Visualizing High-Dimensional Configuration Spaces For Robots: A Comprehensive Approach for Quantitative and Qualitative Analysis

Jorge Ocampo Jimenez, Wael Suleiman

Visualizing High-Dimensional Configuration Spaces For Robots: A Comprehensive Approach for Quantitative and Qualitative Analysis

Jorge Ocampo Jimenez and Wael Suleiman

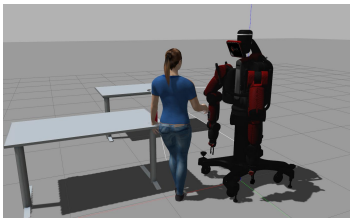
***Abstract*—The reconstruction of Configuration Space (CS) from a limited number of samples plays a vital role in expediting motion planning for random tree algorithms. Traditionally, the evaluation of CS reconstruction is performed through collision checking. However, employing the collision checker as an evaluation measure can be misleading. In particular, a collision checker may exhibit high accuracy even when only a subset of the original CS is reconstructed, limiting the motion planner's ability to find paths comparable to those in the original CS. Additionally, a significant challenge arises when dealing with high-dimensional CSs, as it becomes increasingly difficult, if not impossible, to perform qualitative evaluations when working in dimensions higher than three.**

In this paper, we introduce a novel approach for representing high-dimensional CSs of manipulator robots in a 2D format. Specifically, we leverage the kinematic chain of manipulator robots and the human ability to perceive colors based on hue. This allows us to construct a visualization comprising a series of pairs of 2D projections. We showcase the efficacy of our method in representing a 7-degree-of-freedom CS of a manipulator robot in a 2D projection. This representation provides qualitative insights into the joint boundaries of the robot and the collision state combinations. From a quantitative perspective, we show that the proposed representation not only captures accuracy but also furnishes additional information, enhancing our ability to compare two different high-dimensional CSs during the deployment phase, beyond what is usually offered by the collision checker. The source code is publicly available on our repository [1].

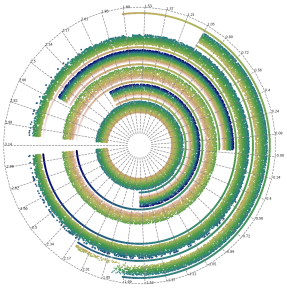
***Index Terms*—Random sampling, CS Representation, Accuracy, Manipulator Robots.**

I. Introduction

In recent years, significant advancements have been made in modeling the reconstruction of configuration space in path planning. Notably, when applying probabilistic random tree algorithms, an accurate approximation of the CS proves instrumental in expediting the discovery of Collision Free (CF)-paths. However, evaluating the quality of reconstruction, particularly for high-dimensional CSs exceeding three dimensions, remains a challenging task. Qualitatively assessing the reconstruction's fidelity and directing sampling toward CF-regions becomes progressively intricate as dimensionality increases. Distinguishing between discrete datasets derived from CF sampled states in the CS of a robot and other types of datasets in unrelated domains is crucial. In the case of the robot's CS, the key distinction arises from its inherent spatial nature, where similarities can be quantified through Euclidean distance measurements between states. However, the primary challenge lies in directly representing the CS as a Riemannian manifold, a subset of $\mathbb{R}^t$ for some $t \geq n$ and $n \geq 3$, an abstraction often unfamiliar to most. Moreover, the point cloud representation may lack a foundational ground plane and may exhibit limitations concerning everyday perceptual bases [1]. In path planning tasks, only CS representations limited to 3d could be shown graphically without using the projection of the path in the working space (WS)[2][3][4]. Another common challenge encountered in working with approximations of CF-CSs is the assessment of reconstruction accuracy. Relying solely on accuracy data from the collision checker to evaluate the reconstruction of the CF-CS can be misleading. In certain scenarios, it's feasible to achieve 100% accuracy in approximating CF-states, even when only a subset of the CF-CS is being reconstructed. This can, in turn, limit the number of CF-paths that can be discovered.



(a) Example of the WS were the human is close to the right arm of the robot. The changes in the WS are reflected in the CS



(b) Original CS representation coming from a uniform random sampler and a collision checker

Fig. 1: 2D visualization of a CF-CS for a 7-degree-of-freedom (DOF) robot manipulator.

Considering these challenges, our objective is to assess the reconstruction of the CS both qualitatively and quantitatively before its implementation in the path planning tasks by proposing a novel visualization of the CF-CS.

According to [5], a visualization serves as a mapping between raw data and visual information, involving a sequence that transforms the data into its final visual representation. In [6], the authors categorize multidimensional data visualization into three main stages:

- Data Transformation: This stage employs data analy-

Authors are with Electrical and Computer Engineering Department, Université de Sherbrooke, Quebec, Canada (e-mail: {Jorge.Ocampo-Jimenez; Wael.Suleiman}@USherbrooke.ca)

[1]https://bitbucket.org/joro3001/multidimensionalplots/

sis techniques to extract meaningful insights from the dataset. Often, this process involves some data reduction, retaining only pertinent information. For instance, methods like Principal Component Analysis (PCA) [7] and factor generation with subgroups [8] are used.

- Visual Mapping: In this stage, the meaningful information obtained is encoded to create a visual representation. Various techniques, such as scatter plots [9] and parallel coordinates [10], are used to convey this information visually.
- View Transformation: Techniques in this stage adapt the display to emphasize critical features or aspects of the data. Examples include splatter plots using Kernel Density Estimation (KDE) [11] and continuous parallel coordinates [12].

For a more detailed discussion of the various types of data visualization, readers can refer to [6].

Our proposed approach primarily falls into the categories of Visual Mapping and View Transformation. The process entails establishing a ground plane that connects all the projections from unidimensional θ_{i+1} to S^1, preserving the relationship with the ground plane θ_i. This ground plane can be likened to a coordinate chart φ that spans the manifold M by leveraging the constraints inherent in rotative manipulators' CSs. By adhering to the definition of the kinematic chain, the forward kinematic process follows a sequence of paired homogeneous transformations, where each rotation matrix depends solely on the last rotation matrix. This line of reasoning suggests that it is possible to plot the CS interactions of each dimension using only the last dependent entry while still representing the complete CS and its dependencies. These dependencies are graphically represented by colors and arc widths, combining elements of scatter plots, radial coordinates, hue and saturation, as illustrated in Fig. 1. Our approach takes advantage

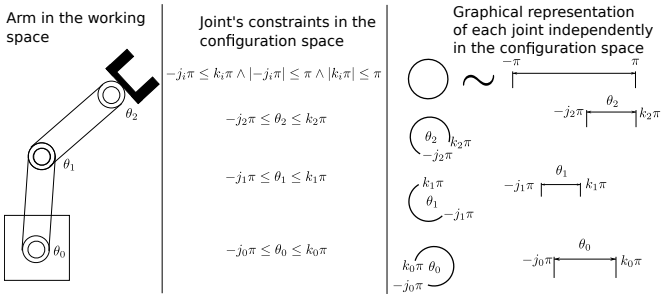


Fig. 2: A 3 DOF robot. We can easily observe the constraints of each joint of the robot by looking at the entry θ_i from each CF-state $\boldsymbol{\theta}$, with $j_i, k_i \in \mathbb{R}$ and $i \in \mathbb{N}$. Each unconstrained joint can be seen as a circle that is isomorphic to a line segment between $-\pi$ and π. It is not possible to observe the dependencies between the joints.

of the isomorphism between a circle and a line segment to represent the angles within the CS as arcs or line segments. This approach allows us to create a representation that effectively emulates the rotations of each joint and the constraints governing them. Notably, this visualization method has the unique ability to reveal constraints as distinct "holes" within specific color dependencies. By employing this isomorphic technique, we achieve a powerful means of conveying the intricate interplay between the robot's joints and their associated limitations.

II. Contributions

The primary contribution of this work lies in the development of a 2D graphical representation for discrete, high-dimensional CSs. This representation serves as a valuable tool for practitioners in robotics and machine learning by aiding them in the visual identification of CF-regions within the CS by:

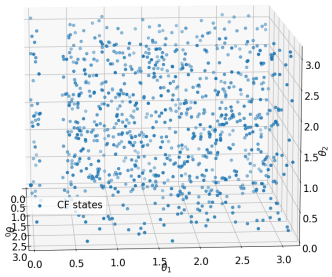
- Visual Identification of CF-Regions: We introduce a novel approach that allows practitioners to visually discern the areas within high-dimensional manipulator robot CSs that are CF. This visual guidance greatly facilitates the qualitative analysis of different CSs.
- Homology Group Detection: The representation enables the detection of homology groups, particularly H_0 and H_1, through the absence or presence of colors and white spaces. For higher-order homology groups (H_k with $k \geq 2$), identification becomes possible by adhering to the color constraints established by the independent joints within the sequence $\theta_0, \theta_1|\theta_0, \theta_1, \theta_2|\theta_1, ..., \theta_i, \theta_{i+1}|\theta_i, ..., \theta_{n-1}, \theta_n|\theta_{n-1}$.
- Qualitative Comparison of CSs: Our proposed representation offers an intuitive means to qualitatively compare two different CSs.
- Local Region Differences: In cases where two CSs exhibit visual similarity, our representation empowers users to employ specific operations within the visualization to identify differences in local regions.
- Encoding of Numerical Information: The representation also encodes numerical information regarding the accuracy of a CS approximation achieved by the original CS's collision checker.
- Metric-Based Evaluation: Given that our 2D representation of high-dimensional CSs can be analyzed using metrics from the field of computer vision, we can measure the proximity of an approximation to the original CS.

III. Representation construction

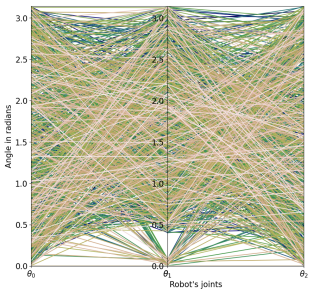
The primary objective of the proposed visualization is to offer both qualitative and quantitative insights into the representation of a CS. Specifically, we aim to capture the essential characteristics of a rotative manipulator robot for motion planning tasks. This includes providing a comprehensive view of CF positions of the robot's joints in response to variations in the robot's current state within the working space. Additionally, our representation allows us to visualize the constraints imposed by obstacles in the environment, further enhancing our understanding of the CS.

A. Kinematic chain and joint interaction

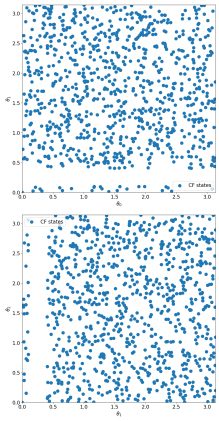
In the context of a rotative manipulator robot's CS, characterized by the presence of $n \in \mathbb{N}$ joints, each CF-state is represented as a tuple in a dataset of size $m \in \mathbb{N}$, the form of



(a) In this example, there is a 3D gap generated by the constraints in θ_1.



(b) In this parallel coordinates representation of the same CS, each line corresponds to an independent state, presented as a unique colored line.



(c) 2D projections of $\theta_0 \times \theta_1$ and $\theta_1 \times \theta_2$.

Fig. 3: 5000 CF-states sampled from a 3D CS. Parallel coordinates can effectively capture the constraints within individual joints, providing insight into each joint's behavior but fail to illustrate the complex interactions between joints. The 2D projections of Fig. 3c capture these interactions, such as the presence of a 3D hole by projecting it in two dimensions with different pair of joints, but the simultaneous interactions across individual samples in all dimensions can be challenging to discern.

$\boldsymbol{\theta}(j) = [\theta_0(j), \theta_1(j), ..., \theta_n(j)]$, where $-\pi \leq \theta_i(j) \leq \pi, j \in \mathbb{N}, j \leq m$. Each entry, denoted as θ_i, can be independently visualized in a 2D representation. This representation employs either polar coordinates in the form of $[\cos(\theta_i(j)), \sin(\theta_i(j))]$ or a 1D bounded line segment within the subset of $[-\pi, \pi]$.

While visualizing $\boldsymbol{\theta}(j)$ provides insight into the rotational boundaries of individual robot joints, it falls short in capturing the expressiveness of constraints imposed by the kinematic chain. For example, if θ_i is restricted to the interval $[-\pi, \pi]$ due to self-collisions or obstacles within the robot's WS, it has a cascading impact on joint θ_{i+1}. However, when we independently plot all potential values of entry θ_{i+1} from a uniformly sampled dataset, it results in an almost continuous representation of the joint. This continuous representation obscures our ability to identify CF regions shaped by constraints originating from θ_i and affecting θ_{i+1}, as depicted in Fig. 2.

To project multidimensional data into lower dimensions without losing the connection between non-consecutive entries of each data point, it is essential to maintain the relationship between the values within the $\boldsymbol{\theta}(j)$ vector. For instance, when conducting a 2D projection, there is a risk of information loss when transitioning between consecutive entries θ_i and θ_{i+1}. This is particularly problematic because, at any given point, we have information only about the values of the previous entry, θ_i. This is showcased in Fig. 3.

An advantage of working with rotative manipulator robots is that each link connected to joint θ_{i+1} can be considered dependent only on the relative angle with respect to θ_i, this can be seen as a series of 2D projections. The set of achievable states for the robot is defined by the values of each θ_i, constrained by homogeneous transformations ${}^{i}_{i+1}T(\theta_i)$.

We propose a solution to address the visualization of high-dimensional data by 2D projections by combining labeling from parallel coordinates with the 2D projection of entry joints $\theta_i \times \theta_{i+1}$ following the order of the homogeneous transformations of a manipulator robot. In this approach, each point in the 2D projection is labeled according to the order of the homogeneous transformations; $\theta_0(j), \theta_i(j), \theta_{i+1}(j), \ldots, \theta_n(j)$. This integration of labeling enables us to preserve the relationship intra-joint of the original data from the representation and simultaneously detect 2D gaps within each projection $\theta_i \times \theta_{i+1}$, as visually represented in Fig. 4. Since we have a discrete number of samples, interpreting the impact of each individual point on the local regions becomes challenging; depending on the spread of the samples, the data points may not effectively convey the topology of the configuration space (CS).

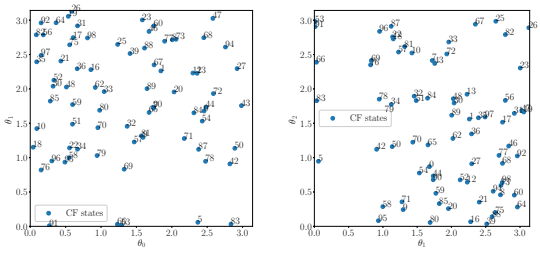
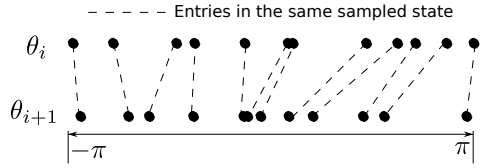


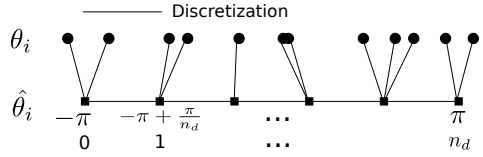
Fig. 4: Proposed data structure. By assigning a unique id to each sampled point in $\theta_0 \times \theta_1$ and $\theta_1 \times \theta_2$, we are able to keep the relationship between joint values of the original dataset $\theta_0 \times \theta_1 \times \theta_2$.

Now that we know that it is feasible to keep most of the n-dimensional CS when building the 2d data structures, the act of visualizing continuous data to a discrete format for the practitioner would result in the loss of information; this happen even when the visualization has the same dimension as the data being plotted. The discretization also creates branching of the values of the original data. This loss of information can be used in our advantage to plot discretization clusters that approximate of the original data by its euclidean distance with a sequence of 2D projections.

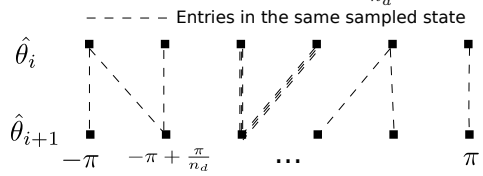
The visual representation of θ_i given $\theta_0, ..., \theta_{i-1}, \theta_{i+1}, ..., \theta_n$ will be subject to discretization when viewed on an output device; which impose limitations on the number of pixels or file size represented as discrete grid. Consequently, the potential for displaying various



(a) Each time that we sample a CF-state $\boldsymbol{\theta}(j)$, we consider that each entry $\theta_i(j)$ has a relationship with the rest of the entries $\theta_k(j)$ of the sample j. This relationships can be ordered following the kinematic chain from θ_0 to θ_n.



(b) The visual representation from the CS will be represented in a discrete output, each $\theta_i(j)$ will be approximated given the resolution of the output device. We can control the discretization before being sent to the output for visualization by distributing uniformly the sampled data given a uniform interval of size $\frac{1}{n_d}$ where $n_d \in \mathbb{N}$.



(c) The new vector states $\hat{\boldsymbol{\theta}}(j) = \boldsymbol{\theta}(j) + \boldsymbol{\sigma}$ will be created by keeping the relationship of the previous $\theta_i(j), \theta_{i+1}(j)$ on the perturbed by discretization values $\hat{\theta}_i(j), \hat{\theta}_{i+1}(j)$.

Fig. 5: Discretization effect when the continous values from the CS are displayed on a discrete output device.

values for each pair of θ_i and θ_{i+1} will be constrained by the discretized state samples. This is achieved through $\hat{\boldsymbol{\theta}}(j) = \boldsymbol{\theta}(j) + \boldsymbol{\sigma}$, where $\boldsymbol{\sigma} = [\sigma_0, ..., \sigma_n]$ and $\sigma_i \in \mathbb{R}$.

The visualized $\hat{\boldsymbol{\theta}}(j)$, will increase the probability of observing $\hat{\boldsymbol{\theta}}(j) = \hat{\boldsymbol{\theta}}(k)$, where $k \leq m$ and $k \neq j$ as the sample size grows, provided the discrete resolution remains constant. This is visually depicted in Fig. 5. The discretization naturally leads to a tree-like structure, reflecting the interdependence of individual joints. The children of the roots θ_i are expressed as follows:

$$\theta_{i+1}|\theta_i(j) = \{\forall \theta_{i+1}(k) : \theta_{i+1}(k) \in \boldsymbol{\theta}(k) \wedge \theta_i(k) = \theta_i(j), \\ \forall k \in \mathbb{N}, 0 \leq k \leq m\} \quad (1)$$

For the sake of generality, we will treat $\boldsymbol{\theta}$ as equal to $\hat{\boldsymbol{\theta}}$, considering that $\boldsymbol{\theta}$ is sampled from a uniform distribution. As n_d, the discretization size of the uniform interval, approaches infinity, we have $\boldsymbol{\theta} \approx \hat{\boldsymbol{\theta}}$.

This tree-like structure approach enables the detection of patterns within the sampled points cloud of states by considering the interdependencies between consecutive joints. This concept of grouping information together is reminiscent of clustering techniques that use Euclidean distances between the various sampled points.

B. Visualization

With the data organized into 2D sequenced projections and clusters, we now turn our attention to effectively conveying this information to the user. The goal is to provide a clear visual representation that allows users to discern CF regions and identify the constraints within the CS.

A key challenge is to uniquely and graphically identify the root values expressed by the trees $\theta_i|\theta_{i+1}$ without inflating the dimensions of the visualization and also preserving the information of each independent sampled state. One elegant solution to address this challenge is to employ color mapping.

Color mapping is a powerful technique that can be likened to a curve within the 3D color space [13]. In our approach, we employ color to represent the Euclidean distance within the CS. Specifically, states that are close in terms of their parent values and current configurations will exhibit similar colors in the visualization. This color mapping not only aids in visually capturing the CS structure but also helps practitioners in clustering states within CF local regions.

Importantly, we maintain the visualization in 2D, as research has shown that 2D visualizations can outperform their 3D counterparts, particularly in tasks related to obtaining an overview of the data [14].

The color mapping strategy involves associating a distinct color with each root tree $\theta_{i+1}|\theta_i$. The creation of these trees is necessary because of the limited color depth available in the output device, which constrains the number of colors that can be effectively represented. Although it is possible to assign a unique real number to each state in the color map, the mapping from the color map to RGB values would result in very close values in the color map domain being assigned to the same RGB color. From the user's perspective, two states with nearly identical RGB colors would appear the same. This process can be likened to discretizing the original data, leading to the formation of a tree-like structure as described in Eq. (1). Given that each tree can be viewed as a 2D division of the $n \times m$ CS grid, it becomes feasible to implement a unique color map for each joint θ_i.

The process begins by dividing the color mapping across the uniform interval of $[-\pi, \pi]$ and the discretization size n_d for each entry i. In practical terms, when we have a parent value $\theta_i(j)$, we select a specific color from the color map. This chosen color is then assigned to all the children $\theta_{i+1}|\theta_i(j)$ of the parent value, creating a clear visual representation of the kinematic constraints within the CS. The color assignment process is shown in Fig. 6.

In order to preserve as much of the information between non-consecutive joints of the original dataset encoded by the relationship between the 2D trees, $\theta_{i+1}|\theta_i$ and the rest of the joints in the sampled state, each time we assign colors to trees of the form $\theta_{i+1}(j)|\theta_i(j)$; where $\theta_{i+1}(j)$ can be also a children of different parent $\theta_i(k)$, we add a slightly different color to the states $\boldsymbol{\theta}(k)$ by a perturbation $\boldsymbol{\epsilon}(k) = \epsilon(k)[1, ..., 1]^T \in \mathbb{R}^n$, $\epsilon(k) \in \mathbb{R}^+$. While introducing noise to the original data may introduce imprecisions to the original model, it can help prevent the unbounded error associated with the possible reconstruction of additional states resulting from the branching of the data. It is worth noting that the error of the perturbed state is bounded by $\|\boldsymbol{e}(k)\|$ given that the perturbation follows $\|\boldsymbol{\theta}(k) + \boldsymbol{e}(k)\| \leq \|\boldsymbol{\theta}(k)\| + \|\boldsymbol{e}(k)\|$.

After assigning the colors to the states, we precede to create

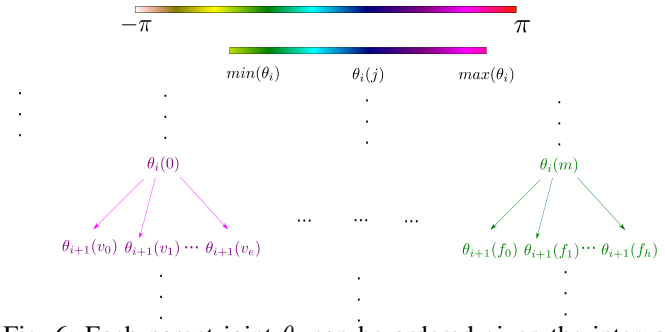


Fig. 6: Each parent joint θ_i can be ordered given the interval $[-\pi, \pi]$. Then we assign a color to each tree $\theta_i|\theta_{i+1}(j)$ by distributing uniformly the value of the joint θ_i given the number of samples in the color space.

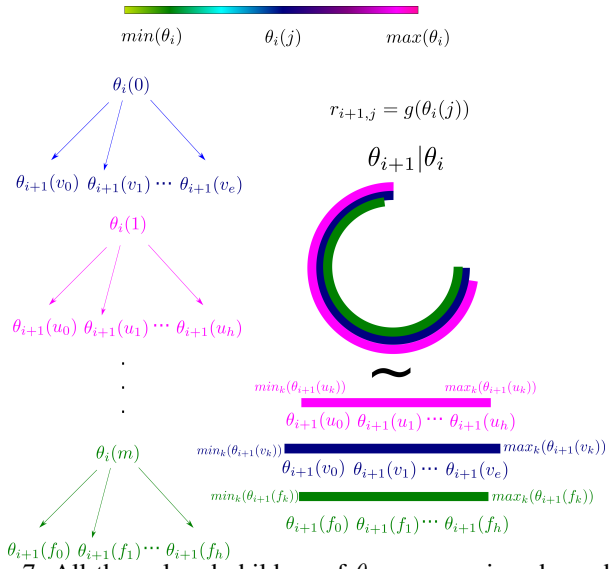


Fig. 7: All the colored children of θ_{i+1} are assigned a radius r_{i+1} by an increasing function $g(\theta_i)$ and are plotted in their respective 2d positions $g(\theta_i)cos(\theta_{i+1}), g(\theta_i)sin(\theta_{i+1})$. The length of each circunference are given by the maximum and minimum $\theta_i(j)$ from each $\theta_{i+1}(j)$

a visual representation within the context of a rotative manipulator robot, we have adopted a method that projects each tree onto a segment of the circumference of S^1. This approach allows us to effectively identify the angles θ_{i+1} at which the joint is in a CF state. Much like the tree-based representation, the color assigned to each arc corresponds to the angle of the parent joint, which is denoted as θ_i. Essentially, each parent $\theta_i(j)$ is visually represented by a dedicated arc on the circumference of S^1. In order to visualize all the colored arcs $\theta_i|\theta_{i+1}(j)$ distinctly, we employ an increasing function denoted as $g(\theta_i|\theta_{i+1}(j))$ to assign a radius $r_{i+1}(j)$ to each joint. This technique ensures that the arcs do not overlap with each other. The length of each arc is determined by the specific kinematic constraints of the joint. The reader can refer to Fig. 7 for a visual representation of this process.

By appropriately scaling each set of arcs $\theta_i|\theta_{i+1}$, it becomes possible to plot the values of all the joints simultaneously. To easily indicate the relationship between the colors and the

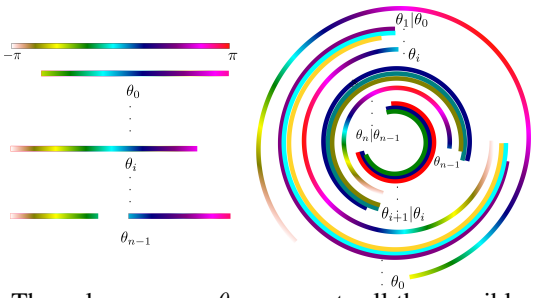


Fig. 8: The color map arc θ_i represents all the possible entries of the joint given the dataset. This help us to localize the original values of the sampled states in radians following the kinematic chain of the robot.

corresponding angle of the parent joint θ_i, a color legend is employed. This legend visually associates colors with their specific angle in the parent joint θ_i.

To further improve clarity and readability, we add a chart φ that connects the entry θ_i and θ_{i+1} in the form of a line segment spanning the interval $[-\pi, \pi]$. This line segment defines the color map for the trees/arcs of the children $\theta_{i+1}|\theta_i$ and represents the boundaries of the entry θ_i, independent of any other entry $\theta_{i+k}, k \in \mathbb{Z}, k \neq 0$. The visual representation of the process is illustrated in Fig. 8.

IV. Representation Results

A 2D graphical representation of high-dimensional CSs provides valuable insight into the similarities between two distinct CSs. However, it's essential to assess the representation's effectiveness in quantitatively measuring the relationship between these CSs, especially when compared to accuracy-based metrics. This evaluation is crucial for the reconstruction of CSs, as it plays a significant role in expediting path planning using random-based algorithms, often complemented by machine learning models.

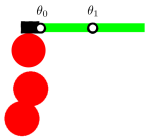
A. Qualitative evaluation

In the context of a simple 2D manipulator robot, our proposed visualization naturally captures the essence of the 2D projection of $\theta_0 \times \theta_1$ as depicted in the trivial example in Fig. 9. The challenge is to demonstrate how effectively this concept translates to high-dimensional CSs.

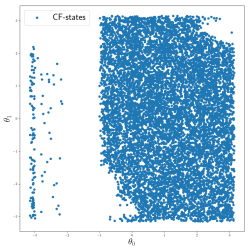
To evaluate the effectiveness of our visualization in high-dimensional CSs, we gathered a dataset of CF states from the CS of a 7-DOF right arm of a Baxter industrial robot. This evaluation included scenarios where a simulated human presence within the robot's WS introduced geometric constraints, which significantly impacted the representation of the CF-CS, as illustrated in Fig. 1.

To estimate collisions between the robot and the simulated human and construct a training dataset, we utilized MoveIt [15] and OMPL (Open Motion Planning Library) [16] for collision checking

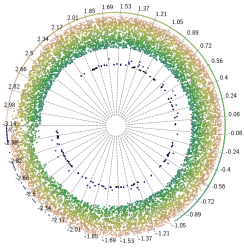
We changed the position and orientation of the human in two different random configurations within the WS of the robot. We sampled 10,000 points for each configuration to construct the CS.



(a) A simulated 2d robot. The links of the robot are the green rectangles. The red circles represent obstacles in the WS.

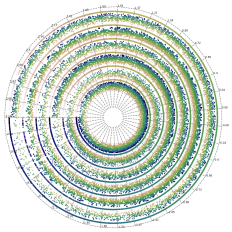


(b) $\theta_0 \times \theta_1$ projection of the 2d CS of the robot in radians.

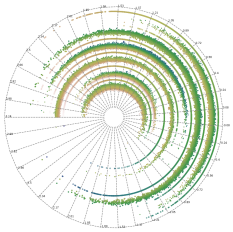


(c) Proposed visualization of the 2d robot. Here the big gap of the $\theta_0 \times \theta_1$ projection is shown as the lack of points with colors in $\theta_1|\theta_0$ following the gradient of the map θ_0

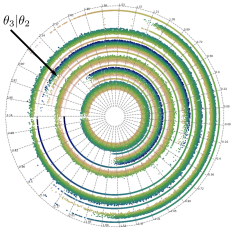
Fig. 9: WS and CF simulation of 2 DOF. Our visualization highlights the changes that comes from the dependency of the two joints: the lack of colors in the 2d arc $\theta_1|\theta_0$ means that the collision states were generated by constraints in θ_0, the lack of continuity of points of the same colors indicates constraints coming from the joint θ_1.



(a) GAN trained with 0 epochs.

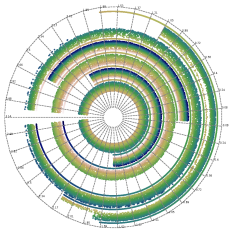


(b) GAN trained with 10 epochs.

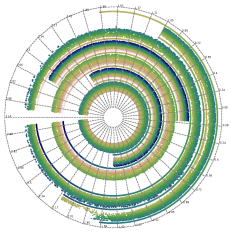


(c) GAN trained with 100 epochs.

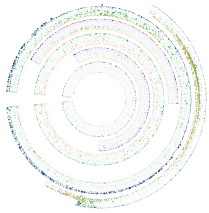
Fig. 10: Reconstruction of the original CS of 7 DOF from Fig. 1 by training a GAN. We can see that given the increasing of epochs the model starts to reproduce more closely the training data.



(a) CS1.



(b) CS2.



(c) Negative image of the substracion of CS1-CS2.

Fig. 11: CSs that look similar when the WS changes. We can use the difference between the CSs visualizations to localize more easily the regions were the CSs diverge.

We proceeded to train a small example to assess the capability of representing high-dimensional data, with the objective of comparing how well a trained model can accurately reproduce the distribution of the original configuration space (CS).

We trained a Generative Adversarial Network (GAN) [17] to develop a model that can generate samples from one of the two configurations. We employed the architecture proposed in [18], with the modification that the workspace condition is now incorporated by simply concatenating 0 or 1 to the input of the latent vector.

We selected the "GistEarth" color map for mapping θ_i and $\theta_{i+1}|\theta_i$. This choice was influenced by the resemblance of this color map to luminance-controlled maps, such as "Rainforest" [19][20]. Luminance is a critical factor as it conveys information about the structural aspects of surfaces in three-dimensional space [21]. Moreover, this color map remains interpretable in grayscale for individuals with color vision deficiencies [19].

As demonstrated in Fig. 10, our visualization representation effectively illustrates the model's progress in learning the original CS as training epochs increase. Notably, the model continues to refine its understanding of the distribution governed by its parent joints. For example, the arc $\theta_3|\theta_2$ illustrates this ongoing learning process, as it has yet to fully capture the interrelationships between the joints, resulting in the distribution of colors in the visualization. This is further evidenced by the multitude of small holes in the original data, depicted in Fig. 1. These gaps denote potential collision configurations that the model is still in the process of lerning them.

Upon closer examination, we observed that when two CSs exhibit variations in only limited regions, it can be challenging for users to pinpoint these specific alterations within the CS. To address this issue, we leverage our image representation to identify such areas of interest. By employing image subtraction techniques, we can effectively extract regions where differ-

ences between two CSs are localized. These areas of interest are indicated to the user, guiding them to investigate specific regions exhibiting dissimilarity, highlighted by the values of the pixels of darker colors. This approach is exemplified in Fig. 11.

B. Quantitative evaluation

In the field of robotics and motion planning, the evaluation of CS reconstructions typically occurs when a machine learning model is put into action in an actual motion planning task. For instance, in the case of random trees-type algorithms, the conventional uniform sampler is often substituted with a model specifically designed to generate CF states. The primary aim here is to minimize the need for extensive collision-checks, thereby accelerating the search for CF-paths. The evaluation of these paths predominantly revolves around two key metrics: the number of CF-paths discovered and the query time. However, this approach comes with a notable challenge.

For instance, when a Generative Adversarial Network (GAN) is employed to guide a sampler, the issue of mode collapse can arise. Mode collapse is a phenomenon where the generator predominantly produces a single sample or a very limited set of closely similar samples [22], thus ignoring a significant portion of the CS. As a result, when the robot is deployed in real-world scenarios beyond the boundaries of the GAN's collapsed representation, it may miss crucial CF-regions or paths. This situation is analogous to relying solely on accuracy and collision checking to measure a subset of the CF-CS. An illustrative example of this phenomenon is depicted in Fig. 12.

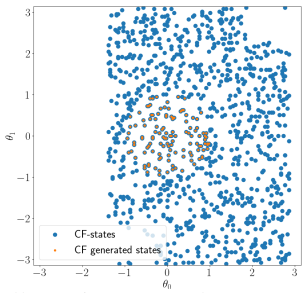


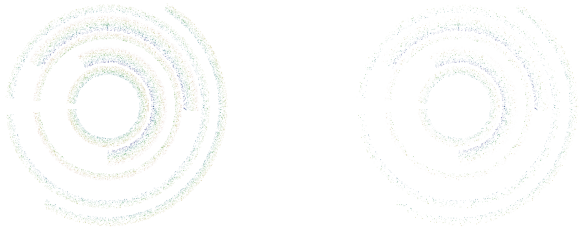
Fig. 12: Mode collapse in GAN: The generator covers a small subset of the true CF-CS; resulting in a scenario where the collision checker achieves 100% accuracy when assessing the generated samples.

Although our visualization offers valuable qualitative insights into high-dimensional CSs, it also has the potential to assess cases where only subsets of the CS are generated. Furthermore, it can serve as an approximation tool, given that it displays 2D projections of data from each pair of joints, θ_i and θ_{i+1}, along with a chart, φ.

First, we conducted a test of our visualization to determine if the visualization is able to captures information about accuracy from the collision checker. In this context, accuracy is defined as the ratio of CF-samples generated by the trained sampler, validated by the collision checker, to the total number of samples queried from the trained model.

We conducted experiments using the 100 different WS/CS pairs from the previous section. In these experiments, we introduced random collision states inside the boundaries of the joints and replacing some of the CF-states of the original CS. The amount of added collision states went from 10% to 100% of the total size of samples of each CF-CS, with 10% increments.

To assess the alignment between accuracy and our visualization approach, we conducted a set subtraction operation, represented as $A \setminus B = A \cap B^c$, between visualizations A and B. During this operation, we replaced all pixels in the perturbed CS visualizations that matched the non-perturbed CS visualizations by setting their RGB values to white. Subsequently, we calculated the ratio of non-white pixels that remained after this replacement to the number of non-white pixels before the replacement in the perturbed visualization. This ratio served as our metric for measuring accuracy. We discretized the data in 500 different uniform intervals between $[-\pi, \pi]$. If the discretization is not made; we will underestimate the accuracy when comparing two visualizations of the same CS but with different sampled states. We can see an example of the discretization effect on $A \setminus B = A \cap B^c$ in Fig. 13.



(a) Visualization of the set $A \setminus B$, here A has a smaller uniform interval during discretization

(b) Visualization of the set $A \setminus B$, here A has a bigger uniform interval during discretization

Fig. 13: The mesure derived from $A \setminus B$ from the visualizations will change depending on the discretization applied to the CS, given that the number of pixels that are in B^c and A will change. In this case, A is a reconstruction of Fig. 1b that has 10% of its states in collision, and B is the original Fig. 1b

Given that each of the 100 different CSs accuracies from visualizations are tested by measuring the correlation with the accuracy from the collision-checker, we present the average correlation using the Fisher-z transform; which results in 0.90 ± 0.10 using the predicted standard error. This strong linear correlation [23] is expected, as we are directly plotting the sampled data projections.

Despite the difference in scale given by the discretization process, the strong correlation indicates that the proposed visualization is able to capture the change in accuracy when dealing with non-complete reconstructions of the CF-CS. This suggests that the visualization captures the accuracy of the collision checker in the context of high-dimensional CSs and how the improvement of the approximation is reflected in the visualization.

The proposed visualization is further tested to measure the difference between two CSs, where one is a subset of the other,

this measurement is not available using the original collision checker. In the context of image generation, there are various methods to compare images. The simplest one is to use the Mean Squared Error (MSE) per pixel. We can compare the visualization subsets of a CS against the original visualization of the CS and measure how well the density of the CS is being reproduced. To evaluate this, we reduced the number of sampled points in each CS from 90% to 10%, generating nine different subsets, and estimated the MSE between the original CS with 100% of its data points and its subsets.

Amount of samples of the subset	MSE
90%	1.88 ± 0.01
80%	1.92 ± 0.01
70%	1.98 ± 0.01
60%	2.05 ± 0.01
50%	2.12 ± 0.01
40%	2.21 ± 0.01
30%	2.30 ± 0.01
20%	2.40 ± 0.01
10%	2.52 ± 0.01

TABLE I: Mean and standard deviations of MSEs of the 100 different CSs compared against their subsets with 90% to 10% of the original data samples. The MSE derived from the visualization effectively gauges the extent to which the subset deviates from the complete model.

As shown in Table I, as the number of points forming the original CS is decreased, the MSE increases. This implies that the visualization can effectively reflect how well a model is being reconstructed based on the extent of the CF-CS. This is a capability not achievable when relying solely on a collision checker as a reference.

V. Conclusions and future work

We introduce a novel 2D representation for high-dimensional CSs of manipulator robots, designed to facilitate the visual detection of regions where CSs differ due to changes in their WS. Our findings demonstrate that our visualization approach can effectively capture metrics like accuracy and error measurements between CS approximations, which broadens the provided information compared to traditional approaches that rely solely on collision checking to assess the difference between two CSs, especially in the context of CS learning. This visualization offers valuable insights to machine learning practitioners in the field of robotics, allowing them to gauge the quality of a model's approximation of the real CS.

Future research is aimed at adapting this visualization representation to various types of robots, such as mobile manipulators, where movement in the 2D plane might be unbounded or there is not a fixed reference to independently represent each state. Another potential application is to directly encode the CS into 2D visualizations for CS-related tasks. There are numerous image-to-image generation algorithms that could be employed to learn the 2D visualization representation of the CS. With sufficient resolution and color depth, it should be possible to obtain an exceptionally close representation of the original CS. This would enable the integration of state-of-the-art machine learning and computer vision techniques for CS reconstruction.

References

[1] M. Chalmers, "Using a landscape metaphor to represent a corpus of documents," in *Spatial Information Theory A Theoretical Basis for GIS*, A. U. Frank and I. Campari, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, pp. 377–390.

[2] M. J. Bency, A. H. Qureshi, and M. C. Yip, "Neural path planning: Fixed time, near-optimal path generation via oracle imitation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 3965–3972.

[3] A. H. Qureshi and M. C. Yip, "Deeply informed neural sampling for robot motion planning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 6582–6588.

[4] R. Reinhart, T. Dang, E. Hand *et al.*, "Learning-based path planning for autonomous exploration of subterranean environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1215–1221.

[5] S. Card, J. Mackinlay, and B. Shneiderman, Eds., *Readings in Information Visualization: Using Vision to Think*. California, USA: Morgan Kaufmann, 1999.

[6] E. Bertini, A. Tatu, and D. Keim, "Quality metrics in high-dimensional data visualization: An overview and systematization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2203–2212, 2011.

[7] K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," Nov. 1901.

[8] C. Turkay, A. Lundervold, A. J. Lundervold *et al.*, "Representative factor generation for the interactive visual analysis of high-dimensional data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2621–2630, 2012.

[9] M. Friendly and D. Denis, "The early origins and development of the scatterplot," *Journal of the History of the Behavioral Sciences*, vol. 41, no. 2, pp. 103–130, Spring 2005.

[10] M. d' Ocagne, "Coordonnées parallèles et axiales méthode de transformation géométrique et procédé nouveau de calcul graphique, déduits de la considération des coordonnées parallèles," *Nature*, vol. 31, pp. 551–552, 1885.

[11] A. Mayorga and M. Gleicher, "Splatterplots: Overcoming overdraw in scatter plots," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 9, pp. 1526–1538, 2013.

[12] J. Heinrich and D. Weiskopf, "Continuous parallel coordinates," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1531–1538, 2009.

[13] P. Kovesi, "Good colour maps: How to design them," *ArXiv*, vol. abs/1509.03700, 2015.

[14] M. Kraus, K. Angerbauer, J. Buchmüller *et al.*, "Assessing 2d and 3d heatmaps for comparative analysis: An empirical study," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, p. 1–14.

[15] D. Coleman, I. A. Şucan, S. Chitta *et al.*, "Reducing the barrier to entry of complex robotic software: a MoveIt! case study," *Journal of Sofware Engineering for Robotics*, vol. 5, no. 1, pp. 3–16, May 2014.

[16] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, December 2012.

[17] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza *et al.*, "Generative Adversarial Nets," in *27th International Conference on Neural Information Processing Systems*, vol. 2, 2014, pp. 2672–2680.

[18] J. Ocampo-Jimenez and W. Suleiman, "Improving path planning performance through multimodal generative models with local critics," *ArXiv*, 2023.

[19] E. van der Velden, A. R. Duffy, D. Croton *et al.*, "Model dispersion with prism: An alternative to mcmc for rapid analysis of models," *The Astrophysical Journal Supplement Series*, vol. 242, no. 2, p. 22, jun 2019.

[20] G. Kindlmann, E. Reinhard, and S. Creem, "Face-based luminance matching for perceptual colormap generation," in *Proceedings of the Conference on Visualization 2002*. USA: IEEE Computer Society, p. 299–306.

[21] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. New York, NY, USA: Henry Holt and Co., Inc., 1982.

[22] L. Metz, B. Poole, D. Pfau *et al.*, "Unrolled generative adversarial networks," *ArXiv*, 2017.

[23] A. Field, *Discovering Statistics Using IBM SPSS Statistics*, 4th ed. Sage Publications Ltd., 2013.