2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
October 1-4, 2023, Oahu, Hawaii, USA

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

Charles L. Clark[1], Mohamed Y. Metwly[2], Jiangbiao He[3], Biyun Xie[4]

***Abstract*— There are currently two existing methods to compute the fault-tolerant workspace of a redundant robot arm for a given set of artificial joint limits. However, both of these methods are very computationally expensive. This article proposes using a mixture density network to learn the probability that a rotation angle belongs to the fault-tolerant rotation ranges. A difference filter is used to remove outlying rotation angles predicted by the network, and the remaining rotation angles are grouped together to generate the fault-tolerant workspace. Because this method is highly computationally efficient, it can be used alongside a genetic algorithm to compute the optimal artificial joint limits to maximize the area of the fault-tolerant workspace for a given robot arm. The predicted fault-tolerant workspace is compared to the actual fault-tolerant workspace, which proves the effectiveness of this algorithm. The computational speed of this proposed algorithm is roughly 390 times faster than the traditional method. Finally, a trajectory is placed within the fault-tolerant workspace predicted by the proposed method, and the experimental results show that this trajectory 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 [4]. 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 [5]. However, kinematic redundancy alone is insufficient to guarantee fault tolerance [6], so motion planning algorithms with intelligent optimization before and after arbitrary joint failures must be developed.

*This work was supported by the National Science Foundation under Grant #2205292 as well as NASA and the NASA Kentucky EPSCoR Program under NASA award number 80NSSC22M0034.

[1]Landon Clark is with the Electrical and Computer Engineering Department, University of Kentucky, Lexington, KY 40506, USA landon.clark@uky.edu

[2]Mohamed Y. Metwly is with the Electrical and Computer Engineering Department, University of Kentucky, Lexington, KY 40506, USA mohamed.metwly@uky.edu

[3]Jiangbiao He is with the Electrical and Computer Engineering Department, University of Kentucky, Lexington, KY 40506, USA jiangbiao.he@uky.edu

[4] Biyun Xie is with the Electrical and Computer Engineering Department, University of Kentucky, Lexington, KY 40506, USA biyun.xie@uky.edu

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 (SMMs) of the target point, which provides a set of artificial joint limits [7]–[9]. 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 [10], [11]. Therefore, the robot will be able to complete the entire end-effector trajectory after a failure. The concept of the fault-tolerant workspace is further extended to reliability maps for probabilistic guarantees of task completion [12].

The key problem with the fault-tolerant workspace is the difficulty in calculating it for a given set of artificial joint limits, which is challenging even for simple planar 3R robots. There are two existing methods which solve this problem. For the first method developed in [13], the conditions of pre-failure workspace boundaries and post-failure workspace boundaries are first identified, and then the 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 [14]. The second method of computing the fault-tolerant workspace is discretizing the half plane 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 of the associated workspace positions are within the fault-tolerant workspace [15].

It can be seen that the above existing methods are very computationally 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 the fault-tolerant workspace of a planar 3R robot for an arbitrary set of artificial joint limits as well as determining the optimal artificial joint limits to maximize the area of the fault-tolerant workspace are studied. The main innovations of this paper are as follows: (1) a new, computationally efficient 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 artificial joint limits that maximize the area of the fault-

2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC) | 979-8-3503-3702-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/SMC53992.2023.10394481

tolerant workspace.

The remainder of this article is organized as follows. Section II introduces the background on computing the fault-tolerant workspace by identifying the fault-tolerant rotation range of the first joint. In Section III, a new method is proposed to predict fault-tolerant workspace of planar 3R robots using mixture density networks. In Section IV, a genetic algorithm is applied to compute the optimal link lengths and artificial joint limits for maximizing the fault-tolerant workspace area. The performance of the proposed methods is validated in Section V. Finally, Section VI concludes this article.

II. Background on Computing Fault-Tolerant Workspace

For each joint i, its artificial joint limits A_i are 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 experiences a locked-joint failure and becomes fixed 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 set of 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, as shown in Fig. 1. 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 rotation 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 $\underline{\theta}_1$ and $\overline{\theta}_1$ are the intersection points of the self-motion manifolds with $\boldsymbol{C}'_A$, which is equivalent to $\boldsymbol{C}_A$ 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

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

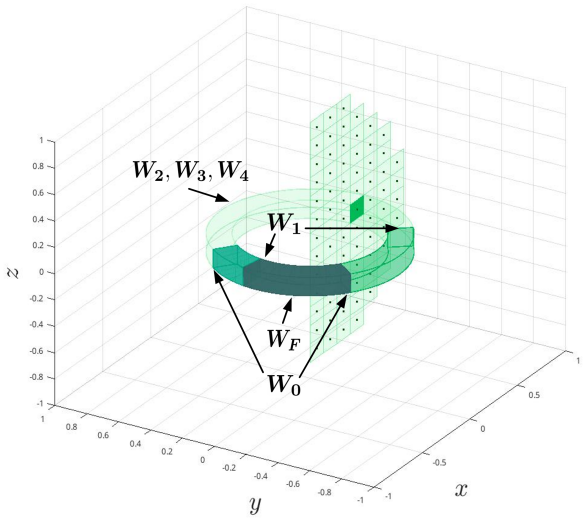


Fig. 1. The fault-tolerant workspace is computed by discretizing a half- plane into equal square grids where the normal of the half-plane is perpendicular to the rotation axis of the first joint. The fault-tolerant rotation range of the first joint is computed for each grid center.

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. This is because the critical points along each of the SMMs belonging to the grid center $\mathbf{x}$ must be computed by an iterative method, and this procedure needs to be repeated for each grid center. A new method is proposed in this section to increase the efficiency of the calculation of the fault-tolerant workspace, which utilizes a supervised learning technique to learn the relationship between artificial joint limits and the corresponding fault-tolerant rotation range of joint one for a given robot.

To predict the fault-tolerant range of joint one, a Gaussian Mixture Model (GMM) is used to represent the distribution of angles within the fault-tolerant rotation ranges for a given grid center $\mathbf{x}$. A GMM can accurately represent an arbitrary number of rotation ranges for each grid center because it is composed of an arbitrary number of normal distributions. First, a mixture density network is used to predict the GMM parameters for an arbitrary workspace location and the predicted GMM is used to sample many potential fault-tolerant rotation angles. Then, these samples are sorted and a difference filter is applied to these sorted rotation angles to remove the outlying rotation angles. Finally, the predicted fault-tolerant rotation angles within the same rotation range

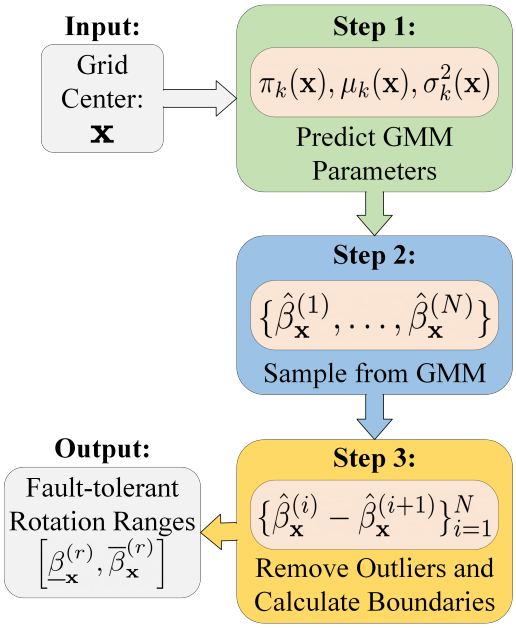


Fig. 2. The flowchart of the proposal method for computing the fault-tolerant rotation ranges of each grid center $\mathbf{x}$ is shown.

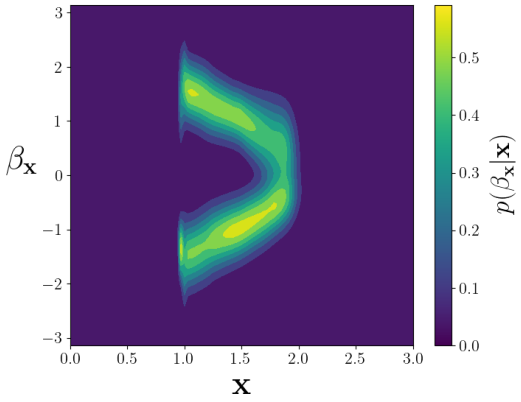


Fig. 3. An example GMM probability distribution indicating the probability that a rotation angle $\beta_{\mathbf{x}}$ is fault tolerant given a grid center $\mathbf{x}$ is shown. The rotation angles with the highest probability of being fault-tolerant are shown as the green and yellow regions.

are grouped together and the lower and upper bounds of the fault-tolerant rotation ranges are identified. The flowchart of the proposed method to compute the fault-tolerant rotation range at a given grid center is shown in Fig. 2.

To predict potential fault-tolerant rotation angles, samples are drawn from the GMM 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 mixture density network as a function of only the workspace location $\mathbf{x}$ for the given robot kinematic parameters and artificial joint limits. This sampling process is repeated to collect N different rotation angle samples for each workspace location $\mathbf{x}$. An example of the probability distributions produced by the GMM for each grid center $\mathbf{x}$ of a planar 3R robot with link lengths of 1m, 1m, and 1m and artificial joint limits of $[-17°, 17°]$, $[-120°, 120°]$, and $[-120°, 120°]$ is shown in Fig. 3. The fault-tolerant rotation angles are represented by the green and yellow areas, which are the regions with high probabilities.

Considering that several of the values sampled from the GMM may be outliers because they are sampled randomly from a probability distribution, a difference filter is then applied to remove these outlying rotation angles. 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, 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 the difference between $\hat{\beta}_{\mathbf{x}}^{(i)}$ and $\hat{\beta}_{\mathbf{x}}^{(i+1)}$ is much larger than the average difference between the other $\hat{\beta}_{\mathbf{x}}$ values, then the values $\hat{\beta}_{\mathbf{x}}^{(i)}$ and $\hat{\beta}_{\mathbf{x}}^{(i+1)}$ are likely to be outliers. In the proposed method, if $\Delta\hat{\beta}_{\mathbf{x}}^{(i)} - \mu_\Delta$ is greater than $\alpha \cdot \sigma_\Delta$, 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. Because many samples that are very close to one another most likely belong to negligible fault-tolerant rotation ranges, the difference filter can also be used to remove these samples. If $\Delta\hat{\beta}_{\mathbf{x}}^{(i)}$ is less than ϵ, where $\epsilon > 0$ is another filter parameter, then the samples $\hat{\beta}_{\mathbf{x}}^{(i)}$ and $\hat{\beta}_{\mathbf{x}}^{(i+1)}$ are both removed. The filtering process can be repeated as many times as necessary to ensure all of the unwanted samples are removed. An example of the difference filter is given in Fig. 4. The unfiltered predicted $\hat{\beta}_{\mathbf{x}}$ values are shown in Fig. 4(a), while the filtered values are shown in Fig. 4(b). This example demonstrates the filters ability to remove outliers and samples that belong to negligible fault-tolerant rotation ranges.

Once the outlying samples have been removed, the remaining samples within the same rotation range are grouped together. This can be accomplished using a procedure similar to the difference filter. Based on the remaining $\hat{\beta}_{\mathbf{x}}$ samples, $\Delta\hat{\boldsymbol{\beta}}_{\mathbf{x}}$ is recomputed along with its mean and variance, μ_Δ and σ_Δ^2. Instead of removing samples that have a large difference between them, these large differences are now used to determine the boundaries of each fault-tolerant rotation range. Therefore, if $\Delta\hat{\beta}_{\mathbf{x}}^{(i)} - \mu_\Delta$ is less than $\gamma \cdot \sigma_\Delta$, 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

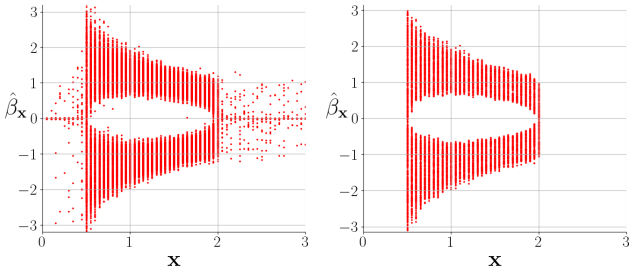


Fig. 4. The predicted $\hat{\beta}_{\mathbf{x}}$ values both before and after the difference filter is applied are shown in (a) and (b), respectively. Very few of the $\hat{\beta}_{\mathbf{x}}$ values remaining after applying the difference filter are outliers or belong to negligible fault-tolerant rotation ranges.

boundaries of the fault-tolerant rotation ranges are found when this condition is violated. These boundaries are used to define the R fault-tolerant rotation ranges for the workspace location $\mathbf{x}$, denoted $\left[\underline{\beta}_{\mathbf{x}}^{(r)}, \overline{\beta}_{\mathbf{x}}^{(r)}\right]$ for $1 \leq r \leq R$.

B. Training Mixture Density Networks

The key component of the above proposed method is to use the mixture density network to learn the relationship between artificial joint limits and the resulting fault-tolerant workspace for a given robot. To achieve this goal, many different combinations of artificial joint limits are used to train the mixture density network. Because the workspace dimension is two, the hyperplane in Fig. 1 is reduced to a line along the x-axis. Therefore, only the workspace locations along the x-axis need to be sampled. The first step in collecting the training data is defining a set of random artificial joint limits. Using these artificial joint limits, the fault-tolerant rotation ranges using the traditional method are computed for a set of workspace positions along the x-axis. This process is repeated to form a dataset of different grid centers, artificial joint limits, and 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 artificial joint limits and the workspace position along the x-axis for which the fault-tolerant rotation range is being predicted for. 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 $y^{(i)}$ occurring 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 the fault-tolerant workspace, the problem of optimizing the artificial joint limits to maximize the area of the fault-tolerant workspace for a given robot is studied in this section. This optimization problem can be formulated as

$$\begin{aligned} \max_{\mathbf{A}} \ & \mathcal{A}_F = \rho \cdot \sum_{\mathbf{x} \in \mathbf{X}} \sum_{r=1}^{R} \mathbf{x} \cdot \left(\overline{\beta}_{\mathbf{x}}^{(r)} - \underline{\beta}_{\mathbf{x}}^{(r)}\right) \\ \text{s.t.} \ & \underline{a}_1 = -\overline{a}_1 \end{aligned} \tag{9}$$

where $\mathbf{A}$ is the set of artificial joint limits, and ρ is the length of the grids along the x-axis. The objective function $\mathcal{A}_F$ is the area of the fault-tolerant workspace. Conceptually, this can be computed by sweeping the grid associated with each workspace location $\mathbf{x}$ about its fault-tolerant rotation range $\left[\underline{\beta}_{\mathbf{x}}^{(r)}, \overline{\beta}_{\mathbf{x}}^{(r)}\right]$. The constraint $\underline{a}_1 = -\overline{a}_1$ is added because the area of the fault-tolerant workspace is affected only by the difference of the limits of joint one.

Because this optimization objective is very complicated and contains many local maxima, the best choice of optimization algorithms is a global optimization method such as a genetic algorithm. To make use of a genetic algorithm to solve this problem, the concepts of population, fitness, crossover, and mutation must be defined. The population in this work is composed of different sets of 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 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. To mutate a set of artificial joint limits, one of the joints is chosen at random and its artificial joint limits are set to random values, making sure to keep the upper joint limit larger than the lower limit. By solving this optimization problem using a genetic algorithm, the artificial joint limits which maximize the area of the fault-tolerant workspace for a given robot can be determined.

V. Results

To validate the ability of the proposed method to produce accurate fault-tolerant workspaces, a dataset is created using a planar 3R robot with link lengths of 1.25m, 0.5m, and 1.25m; 20,000 different sets of artificial joint limits; and the fault-tolerant rotation ranges associated with each set of artificial joint limits for the given robot. The mixture density network used to learn the GMM parameters is a deep feedforward neural network with 3 hidden layers having 100 neurons each. The activation function for each of these layers is the leaky ReLU function. The final layer is a linear layer with $3 \times K$ outputs: π_k, μ_k, and σ_k^2, where $K = 5$ in these experiments. Once the mixture density network is trained

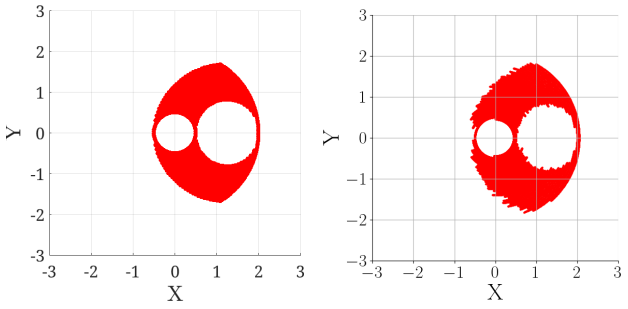


Fig. 5. The actual (a) and predicted (b) fault-tolerant workspaces of the planar 3R robot with link lengths of 1.25m, 0.5m, 1.25m and artificial joint limits of $[0°, 0°]$, $[-180°, 180°]$, and $[-180°, 180°]$ are shown.

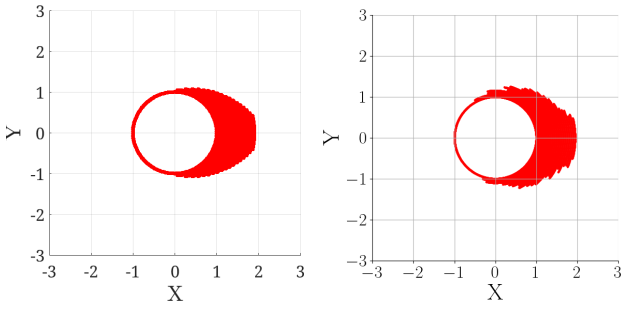


Fig. 6. The actual (a) and predicted (b) fault-tolerant workspaces of the planar 3R robot with link lengths of 0.5, 1.0m, 1.5m and artificial joint limits of $[-68.8°, 68.8°]$, $[-126.1°, 126.1°]$, and $[-114.6°, 114.6°]$ are shown.

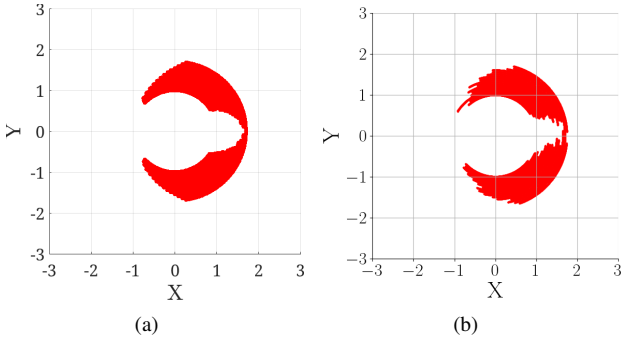


Fig. 7. The actual (a) and predicted (b) fault-tolerant workspaces of the planar 3R robot with link lengths of 0.5, 1.0m, 1.5m and artificial joint limits of $[-11.4°, 11.4°]$, $[-126.1°, 126.1°]$, and $[-137.5°, 137.5°]$ are shown.

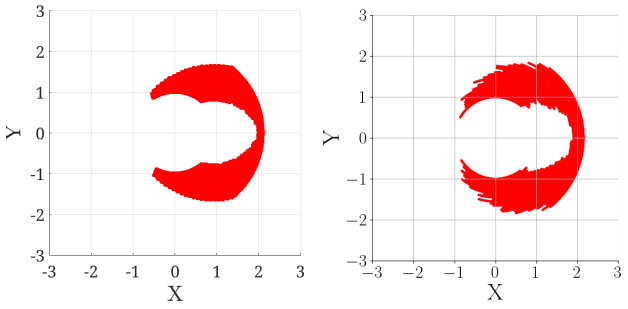
on this dataset, its performance is analyzed by using it to predict the fault-tolerant workspace of the above planar 3R robot with artificial joint limits of $[0°, 0°]$, $[-180°, 180°]$, and $[-180°, 180°]$. The actual and predicted fault-tolerant workspaces are shown in Fig. 5, where the predicted fault-tolerant workspace is very similar in shape to the actual fault-tolerant workspace. The same training process is used to learn the fault-tolerant workspace of the robot with link lengths of 0.5m, 1.0m, and 1.5m given arbitrary artificial joint limits. Two examples of the actual and predicted fault-tolerant workspaces are shown in Fig. 6 and Fig. 7, where the predicted fault-tolerant workspaces are again very similar to the actual fault-tolerant workspace. A high-performance cluster node with 192GB of RAM and a 48-core, 2.1-GHz Intel Xeon CPU is used to compute the actual fault-tolerant workspace. The predicted fault-tolerant workspace is computed on a laptop with an Nvidia MX150 GPU and 3GB of VRAM. The computational time required to compute the actual fault-tolerant workspace was roughly 60 seconds, while the proposed method required only 0.15 seconds.

The proposed method is also validated by computing $\mathcal{A}_F$ as a part of the fault-tolerance optimization process described in Section IV. The goal of this optimization is to compute the optimal artificial joint limits which maximize $\mathcal{A}_F$ for a robot with link lengths of 0.5m, 1m, and 1.5m. Because the mixture density network is used during each iteration of the genetic algorithm, its computational efficiency greatly impacts the speed of the optimization process. With a population size of 100 sets of artificial joint limits, a single iteration of the genetic algorithm using the mixture density network takes 15.41 seconds, while it takes roughly 6000 seconds to calculate the area of the fault-tolerant workspace for these joint limits using the method from [15]. The final output of the genetic algorithm is the optimal artificial joint limits $[-11.4°, 11.4°]$, $[-126.1°, 126.1°]$, and $[-137.5°, 137.5°]$ which maximize the area of the fault-tolerant workspace with the result $\mathcal{A}_F = 4.65\text{m}^2$. Both the actual and predicted fault-tolerant workspaces are shown in Fig. 8. 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 DOF Kinova Gen3 robot arm with joints 1, 3, 5, and 7 locked, which reduces the arm to a planar 3R robot. The proposed fault-tolerant workspace prediction method is used to search for the artificial joint limits which produce a fault-tolerant workspace that contains the example fault-tolerant trajectory. The produced artificial joint limits are $[-30°, 30°]$, $[-100°, 100°]$, and $[-100°, 100°]$. Once the fault-tolerant workspace is computed, the example fault-tolerant trajectory is placed inside of it as shown by the green points in Fig. 9(a)-9(c). It can be seen that the robot is able to complete the entire trajectory after joint 2 is locked in Fig. 9(a), joint 4 is locked in Fig. 9(b), joint 6 is locked in Fig. 9(c). By contrast, a trajectory is placed outside of the fault-tolerant workspace, as shown by the green points in Fig. 9(d)-9(f). Although the task is completed when joint 6 is locked in Fig. 9(f), the task fails when joint 2 is locked in Fig. 9(d) and when joint 4 is locked in Fig. 9(e).

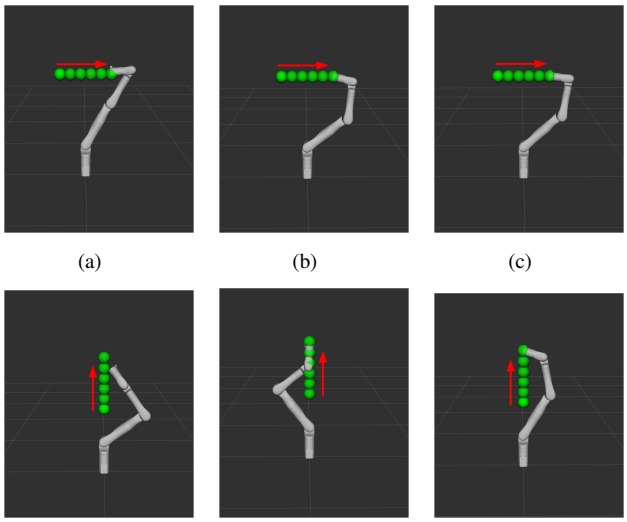
VI. Conclusion

This paper develops an innovative, efficient method of predicting the fault-tolerant workspace of planar 3R robots



(a) (b)

Fig. 8. The actual (a) and predicted (b) fault-tolerant workspaces of the planar 3R robot with link lengths of 0.5, 1.0m, 1.5m and artificial joint limits of $[-21.0°, 21.0°]$, $[-96.5°, 101.1°]$, and $[-111.2°, 108.9°]$ are shown. These optimal artificial joint limits produce a fault-tolerant workspace with an area of $\mathcal{A}_F = 4.65\text{m}^2$



(a) (b) (c)

(d) (e) (f)

Fig. 9. 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).

based on mixture density networks. Due to the computational efficiency of the proposed algorithm, it can be used alongside a genetic algorithm to solve for the optimal artificial joint limits of a robot to maximize the area of the fault-tolerant workspace. This algorithm is validated on two different robots with link lengths of 1.25m, 0.5m, and 1.25m; and 0.5m, 1.0m, and 1.5m using several different sets of artificial joint limits. The predicted fault-tolerant workspace is very close to the actual fault-tolerant workspace, while the computational speed is roughly 390 times faster than the existing method. The optimal artificial joint limits, $[-21.0°, 21.0°]$, $[-96.5°, 101.1°]$, and $[-111.2°, 108.9°]$, of the latter robot were solved for using the genetic algorithm, producing the fault-tolerant workspace with an area of 4.65m^2. To further validate this method, an experiment is performed on a Kinova robot arm. The results show that the trajectory placed inside the predicted fault-tolerant workspace can still be completed despite arbitrary joint failures. This algorithm will be extended to robots with high degrees of freedom in future works.

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] N. Tan, Z. Zhong, P. Yu, Z. Li, and F. Ni, "A discrete model-free scheme for fault-tolerant tracking control of redundant manipulators," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 8595–8606, 2022.

[5] Z. Li, C. Li, S. Li, and X. Cao, "A fault-tolerant method for motion planning of industrial redundant manipulator," *IEEE transactions on industrial informatics*, vol. 16, no. 12, pp. 7469–7478, 2019.

[6] 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.

[7] 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.

[8] B. Xie, J. Zhao, and Y. Liu, "Fault tolerant motion planning of robotic manipulators based on a nested rrt algorithm," *Industrial Robot: An International Journal*, vol. 39, no. 1, pp. 40–46, 2012.

[9] B. Xie and A. A. Maciejewski, "Maximizing the probability of task completion for redundant robots experiencing locked joint failures," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 616–625, 2021.

[10] S. Tosunoglu and V. Monteverde, "Kinematic and structural design assessment of fault-tolerant manipulators," *Intelligent Automation & Soft Computing*, vol. 4, no. 3, pp. 261–268, 1998.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.