Short Papers

Kinematic Design of Redundant Robotic Manipulators for Spatial Positioning that are Optimally Fault Tolerant

Khaled M. Ben-Gharbia, Anthony A. Maciejewski, and Rodney G. Roberts

***Abstract*—This work presents a method for identifying all the kinematic designs of spatial positioning manipulators that are optimally fault tolerant in a local sense. We use a common definition of fault tolerance, i.e., the post-failure Jacobian possesses the largest possible minimum singular value over all possible single locked-joint failures. The large family of physical manipulators that can achieve this optimally failure tolerant configuration is then parameterized and categorized. We develop a general computational technique to evaluate the resulting manipulators in terms of their global kinematic properties, with an emphasis on failure tolerance. Several manipulators with a range of desirable kinematic properties are presented and analyzed, with a specific example of optimizing over a given class of manipulators that possess a specified kinematic constraint.**

***Index Terms*—Fault-tolerant robots, redundant robots, robot kinematics.**

I. Introduction

The design and operation of fault-tolerant manipulators is critical for applications in remote and/or hazardous environments where repair is not possible. Component failures for robots that are employed in structured and benign environments where regular maintenance can be performed are relatively rare. However, there are many important applications, although less common, where this is not true, e.g., in space exploration [1], [2], underwater exploration [3], and nuclear waste remediation [4], [5].[1] The failure rates for components in such harsh environments are relatively high [7]–[9], and maintenance is not possible. Many of these component failures will result in a robot's joint becoming immobilized, i.e., a locked-joint failure mode [10], [11]. In addition, component failures that result in other common failure modes, e.g., free-swinging joint failures [12], [13], are frequently transformed into the locked-joint failure mode by failure recovery mechanisms that employ fail safe brakes [14]. Because of the severe consequences of such failures there has been a great deal of research to improve manipulator reliability [9], [15], design fault-tolerant robots [16], [17], and determine mechanisms for analyzing [18], detecting [11], [19], identifying [20]–[22], and recovering [23]–[26] from failures.

A large body of work on fault-tolerant manipulators has focused on the properties of kinematically redundant robots, either in serial or parallel form [27]–[31]. These analyses have been performed both on the local properties that are associated with the manipulator Jacobian [32]–[36] and the global characteristics such as the resulting workspace following a particular failure [37]–[40]. (Clearly both local and global kinematic properties are related, e.g., workspace boundaries correspond to singularities in the Jacobian.)

In this work it is assumed that one is given a set of local performance constraints, defined by a desired Jacobian, that require a manipulator to function in a configuration that is optimal under normal operation and after an arbitrary single joint fails, and is locked in position. In our previous work [41] and [42], it was shown that there exist multiple different physical planar manipulators that correspond to the same optimally fault-tolerant Jacobian. In this work we build on our previous results [43] and consider the Jacobian for an optimally fault-tolerant, spatial positioning manipulator that possesses four degrees of freedom (DOFs) and show that there is a much greater degree of design flexibility. We characterize entire families of manipulators that correspond to this specific Jacobian and analyze multiple categories. One can then use global characteristics to distinguish between multiple manipulators that meet the local design constraints.

The remainder of this paper is organized in the following manner. A local definition of failure tolerance centered on desirable properties of the manipulator Jacobian is mathematically defined in the next section. In Section III, the set of all 6×4 Jacobian matrices that include an optimally fault-tolerant 3×4 spatial positioning sub-Jacobian are characterized. This characterization is then used to determine the family of Denavit and Hartenberg (DH) parameters that represent physical manipulators with the optimally fault-tolerant property. In Section IV, we describe how one can evaluate a particular robot design (that is generated from the optimal Jacobian) in terms of its global kinematic properties, especially with regard to failure tolerance. The following section illustrates such an analysis on several categories of robots. Finally, the conclusions of this work are presented in Section VI.

II. Background on Optimally Fault-Tolerant Jacobians

In this section, we briefly review our definition of an optimally fault-tolerant Jacobian [43]. The dexterity of manipulators is frequently quantified in terms of the properties of the manipulator Jacobian matrix that relates end-effector velocities to joint angle velocities. The Jacobian will be denoted by the $m \times n$ matrix J, where m is the dimension of the task space, and n is the number of DOFs of the manipulator. For redundant manipulators, $n > m$ and the quantity $n - m$ is the degree of redundancy. The manipulator Jacobian can be written as a collection of columns

$$J_{m\times n} = [\, j_1 \quad j_2 \quad \cdots \quad j_n \,] \tag{1}$$

where j_i represents the end-effector velocity due to the velocity of joint i. For an arbitrary single joint failure at joint f, assuming that the failed joint can be locked, the resulting m by $n-1$ Jacobian will be missing the fth column, where f can range from 1 to n. This Jacobian will be denoted by a preceding superscript so that, in general,

$${}^{f}J_{m\times(n-1)} = [\, j_1 \quad j_2 \quad \cdots \quad j_{f-1} \quad j_{f+1} \quad \cdots \quad j_n \,]. \tag{2}$$

Manuscript received November 30, 2012; revised March 28, 2013; accepted June 2, 2013. Date of publication July 3, 2013; date of current version September 30, 2013. This paper was recommended for publication by Associate Editor Y. Choi and Editor W. K. Chung upon evaluation of the reviewers' comments. This work was supported in part by the National Science Foundation under Contract IIS-0812437.

K. M. Ben-Gharbia and A. A. Maciejewski are with the Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523-1373 USA (e-mail: khaled.ben-gharbia@colostate.edu; aam@colostate.edu).

R. G. Roberts is with the Department of Electrical and Computer Engineering, Florida A&M–Florida State University, Tallahassee, FL 32310-6046 USA (e-mail: rroberts@eng.fsu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2013.2266855

[1]One recent example is the Fukushima nuclear reactor accident, where robot component failures were not only likely, but inevitable [6].

The properties of a manipulator Jacobian are frequently quantified in terms of the singular values, denoted σ_i, which are typically ordered so that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m \geq 0$. Most local dexterity measures can be defined in terms of simple combinations of these singular values such as their product (determinant) [44], sum (trace), or ratio (condition number) [45]–[47]. The most significant of the singular values is σ_m, the minimum singular value, because it is by definition the measure of proximity to a singularity and tends to dominate the behavior of both the manipulability (determinant) and the condition number. The minimum singular value is also a measure of the worst case dexterity over all possible end-effector motions.

The definition of failure tolerance used in this work is based on the worst case dexterity following an arbitrary locked-joint failure. Because ${}^f\sigma_m$ denotes the minimum singular value of ${}^fJ$, ${}^f\sigma_m$ is a measure of the worst case dexterity if joint f fails. If all joints are equally likely to fail, then a measure of the worst case failure tolerance is given by

$$\mathcal{K} = \min_{f=1}^{n}({}^f\sigma_m). \tag{3}$$

Physically, this amounts to minimizing the worst-case increase in joint velocity when a joint is locked, and the others must accelerate to maintain the desired end-effector trajectory. In addition, maximizing $\mathcal{K}$ is equivalent to locally maximizing the distance to the post-failure workspace boundaries [7]. To insure that manipulator performance is optimal prior to a failure, an optimally failure tolerant Jacobian is further defined as having all equal singular values due to the desirable properties of isotropic manipulator configurations [45]–[47]. Under these conditions, to guarantee that the minimum ${}^f\sigma_m$ is as large as possible they should all be equal. It is easy to show [33] that the worst-case dexterity of an isotropic manipulator that experiences a single joint failure is governed by the inequality

$$\min_{f=1}^{n}({}^f\sigma_m) \leq \sigma\sqrt{\frac{n-m}{n}} \tag{4}$$

where σ denotes the norm of the original Jacobian. The best case of equality occurs if the manipulator is in an optimally failure tolerant configuration. The above inequality makes sense from a physical point of view because it represents the ratio of the degree of redundancy to the original number of DOFs.

Using the above definition of an optimally failure tolerant configuration, one can identify the structure of the Jacobian that is required to obtain this property [48].[2] In particular, one can show that the optimally failure tolerant criteria requires that each joint contributes equally to the null space of the Jacobian transformation [34], [35]. Physically, this means that the redundancy of the robot is uniformly distributed among all the joints so that a failure at any one joint can be compensated for by the remaining joints. Therefore, in this work an optimally failure tolerant Jacobian is defined as being isotropic, i.e., $\sigma_i = \sigma$ for all i, and having a maximum worst-case dexterity following a failure, i.e., one for which ${}^f\sigma_m = \sigma\sqrt{\frac{n-m}{n}}$ for all f. Because the manipulator Jacobian is assumed to be isotropic, the second condition is equivalent to the columns of the Jacobian having equal norms.

[2]Note that our approach does not depend on our choice of fault-tolerance measure. Any fault-tolerant measure, e.g. relative manipulability, can be used to define a locally optimally failure tolerant Jacobian. In fact, any local desired property that is defined by a Jacobian can be used in our approach.

For the case of a spatial positioning manipulator with four joints, an optimally failure tolerant configuration is given by

$$J_v = \begin{bmatrix} -\sqrt{\frac{3}{4}} & \sqrt{\frac{1}{12}} & \sqrt{\frac{1}{12}} & \sqrt{\frac{1}{12}} \\ 0 & -\sqrt{\frac{2}{3}} & \sqrt{\frac{1}{6}} & \sqrt{\frac{1}{6}} \\ 0 & 0 & -\sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \end{bmatrix} \tag{5}$$

where J_v represents the linear velocity portion of a manipulator Jacobian. The null space at this configuration is given by $\frac{1}{2}[1\ \ 1\ \ 1\ \ 1]^T$, which illustrates that each joint contributes equally to the null-space motion, thus distributing the redundancy proportionally to all DOFs. If the four possible single locked-joint failures are considered, one can show that

$${}^f\sigma_3 = \mathcal{K}_{\max} = \frac{1}{2} \tag{6}$$

for $f = 1$ to 4, which satisfies the optimally failure tolerant criterion.

Equation (5) is in fact a canonical form that essentially characterizes all optimally fault-tolerant 3×4 Jacobians J_v. Optimal fault tolerance requires that the components of the unit length null vector have the same magnitude. Without loss of generality, these components can be taken to be equal to each other by multiplying columns of J_v by -1, if necessary. Multiplying a column of the Jacobian by -1 corresponds to redefining that axis of rotation or translation to be in the opposite direction, which does not essentially change the manipulator design. Hence, we can always assume that the null space of an optimally fault-tolerant J_v is given by the unit length null vector $\hat{n}_{J_v} = \frac{1}{2}[1\ \ 1\ \ 1\ \ 1]^T$. With this choice of $\hat{n}_{J_v}$ along with the isotropy condition $\sigma_i = \sigma = 1$, we have that

$$J_v^T J_v = I - \hat{n}_{J_v}\hat{n}_{J_v}^T = \begin{bmatrix} \frac{3}{4} & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4} & \frac{3}{4} & -\frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4} & -\frac{1}{4} & \frac{3}{4} & -\frac{1}{4} \\ -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & \frac{3}{4} \end{bmatrix} \tag{7}$$

implying that the columns of J_v each have length $\sqrt{\frac{3}{4}}$, and the dot product of any two distinct columns is equal to $-\frac{1}{4}$. Furthermore, since fault tolerance dictates that any three columns of J_v are linearly independent, one can apply the QR factorization to determine a unique orthogonal matrix Q so that QJ_v is upper triangular with negative values along its main diagonal. Applying this orthogonal matrix merely rotates and/or reflects the base coordinate frame so that a rotated/reflected manipulator is obtained. The only 3×4 matrix satisfying these conditions, i.e., isotropy with $\sigma = 1$, a null vector with equal components, and an upper triangular form with negative components along the main diagonal, is (5). Hence, any optimally fault-tolerant Jacobian J_v can be written as (5) by performing a series of suitable coordinate transformations.

The next section will illustrate how to characterize the set of all 6×4 Jacobian matrices that have the linear velocity portion given by J_v in (5). Once we have all these possible 6×4 Jacobians, we will be able to determine the DH parameters for the physical robots.

III. Characterizing Fault-Tolerant Four-Degree-Of-Freedom Spatial Positioning Manipulators

Our goal in this section, as in [43], is to determine all possible Jacobians of the form

$$J_{6\times 4} = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix}. \tag{8}$$

The orientational velocity portion, J_ω, is somewhat arbitrary because it does not affect the positional fault-tolerance properties. However, one must consider the constraint that each column of J_ω is orthogonal to the corresponding column of J_v. The ith column of J in (8) can be written as

$$j_i = \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} \tag{9}$$

where v_i and ω_i are three-dimensional vectors that describe the linear and angular velocities respectively. By applying the constraints that ω_i is of unit norm and orthogonal to v_i, one can characterize all valid ω_i's by a circle centered at the origin, and parameterized by a function of an angle that we denote β_i.

To illustrate this, consider the first column of $J_{6\times 4}$. Let $\omega_1 = [\,\omega_{11} \quad \omega_{21} \quad \omega_{31}\,]^T$. Because ω_1 and v_1 are orthogonal,

$$\omega_1^T v_1 = [\,\omega_{11} \quad \omega_{21} \quad \omega_{31}\,] \begin{bmatrix} -\sqrt{\frac{3}{4}} \\ 0 \\ 0 \end{bmatrix} = 0 \tag{10}$$

so that $\omega_{11} = 0$. Because ω_1 is a normalized vector, $\omega_{21}^2 + \omega_{31}^2 = 1$, it follows that ω_1 can be written as

$$\omega_1 = \begin{bmatrix} 0 \\ \cos(\beta_1) \\ \sin(\beta_1) \end{bmatrix}. \tag{11}$$

Similarly, one can find that ω_2 is described by the following equation:

$$\omega_2 = \begin{bmatrix} \omega_{12} \\ \omega_{22} \\ \omega_{32} \end{bmatrix} = \begin{bmatrix} \omega_{12} \\ \sqrt{\frac{1}{8}}\omega_{12} \\ \pm\sqrt{1 - \frac{9}{8}\omega_{12}^2} \end{bmatrix} \tag{12}$$

where $|\omega_{12}| \leq \frac{2\sqrt{2}}{3}$. One can write (12) as a function of β_2 so that

$$\omega_2 = \begin{bmatrix} \frac{2\sqrt{2}}{3}\cos(\beta_2) \\ \frac{1}{3}\cos(\beta_2) \\ \sin(\beta_2) \end{bmatrix}. \tag{13}$$

Likewise, ω_3 and ω_4 can be written as a functions of β_3 and β_4, respectively, as

$$\omega_3 = \begin{bmatrix} \frac{2\sqrt{2}}{3}\sin(\beta_3) \\ -\left(\frac{\sqrt{3}}{2}\cos(\beta_3) + \frac{1}{6}\sin(\beta_3)\right) \\ -\frac{1}{2}\cos(\beta_3) + \frac{\sqrt{3}}{6}\sin(\beta_3) \end{bmatrix} \tag{14}$$

and

$$\omega_4 = \begin{bmatrix} \frac{2\sqrt{2}}{3}\sin(\beta_4) \\ -\left(\frac{\sqrt{3}}{2}\cos(\beta_4) + \frac{1}{6}\sin(\beta_4)\right) \\ -\left(-\frac{1}{2}\cos(\beta_4) + \frac{\sqrt{3}}{6}\sin(\beta_4)\right) \end{bmatrix}. \tag{15}$$

In all of the above equations, β_i can be any value between 0° and 360°.

Now that the set of possible ω_i's has been characterized, our next step is to determine the DH parameters for the corresponding robots as functions of the β_i's. The link parameters of twist (α_i) and length (a_i) for link i are determined from the i and $i+1$ coordinate frames. Therefore, they are affected by the β_i and β_{i+1} parameters, i.e.,

$$\alpha_i = f_{\alpha_i}(\beta_i, \beta_{i+1}) \qquad \text{and} \tag{16}$$

$$a_i = f_{a_i}(\beta_i, \beta_{i+1}). \tag{17}$$

For example, there is considerable flexibility in selecting α_1 and a_1, i.e., the twist angle can be set anywhere from 0° to 180°, and the magnitude of the length can be anywhere from 0 to $\sqrt{3}$. (See [43] for plots of these functions.) Because the tool, i.e., fifth, coordinate frame is arbitrary, we assume it to be in the same orientation as the fourth, so that

$$\alpha_4 = 0 \qquad \text{and} \tag{18}$$

$$a_4 = \sqrt{3}/2. \tag{19}$$

The joint parameters of the rotation angle (θ_i) and offset (d_i) for joint i are determined from the $i-1, i$, and $i+1$ coordinate frames; so they are influenced by the β_{i-1}, β_i, and β_{i+1} parameters, i.e.,

$$\theta_i = f_{\theta_i}(\beta_{i-1}, \beta_i, \beta_{i+1}) \qquad \text{and} \tag{20}$$

$$d_i = f_{d_i}(\beta_{i-1}, \beta_i, \beta_{i+1}). \tag{21}$$

For the first coordinate frame, θ_1 and d_1 are arbitrary so they can be assumed to be zero because we can select the orientation of the zeroth coordinate frame. At the fourth coordinate frame, the joint parameters are not functions of the fifth coordinate frame, i.e.,

$$\theta_4 = f_{\theta_4}(\beta_3, \beta_4) \qquad \text{and} \tag{22}$$

$$d_4 = f_{d_4}(\beta_3, \beta_4), \tag{23}$$

because it is selected to be aligned with the fourth.

The exact values of the DH parameters for a given set of β_i's can be computed using the algorithm that is presented in [41]. Clearly, there is an infinite family of robots that correspond to (5). The next section will discuss how to compute the global failure tolerance properties of the various possible physical robots that can be generated from this single optimal Jacobian.

IV. Computing Global Fault-Tolerance Properties

Different combinations of $(\beta_1,\ \beta_2,\ \beta_3,\ \beta_4)$ correspond to different potential robots (in terms of their DH parameters). While these robots have the same desired optimal local fault-tolerant design point, they are quite different in terms of their global properties. Not only is the size of the workspace quite different, but more importantly if one is concerned with fault tolerance, there is considerable difference in how the value of the fault tolerance measure varies away from the design point.

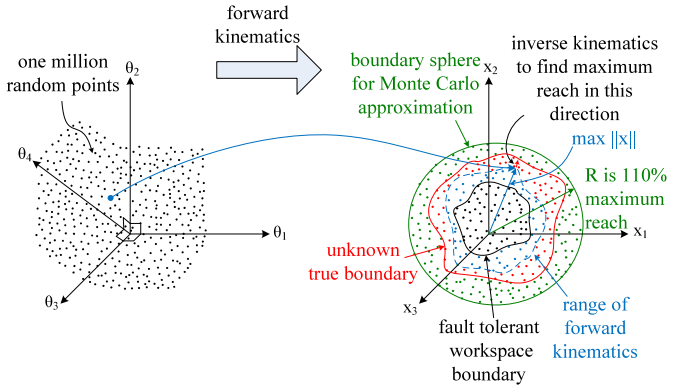

Fig. 1. An illustration of how Monte Carlo integration algorithm is used to compute the volume of both the fault-tolerant workspace and the entire reachable workspace.

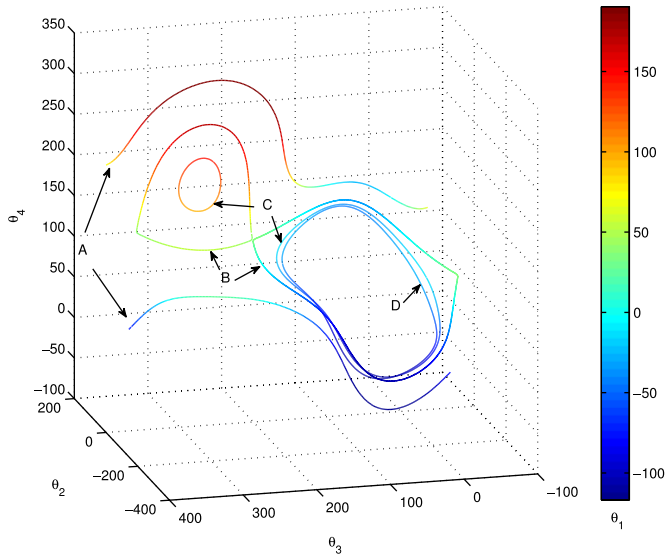

Fig. 2. Self-motion manifolds for the robot that are generated from $(\beta_1,\ \beta_2,\ \beta_3,\ \beta_4) = (200°,\ 220°,\ 130°,\ 90°)$, where the value of θ_1 is indicated using color. The points A, B, C, and D lie on a line segment in the workspace, where A is one endpoint that is located at $[x, y, z] = [-0.04, -0.82, 0.22]$, D is the other endpoint at $[x, y, z] = [-0.16, -1.79, 0.47]$, and points B and C are located 57% and 80% of the way from A to D, respectively.

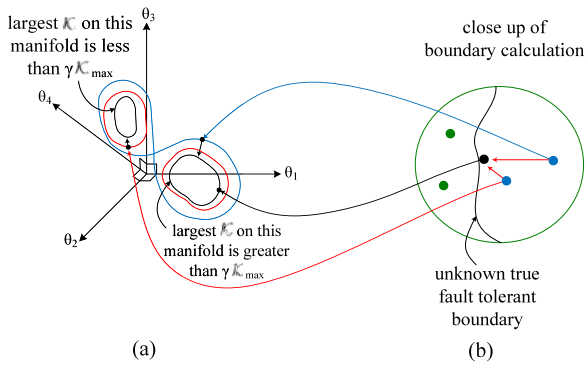

Fig. 3. To evaluate the maximum value of $\mathcal{K}$ at a workspace location [shown in black in (b)], we use multiple randomly generated configurations (two of which are shown in blue) in order to evaluate the multiple self-motion manifolds associated with a workspace location. The configuration that corresponds to the largest value of $\mathcal{K}$ for each of these workspace locations is shown in (a). Note that the closest random point may not be associated with the manifold that has the largest value of $\mathcal{K}$.

To compare different physical robot designs, we use a measure of how the fault tolerance varies across the entire workspace. Specifically, we compute the volume of the workspace that has a $\mathcal{K}$ greater than or equal to a given fraction of the maximum, i.e., $\mathcal{K} \geq \gamma\ \mathcal{K}_{\max}$, where $0 \leq \gamma \leq 1$ is a user defined parameter. (For all of the results shown in the following examples we use $\gamma = 0.9$.) We denote this fault-tolerant volume by V_{f} and then divide it by the total reachable volume, denoted V_{r}, to obtain a normalized global measure of fault tolerance that can be used to compare different robots.

One of the best techniques for computing an estimate for the volume of an unknown three-dimensional shape is to use the Monte Carlo integration algorithm (see Fig. 1). We first generate one million uniformly distributed random configurations in the joint space, where $0 \leq \theta_i < 2\pi$ for all i, that are transformed to the workspace using forward kinematics. We then estimate the maximum reach of the manipulator by picking the point with the largest norm and use inverse kinematics to drive the robot until its Jacobian is singular and we have reached a workspace boundary. We use a sphere whose radius R is 110% of this maximum reach as the boundary for our Monte Carlo integration. We then generate 10,000 uniformly distributed random points within this sphere and determine if they are reachable. We determine reachability by performing iterative inverse kinematics starting with the joint configuration (from the one million configurations initially generated) whose workspace location is the closest to the Monte Carlo point being evaluated. An estimate for the total reachable workspace volume is then given by

$$V_{\mathrm{r}} = \left(\frac{n_{\mathrm{r}}}{10,000}\right)\frac{4}{3}\pi R^3 \tag{24}$$

where n_{r} is the number of reachable points. For each of these reachable points, we then need to determine that location's maximum value of $\mathcal{K}$.

Determining the maximum value of $\mathcal{K}$ for a workspace location is complicated by the fact that one must consider all of the possible preimages for this location. Even determining the number of manifolds associated with a location, and whether these manifolds are open or closed, is not trivial. For example, consider the case illustrated in Fig. 2 that shows all the configurations for one optimal robot design at four distinct locations (identified by A, B, C, and D) that lie on a straight line segment within the workspace. (Points A and D represent the two endpoints of the line segment with B and C being 57% and 80% of the distance from A to D, respectively.) The possible configurations for being at point A are represented by two open manifolds, which then combine into a single closed manifold at point B, which then becomes two disjoint manifolds (as illustrated by point C), with one of them ultimately disappearing at point D. This illustrates that one cannot simply track a locally optimal value of $\mathcal{K}$ because these local optima may disappear, as they do in this example. In order to accurately estimate the maximum value of $\mathcal{K}$ for a workspace location, one must make sure to consider all of the possible configurations that are associated with all of the possible manifolds for that location.

To deal with this situation, we select several joint configurations (from the one million random configurations initially generated) whose workspace locations are near the reachable workspace point whose maximum value of $\mathcal{K}$ is being evaluated. This increases the probability that all self-motion manifolds associated with this workspace location will be represented. (See Fig. 3.) We then perform iterative inverse kinematics to drive each of these configurations exactly to the point being evaluated. Once the exact end-effector position is achieved, we

map out the entire self-motion manifold by stepping along the null vector of the Jacobian. As we compute the configurations along the manifold, we calculate each configuration's value of $\mathcal{K}$ and save the maximum. If the maximum value of $\mathcal{K}$ over all manifolds associated with this point is greater than $\gamma\ \mathcal{K}_{\max}$, then this reachable workspace point is included in the count for the fault-tolerant workspace volume, denoted $n_{\rm f}$.[3] The fraction of the total workspace that is fault tolerant, denoted $W_{\mathcal{K}}$, can then be easily estimated by the ratio $n_{\rm f}/n_{\rm r}$. The next section will use this measure to evaluate families of robots that have commonly used values for link twists.

V. Examples Using Manipulators With Common Link Twist Parameters

A. Background

When designing a robot's kinematics there are many factors that must be considered. These factors may limit the range of desirable values for the joint parameters. In this section, we use the example of where one may be interested in limiting the joint twist values, i.e., setting α_i's to $\pm 90^\circ, 0^\circ$, or 180° as is common in many commercial manipulators. (Because α_4 is already set to zero [see (18)], we only need to consider $i = 1, 2, 3$.)

Recall that the parameter α_i is defined as the angle between the rotation axes of joints i and $i+1$, which is the same as ω_i and ω_{i+1}, respectively. Therefore, one can use the dot product between the appropriate pair of equations (11) and (13)–(15) to determine the values of the β_i's that result in the desired α_i's.

B. Link Twist $\alpha_i = \pm 90^\circ$

To determine the relationship between β_i and β_{i+1} for $\alpha_i = \pm 90^\circ$, one can solve for the case where the dot product between ω_i and ω_{i+1} is equal to zero. In the case of $\alpha_1 = \pm 90^\circ$, using (11) and (13) results in

$$\beta_2 = -\arctan\left(\frac{1}{3\tan(\beta_1)}\right) + k\pi \tag{25}$$

where $k = 0, 1$, i.e., one value of k results in $\alpha_1 = 90^\circ$ and the other results in $\alpha_1 = -90^\circ$. Because β_1 is an arbitrary parameter, there is a set of solutions that are parameterized by β_1. However, the space of all possible robots is spanned by $0 \le \beta_1 < 180$. (This is because when $180 \le \beta_1 < 360$, the resulting robots are mirror versions of those obtained when $0 \le \beta_1 < 180$.)

Similarly, we can find the solutions of when α_2 and α_3 are equal to $\pm 90^\circ$ using (13)–(15) so that

$$\beta_3 = \arctan\left(\frac{\sqrt{3} + 3\tan(\beta_2)}{5 + \sqrt{3}\tan(\beta_2)}\right) + k\pi \tag{26}$$

and

$$\beta_4 = -\arctan\left(\frac{3 + \sqrt{3}\tan(\beta_3)}{5\tan(\beta_3) + \sqrt{3}}\right) + k\pi \tag{27}$$

where $k = 0, 1$.

Note that when $k = 1$, the direction of ω_{i+1} flips from what it was when $k = 0$, and thus the sign of α_i will change. However, one can add 180° to β_i with $k = 1$ so that both ω_i and ω_{i+1} are flipped, and the sign of α_i will stay the same. Thus, there will be two sets of values for β_2 and β_3 that will satisfy (26) and result in $\alpha_2 = 90^\circ$, and two sets of values for β_2 and β_3 that will satisfy (26) and result in $\alpha_2 = -90^\circ$. The same is true for (27).

[3]Clearly, if one finds a configuration where $\mathcal{K} \ge \gamma\ \mathcal{K}_{\max}$, then there is no need to continue evaluating the manifold. Likewise, there is no need to evaluate a manifold multiple times.

TABLE I
Values of β_i and β_{i+1} to Make $\alpha_i = 0^\circ$ or 180°

i	α_i [degrees]	$(\beta_i,\ \beta_{i+1})$ [degrees]
1	0	(90, 90)
	180	(90, 270)
2	0	(30, 120), (210, 300)
	180	(30, 300), (210, 120)
3	0	(60, 60), (240, 240)
	180	(60, 240), (240, 60)

TABLE II
Size of Robot Groups With All Possible Combinations of α_i Being $\pm 90^\circ, 0^\circ$, or 180°

Robot Group	Relationship between joint axes $i-1$ and i $i = (1, 2, 3, 4)$	Size of Group
1	($\parallel$, $\parallel$, $\parallel$, $\parallel$)	0
2	($\parallel$, $\parallel$, $\perp$, $\parallel$)	0
3	($\perp$, $\parallel$, $\parallel$, $\parallel$)	0
4	($\parallel$, $\perp$, $\parallel$, $\parallel$)	8
5	($\parallel$, $\perp$, $\perp$, $\parallel$)	8
6	($\perp$, $\parallel$, $\perp$, $\parallel$)	8
7	($\perp$, $\perp$, $\parallel$, $\parallel$)	8
8	($\perp$, $\perp$, $\perp$, $\parallel$)	∞

C. Link Twist $\alpha_i = 0^\circ$ or 180°

To solve for α_i being equal to 0° or 180°, one can set the dot product of ω_i and ω_{i+1} to 1 or -1, respectively. In contrast to the subsection above, the solutions of these equations result in discrete values of β_i and β_{i+1}. Table I shows the values of β_i and β_{i+1} for all cases of $\alpha_i = 0^\circ$ or $\alpha_i = 180^\circ$. Note that there are two sets of β_i and β_{i+1} for a given desired α_i, except for α_1 where there is only one set of β_1 and β_2. (This is because the other two sets of β_1 and β_2 result in mirror versions of the corresponding robots.)

It should be noted that it is not possible to arbitrarily set α_i's to be 0° and/or 180° at the same time. For example, setting $\alpha_2 = 0^\circ$ requires $\beta_2 = 30^\circ$ or 210°, which is not consistent with the values that are required to make $\alpha_1 = 0^\circ$. It also requires $\beta_3 = 120^\circ$ or 300°, so that it is not possible to make $\alpha_3 = 0^\circ$. By comparing all of the constraints on the values of β_i and β_{i+1} for $i = 1, 2$, it can be concluded that it is not possible to have $\alpha_i = 0^\circ$ and also have α_{i+1} be either 0° or 180°. The same is true for $\alpha_i = 180^\circ$.

D. Manipulator Categories

The above subsection provide the equations for determining the required β_i and β_{i+1} values to achieve a desired α_i, i.e., solving equation (16) for $\alpha_i = \pm 90^\circ, 0^\circ$, and 180°. However, because selecting a desired α_i restricts the range of possible values for β_i *and* β_{i+1} it is not possible to arbitrarily select all three of the α_i values. For example, if one selects the value of α_1 and α_3, then the values of $\beta_1, \beta_2, \beta_3$, and β_4 are all specified so that the choices for α_2 are limited. In this subsection, we determine if a particular combination of α_i's results in a feasible kinematic design with the desired fault-tolerant Jacobian and, if so, how many such designs exist.

Even with restricting α_i to $\pm 90^\circ, 0^\circ$, or 180°, there still exists a large number of possible robots. In order to analyze them further, we organize them into groups. Because $\alpha_4 = 0$, the total number of different combinations of setting α_i to one of these four values is $4^3 = 64$. We organize these 64 combinations into eight robot groups, based on whether an α_i results in adjacent joint axes being parallel ($\parallel$) or perpendicular ($\perp$), i.e., whether $\alpha_i = 0^\circ, 180^\circ$, or $\alpha_i = \pm 90^\circ$,

respectively. Table II enumerates these eight groups, along with the total number of robots that they include.

Note that some robot structures are not feasible because they are not physically able to result in the optimally fault-tolerant Jacobian given by (5). For example, robots in group 1 have all of their joints parallel so that they result in planar manipulators, which clearly are not capable of the desired Jacobian. Likewise, robots in groups 2 and 3 result in planar substructures that make it physically impossible to achieve the desired Jacobian.

For cases where there are not three joint axes in parallel, one is able to identify multiple feasible robot designs. For groups 4–7, there exist eight unique robots in each group. To determine all of the β_i parameters that result in the specified joint twist values (α_i's) one can use Table I and (25)–(27). In some cases, the inverses of (25) and (26), i.e.,

$$\beta_1 = -\arctan\left(\frac{1}{3\tan(\beta_2)}\right) \quad \text{and} \tag{28}$$

$$\beta_2 = -\arctan\left(\frac{\sqrt{3}-5\tan(\beta_3)}{3-\sqrt{3}\tan(\beta_3)}\right) + k\pi \tag{29}$$

where $k = 0, 1$, are useful. Once all the β_i parameters are determined, then J_w is defined and one can calculate the remaining DH parameters that describe the robot.

We illustrate this procedure with a specific example. Consider robot group 6. Because joint axes 2 and 3 are parallel, i.e., $\alpha_2 = 0^\circ, 180^\circ$, β_2 and β_3 are constrained to the discrete values given in Table I. Therefore, our strategy is to start with each of these possible values and then evaluate (28) and (27) to determine the required values of β_1 and β_4. Specifically, for $\alpha_2 = 0^\circ, (\beta_2, \beta_3) = (30^\circ, 120^\circ)$ or $(210^\circ, 300^\circ)$. Then, to set α_1 and α_3 to $\pm 90^\circ$, we use (28) and (27), respectively. This results in

$$\begin{aligned}(\beta_1, \beta_2, \beta_3, \beta_4) &= (150^\circ, 30^\circ, 120^\circ, 0^\circ) \quad \text{or} \\ &= (150^\circ, 30^\circ, 120^\circ, 180^\circ) \text{ or} \\ &= (150^\circ, 210^\circ, 300^\circ, 0^\circ) \quad \text{or} \\ &= (150^\circ, 210^\circ, 300^\circ, 180^\circ)\end{aligned} \tag{30}$$

so that there are four possible robot combinations. One can apply the analogous procedure for $\alpha_2 = 180^\circ$, which also results in four robots so that the size of this group is eight.

Finally, robot group 8 is unique in that it is parameterized by β_1 using (25)–(27), and so there are an unlimited number of robots in this group. In the next subsection, we evaluate all of the robots from these groups to identify an optimal design in terms of its global fault-tolerance capabilities.

E. Global Fault Tolerance Analysis

We computed the global fault-tolerant properties for all of the robots represented in Table II. For robot groups 4–7, this simply involved evaluating each of the possible robot designs for $W_{\mathcal{K}}$. (Recall that $W_{\mathcal{K}}$ is the percentage of the workspace that has a fault-tolerance value that is greater than or equal to $\gamma = 0.9$ of the maximum.) For each group, we determined the robot (in terms of its β_i's parameters) that had the largest value of $W_{\mathcal{K}}$. We show these results in Table III along with a depiction of the resulting robots in the optimal fault-tolerant design configuration in Fig. 4. Note that the best robot from each group varies from a maximum of $W_{\mathcal{K}} = 67\%$ to a minimum of 30%.

In contrast, determining the best robot from group 8 required an optimization over the independent variable β_1. However, this resulted in by far the best global fault-tolerant robot with a $W_{\mathcal{K}} = 75\%$ for the robot

TABLE III
BEST ROBOT CONFIGURATIONS AMONG THE INDIVIDUAL ROBOT GROUPS 4–7 THAT HAVE ONE ADDITIONAL $\alpha_i = 0$ IN TABLE II

Robot Group	$(\alpha_1,\ \alpha_2,\ \alpha_3,\ \alpha_4)$ [degrees]	$(\beta_1,\ \beta_2,\ \beta_3,\ \beta_4)$ [degrees]	V_{r} [m^3]	$W_{\mathcal{K}}$ [%]
6	$(-90,\ 0,\ -90,\ 0)$	$(150,\ 30,\ 120,\ 180)$	251	67
4	$(0,\ 90,\ 0,\ 0)$	$(90,\ 90,\ 240,\ 240)$	316	59
7	$(90,\ 90,\ 0,\ 0)$	$(0,\ 270,\ 240,\ 240)$	175	48
5	$(0,\ 90,\ 90,\ 0)$	$(90,\ 90,\ 240,\ 330)$	227	30

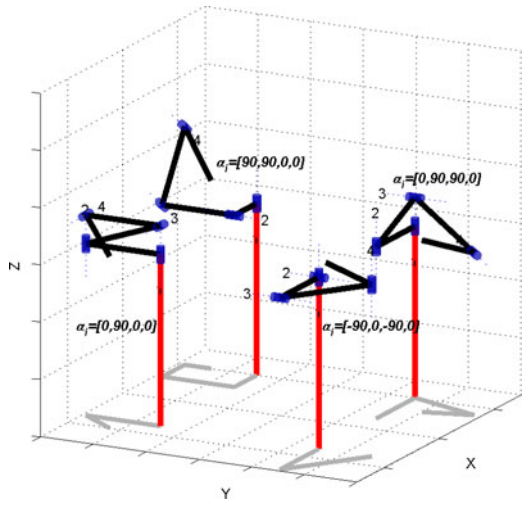

Fig. 4. Optimal fault-tolerant configurations at the design point of Table III robots. (The figure was generated using the Robotics Toolbox described in [49].)

TABLE IV
DH PARAMETERS OF THE GLOBALLY OPTIMAL ROBOT THAT CORRESPONDS TO $(\beta_1, \beta_2, \beta_3, \beta_4) = (30^\circ, 330^\circ, 0^\circ, 300^\circ)$

i	α_i [degrees]	a_i [m]	d_i [m]	θ_i [degrees]
1	90	$\sqrt{2}$	0	0
2	-90	$\sqrt{2}$	1	180
3	90	$\sqrt{2}$	-1	180
4	0	$\sqrt{3}/2$	$1/2$	145

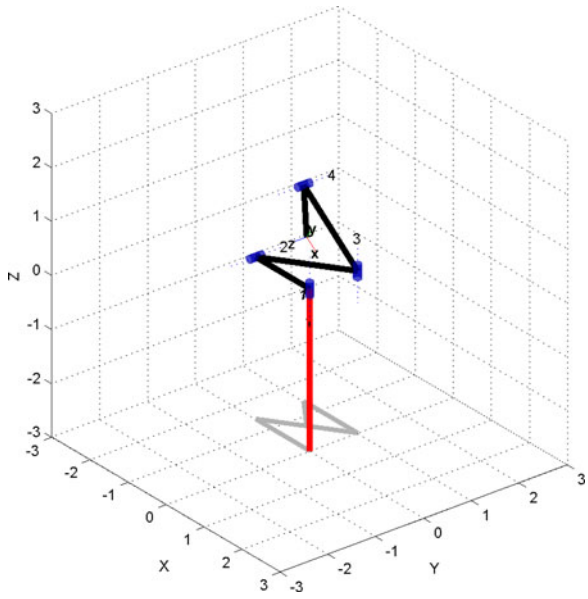

Fig. 5. Locally optimal fault-tolerant configuration at the design point of the globally optimal robot defined in Table IV. (The figure was generated using the Robotics Toolbox described in [49].)

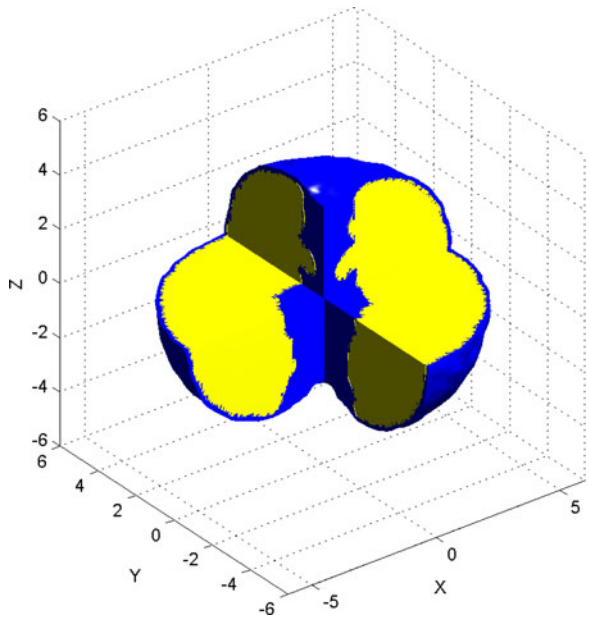

Fig. 6. Workspace volumes of the globally optimal robot defined in Table IV. (Blue) Total reachable workspace volume. (Yellow) Fault-tolerant workspace volume.

given by $(\beta_1, \beta_2, \beta_3, \beta_4) = (30^\circ, 330^\circ, 0^\circ, 300^\circ)$, which corresponds to

$$J_\omega = \begin{bmatrix} 0 & \sqrt{\frac{2}{3}} & 0 & -\sqrt{\frac{2}{3}} \\ \sqrt{\frac{3}{4}} & \sqrt{\frac{1}{12}} & -\sqrt{\frac{3}{4}} & -\sqrt{\frac{1}{12}} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{bmatrix}. \tag{31}$$

The DH parameters for this robot are given in Table IV and an image of the robot in its optimal design configuration is shown in Fig. 5, where the base coordinate frame has been rotated to align the first joint axis with the z-axis, resulting in the following Jacobian

$$J_{6\times 4} = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ \hline 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix}. \tag{32}$$

This globally optimal fault-tolerant robot design has a total maximum reach and reachable volume, V_r, of 5.5 m and 560 m^3, respectively. An illustration of both the reachable and the fault-tolerant workspace are given in Fig. 6, where the three-dimensional volume is shown with multiple cross sections to better visualize the two different volumes. Clearly, one can see that a $W_\mathcal{K} = 75\%$ results in a significant amount of the reachable volume being fault tolerant as well.

F. Jacobian Column Permutation and/or Sign Change Effects

The analysis so far has been based on the positional Jacobian J_v given in (5). The family of manipulator Jacobians with a compatible orientational Jacobian J_ω was characterized based on parameterizing the rotation axes with $\beta_i, i = 1, 2, 3, 4$ given in equations (11) and (13)–(15). There are $4! \times 2^4 = 384$ different versions of (5), where $4! = 24$ represents the total number of permutation operations, and 2^4 is the number of all possible cases of alternating the column signs. Performing these operations, particularly permuting the columns, on an arbitrary Jacobian will typically result in a Jacobian that corresponds to a completely different manipulator [41], [42]. However, as these operations do not collectively change the family of fault-tolerant robots identified using (5), the robot geometries do not essentially change. Consequently, one need only consider the unmodified Jacobian described by (5) with J_ω as a function of the β_i's to optimize the whole family of robots. Fig. 7 illustrates the kinematic equivalency of these various robots.

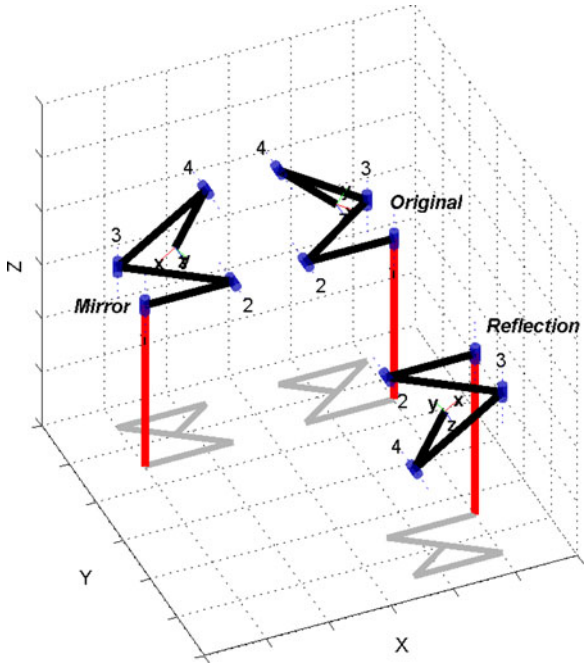

Fig. 7. Kinematically equivalent robots generated from a permutation of the columns (and/or multiplication of ± 1) of the Jacobian. (The robot labeled "original" is the same as in Fig. 5.)

VI. Conclusion

This work has shown that one can parameterize the infinite family of four-DOF spatial positioning manipulators that correspond to an optimally failure-tolerant Jacobian. We then presented a method for evaluating the global fault-tolerant properties of the resulting manipulator designs and used this measure to illustrate how one would optimize the kinematic design for a given family of manipulators to obtain a robot that has a high degree of failure tolerance over its entire workspace, in addition to being locally optimal.

References

[1] E. C. Wu, J. C. Hwang, and J. T. Chladek, "Fault-tolerant joint development for the space shuttle remote manipulator system: Analysis and experiment," *IEEE Trans. Robot. Automat.*, vol. 9, no. 5, pp. 675–684, Oct. 1993.

[2] G. Visentin and F. Didot, "Testing space robotics on the Japanese ETS-VII satellite," ESA Bull.-Eur. Space Agency, vol. 99, pp. 61–65, Sep. 1999.

[3] P. S. Babcock and J. J. Zinchuk, "Fault-tolerant design optimization: Application to an autonomous underwater vehicle navigation system," in

Proc. Symp. Autonom. Underwater Vehicle Technol., Washington, DC, USA, Jun. 5–6, 1990, pp. 34–43.
[4] R. Colbaugh and M. Jamshidi, "Robot manipulator control for hazardous waste-handling applications," *J. Robot. Syst.*, vol. 9, no. 2, pp. 215–250, 1992.
[5] W. H. McCulloch, "Safety analysis requirements for robotic systems in DOE nuclear facilities," in *Proc. 2nd Specialty Conf. Robot. Challeng. Environ.*, Albuquerque, NM, USA, Jun. 1–6, 1996, pp. 235–240.
[6] H. Nakata, "Domestic robots failed to ride to rescue after No. 1 plant blew," *Jpn. Times*, Jan. 6, 2012.
[7] K. N. Groom, A. A. Maciejewski, and V. Balakrishnan, "Real-time failure-tolerant control of kinematically redundant manipulators," *IEEE Trans. Robot. Autom.*, vol. 15, no. 6, pp. 1109–1116, Dec. 1999.
[8] Reliability Information Analysis Center, "Nonelectronic parts reliability data," Defense Technical Information Center / Air Force Research Lab, Rome, NY, USA, no. NPRD-2011, 2011.
[9] B. S. Dhillon A. R. M. Fashandi and K. L. Liu, "Robot systems reliability and safety: A review," *J. Qual. Mainten. Eng.*, vol. 8, no. 3, pp. 170–212, 2002.
[10] Y. Ting, S. Tosunoglu, and B. Fernandez, "Control algorithms for fault-tolerant robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 8–13, 1994, vol. 2, pp. 910–915.
[11] M. L. Visinsky, J. R. Cavallaro, and I. D. Walker, "A dynamic fault tolerance framework for remote robots," *IEEE Trans. Robot. Automat.*, vol. 11, no. 4, pp. 477–490, Aug. 1995.
[12] J. D. English and A. A. Maciejewski, "Fault tolerance for kinematically redundant manipulators: Anticipating free-swinging joint failures," *IEEE Trans. Robot. Automat.*, vol. 14, no. 4, pp. 566–575, Aug. 1998.
[13] J. D. English and A. A. Maciejewski, "Failure Tolerance through active braking: A kinematic approach," *Int. J. Robot. Res.*, vol. 20, no. 4, pp. 287–299, Apr. 2001.
[14] 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," *Fus. Eng. Design*, vol. 84, no. 7, pp. 1420–1424, 2009.
[15] D. L. Schneider, D. Tesar, and J. W. Barnes, "Development & testing of a reliability performance index for modular robotic systems," in *Proc. Annu. Rel. Maintain. Symp.*, Anaheim, CA, USA, Jan. 24–27, 1994, pp. 263–271.
[16] C. J. J. Paredis and P. K. Khosla, "Designing fault-tolerant manipulators: How many degrees of freedom?," *Int. J. Robot. Res.*, vol. 15, no. 6, pp. 611–628, Dec. 1996.
[17] S. Tosunoglu and V. Monteverde, "Kinematic and structural design assessment of faul-tolerant manipulators," *Intell. Automat. Soft Comput.*, vol. 4, no. 3, pp. 261–268, 1998.
[18] C. Carreras and I. D. Walker, "Interval methods for fault-tree analysis in robotics," *IEEE Trans. Robot. Automat.*, vol. 50, no. 1, pp. 3–11, Mar. 2001.
[19] L. Notash, "Joint sensor fault detection for fault tolerant parallel manipulators," *J. Robot. Syst.*, vol. 17, no. 3, pp. 149–157, 2000.
[20] M. Leuschen, I. Walker, and J. Cavallaro, "Fault residual generation via nonlinear analytical redundancy," *IEEE Trans. Control Syst. Tech.*, vol. 13, no. 3, pp. 452–458, May 2005.
[21] M. Anand, T. Selvaraj, S. Kumanan, and J. Janarthanan, "A hybrid fuzzy logic artificial neural network algorithm-based fault detection and isolation for industrial robot manipulators," *Int. J. Manuf. Res.*, vol. 2, no. 3, pp. 279–302, 2007.
[22] D. Brambilla, L. Capisani, A. Ferrara, and P. Pisu, "Fault detection for robot manipulators via second-order sliding modes," *IEEE Trans. Ind. Electron.*, vol. 55, no. 11, pp. 3954–3963, Nov. 2008.
[23] J. Park, W.-K. Chung, and Y. Youm, "Failure recovery by exploiting kinematic redundancy," in *Proc. 5th Int. Workshop Robot Human Commun.*, Tsukuba, Japan, Nov. 11–14, 1996, pp. 298–305.
[24] X. Chen and S. Nof, "Error detection and prediction algorithms: Application in robotics," *J. Intell. Robot. Syst.; Robot. Syst.*, vol. 48, no. 2, pp. 225–252, 2007.
[25] M. Ji and N. Sarkar, "Supervisory fault adaptive control of a mobile robot and its application in sensor-fault accommodation," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 174–178, Feb. 2007.
[26] A. De Luca and L. Ferrajoli, "A modified Newton-Euler method for dynamic computations in robot fault detection and control," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2009, pp. 3359–3364.
[27] J. E. McInroy, J. F. O'Brien, and G. W. Neat, "Precise, fault-tolerant pointing using a Stewart platform," *IEEE/ASME Trans. Mechatronics*, vol. 4, no. 1, pp. 91–95, Mar. 1999.
[28] M. Hassan and L. Notash, "Optimizing fault tolerance to joint jam in the design of parallel robot manipulators," *Mech. Mach. Theory*, vol. 42, no. 10, pp. 1401–1417, 2007.
[29] Y. Chen, J. E. McInroy, and Y. Yi, "Optimal, fault-tolerant mappings to achieve secondary goals without compromising primary performance," *IEEE Trans. Robot.*, vol. 19, no. 4, pp. 680–691, Aug. 2003.
[30] Y. Yi, J. E. McInroy, and Y. Chen, "Fault tolerance of parallel manipulators using task space and kinematic redundancy," *IEEE Trans. Robot.*, vol. 22, no. 5, pp. 1017–1021, Oct. 2006.
[31] J. E. McInroy and F. Jafari, "Finding symmetric orthogonal Gough–Stewart platforms," *IEEE Trans. Robot.*, vol. 22, no. 5, pp. 880–889, Oct. 2006.
[32] C. L. Lewis and A. A. Maciejewski, "Dexterity optimization of kinematically redundant manipulators in the presence of failures," *Comput. Electr. Eng.*, vol. 20, no. 3, pp. 273–288, May 1994.
[33] A. A. Maciejewski, "Fault tolerant properties of kinematically redundant manipulators," in *Proc. IEEE Int. Conf. Robot. Automat.*, Cincinnati, OH, USA, May 13–18, 1990, pp. 638–642.
[34] R. G. Roberts and A. A. Maciejewski, "A local measure of fault tolerance for kinematically redundant manipulators," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 543–552, Aug. 1996.
[35] R. G. Roberts, H. G. Yu, and A. A. Maciejewski, "Fundamental limitations on designing optimally failure-tolerant kinematically redundant manipulators," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1124–1237, Oct. 2008.
[36] R. G. Roberts, "On the local fault tolerance of a kinematically redundant manipulator," *J. Robot. Syst.*, vol. 13, no. 10, pp. 649–661, Oct. 1996.
[37] C. J. J. Paredis and P. K. Khosla, "Fault tolerant task execution through global trajectory planning," *Rel. Eng. Syst. Safety*, vol. 53, pp. 225–235, 1996.
[38] C. L. Lewis and A. A. Maciejewski, "Fault tolerant operation of kinematically redundant manipulators for locked joint failures," *IEEE Trans. Robot. Autom.*, vol. 13, no. 4, pp. 622–629, Aug. 1997.
[39] R. S. Jamisola, Jr., A. A. Maciejewski, and R. G. Roberts, "Failure-tolerant path planning for kinematically redundant manipulators anticipating locked-joint failures," *IEEE Trans. Robot.*, vol. 22, no. 4, pp. 603–612, Aug. 2006.
[40] R. G. Roberts, R. J. Jamisola, Jr., and A. A. Maciejewski, "Identifying the failure-tolerant workspace boundaries of a kinematically redundant manipulator," in *Proc. IEEE Int. Conf. Robot. Autom.*, Rome, Italy, Apr. 10–14, 2007, pp. 4517–4523.
[41] K. M. Ben-Gharbia, A. A. Maciejewski, and R. G. Roberts, "An illustration of generating robots from optimal fault-tolerant Jacobians," in *Proc. 15th IASTED Int. Conf. Robot. Appl.*, Cambridge, MA, USA, Nov. 1–3, 2010, pp. 461–468.
[42] K. M. Ben-Gharbia, R. G. Roberts, and A. A. Maciejewski, "Examples of planar robot kinematic designs from optimally fault-tolerant Jacobians," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 9–13, 2011, pp. 4710–4715.
[43] K. M. Ben-Gharbia, A. A. Maciejewski, and R. G. Roberts, "Examples of spatial positioning redundant robotic manipulators that are optimally fault tolerant," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Anchorage, AK, USA, Oct. 9–12, 2011, pp. 1526–1531.
[44] T. Yoshikawa, "Manipulability of robotic mechanisms," *Int. J. Robot. Res.*, vol. 4, no. 2, pp. 3–9, 1985.
[45] C. A. Klein and B. E. Blaho, "Dexterity measures for the design and control of kinematically redundant manipulators," *Int. J. Robot. Res.*, vol. 6, no. 2, pp. 72–83, 1987.
[46] C. Klein and T. A. Miklos, "Spatial robotic isotropy," *Int. J. Robot. Res.*, vol. 10, no. 4, pp. 426–437, 1991.
[47] K. E. Zanganeh and J. Angeles, "Kinematic isotropy and the optimum design of parallel manipulators," *Int. J. Robot. Res.*, vol. 16, no. 2, pp. 185–197, Apr. 1997.
[48] A. A. Maciejewski and R. G. Roberts, "On the existence of an optimally failure tolerant 7R manipulator Jacobian," *Appl. Math. Comput. Sci.*, vol. 5, no. 2, pp. 343–357, 1995.
[49] P. I. Corke, "A robotics toolbox for MATLAB," *IEEE Robot. Autom. Mag.*, vol. 3, no. 1, pp. 2–32, Mar. 1996.