Predicting Fault-Tolerant Workspace of Planar 3R Robots Experiencing Locked Joint Failures Using Mixture Density Networks

Charles L. Clark and Biyun Xie

***Abstract*— Currently, there are two existing methods to compute the fault-tolerant workspace of a redundant robot arm. However, both of these methods are very computationally expensive. This work proposes using a mixture density network to learn the probability of a rotation angle belonging to the fault-tolerant rotation ranges, and combining rotation angles having high probabilities together to generate the fault-tolerant workspace. Because this method is computationally efficient, it can be used alongside a genetic algorithm to compute the optimal link lengths and artificial joint limits to maximize the area of the fault-tolerant workspace. The predicted fault-tolerant workspace is compared to the actual fault-tolerant workspace, which shows they are both very similar. Finally, the proposed method is used to generate a trajectory that is tolerant to arbitrary joint failures.**

I. Introduction

Robots are well-suited for replacing human workers in hazardous, isolated, and remote tasks, such as nuclear waste cleanup [1], space exploration [2], and disaster relief [3]. Nevertheless, these environments present very challenging conditions, including extreme temperatures, high radiation levels, and unstable structures, which can lead to frequent joint malfunctions. Furthermore, because these environments are inaccessible to humans for repair purposes, ensuring the dependability and resilience of robotic systems requires fault tolerance. One possible approach to achieve fault tolerance is to use kinematically redundant robots, which have more degrees of freedom (DOFs) than are required to accomplish the assigned tasks. However, kinematic redundancy alone is insufficient to guarantee fault tolerance [4], so motion planning algorithms with intelligent optimization before and after arbitrary joint failures must be developed.

The two most common types of tasks are point-to-point tasks, such as pick and place tasks, and trajectory-following tasks, such as arc welding tasks. For point-to-point tasks, fault tolerance can be simply guaranteed by constraining the joints moving inside the bounding boxes enclosing the self-motion manifolds of the target point, which provides a set of artificial joint limits [5]. For trajectory-following tasks, the most efficient way to guarantee fault tolerance is to locate the end-effector trajectory within the fault-tolerant workspace, which is the workspace that can be achieved by the robot both before and after an arbitrary joint failure for a given set of artificial joint limits [6]. Therefore, the robot will be able to complete the entire end-effector trajectory after a failure.

*This work was supported by the NSF Foundation under Grant 2205292.

Charles L. Clark and Biyun Xie are with the Electrical and Computer Engineering Department, University of Kentucky, Lexington, KY 40506, USA landon.clark@uky.edu, biyun.xie@uky.edu

The key problem is how to compute fault-tolerant workspace for a given set of artificial joint limits, which is not an easy job even for planar 3R robots. There are two existing methods. For the first method developed in [7], the conditions of pre-failure workspace boundaries and post-failure workspace boundaries are first identified, and then final fault-tolerant workspace boundaries are obtained by taking the intersections of all the curves. Based on this method, a gradient ascent method is applied to maximize the failure tolerant workspace area [8]. The concept of fault-tolerant workspace is further extended to reliability maps for probabilistic guarantees of task completion [9]. The other method of computing fault-tolerant workspace is discretizing the half plan whose normal is perpendicular to the rotation axis of the first joint, and then determining the rotation range of the first joint to guarantee all the associated workspace positions are within the fault-tolerant workspace [10].

It can be seen that both of the above existing methods are very computational expensive. For the first method, it is numerically challenging to compute the intersections of all potential boundaries. For the other method, the procedure of identifying the fault-tolerant rotation range of the first joint needs to be repeated for each sampled cell in the workspace. In this article, the problems of efficiently computing fault-tolerant workspace of planar 3R robots and design optimally fault-tolerant planar 3R robots and artificial joint limits are studied. The main contributions of this paper are as follows: (1) a new method based on mixture density networks is developed to compute the fault-tolerant workspace of planar 3R robots. (2) A genetic algorithm is applied to identify the optimal robot kinematic structure parameters and artificial joint limits to maximize the fault-tolerant workspace area.

II. Background on Computing Fault-Tolerant Workspace

For each joint i, its artificial joint limits A_i is defined as $A_i = [\underline{a}_i, \overline{a}_i]$, so the pre-failure configuration space is $\boldsymbol{C}_A = A_1 \times \cdots \times A_n$, where n is the number of joints. The pre-failure workspace $\boldsymbol{W_0}$ can be computed as $\boldsymbol{W_0} = f(\boldsymbol{C}_A)$, where f is the forward kinematics function. If joint i is locked at $q_i = \theta_i$, where $\theta_i \in [\underline{a}_i, \overline{a}_i]$, the artificial joint limits of the remaining joints are released, so the post-failure configuration space is given by ${}^i\boldsymbol{C}(\theta_i) = \{\boldsymbol{q} \in \boldsymbol{C} | q_i = \theta_i\}$. Therefore, the post-failure workspace $\boldsymbol{W_i}$, which is defined as the reachable workspace after joint i is locked at any angle between its artificial joint limits, is given by $\boldsymbol{W_i} = \bigcap_{\underline{a}_i \leqslant \theta_i \leqslant \overline{a}_i} f({}^i\boldsymbol{C}(\theta_i))$. Finally, the fault-tolerant workspace $\boldsymbol{W_F}$, which is the reachable workspace locations both before and after an arbitrary failure, is defined as $\boldsymbol{W_F} = \bigcap_{i \in \boldsymbol{F} \cup 0} \boldsymbol{W_i}$, where $\boldsymbol{F}$ is a set of the locked joints.

A general method of calculating the fault-tolerant workspace for a given set of artificial joint limits is discretizing a half-plane into equal square grids where the normal of the half-plane is perpendicular to the rotation axis of the first joint. The fault-tolerant rotation range of the first joint $\beta_{\mathbf{x}} = \left[\underline{\beta}_{\mathbf{x}}, \overline{\beta}_{\mathbf{x}}\right]$ is identified for each grid center, and the positional fault-tolerant workspace can be obtained by rotating each grid from $\underline{\beta}_{\mathbf{x}}$ to $\overline{\beta}_{\mathbf{x}}$.

The fault-tolerant rotation range for each grid center is given by

$$\left[\underline{\beta}_{\mathbf{x}}, \overline{\beta}_{\mathbf{x}}\right] = \bigcap_{i=0}^{n} \left[\underline{\beta}_{i}, \overline{\beta}_{i}\right], \tag{1}$$

where $\left[\underline{\beta}_{i}, \overline{\beta}_{i}\right]$ is the ration range after joint i is locked. The pre-failure rotation range $\left[\underline{\beta}_{0}, \overline{\beta}_{0}\right]$ is determined by

$$\left[\underline{\beta}_{0}, \overline{\beta}_{0}\right] = \left[\underline{a}_1 - \overline{\theta}_1, \overline{a}_1 - \underline{\theta}_1\right], \tag{2}$$

where $\underline{a}_1$ and $\overline{a}_1$ are the intersection points of the self-motion manifolds with $\boldsymbol{C}'_A$, which is the $\boldsymbol{C}_A$ with the artificial joint limits on θ_1 released. The rotation range after the first joint is locked, i.e., $\left[\underline{\beta}_{1}, \overline{\beta}_{1}\right]$, can be computed by

$$\left[\underline{\beta}_{1}, \overline{\beta}_{1}\right] = \left[\overline{a}_1 - \Theta_{1max}, \underline{a}_1 - \Theta_{1min}\right], \tag{3}$$

where $[\Theta_{1max}, \Theta_{1min}]$ denotes the range in $\Theta_1 = \bigcup_{\#\text{of SMMs}} [\theta_{1min}, \theta_{1max}]$ that contains A_1. Finally, the rotation range after the other joints are locked, i.e., $\left[\underline{\beta}_{j}, \overline{\beta}_{j}\right]$, $j = \{2, 3, \cdots, n\}$, is given by

$$\left[\underline{\beta}_{j}, \overline{\beta}_{j}\right] = [-\pi, \pi]. \tag{4}$$

III. Predicting the Fault-Tolerant Workspace

A. Predicting Fault-Tolerant Rotation Ranges

It can be seen that the computation of the fault-tolerant rotation range $\beta_{\mathbf{x}}$ is very complicated and time consuming. The proposed method to increase the efficiency of the calculation of the fault-tolerant workspace utilizes a supervised learning technique, known as a mixture density network, to learn the relationship between kinematic parameters of a robot, artificial joint limits, and the corresponding fault-tolerant workspace. As the number of fault-tolerant rotation ranges associated with each workspace location are not necessary the same, mixture density networks are well-applied to this problem because they are designed to model one-to-many relationships.

To apply an mixture density network to this problem, the parameters of a Gaussian Mixture Model (GMM) are learned and used to predict $\beta_{\mathbf{x}}$ values within the fault-tolerant rotation range for an arbitrary workspace location $\mathbf{x}$. The GMM is sampled using the following equation

$$\hat{\beta}_{\mathbf{x}} \sim \sum_{k}^{K} \pi_k(\mathbf{x}) \mathcal{N}(\mu_k(\mathbf{x}), \sigma_k^2(\mathbf{x})) \tag{5}$$

where K represents the number of mixture components in the GMM, $\pi_k(\mathbf{x})$ represents the probability that $\hat{\beta}_{\mathbf{x}}$ belongs to the k^{th} mixture component, $\mu_k(\mathbf{x})$ represents the mean of the k^{th} mixture component, and $\sigma_k^2(\mathbf{x})$ represents the variance of the k^{th} mixture component. Each of the GMM components are computed by the MDN as a function of only the workspace location $\mathbf{x}$ assuming the robot's kinematic parameters and artificial joint limits are kept constant.

The first step in solving for the fault-tolerant rotation ranges is sampling N different $\hat{\beta}_{\mathbf{x}}$ values for each workspace location $\mathbf{x}$. Because several of these samples may be outliers, a difference filter is then applied to remove these values. The difference filter is developed as follows. First, the sampled $\hat{\beta}_{\mathbf{x}}$ values are sorted, and the difference between adjacent samples is taken as follows

$$\Delta\hat{\boldsymbol{\beta}}_{\mathbf{x}} = \{\hat{\beta}_{\mathbf{x}}^{(1)} - \hat{\beta}_{\mathbf{x}}^{(2)}, \ldots, \hat{\beta}_{\mathbf{x}}^{(N-1)} - \hat{\beta}_{\mathbf{x}}^{(N)}\} \tag{6}$$

where N is the number of sampled $\hat{\beta}_{\mathbf{x}}$ values, and $\hat{\beta}_{\mathbf{x}}^{(i)}$ represents the i^{th} largest sample. The mean and variance of these differences are then computed, where these values are denoted as μ_Δ and σ_Δ^2, respectively. This information is then used to determine which sampled $\hat{\beta}_{\mathbf{x}}$ values should be removed. If $|\Delta\hat{\boldsymbol{\beta}}_{\mathbf{x}}^{(i)} - \mu_\Delta|$ is greater than $\alpha \cdot \sigma_\Delta^2$, where $\alpha > 0$ is a filter parameter, then the samples $\hat{\beta}_{\mathbf{x}}^{(i)}$ and $\hat{\beta}_{\mathbf{x}}^{(i+1)}$ are both removed. Conceptually, this represents the removal of outliers, as sampled $\hat{\beta}_{\mathbf{x}}$ values that are much larger or smaller from the other sampled values are removed. This process can be repeated as many times as necessary to ensure the removal of outliers.

Once the outlying samples have been removed, the fault-tolerant rotation ranges are calculated using the remaining $\hat{\beta}_{\mathbf{x}}$ samples. The values of $\Delta\hat{\boldsymbol{\beta}}_{\mathbf{x}}$ are used to determine this range using a process similar to the difference filter. Once again, the mean and variance of $\Delta\hat{\boldsymbol{\beta}}_{\mathbf{x}}$ are computed. However, if $|\Delta\hat{\boldsymbol{\beta}}_{\mathbf{x}}^{(i)} - \mu_\Delta|$ is less than $\gamma \cdot \sigma_\Delta^2$, where $\gamma > 0$ is not necessarily equal to α, then $\hat{\beta}_{\mathbf{x}}^{(i)}$ and $\hat{\beta}_{\mathbf{x}}^{(i+1)}$ are considered to belong to the same rotational range. The boundaries of the fault-tolerant rotation ranges are found when this condition is violated. These boundaries are used to define a set of R fault-tolerant rotation ranges, denoted $\left[\underline{\beta}_{\mathbf{x}}^{(r)}, \overline{\beta}_{\mathbf{x}}^{(r)}\right]$ for $1 \leq r \leq R$.

B. Training Mixture Density Networks

To ensure the mixture density network learns the relationship between kinematic parameters, artificial joint limits, and the resulting fault-tolerant workspace, many different combinations of kinematic parameters and artificial joint limits are used to train the mixture density network. Because this work considers only planar 3R robots, link lengths are the only kinematics parameters necessary to describe each robot's kinematic structure. Accordingly, the fault-tolerant workspace shape is based only on the ratio between the link lengths, so the total sum of the link lengths is kept constant. The first step in collecting the training data is defining a robot by a set of random link length ratios. A random set of artificial joint limits is then defined. Using this combination of kinematic parameters and artificial joint limits, the fault-tolerant rotation ranges are computed for a set of workspace positions along the x-axis. This process is repeated to form a dataset of different combinations of kinematic parameters and artificial joint limits and the resulting fault-tolerant rotation ranges.

After the dataset is created, the mixture density network is trained on the resulting data. The inputs to the mixture density network are the robot's kinematic parameters, its artificial joint limits, and the workspace position along the x-axis for which the fault-tolerant rotation range is being predicted. The outputs of the mixture density network are K GMM mixture components, π_k, μ_k, and σ_k^2, relative to the input parameters. The mixture density network is trained using backpropagation to minimize the following negative log-likelihood error

$$\mathcal{L}(\mathbf{Z},\mathbf{Y}) = -\frac{1}{N}\sum_{i=1}^{N}\log(p(y^{(i)}|\mathbf{z}^{(i)})) \tag{7}$$

where $\mathbf{Z}$ represents the set of input parameters, $\mathbf{Y}$ represents the set of $\beta_{\mathbf{x}}$ values uniformly distributed between the fault-tolerant rotation ranges of the given workspace location, and N represents the total size of the dataset. The probability of $y^{(i)}$ occuring given $\mathbf{z}^{(i)}$ is derived from the equation of the GMM as follows

$$p(y|\mathbf{z}) = \sum_{k}^{K}\frac{\pi_k(\mathbf{z})}{\sqrt{2\pi}\sigma_k(\mathbf{z})}\exp\left(\frac{-\|y-\mu_k(\mathbf{z})\|^2}{2\sigma_k^2(\mathbf{z})}\right). \tag{8}$$

IV. Optimizing Robot Parameters and Artificial Joint Limits

After the above method is developed to efficiently compute fault-tolerant workspace, the problem of optimizing robot kinematic structure parameters and artificial joint limits to maximize the area of the fault-tolerant workspace is studied in this section. The area of the fault-tolerant workspace can be computed as follows

$$\mathcal{A}_F = \sum_{\mathbf{x}\in\mathbf{X}}\sum_{r=1}^{R}\|\mathbf{x}\|\cdot\|\overline{\beta}_{\mathbf{x}}^{(r)} - \underline{\beta}_{\mathbf{x}}^{(r)}\| \tag{9}$$

where $\mathbf{X}$ represents the set of all sampled workspace locations, $\|\mathbf{x}\|$ represents the distance from the origin to the workspace position $\mathbf{x}$, and R represents the total number of predicted rotational ranges at the given workspace position. Conceptually, this formula represents the area obtained by sweeping the grid associated with each workspace location about the fault-tolerant rotation ranges.

Because this optimization objective is very complicated and contains many local minima, the best choice of optimization algorithms is a global optimization method such as a genetic algorithm. To make use of a genetic algorithm, the concepts of population, fitness, crossover, and mutation must be defined. The population in this work is composed of combinations of link length ratios and artificial joint limits. The fitness of each member of the population is defined as the $\mathcal{A}_F$ value computed for that specific member. To perform crossover between link length ratios, the ratios can be added together, then normalized so their sum is one. To perform crossover between two sets of artificial joint limits, the new upper limit of each joint is the average of the two upper limits of each set of joint limits for that joint, and likewise for the lower limits. For the link length ratios, mutation is performed by adding some noise to the link length ratios and normalizing as described above. To mutate a set of artificial joint limits, one of the joints is chosen at random and its artificial joint limits are randomized, making sure to keep the upper joint limit larger than the lower limit.

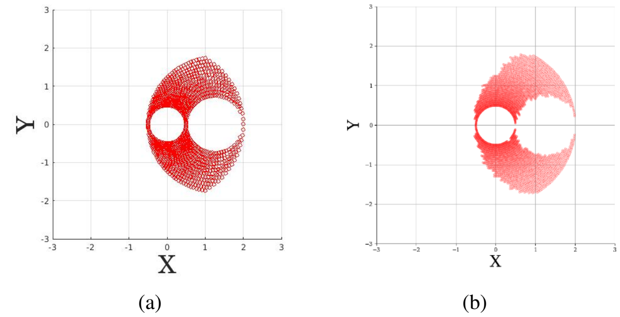
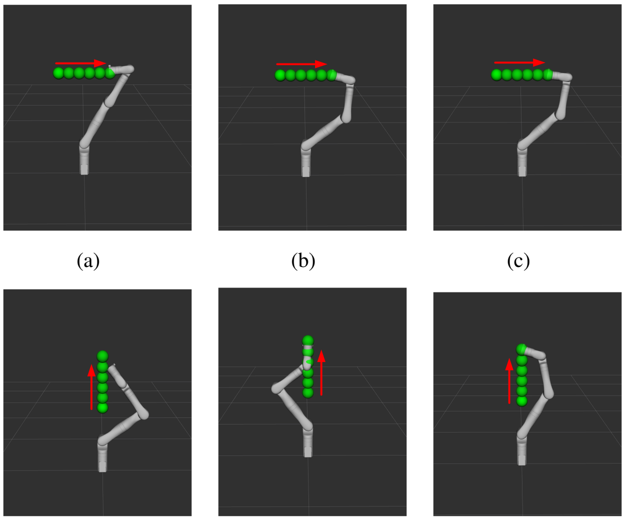


Fig. 1. The actual (a) and predicted (b) fault-tolerant workspaces of the planar 3R robot with link lengths of 1.25m, 0.5m, 1.25m and artificial joint limits of $[0°, 0°]$, $[-180°, 180°]$, and $[-180°, 180°]$ are shown.

V. Results

To validate the ability of the proposed method to produce accurate fault-tolerant workspaces, the fault-tolerant rotation ranges of 150 planar 3R robots, which is a reasonable size for a training dataset, with random link length ratios and constant artificial joint limits of $[0°, 0°]$, $[-180°, 180°]$, and $[-180°, 180°]$ were used to create the training dataset. Because this work focuses on planar 3R robots, the workspace grid is simply the x-axis. Once the mixture density network was trained on this dataset, its performance was analyzed by using it to predict the fault-tolerant workspace of the planar 3R robot with the above artificial joint limits and link lengths of 1.25m, 0.5m, 1.25m. The actual and predicted fault-tolerant workspaces are shown in Fig. 1, where the predicted fault-tolerant workspace is very similar in shape to the actual fault-tolerant workspace. The computational time required to compute the actual fault-tolerant workspace was roughly 60 seconds, while the proposed method required only 0.67 seconds.

The proposed method is also used to compute $\mathcal{A}_F$ as a part of the fault-tolerance optimization process described in Section IV. After being trained on the above dataset, the mixture density network is used to determine the fitness of a set of robots with a randomly generated link length ratios and the same artificial joint limits as the previous example. Because the mixture density network is used during



(a) (b) (c)

(d) (e) (f)

Fig. 2. The comparison of the fault-tolerant and fault-intolerant trajectories is shown. For the horizontal trajectory within the fault-tolerant workspace, the robot is able to complete the trajectory despite failures of joint 2 in (a), joint 4 in (b), and joint 6 in (c). For the vertical trajectory outside of the fault-tolerant workspace, the robot is still able to complete the task is given a failure of joint 6 in (f), but is not able to complete the trajectory given failures of joint 2 in (d) and joint 4 in (e).

each iteration of the genetic algorithm, its computational efficiency greatly impacts the speed of the optimization process. With a population size of 100 robots, a single iteration of the genetic algorithm using the mixture density network takes 20.45 seconds, while it takes roughly 6000 seconds to calculate the area of the fault-tolerant workspace for all of these robots using the method from [10]. The final output of the genetic algorithm is a robot with link lengths of 1.15m, 0.5m, and 1.35m. The first and last links are just 0.1m shorter and longer, respectively, than the analytically-derived solution found in [8] with the given artificial joint limits. This example demonstrates the efficiency and accuracy of the proposed method.

The proposed method is finally validated by placing a trajectory inside of the predicted fault-tolerant workspace, placing another trajectory outside of the fault-tolerant workspace, and demonstrating that the trajectory inside the fault-tolerant workspace is achievable after an arbitrary joint failure, while the other trajectory is not guaranteed to be achievable. This experiment is performed using the 7 DOFs Kinova Gen3 robot arm with joints 1, 3, 5, and 7 locked, which reduces the arm to a planar 3R robot. The proposed optimization method is used to find the optimal the artificial joint limits which maximize the fault-tolerant workspace. The produced artificial joint limits are $[-30°, 30°]$, $[-100°, 100°]$, and $[-100°, 100°]$. Once the fault-tolerant workspace is computed, a trajectory is placed inside of it as shown by the green points in Fig. 2(a)-2(c). It can be seen that the robot is able to complete the entire trajectory after joint 2 is locked in Fig. 2(a), joint 4 is locked in Fig. 2(b), joint 6 is locked in Fig. 2(c). By contrast, a trajectory is placed outside of the fault-tolerant workspace, as shown by the green points in Fig. 2(d)-2(f). Although the task is completed when joint 6 is locked in Fig. 2(f), the task fails when joint 2 is locked in Fig. 2(d) and when joint 4 is locked in Fig. 2(e).

VI. Conclusion

This paper studies the problems of efficiently computing fault-tolerant workspace and identifying the optimal robot kinematic structure parameters and artificial joint limits to maximize the area of the fault-tolerant workspace. It shows that mixture density networks can accurately and efficiently predict the fault-tolerant workspace of planar 3R robots. Therefore, this method can be applied to optimize the artificial joint limits and link lengths of planar 3R robots, which results in an optimal robot with link lengths of 1.15m, 0.5m, and 1.35m for the artificial joint limits of $[0°, 0°]$, $[-180°, 180°]$, and $[-180°, 180°]$. In experiments, a trajectory is placed inside the predicted fault-tolerant workspace, and the robot is able to complete the task for an arbitrary joint failure. The proposed method is a general method that can be applied to robots with many DOFs working in high dimensional workspaces and experiencing multiple joint failures.

References

[1] I. Vitanov, I. Farkhatdinov, B. Denoun, F. Palermo, A. Otaran, J. Brown, B. Omarali, T. Abrar, M. Hansard, C. Oh *et al.*, "A suite of robotic solutions for nuclear waste decommissioning," *Robotics*, vol. 10, no. 4, p. 112, 2021.

[2] Y. Gao and S. Chien, "Review on space robotics: Toward top-level science through space exploration," *Science Robotics*, vol. 2, no. 7, pp. 50–74, 2017.

[3] J. Delmerico, S. Mintchev, A. Giusti, B. Gromov, K. Melo, T. Horvat, C. Cadena, M. Hutter, A. Ijspeert, D. Floreano *et al.*, "The current state and future outlook of rescue robotics," *Journal of Field Robotics*, vol. 36, no. 7, pp. 1171–1191, 2019.

[4] A. A. Maciejewski and R. G. Roberts, "On the existence of an optimally failure tolerant 7r manipulator jacobian," *Applied Math and Computer Science*, vol. 5, no. 2, pp. 343–357, 1995.

[5] C. L. Lewis and A. A. Maciejewski, "Fault tolerant operation of kinematically redundant manipulators for locked joint failures," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 4, pp. 622–629, 1997.

[6] J. Zhao, K. Zhang, and X. Yao, "Study on fault tolerant workspace and fault tolerant planning algorithm based on optimal initial position for two spatial coordinating manipulators," *Mechanism and Machine Theory*, vol. 41, no. 5, pp. 584–595, 2006.

[7] R. C. Hoover, R. G. Roberts, A. A. Maciejewski, P. S. Naik, and K. M. Ben-Gharbia, "Designing a failure-tolerant workspace for kinematically redundant robots," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1421–1432, 2014.

[8] A. M. Bader and A. A. Maciejewski, "Maximizing the failure-tolerant workspace area for planar redundant robots," *Mechanism and Machine Theory*, vol. 143, p. 103635, 2020.

[9] H. Abdi, A. A. Maciejewski, and S. Nahavandi, "Reliability maps for probabilistic guarantees of task motion for robotic manipulators," *Advanced Robotics*, vol. 27, no. 2, pp. 81–92, 2013.

[10] A. M. Bader and A. A. Maciejewski, "A hybrid approach for estimating the failure-tolerant workspace size of kinematically redundant robots," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 303–310, 2020.