IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

An Algorithm to Design Redundant Manipulators of Optimally Fault-Tolerant Kinematic Structure*

Ahmad A. Almarkhi[1], Anthony A. Maciejewski[1], and Edwin K. P. Chong[1]

***Abstract*— One measure of the global fault tolerance of a redundant robot is the size of its self-motion manifold. If this size is defined as the range of its joint angles, then the optimal self-motion manifold size for an n-degree-of-freedom (DoF) robot is $n \times 2\pi$, which is not typical for existing robot designs. This paper presents a novel two-step algorithm to optimize the kinematic structure of a redundant manipulator to have an optimal self-motion manifold size. The algorithm exploits the fact that singularities occur on large self-motion manifolds by optimizing the robots kinematic parameters around a singularity. Because a gradient for the self-motion manifold size does not exist, the kinematic parameter optimization uses a coordinate-ascent procedure. The algorithm was used to design 4-DoF, 7-DoF, and 8-DoF manipulators to illustrate its efficacy at generating optimally fault-tolerant robots of any kinematic structure.**

I. INTRODUCTION

Fault tolerance has been very important to the design and operation of manipulators for mission critical applications, where maintenance and repair are not feasible and a failure could result in a catastrophe. Failures are less likely for robots used in controlled environments where maintenance is relatively easy. However, reliability is critical for robots used in search and rescue operations [1]. Previous work has shown that the robot failure rates in severe environments are high [2] [3] and robot availability is as low as 50% [4]. To address some of these issues, researchers have studied the fault-tolerant control of actuators, e.g., in automated underwater vehicles [5] [6]. Also, Fault-tolerant control for multirobot systems with undetected failures was discussed in [7]. Because an entire critical mission can be jeopardized due to an unrepairable failure [8], redesigning robots to make them more fault tolerant is an important area of research [9].

Several aspects of robot fault-tolerance have been considered, such as fault detection, identification, and analysis, as surveyed in [10]. The most commonly occurring failures modes are the locked-joint failure [11], which will be considered in this work, and the free-swinging joint failure [12]. The latter mode is often transformed into the locked-joint mode by activation of fail-safe brakes [13]. Failure tolerance, necessarily, requires some level of redundancy. It can be achieved by duplicating parts that are more likely to fail (structural redundancy) [14], by human intervention to assess and overcome faults (functional redundancy), by analyzing working sensors to recover lost sensor information, e.g., integrating a tachometer signal to recover position (analytical redundancy) [10], or by designing a robot with more degree of freedom (DoFs) than the minimum needed to execute a certain task (kinematic redundancy). In this paper we focus on kinematic redundancy.

Quantifying measures of fault-tolerance for kinematically redundant robots has been extensively studied, focusing on two types, i.e., local and global measures. The local fault-tolerance measures are commonly based on the singular value decomposition (SVD) of the Jacobian matrix of a failed robot. These measures include the minimum singular value [15], the robot manipulability [16], and the condition number [17]. These local properties can be optimized by utilizing the kinematic redundancy. For example, the gradient of a singular value can be used to reconfigure a robot to satisfy a desired local fault-tolerance measure [18]. In addition, the local measures are widely used to design and control kinematically redundant fault-tolerant robots [19]–[21].

Global fault-tolerance measures typically quantify the fault-tolerant workspace. This makes these measures more suitable for pick-and-place tasks. For these types of tasks, a global measure can be used to identify the optimal fault-tolerant workspace locations [15], i.e., locations that are reachable both before and after a failure occurs. This is assured by limiting the robot to operate within software-imposed joint limits that are determined from the boundaries of the robot's self-motion manifold.

Designing a fault-tolerant workspace that is reachable for any trajectory both pre- and post-failure is more difficult. A procedure for computing the boundaries of the fault-tolerant workspace was presented in [22]. The fault-tolerant workspace can be maximized by determining the optimal artificial joint limits for a robot. This has been done by employing the gradient of the fault-tolerant workspace size as a function of the joint limits [23]. These global properties can be used to assess and optimize kinematic parameters of a robot by modifying its structure to improve its fault tolerance [24].

In our previous work [25], it was shown how one can determine the most fault-tolerant location for a given robot by identifying its largest self-motion manifold, where the manifold size depends on the ranges of all joint angles. In this work, we suggest a systematic procedure to generate robots of optimal kinematic structure by maximizing their self-motion manifold size. Because the largest self-motion manifold of a given manipulator is usually not optimal, i.e., not all joints span a 2π range on the manifold, we present an efficient algorithm to optimize the kinematic structure

*This work was not supported by any organization

[1]The authors are with the Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, Co. 80523-1373 USA almarkhi@rams.colostate.edu, aam@colostate.edu, edwin.chong@colostate.edu

of a robot to achieve the theoretically optimal self-motion manifold size.

The rest of the paper is organized in the following manner. A definition of a global fault tolerance suitable for a wide range of robotic applications is described in Section II. In Section III, we present a new optimization algorithm that can design an optimal kinematically fault-tolerant robot from any given baseline robot. In the following section, we illustrate the results of applying our algorithm on a 4-DoF, a 7-DoF, and an 8-DoF robot. Finally, we present our conclusions in Section V.

II. BACKGROUND ON SELF-MOTION MANIFOLDS[1]

A. Preliminaries

Generally, the forward kinematics of a robot is a function of its joint angles

$$\boldsymbol{x} = f(\boldsymbol{\theta}) \tag{1}$$

where $\boldsymbol{x}$ is an m-dimensional vector representing the end-effector location and $\boldsymbol{\theta}$ is an n-dimensional vector representing the joint angles. The inverse kinematics of a robot can be represented as

$$\boldsymbol{\theta} = f^{-1}(\boldsymbol{x}). \tag{2}$$

For non-redundant manipulators, the inverse-kinematic solution is a limited number of distinct solutions, but for redundant robots, i.e., $n > m$, where $n - m$ is the degree of redundancy, the inverse-kinematic solution for a certain desired end-effector location $\boldsymbol{x}_d$ can be a number of continuous sets of solutions in the joint space. Each continuous set of solutions is a self-motion manifold. The upper bound on the number of the self-motion manifolds for redundant spherical, positional, and spatial manipulators is 2, 4, and 16, respectively [26]. At the velocity level, the forward kinematics of a robot can be rewritten as

$$\dot{\boldsymbol{x}} = \boldsymbol{J}\dot{\boldsymbol{\theta}} \tag{3}$$

where $\dot{\boldsymbol{x}}$ is the end-effector velocity, $\boldsymbol{J}$ is the $m \times n$ robot Jacobian matrix, and $\dot{\boldsymbol{\theta}}$ represents the joint velocities.

The singular value decomposition (SVD) of $\boldsymbol{J}$ can be written as

$$\boldsymbol{J} = \sum_{i=1}^{r} \sigma_i \hat{\boldsymbol{u}}_i \hat{\boldsymbol{v}}_i^{\top} \tag{4}$$

where σ_i's are the ordered singular values, i.e., $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{r-1} \geq \sigma_r \geq \cdots \geq \sigma_m \geq 0$, the vectors $\hat{\boldsymbol{u}}_i$ and $\hat{\boldsymbol{v}}_i$ represent the output and input singular vectors, respectively, and r denotes the rank of $\boldsymbol{J}$, where $r < m$ for a singular robot ($\sigma_i = 0$ for $i > r$). The self-motion manifold(s) of a robot can be computed by solving

$$\boldsymbol{J}\dot{\boldsymbol{\theta}} = \boldsymbol{0} \tag{5}$$

for all possible $\dot{\boldsymbol{\theta}}$ values. We are not interested in the trivial solution $\dot{\boldsymbol{\theta}} = \boldsymbol{0}$.

[1]Much of the material in this section is adapted from [25] and is included here for completeness.

Typically, redundant manipulators have $(n - m)$-dimensional self-motion manifolds.[2] At singular configurations where self-motion manifolds connect, the number of manifolds that are connecting is one greater than the rank of the singularity. In the case that $n - m = 1$, the one-dimensional self-motion manifolds associated with a desired end-effector location $\boldsymbol{x}_d$ can be computed by solving

$$\Delta\boldsymbol{\theta} = \gamma\hat{\boldsymbol{v}}_n + \boldsymbol{J}^{+}\Delta\boldsymbol{x}_e \tag{6}$$

where $\Delta\boldsymbol{\theta}$ is the change in the joint angles, γ is a real positive scalar that represents the step size along the manifold, $\hat{\boldsymbol{v}}_n$ is the n^{th} input singular vector that represents the one-dimensional null vector of the robot's Jacobian, and $\boldsymbol{J}^{+}\Delta\boldsymbol{x}_e$ is an error correction term where $\boldsymbol{J}^{+}$ is the pseudoinverse of the Jacobian matrix[3] and $\Delta\boldsymbol{x}_e$ is the end-effector error, i.e., the difference between $f(\boldsymbol{\theta})$ and the desired location, $\boldsymbol{x}_d$.

The topology of self-motion manifolds of high-DoF redundant robots can be very complicated. That is not only because visualizing all dimensions of the manifold(s) on one plot is infeasible, but also because it is common for a self-motion manifold to contain multiple singularities of different ranks. In Fig. 1, we show an abstract sketch of an actual one-dimensional self-motion manifold for a 7-DoF robot. This is considered one manifold because it is one set of continuous joint-space solutions for a specific workspace location. One can note that singularities (indicated in black and red for rank-1 and rank-2 singularities, respectively) are where the one-dimensional self-motion manifold branches out in one or more additional directions. The topology is more complicated for multi-dimensional self-motion manifolds. For instance, one can observe that for an 8-DoF robot, the self-motion manifold of a typical end-effector location is two-dimensional. This means that a rank one singularity is now a line, not a point. The topology is even more complicated for higher degree-of-redundancy manipulators.

For redundant manipulators where $n - m > 1$, a self-motion manifold associated with a typical end-effector location is $(n - m)$-dimensional, likewise, the null space of the Jacobian matrix is $(n - m)$-dimensional. In these cases, computing a self-motion manifold is challenging. Fortunately, one can efficiently find an estimate of a bounding box on the ranges of all joint angles. This estimate can be found by modifying (6) to

$$\Delta\boldsymbol{\theta} = \gamma\boldsymbol{N}_J\hat{\boldsymbol{e}}_i + \boldsymbol{J}^{+}\Delta\boldsymbol{x}_e \tag{7}$$

where $\boldsymbol{N}_J$ is a projection operator onto the $(n - m)$-dimensional null space of the Jacobian matrix and $\hat{\boldsymbol{e}}_i$ is a basis vector along the i^{th} joint angle, where $1 \leq i \leq n$ [15].

[2]At singular configurations (singularities) associated with workspace boundaries, a self-motion manifold may be of a lower dimension.

[3]Rather than exactly computing $\boldsymbol{J}^{+}$, we employ the Damped Least Squares (DLS) technique to efficiently compute the inverse kinematics in a numerically stable manner. DLS is able to deal with singular configurations as well as the ill-conditioned transition between singular and non-singular configurations [27].

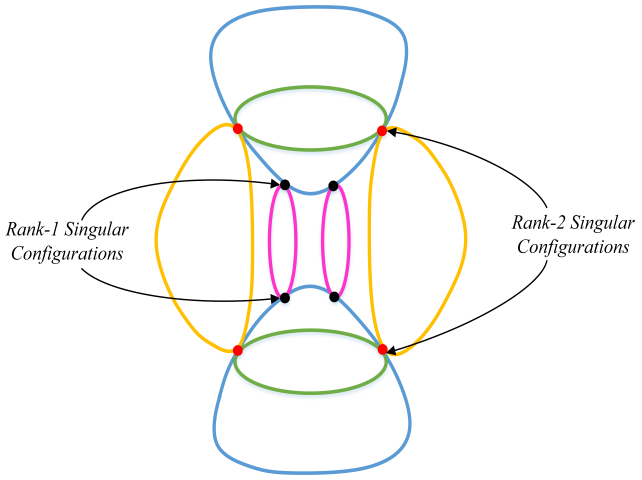

Fig. 1. This sketch illustrates the topology of a one-dimensional self-motion manifold comprised of multiple previously disjoint manifolds (for the Mitsubishi PA-10 7-DoF robot). The connection points between the previously disjoint manifolds are the singular configurations. One can observe that rank-1 singularities (in black) occur when two one-dimensional manifolds touch, while Rank-2 singularities (in red) occur when three one-dimensional manifolds touch. Note that visualizing the actual self-motion manifolds is challenging because there is typically no one projection that can show all self-motion manifolds distinctly.

The null space projection is given by

$$\boldsymbol{N}_J = \sum_{i=r+1}^{n} \hat{\boldsymbol{v}}_i \hat{\boldsymbol{v}}_i^\top. \tag{8}$$

One can approximate the joint-angle ranges at a specific task location by iteratively solving (7) for $i = 1$ to n with $\pm\hat{\boldsymbol{e}}_i$. These joint ranges can be used to compute the size of the self-motion manifold. One can terminate the iterations while solving (7) when either a joint i spans a 2π range or the projection of $\hat{\boldsymbol{e}}_i$ onto the null space becomes zero. In this case, this could be a local minimum so that this measure is a lower bound on the range of joint i.

B. Size of Self-motion Manifolds

The size of a self-motion manifold can be measured differently for different applications. To distinguish between self-motion manifolds of different shapes and sizes, [28] suggested plotting the angular distance along a self-motion manifold versus the angular distance from the origin of the manifold. However, this is not practical for manifolds with complicated topologies as well as for multi-dimensional self-motion manifolds. By iteratively solving either (6) or (7) for a specific task location, one can determine the ranges of all joint angles over the manifolds. These ranges represent a bounding box, where its volume can be used as a global fault-tolerance measure [15] for that location. Because it is not uncommon for some robot joints to have a zero range over a self-motion manifold, we compute the self-motion manifold size by summing up the ranges of all joint angles over that manifold. In general, a redundant robot has multiple self-motion manifolds associated with a workspace location. Therefore, one needs to consider the union of all angle ranges on these manifolds. At an optimally fault tolerant task location the self-motion manifold size for an n-DoF robot must be $n2\pi$, i.e., each joint spans a 2π range.

III. GENERATING OPTIMALLY FAULT TOLERANT ROBOT DESIGNS

As described above, large self-motion manifolds occur at singular configurations because previously disjoint manifolds are combined.[4] We exploit this observation to develop a procedure for identifying robot kinematic designs that have optimally fault tolerant self-motion manifolds. Generally, this is an iterative procedure where we first drive the robot to a singularity, evaluate it's self motion manifold, and then adjust the kinematic parameters, i.e., Denavit and Hartenberg (DH) parameters, to increase the size of the manifold.

At a singularity, one or more singular values of the robot's Jacobian matrix, $\boldsymbol{J}$, are zero. Therefore, to drive a robot into a singularity we employ a gradient-descent technique on a singular value to find the robot's singular joint configuration. From (4), one can express any $\boldsymbol{\sigma}_i$ as

$$\sigma_i = \hat{\boldsymbol{u}}_i^\top \boldsymbol{J} \hat{\boldsymbol{v}}_i. \tag{9}$$

By differentiating (9) with respect to time, one obtains

$$\dot{\sigma}_i = \dot{\hat{\boldsymbol{u}}}_i^\top \boldsymbol{J} \hat{\boldsymbol{v}}_i + \hat{\boldsymbol{u}}_i^\top \dot{\boldsymbol{J}} \hat{\boldsymbol{v}}_i + \hat{\boldsymbol{u}}_i^\top \boldsymbol{J} \dot{\hat{\boldsymbol{v}}}_i \tag{10}$$

where the first and the third terms vanish due to the fact that the first order change in a singular vector is orthogonal to the vector itself. Thus, (10) can be further simplified to

$$\dot{\sigma}_i = \hat{\boldsymbol{u}}_i^\top \dot{\boldsymbol{J}} \hat{\boldsymbol{v}}_i. \tag{11}$$

The partial derivative of σ_i with respect to any joint angle θ_k can be written as

$$\frac{\partial \sigma_i}{\partial \theta_k} = \hat{\boldsymbol{u}}_i^\top \frac{\partial \boldsymbol{J}}{\partial \theta_k} \hat{\boldsymbol{v}}_i \tag{12}$$

where

$$\frac{\partial \boldsymbol{J}}{\partial \theta_k} = \left[\frac{\partial \boldsymbol{j}_1}{\partial \theta_k}, \frac{\partial \boldsymbol{j}_2}{\partial \theta_k}, \cdots, \frac{\partial \boldsymbol{j}_n}{\partial \theta_k} \right]. \tag{13}$$

The partial derivative of the i^{th} column of the Jacobian matrix is given by [18], [29]

$$\frac{\partial \boldsymbol{j}_i}{\partial \theta_k} = \begin{cases} \begin{bmatrix} (\boldsymbol{z}_k^\top \boldsymbol{p}_i)\boldsymbol{z}_i - (\boldsymbol{z}_k^\top \boldsymbol{z}_i)\boldsymbol{p}_k \\ \boldsymbol{z}_k \times \boldsymbol{z}_i \end{bmatrix}, & k < i \\ \begin{bmatrix} (\boldsymbol{z}_i^\top \boldsymbol{p}_k)\boldsymbol{z}_k - (\boldsymbol{z}_k^\top \boldsymbol{z}_i)\boldsymbol{p}_k \\ \boldsymbol{0} \end{bmatrix}, & k \geq i \end{cases} \tag{14}$$

where $\boldsymbol{z}_i$ is the axis of rotation of the i^{th} robot joint and $\boldsymbol{p}_i$ is the position vector from the joint origin to the end-effector. Using (12), (13), and (14) one can compute the gradient of any σ_i of the Jacobian matrix as

$$\nabla \sigma_i = \left[\frac{\partial \sigma_i}{\partial \theta_1}, \frac{\partial \sigma_i}{\partial \theta_2}, \cdots, \frac{\partial \sigma_i}{\partial \theta_n} \right]. \tag{15}$$

[4] Typically, self-motion manifolds with internal singular configurations will be larger than average. Manifolds that include reach singularities will be smaller than average

Moving in the negative direction of this gradient allows one to decrease any desired singular value.

Maximizing a self-motion manifold's size of a robot can be done by employing a two-step iterative procedure. The first step is to drive the robot to a singular configuration from a random starting configuration $\boldsymbol{\theta}^{(0)}$. In this step, one can use the gradient descent of a singular value σ_i until it reaches zero, i.e.,

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \delta_k \nabla \sigma_i \tag{16}$$

where $\boldsymbol{\theta}^{(k)}$ is the current joint configuration, $\boldsymbol{\theta}^{(k+1)}$ is the next joint configuration, and δ_k is an adaptive step size. One can iteratively solve (16) to guarantee that the robot converges to a desired singularity. Once the robot is at a singularity, we compute the size of the self-motion manifolds for the current end-effector location.

The second step is to optimize the robot's kinematic structure so that the self-motion manifold size is maximized to an optimal value. A gradient for the self-motion manifold size does not exist, however, one can employ a gradient-free optimization technique. The optimization in this step can be formulated as

$$\boldsymbol{p}^* = \arg\max_{\boldsymbol{p} \in \boldsymbol{R}^l} f(\boldsymbol{p}) \tag{17}$$

where $\boldsymbol{p}$ is an l-dimensional vector of the robot's DH parameters, except for the joint angles, namely the link lengths ($\boldsymbol{a}$), the link displacements ($\boldsymbol{d}$), and the link twists ($\boldsymbol{\alpha}$), where $\boldsymbol{p} = [\boldsymbol{a}^\top : \boldsymbol{d}^\top : \boldsymbol{\alpha}^\top]^\top$ and $l = 3n$, where n is the number of DoF. The function $f(\boldsymbol{p})$ is a nonlinear function representing the self-motion manifold size. In this step, one can perform a *coordinate-ascent* procedure along all the DH parameters that are subject to optimization.

The coordinate ascent can be done by sequentially changing (increasing or decreasing) a DH parameter, $\boldsymbol{p}_i$. Once a $\boldsymbol{p}_i$ value is slightly changed, the robot needs to be driven back to a singularity using (16) and then the change in the self-motion manifold size is evaluated. If the size increases, one can keep updating the same $\boldsymbol{p}_i$. If the size decreases or does not change, one should step back by retrieving the last good $\boldsymbol{p}_i$ value and start changing the next DH parameter, $\boldsymbol{p}_{i+1}$. This process can be formulated as

$$\boldsymbol{p}_i^{(k+1)} = \boldsymbol{p}_i^{(k)} + \beta_k \boldsymbol{p}_i^{(k)} \tag{18}$$

where $\boldsymbol{p}_i^{(k+1)}$ is the next value of a DH parameter, $\boldsymbol{p}_i^{(k)}$ is the current value of the DH parameter, and $\beta^{(k)}$ is a user-defined step size, where β_k can be positive or negative. These two steps should be performed alternately on all DH parameters until one either obtains an optimal robot, or a sweep of all elements of $\boldsymbol{p}$ results in no improvement of $f(\boldsymbol{p})$. One should be aware that there could be a local maxima where there is no change to any DH parameter that will improve the self-motion manifold size. The pseudocode to implement this procedure is given in Algorithm 1.

There are several comments that should be pointed out about the behavior of this algorithm. First, it is important to note that the rank of the singularity being used can affect the kinematic structure of the resulting optimal robot. This means that optimizing around a rank-1 singularity could generate completely different optimal robots from the ones generated by optimizing around a singularity of higher rank.[5] Obviously, different starting joint configurations, $\boldsymbol{\theta}^{(0)}$s, may result in different optimal robots. Furthermore, using the same starting configuration, singularity rank, and step size may have drastically different results for a positive step size versus a negative step size. Finally, the algorithm may converge to an optimal robot from the first coordinate-ascent sweep through the DH parameters, however, in some cases, it may take several sweeps through the DH parameters to converge. The results of applying the above procedure for generating optimal redundant robots are illustrated for 4-DoF, 7-DoF, and 8-DoF robots in the next section.

Algorithm 1 Generate Kinematically Fault-tolerant Robots

start with DH parameters of a baseline $\boldsymbol{n}$-joint redundant robot. $\{\boldsymbol{p} = [\boldsymbol{a} : \boldsymbol{d} : \boldsymbol{\alpha}]\ \}$
initialize $\boldsymbol{\theta}^{(0)}$ $\{$*random joint-space configuration*$\}$
find $\boldsymbol{\theta}^{(sing)}$ $\{$*drive the robot to a singularity*$\}$
compute $\boldsymbol{S}_{init}$ $\{$*initial self-motion manifold size*$\}$
$\boldsymbol{S}_{large} = \boldsymbol{S}_{init}$ $\{$*save initial self-motion manifold size*$\}$
for $i = 1$ to $\boldsymbol{n} \times 3$ **do** $\{$*for DH parameters:*$[\boldsymbol{a} : \boldsymbol{d} : \boldsymbol{\alpha}]\}$
  $\boldsymbol{S}_{begin} = \boldsymbol{S}_{large}$ $\{$*save $\boldsymbol{S}_{large}$ beginning of sweep*$\}$
  $\boldsymbol{dh}_{org} = \boldsymbol{p}_i$ $\{$*save $\boldsymbol{p}_i$ in $\boldsymbol{dh}_{org}$*$\}$
  $\boldsymbol{p}_i^{(k+1)} = \boldsymbol{p}_i^{(k)} + \beta_k \boldsymbol{p}_i^{(k)}$ $\{$*perturb DH parameter $\boldsymbol{p}_i$*$\}$
  find $\boldsymbol{\theta}^{(sing)}$ $\{$*drive the robot to a singularity*$\}$
  compute $\boldsymbol{S}_{new}$ $\{$*the new self-motion manifold size*$\}$
  while $\boldsymbol{S}_{new} > \boldsymbol{S}_{large}$ **do**
    $\boldsymbol{S}_{large} = \boldsymbol{S}_{new}$ $\{$*update $\boldsymbol{S}_{large}$ with new value*$\}$
    $\boldsymbol{dh}_{org} = \boldsymbol{p}_i$ $\{$*save $\boldsymbol{p}_i$ in $\boldsymbol{dh}_{org}$*$\}$
    $\boldsymbol{p}_i^{(k+1)} = \boldsymbol{p}_i^{(k)} + \beta_k \boldsymbol{p}_i^{(k)}$ $\{$*perturb parameter $\boldsymbol{p}_i$*$\}$
    find $\boldsymbol{\theta}^{sing}$ $\{$*drive the robot to a singularity*$\}$
    compute $\boldsymbol{S}_{new}$
    if ($\boldsymbol{S}_{new} \approx n \times 2\pi$) **then** $\{$*robot is optimal*$\}$
      save the optimal robot $\{[\boldsymbol{a} : \boldsymbol{d} : \boldsymbol{\alpha} : \boldsymbol{\theta}^{(sing)}]\}$
    end if
  end while
  $\boldsymbol{p}_i = \boldsymbol{dh}_{org}$ $\{$*reset to the last good DH parameters*$\}$
end for
if ($\boldsymbol{S}_{new} \leq \boldsymbol{S}_{begin}$) **then** $\{$*sweep didn't improve $\boldsymbol{S}_{begin}$*$\}$
  go to 2 $\{$*select different starting joint configurations*$\}$
else $\{$*do another sweep*$\}$
  go to 6
end if

IV. RESULTS

A. Four-DoF Robots

The algorithm is first used to generate optimally fault-tolerant 4-DoF spatial positioning robots. We start with a

[5] Using the gradient descent of a singular value to drive a robot to a high-rank singularity may result in unwanted behavior when two or more singular values become nearly equal. In this case, a special procedure should be employed to guarantee efficient convergence [30].

baseline robot that was designed to be globally optimal with respect to a local fault-tolerance measure [31]. The DH parameters of this robot are listed in Table I.

TABLE I

DH PARAMETERS OF THE BASELINE 4-DOF ROBOT

$Link_i$	α_i [degrees]	a_i [meters]	d_i [meters]	θ_i [degrees]
1	90	1.41	0	0
2	−90	1.41	1	180
3	90	1.41	−1	180
4	0	1.22	0.50	145

We first evaluated the global fault-tolerance of this robot, i.e., its largest self-motion manifold size. Fig. 2(a) shows the maximal self-motion manifold in θ_2, θ_3, and θ_4 projection, where the singular configuration on the manifold is indicated in red. In subfigure (b), we show the angle ranges of the robot's joints on this self-motion manifold. It is clear that only θ_4 has an optimal range of 2π. The rest of the joints θ_1, θ_2, and θ_3 are not optimal.

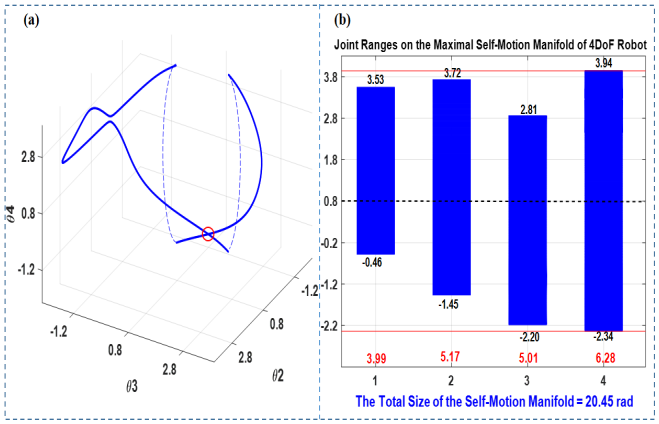

Fig. 2. In this figure, the maximal self-motion manifold of the baseline 4-DoF robot is illustrated. In (a) this manifold is shown projected into the θ_2, θ_3, and θ_4 subspace. From this projection one can see that this manifold contains only one rank-1 singularity, shown in red. Note that this robot has one continuous self-motion manifold where the dotted blue lines show the continuity of θ_4. The ranges of each of the joints is shown in (b) where only θ_4 has a range of 2π. Joints 1, 2, and 3 have ranges of 3.99, 5.17, and 5.01 radians, respectively, as indicated in red. The total size of this self-motion manifold is 20.45 radians.

By employing the optimization procedure explained in Section III, we were able to modify the kinematics of the baseline robot to generate many robots that have an optimal self-motion manifold at a particular workspace location. Because this is a 4-DoF robot, an optimal self-motion manifold size is 25.13 radians, i.e., $4 \times 2\pi$. The DH parameters of an example optimal robot are listed in Table II.

TABLE II

THE DH PARAMETERS OF AN EXAMPLE OPTIMAL 4-DOF ROBOT

$Link_i$	α_i [degrees]	a_i [meters]	d_i [meters]	θ_i [degrees]
1	130.52	1.41	0	0
2	−90	1.77	1	25.87
3	90	1.84	−1	159.78
4	0	1.52	0.55	−111.90

Comparing the optimal robot in Table II with the baseline robot in Table I, one can observe that the algorithm has changed the first twist angle, α_1 from 90° to 130.52°. The other notable change was in the link lengths a_2, a_3, and a_4 as well as the last link displacement, d_4. Note that there are multiple possible values for θ, i.e., any joint configuration on the self-motion manifold associated with this workspace location. The value shown in the table corresponds to the singularity on this manifold that was identified by the algorithm. The robot at this workspace location has only one self-motion manifold, as shown in Fig. 3.

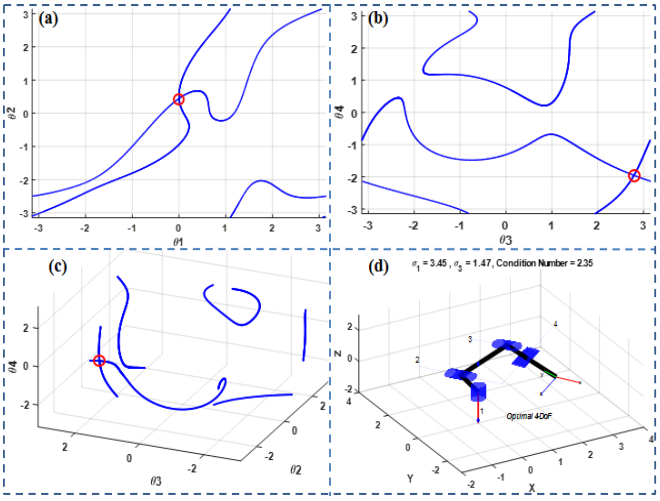

Fig. 3. In this figure, the optimal 4-DoF robot and its optimal self-motion manifold are illustrated. Subfigures (a) and (b) show projections in $\theta 1$, θ_2 and θ_3, θ_4, respectively, where the manifold is continuous and all joints span a 2π range. Thus, it is easy to see that the total size of the optimal self-motion manifold is 25.13 radians. In (c), the optimal self-motion manifold is presented in the same θ_2, θ_3, and θ_4 projection as Fig. 2(a). In (d), the robot is shown in a configuration on the self-motion manifold where a local dexterity measure, i.e., the condition number, is best ($\sigma_1/\sigma_3 = 2.35$). The rank-1 singularity on the manifold is marked with a red circle in (a), (b), and (c).

It is easy to see from subfigures (a) and (b) that the range of all joints is 2π. A rank-1 singularity that occurs on this self-motion manifold is indicated with a red circle. Subfigure (c) shows the same configuration-space projection as in Fig. 2(a) to illustrate how different the self-motion manifolds are for these two robots. Note that on this optimal self-motion manifold, one can elect to operate the manipulator in a configuration that optimizes an additional preferred dexterity measure, e.g., the condition number ($\sigma_{max}/\sigma_{min}$). In subfigure (d), we show the robot in a configuration with the best condition number on this self-motion manifold, where $\sigma_{max}/\sigma_{min} = 2.35$.

B. Seven-DoF Robots

To redundantly operate in a six-dimensional workspace consisting of both position and orientation, one needs a manipulator of at least 7 DoFs. A common 7-DoF redundant robot is the Mitsubishi PA-10, which has a kinematic design that is similar to the human arm. Unfortunately, because the arm has a three-joint spherical shoulder and a three-joint

spherical wrist that are connected by a single rotational elbow joint, the PA-10 is fault intolerant with respect to the elbow, i.e., joint 4. Therefore, joint 4 has a zero range on the self-motion manifold(s) of any workspace location. We show that this kinematically fault-intolerant structure can be used as a baseline to generate optimal 7-DoF robots.

The DH parameters of the PA-10 robot are given in Table III, with the last link displacement, d_7 set equal to d_3. The robot at the joint configurations that are given in

TABLE III
THE DH PARAMETERS OF THE PA-10 ROBOT IN MAXIMAL CONFIGURATIONS

$Link_i$	α_i [degrees]	a_i [meters]	d_i [meters]	θ_i [degrees]
1	-90	0	0	0
2	90	0	0	0
3	-90	0	0.45	± 90
4	90	0	0	154.16
5	-90	0	0.5	± 90
6	90	0	0	0
7	0	0	0.45	0

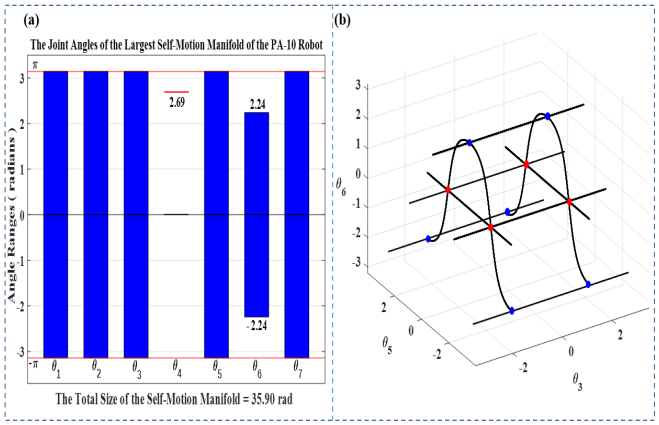

Fig. 4. The maximal self-motion manifold of the PA-10 robot is shown in this figure. In (a) the ranges of each of the joints is shown where $\theta_1, \theta_2, \theta_3, \theta_5$, and θ_7 are 2π, the range of θ_6 is 4.48 radians (± 2.24), and the range of of θ_4 is zero. In (b) the one dimensional self-motion manifold is shown projected into the θ_3, θ_5, and θ_6 subspace. From this projection one can see that this manifold contains four rank-1 and four rank-2 singularities, shown in blue and red respectively. Note that the two blue singularities at the bottom of the figure, i.e., where $\theta_6 = -2.24$ radians, are shown twice at both $\theta_5 = \pm\pi$. The rank-2 singularities occur when $\boldsymbol{\theta} = [0, 0, \frac{\pm\pi}{2}, \theta_4, \frac{\pm\pi}{2}, 0, 0]$, where in this case $\theta_4 = 2.69$ radians, $(154.16°)$.

Table 4 has a maximal self-motion manifold with size of 35.90 radians. The angles θ_3 and θ_5 can be ± 90 for the same end-effector location.[6] At this workspace location, the PA-10 has two large self-motion manifolds that are identical in terms of their joint ranges. Joint 4 has a range of zero on both self-motion manifolds with a constant value of $\pm 154.16°$ for up-elbow and down-elbow configurations, respectively. A bar plot of the joint ranges is presented in Fig. 4(a).[7] A projection of one of the manifolds in θ_3, θ_5, and θ_6 space is presented in Fig. 4(b). It is clear from the figure that θ_4 and θ_6 have non-optimal ranges.

We employed our algorithm on this fault-intolerant baseline 7-DoF manipulator to generate robots with optimal kinematic structures that have self-motion manifolds of optimal size, i.e., $7 \times 2\pi = 43.98$ radians. In Table IV, we present the DH parameters of an example optimal 7-DoF robot. By comparing Table III and Table IV, one can

TABLE IV
THE DH PARAMETERS OF AN EXAMPLE OPTIMAL 7-DOF ROBOT

$Link_i$	α_i [degrees]	a_i [meters]	d_i [meters]	θ_i [degrees]
1	-99.24	0.43	0	0
2	90	0	0	-72.25
3	-90	0	0.54	36.81
4	117	0.26	0.32	81.10
5	-90	0.54	0.50	-21.11
6	118.14	0.18	0.16	71.50
7	7.22	0.16	0.50	1.01

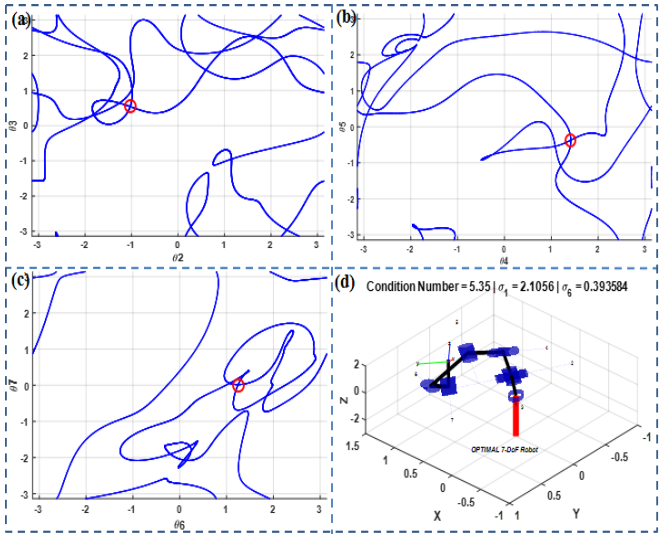

Fig. 5. The optimal self-motion manifold of the fault-tolerant 7-DoF robot is shown in this figure. A projection in θ_2, θ_3 is shown in (a), θ_3, θ_4 in (b), and θ_6, θ_7 in (c). The only singularity (which is of rank 1) on this manifold is indicated with a red circle. In (d) the optimal 7-DoF robot is presented in a configuration where its condition number is minimal, i.e., 5.35.

observe that the algorithm has significantly changed the DH parameters to generate an optimal robot. The algorithm has modified all link lengths except a_2 and a_3, displacements, d_3, d_4, d_6, and d_7, and twists α_1, α_4, α_6, and α_7. In addition, the algorithm automatically generates singular joint configurations. In Fig. 5, we present the optimal self-motion manifold of this robot.

The fault-tolerant 7-DoF robot presented in Table IV has only one continuous self-motion manifold at the optimal end-effector location. From Fig 5, one can note that all robot joints have a range of 2π. This manifold only contains one singularity, marked with a red circle, that is of rank 1, while all other intersections that appear in the figure are due to the projections.

[6]For this workspace location, there are up-elbow and down-elbow configurations where the robot cannot move from one configuration to the other without changing the end-effector location.

[7]Fig. 4 appears in our previous work [25] where we identified the largest self-motion manifold of the PA-10 robot.

C. Eight-DoF Robots

To show the merit of using our algorithm to optimize the kinematic structure of high-DoF robots, we illustrate its efficacy on an 8-DoF manipulator. We arbitrarily used an 8-DoF robot that has a 3-joint shoulder, a 2-joint elbow, and a 3-joint wrist. The DH parameters of this baseline robot are given in Table V.

TABLE V
THE DH PARAMETERS OF THE BASELINE 8-DOF ROBOT

$Link_i$	α_i [degrees]	a_i [meters]	d_i [meters]	θ_i [degrees]
1	-90	0	0	85.34
2	90	0	0	0
3	-90	0	0.54	-90
4	90	0	0	-134.50
5	-90	0	0	134.78
6	90	0	0.50	-44.78
7	-90	0	0	0
8	0	0	0.45	-12.86

This eight DoF robot at this joint configuration has the maximal self-motion manifold size of 40.80 radians. The robot and its largest self-motion manifold are both shown in Fig. 6.

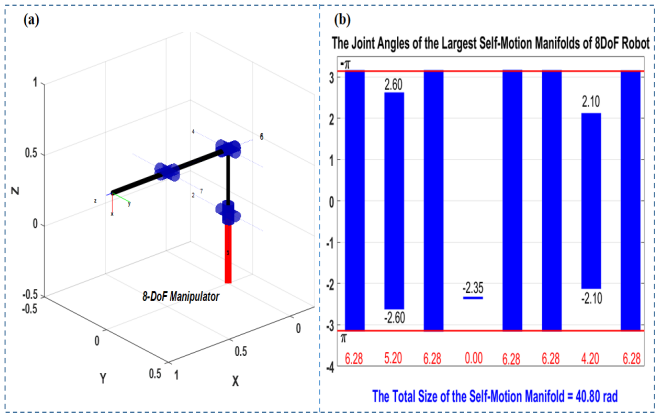

Fig. 6. In (a) the 8-DoF robot is illustrated in a configuration where $\boldsymbol{\theta} = [0, 0, 0, \frac{\pi}{2}, 0, 0, 0, 0]$ to show its structure. The joint ranges on the largest self-motion manifold are shown as a bar plot in (b) where $\theta_1, \theta_3, \theta_5, \theta_6$, and θ_8 have a 2π range. The range of θ_2 is 5.20 radians (± 2.60), the range of θ_7 is 4.20 radians (± 2.10), while θ_4 has a zero range with a fixed value at $\theta_4 = -2.35$ radians.

One should note that in this case, the self-motion manifolds are typically two dimensional, but are of a higher dimension at singularities. In this case, computing a self-motion manifold is challenging, but one can employ (7) to determine an estimate of the ranges of all joints. The optimal self-motion manifold size is $8 \times 2\pi = 50.27$ radians. In Table VI, we list an example 8-DoF optimal robot that was generated by employing Algorithm 1. This robot is shown in Fig. 7 in the zero configuration and in a relatively dexterous configuration where the condition number is 7.18. One can immediately observe that the algorithm has introduced a single change in link twist parameters by changing α_6 from 90 to 105.30 degrees. Also, two link lengths, a_1 and a_5 were introduced as well as other minor changes in the link displacement parameters.

TABLE VI
THE DH PARAMETERS OF THE OPTIMAL 8-DOF ROBOT

$Link_i$	α_i [degrees]	a_i [meters]	d_i [meters]	θ_i [degrees]
1	-90	0.45	0	0
2	90	0	0	-83.60
3	-90	0	0.45	180
4	90	0	0	90
5	-90	0.43	0	-168
6	105.30	0	0.52	-90
7	-90	0	0.17	0
8	0	0	0.45	-29.05

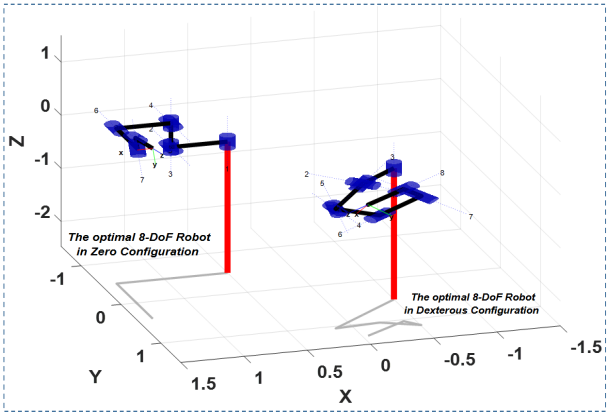

Fig. 7. The kinematic design of an optimally fault-tolerant 8-DoF robot is shown in this figure. The robot is presented in the zero configuration to illustrate how various joints are connected (left). The robot is shown in a dexterous configuration where the condition number of the Jacobian matrix is 7.18 (right).

D. Discussion

Now that we have robots of different DoFs that are optimal in terms of their global fault-tolerance measure, i.e., their self-motion manifold size, we evaluate the quality of these self-motion manifolds with respect to a local dexterity measure. This is done by evaluating the condition number ($\sigma_{max}/\sigma_{min}$) of the Jacobian matrix for configurations on the self-motion manifold and estimating how much of the manifold can meet a certain threshold. For example, for the 4-DoF robot in Fig. 3, over 90% of the self-motion manifolds have condition numbers less than 10, and for the 7-DoF robot in Fig. 5 it is over 20%. This illustrates that robot designers have significant flexibility in satisfying multiple criteria, i.e., having a robot that is both optimally fault tolerant and meets pre-failure dexterity design specifications.

V. CONCLUSIONS AND FUTURE WORK

The goal of the work described here is to identify redundant robot kinematic designs that possess optimally fault tolerant locations of operation within their workspace. If the definition of fault tolerance is reachability of a task location after any arbitrary locked-joint failure occurs, then this corresponds to identifying robots that possess self-motion manifolds that span 2π in every joint angle for that task location. We exploit the fact that such large self-motion manifolds are more likely when they contain singular configurations, because singularities occur when previously disjoint manifolds are connected. Our novel algorithm for

identifying optimal kinematic designs alternates between driving the robot to a singular configuration and modifying the kinematic parameters by using a coordinate-ascent algorithm to increase the self-motion manifold size.

The premise in this work is that a robot designer already has a baseline robot kinematics in mind, and one would like to improve the fault tolerance of that design, without changing it radically. Remarkably, the algorithm was able to find multiple optimal designs from 4-DoF, 7-DoF, and 8-DoF baseline robots. In our future work, we will be investigating ways to evaluate and characterize all of these optimally fault-tolerant robots and determine whether they can also be optimized for additional desirable objectives. In addition, it is important to point out that the proposed algorithm results in an optimal fault tolerant configuration, which is appropriate for critical workspace locations involved in pick-and-place tasks. For other types of tasks, one may want to guarantee a specified high-level of fault tolerance over a given workspace volume.

REFERENCES

[1] F. Matsuno and S. Tadokoro, "Rescue robots and systems in Japan," in *IEEE Int. Conf. on Robotics and Biomimetics*, pp. 12–20, 2004.

[2] S. Cheng and B. Dhillon, "Reliability and availability analysis of a robot-safety system," *Journal of Quality in Maintenance Engineering*, 2011.

[3] D. Mahar, W. Fields, J. Reade, P. Zarubin, and S. McCombie, "Nonelectronic parts reliability data," *Reliability Information Analysis Center*, 2011.

[4] J. Carlson and R. R. Murphy, "How UGVs physically fail in the field," *IEEE Trans. on Robotics*, vol. 21, no. 3, pp. 423–437, 2005.

[5] T. K. Podder, G. Antonelli, and N. Sarkar, "Fault tolerant control of an autonomous underwater vehicle under thruster redundancy: Simulations and experiments," in *IEEE Int. Conf. on Robotics and Automation*, pp. 1251–1256, 2000.

[6] N. Ranganathan, M. I. Patel, and R. Sathyamurthy, "An intelligent system for failure detection and control in an autonomous underwater vehicle," *IEEE Trans. on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 31, no. 6, pp. 762–767, 2001.

[7] H. Park and S. A. Hutchinson, "Fault-tolerant rendezvous of multirobot systems," *IEEE Trans. on Robotics*, vol. 33, no. 3, pp. 565–582, 2017.

[8] K. Nagatani, S. Kiribayashi, Y. Okada, K. Otake, K. Yoshida, S. Tadokoro, T. Nishimura, T. Yoshida, E. Koyanagi, M. Fukushima, and S. Kawatsuma, "Emergency response to the nuclear accident at the Fukushima Daiichi nuclear power plants using mobile rescue robots," *Journal of Field Robotics*, vol. 30, no. 1, pp. 44–63, 2013.

[9] K. Nagatani, S. Kiribayashi, Y. Okada, S. Tadokoro, T. Nishimura, T. Yoshida, E. Koyanagi, and Y. Hada, "Redesign of rescue mobile robot Quince," in *IEEE Int. Symposium on Safety, Security, and Rescue Robotics*, pp. 13–18, 2011.

[10] M. L. Visinsky, J. R. Cavallaro, and I. D. Walker, "Robotic fault detection and fault tolerance: A survey," *Reliability Engineering & System Safety*, vol. 46, no. 2, pp. 139–158, 1994.

[11] Y. Ting, S. Tosunoglu, and B. Fernandez, "Control algorithms for fault-tolerant robots," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pp. 910–915, IEEE, 1994.

[12] J. D. English and A. A. Maciejewski, "Fault tolerance for kinematically redundant manipulators: Anticipating free-swinging joint failures," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 566–575, 1998.

[13] P. Nieminen, S. Esque, A. Muhammad, J. Mattila, J. Väyrynen, M. Siuko, and M. Vilenius, "Water hydraulic manipulator for fail safe and fault tolerant remote handling operations at iter," *Fusion Engineering and Design*, vol. 84, no. 7-11, pp. 1420–1424, 2009.

[14] V. Monteverde and S. Tosunoglu, "Effect of kinematic structure and dual actuation on fault tolerance of robot manipulators," in *IEEE Int. Conf. on Robotics and Automation*, pp. 2902–2907, 1997.

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

[16] R. G. Roberts and A. A. Maciejewski, "A local measure of fault tolerance for kinematically redundant manipulators," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 4, pp. 543–552, 1996.

[17] H. Abdi and S. Nahavandi, "Minimum reconfiguration for fault tolerant manipulators," in *Proceedings of the 34th Annual Mechanisms and Robotics Conference, Parts A and B*, pp. 1345–1350, 2010.

[18] K. N. Groom, A. A. Maciejewski, and V. Balakrishnan, "Real-time failure-tolerant control of kinematically redundant manipulators," *IEEE Trans. on Robotics and Automation*, vol. 15, no. 6, pp. 1109–1115, 1999.

[19] F. Hammond, "Synthesis of k th order fault-tolerant kinematically redundant manipulator designs using relative kinematic isotropy," *International Journal of Adaptive and Innovative Systems*, vol. 2, no. 1, pp. 73–96, 2014.

[20] C. S. Ukidve, J. E. McInroy, and F. Jafari, "Using redundancy to optimize manipulability of stewart platforms," *IEEE/ASME Transactions on Mechatronics*, vol. 13, no. 4, pp. 475–479, 2008.

[21] Y. She, W. Xu, H. Su, B. Liang, and H. Shi, "Fault-tolerant analysis and control of ssrms-type manipulators with single-joint failure," *Acta Astronautica*, vol. 120, pp. 270–286, 2016.

[22] 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 Trans. on Automation Science and Engineering*, vol. 12, no. 4, pp. 1421–1432, 2015.

[23] 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.

[24] K. M. Ben-Gharbia, A. A. Maciejewski, and R. G. Roberts, "Modifying the kinematic structure of an anthropomorphic arm to improve fault tolerance," in *IEEE Int. Conf. on Robotics and Automation*, pp. 1455–1460, 2015.

[25] A. A. Almarkhi and A. A. Maciejewski, "Maximizing the size of self-motion manifolds to improve robot fault tolerance," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2653–2660, 2019.

[26] J. W. Burdick, *Kinematic analysis and design of redundant robot manipulators*. PhD thesis, Department of Computer Science, Stanford University, 1988.

[27] A. A. Maciejewski and C. A. Klein, "The singular value decomposition: Computation and applications to robotics," *The International Journal of Robotics Research*, vol. 8, no. 6, pp. 63–79, 1989.

[28] K. M. Ben-Gharbia, A. A. Maciejewski, and R. G. Roberts, "Modifying the kinematic structure of an anthropomorphic arm to improve fault tolerance," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1455–1460, IEEE, 2015.

[29] C. A. Klein and L.-C. Chu, "Comparison of extended Jacobian and Lagrange multiplier based methods for resolving kinematic redundancy," *Journal of Intelligent & Robotic Systems*, vol. 19, no. 1, pp. 39–54, 1997.

[30] A. A. Almarkhi and A. A. Maciejewski, "Singularity analysis for redundant manipulators of arbitrary kinematic structure," *International Conference on Informatics in Control, Automation and Robotics (ICINCO 2019)*, pp. 42–49, 2019.

[31] K. M. Ben-Gharbia, A. A. Maciejewski, and R. G. Roberts, "Kinematic design of redundant robotic manipulators for spatial positioning that are optimally fault tolerant," *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1300–1307, 2013.