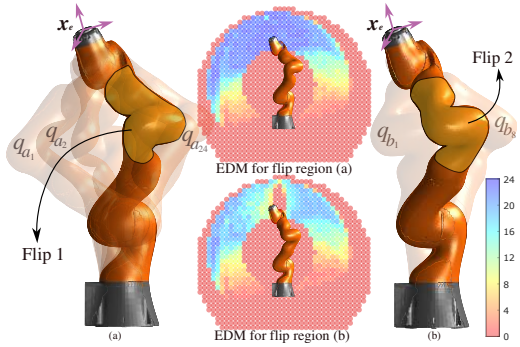

Fig. 1. Enhanced Dexterity Maps (EDMs) allow us to strategically select flips depending on the task and feasible nullspace capabilities. Although there exists no feasible nullspace motion from $\boldsymbol{q}_{a,i}$ to $\boldsymbol{q}_{b,j}$ due to the highlighted flips, our framework is able to analyze the various nullspace solutions in (a) 24 possible solutions and (b) 8 possible solutions for the same EE pose $\boldsymbol{x}_e$. The color spectrum denotes dexterity values from low (0) to high (24).

A. Related Work

Reachability and Dexterity maps capture the workspace quality in terms of feasible end-effector poses [13]. One of the first ones to propose a measure of manipulability was Yoshikawa [14]. The authors in [15] extended it to the global context, thereby introducing the global conditioning index. The benefits of such analyses include workspace quality assessment, robot-base placement [4], [16], and task-oriented optimal pose selection [17], robotic machining of complex parts [18], and motion optimization of surgical robots [19].

A mathematically consistent definition of the reachable workspace is introduced in [20], which neglects singular points. This was further enhanced in [21], where the authors developed an offline computation of the capability map. In addition to feasibility, the inherent data structure of reachability and dexterity maps allow for a real-valued metric that implicitly captures robot's manipulation capabilities [22]. These performance metrics can be task-specific and/or agnostic to them, as is the case of manipulability-informed reachability maps, also known as manipulability maps [6]. Informed maps have also been used for bimanual manipulation skills [7] and even to describe human manipulation capabilities in terms of a quality index for biomechanics features [23].

These maps can additionally capture the proximity to singularities and/or joint limits in the form of a quality index per voxel in a discretized $\mathrm{SE}(3)$ space. In [8], [24], the authors include penalties for self-collisions, joint-limits, collision avoidance, and task objectives for a grasp planning

TABLE I

DESCRIPTION OF MAIN NOTATIONS USED IN THE PAPER

Notation	Corresponding Description
$\boldsymbol{q} \in \boldsymbol{Q}^{\mathcal{CN}}$	$\triangleq$ Configuration in continuous nullspace band
$\boldsymbol{x} \in \mathcal{X}_{\text{EE}}$	$\triangleq$ EE pose in the set of EE poses
$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{q})$	$\triangleq$ Forward kinematic function (FKM)
$\boldsymbol{J_q}$	$\triangleq$ Jacobian corresponding to configuration $\boldsymbol{q}$
$\mathcal{N}(\boldsymbol{J_q})$	$\triangleq$ Nullspace of the Jacobian
$\mathcal{CN}$	$\triangleq$ Continuous Nullspace
$\mathcal{FN}$	$\triangleq$ Flip Nullspace
ρ	$\triangleq$ Index for end effector position
γ	$\triangleq$ Index for orientation direction
ϕ	$\triangleq$ Index for rotation about z-axis of EE
η	$\triangleq$ Different configurations within same CNB
ζ	$\triangleq$ Index for flip region of current CNB
$\mathcal{Q}_{\text{EDM}}$	$\triangleq$ EDM data structure for configurations
$\mathcal{S}_{\text{EDM}}$	$\triangleq$ EDM data structure for solution status

scenario with a humanoid robot. Differently, a method inspired by biology is proposed in [25], where the authors try to learn the workspace reachability map using a neural network architecture. Similarly, an inverse reachability map for base placement analysis of fixed base and mobile robots is presented in [26]. However, as highlighted, almost all of the maps do not associate continuous joint configurations to a discretized task space voxel in order to move to neighboring poses. To the best of the authors' knowledge, data structures in existing literature carrying either binary status or capability information like manipulability, distance to singularities etc. are only based on a 3D/6D workspace voxelized data structure. As a result of neglecting critical posture information, they failed to capture the number and quality of individual nullspace solutions and disjoint flip solutions, as depicted in Fig. 1.

B. Statement of Contributions

In this work, in contrast, we propose an augmented data structure that also incorporates the distribution of feasible swivel angles of a serial manipulator and its discontinuous regions. Our main contributions are:

1) The definition and systematic analysis of the kinematic topology through nullspace exploration;
2) An algorithmic framework that generates unified dexterity maps based on the measures of the continuous nullspace and flip nullspace;
3) Demonstration that our developed concept is general enough to accommodate additional metrics for performance like joint limits and manipulability. Existing maps are shown to be a special case of the introduced EDM;
4) An open-source toolbox for the community that displays the entire information as a panel.[1]

II. ENHANCED DEXTERITY ANALYSIS

This section introduces concepts concerning the robot nullspace at a given end-effector (EE) pose. These concepts shape the backbone of the enhanced dexterity maps and the proposed automatic algorithms. Further open discussions related to the same are addressed in Section VI.

A. Continuous Nullspaces

Anthropomorphic features in 7-DoFs systems led to an additional degree of redundancy [27], which is often explored for nullspace motion, the joint motions within the nullspace of the task Jacobian [28], also known as self-motion.

Given a generalized workspace coordinate $\boldsymbol{x} \in \text{SE}(3)$ containing positions and orientation coordinates (6 degrees of freedom in total), and a joint configuration $\boldsymbol{q} \in \mathbb{R}^n$, the forward kinematic function can be expressed as $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{q})$. Now, the kinematic redundancy ($n > 6$) can be resolved using the definitions below.

Definition 1. Continuous Nullspace ($\mathcal{CN}$): *Given joint configurations $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$, they are said to be in the same continuous nullspace if and only if they (i) lead to the same pose (ii) connected by nullspace motion (iii) being sufficiently close in joint-space. Formally, $\boldsymbol{q}_2$ is reachable from $\boldsymbol{q}_1$ with arbitrarily small changes in norm ($\varepsilon \in \mathbb{R}_{>0}$) with first-order difference equation being mapped to the nullspace of the Jacobian of $\boldsymbol{q}_1$, that is,*

$$\mathcal{CN}(\boldsymbol{q}_1, \boldsymbol{q}_2) : \forall \varepsilon > 0, \boldsymbol{f}(\boldsymbol{q}_1) = \boldsymbol{f}(\boldsymbol{q}_2) \\ ||\boldsymbol{q}_2 - \boldsymbol{q}_1|| < \varepsilon, \boldsymbol{q}_2 - \boldsymbol{q}_1 \in \mathcal{N}(\boldsymbol{J}_{\boldsymbol{q}_1}) \tag{1}$$

Using an analogy between the robot's and human's arms, researchers proposed the definition of *Swivel Angle* [29]–[31], which indicates the angular difference of the arm plane and the reference plane, as depicted in Fig. 2. This posture-dependent strategy provides a standard approach to classify different configurations during a nullspace motion and is adopted for our EDM encoding.

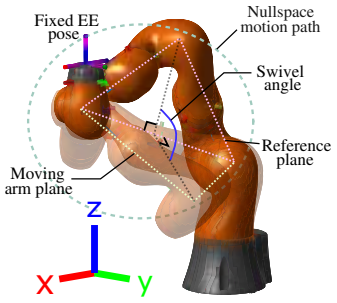

Fig. 2. Swivel angle definition for iiwa 14. The transparent robot depicts one instance of the nullspace motion.

B. Continuous Nullspace Band and Flip Nullspaces

After considering definition 1, the natural idea is to group together all $\mathcal{CN}$ configurations achieved within a contiguous nullspace motion into a set. This individual set is herein defined as a single *Continuous Nullspace Band* (CNB).

Definition 2. Continuous Nullspace Band: *The continuous nullspace bands $\boldsymbol{Q}^{\mathcal{CN}}$ are a subset of configurations with same EE pose $\boldsymbol{x}^*$. For any two configurations $\hat{\boldsymbol{q}}$, $\tilde{\boldsymbol{q}}$ in this set, there is a sequence $(\boldsymbol{q}_i)$ starting from $\hat{\boldsymbol{q}}$, such that it converges to $\tilde{\boldsymbol{q}}$ through infinitesimal motions, and each term $\boldsymbol{q}_{i+1}$ is in continuous nullspace with the previous term $\boldsymbol{q}_i$.*

$$\forall \hat{\boldsymbol{q}}, \tilde{\boldsymbol{q}} \in \boldsymbol{Q}^{\mathcal{CN}}, \exists\, (\boldsymbol{q}_i) \subset \boldsymbol{Q}^{\mathcal{CN}} : \\ \boldsymbol{x}^* = \boldsymbol{f}(\hat{\boldsymbol{q}}) = \boldsymbol{f}(\tilde{\boldsymbol{q}}),\ \boldsymbol{q}_1 = \hat{\boldsymbol{q}},\ \lim_{i \to \infty} \boldsymbol{q}_i = \tilde{\boldsymbol{q}},\ \mathcal{CN}(\boldsymbol{q}_i, \boldsymbol{q}_{i+1}) \tag{2}$$

To build the CNB, we perform a virtual nullspace motion doing a full revolution of the swivel angle (see Fig. 2). During this procedure, self-collisions and joint limits are temporarily ignored but marked. Notice that the swivel angle and its connection to the configuration in CNB is crucial. It provides an intuitive way to connect the general robot workspace posture and the CNB in the configuration space for high DoF robot arms. As shown in Fig. 3, this mapping has a bijective nature (at least in the numerical sense) in a 7

[1]Link for the toolbox: https://gitlab.lrz.de/edm-project/edm-prototype

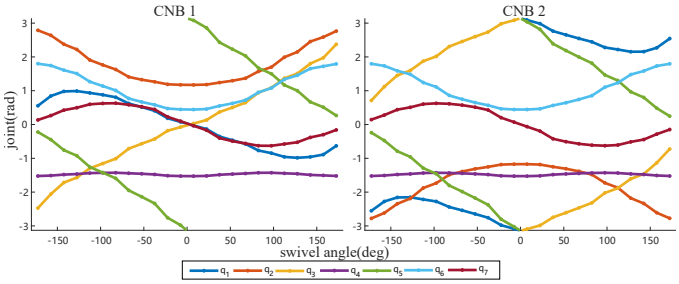

Fig. 3. Example of configurations of CNBs changing along swivel angle. (Franka-Emika for EE position [0.65 0 0.175] and orientation [1 0 0 0].) We can notice the bijectivity between configuration and swivel angle in CNB. Also, comparing two CNBs, some joints are shifted ($\boldsymbol{q}_1, \boldsymbol{q}_3$) or flipped ($\boldsymbol{q}_2$).

DoF robot.

However, also from Fig. 3, we notice that multiple CNBs that are distant from each other can map to the same EE pose. These distinct CNBs $Q_1^{\mathcal{CN}}, Q_2^{\mathcal{CN}}, \cdots$ are mostly characterized by flipping, disjoint shift or mirrored relationship in the configuration space. Therefore, we call this kind of relationship *Flip Nullspace*.

Definition 3. Flip Nullspace ($\mathcal{FN}$): *For multiple distinct CNBs* $\{\boldsymbol{Q}_i{}^{\mathcal{CN}}, \cdots\}$ *not connected by continuous nullspace and projecting into the same EE pose* $\boldsymbol{x}^*$ *with the kinematic function* $\boldsymbol{f}$*, each CNB is the Flip Nullspace* $\mathcal{FN}$ *of other CNBs.*

$$\left\{\boldsymbol{Q}_1{}^{\mathcal{CN}}, \cdots, \boldsymbol{Q}_n{}^{\mathcal{CN}}\right\} \xrightarrow{\boldsymbol{f}} \boldsymbol{x}^* \tag{3}$$

$$\boldsymbol{Q}_i{}^{\mathcal{CN}} \xleftrightarrow{\mathcal{FN}} \boldsymbol{Q}_j{}^{\mathcal{CN}}, i \neq j$$

For all feasible end-effector poses in the workspace, the collection of each CNB forms the corresponding *flip region*.

As a simple example for both nullspaces, consider a planar three-link robot shown in Fig. 4. The robot has 1 extra DoF, therefore, each configuration has $\mathcal{CN}$ for the robot arm to explore. Furthermore, the robot can obtain 2 different solutions by flipping the first and second link frames up and down. Thus, they are $\mathcal{FN}$ of each other.

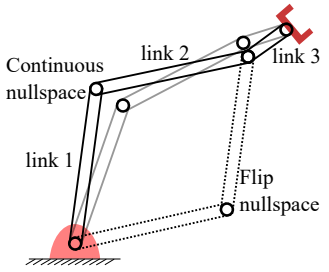

Fig. 4. Continuous and flip nullspace shown for a planar 3-link robot. Note that the task relates to position only.

We can gather deeper insights from this example in the workspace and configuration space as illustrated in Fig. 5. Two CNBs marked in configuration space are all projected in the same EE pose through kinematic function. Configurations $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ belongs to the same CNB $\boldsymbol{Q}_1^{\mathcal{CN}}$ in same flip region, so that they can reach each other with continuous nullspace motion. $\boldsymbol{q}_3$ resides in a different flip and thus belongs to a distinct CNB $\boldsymbol{Q}_2^{\mathcal{CN}}$.

C. Distance Metrics

Distance is an important metric for clustering. The distance metrics that we employ for sorting CNBs in Alg. 2 are:

- *Joint2CNB distance*: distance between a joint configuration $\boldsymbol{q}$ and a CNB $\boldsymbol{Q}^{\mathcal{CN}}$. It is defined by the minimum of the H_∞ norm between the configuration and every possible configuration in CNB, that is,
$$\text{dist}(\boldsymbol{q}, \boldsymbol{Q}^{\mathcal{CN}}) = \min_i(||\boldsymbol{q} - \boldsymbol{q}_i||_\infty), \ \boldsymbol{q}_i \in \boldsymbol{Q}^{\mathcal{CN}}$$
- *CNB2CNB distance*: is the distance between two CNBs $\boldsymbol{Q}_1^{\mathcal{CN}}, \boldsymbol{Q}_2^{\mathcal{CN}}$. It is defined by the minimum of Joint2CNB distance between each configuration of one CNB to another CNB:
$$\text{dist}(\boldsymbol{Q}_1^{\mathcal{CN}}, \boldsymbol{Q}_2^{\mathcal{CN}}) = \min_i(\text{dist}(\boldsymbol{q}_i, \boldsymbol{Q}_2^{\mathcal{CN}}), \boldsymbol{q}_i \in \boldsymbol{Q}_1^{\mathcal{CN}}$$

Though the CNB2CNB distance definition here is sequential, we are looking for the closest distance between two CNBs, therefore the distance is symmetric.

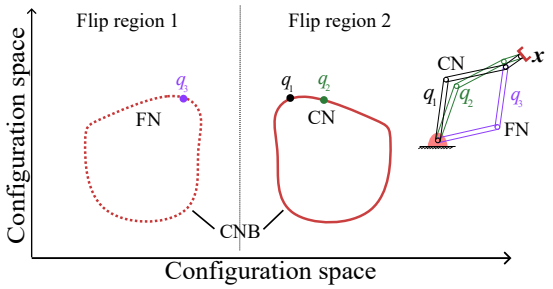

Fig. 5. Schematic of two continuous nullspace bands for the same EE pose to demonstrate continuous nullspace (different configuration in an CNB) and flip nullspace (different CNBs). Notice that EE orientation is neglected.

D. Data Structure of Targeted EDM

Here we detail the data structure to build the EDM. The map contains two sets of data structures controlled by five separate indices: (I) solution status $\mathcal{S}_{\text{EDM}} = \{s(\rho, \gamma, \phi, \eta, \zeta)\}$ and (II) configuration $\mathcal{Q}_{\text{EDM}} = \{\boldsymbol{q}(\rho, \gamma, \phi, \eta, \zeta)\}$.

Indices ρ, γ, and ϕ are responsible for the voxel discretization of the EE pose in the workspace $\mathbb{R}^3 \times S^2 \times S^1$, as shown in Fig. 6. Precisely, ρ depicts the EE position, γ is the pointing orientation of EE, and ϕ is the rotation around the center axis of EE. In accordance with Def. 2, index η depicts different configurations within the same CNB, reflecting the section of the swivel angle along a full rotation distribution. Index ζ is the flip region the current CNB resides in.

Each element of the configuration $\mathcal{Q}_{\text{EDM}}$ stores a configuration $\boldsymbol{q}$ that matches the definition of the given indexes if it exists. Each element of solution status $\mathcal{S}_{\text{EDM}}$ records one of the possible status for different indices $s \in \{\text{hasSolution}, \text{noSolution}, \text{jointLimit}, \text{selfCollision}\}$ for the corresponding configuration in $\mathcal{Q}_{\text{EDM}}$. In application, it is used as a binary value that equals 1 if it is feasible (hasSolution) and 0 for other cases.

This data structure is designed with extra indices η, ζ that decouples both nullspaces. It provides an adequate structure to represent dexterity. This allows an advanced reasoning over nullspace performances. For instance, feasible swivel configurations, possible swept area during a path planning task, and the best flip region for different tasks.

III. Automatic Generation of EDMs

This section illustrates the building blocks and steps towards the design of the EDM. The proposed approach, similar to concepts of reachability and manipulability [32], builds toward a voxelized data structure of $\mathcal{S}_{\text{EDM}}$ and $\mathcal{Q}_{\text{EDM}}$.

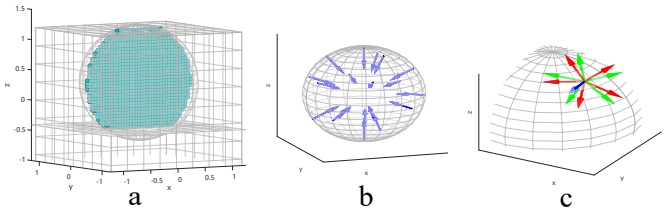

Fig. 6. (a) An x-z slice of discretized positions under basic geometric shapes. (b) The $N_{\text{dir}} = 20$ directions around a reachability sphere for one position voxel. (c) $N_{\text{rot}} = 5$ rotations around the axis in one direction.

A. Discretization of the End-Effector Workspace

The end-effector pose is evenly discretized in N_{pos} positions, N_{dir} Cartesian directions, and N_{rot} rotations, which could be expressed as a set of rigid body transformations given by $\mathcal{X}_{\text{EE}} = \{\boldsymbol{x}(\rho, \gamma, \phi)\}$ where each element is indexed in correspondence to the targeted data structure. Fig. 6 illustrates the procedure as voxelized grids of the SE(3) space.

The position discretization is estimated by the intersection of primitive geometric shapes like spheres or cubes according to manufacturer specifications. The discretization of the orientation direction follows the idea of reachability sphere [21]. The directional vector is distributed on the sphere's surface using the spiral point algorithm [33]. Finally, rotation along the resulting vector is evenly discretized, completing the orientation description.

B. Generate Unsorted Enhanced Dexterity Map (UEDM)

The main intention of the `generateUEDM` function is to sample CNBs for each pose of the discretized EE-workspace and obtain the validity state for each sampled configuration. The function takes the robot information $\boldsymbol{r}$, the EE poses $\mathcal{X}_{\text{EE}}$, and the number of iterations n_{t} as input. The outputs are configurations $\mathcal{Q}_{\text{UEDM}}$ and validity state $\mathcal{S}_{\text{UEDM}}$.

As depicted in Alg. 1, for each end-effector pose $\boldsymbol{x}(\rho, \gamma, \phi)$, we solve the IK problem n_{t} times to obtain potential configurations $\boldsymbol{q}_{\text{ik}}$ multiple times (line 5). We use a numeric IK solver using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) gradient projection algorithm for generality [34].

Each time a new solution $\boldsymbol{q}_{\text{ik}}$ is found, it is first compared with existing CNB for the same EE pose to determine if it is reachable with continuous nullspace motion (line 9). If not, $\boldsymbol{q}_{\text{ik}}$ is then sampled as a new CNB $\boldsymbol{Q}_{\text{ik}}$ with N_{sec} samplings. To make the result evenly represent a full revolution of the swivel angle, each of the N_{sec} configurations should fall into a distinct $2\pi/N_{\text{sec}}$ section for the swivel angle. Afterward, $\boldsymbol{Q}_{\text{ik}}$ receives a flip region index ζ_{ik} to distinguish it from other existing CNBs. In the end, this CNB $\boldsymbol{Q}_{\text{ik}}$ is collected into the UEDM.[2] When the repetition is terminated, the function calculates each configuration's validity state $\mathcal{S}_{\text{UEDM}}$.

C. Sorting UEDMs into incomplete EDMs

The UEDM data structure should use index ζ to describe CNBs in different flip regions. Nonetheless, as shown in line 11 of the Alg. 1, the index ζ in UEDM is incremental for each pose based on the random sequence of the IK solver.

[2]The colon symbol has been used throughout this work to indicate all elements in the corresponding index like Matlab, as seen in line 12 of Alg. 1.

Algorithm 1: Generate UEDM with IK

1 **function** `generateUEDM`($\mathcal{X}_{\text{EE}}$,n_{t},$\boldsymbol{r}$)
2 $\quad\mathcal{Q}_{\text{UEDM}}, \mathcal{S}_{\text{UEDM}} \leftarrow$ empty value matrix;
3 $\quad$**for** 1 *to* n_{t} **do**
4 $\quad\quad$**foreach** $\boldsymbol{x}(\rho, \gamma, \phi)$ *in* $\mathcal{X}_{\text{EE}}$ **do**
5 $\quad\quad\quad\boldsymbol{q}_{\text{ik}}$,flag $\leftarrow$ solve IK($\boldsymbol{r}$,$\boldsymbol{x}(\rho, \gamma, \phi)$);
6 $\quad\quad\quad$**if** flag = *"best available"* **then**
7 $\quad\quad\quad\quad$**continue**;
8 $\quad\quad\quad$**else if** flag = *"success"* **then**
9 $\quad\quad\quad\quad$**if** $\boldsymbol{q}_{\text{ik}}$ *unreachable with current CNBs* **then**
10 $\quad\quad\quad\quad\quad\boldsymbol{Q}_{\text{ik}} \leftarrow$ sample CNB on $\boldsymbol{q}_{\text{ik}}$;
11 $\quad\quad\quad\quad\quad\zeta_{\text{ik}} \leftarrow$ new empty ζ index;
12 $\quad\quad\quad\quad\quad\boldsymbol{q}_{\text{UEDM}}(\rho, \gamma, \phi, :, \zeta_{\text{ik}}) \leftarrow \boldsymbol{Q}_{\text{ik}}$;
13 $\quad\mathcal{S}_{\text{UEDM}} \leftarrow$ validity state of $\mathcal{Q}_{\text{UEDM}}$;
14 $\quad$**return** $\mathcal{Q}_{\text{UEDM}}$,$\mathcal{S}_{\text{UEDM}}$;

Algorithm 2: Generate EDM by sorting UEDM

1 **function** `UEDM2EDM`($\mathcal{Q}_{\text{UEDM}}$,$\mathcal{S}_{\text{UEDM}}$,$\theta$)
2 $\quad\mathcal{Q}_{\text{EDM}}, \mathcal{S}_{\text{EDM}} \leftarrow$ empty value matrix;
3 $\quad$**foreach** γ, ϕ *in* $\mathcal{Q}_{\text{UEDM}}$ **do**
4 $\quad\quad\rho_s \leftarrow$ start position of γ, ϕ;
5 $\quad\quad\mathcal{Q}_{\text{EDM}} \leftarrow$ import $\boldsymbol{q}_{\text{UEDM}}(\rho_s, \gamma, \phi, :, :)$;
6 $\quad\quad\mathcal{G} \leftarrow$ neighbor of ρ_s;
7 $\quad\quad$**for** $\rho_c \leftarrow$ *element of* $\mathcal{G}$ **do**
8 $\quad\quad\quad\mathcal{G} \leftarrow$ append neighbor of i_c that not in $\mathcal{G}$;
9 $\quad\quad\quad\{\boldsymbol{Q}\} \leftarrow$ sort ζ index of $\boldsymbol{q}_{\text{UEDM}}(\rho_c, \gamma, \phi, :, :)$ with threshold θ;
10 $\quad\quad\quad\boldsymbol{q}_{\text{EDM}}(\rho_c, \gamma, \phi, :, :) \leftarrow \{\boldsymbol{Q}\}$;
11 $\quad\mathcal{S}_{\text{EDM}} \leftarrow$ validity state of $\mathcal{Q}_{\text{EDM}}$;
12 $\quad$**return** $\mathcal{Q}_{\text{EDM}}$,$\mathcal{S}_{\text{EDM}}$;

This section presents a solution to the sorting scheme. The `UEDM2EDM` function sorts index ζ into a same flip region sequence for all poses according to the distance metrics introduced in subsection II-C. This then converts UEDM into EDM. As inputs, it takes the robot data structure $\boldsymbol{r}$, a distance threshold θ, and the UEDM data $\mathcal{Q}_{\text{UEDM}}$, $\mathcal{S}_{\text{UEDM}}$.

As shown in Alg. 2, we first initialize empty matrices $\mathcal{Q}_{\text{EDM}}$ and $\mathcal{S}_{\text{EDM}}$ for EDM (line 2). Then, the algorithm executes iterations for each direction γ and rotation ϕ. For each combination, a position ρ_s is selected as the start position of Breadth First Search (BFS) [35] visit and the initial comparison reference. (line 4-6) To implement BFS visit, we pop a current position ρ_c from queue $\mathcal{G}$ while pushing its neighbor and unvisited positions into $\mathcal{G}$. Each CNB of each ρ_c in UEDM is compared with a neighboring reference position in EDM, sorted in a specific sequence, and saved in the EDM (line 9-10). Concerning the comparison and sorting, if a CNB in ρ_c has a low distance within threshold θ with exactly one CNB of the reference position, they are expected to be in the same flip region ζ. This comparison continues until all positions are visited, and all CNBs are sorted. A schematic overview of the sorting algorithm is shown in Fig 7.

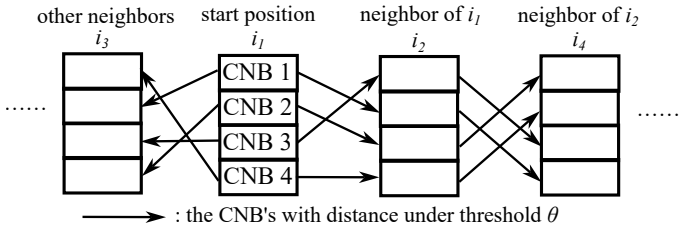

Fig. 7. Schematic of sorting CNBs in a same sequence by comparing distance. The distance between CNBs are compared between each neighbor position from near to far in whole map.

Full EDM Algorithm[3]: The complete algorithm (see Alg. 3) is therefore built from the three previous steps, that is, computing the voxelized structure, the generation of unsorted enhanced dexterity map followed by the sorting algorithm.

Algorithm 3: Complete EDM Algorithm

procedure *generate EDM*($\boldsymbol{r}$,n_t,θ)
  $\mathcal{X}_\mathrm{EE} \leftarrow$ discretize the workspace;
  $\mathcal{Q}_\mathrm{UEDM}, \mathcal{S}_\mathrm{UEDM} \leftarrow$ `generateUEDM`($\mathcal{X}_\mathrm{EE}$,n_t,$\boldsymbol{r}$);
  $\mathcal{Q}_\mathrm{EDM}, \mathcal{S}_\mathrm{EDM} \leftarrow$ `UEDM2EDM`($\mathcal{Q}_\mathrm{UEDM}, \mathcal{S}_\mathrm{UEDM}$,$\theta$);
  return $\mathcal{Q}_\mathrm{EDM}$,$\mathcal{S}_\mathrm{EDM}$;

D. Design Parameters

The EDM algorithm, similar to other informed-based maps, includes discretization parameters to define how fine or coarse are the voxels, which leads to a trade-off between continuous approximation and time and space complexity. Herein, EDM was deployed with N_dir=20, N_rot=5, N_sec=24 and the radius of reachability sphere is 2.5 cm. Parameter n_t in Alg. 1 controls the iteration to search for different flip regions. Herein, an empirical $1.5\times$ the number of flip regions have shown to be a good balance, e.g., we used n_t=6 for Franka and n_t=12 for Kuka. Parameter θ in Alg. 2 controls the sorting threshold. It is selected as $360/N_\mathrm{sec}$=15° to compensate sampling uncertainty for CNB.

IV. Connection with Existing Maps

EDM can naturally derive existing or even undiscovered kinematic or Jacobian-based capability maps. More specifically, by applying trivial operations on the different indices $\rho, \gamma, \phi, \eta, \zeta$ on $\mathcal{S}_\mathrm{EDM}$ $\mathcal{Q}_\mathrm{EDM}$ in a specific sequence, for instance average, median, bounded min-max, NN-based function, logical any, logical all, or even random, any existing capability map can be emulated.

It is worth noting that all existing literature neglects the flip region ζ information, since the systematic CNB sampling is a novel feature in capability maps introduced in this work. In general, most of existing algorithms reduce that information through a heuristic of *calculate-and-update* strategy, which updates a quality index result during iterations for the same point. This method ignores the differences between nullspace solutions and represent them in a single value.

Table II categorizes some existing literature that designs maps to evaluate the workspace performance. We summarize these maps in three large categories based on how many indices are available in their data structure. These maps are further categorized as to whether it provides a binary status or a capability value for each data. Some works generate a map with binary status but then propose a capability metric by collecting information along specific indices. Hence, they appear twice in the table. This categorization highlights the uniqueness of the proposed EDM. It is clear it provides information on all indices with binary status $\mathcal{S}_\mathrm{EDM}$ and also configurations $\mathcal{Q}_\mathrm{EDM}$ for further capability calculation. Here, we also propose a metric for enhanced dexterity based on EDM.

[3]Due to page limit, please visit https://gitlab.lrz.de/edm-project/extra-files for our supplementary material containing further explanation.

TABLE II
Comparison between the different existing maps capturing reachability and dexterity information.

Map data structure	Indexes $\rho, \gamma, \phi, \eta, \zeta$	Binary information	Capability information
Position-based	(✓✗✗✗✗)	–	[16], [21], [26]
Reachability	(✓✓✗✗✗)	[2], [3], [21], [26]	–
Dexterity	(✓✓✓✗✗)	[17]	[4]–[8], [23]
$\mathcal{CN}$-informed	(✓✓✓✓✗)	–	–
EDM	(✓✓✓✓✓)	This paper	This paper

The Enhanced Dexterity Metric: To demonstrate the full information of EDM in the easiest way possible, we propose the *enhanced dexterity metric* $D_{\mathcal{ED}}$ that indicates the number of feasible configuration m while retaining the rest of indices $\rho, \gamma, \phi, \zeta$, that is, the available contiguous swivel angle distribution in a given pose for a given flip region.

$$D_{\mathcal{ED}}(\rho, \gamma, \phi, \zeta) = \sum_{\eta} s(\rho, \gamma, \phi, \eta, \zeta)$$

Figs. 8 and 9 demonstrate two examples of enhanced dexterity metric for a specific direction γ and rotation ϕ with different flips ζ. The exact analysis of the map is described in subsection V-A. Note the max value of the maps is $N_\mathrm{sec} = 24$.

V. Results and Practical Application Scenarios

A. General Map Analysis

We present and analyze the enhanced dexterity metric along a specific direction and rotation with different flips for the Franka Emika research and the KUKA iiwa 14 robots.

Firstly, Fig. 8 depicts the enhanced dexterity metric for Franka Emika's 4 flipped index. Note that flip region indices 3 and 4 lead to poor dexterity performance. In a real application, such flip regions should be avoided. IK solvers and path planners should keep the robot along the flip region index 1 and 2. Particularly, flip region index 2 leads to better dexterity in the proximity of the x-axis[4].

Secondly, Fig. 9 shows the 3D distribution of the $D_{\mathcal{ED}}$ metric for the KUKA iiwa 14. The first assessment is that iiwa has more flip regions ζ compared to the Franka-Emika robot. All 8 solutions with different flip indices have similar reachable edges with small differences occasionally.

Finally, comparing different flips in both figures, we identify some patterns for the kinematic structure stemming from the sliced dexterity map: (a) The outer edge of the reachability area is similar among different flips; (b) The most dexterous part of the workspace always gathers to a side according to the EE direction; (c) Comparing different flips, some have unreachable holes or relatively low dexterity strips in the workspace that could be attributed to joint limits. To conclude, EDM handles enhanced dexterity with two different indices, which helps plan paths in different flips.

[4]It is important to highlight that such a solution is only one slice of EDM for a specific direction and rotation. However, the figure pattern remains the same along different directions and rotations.

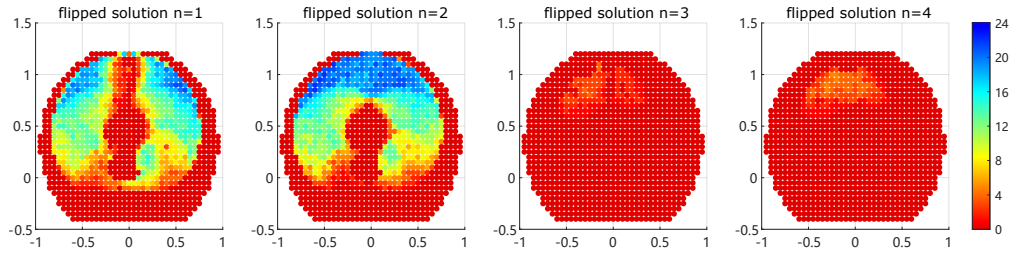

Fig. 8. The x-z plane of sliced dexterity metric for Franka Emika robot for a given orientation (direction index $\gamma = 3$, rotation index $\phi = 1$)

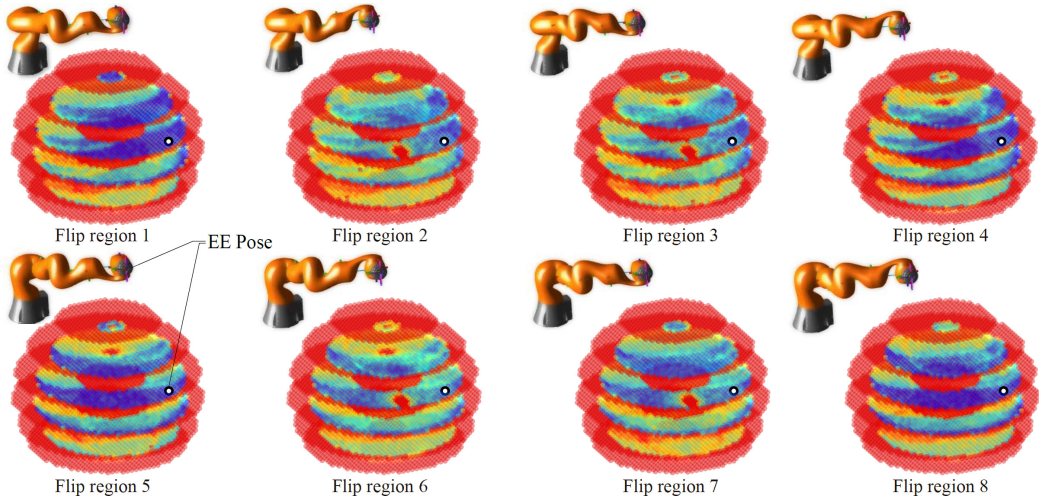

Fig. 9. Sliced dexterity metric of iiwa 14 with orientation $[0.5038, -0.4694, -0.3848, -0.6146]$ (direction $\gamma = 8$, rotation $\phi = 1$). The EE position $[-0.3547, 0.7263, 0.5934]$ is highlighted with the configuration shown in the upper left for each flipped solution. The slices in the x-y plane are taken at intervals of $-0.2, 0.0, 0.3, 0.6, 0.9$, and 1.2m in the z axis.

B. Case Study #1: Ablation study

This experiment demonstrates that EDM can be used for task-relevant start and goal configurations for more robust task-specific motion planning. Specifically, the task is to perform a pick and place motion for a fixed base robot having a static reachable start $\boldsymbol{T}_s$ and end $\boldsymbol{T}_e$. To achieve this, the robot needs to be aligned in a pre-grasping configuration $\boldsymbol{q}_s$ and comprehend the place configuration $\boldsymbol{q}_e$. This task was designed to fully demonstrate planners exploiting nullspace performance during the path to avoid complexities like hitting joint limits, getting into singular configurations, and passing through regions with low manipulability. A bi-directional RRT is used to connect the start and goal [36].

This planning process was compared based on (a) **manip** – maps similar to existing state-of-the-art manipulability map [6] that assigns a score to each EE pose, and acts as a baseline, (b) **manip+** – a new map based on the manipulability map that scores different continuous nullspace configurations for the same EE pose with an average of all flips, and (c) full **EDM-map** with an enhanced dexterity metric.[5] The selection metric is a weighted linear combination of the minimum singular value and the minimum distance of joints to its limit, which contains similar information of Jacobian penalization in [6].

We implemented multiple experiments for the two robots under different start and end poses. Multiple criteria are used for evaluation to ensure comparable performance between different planning methods. Table III shows the planning time. Fig. 10 showcase the planned path distance, joint limits distance, and manipulability in one planning mission. The following are the inferences: (i) The time spent on RRT planning is smaller and more stable with EDM. (ii) The distance in configuration space and Cartesian space is smaller using EDM-based planning. (iii) The distance from joint limits is significantly larger. (iv) The minimum singular value is larger on average, and also the data is more stable.

[5]The maps manip and manip+ can be derived as a special case of the EDM.

TABLE III

MEAN AND STANDARD DEVIATION OF TIME CONSUMED FOR THE RRT PLANNER WITH DIFFERENT METHODS. *Unit:* SECOND(S).

20×5 attempts	manip	manip+	EDM
Franka Emika	1.0296±0.3319	0.8542±0.3580	0.5294±0.2611
KUKA iiwa 14	3.2956±0.8219	3.3165±0.8580	2.3660±0.6072

Since the planning task here considers stationary start and end poses, the manip map fails to provide any information corresponding to the re-adjustment of the joints. The manip+ map encodes continuous nullspace configurations and thus yields improved choices. However, these maps cannot straightforwardly render flip region information, resulting in the robot traversing through different flips. This induces relatively unsatisfactory performance. Employment of EDM allows us to strategically select better configurations in the same flipped solution, therefore, has the best and most reliable performance.

C. Case Study #2: Grasp planning

A grasp planning scenario is another application of EDM that can be used to strengthen our claims. For simplicity, we consider planning in a 2D snapshot in slice of $z = 0.6m$. As Fig. 11 shows, suppose we want to move a object (grey hexagon in the middle) from one point (center of green

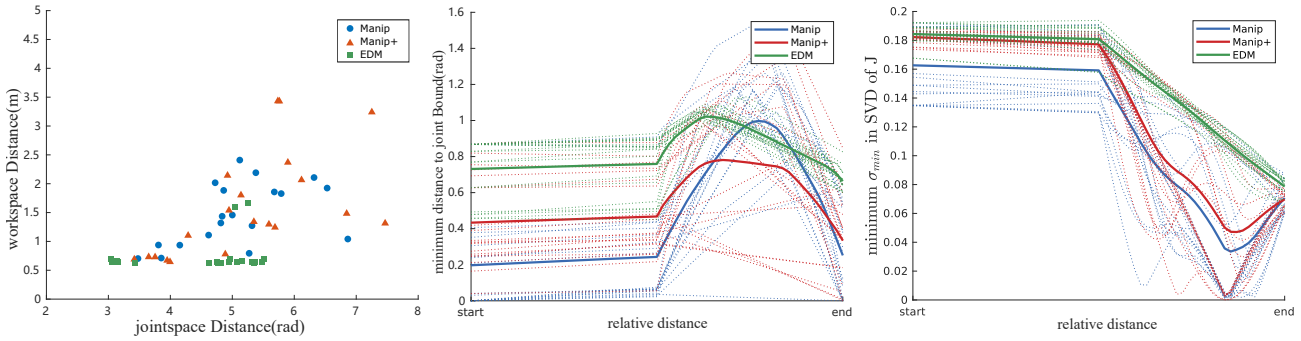

Fig. 10. Performance comparison between planning methods with KUKA iiwa 14. (a) path lengths (b) distance from joint limits along path (c) minimum singular value along path. Solid lines indicate average values, dashed lines denote different trials.

circle) in the workspace to another (center of beige circle). By employing EDM, we compute the average dexterity value of the surrounding regions pointing into the grey hexagon along the arrow as feasible grasp direction. The result is compared between different flip regions. In this specific case, we consider the flip regions with index 1,2,5,6. It is straightforward to observe that the outer side direction is difficult to grasp in all flip regions because it is close to the reachability edge. However, grasping the inner side of the prism has different dexterity performance in different flip regions. Flip region with index 2,5,6 have relatively poor performance in either start, end, or both positions, while only flip region 1 could grasp the prism from more directions with higher dexterity. These performance differences, as seen in Fig. 9, arise from low dexterity holes or stripes due to joint limits. Planning with EDM allows the best flip region selection to exploit the grasping dexterity and thereby avoid a long path that crosses between different flip regions.

D. Case Study #3: Base Placement

Lastly, we also perform a quantitative base placement analyses comprising of 100 different runs similar to [8] for evaluation of the robustness capabilities of our approach. In terms of joint-space distance, we obtain an average improvement of $\approx 30\%$. This is expected since we ensure same flip region. For manipulability and minimum distance to joint limits we secure enhancements of 33.65% and 55.29% respectively. The reason for this is that existing maps average CNB and flip region performances which we instead explicitly evaluate. This exemplifies the benefits of our full EDM map as compared to non-flip-informed ones.

VI. Open Discussions

This section details some interesting phenomena that is connected to other studies and will be considered in the future.

Firstly, our work combines and generalizes multiple concepts presented in separate papers on robot capability. For instance, the idea of CNB corresponds to the term self-motion manifold, whose topology is a ring for 7DoF arm and a torus for 8DoF arm. Similarly, the flip region corresponds to the idea of c-bundle [37]. Expressing the concept from a topological perspective contributes to the discussion but limits the applicability of the algorithm to known and limited cases. Our research uses less prior information to build large-scale maps containing topology information for various robots, while current relevant studies are either small-scale

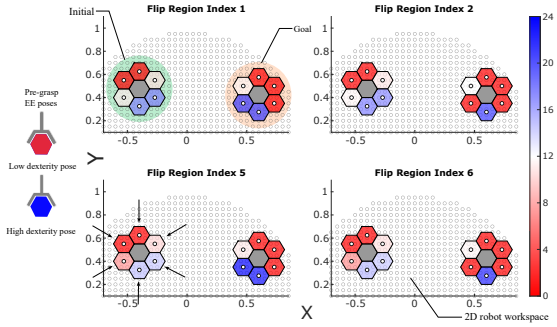

Fig. 11. Grasp-planning application example: Here, the intention is to apply EDM for robust grasp planning for standing prism (grey) at pick (light green) and release (light beige) locations using iiwa 14. Only best 4 flipped regions are considered here. The arrows depict different pre-grasp poses. The colormap spectrum represents a transition from low (red) to high (blue) dexterity performances.

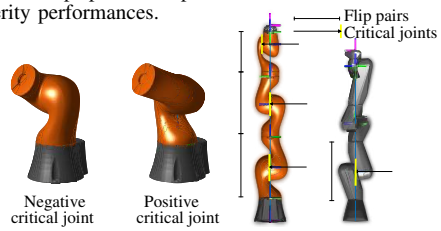

Fig. 12. Left: example of positive and negative critical joint with nearly same end pose. right: Analysis on critical joint for KUKA iiwa 14 and Franka Emika research robot.

[38], or robot-specific and require too much human decision [37], or do not focus on and cannot be trivially extended to the map for global topology [39], [40]. In future work, a promising direction would be to continue to combine the results from these different fields.

Secondly, we want to highlight the correlation between kinematic structure and flip region. From a geometric perspective, a class of robot kinematics is noteworthy : *(2n+1)R Circular Manipulators* [31]. These manipulators have a spherical shoulder, multiple no-offset revolute elbow, and a spherical wrist. In other words, its DH table is zero in orthogonal distances and even-numbered offsets. These manipulators always produce flip nullspace by flipping a pair of joints. As shown in Fig. 12, in either of the two states, the even-numbered joint of a flip pair has nearly identical absolute value but a different sign. As a result, the sign of even joints are critical to identify different flip regions for (2n+1)R circular manipulators.

In contrast, many manipulators do not satisfy (2n+1)R circular definition because of the joint offsets, like the Franka

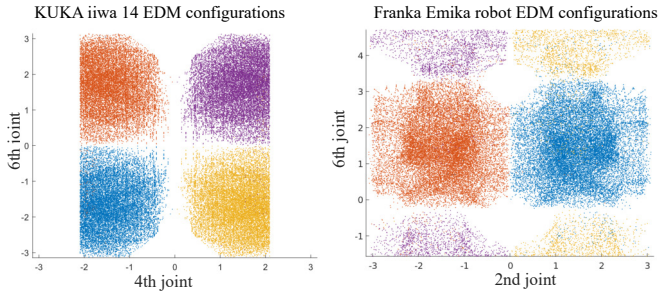

Fig. 13. Left: value of joint $4, 6$ for configurations in 4 different flips in iiwa 14. Right: the value of joint $2, 6$ for all configurations in Franka Emika research robot. Colors represent distinct flips.

Emika research robot, ABB GoFa, and Kassow robots. They cannot be decomposed into flip pairs; therefore, the sign analysis does not hold. However, as shown in Fig. 13, we can still observe some joints that indicate a clear edge between different flipped regions in both kind of robots. Finding these edges with analytical or geometric methods for robots with other kinematics could be an interesting future direction.

VII. Conclusion

This paper presents automatic algorithms for the generation of maps that encode embodied geometric capabilities of serial chain manipulators. These maps provide new insights into existing robots capabilities thanks to the new concepts of continuous and flip nullspace. Our algorithms enable the robot to systematically select more robust task-specific configurations across the scenarios with different case studies. As future works, we also aim to incorporate safe planning around humans by exploring safety-aware metrics in grid-space, for instance, by extending our previous work [41].

References

[1] J. Lachner, V. Schettino, F. Allmendinger, M. D. Fiore, F. Ficuciello, B. Siciliano, and S. Stramigioli, "The influence of coordinates in robotic manipulability analysis," *Mech. and machine theory*, 2020.

[2] J. Dong and J. C. Trinkle, "Orientation-based reachability map for robot base placement," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015.

[3] Y. Yang, L. Zhang, X. Cheng, J. Pan, and R. Yang, "Compact reachability map for excavator motion planning," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

[4] T. Birr, C. Pohl, and T. Asfour, "Oriented surface reachability maps for robot placement," in *IEEE Intl. Conf. on Robo. and Auto. (ICRA)*, 2022.

[5] K. Leibrandt, L. da Cruz, and C. Bergeles, "Designing robots for reachability and dexterity: Continuum surgical robots as a pretext application," *IEEE Transactions on Robotics*, 2023.

[6] N. Vahrenkamp, T. Asfour, G. Metta, G. Sandini, and R. Dillmann, "Manipulability analysis," in *IEEE-RAS Humanoids*, 2012.

[7] W. Thibault, V. Rajendran, and K. Mombaur, "Bimanual manipulation workspace analysis of humanoid robots with object specific coupling constraints," in *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*. IEEE, 2022, pp. 141–148.

[8] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Robot placement based on reachability inversion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 1970–1975.

[9] H. Kim, L. M. Miller, A. Al-Refai, M. Brand, and J. Rosen, "Redundancy resolution of a human arm for controlling a seven dof wearable robotic system," in *2011 Annual international conference of the IEEE engineering in medicine and biology society*. IEEE, 2011.

[10] S. Kim, C. Kim, and J. H. Park, "Human-like arm motion generation for humanoid robots using motion capture database," in *2006 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*. IEEE.

[11] E. Sennesh, J. Theriault, D. Brooks, J.-W. van de Meent, L. F. Barrett, and K. S. Quigley, "Interoception as modeling, allostasis as control," *Biological Psychology*, vol. 167, p. 108242, 2022.

[12] J. U. Korein, *A geometric investigation of reach*. MIT press, 1986.

[13] Z. Wang, S. Ji, Y. Li, and Y. Wan, "A unified algorithm to determine the reachable and dexterous workspace of parallel manipulators," *Robotics and Computer-Integrated Manufacturing*, 2010.

[14] T. Yoshikawa, "Manipulability of robotic mechanisms," *The international journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

[15] C. Gosselin and J. Angeles, "A Global Performance Index for the Kinematic Optimization of Robotic Manipulators," *Journal of Mechanical Design*, vol. 113, no. 3, pp. 220–226, 09 1991.

[16] T. Sandakalum, N. X. Yao, and M. H. Ang, "Inv-reach net: Deciding mobile platform placement for a given task," in *2022 IEEE-RAS Humanoids*. IEEE, 2022.

[17] O. Porges, T. Stouraitis, C. Borst, and M. A. Roa, "Reachability and capability analysis for manipulation tasks," in *ROBOT2013: First Iberian Robotics Conference: Advances in Robotics*, 2014.

[18] G. Wang, W. Li, C. Jiang, D. Zhu, Z. Li, W. Xu, H. Zhao, and H. Ding, "Trajectory planning and optimization for robotic machining based on measured point cloud," *IEEE Transactions on robotics*, 2021.

[19] Y. Kong, J. Wang, N. Zhang, S. Song, and B. Li, "Dexterity analysis and motion optimization of in-situ torsionally-steerable flexible surgical robots," *IEEE Robotics and Automation Letters*, 2022.

[20] K. C. Gupta and B. Roth, "Design Considerations for Manipulator Workspace," *Journal of Mechanical Design*, vol. 104, no. 4, 10 1982.

[21] F. Zacharias, C. Borst, and G. Hirzinger, "Capturing robot workspace structure: representing robot capabilities," in *IEEE/RSJ Int. Conf. Intelligent Robots & Systems*, 2007.

[22] S. Patel and T. Sobh, "Manipulator Performance Measures - A Comprehensive Literature Survey," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 77, 2014.

[23] L. F. Figueredo, R. C. Aguiar, L. Chen, S. Chakrabarty, M. R. Dogar, and A. G. Cohn, "Human comfortability: Integrating ergonomics and muscular-informed metrics for manipulability analysis during human-robot collaboration," *IEEE Robotics and Automation Letters*, 2020.

[24] N. Vahrenkamp, E. Kuhn, T. Asfour, and R. Dillmann, "Planning multi-robot grasping motions," in *IEEE-RAS Humanoids*, 2010.

[25] L. Jamone, L. Natale, K. Hashimoto, G. Sandini, and A. Takanishi, "Learning the reachable space of a humanoid robot: a bio-inspired approach," in *4th IEEE RAS & EMBS Int. Conf. Biomedical Robotics and Biomechatronics (BioRob)*, 2012.

[26] A. Makhal and A. K. Goins, "Reuleaux: Robot base placement by reachability analysis," in *IEEE Int. Conf. Robo. Computing (IRC)*, 2018.

[27] A. Ghosal, "Resolution of redundancy in robots and in a human arm," *Mechanism and Machine Theory*, vol. 125, pp. 126–136, 2018.

[28] K. M. Lynch and F. C. Park, *Modern robotics*. Cambridge University Press, 2017.

[29] D. Tolani, A. Goswami, and N. I. Badler, "Real-time inverse kinematics techniques for anthropomorphic limbs," *Graphical models*, 2000.

[30] J. Jiang, Y. Wang, Y. Jiang, H. Xie, H. Tan, and H. Zhang, "A robust visual servoing controller for anthropomorphic manipulators with field-of-view constraints and swivel-angle motion: Overcoming system uncertainty and improving control performance," *IEEE Robotics & Automation Magazine*, vol. 29, no. 4, pp. 104–114, 2022.

[31] Z. Li, M. Brandstötter, and M. Hofbaur, "Kinematic redundancy analysis for (2n+1)r circular manipulators," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 755–767, 2022.

[32] R. M. Murray, Z. Li, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 2017.

[33] E. B. Saff and A. B. J. Kuijlaars, "Distributing many points on a sphere," *The Mathematical Intelligencer*, vol. 19, pp. 5–11, 12 1997.

[34] R. Fletcher, *Practical methods of optimization*. J Wiley & Sons, 2000.

[35] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2022.

[36] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *IEEE Int. Conf. Robot. & Autom.*, 2000.

[37] C. L. Lück, "Self-motion representation and global path planning optimization for redundant manipulators through topology-based discretization," *Journal of Intelligent and Robotic Systems*, 1997.

[38] J. Burdick, "On the inverse kinematics of redundant manipulators: characterization of the self-motion manifolds," in *Proceedings, 1989 Intl. Conf. on Robotics and Automation (ICRA)*, 1989.

[39] A. M. Zanchettin and P. Rocco, "A general user-oriented framework for holonomic redundancy resolution in robotic manipulators using task augmentation," *IEEE Transactions on Robotics*, 2011.

[40] A. Albu-Schäffer and A. Sachtler, "Redundancy resolution at position level," *arXiv preprint arXiv:2202.09869*, 2022.

[41] R. Laha, W. Wu, R. Sun, N. Mansfeld, L. F. Figueredo, and S. Haddadin, "S*: On safe and time efficient robot motion planning," in *2023 IEEE Intl. Conf. on Robot. and Auto (ICRA)*. IEEE, 2023.