Kinematic Design of Manipulators with Seven Revolute Joints Optimized for Fault Tolerance

Khaled M. Ben-Gharbia, *Member, IEEE*, Anthony A. Maciejewski, *Fellow, IEEE*, and Rodney G. Roberts, *Senior Member, IEEE*

***Abstract*—A local definition of fault tolerance, based on properties of the manipulator Jacobian, is used to generate the kinematics of seven degree-of-freedom (DOF) revolute joint manipulators. The measure of fault tolerance used is the smallest singular value over all possible Jacobians resulting from single locked joint failures. The canonical form for an optimal fault-tolerant Jacobian that maximizes this measure has been previously identified. It has also been known that it is not possible to generate a seven DOF revolute manipulator that corresponds to this theoretically optimal Jacobian. However, in this paper, it is shown how to generate physically realizable Jacobians that are very close to being optimal. It is further shown that there exist 7! different manipulators, from a single Jacobian, that have the same local fault tolerance properties. To evaluate the global properties of these different manipulators, a technique for computing six-dimensional fault-tolerant workspaces is presented. The size of these workspaces vary significantly among these 7! manipulators.**

***Index Terms*—Fault-tolerant robots, locked joint failure, redundant robots, robot kinematics, workspace analysis.**

I. Introduction

TO MAXIMIZE the usefulness of robots applied in remote or hazardous environments one must consider how such robots should be operated before, during, and after failures. This is due to the fact that component failures are common [1]–[3] in applications such as space exploration [4], [5], underwater exploration [6], and nuclear waste remediation [7]. One notable example is the Fukushima nuclear reactor accident [8], [9]. Most component failures either result in a locked joint failure mode [10] or are transformed into this mode by failure recovery mechanisms that employ fail-safe brakes [11], e.g., free-swinging failures [12], [13].

Some aspects of previous work on dealing with failures include improving manipulator reliability [3], designing fault-tolerant robots [14]–[18], and determining mechanisms for analyzing [19], detecting [10], [20]–[22], identifying [23], [24], and recovering [25]–[27] from failures. One prerequisite for maintaining the ability to perform general motion after a joint failure is kinematic redundancy [28]–[30]. This redundancy impacts the performance of the manipulator both locally [31]–[34] and in terms of its workspace [35]–[37].

In this paper, it is assumed that the desired local performance of a manipulator, as described by its Jacobian, is specified. In particular, one would like to maximize the capability of the manipulator for a wide range of tasks. Therefore, prior to a failure, an optimal Jacobian would be one that is isotropic. After a locked joint failure, one would like to maintain as much of this capability as possible. We mathematically define this concept of fault tolerance and identify its optimal Jacobian in the next section. Once one has identified a desired optimal Jacobian, there will be a family of different manipulator kinematics that locally possess the properties of this Jacobian [38], [39]. To evaluate the properties of these different designs away from the optimal configuration, we designed two algorithms for computing a measure of fault tolerance throughout a six-dimensional workspace volume. We then use these algorithms to compare the global characteristics of the large number of seven degree-of-freedom (DOF) manipulators that possess the same locally optimal fault-tolerant Jacobian.

The remainder of this paper is organized as follows. In the next section we review a mathematical definition of local failure tolerance that is based on desirable properties of the manipulator Jacobian. In Section III, we describe how one can generate physically realizable families of robots that correspond as closely as possible to the properties of the optimal Jacobian. The following section presents algorithms for computing the fault tolerance measures of a robot design throughout its six-dimensional workspace. In Section V, we explore the large number (7!) of seven DOF manipulators that correspond to a single Jacobian and identify correlations between kinematic parameters and fault-tolerant properties. We then use the algorithms from Section IV to evaluate and compare their six-dimensional fault-tolerant workspaces. The conclusions of this paper 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 as presented in [40].

Manuscript received March 23, 2015; accepted August 8, 2015. Date of publication November 30, 2015; date of current version September 14, 2016. This work was supported in part by the National Science Foundation (NSF) under Contract IIS-0812437. This research utilized the CSU ISTeC Cray HPC System supported by NSF Grant CNS-0923386. This paper was recommended by Associate Editor S. Nahavandi.

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/TSMC.2015.2497439

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 degrees of freedom 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

$$J_{m\times n} = \begin{bmatrix} j_1 & j_2 & \cdots & j_n \end{bmatrix} \tag{1}$$

where j_i represents the end-effector velocity due to the velocity of joint i. For an arbitrary single locked joint failure at joint f, 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

$${}^{f}J_{m\times(n-1)} = \begin{bmatrix} j_1 & j_2 & \cdots & j_{f-1} & j_{f+1} & \cdots & j_n \end{bmatrix}. \tag{2}$$

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), sum (trace), or ratio (condition number) [41]. 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 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 ${}^{f}J$, ${}^{f}\sigma_m$ 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

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

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 [1]. 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 [41]. 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 [31] that the worst-case dexterity of an isotropic manipulator that experiences a single joint failure is governed by the inequality

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

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 degrees of freedom.

Using the above definition of an optimally failure tolerant configuration one can identify the structure of the Jacobian required to obtain this property [42].[1] In particular, one can show that the optimally failure tolerant criterion requires that each joint contributes equally to the null space of the Jacobian transformation [32], [33]. 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 paper, 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. The second condition is equivalent to the columns of the Jacobian having equal norms.

For the case of a seven DOF fully spatial manipulator, the canonical structure of an optimally failure tolerant configuration is given by [42]

$$J = \begin{bmatrix} -\sqrt{\frac{6}{7}} & \sqrt{\frac{1}{42}} & \sqrt{\frac{1}{42}} & \sqrt{\frac{1}{42}} & \sqrt{\frac{1}{42}} & \sqrt{\frac{1}{42}} & \sqrt{\frac{1}{42}} \\ 0 & -\sqrt{\frac{5}{6}} & \sqrt{\frac{1}{30}} & \sqrt{\frac{1}{30}} & \sqrt{\frac{1}{30}} & \sqrt{\frac{1}{30}} & \sqrt{\frac{1}{30}} \\ 0 & 0 & -\sqrt{\frac{4}{5}} & \sqrt{\frac{1}{20}} & \sqrt{\frac{1}{20}} & \sqrt{\frac{1}{20}} & \sqrt{\frac{1}{20}} \\ 0 & 0 & 0 & -\sqrt{\frac{3}{4}} & \sqrt{\frac{1}{12}} & \sqrt{\frac{1}{12}} & \sqrt{\frac{1}{12}} \\ 0 & 0 & 0 & 0 & -\sqrt{\frac{2}{3}} & \sqrt{\frac{1}{6}} & \sqrt{\frac{1}{6}} \\ 0 & 0 & 0 & 0 & 0 & -\sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \end{bmatrix}. \tag{5}$$

Unfortunately, no manipulator that consists of seven revolute joints can possess this Jacobian. In the next section, we will discuss how one can determine such manipulators whose Jacobians have properties that are close to optimal.

III. Generating Robot Designs

Although the canonical form in (5) has the desirable property of fault tolerance, it corresponds to a manipulator possessing three prismatic joints and four joints that are capable of an arbitrary screw motion [42]. The columns of a manipulator Jacobian for a robot consisting of only revolute joints have a more restrictive algebraic structure. In particular, the vector ω_i corresponding to the last three components of a column j_i must have unit length and must be perpendicular to the vector v_i consisting of the first three components of that column. Furthermore, the isotropy condition and the condition that the columns of J have equal norms require that the v_i's also have unit norm. Including this additional constraint on the v_i's gives a total of 21 constraints corresponding to $\|v_i\| = 1$, $\|\omega_i\| = 1$, and $v_i \cdot \omega_i = 0$ for $i = 1, 2, \ldots, 7$. If an isotropic configuration exists, then it follows that $\sigma = \sqrt{7/3} \approx 1.5275$. As noted in [42], we were not able to find an isotropic revolute

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

manipulator Jacobian for which (4) achieves its upper bound. Instead, we identified manipulator Jacobians that were close to being ideally fault-tolerant.

First, we determined a manipulator Jacobian satisfying the 21 constraints on the columns that was closest to satisfying the isotropy condition in the sense described in [42]. In this case, the objective function was the sum of the squares of the 21 unique constraints given by $JJ^T - \frac{n}{3}I = 0$. This resulted in an optimal manipulator Jacobian

$$J = \begin{bmatrix} 1 & 0.43 & 0.75 & -0.54 & 0.14 & 0.33 & -0.38 \\ 0 & -0.60 & 0.65 & 0.46 & -0.79 & -0.19 & -0.81 \\ 0 & -0.67 & -0.14 & -0.70 & 0.60 & -0.93 & -0.46 \\ 0 & 0.77 & 0.15 & 0.84 & 0.58 & -0.69 & -0.43 \\ 1 & -0.15 & -0.36 & 0.33 & -0.42 & -0.72 & 0.59 \\ 0 & 0.62 & -0.92 & -0.43 & -0.69 & -0.10 & -0.68 \end{bmatrix}. \tag{6}$$

This Jacobian has a $\mathcal{K} = 0.5196$, whereas the optimal value of $\mathcal{K}$ from (4) is 0.5774. Furthermore, while it does minimize the objective function, the value of the objective function was not zero and hence (6) is not isotropic. However, its singular values only range between $\sigma_1 = 1.5829$ and $\sigma_m = \sigma_6 = 1.4726$. This compares reasonably well to the Jacobian in (5) that is isotropic with $\sigma = 1.5275$.

We next determined a manipulator Jacobian by maximizing $\mathcal{K}$ subject to all of the ${}^f\sigma_6$ for $f = 1, 2, \ldots, 7$ being equal and all of the components of the null vector having the same magnitude, using a solution to the 21 equations as an initial condition. One resulting solution is

$$J = \begin{bmatrix} 0 & -0.28 & -0.42 & 0.89 & -0.28 & -0.72 & 0.80 \\ 1 & -0.13 & -0.91 & 0.14 & -0.45 & 0.648 & -0.29 \\ 0 & 0.95 & 0.07 & -0.43 & -0.85 & -0.26 & 0.52 \\ 0 & -0.85 & 0.79 & -0.003 & -0.85 & 0.52 & 0.39 \\ 0 & -0.42 & -0.33 & -0.95 & 0.53 & 0.26 & 0.92 \\ 1 & -0.31 & 0.52 & -0.31 & 0.004 & -0.81 & -0.09 \end{bmatrix} \tag{7}$$

${}^f\sigma_6 = 0.5714$ for all $f = 1, 2, \ldots, 7$ and $n_J = \frac{1}{\sqrt{7}}[1\ 1\ 1\ 1\ 1\ 1\ 1]^T$. As a tradeoff $\sigma_1 = 1.6455$ and $\sigma_m = \sigma_6 = 1.4169$ deviate more than (6) from the isotropic value $\sigma = 1.5275$. In both approaches all calculated solutions converged to their respective optimal values.

Using the technique described in [43], one can generate the Denavit and Hartenberg (DH) parameters for a robot with a prescribed Jacobian. For example, Table I illustrates this for the Jacobian in (6) with the robot described by these parameters depicted in Fig. 1, which we will refer to as the robot from (6).

As pointed out in [38] and [39], the operation of postmultiplying a Jacobian by a permutation matrix that permutes the columns changes the physical parameters of the corresponding robot but does not affect its fault tolerance properties. Therefore, because we are designing seven DOF manipulators, there are 7! different robot designs.

TABLE I
DH PARAMETERS OF THE ROBOT FROM (6)

i	α_i [degrees]	a_i [m]	d_i [m]	θ_i [degrees]
1	-98	0.17	0	0
2	-114	1.42	1.67	62
3	-66	1.42	-0.69	126
4	50	0.56	-1.77	-28
5	-92	1.32	2.42	-172
6	-93	1.27	-0.38	88
7	0	1	0.95	152

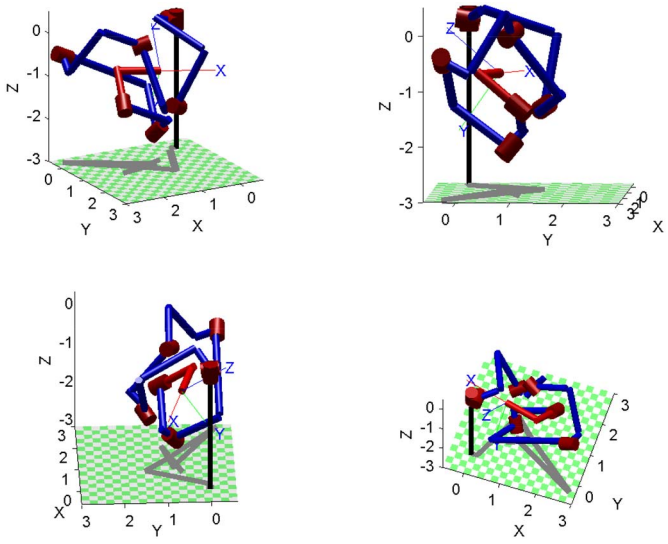

Fig. 1. Configuration of a robot that was generated to have the locally optimal fault-tolerant Jacobian in (6), referred to here as the robot from (6). (The figure was generated using the Robotics toolbox described in [44].)

IV. COMPUTING GLOBAL FAULT TOLERANCE PROPERTIES

A. Preliminaries

The above section shows that there are multiple Jacobians, and therefore multiple manipulator designs, that share the same local fault tolerance properties. To distinguish between them, one would select a specific Jacobian and then calculate the corresponding physical robot in order to evaluate its global properties, especially how the fault tolerance measure varies across the workspace. Even though we are designing a fully general spatial manipulator with a six-dimensional workspace consisting of both position and orientation, it is also useful to consider the three-dimensional maximum reachable workspace volume. Specifically, in this work, both the three-dimensional reachable volume and six-dimensional 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, are computed. (For all of the results shown in the following examples, $\gamma \approx 0.4$ is used and $\mathcal{K}_{\max} = 0.5196$ or 0.5714, depending on whether the Jacobian in (6) or (7) is used, respectively.) The most difficult portion of this calculation is computing the six-dimensional volume, which is discussed in the following subsection.

B. Calculating Six-Dimensional Volume

The six-dimensional workspace volume, denoted V_{6d}, of a fully spatial robotic manipulator can be determined by the integration of the orientation volume, denoted V_o and measured

in rad^3, over the reachable workspace volume, denoted V_r and measured in m^3. The six-dimensional workspace volume corresponding to a small volume element of the reachable workspace centered at the three-dimensional cartesian position x is approximately equal to

$$\Delta V_{6d} \approx V_o(x)\Delta V_r(x) \tag{8}$$

where $V_o(x)$ is the orientational volume corresponding to the point x and where ΔV_r is the volume of the small volume element containing the workspace point x. To obtain the six-dimensional workspace volume over the complete reachable workspace, we use a Riemann sum

$$V_{6d} \approx \sum_{i=1}^{N_r} V_o(P_i)\Delta V_r(P_i) = \frac{V_r}{N_r}\sum_{i=1}^{N_r} V_{o_i} \tag{9}$$

where the P_i are points contained in the individual volume elements determined by the integration grid, $V_{o_i} = V_o(P_i)$, and we assume in our case that $\Delta V_r(P_i) = V_r/N_r$, where N_r is the total number of grid (or sampled) points within the reachable workspace V_r. Note that V_{6d} is measured in units of m^3 rad^3.

The following subsection discusses two ways of computing the individual orientation volume segments V_{o_i}. In both cases, we use Monte Carlo integration with orientations represented by unit quaternions, denoted $q = [s,\ v_x,\ v_y,\ v_z]$. They differ in how the sampling is performed.

C. Calculating Orientation Volume

1) Parameterized Sampling of Quaternions: One simple way to sample orientations is to use spherical polar coordinates to parameterize unit quaternions [45], that is

$$\begin{aligned} s &= \cos(\psi) \\ v_x &= \sin(\psi)\cos(\phi) \\ v_y &= \sin(\psi)\sin(\phi)\cos(\theta) \\ v_z &= \sin(\psi)\sin(\phi)\sin(\theta) \end{aligned} \tag{10}$$

with $0 < \psi < \pi/2$, $0 < \phi < \pi$, and $0 < \theta < 2\pi$, to represent all possible orientations. When using this parameterization of quaternions to represent orientations, the volume integral element to calculate a reachable orientation volume is

$$\sin^2(\psi)\sin(\phi)d\psi d\phi d\theta. \tag{11}$$

To calculate the orientation volume V_{o_i} at a position P_i within a reachable workspace, we use Monte Carlo integration. To do so, we generate N_o quaternions whose spherical polar coordinates are uniformly distributed within the full ranges of ψ, ϕ, and θ. Each orientation is then evaluated to see if it is achievable, with the total denoted N_{o_i}. The orientation volume is then calculated using

$$V_{o_i} \approx \pi^3 \frac{1}{N_o}\sum_{j=1}^{N_{o_i}} \sin^2(\psi_j)\sin(\phi_j) \tag{12}$$

where ψ_j and ϕ_j are the spherical polar coordinates of achievable orientation j. Note that the maximum orientation volume is π^2, which can be obtained from integrating (11) over the total range of ψ, ϕ, and θ.

To improve the accuracy we estimate the achievable range of ψ, ϕ, and θ using a low-resolution sampling. We then resample at a higher resolution within the restricted range.[2]

2) Uniformly Sampling Unit Quaternions: Rather than using a parameterized sampling of quaternions, one can directly sample a sphere in four-dimensional space, i.e., a 3-sphere, to generate N_o uniformly distributed quaternions on its surface. Even though the surface area of a 3-sphere is given by $2\pi^2$, we only need half of the surface to represent uniquely all possible orientations. This is because, for a unit quaternion, we only need the scalar component to be in the range $0 \le s \le 1$, while the elements of the axis of rotation, v_x, v_y, and v_z range between -1 and 1. Consequently, the maximum orientation volume, as noted earlier, is given by $V_{o_{\max}} = \pi^2$. If at a position P_i there are N_{o_i} quaternions that are achievable, then the orientation volume is approximately given by

$$V_{o_i} \approx V_{o_{\max}}\frac{N_{o_i}}{N_o} = \pi^2\frac{N_{o_i}}{N_o}. \tag{13}$$

In order to generate N_o uniformly distributed quaternions on the surface of a 3-sphere [46], for each generated q, we select s and v_x as independent random variables uniformly distributed between [0, 1] and [-1, 1], respectively, under the constraint that $S_1 = s^2 + v_x^2 < 1$. We then compute two different independent uniform variables, v_y' and v_z', between [-1, 1], under the constraint that $S_2 = v_y'^2 + v_z'^2 < 1$. Then

$$\begin{aligned} q &= [s,\ v_x,\ v_y,\ v_z] \\ &= \left[s,\ v_x,\ \left(\sqrt{\frac{1-S_1}{S_2}}\right)v_y',\ \left(\sqrt{\frac{1-S_1}{S_2}}\right)v_z'\right]. \end{aligned} \tag{14}$$

We compared the uniform sampling approach to the parameterized sampling approach using a hundred randomly generated positions in the workspace of the robot from (6) and determined that the uniform sampling approached had both slightly higher accuracy and slightly lower computation time, so this approach is used in all orientation volume calculations in this paper. Clearly, the accuracy of any Monte Carlo technique is a function of the number of samples, N_o. Based on experimentation, we use $N_o = 1000$ as a compromise between accuracy and computation time.

D. Workspace Volume Estimation Algorithms

1) Overview: To determine an estimate for the workspace volumes, we develop two algorithms based on Monte Carlo integration. The first, denoted Algorithm A, which is more straightforward, uses direct sampling within the six-dimensional workspace. It is appropriate if one only needs information about six-dimensional volumes. The second, denoted Algorithm B, uses a decomposed sampling technique where a number of samples are associated with each position in order to obtain orientation volume information as a function of position. They both are still equivalent in terms of evaluating the six-dimensional volumes. However, Algorithm B, will be shown to be more efficient in terms of evaluating both the three- and six-dimensional volumes. Furthermore, Algorithm B can represent the six-dimensional

[2]The achievable range is extended by 5° on each end to increase the probability of enclosing the entire reachable orientation volume.

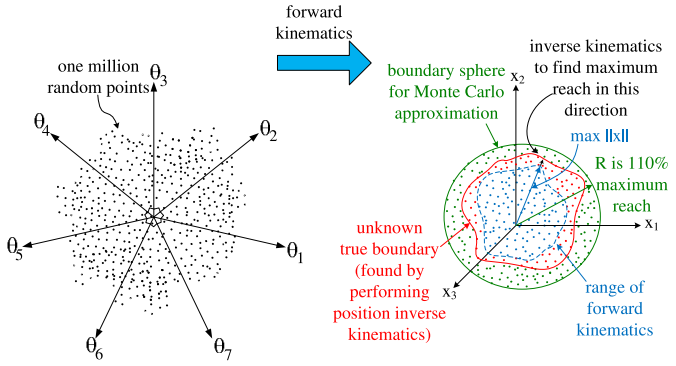

Fig. 2. Illustration of how Monte Carlo integration is used to compute the volume of the reachable workspace. This is done by first determining the maximum reach of the robot by performing forward kinematics on one million random joint configurations. Inverse kinematics is then performed on the configuration whose end effector is furthest from the origin to increase this distance to a workspace boundary, which estimates the maximum reach. The Monte Carlo integration is then performed within a sphere of radius R that is 110% of the maximum reach.

volume by plotting in color the orientation volume for every position in the three-dimensional workspace, which is not possible using Algorithm A. The average difference between their measures was less than 1%. In both cases, we use a total of 10^7 samples, which we have experimentally determined to be sufficient for the Monte Carlo integration to converge [40]. In Algorithm B, 10^4 samples are used within the three-dimensional position workspace with 10^3 samples used in the orientation space associated with each position sample point.

2) Algorithm A (Direct Sampling in Six-Dimensional Space): The first algorithm that we developed uses direct sampling in the six-dimensional workspace to implement Monte Carlo integration. To make our sampling efficient, we first compute a maximum radius for the reachable workspace. To do this, we 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. Then the maximum reach $R_{\max}$ of the manipulator is estimated by picking the point whose position is furthest from the origin and using inverse kinematics on only the linear velocity portion of the Jacobian to try to increase this distance until the robot is at its workspace boundary where its Jacobian is singular. The process for doing this calculation is illustrated in Fig. 2. Once $R_{\max}$ has been determined, we randomly select N samples directly in the six-dimensional workspace, which consists of positions and quaternions that represent orientations. The position part is uniformly sampled within a sphere whose radius R is 110% of $R_{\max}$ so that the maximum volume, $V_{r_{\max}} = \frac{4}{3}\pi R^3$. The quaternion part is sampled using the uniform sampling approach described in Section IV-C2. Denote N_{3d} as the number of samples that are reachable for the given position irrespective of orientation. Then the reachable workspace volume V_r can be approximated by

$$V_r \approx \left(\frac{N_{3d}}{N}\right) V_{r_{\max}} \approx \left(\frac{N_{3d}}{N}\right) \frac{4}{3}\pi R^3. \quad (15)$$

Denote N_{6d} as the number of samples that are reachable for both the given position and orientation. Then the six-dimensional workspace volume V_{6d} can be approximated by

$$V_{6d} \approx \left(\frac{N_{6d}}{N}\right) V_{r_{\max}} V_{o_{\max}} \approx \left(\frac{N_{6d}}{N}\right) \frac{4}{3}\pi R^3 \cdot \pi^2. \quad (16)$$

In practice, we first try to determine if a randomly generated position and orientation is achievable by starting at random configurations and iteratively performing inverse kinematics using the Jacobian. If it is, then it is included in both the count for N_{3d} and N_{6d}. If it is not, then we perform iterative inverse kinematics using only the position portion of the Jacobian to determine if the position is reachable irrespective of orientation. If it is, then it is included in the count for N_{3d}.

To calculate the fault-tolerant six-dimensional volume, we need to determine the number of points where $\mathcal{K} \geq \gamma\ \mathcal{K}_{\max}$ (recall that $\gamma \approx 0.4$ is used), which is denoted N_{FT}. Determining whether a point satisfies this condition requires that one check all of the robot configurations in all self-motion manifolds associated with this point. Techniques for doing this are described in Section IV-D4. Once N_{FT} is computed, the six-dimensional volume is approximated by

$$V_{\mathrm{FT}} \approx \left(\frac{N_{\mathrm{FT}}}{N}\right) \frac{4}{3}\pi R^3 \cdot \pi^2. \quad (17)$$

Calculating the fault-tolerant three-dimensional volume is complicated, because one must determine if there is even one configuration on any possible four-dimensional manifold associated with this position whose Jacobian satisfies $\mathcal{K} \geq \gamma\ \mathcal{K}_{\max}$. The sum of all positions for which such a configuration exists is denoted $N_{\mathrm{FT}_{3d}}$, so that

$$V_{\mathrm{FT}_{3d}} \approx \left(\frac{N_{\mathrm{FT}_{3d}}}{N}\right) \frac{4}{3}\pi R^3 \quad (18)$$

where $V_{\mathrm{FT}_{3d}}$ denotes an estimate for the three-dimensional workspace volume that is fault-tolerant. One way to explore the four-dimensional manifold is to sample the orientation space associated with a position, and evaluate $\mathcal{K}$ while traversing a one-dimensional manifold. However, this is very computationally expensive. The following section describes an approach that is more efficient if one wants to compute all four volumes, i.e., (15)–(18).

3) Algorithm B (Decomposed Sampling of Six-Dimensional Space): An alternate approach to directly sampling the six-dimensional workspace volume is to decompose the workspace into two three-dimensional spaces and perform Monte Carlo integration on both. We sample N_p points in the three-dimensional position workspace and for each reachable position we sample N_o points in the orientation space to determine the associated orientation volume, using one of the approaches discussed in Section IV-C.

Similarly to Algorithm A, we determine $R_{\max}$ first in order to define the sampling sphere whose radius R is 110% of $R_{\max}$. One can then apply (15) to calculate the three-dimensional position volume by replacing N with N_p. For each of these reachable points one still needs to determine the orientation

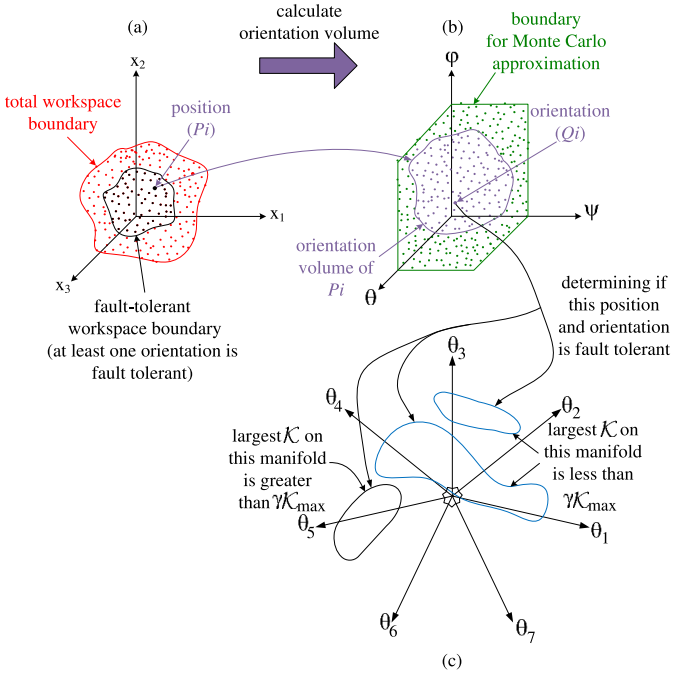

Fig. 3. Illustration of how Monte Carlo integration is used to compute the six-dimensional volumes of both the fault-tolerant workspace and the reachable workspace. For each reachable workspace position P_i in (a) we use Monte Carlo integration to evaluate the achievable orientation volume at that P_i as shown in (b). To evaluate the fault tolerance of a P_i and Q_i in (b) one needs to identify the maximum value of $\mathcal{K}$ for all self-motion manifolds associated with that P_i and Q_i, as shown in (c). The three-dimensional fault-tolerant volume contains all of the positions that have at least one fault-tolerant orientation.

volume using one of the approaches that were discussed in Section IV-C. Then, one can directly use (9) to calculate the total reachable six-dimensional volume.[3]

Because Algorithm B uses decomposed sampling, $V_{\mathrm{FT}_{3d}}$ is easily determined during the process for computing V_{FT}. Fig. 3 illustrates the process for computing these two volumes. We first describe how to compute the fault-tolerant orientation volume associated with a reachable position P_i, denoted $V_{\mathrm{FT}_{oi}}$. For each P_i shown in (a), there is an associated N_{oi} reachable orientations as illustrated in (b). For each reachable position and orientation there are multiple robot configurations which typically occur in multiple self-motion manifolds as illustrated in (c). If any of these configurations have a $\mathcal{K} \geq \gamma\ \mathcal{K}_{\max}$ then this orientation should be included in the sum of all such orientations, denoted $N_{\mathrm{FT}_{oi}}$. One can now compute $V_{\mathrm{FT}_{oi}}$ using

$$V_{\mathrm{FT}_{oi}} \approx \pi^2 \frac{N_{\mathrm{FT}_{oi}}}{N_o} \tag{19}$$

which is analogous to (13). Therefore, one can use (18) to determine $V_{\mathrm{FT}_{3d}}$ by replacing N with N_p, where $N_{\mathrm{FT}_{3d}}$ is now the sum of all of reachable positions whose $V_{\mathrm{FT}_{oi}} > 0$. Similar to (9) V_{FT} can be approximated by

$$V_{\mathrm{FT}} \approx \frac{V_{\mathrm{FT}_{3d}}}{N_{\mathrm{FT}_{3d}}} \sum_{i=1}^{N_{\mathrm{FT}_{3d}}} V_{\mathrm{FT}_{o_i}}. \tag{20}$$

[3]If one is not interested in the orientation volume associated with each position, then one can directly compute (16) using $N = N_p N_o$ and $N_{6d} = \sum_{i=1}^{N_{3d}} N_{oi}$.

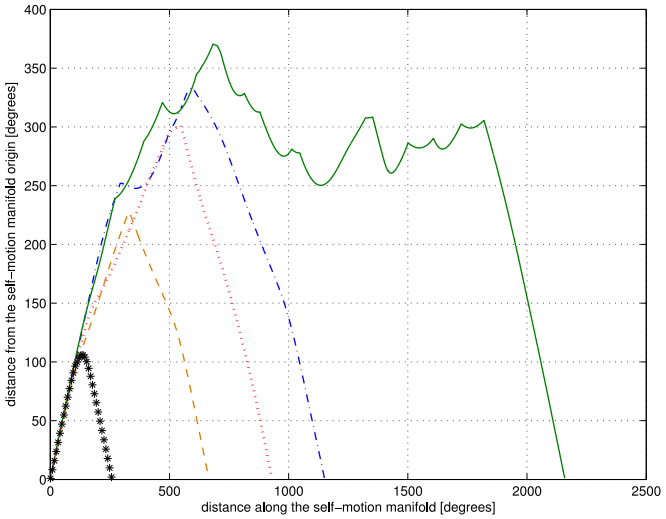

Fig. 4. Example of five self-motion manifolds for the robot from (6) where its end effector is located four meters from the design point in the positive y direction. The independent axis is a measure of the size of the manifold (in degrees) with the dependent axis being the distance from an arbitrary origin on a manifold (giving some sense of its shape).

The most difficult part of determining if a six-dimensional position and orientation satisfies the fault tolerance criterion $\mathcal{K} \geq \gamma\ \mathcal{K}_{\max}$ is to identify and evaluate all self-motion manifolds associated with that location. This is the topic of the next subsection.

4) Maximizing $\mathcal{K}$: In this subsection, we discuss how one can identify a robot configuration that maximizes $\mathcal{K}$ for a given point, i.e., position and orientation, and thus determine if $\mathcal{K} \geq \gamma\ \mathcal{K}_{\max}$.

The first approach is to evaluate every configuration for every self-motion manifold to determine the maximum value of $\mathcal{K}$ at a workspace point [see Fig. 3(c)]. This is not easy, because even determining how many self-motion manifolds exist is not trivial. Our approach to identifying all manifolds is to use multiple random configurations whose locations are close to the point that we are evaluating. Recall that we have already computed the forward kinematics mapping of one million samples in the joint space that were used to estimate the maximum reach $R_{\max}$. It is likely that these samples will include all self-motion manifolds when the sampling rate is high enough. However, it is still possible to miss a manifold, especially if it is small, and increasing the number of samples is computationally expensive. Fig. 4 shows an example for a typical point where five different self-motion manifolds were identified. They are graphed by stepping along the manifold and plotting the absolute value of the difference from an arbitrarily assigned start configuration. The fact that each plot returns to zero indicates that all five of the self-motion manifolds are closed curves, which is not necessarily true.

This approach is relatively straightforward for an isolated point in the workspace; however, if one is concerned with continuous trajectories of the end effector in the workspace, the situation becomes more complicated if one wants to track the maximum. This is because adjacent points in the workspace may have maximum $\mathcal{K}$ values that are associated with configurations that are not adjacent in the joint space. This means

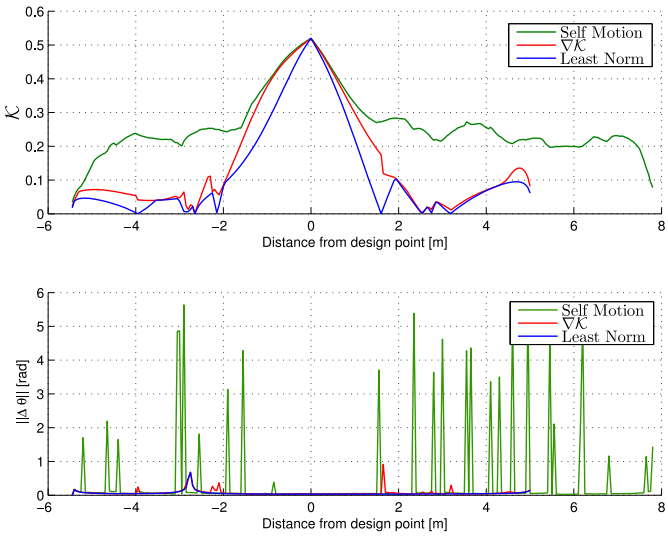

Fig. 5. Plot of the maximum value of $\mathcal{K}$ and the rate of joint displacement for the robot from (6), for a trajectory along the y-axis away from the design point while keeping the orientation constant. The ability to maintain a large value of $\mathcal{K}$ far from the design point comes at the expense of very large joint motion. In fact, the magnitude of configuration change curve $\|\Delta\theta\|$ for the technique that tracks the maximum over all self-motion manifolds is scaled down by a factor of ten.

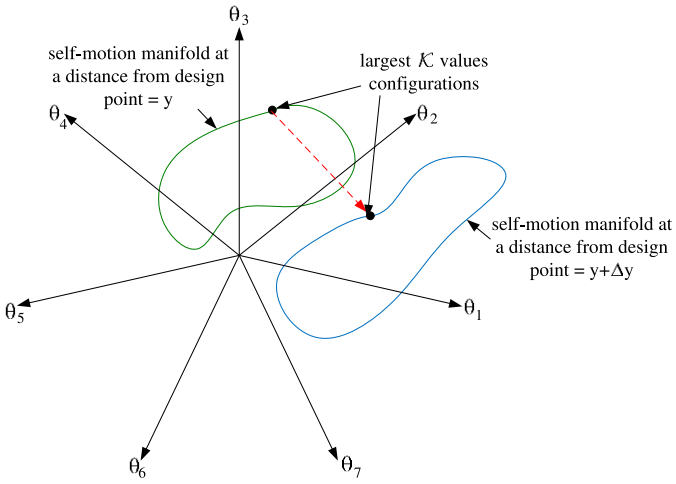

Fig. 6. Illustration of the large value of joint motion that can occur when tracking the globally optimal fault-tolerant configuration. This motion is due to a switch between self-motion manifolds. This reconfiguration will also require motion of the end effector.

TABLE II
COMPARISON OF $\mathcal{K}$ MAXIMIZATION TECHNIQUES USING THE ROBOT FROM (6)

Optimization	$V_{FT_{3d}}$ [m^3]	$V_{FT_{6d}}$ [m^3·rad^3]
Null Motion	2809	8041
$\nabla\mathcal{K}$	2146	487
pseudo inverse	1653	156

that it is not possible to track the maximum $\mathcal{K}$ trajectory without large discontinuities in joint configuration. These jumps in configuration can be either between self-motion manifolds or within a single manifold.

One way of dealing with this issue is to use the gradient projection technique with $\nabla\mathcal{K}$ as described in [1]. It maximizes $\mathcal{K}$ locally, depending on the starting configuration, as opposed to optimizing $\mathcal{K}$ globally. In this work, we always chose the starting configuration to be the optimally fault-tolerant design configuration. This approach is much faster than searching for the global optimal across all self-motion manifolds and it bounds the joint velocity, which makes it applicable for real-time implementation. If one is concerned with locally minimal joint velocity then one can use the pseudoinverse solution, once again, starting from the design configuration.

To illustrate the differences between the above techniques for maximizing $\mathcal{K}$, we selected an example trajectory that moves the end effector along a straight line in the y direction, through the design point, while maintaining constant orientation. Fig. 5 presents the maximum value of $\mathcal{K}$ and norm of the configuration change, $\|\Delta\theta\|$, along this trajectory for the three techniques. Within ± 1 m of the design point the $\nabla\mathcal{K}$ tracking approach is essentially the same as searching all self-motion manifolds because the locally optimal point is globally optimal. Beyond ± 2 m the globally optimal value of $\mathcal{K}$ can be maintained at a relatively large value, i.e., $\mathcal{K} \approx 0.2$ between -4 m and 7 m; however, it requires many transitions between disjoint self-motion manifolds that result in large joint displacements. These large displacements can be alleviated by using the $\nabla\mathcal{K}$ approach (especially if one does not want to exactly track the local optimal); however, the locally optimal value of $\mathcal{K}$ is relatively small outside of ± 2 m. In other words, to obtain larger values of $\mathcal{K}$, one must switch self-motion manifolds. This will require deviation from a purely y velocity trajectory. This is illustrated in Fig. 6 that shows the joint displacement required to move between two adjacent locations, e.g., from $y = 1.50$ m to $y = 1.55$ m, with $\mathcal{K}$ at a global maximum. The total joint displacement for this motion is 3.7 rad and results in an end-effector motion of 2.7 m from the desired linear trajectory.

Fig. 5 illustrated the variation in $\mathcal{K}$ over a specific trajectory. To evaluate how the three techniques behave over the entire workspace we computed both the three-dimensional and six-dimensional fault-tolerant volumes where $\mathcal{K} \geq \gamma\ \mathcal{K}_{\max} = 0.2$. The results are shown in Table II. In the remainder of this paper, we maximize $\mathcal{K}$ by searching all of the self-motion manifolds.

V. EXPLORATION OF DIFFERENT ROBOT DESIGNS

A. Comparing Robot Designs

Now that we have a way to measure the global workspace volumes of interest, we have a way of comparing various different robot designs that are all locally optimal. In Section III, we discussed two different definitions of "optimal" Jacobians, i.e., those given in (6) and (7), from which the kinematic parameters of a manipulator can be determined. Regardless of which definition, i.e., Jacobian, we use, there are $7! = 5040$ different permutations that result in an equal number of unique robot designs, which obviously still possess the same locally optimal fault-tolerant property. To illustrate one aspect of the wide variation among all of the robot designs, Fig. 7 is a plot of the maximum reach as a function of the permutation

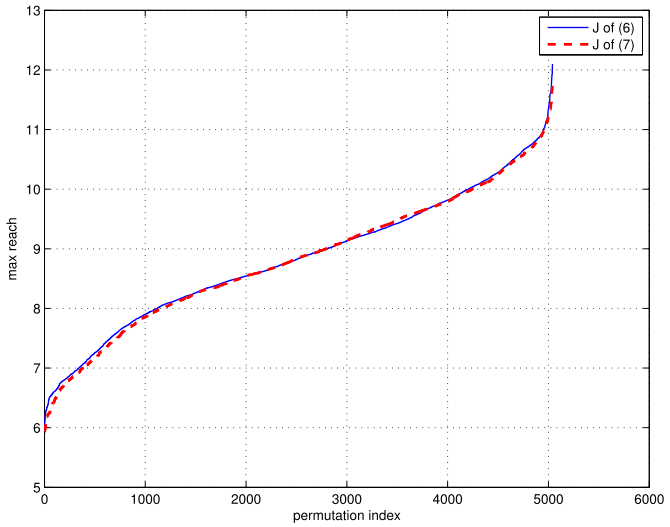

Fig. 7. Plot of the maximum reach for all possible robots generated from the 7! column permutations of (6) and (7).

TABLE III
DH PARAMETERS OF MIN ROBOT FROM (6)

i	α_i [degrees]	a_i [m]	d_i [m]	θ_i [degrees]
1	-98	0.17	0	0
2	-86	1.57	-0.03	65
3	92	0.50	-1.07	-148
4	-83	0.61	-0.39	-49
5	-66	1.42	0.72	109
6	-75	0	0.91	-10
7	0	1	-0.001	180

index, which is ordered from minimum to maximum value of $R_{\max}$, for both Jacobians given in (6) and (7). It is not surprising that the variation of $R_{\max}$ is similar for both Jacobians, because both optimizations require that every column is of equal norm.

There appears to be a rough relationship between the maximum robot reach, $R_{\max}$, and the normalized workspace volumes. We will illustrate this with robot designs that have minimum, maximum, and mid-range values of $R_{\max}$. It turns out that the robots designed directly from (6) and (7) are in the mid-range, with $R_{\max} = 9.4$ and $R_{\max} = 8.7$, respectively. The minimum and maximum $R_{\max}$ robots for (6) result from permutations $[j_1\ j_2\ j_5\ j_7\ j_4\ j_3\ j_6]$ and $[j_2\ j_7\ j_3\ j_5\ j_4\ j_1\ j_6]$, respectively. The DH parameters for robots generated from these Jacobians are given in Tables III and IV, respectively. The minimum and maximum $R_{\max}$ robots for (7) result from permutations $[j_5\ j_1\ j_4\ j_6\ j_2\ j_3\ j_7]$ and $[j_3\ j_5\ j_2\ j_4\ j_7\ j_1\ j_6]$, respectively.

Table V presents the three-dimensional and six-dimensional volumes using percentages out of the maximum to give some intuition about the relative size of these workspaces for these six different robots. Arguably, the best robot design is given by the minimum $R_{\max}$ of (6), where the maximum reachable three-dimensional volume is 99% of a sphere of radius $R_{\max}$, indicating that this robot's reachable workspace is almost spherical. The six-dimensional workspace volume is 49% of the maximum six-dimensional volume, i.e., $\frac{4}{3}\pi R_{\max}^3 \cdot \pi^2$.

TABLE IV
DH PARAMETERS OF MAX ROBOT FROM (6)

i	α_i [degrees]	a_i [m]	d_i [m]	θ_i [degrees]
1	147	1.00	0	0
2	-69	0.87	-3.98	103
3	-29	0.02	4.46	-42
4	50	0.56	-5.36	117
5	-71	1.38	2.76	153
6	136	1.26	-2.62	136
7	0	1	-1.21	0

TABLE V
COMPARISON OF ROBOT WORKSPACE VOLUMES

		Robots from J of (6)			Robots from J of (7)		
		Max reach R_{max} [m]			Max reach R_{max} [m]		
		min	mid	max	min	mid	max
		6.0	9.4	12.1	5.9	8.7	11.7
Volumes[%]	V_r	99	96	62	97	95	68
	$V_{FT_{3d}}$	81	81	51	81	73	59
	V_{6d}	49	46	26	49	41	28
	V_{FT}	27	24	12	26	19	13

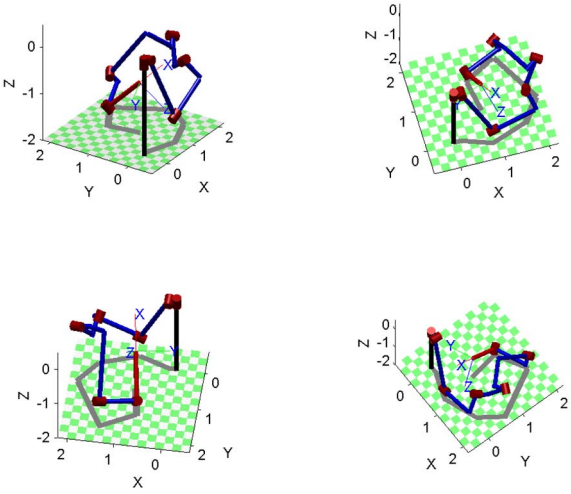

Fig. 8. Locally optimal fault-tolerant configuration at the design point of min robot from (6). Note the more evenly distributed links of this robot configuration as compared to that of Fig. 1.

In other words, within the spherical workspace, this robot is capable of achieving approximately half of all possible orientations. For relatively high degrees of fault tolerance, i.e., $\mathcal{K} > 0.2$, the robot would be limited to only 27% of the maximum workspace. However, if one is only concerned with a spatial positioning task, the three-dimensional volume with $\mathcal{K} > 0.2$ is quite large, i.e., 81%.

This "best" robot design is shown in its optimal configuration in Fig. 8. From Fig. 8, one can see that the robot is not folded up on itself like the robot from (6) in Fig. 1. This is typical of min robot designs because the joints of a robot generated from an optimally fault-tolerant Jacobian are constrained to lie on a sphere that is centered at the end effector. Consequently, the robot designs will become increasingly folded up on themselves as their $R_{\max}$ increases. Note that the min robot for the Jacobian from (7) is also the best, in the sense of having the largest volumes.

To visualize the six-dimensional volume, we use a three-dimensional volume plot, where for each point we use color to represent the orientation volume at that point. One can

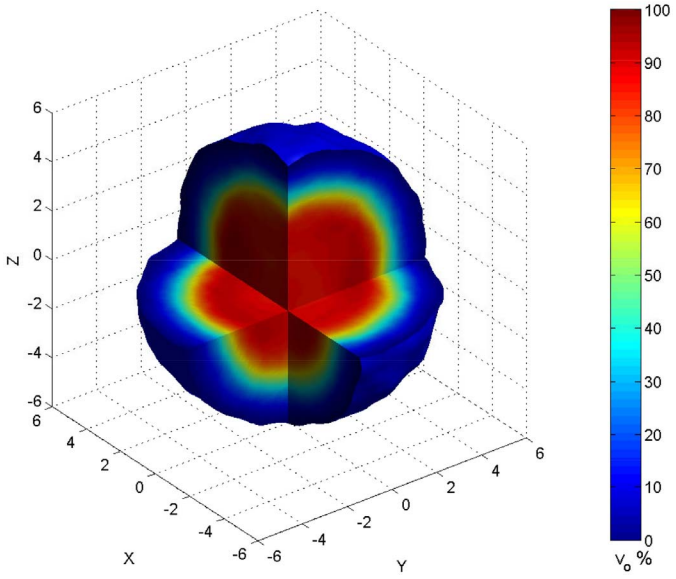

Fig. 9. Six-dimensional fault-tolerant volume of $\mathcal{K} > 0.2$ for the min robot from (6). The orientation volume is represented by its color within the three-dimensional volume as a percentage of $V_{o_{\max}} = \pi^2$.

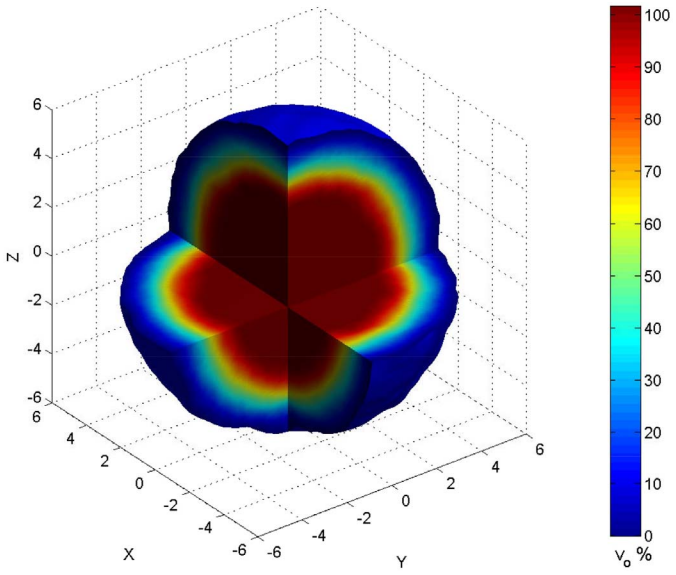

Fig. 10. Six-dimensional reachable volume for the min robot from (6). This is very similar to the fault-tolerant volume for $\mathcal{K} \neq 0$. The orientation volume is represented by its color within the three-dimensional volume as a percentage of $V_{o_{\max}} = \pi^2$.

visualize the three-dimensional workspace by simply ignoring the color map. Fig. 9 illustrates the fault-tolerant six-dimensional volume for the min robot from (6), where the three-dimensional volume part is shown with multiple cross-sections at the design point to better visualize the interior. The color map represents the orientation volume distribution within the three-dimensional volume, where the orientation volume is represented by its percentage of the maximum value $V_{o_{\max}} = \pi^2$. One can see how the largest values of fault-tolerant orientation volume are concentrated around the design point. Fig. 10 illustrates the total reachable volume, which is similar to the volume where $\mathcal{K} > 0$, because there are large self-motion manifolds that make it easy to find a configuration where $\mathcal{K} \neq 0$.

VI. Conclusion

This paper has explored the kinematic design of optimally fault-tolerant manipulators. It has focused on seven DOF redundant manipulators used for fully general spatial motion. Two different approaches for identifying physically feasible designs were explored. Both approaches identify locally optimal robot configurations based on properties of the Jacobian. A technique for efficiently computing six-dimensional workspace volumes was then developed to determine how these fault-tolerant properties vary throughout the workspace. This provides a quantitative measure for comparing different locally optimal designs. This is important because each locally optimal Jacobian will result in 7! unique manipulators designs. It was shown that robot designs with small normalized maximum reach tend to have larger fault-tolerant workspace volumes. Such kinematic designs also tend to be more well configured at the optimal design point.

References

[1] 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.

[2] Reliability Information Analysis Center, "Nonelectronic parts reliability data," Air Force Res. Lab., Defense Tech. Inf. Center, Rome, NY, USA, Tech. Rep. NPRD-2011, 2011.

[3] S. Cheng and B. S. Dhillon, "Reliability and availability analysis of a robot-safety system," *J. Qual. Maint. Eng.*, vol. 17, no. 2, pp. 203–232, 2011.

[4] 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. Autom.*, vol. 9, no. 5, pp. 675–684, Oct. 1993.

[5] 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.

[6] S. Soylu, B. J. Buckham, and R. P. Podhorodeski, "Redundancy resolution for underwater mobile manipulators," *Ocean Eng.*, vol. 37, nos. 2–3, pp. 325–343, 2010.

[7] 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. 1996, pp. 235–240.

[8] S. Kawatsuma, M. Fukushima, and T. Okada, "Emergency response by robots to Fukushima-Daiichi accident: Summary and lessons learned," *Ind. Robot Int. J.*, vol. 39, no. 5, pp. 428–435, 2012.

[9] K. Nagatani *et al.*, "Emergency response to the nuclear accident at the Fukushima Daiichi nuclear power plants using mobile rescue robots," *J. Field Robot.*, vol. 30, no. 1, pp. 44–63, 2013.

[10] M. L. Visinsky, J. R. Cavallaro, and I. D. Walker, "A dynamic fault tolerance framework for remote robots," *IEEE Trans. Robot. Autom.*, vol. 11, no. 4, pp. 477–490, Aug. 1995.

[11] P. Nieminen *et al.*, "Water hydraulic manipulator for fail safe and fault tolerant remote handling operations at ITER," *Fusion Eng. Design*, vol. 84, nos. 7–11, pp. 1420–1424, 2009.

[12] J. D. English and A. A. Maciejewski, "Fault tolerance for kinematically redundant manipulators: Anticipating free-swinging joint failures," *IEEE Trans. Robot. Autom.*, 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] 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.

[15] S. Tosunoglu and V. Monteverde, "Kinematic and structural design assessment of fault-tolerant manipulators," *Intell. Autom. Soft Comput.*, vol. 4, no. 3, pp. 261–268, 1998.

[16] C.-M. Lin and C.-H. Chen, "Robust fault-tolerant control for a biped robot using a recurrent cerebellar model articulation controller," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 1, pp. 110–123, Feb. 2007.

[17] J.-M. Yang, "Fault-tolerant gaits of quadruped robots for locked joint failures," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 32, no. 4, pp. 507–516, Nov. 2002.

[18] S. K.-K. Chu and G. K.-H. Pang, "Comparison between different model of hexapod robot in fault-tolerant gait," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 32, no. 6, pp. 752–756, Nov. 2002.

[19] C. Carreras and I. D. Walker, "Interval methods for fault-tree analysis in robotics," *IEEE Trans. Rel.*, vol. 50, no. 1, pp. 3–11, Mar. 2001.

[20] L. Notash, "Joint sensor fault detection for fault tolerant parallel manipulators," *J. Robot. Syst.*, vol. 17, no. 3, pp. 149–157, 2000.

[21] S. N. Huang, K. K. Tan, and T. H. Lee, "Automated fault detection and diagnosis in mechanical systems," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 6, pp. 1360–1364, Nov. 2007.

[22] L. V. Allen and D. M. Tilbury, "Anomaly detection using model generation for event-based systems without a preexisting formal model," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 3, pp. 654–668, May 2012.

[23] M. L. Leuschen, I. D. Walker, and J. R. Cavallaro, "Fault residual generation via nonlinear analytical redundancy," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 3, pp. 452–458, May 2005.

[24] L. M. Capisani, A. Ferrara, A. F. de Loza, and L. M. Fridman, "Manipulator fault diagnosis via higher order sliding-mode observers," *IEEE Trans. Ind. Electron.*, vol. 59, no. 10, pp. 3979–3986, Oct. 2012.

[25] X. W. Chen and S. Y. Nof, "Error detection and prediction algorithms: Application in robotics," *J. Intell. Robot. Syst.*, vol. 48, no. 2, pp. 225–252, 2007.

[26] 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.

[27] 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. Autom.*, Kobe, Japan, May 2009, pp. 3359–3364.

[28] L. Notash, "A methodology for actuator failure recovery in parallel manipulators," *Mech. Mach. Theory*, vol. 46, no. 4, pp. 454–465, 2011.

[29] A. A. Allais, J. E. McInroy, and J. F. O'Brien, "Locally decoupled micromanipulation using an even number of parallel force actuators," *IEEE Trans. Robot.*, vol. 28, no. 6, pp. 1323–1334, Dec. 2012.

[30] J. E. McInroy and F. Jafari, "Finding symmetric orthogonal Gough-Stewart platforms," *IEEE Trans. Robot.*, vol. 22, no. 5, pp. 880–889, Oct. 2006.

[31] A. A. Maciejewski, "Fault tolerant properties of kinematically redundant manipulators," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1. Cincinnati, OH, USA, May 1990, pp. 638–642.

[32] 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.

[33] R. G. Roberts, H. G. Yu, and A. A. Maciejewski, "Fundamental limitations on designing optimally fault-tolerant redundant manipulators," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1224–1237, Oct. 2008.

[34] 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.

[35] 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.

[36] 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.

[37] 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. Autom. Sci. Eng.*, vol. 12, no. 4, pp. 1421–1432, Oct. 2015.

[38] 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 Trans. Robot.*, vol. 29, no. 5, pp. 1300–1307, Oct. 2013.

[39] K. M. Ben-Gharbia, A. A. Maciejewski, and R. G. Roberts, "A kinematic analysis and evaluation of planar robots designed from optimally fault-tolerant Jacobians," *IEEE Trans. Robot.*, vol. 30, no. 2, pp. 516–524, Apr. 2014.

[40] K. M. Ben-Gharbia, A. A. Maciejewski, and R. G. Roberts, "An example of a seven joint manipulator optimized for kinematic fault tolerance," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, San Diego, CA, USA, Oct. 2014, pp. 802–807.

[41] 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.

[42] 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.

[43] 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. 2010, pp. 453–460.

[44] P. I. Corke, "A robotics toolbox for MATLAB," *IEEE Robot. Autom. Mag.*, vol. 3, no. 1, pp. 24–32, Mar. 1996.

[45] C. Chen and D. Jackson, "Parameterization and evaluation of robotic orientation workspace: A geometric treatment," *IEEE Trans. Robot.*, vol. 27, no. 4, pp. 656–663, Aug. 2011.

[46] G. Marsaglia, "Choosing a point from the surface of a sphere," *Ann. Math. Stat.*, vol. 43, no. 2, pp. 645–646, 1972.

Khaled M. Ben-Gharbia (S'11–M'15) received the B.S. and M.S. degrees in Electrical and Electronic engineering from the University of Tripoli (formerly Al-Fateh University), Tripoli, Libya, in 2003 and 2008, respectively, and the Ph.D. degree in electrical engineering from Colorado State University, Fort Collins, CO, USA, in 2014.

He is currently with Wolf Robotics LLC, Fort Collins.

Anthony A. Maciejewski (F'05) received the B.S.E.E., M.S., and Ph.D. degrees in Electrical Engineering from The Ohio State University, Columbus, OH, USA, in 1982, 1984, and 1987, respectively.

From 1988 to 2001, he was a Professor of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA. He is currently the Department Head of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO, USA.

Rodney G. Roberts (SM'02) received the B.S. degree in electrical engineering and the B.S. degree in mathematics from the Rose-Hulman Institute of Technology, Terre Haute, IN, USA, in 1987, and the M.S.E.E. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1988 and 1992, respectively.

From 1992 to 1994, he was a National Research Council Fellow with Wright Patterson Air Force Base, Dayton, OH, USA. Since 1994, he has been with the College of Engineering, Florida Agricultural and Mechanical University–The Florida State University, Tallahassee, FL, USA, where he is currently a Professor of Electrical and Computer Engineering. His current research interests include robotics and image processing.