



Planning Fail-Safe Trajectories for Space Robotic Arms

Oliver Porges¹, Daniel Leidner² and Máximo A. Roa^{2*}

¹Agile Robots, Munich, Germany, ²Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany

A frequent concern for robot manipulators deployed in dangerous and hazardous environments for humans is the reliability of task executions in the event of a joint failure. A redundant robotic manipulator can be used to mitigate the risk and guarantee a post-failure task completion, which is critical for instance for space applications. This paper describes methods to analyze potential risks due to a joint failure, and introduces tools for fault-tolerant task design and path planning for robotic manipulators. The presented methods are based on off-line precomputed workspace models. The methods are general enough to cope with robots with any type of joint (revolute or prismatic) and any number of degrees of freedom, and might include arbitrarily shaped obstacles in the process, without resorting to simplified models. Application examples illustrate the potential of the approach.

Keywords: manipulation planning, robotic arms, fail-safe trajectories, fault-tolerant manipulators, space manipulator

OPEN ACCESS

Edited by:

Craig R. Caignan,
University of Maryland, United States

Reviewed by:

Abhijit Makhil,
Honeywell, United States
Biyun Xie,
University of Kentucky, United States

*Correspondence:

Máximo A. Roa
maximo.roa@dlr.de

Specialty section:

This article was submitted to
Space Robotics,
a section of the journal
Frontiers in Robotics and AI

Received: 14 May 2021

Accepted: 20 September 2021

Published: 30 November 2021

Citation:

Porges O, Leidner D and Roa MA
(2021) Planning Fail-Safe Trajectories
for Space Robotic Arms.
Front. Robot. AI 8:710021.
doi: 10.3389/frobt.2021.710021

1 INTRODUCTION

Robotic manipulators are a convenient tool for deployment in dangerous and hazardous environments for humans, for applications such as planetary exploration, on-orbit servicing, de-orbiting of uncooperative targets, and hazardous material handling in nuclear or chemical disaster/waste sites. High cost and low to none maintenance possibilities place strong demands on reliability for robots deployed in such scenarios. The economic cost of a failure in those environments is usually very high, as analyzed in Ellery et al. (2008). Thus, one of the major concerns in these situations are the post-failure capabilities of the robotic manipulator. Even though the redundancy of electronics and mechanics of the joints significantly reduces the risk, the use of a redundant manipulator adds another layer of safety for the operation. This paper is focused on the description of methods and tools for risk analysis and fault-tolerant path planning to ensure the post-failure task execution for such robotic manipulators.

Most of the literature on safety-critical operations of robotic manipulators considers that if a failure occurs, it would be a locked joint. A free-swinging joint is another possible failure, but unlikely to occur given the considerations employed in the mechatronic design of joints. Early works on fault-tolerant path planning or task design establish a link between robustness to failures and kinematic dexterity based on the minimum singular value of the post-failure Jacobian, as presented in Lewis and Maciejewski (1994). This post-failure dexterity is referred to as the *kinematic failure-tolerance measure (kfm)*. The *kfm* values are analyzed throughout the configuration space (C-space) in order to identify configurations of optimal fault-tolerance. An inverse of the optimal *kfm* is used to track an end-effector path to anticipate failures and guarantee post-failure task execution. The methods are demonstrated on a planar manipulator with three DoF (Degrees of Freedom) realizing a 2-DoF task without obstacles.

Global methods for fail-safe path planning have also been considered. A global path planning approach for post-failure operation by using a redundancy resolution algorithm was proposed in Paredis and Khosla (1996). At each point along a desired end-effector path, a set of acceptable fault-tolerant configurations is estimated. A connectivity graph is constructed from this set, capturing the pre-image topology along the path. In case of failure, a graph search discovers alternative trajectories to reach the goal end-effector pose. The approach is demonstrated on a 4-DoF robot. Another global approach in Lewis and Maciejewski (1997) discusses the conditions of existence of fault-tolerant regions of operation in C-space around the start and goal poses. The authors overcome computational limitations by computing off-line the self-motion manifolds, which are used to modify the task by changing the goal pose so that larger self-motion manifolds are used during the task execution. The path generation method artificially imposes restrictions on joint limits to avoid possible failure scenarios. The analysis and proof of existence of a fault-tolerant path is done using intersections of self-motion bounding boxes in C-space, representing safe ranges for operation. The self-motion manifolds are discovered through a sampling approach, and they are explored by executing a spiral motion along the manifold to determine its boundary and connectivity. Joint configuration constraints on fault-tolerant motion are derived from the intersection of start and goal self-motion bounding boxes. Potential collisions of the robot are not included in this method.

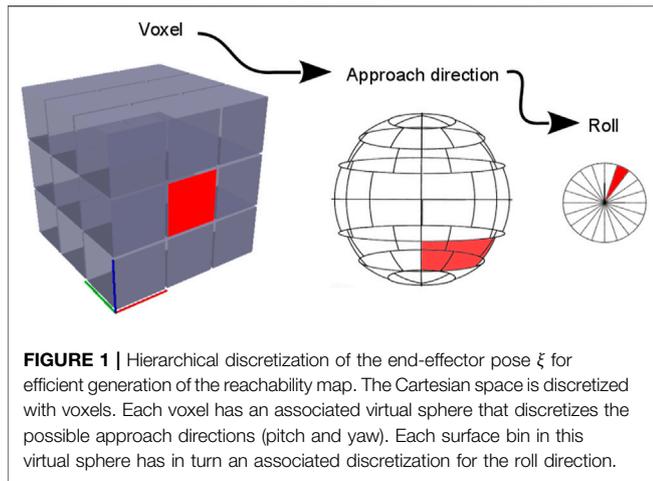
A hierarchical method for failure analysis using a least-constraint framework is presented in Ralph and Pai (1997). Here, the main task is to reach the goal, and the secondary task is resolving the null-space configuration to maximize the utility of redundancy. The task is formulated as a set of constraints rather than using waypoints, and introduces a performance measure called *longevity*, which reflects the connectivity of the C-space after a given failure. A dynamic programming task computes the *longevity* and searches for a post-failure recovery path using pre-computed recovery strategies (*ease of recovery*) required to keep the functionality of the robot. The method is demonstrated on a 4-DoF robot without obstacles; an extension of this framework to a 5-DoF robot is presented in Ralph and Pai (1999). A fault-tolerant path planning method for a 7-DoF manipulator is presented in Jamisola et al. (2006). A locked-joint failure at a start configuration defines *failure hyperplanes*. Intersections of these hyperplanes with the goal pose self-motion manifold define a failure hypercuboid. A web of paths is generated from the start configuration to a point on the goal pose self-motion manifold using monotonic paths. After checking continuity and collisions, an obstacle-free web of paths is created, defining a *failure surface*. While the manipulator configuration stays on the failure surface, it is guaranteed that the goal end-effector pose can be reached after any failure occurs. A measure for global failure tolerance is derived based on the joint range excursion in the self-motion manifold. The maximization of the size of the self-motion manifolds was recently proposed in Alkmarkhi and Maciejewski (2019), by using singularities to identify where large manifolds might exist.

In summary, previous works have identified the necessary conditions under which a redundant manipulator can fulfill a task after a joint failure, exploiting links between dexterity, manipulability and robustness to failures. The methods are commonly based on the analysis of pre-image and self-motion manifold estimation using discrete sampling and off-line computations of certain aspects of the algorithms (to overcome the high computational costs). The constraints derived from such analysis are used for path verification and planning. Several measures of robustness to failure were introduced. Obstacle avoidance, if addressed, relies on projection of primitives in the C-space. The path-planning approaches are typically augmented with discrete sampling or with a null-space controller. The analysis of failure robustness is performed mainly on a case by case basis. Some works have considered as well a global overview, as the one provided here, but generic considerations of obstacles in the environment and coupling of the methods with path planning algorithms are scarce in previous works.

This paper builds upon the necessary conditions for existence of a fault-tolerant path and presents a new approach to fail-safe path planning and task design. Our framework is based on a global workspace overview, which allows a direct link between the end-effector pose and its pre-image bounds in C-space. Not relying on local exploration, all self-motion manifolds (which compose the pre-image) for fault-tolerant path planning are revealed, regardless of the current joint configuration. Pre-image bounds are pre-computed in the offline phase and used to generate a model that provides information about robot redundancy within the workspace and the effects of failures on the end-effector positioning ability. This information is later used to provide the required constraints for the on-line fault-tolerant path planning process. Thus, the method is applicable to arbitrary kinematic designs and to large DoF systems without further adaptation. The environment around the manipulator (e.g. the satellite body) can also be considered in the workspace model and in the computation of the self-motion manifolds. For this purpose, collision detection does not rely on C-space projection but it uses directly the geometry of the environment, therefore, it is possible to use any common collision detection method within the approach. Our implementation uses a point-to-primitive check, which makes the method suitable for online applications utilizing depth sensor input directly. The framework is applied to different scenarios that demonstrate the potential of the proposed approach.

2 REACHABILITY AND CAPABILITY MAPS

A redundant manipulator has at least one DoF more than the number of DoF required for the task at hand. In this paper, a set of joint values for the manipulator is denoted as $\theta \in \mathbb{R}^{N_{\text{dof}}}$ (with N_{dof} representing the number of joints), which corresponds to a point in the C-space for the manipulator. Joint values θ are bounded by a set of joint limit pairs denoted as $Q = \{\theta_{\min}, \theta_{\max}\}$. A forward kinematics function maps the C-space configuration θ to a task space pose $\xi \in SE(3)$. An inverse kinematics function maps $\xi \rightarrow \theta$, and it typically provides multiple solutions for redundant



manipulators. As there can be many θ mapping to one particular ξ , self-motion manifolds are present, as discussed in Burdick (1989).

The workspace \mathcal{W} of a manipulator refers to every possible ξ that can be reached with a configuration θ within the joint limits Q . The workspace \mathcal{W} can be represented in a discrete way through the reachability map. It captures all poses reachable by the robot's end-effector frame with respect to the robot base. These maps were initially introduced in Zacharias et al. (2007) and Diankov (2010), and have been used in multiple applications including workspace analysis and robot design in Porges et al. (2015), pre-filtering of inverse kinematics queries in Porges et al. (2014), robot base positioning as in Zacharias et al. (2009b); Vahrenkamp et al. (2013), bi-manual manipulation planning in Sundaram et al. (2016); Vahrenkamp et al. (2009), path generation or validation in Zacharias et al. (2009a) and humanoid robot foot-step planning in Werner et al. (2016).

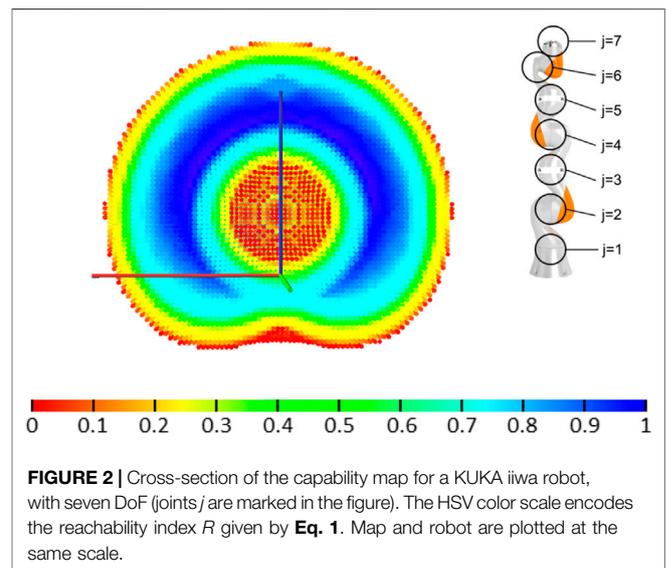
To obtain the reachability map, the workspace is discretized in a hierarchy that decomposes the pose $\xi \in SE(3)$ into a translation and a rotation, $\xi \rightarrow (t, R)$, where $t \in \mathbb{R}^3$, $R \in SO(3)$. Considering the Cartesian position $t \rightarrow x, y, z$ and Euler angles $R \rightarrow \alpha, \beta, \gamma$ (roll, pitch, yaw), the mapping functions $f(t) \rightarrow V_i$ and $g(R) \rightarrow v_{i,p}$ discretize the pose ξ using $i, p \in \mathbb{N}$, where V_i is the i -th voxel and $v_{i,p}$ is a particular bin value of voxel V_i . An illustration of the discretization mechanism is presented in Figure 1. The workspace is discretized with voxels of suitable size. Each voxel has an associated virtual sphere that discretizes the end-effector pointing direction (pitch and yaw, β and γ Euler angles) within each voxel. Each discretized pair of β, γ has an associated grid for discretizing the roll angle α . The reachability map stores binary values for each bin, where a bin represents a small range of end-effector poses in $SE(3)$: The stored value is one for reachable or zero for unreachable bins. The level of detail of a reachability map is then represented by three quantities, voxel size (e.g. 0.025m), number of approach directions (e.g. 200) and number of roll bins (e.g. 30, thus mapping $200 \times 30 = 6,000$ orientations in each voxel).

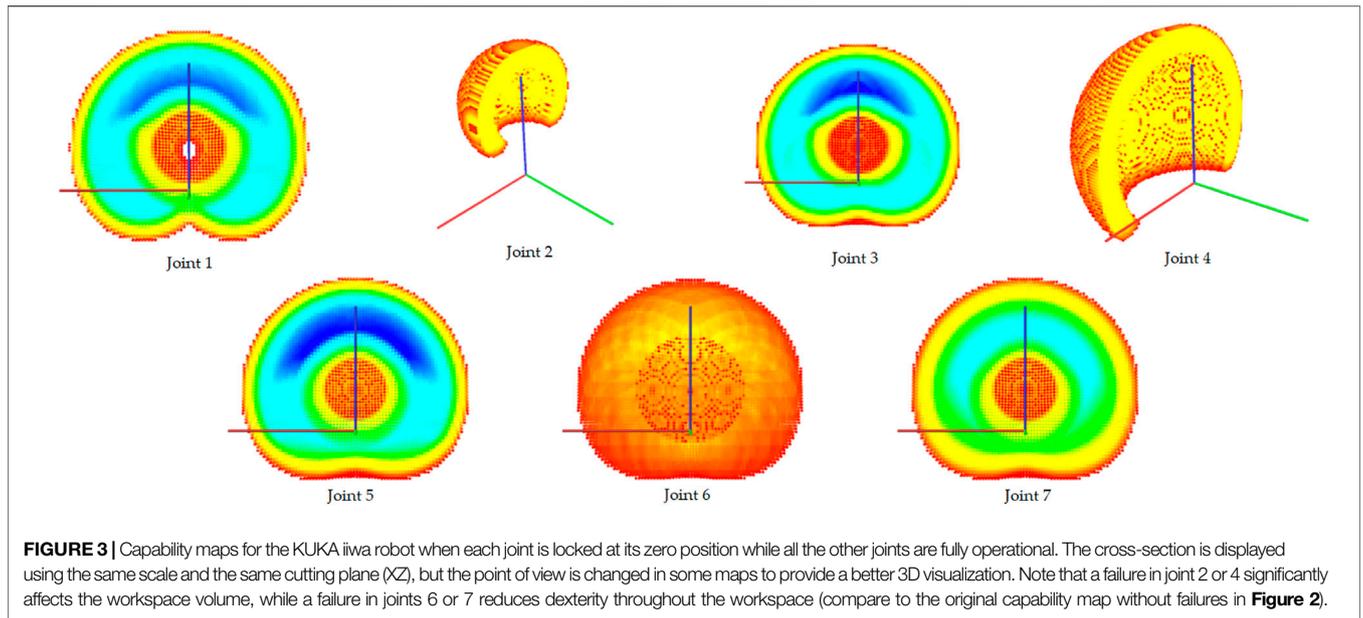
This paper exploits two fundamental functionalities of reachability maps: pose existence queries and interpreter functions. A query to a map is denoted as $\mathcal{W}(\xi)$, and performs the mapping of pose ξ into a bin and retrieves its binary value, which indicates whether the pose is reachable or not. The speed of this query process is what makes the reachability maps suitable for on-line applications on robots. The current implementation from Porges et al. (2014) handles typically about 2000 queries per millisecond. Interpreter functions help to visualize the workspace model data based on a desired property. A commonly used interpreter function called reachability index R summarizes the results for all discretized orientations inside a voxel in one metric, thus enabling the workspace visualization. This index is defined by

$$R(V_i) = \frac{\sum_{p=0}^{n_b} v_{i,p}}{n_b} \quad (1)$$

Where n_b is the number of all discretized orientations inside each voxel (approach directions times roll directions). The range of the reachability index is $R(V_i) \in [0, 1]$, from totally unreachable ($R = 0$) to 100% reachable orientations ($R = 1$) in a voxel.

The reachability index reflects a local dexterity in the task-space, as it indicates the ability of the manipulator to reach different position and orientations of the end effector within the local neighborhood of the current pose. Note that reachability strictly refers here to the ability to reach all possible orientations of the end effector at a given point in the manipulator workspace, as a difference to the manipulability measure derived from the Jacobian condition number in Yoshikawa (1985), which describes the ability of the manipulator to move and apply forces in arbitrary directions, as described in Patel and Sobh (2015). Further comparison on reachability and manipulability is provided in Zacharias (2012). A 3D visualization of voxels with a color-coded reachability value is called a capability map, as illustrated in Figure 2 using a HSV color scale: Regions with low reachability are colored in red, while high reachability regions are colored in blue. To provide a sense of





scale, the capability maps shown in this paper always include a reference frame, with axis of 1m length, depicting the robot base position and orientation. Generation methods, performance and prediction accuracy of these maps are discussed in detail in Porges et al. (2014).

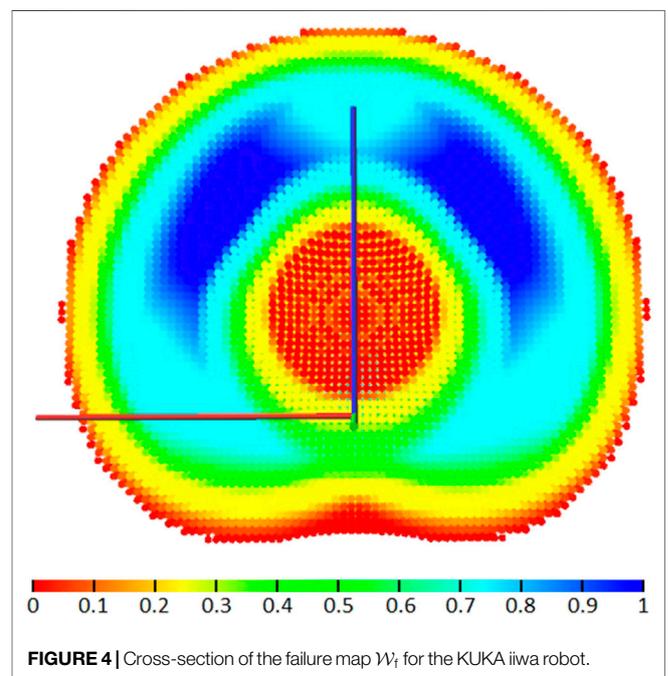
3 FAILURE MAPS

Since the reachability maps depend on the kinematic design of the manipulator, they are usually computed off-line, and the information can later be queried for online processes. In case that an arbitrary joint of the robot is locked, a set \mathcal{F} containing a large number of reachability maps can be generated to represent the effect of the locked joint on the robot workspace. Let $\mathcal{W}_{j,l}$ be a reachability map where l is the locked position of the j -th joint,

$$\mathcal{F} = \{ \mathcal{W}_{j,l} \mid j \in (1, N_{\text{dof}}), l \in (l_{\text{min}}, l_{\text{max}}) \} \quad (2)$$

Where $l_{\text{min}}, l_{\text{max}} \in \mathcal{Q}_j$ are the joint limits of the locked joint. For illustrating the methods in this paper, the 7-DoF KUKA iiwa robot¹ is used as a prototypical example. To generate \mathcal{F} , each joint is locked, one at a time, at different successive positions within the joint limits using a given resolution. This paper uses a resolution of one degree, which leads to generating 2097 reachability maps. The maps are generated using forward kinematics, which provides the least amount of false positive values in prediction of pose reachability, as discussed in Porges et al. (2014).

Joint locks reduce the capabilities of the manipulator. They shrink the post-failure workspace and reduce the manipulator's dexterity. **Figure 3** illustrates the influence of a potential failure at



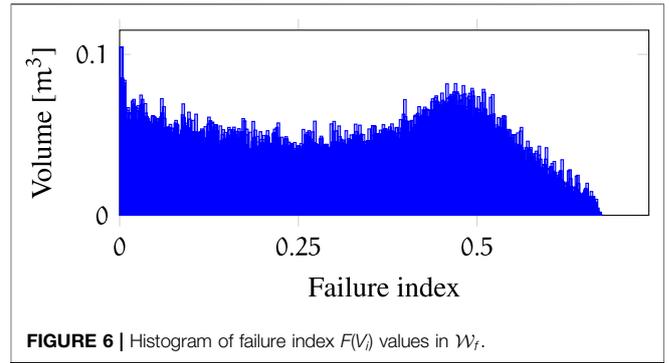
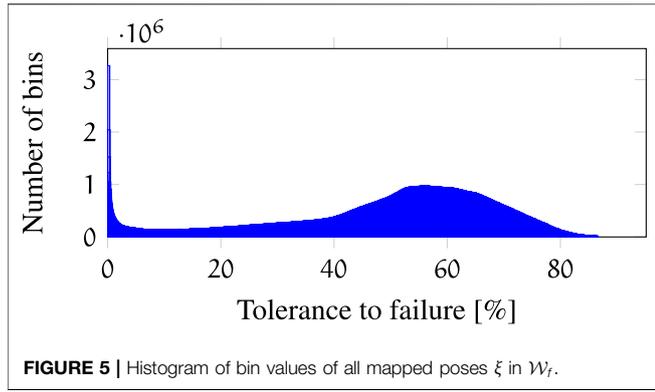
different joints locked at zero position on the workspace of the KUKA iiwa robot.

Some workspace locations are more affected than others by joint failures. To gain a full overview for assessing the risk, workspaces from the set \mathcal{F} are merged into one failure map \mathcal{W}_f by

$$\mathcal{W}_f = \sum_{j=1}^{N_{\text{dof}}} \sum_{l_{\text{min}}}^{l_{\text{max}}} \mathcal{W}_{j,l} \quad (3)$$

Two reachability maps can be merged as described in Sundaram et al. (2016): each bin $v_{i,p} \in \mathbb{N}$ of the resulting map

¹<https://www.kuka.com/en-de/products/robot-systems/industrial-robots/lbr-iiwa>



\mathcal{W}_f holds an integer value reflecting the number of maps from the set \mathcal{F} that identify this particular bin as reachable. A new interpreter function called failure index $F(V_i)$ is computed in an analogue manner to the reachability index, and defined as

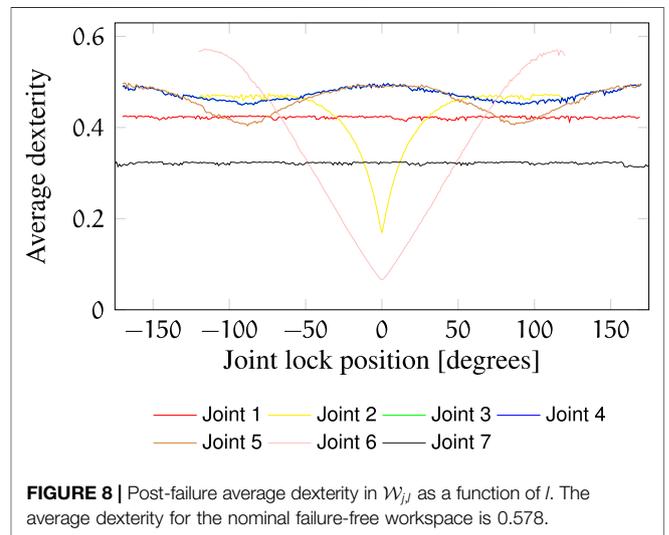
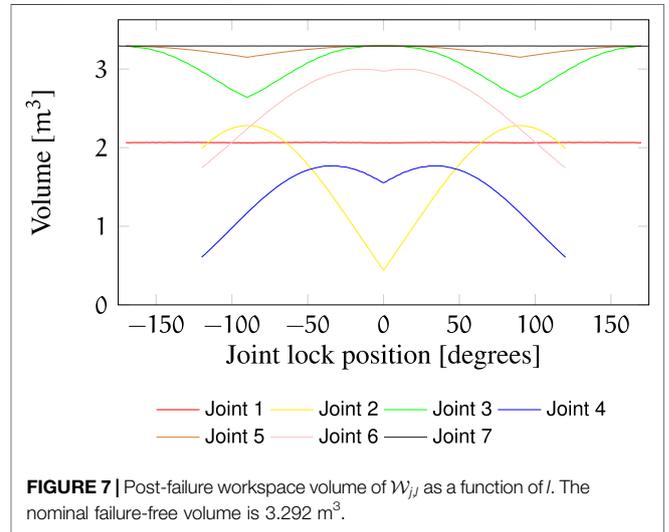
$$F(V_i) = \frac{\sum_{p=1}^{n_b} V_{i,p}^f}{n_m * n_b} \quad v_{i,p}^f = \sum_{k=1}^{n_m} v_{i,p}^{(k)} \quad (4)$$

where $v_{i,p}^{(k)}$ is the p -th bin value of voxel i in reachability map k , and n_m represents the number of maps generated, i.e. the number of points used to discretize the joint ranges Q . **Figure 4** shows the cross-section of the failure map for a KUKA iiwa. With the HSV color scale, red areas are less robust to a general joint failure, while blue areas are reachable in most failure cases. The combined failure map \mathcal{W}_f is used for identifying Cartesian positions of lower and higher risk due to joint failures, so that locations with higher risk are avoided when performing critical operations.

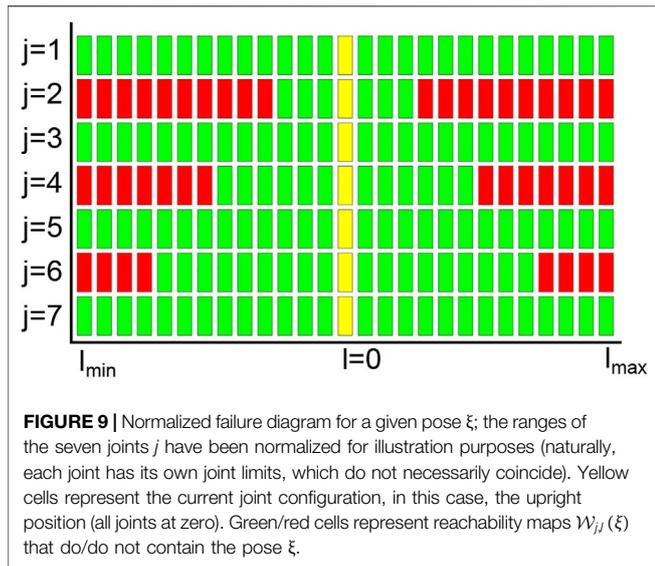
The map in **Figure 4** appears intuitive due to its symmetry, as it is generated for a robot moving in a collision-free environment. For a real use case, the workspace models can be generated considering collisions with the known environment, e.g. the mobile base of the manipulator or the satellite for a space robot arm, as presented in Porges et al. (2015). A small obstacle can cause large reachability reductions away from the obstacle's vicinity.

The failure index defined in (4) summarizes in a single index the influence of the joint failure across all orientations. The index can be tuned to study failures along a specific subset of orientations, or just in a single orientation, by changing the limits of the sum in (4).

Individual bin values in \mathcal{W}_f indicate the robustness of the end-effector pose. The bin value $v_{i,p}$ is calculated as a sum of all maps from \mathcal{F} containing the pose ξ associated with it, therefore, $v_{i,p}$ quantifies the pre-image footprint within Q . In general, the larger the bin value $v_{i,p}$, the more failure-tolerant behavior can be achieved. For example, particular values $v_1 = 101$ and $v_2 = 1622$ mean that the pose associated with v_1 is tolerant to failure in 4.8% of the cases, while v_2 is failure-tolerant in 77.3% of the 2097 discretized cases.



Other measures derived from \mathcal{W}_f help to evaluate the robot kinematics. A histogram of the bin values is presented in **Figure 5**. The maximum bin value in \mathcal{W}_f is 1813 out of 2097 original joint



failure maps, therefore, the maximum possible robustness of a particular pose in the workspace is 86.46% and the highest failure index of a voxel is 0.673. The area under the curve is correlated to the 6-dimensional volume that indicates the portion of task space robust to failures. A histogram of the failure index is presented in **Figure 6**. The area under the curve represents the 3-dimensional volume (correlated to the number of voxels) that indicates the portion of the workspace robust to failures. These histograms are useful for instance to compare different kinematic designs and verify which ones provide larger volumes of high tolerance to failures.

Additional indicators of failure robustness are the post-failure workspace volume (**Figure 7**) and the post-failure average dexterity (**Figure 8**). The post-failure workspace volume shown in **Figure 7** helps to identify joint value ranges in which a joint failure is dangerous. In this case, joint 2 around a joint value of 0 is critical. A cross section of this particular post-failure capability map is shown in **Figure 3**. The post-failure workspace volume has almost no change for failures in joints 1 and 7. On the other hand, the average dexterity, shown in **Figure 8**, is significantly reduced for joints 2 and 6 around the joint value 0. A cross section of the capability map for these failures is shown in **Figure 3**, for the joints locked at zero position. When designing the kinematic structure, these indicators can be used to avoid concentration of risks in one zone of the C-space. How critical are these C-space locations can only be evaluated for a particular task specification.

4 FAILURE DIAGRAMS

Previous sections described a metric for failure robustness, and a global overview of the redundancy of the manipulator kinematics with insight into risks related to failures of single joints. In order to guide the task design or to plan fault-tolerant paths, more

insight into the C-space is required. For this purpose, a new tool called failure diagram, denoted as $D(\xi)$, is introduced. It is a function of the end-effector pose ξ , and can be generated from the set \mathcal{F} . The set \mathcal{F} is queried with additional parameters that specify the joint and its locked position $\mathcal{W}_{j,l}(\xi)$. The failure diagram $D(\xi)$ is defined as

$$D(\xi) = \{\mathcal{W}_{j,l}(\xi) \mid j \in (1, N_{\text{dof}}), l \in (l_{\text{min}}, l_{\text{max}})\} \quad (5)$$

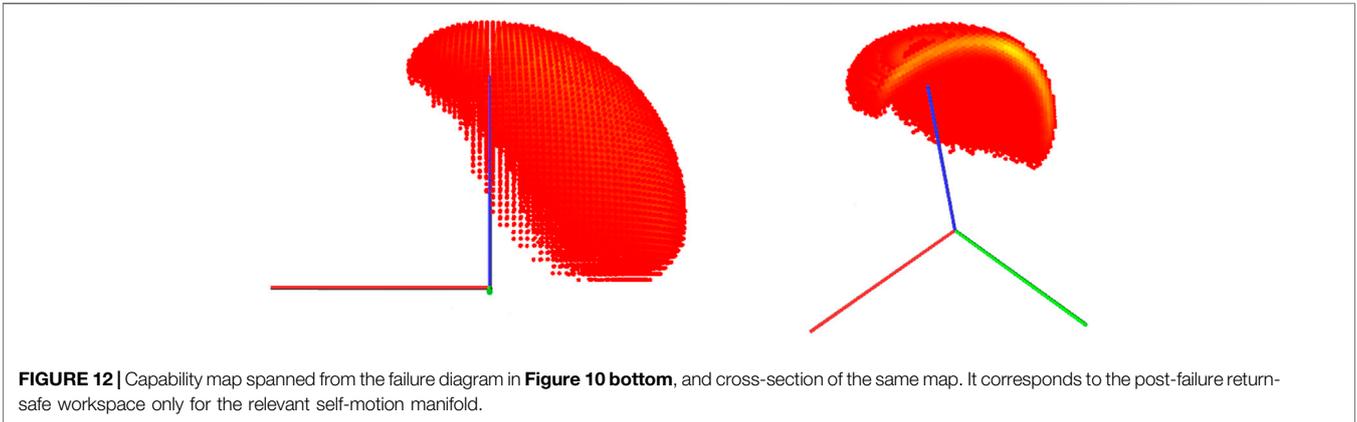
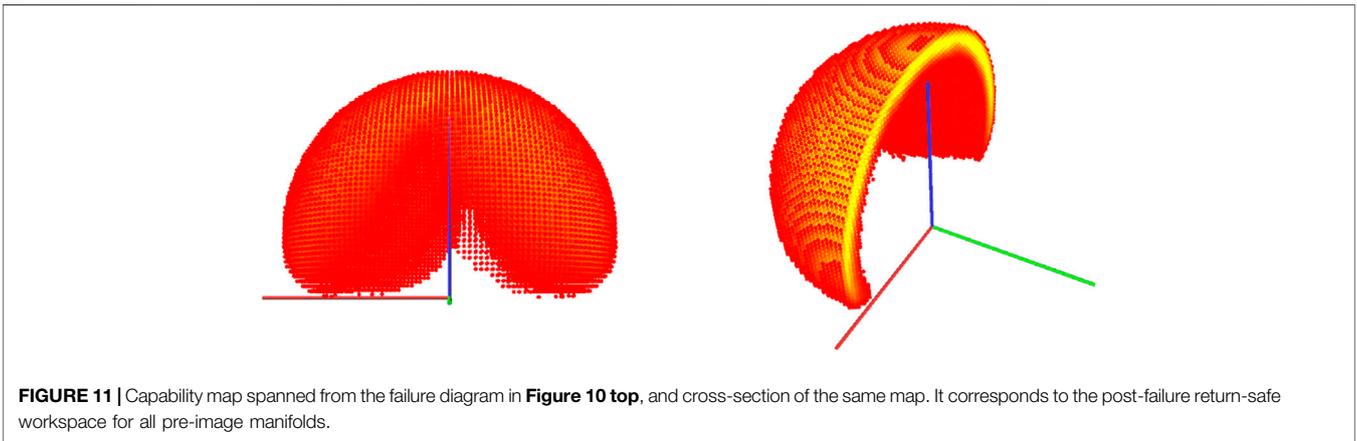
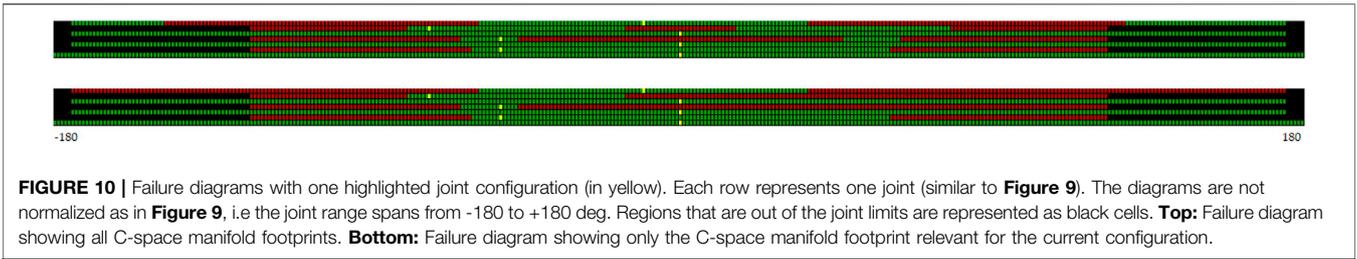
An illustrative failure diagram is presented in **Figure 9**. Each horizontal row represents one joint within its range of motion. Every colored cell corresponds to a reachability map $\mathcal{W}_{j,l}$ with j, l according to the plot axes. The color of the cell indicates the reachability of the pose ξ in the referenced reachability map, green for reachable and red for unreachable. The pattern of the diagram shows continuous ranges of green and red cells in each joint, hereafter called allowed and forbidden ranges, respectively. This pattern is the pre-image footprint of ξ in Q . Multiple allowed ranges within one row (joint) indicate disconnected manifolds of the pre-image. To illustrate this point, a typical failure diagram (non-normalized) for the KUKA iiwa is presented in **Figure 10** top.

Number and structure of distinct self-motion manifolds within a pre-image of a redundant manipulator are examined in Burdick (1989). Two self-motion manifolds are either (C-space) disconnected or connected through a singularity. Each self-motion manifold is continuous in its own. As a result, if a joint is moved within its allowed ranges on the diagram $D(\xi)$, it is guaranteed that the pose ξ will be post-failure reachable. If any joint value falls outside the allowed ranges and a failure occurs there, the pose of $D(\xi)$ is guaranteed to be unreachable. Transitioning of one joint value between two allowed ranges is, therefore, not possible. The combination of allowed ranges (one for each joint) is not arbitrary. A valid combination of allowed ranges is determined by the existence of joint values θ for pose ξ that lie within these ranges, i.e. the allowed range selection is derived from the pre-image of ξ . This selection process is illustrated in **Figure 10**. The joint values in **Figure 10** (top, indicated in yellow) are from the pre-image of ξ for which this diagram was generated. Based on this robot configuration, a selection of allowed ranges for fail-safe operation is shown in **Figure 10** (bottom). Note that this failure diagram is obtained starting from capability maps, which have some chosen discretization level. In our case, we are using a voxel size of 0.025 m, with 200 approach directions and 30 roll bins. The failure diagram is then the pre-image of all the poses that fall into the bin containing the desired pose ξ . This tolerance to reaching the desired pose is of course task-dependent, and should be set by the user. To increase numerical accuracy, a finer resolution for discretization should be used, which leads to higher computational costs.

The reachability map arising from $D(\xi)$ is denoted by $W(D(\xi))$. The generation pipeline is described by

$$\theta \rightarrow \xi \rightarrow D(\xi) \rightarrow W(D(\xi)) \quad (6)$$

Given a desired configuration θ , the pose ξ is computed. The failure diagram $D(\xi)$ for this pose is later used to generate the



reachability map $W(D(\xi))$. To illustrate the pipeline, **Figure 11** shows $W(D(\xi))$ generated from the failure diagram of **Figure 10 top**. It contains all poses of the end-effector considered fail-safe with respect to the pose associated with the diagram $D(\xi)$. These poses can be reached only if the robot configuration stays within a set of allowed joint ranges (one for each joint). **Figure 12** shows the associated reachability map $W(D(\xi))$ for those poses, which are a subset of **Figure 11**. For practical reasons, the design process described in **Eq. 6** starts from configuration θ .

During task design, we start with an arbitrary inverse kinematics solution for reaching the desired pose ξ . Failure diagram $D(\xi)$ reveals all allowed ranges, and the robot configuration identifies the valid ranges of operation for each joint. $W(D(\xi))$ spanned from these identified ranges describes

the task space for fail-safe operations. If a different combination of allowed ranges is desired (e.g. to move to a range with larger footprint in Q), the existence of such θ that selects the desired ranges has to be verified. Note that a post-failure path existence is guaranteed from an arbitrary pose $\xi_a \in W(D(\xi)), \theta \in D(\xi)$ to ξ and not vice-versa. The spanned workspace $W(D(\xi))$ is therefore called post-failure return-safe (*pfrs*) workspace.

Post-failure reachability can be guaranteed for multiple poses simultaneously. Let us have two end-effector poses ξ_1 and ξ_2 and their respective failure diagrams $D(\xi_1)$ and $D(\xi_2)$. An operator $D(\xi_1) \& D(\xi_2)$ is defined analogically to the logical operator $\&$, where green cells represent the logical value one and red cells represent the value zero. The result of the operation $\&$ is a merged failure diagram $D(\xi_1 \& \xi_2)$. A failure-tolerant path from ξ_1 to ξ_2

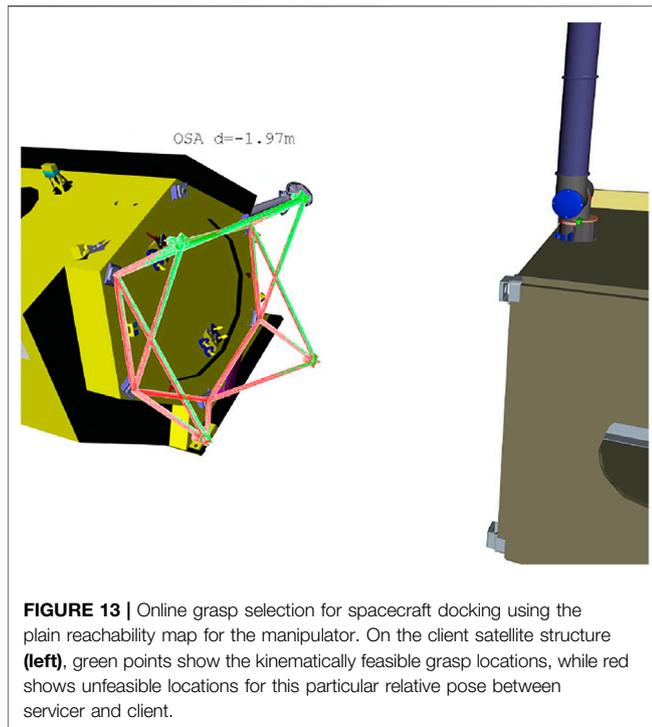


FIGURE 13 | Online grasp selection for spacecraft docking using the plain reachability map for the manipulator. On the client satellite structure (left), green points show the kinematically feasible grasp locations, while red shows unfeasible locations for this particular relative pose between servicer and client.

exists if the diagram $D(\xi_1 \& \xi_2)$ contains an allowed range for each joint. Similar to the previous case, a new reachability map is spanned as $W(D(\xi_1 \& \xi_2))$. Such a map represents a safe workspace to both poses ξ_1 and ξ_2 , which is significantly smaller with respect to the individual *pfirs* workspaces $W(D(\xi_1))$ and $W(D(\xi_2))$. If the robot configuration stays within the boundaries of $D(\xi_1 \& \xi_2)$, ξ_1 and ξ_2 are guaranteed to be post-failure reachable, i.e., a path between ξ_1 and ξ_2 will exist after any single joint failure.

5 APPLICATIONS

To demonstrate the design and planning capabilities of the presented tools, three different use cases are discussed: online grasp selection, path planning, and base positioning for a robot.

5.1 Online Grasp Selection

The increasing amount of orbital debris is a major concern for space missions nowadays. Many dysfunctional satellites are occupying valuable orbits, and potential collisions with them might create more dangerous debris. Robotic manipulators could be the key technology to enable de-orbiting of uncooperative targets like ENVISAT, as discussed in Jaekel et al. (2015). On-orbit servicing is a promising technology to try to keep the orbits clean. Manned on-orbit service missions helped to keep the Hubble telescope afloat and functional, however, at a very high cost. Robotic manipulators could provide the means of servicing and refueling satellites at a reasonable cost, thus prolonging their lifetime in orbit. In a de-orbiting or servicing mission for satellites, a servicer satellite chases a target, or client satellite. The goal is to capture the target with the manipulator

and perform a docking maneuver with the servicer. The target is not necessary cooperative, as it can be tumbling at high rotational velocities. High tumbling velocity and/or significant mass of the target lead to a high risk of damaging the manipulator on contact.

Reachability maps were previously used for selecting a grasp on the target satellite structure such that the docking maneuver is feasible, as presented in Porges et al. (2015). The capability map is used for obtaining a set of feasible grasps on the target structure. **Figure 13** shows for instance the bar structure mounted on the TerraSAR-X satellite that should be grasped to pull the satellite for docking with the servicer.

Now, the method can be enhanced using the tools presented here such that the chosen grasp pose is robust to joint failures. The process described in (6) is followed, i.e., the failure diagram $D(\xi)$ of the desired end-effector pose for the docking maneuver is computed. The *pfirs* workspace $W(D(\xi))$ using only ranges of failure diagram identified by θ is generated. Now, with $W(D(\xi))$ we can choose the best grasp location on the structure. Furthermore, the grasp selection can include the robustness-to-failure as a quality measure, utilizing the bin values $v_{i,p}$ from W_f . The grasp selected by maximizing the $v_{i,p}$ value is the one that maximizes the *pfirs*. After a successful grasp, path planning procedures robust to failures can be applied, as described later in Section 5.2, to ensure that the docking maneuver is performed safely. Task planning modules are forced to only assume grasps within the safe workspace, thus guaranteeing post-failure docking feasibility.

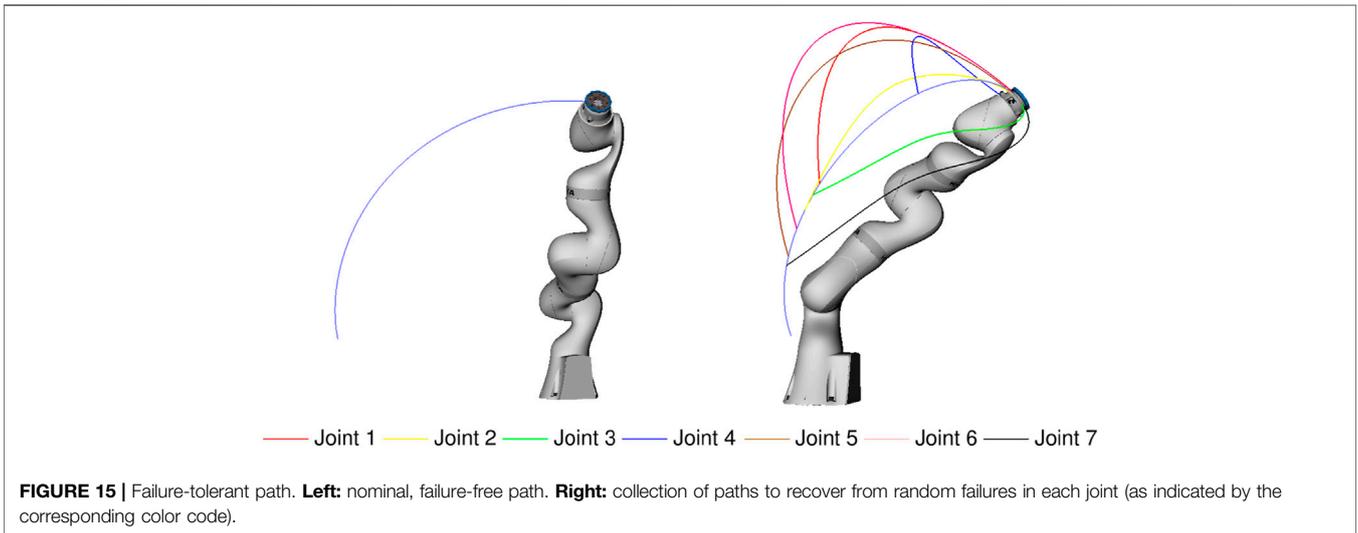
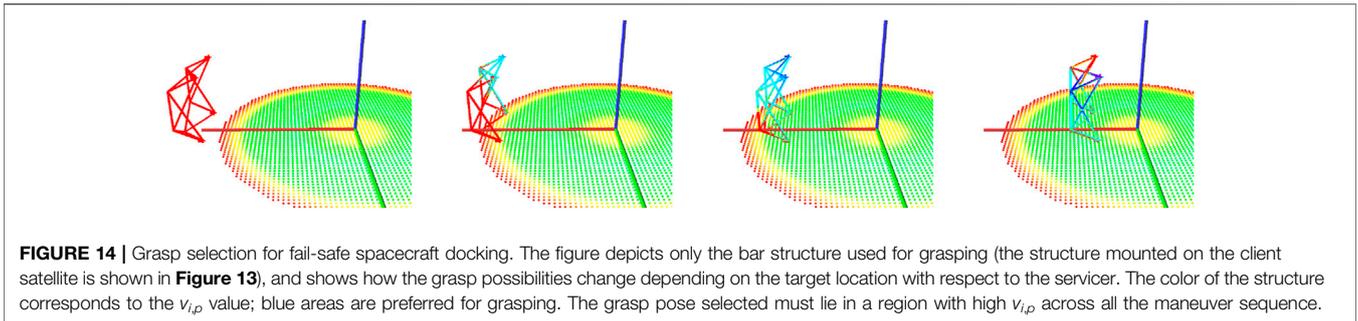
The use of the combined failure map is depicted in **Figure 14**. The query poses are defined by pre-computed grasps on the target surface and passed through the *pfirs* workspace to obtain kinematically feasible and fail-safe grasp locations.

5.2 Path Planning

The necessary conditions for the existence of unidirectional and bi-directional fault-tolerant paths between two particular configurations were introduced in Section 4. Assuming that these conditions are met, a failure diagram can be used to generate a fault-tolerant motion and post-failure recovery path to reach the desired goal.

Given an initial and a desired goal, the first step is then verifying that there is a fail-safe tolerant path between them, as explained in Section 4. If this is the case, then a path can be obtained using any traditional path planning method, for instance, Rapidly-exploring Random Trees (RRT), which relies on random sampling of the C-space (LaValle et al., 2001). The only condition required to integrate the failure diagrams into the RRT planning approach is to use the bounds provided by the failure diagram (e.g. the bounds in **Figure 10** bottom) for retrieving samples for the RRT process. In other words, each random joint position can only be retrieved from the joint ranges provided by the failure diagram for the desired end effector pose. This integration leads to a failure-tolerant RRT variant, as samples entering the RRT building process are inherently fail-safe.

On the onset of failure, the instantaneous joint configuration lies within the failure diagram, and the existence of a path to the goal pose is guaranteed. To obtain such path, the RRT planning



process can proceed further by randomly sampling the remaining $N_{\text{dof}} - 1$ joints using still the joint ranges provided by the failure diagram; the locked joint does not change its value in the successive robot motions.

An example of this approach is shown in **Figure 15**. A goal pose is defined, and a random feasible configuration is chosen as starting configuration. We could have used an RRT to plan the path between both poses; for simplicity, we used a simple joint interpolator to execute the path between the start and the goal configurations, respecting the joint bounds given by the failure diagram. This nominal path is visualized in **Figure 15**. Several random failures are simulated, one per each joint, during the path execution. For each one of them, we use an RRT planner for the remaining active joints to find an after-failure path to achieve the goal pose ξ , as shown in **Figure 15** and in the attached video. As a result, the end-effector travels within the associated *pfrs* workspace depicted in **Figure 12**, and the end effector pose is achieved, within the tolerances given by the discretization used to build the required reachability maps.

5.3 Task Design and Base Positioning

A robotic manipulator mounted on a planetary rover allows tasks such as collection of soil samples, self-maintenance, or image-

based diagnostics with an in-hand camera. Taking samples usually requires scratching and drilling on a rock or into the ground. Once samples have been collected, they need to be transported for devolatilization to a number of scientific instruments, and optionally to the storage area for later retrieval. Additional constraints on required end-effector poses can be given by actions such as visual self-inspection, where a predefined camera pose is desired.

The application of reachability maps for defining the mounting point of a manipulator on the rover structure was presented in Porges et al. (2015). Deciding the mounting point for the arm is a one-time task. However, if we use the failure maps considering the effect of collisions with the body on the resulting manipulator workspace, we can obtain a configuration that guarantees the execution of the tasks even in the case of a joint failure. The poses for the on-board instruments and for taking diagnostic pictures can be chosen based on bins with high values $v_{i,p}$. Maximizing $v_{i,p}$ also maximizes the *pfrs* workspace that spans from the associated poses. Bi-directional fail-safety for post-failure reachability can be guaranteed by using fused failure diagrams. The *pfrs* workspace can also be used to position the rover with respect to the task by following the inversion of reachability for base positioning introduced in Vahrenkamp et al. (2013). This

workspace inversion is done only once, hence, it is not computationally demanding at the time of application. Thus, the method would provide a full 6-DoF base placement solution with respect to the task.

6 DISCUSSION

This paper demonstrated how the concept of reachability maps is applied for analysis and design of fail-safe operations of robotic manipulators. Using failure maps, general dangerous zones can be identified both in task space (with *pfrs* workspaces) and C-space (with failure diagrams), and zones with high pre-failure redundancy can be selected for critical operations.

The reachability maps are based on a discretization of the workspace; its prediction accuracy was previously studied in Porges et al. (2014). In the case of failure diagrams, which is the crucial component for the computation of a failure-tolerant path, the chosen discretization must be considered carefully, as it also defines the tolerance allowed for reaching the desired end effector pose after a failure occurs. For the computation of failure diagrams, it is suggested to generate the reachability maps using forward kinematics, since it leads mostly to false positives on the boundary of the real workspace, as explained in detail in Porges et al. (2014). After the generation, the allowed ranges in the failure diagram (green cells in **Figure 10** top) can be reduced by one bin on each side of the range to obtain conservative bounds for the fail-safe paths.

The computational requirements for the initial generation phase depend on the workspace volume (to the third power), as explained in Porges et al. (2014). In the case of KUKA iiwa, which was the robot used for the examples presented in this paper, approximately 4.5 CPU h (single core) were required for generating each one of the 2097 reachability maps with a locked joint, accounting for a total of approximately 9500 CPU h of a standard desktop computer. Generation of failure diagrams $D(\xi)$ typically takes under 1 s, and spanning a fail-safe workspace $W(D(\xi))$ takes in the order of 5–10 min (volume-dependent). Although this computational cost is very large, this process is entirely performed offline, and the results are directly used for online queries or fail-safe path computations, as presented in Section 5.

The introduction of failure diagrams allows the search of a fault-tolerant path to reach one or multiple target poses. It can also be used to add constraints for established planning and execution methods, to find paths in case that a joint failure occurs. The planning time of fail-safe paths using RRTs

extended with the failure diagrams is not strongly affected by using the failure diagrams, as they only modify the valid ranges for each joint used for the RRT sampling procedure. Therefore, the expected planning times are in the order of seconds, and depend on the particular situation. In general, a fail-safe path is found faster than an unconstrained path, since the search space is reduced by the boundaries imposed to get a fail-safe path.

The presented methods scale to kinematic chains with any number of DoF and joint types without further adaptations. Obstacles known at generation time can be incorporated into the computation of the initial reachability maps, while runtime obstacles can be incorporated according to the selected planning method.

While the presented methods and tools work for single joint failures, they are not easily extensible to two or more simultaneous failures due to the combinatorial increment of possible failure modes, which entails a much higher computational complexity. However, chances of two or more joint failures are slim, due to the redundancy used in the design of the robotic manipulators for space.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors upon request.

AUTHOR CONTRIBUTIONS

OP developed and implemented the method. DL helped to setup the examples with the DLR iiwa. MR co-developed the method.

ACKNOWLEDGMENTS

The authors want to thank Dr. Roberto Lampariello and Dr. Markus Grebenstein for initial discussions regarding the design of failure tolerant systems.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/frobt.2021.710021/full#supplementary-material>

REFERENCES

- Alkmarkhi, A., and Maciejewski, A. A. (2019). Maximizing the Size of Self-Motion Manifolds to Improve Robot Fault-Tolerance. *IEEE Robotics Automation Lett.* 4, 2653–2660. doi:10.1109/LRA.2019.2913994
- Burdick, J. W. (1989). On the Inverse Kinematics of Redundant Manipulators: Characterization of the Self-Motion Manifolds, in IEEE Int. Conf. Robotics and Automation, 264–270. doi:10.1109/ROBOT.1989.99999
- Diankov, R. (2010). Automated Construction of Robotic Manipulation Programs, (Pittsburgh, Pennsylvania: Carnegie Mellon University). Ph.D. thesis. ISBN: 978-1-124-53547-0.
- Ellery, A., Kreisel, J., and Sommer, B. (2008). The Case for Robotic On-Orbit Servicing of Spacecraft: Spacecraft Reliability Is a Myth. *Acta Astronautica* 63, 632–648. doi:10.1016/j.actaastro.2008.01.042

- Jaekel, S., Lampariello, R., Panin, G., Sagardia, M., Brunner, B., Porges, O., et al. (2015). Robotic Capture and De-orbit of a Heavy, Uncooperative and Tumbling Target in Low Earth Orbit, in Workshop on Advanced Space Technologies for Robotics and Automation - ASTRA.
- Jamisola, R. S., Maciejewski, A. A., and Roberts, R. G. (2006). Failure-tolerant Path Planning for Kinematically Redundant Manipulators Anticipating Locked-Joint Failures. *IEEE Trans. Robot.* 22, 603–612. doi:10.1109/tro.2006.878959
- LaValle, S., and Kuffner, J. (2001). Randomized Kinodynamic Planning. *Int. J. Robotics Research.* 20, 378–400. doi:10.1177/02783640122067453
- Lewis, C. L., and Maciejewski, A. A. (1994). Dexterity Optimization of Kinematically Redundant Manipulators in the Presence of Joint Failures. *Comput. Electr. Eng.* 20, 273–288. doi:10.1016/0045-7906(94)90021-3
- Lewis, C. L., and Maciejewski, A. A. (1997). Fault Tolerant Operation of Kinematically Redundant Manipulators for Locked Joint Failures. *IEEE Trans. Robot. Automat.* 13, 622–629. doi:10.1109/70.611335
- Paredis, C. J. J., and Khosla, P. K. (1996). Fault Tolerant Task Execution through Global Trajectory Planning. *Reliability Eng. Syst. Saf.* 53, 225–235. doi:10.1016/s0951-8320(96)00050-6
- Patel, S., and Sobh, T. (2015). Manipulator Performance Measures - a Comprehensive Literature Survey. *J. Intell. Robot. Syst.* 77, 547–570. doi:10.1007/s10846-014-0024-y
- Porges, O., Lampariello, R., Artigas, J., Wedler, A., Borst, C., and Roa, M. A. (2015). Reachability and Dexterity: Analysis and Applications for Space Robotics, in Workshop on Advanced Space Technologies for Robotics and Automation - ASTRA.
- Porges, O., Stouraitis, T., Borst, C., and Roa, M. A. (2014). Reachability and Capability Analysis for Manipulation Tasks, in ROBOT2013: First Iberian Robotics Conference (Berlin, Germany: Springer), 703–718. doi:10.1007/978-3-319-03653-3_50
- Ralph, S. K., and Pai, D. K. (1999). Computing Fault Tolerant Motions for a Robot Manipulator, in IEEE Int. Conf. Robotics and Automation, 486–493. doi:10.1109/ROBOT.1999.770024
- Ralph, S. K., and Pai, D. K. (1997). Fault Tolerant Locomotion for Walking Robots, in IEEE Int. Symp. Computational Intelligence in Robotics and Automation, 130–137. doi:10.1109/CIRA.1997.613849
- Sundaram, A. M., Porges, O., and Roa, M. A. (2016). Planning Realistic Interactions for Bimanual Grasping and Manipulation, in IEEE-RAS Int. Conf. Humanoid Robots, 987–994. doi:10.1109/humanoids.2016.7803392
- Vahrenkamp, N., Asfour, T., and Dillmann, R. (2013). Robot Placement Based on Reachability Inversion, in IEEE Int. Conf. Robotics and Automation, 1970–1975. doi:10.1109/icra.2013.6630839
- Vahrenkamp, N., Berenson, D., Asfour, T., Kuffner, J., and Dillmann, R. (2009). Humanoid Motion Planning for Dual-Arm Manipulation and Re-grasping Tasks, in IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 2464–2470. doi:10.1109/iros.2009.5354625
- Werner, A., Henze, B., Rodriguez, D. A., Gabaret, J., Porges, O., and Roa, M. A. (2016). Multi-contact Planning and Control for a Torque-Controlled Humanoid Robot, in IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 5708–5715. doi:10.1109/iros.2016.7759840
- Yoshikawa, T. (1985). Manipulability of Robotic Mechanisms. *Int. J. Robotics Res.* 4, 3–9. doi:10.1177/027836498500400201
- Zacharias, F., Borst, C., and Hirzinger, G. (2007). Capturing Robot Workspace Structure: Representing Robot Capabilities, in IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 3229–3236. doi:10.1109/iros.2007.4399105
- Zacharias, F., Borst, C., and Hirzinger, G. (2009a). Online Generation of Reachable Grasps for Dexterous Manipulation Using a Representation of the Reachable Workspace, in IEEE Int. Conf. Advanced Robotics, 1–8.
- Zacharias, F. (2012). *Knowledge Representations for Planning Manipulation Tasks*, Vol. 16. Berlin, Germany: Springer. doi:10.1007/978-3-642-25182-5
- Zacharias, F., Sepp, W., Borst, C., and Hirzinger, G. (2009b). Using a Model of the Reachable Workspace to Position mobile Manipulators for 3-d Trajectories, in IEEE-RAS Int. Conf. Humanoid Robots, 55–61. doi:10.1109/ichr.2009.5379601

Conflict of Interest: OP was employed by the company Agile Robots

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2021 Porges, Leidner and Roa. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.