Designing a Failure-Tolerant Workspace for Kinematically Redundant Robots

Randy C. Hoover, *Member, IEEE*, Rodney G. Roberts, *Senior Member, IEEE*, Anthony A. Maciejewski, *Fellow, IEEE*, Priya S. Naik, *Member, IEEE*, and Khaled M. Ben-Gharbia, *Student Member, IEEE*

***Abstract*—Kinematically redundant manipulators are inherently more robust to locked joint failures than non-redundant manipulators. However, if poorly designed, performance degradation may still occur in the presence of a single locked joint. This paper presents a technique for designing a desired operating workspace for a kinematically redundant manipulator that can be guaranteed after the occurrence of an arbitrary single locked joint failure. The existence of such a workspace, called a *failure-tolerant workspace*, will be guaranteed by imposing a suitable set of artificial joint limits prior to a failure. Conditions are presented that characterize end-effector locations within the *failure-tolerant* region. Based on these conditions, an algorithm for computing the failure-tolerant workspace is presented. The algorithm is based upon identifying the boundaries of the failure-tolerant workspace. Examples are presented to illustrate the application of the proposed algorithm to various manipulator design problems.**

***Note to Practitioners*—This work is motivated by situations where an automation engineer would like to employ a robot to autonomously perform tasks in remote or hazardous environments where failures are inevitable, e.g., in space exploration and nuclear waste remediation. In order to be able to complete the desired tasks for as long as possible, one would like to make the robot tolerant to failures in its joints. One technique for doing this is to use a robot with more than the minimum number of joints required to complete the assigned tasks. However, it has been previously shown that if one does not properly control the motion of the robot prior to a failure, the remaining joints may not be able to guarantee completion of the desired tasks. This work presents a technique for producing such guarantees. In particular, it shows how one can apply artificial limits on the range of the robot's joints prior to a failure to ensure that all tasks can still be performed after a failure occurs. Furthermore, it shows how to identify an optimal set of artificial limits that balances the tradeoff between pre-failure performance and post-failure capabilities. This technique is illustrated with two examples using a robot obtained from considering three joints of the commercial PA-10 robot.**

***Index Terms*—Failure tolerant, kinematically redundant manipulators, workspace design.**

Manuscript received October 09, 2013; revised February 11, 2014; accepted June 19, 2014. Date of publication August 01, 2014; date of current version October 02, 2015. This paper was recommended for publication by Associate Editor X. Chen and Editor J. Wen upon evaluation of the reviewers' comments. This work was supported in part by the National Science Foundation under Contract IIS-0812437. This paper was presented in part at the Ninth IEEE International Conference on Automation Science and Engineering (CASE 2013), Madison, WI, USA, in part at the 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, and the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA.

R. C. Hoover is with the Department of Electrical and Computer Engineering, South Dakota School of Mines and Technology, Rapid City, SD 57701 USA (e-mail: randy.hoover@sdsmt.edu).

R. G. Roberts is with the Department of Electrical and Computer Engineering, College of Engineering, Florida Agricultural and Mechanical University—The Florida State University, Tallahassee, FL 32310-6046 USA (e-mail: rroberts@eng.fsu.edu).

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

Digital Object Identifier 10.1109/TASE.2014.2337935

I. Introduction

ROBOTIC manipulators are used extensively for various applications that require autonomous or semi-autonomous operation. This is especially true in remote and/or hazardous environments such as in underwater [1] and space exploration [2] or in nuclear waste remediation [3]. Because such environments are dangerous for humans to work in, this also makes repair or maintenance difficult and expensive. Therefore, it is extremely important for robots that work in such environments to be able to withstand component failures and continue working with minimal degradation in performance. One recent example is the Fukushima nuclear reactor accident, where robot component failures were not only likely, but inevitable [4], [5]. This paper presents a technique for designing a desired operating workspace for a kinematically redundant manipulator, that can be guaranteed after the occurrence of an arbitrary single locked joint failure.

We assume that the robotic manipulators applied in hazardous environment have already been designed using techniques for assessing and increasing their reliability [6], [7], e.g., the use of fault tree analysis [8]. Furthermore, methods for fault detection [9]–[11] and identification [12] are employed, so that either the failure itself or the recovery mechanism [13], [14] will lock the affected joint, with minimal internal shock [15]. These assumptions allow us to even model failures that naturally result in a loss of torque, i.e., free-swinging failures [16]. We further assume that the manipulator is a serial mechanism, although there is extensive work on failure tolerance for parallel manipulators [17]–[21].

Our goal in this work is to guarantee post-failure operation in a desired region of the workspace. This is distinct, but related, to fault-tolerance techniques that are based on local properties of a manipulator's current configuration associated with the Jacobian [22]–[24]. It is possible to traverse a desired trajectory while locally optimizing such properties [25], [26], however, there is no guarantee regarding global behavior. There do exist path planning approaches that guarantee successful task completion in the presence of locked-joint failures [27]–[30], however, they are limited to point to point trajectories.

In [31], it is proven that for a manipulator operating in an m-dimensional workspace, at least $m+2$ joints are required to ensure fault tolerance if the joint motion is unconstrained. In [32], it was shown that one can precompute a set of artificial joint limits that would guarantee the reachability of desired end effector locations, even with a single degree of redundancy. The technique in [33] and [34] builds upon this concept of artificial joint limits to guarantee that entire regions of the workspace are reachable after a locked-joint failure. Our approach here extends this technique to calculate the boundaries of failure-tolerant workspaces, i.e., regions of the workspace that can be guaranteed to be reachable after an arbitrary single joint failure. A judicious choice of artificial joint limits imposed prior to a failure, which are then released after the failure, allows one to design a desired post-failure workspace.

The remainder of this paper is organized as follows. In Section II, the fault tolerant workspace problem is mathematically formulated. In Section III, the problem of identifying the pre-failure workspace is described. The boundaries of this workspace are identified by kinematic singularities and joint-limit singularities. New techniques for identifying workspace boundaries due to multiple joint limits are described. In Section IV, conditions for the fault tolerant workspace are given. These conditions are used in Section V to identify potential boundaries for the fault tolerant workspace. Section VI explains how the true workspace boundaries are identified. Section VII gives a short algorithm to summarize the entire procedure which is illustrated with examples in Section VIII, and conclusions appear in Section IX.

II. Problem Formulation

Let the kinematic function mapping the joint space $\mathcal{C} \subset \mathbb{R}^n$ to the workspace $\mathcal{W} \subset \mathbb{R}^m$ be denoted by $\mathbf{f} : \mathcal{C} \to \mathcal{W}$. In this work, we will assume that the configuration space $\mathcal{C}$ has the form $\mathcal{C}_{\boldsymbol{B}} = B_1 \times \cdots \times B_n$. If joint i has no physical joint limits, $B_i = \mathbb{R}$, and if joint i does have physical joint limits, $B_i = [\underline{b}_i, \bar{b}_i]$ where $\underline{b}_i < \bar{b}_i$. Initially, we introduce artificial limits for each joint so that the i-th joint $q_i \in A_i = [\underline{a}_i, \bar{a}_i]$. If it can be safely assumed that joint i will not fail then we can set $A_i = B_i$. The set $\mathcal{C}_{\boldsymbol{A}} = A_1 \times \cdots \times A_n$ denotes the region of the configuration space corresponding to the artificial joint limits. The joint space prior to a failure is then simply $\mathcal{C}_{\boldsymbol{A}}$. Once a locked-joint failure occurs, the artificial joint limits are released and the robot is constrained to operate on a failure-induced hyperplane. This of course has a significant impact on the resulting reachable workspace. There are generally end-effector locations that were reachable prior to the failure that are no longer reachable after a failure. There may also be areas of the workspace that were formerly unreachable but that, in spite of the locked joint, become reachable after releasing the artificial joint limits of the non-failed joints. The *fault tolerant workspace* is defined as the part of the workspace that is reachable prior to and after any single locked-joint failure where the joint failure can occur at any configuration in $\mathcal{C}_{\boldsymbol{A}}$.

To illustrate the significance of the fault tolerant workspace, consider a planar 3R manipulator with equal length links.[1]

[1]One real application for a planar manipulator is given in [5], where the task space that was considered to sample contaminated water and install a water gauge inside a damaged reactor was a plane.

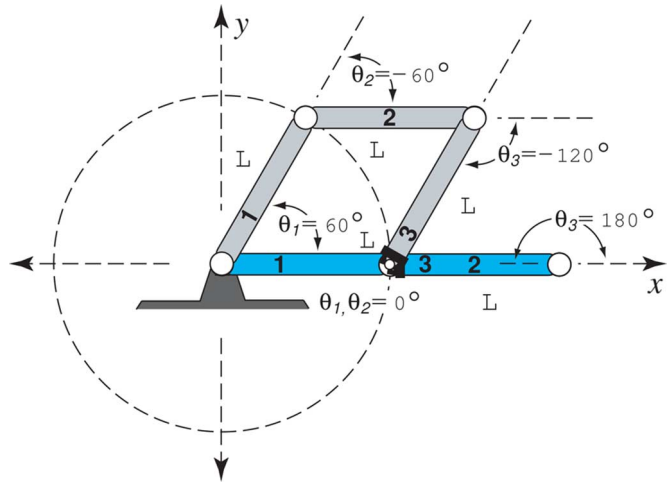


Fig. 1. Two configurations of a planar 3R robot with equal link lengths of L meters. The configurations shown are from an infinite family of configurations resulting in the end-effector position $[L, 0]^T$. The first configuration, $\boldsymbol{\theta} = [60°, -60°, -120°]^T$, is fault tolerant but the second configuration, $\boldsymbol{\theta} = [0°, 0°, 180°]^T$, is fault intolerant to a locked-joint failure in the third joint as this will restrict the end effector to remain on the circle shown regardless of the values of the remaining two healthy joints.

Without artificial or physical joint limits, the fault tolerant workspace is quite small. This can be clearly seen in Fig. 1, where two configurations corresponding to the same end-effector location on the unit circle are shown. While the first configuration is relatively fault tolerant to a locked-joint failure, the second configuration is fault intolerant to joint 3 being locked, in which case the end effector is constrained to remain on the dashed unit circle shown in the figure. In fact, it was shown in [32] that this unit circle is the only region of the workspace that is guaranteed to be reachable following any possible locked-joint failure. However, it will be shown in Section VI that the fault tolerant workspace can be significantly increased by simply enforcing artificial joint limits that can be released after a failure.

In some cases, only certain joints are prone to failures. Let the *failure index set* $\mathbf{F} \subset \{1, 2, \ldots, n\}$ denote the joint labels of the failure-prone joints. We will assume that those joints that are not contained in $\mathbf{F}$ will remain healthy throughout the robot's mission. Our goal then is to determine the fault tolerant workspace corresponding to at most one locked-joint failure where any joint $i \in \mathbf{F}$ can fail. Mathematically, this problem can be formulated in the following way. Prior to a joint failure, the robot's operating configuration space is $\mathcal{C}_{\boldsymbol{A}}$ and the pre-failure workspace is given by

$$\mathcal{W}_{\mathbf{0}} = \mathbf{f}(\mathcal{C}_{\boldsymbol{A}}) = \{\mathbf{x} = \mathbf{f}(\mathbf{q}) \mid \mathbf{q} \in \mathcal{C}_{\boldsymbol{A}}\}. \tag{1}$$

If the i-th joint is locked at $q_i = \theta_i$ and the remaining artificial joint limits are released, the resulting reduced configuration space is given by

$${}^i\mathcal{C}(\theta_i) = \{\mathbf{q} \in \mathcal{C}_{\boldsymbol{B}} \mid q_i = \theta_i\}. \tag{2}$$

Geometrically, one can consider ${}^i\mathcal{C}(\theta_i)$ to be the intersection of the hyperplane given by $q_i = \theta_i$ with the feasible configuration space $\mathcal{C}_{\boldsymbol{B}}$. Because the artificial joint limits were enforced prior to the failure, we have that $\underline{a}_i \leq \theta_i \leq \bar{a}_i$. It is assumed that the failure can occur anywhere in this interval and the joint is

locked at that configuration. Hence, the guaranteed workspace following a locked-joint failure of joint i subject to the artificial joint limits is

$$\boldsymbol{\mathcal{W}_i} = \bigcap_{\underline{a}_i \le \theta_i \le \bar{a}_i} \mathbf{f}\left({}^i\boldsymbol{\mathcal{C}}(\theta_i)\right). \tag{3}$$

The fault tolerant workspace is then the intersection of the pre-failure workspace $\boldsymbol{\mathcal{W}_0}$ and the various post-failure fault tolerant workspaces $\boldsymbol{\mathcal{W}_i}, i \in \mathbf{F}$

$$\boldsymbol{\mathcal{W}_F} = \bigcap_{i \in F \cup \{0\}} \boldsymbol{\mathcal{W}_i}. \tag{4}$$

Our goal is to determine $\boldsymbol{\mathcal{W}_F}$.

Unfortunately, finding $\boldsymbol{\mathcal{W}_F}$ directly is generally impossible, so the approach taken here will be to identify necessary conditions for its boundaries. Recall that a boundary point of a subset $\boldsymbol{\mathcal{S}}$ of $\mathbb{R}^m$ is a point $\mathbf{x}$ such that every open neighborhood of $\mathbf{x}$ contains at least one point in $\boldsymbol{\mathcal{S}}$ and at least one point not in $\boldsymbol{\mathcal{S}}$. Although the boundary of a general point set can be quite complicated, the boundary sets considered in this work are simple and correspond to simple curves or hyper-surfaces, depending on the dimension of the workspace. These boundaries are determined by first identifying candidate boundary sets. These candidate boundary sets correspond to a limitation of the manipulator's ability to move its end effector arbitrarily. In particular, the candidate workspace boundaries prior to a failure are related to the concepts of kinematic singularities and joint-limit singularities. In the next section, we will describe how one can identify these candidate boundaries. Conditions are then introduced in Section IV and are subsequently applied in Section V to identify the fault tolerant workspace boundaries.

III. Identifying the Pre-Failure Workspace Boundaries

Before developing criteria for identifying the fault tolerant workspace boundary, we discuss the problem of identifying the workspace boundary of a healthy robot without any failures. These boundaries are located by identifying two types of singularities: kinematic singularities and joint-limit singularities.

The local motion of the end effector is characterized by the Jacobian equation

$$\mathbf{v} = J\dot{\mathbf{q}} \tag{5}$$

where $\mathbf{v}$ denotes the end-effector velocity, $\dot{\mathbf{q}}$ denotes the joint velocity, and J denotes the manipulator Jacobian. If the manipulator Jacobian has full rank and the joints are unconstrained, then the end effector can move locally in any direction, indicating that the end effector is located in the interior of the workspace. Configurations where J does not have full rank, i.e., when the manipulator does not have full end-effector control, are called *kinematic singularities*. These configurations identify candidate workspace boundary points.

When the manipulator has joint limits, the columns of the manipulator Jacobian corresponding to the constrained joints can only contribute in one direction (a positive amount if the joint is in a lower limit and a negative amount if the joint is in an upper limit). In some cases, this may cause the loss of full end-effector motion even if the manipulator is not in a kinematic singularity. The joint configuration in this case is said to be a *joint-limit singularity* (the term *semi-singularity* is also used [35]). At a workspace boundary, the manipulator is either in a kinematic singularity or a joint-limit singularity. This observation can be used to find the workspace boundaries of the manipulator. It is important to note that the end-effector location corresponding to a kinematic singularity or a joint-limit singularity is not necessarily a boundary point.

Kinematic singularities are relatively easy to find; they are simply those configurations where the manipulator Jacobian does not have full rank. For a kinematically redundant manipulator, these configurations are characterized by $\det(JJ^T) = 0$. This is further simplified for the case of a single degree of redundancy by the fact that $\det(JJ^T) = \|\mathbf{n}_J\|_2^2 = n_{J1}^2 + \cdots + n_{Jn}^2$, where $\mathbf{n}_J$ is the canonical null vector of J, i.e., the null vector obtained by taking the equivalent of the cross product of the rows of J. Hence, the kinematic singularities are those configurations for which each element n_{Ji} of $\mathbf{n}_J$ is zero.

Identifying joint-limit singularities is a more challenging problem, particularly when more than one joint is at its limit. As with kinematic singularities, joint-limit singularities are characterized by a local loss of full end-effector motion. In this case, full motion control is lost because one or more joints are at their limits. To more clearly see the reason for the problem, we note that pseudoinverse control $\dot{\mathbf{q}} = J^+\mathbf{v}$ is typically insufficient for full end-effector control when one or more joints are at their limit. Indeed, if the end-effector velocity $\mathbf{v}$ is feasible under pseudoinverse control where at least one constrained joint has a nonzero joint velocity moving that joint away from its limit, then the end-effector velocity $-\mathbf{v}$ would not be feasible as it would require the constrained joint to move past its limit. In such cases, one must rely on the null space to add enough joint velocity to meet the required joint velocity constraints while still achieving the desired end-effector motion.

It can be shown that if a manipulator has a single degree of redundancy and is not in a kinematic singularity, then the family of joint velocities that result in the end-effector velocity $\mathbf{v}$ is given by

$$\dot{\mathbf{q}} = J^+\mathbf{v} + \alpha\mathbf{n}_J \tag{6}$$

where α is an arbitrary scalar. In order for (6) to be feasible given the joint-limit conditions, the components corresponding to the actively constrained joints must have the appropriate sign, i.e., if joint i is at its upper (lower) limit, the joint velocity $\dot{q}_i$ must be non-positive (non-negative). If the i-th component of $\mathbf{n}_J$ is nonzero, then by choosing an appropriate value for α will allow (6) to be feasible. On the other hand, if the i-th component of $\mathbf{n}_J$ is zero, then no amount of the null vector can be added in (6) to adjust the sign of the $\dot{q}_i$ and consequently, certain end-effector velocities will not be possible. If two or more joint are at their limits, then the relative directions of the columns of the manipulator Jacobian must be examined. For example, if joints i and $j (i \neq j)$ are at their upper limits, then the null space term can compensate for the joint limits provided that n_{Ji} and n_{Jj} are nonzero and of the same sign. Otherwise, certain end-effector velocities cannot be achieved. However, if joint i is at its upper limit and joint j is at its lower limit, arbitrary end-effector velocities can be achieved if and only if n_{Ji} and n_{Jj}

are nonzero and of the opposite sign. The generalization to more joints being at their limits is obvious. Similar results hold for robots with higher degrees of redundancy, e.g., if two joints are at their upper limits, there must be a vector of the null space of J for which the corresponding components are nonzero and of the same sign [36].

There are a significant number of cases to check when identifying potential joint-limit singularities when multiple joints are at their limits. These cases can be enumerated based on three possibilities for each joint: a given joint can be at its lower limit, its upper limit, or at neither limit. This gives a total of 3^n possibilities including the case when no joint is at its limit. Fortunately, for robots with a single degree of redundancy, cases when three or more joints are at their limits are implicitly identified when checking the cases for two joints at their limits. Thus one only needs to check for kinematic singularities, cases when a single joint is at its lower or upper limit, and cases when two joints are at one of the four possible combinations of lower and/or upper limits, giving a total of $1+2n+2^2\binom{n}{2}=1+2n^2$ cases to consider.

IV. Conditions for Failure Tolerant Workspace Locations

Whether or not a given end-effector location is in the fault tolerant workspace is completely determined by its pre-image, i.e., the family of configurations corresponding to that workspace location. The pre-image of a workspace location $\mathbf{x} \in \boldsymbol{\mathcal{W}}$ is the set

$$\mathbf{f}^{-1}(\mathbf{x}) = \{\mathbf{q} \in \boldsymbol{\mathcal{C}_B} \mid \mathbf{f}(\mathbf{q}) = \mathbf{x}\}. \tag{7}$$

The reachability of the end-effector location $\mathbf{x}$ after a locked-joint failure has occurred is determined by whether or not the joint value of the locked joint falls within that individual joint range of the set $\mathbf{f}^{-1}(\mathbf{x})$. In particular, the end-effector location $\mathbf{x}$ is still reachable when joint i is locked at $q_i = \theta_i$ if and only if θ_i is contained in the projection of $\mathbf{f}^{-1}(\mathbf{x})$ onto the i-th axis. The projection of a set $\mathbf{S} \subset \mathbb{R}^n$ is given by

$$P_i[\mathbf{S}] = \left\{ s_i \mid \mathbf{s} = [\, s_1 \quad s_2 \quad \cdots \quad s_n \,]^T \in \mathbf{S} \right\}. \tag{8}$$

Note that $P_i[\mathbf{S}]$ is a subset of $\mathbb{R}$ that can be thought of as the projection of $\mathbf{S}$ onto the i-th axis. Thus the end-effector location $\mathbf{x}$ can be reached after a locked-joint failure $q_i = \theta_i$ if and only if $\theta_i \in P_i[\mathbf{f}^{-1}(\mathbf{x})]$.

We can now formally state the characterizing conditions for the fault tolerant workspace $\boldsymbol{\mathcal{W}}_F$. A workspace region is failure tolerant to a single failure in joint $i \in \mathbf{F}$ for a given $\boldsymbol{\mathcal{C}}_A$ and a given $\boldsymbol{\mathcal{C}}_B$ if and only if the following two conditions hold:

Condition 1. Reachability prior to a failure: For any $\mathbf{x} \in \boldsymbol{\mathcal{W}}_F$,

$$\boldsymbol{\mathcal{C}}_A \cap \mathbf{f}^{-1}(\mathbf{x}) \neq \emptyset. \tag{9}$$

Condition 2. Reachability after a failure: For any $\mathbf{x} \in \boldsymbol{\mathcal{W}}_F$,

$$A_i \subset P_i[\mathbf{f}^{-1}(\mathbf{x})] \text{ for } i \in \mathbf{F}. \tag{10}$$

Condition 1 is simple enough; it merely says that any $\mathbf{x} \in \boldsymbol{\mathcal{W}}_F$ should be reachable prior to a failure, i.e., there is at least one configuration in the pre-image of $\mathbf{x}$ that is contained in the pre-failure configuration space $\boldsymbol{\mathcal{C}}_A$. Equation (9) is equivalent to $\mathbf{x} \in \boldsymbol{\mathcal{W}}_0$. Condition 2 is somewhat more complicated. In order for the manipulator to be capable of reaching a workspace configuration $\mathbf{x}$ following a locked-joint failure in joint i that can occur at any angle within its specified artificial limits, the pre-image of $\mathbf{x}$ must have at least one configuration whose i-th component is equal to that joint value. Condition 2 insures that this is true for all joint values within the individual artificial joint limits for each failure-prone joint. If Conditions 1 and 2 hold, the end-effector location $\mathbf{x}$ is failure tolerant to a locked-joint failure of any joint in the set $\mathbf{F}$. Equation (10) is equivalent to $\mathbf{x} \in \boldsymbol{\mathcal{W}}_i$ for $i \in \mathbf{F}$.

V. Identifying the Potential Fault Tolerant Workspace Boundaries

In Section II, we formulated the problem of identifying the fault tolerant workspace in terms of the intersection of the pre-failure workspace with the intersection of a continuous family of images of hypersurfaces ${}^i\boldsymbol{\mathcal{C}}(\theta_i)$ as θ_i is varied over $A_i, i \in \mathbf{F}$. Although mathematically correct, this approach is not a feasible method for identifying the fault tolerant workspace. In Section III, characterizing conditions based on the pre-images of workspace locations were given. Since closed form expressions for pre-images are difficult if not impossible to obtain, this approach is also not a feasible method for determining the fault tolerant workspace. Instead, we will use these conditions to identify candidate boundaries of the fault tolerant workspace.

Based on the two conditions introduced in Section III, we can develop necessary conditions for identifying the boundaries of the fault tolerant workspace $\boldsymbol{\mathcal{W}}_F$. Condition 1 relates to the workspace prior to a failure, and the techniques for finding its workspace boundaries were already developed in Section III. Condition 2 can be used to identify additional potential boundaries related to a locked-joint failure. Once all of the potential boundaries have been identified, one can readily test these candidate boundaries to determine the real boundaries of the fault tolerant workspace.

When a given location $\mathbf{x}$ is within the failure tolerant workspace, its pre-image satisfies Condition 2 for $i \in \mathbf{F}$; otherwise, Condition 2 will not be satisfied. Our goal is to identify those workspace locations that form the boundary between where Condition 2 is satisfied and where it is violated, i.e., a potential boundary of the fault tolerant workspace. This can occur in two ways: (Case I) the projection of the pre-image for this workspace location fails to contain an endpoint of A_i or (Case II) the projection of its pre-image becomes disjoint within A_i. We will systematically identify the complete set of these potential boundaries and introduce a procedure to extract the true boundary.

While the formulation of Condition 2 is based on the concept of a pre-image, when identifying boundaries, it is more convenient to work with self-motion manifolds. The *self-motion manifolds* of an end-effector location $\mathbf{x}$ are the disjoint, connected

subsets of the pre-image $\mathbf{f}^{-1}(\mathbf{x})$. For a manipulator with r degrees of redundancy

$$\mathbf{f}^{-1}(\mathbf{x}) = \bigcup_{i=1}^{N} \boldsymbol{\mathcal{M}}_i \tag{11}$$

where $\boldsymbol{\mathcal{M}}_i$ is the i-th r-dimensional self-motion manifold in the inverse kinematic pre-image such that $\boldsymbol{\mathcal{M}}_i \cap \boldsymbol{\mathcal{M}}_j = \emptyset$ when $i \neq j$ and N is the number of self-motion manifolds [37].[2] The term self-motion is meant to suggest the fact that as the robot moves along a self-motion manifold in the joint space, the end-effector location remains fixed so that the robot only moves itself and not an object located at the end effector.

When identifying the workspace boundaries, we will at times examine the minimum and maximum values of the different joint variables θ_i on the self-motion manifolds. For manipulators with a single degree of redundancy, the self-motion manifolds are smooth one-dimensional curves whose tangent vector is given by the null vector $\mathbf{n}_J$. Consequently, at minimum and maximum values of θ_i along the self-motion curve, we have $n_{Ji} = 0$. For manipulators with higher degrees of redundancy, the tangent space of the self-motion is given by the null space of the manipulator Jacobian.

We will now consider how to evaluate the potential boundaries characterized by Cases I and II. Case I has two sub-cases (a) and (b) that are illustrated in Figs. 2 and 3, respectively. Fig. 2 illustrates the sub-case I(a) when the violation occurs inside $\boldsymbol{\mathcal{C}}_B$. In this case, the self-motion manifold is tangent to the hyperplane corresponding to the artificial joint limit for joint i. When this occurs, $n_{Ji} = 0$. Thus, setting q_i to an artificial joint limit and solving for the remaining joints subject to the constraint $n_{Ji}(\mathbf{q}) = 0$ gives candidate joint configurations that may correspond to boundary points of the fault tolerant workspace. One can see from Fig. 2 that, in some sense, this can be viewed as an optimization problem in which one looks for extremal values of q_i along a self-motion manifold and those self-motion manifolds where these extremal values happen to occur on a hyperplane corresponding to an artificial joint limit are identified as possibly corresponding to a potential fault tolerant workspace boundary.

In Case I(b), illustrated in Fig. 3, the violation of Condition 2 occurs because the self-motion manifold exits $\boldsymbol{\mathcal{C}}_B \subset \mathbb{R}^n$ before the projection of the self-motion onto the i-th axis can cover A_i so that the self-motion manifold cannot be reached if joint i is locked at the endpoint of A_i subject to the physical requirements that the manipulator must operate within $\boldsymbol{\mathcal{C}}_B$. This case is characterized by the failure prone joint i being at an endpoint of A_i and some other joint $j \neq i$ being at an endpoint of B_j, where once again B_j is the set of joint values that are within the physical limits of joint j. In terms of a constrained optimization problem, Fig. 3 depicts a self-motion manifold for which the i-th component is maximized subject to the constraint $\mathbf{q} \in \boldsymbol{\mathcal{C}}_B$ where at least one of the constraints (in this case, the constraint on joint j) is active.

[2]The complexity of computing the self-motion manifold increases dramatically with an increase in the degree of redundancy. One would need to employ a technique similar to that described in [32].

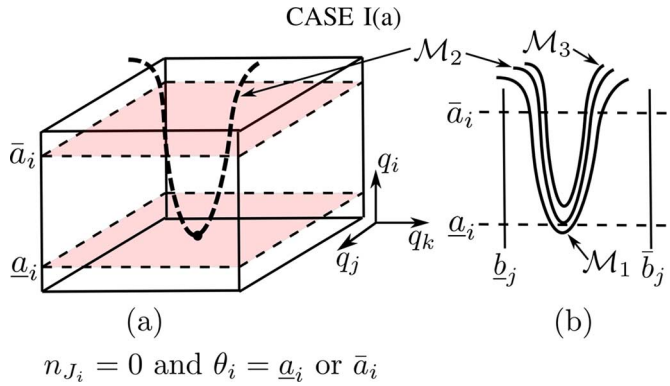


Fig. 2. An example of a self-motion manifold on the verge of violating Condition 2 at an endpoint of $A_i = [\underline{a}_i, \bar{a}_i]$. Arbitrarily near this self-motion manifold are self-motion manifolds that rise above the hyperplane $q_i = \underline{a}_i$, indicating that the corresponding end-effector location may not be within the fault tolerant workspace. Self-motion manifolds that dip below the same hyperplane can also be found arbitrarily near the self-motion manifold shown in the figure. Hence, the corresponding end-effector location may be a boundary point of the fault tolerant workspace. Since the self-motion manifold is tangent to the hyperplane, it follows that $n_{Ji} = 0$ when the self-motion manifold touches the hyperplane given by $q_i = \underline{a}_i$.

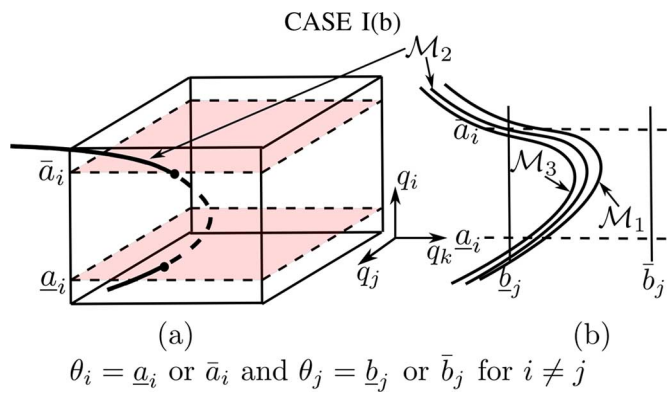


Fig. 3. Another example of a self-motion manifold on the verge of violating Condition 2 at an endpoint of $A_i = [\underline{a}_i, \bar{a}_i]$. The corresponding end-effector location may be a boundary point of the fault tolerant workspace. In this case, there will be similar self-motion manifolds arbitrarily close by that fail Condition 2 as they exit the configuration space $\boldsymbol{\mathcal{C}}_B \subset \mathbb{R}^n$ below the hyperplane given by $\theta_i = \bar{a}_i$. On the other hand, there will also be other self-motion manifolds arbitrarily close by that exit in such a way that Condition 2 is satisfied.

Case II also contains two sub-cases, (a) and (b) that are illustrated in Figs. 4 and 5, respectively, where Condition 2 fails within the interior of A_i. Fig. 4 illustrates the case where the self-motion manifold exits $\boldsymbol{\mathcal{C}}_B$ at a point where the manifold's projection is in the interior of A_i. Consequently, there are self-motion manifolds arbitrarily near the self-motion manifold in Fig. 4 that come outside of $\boldsymbol{\mathcal{C}}_B$ so that their projections onto the i-th axis are disconnected inside A_i.

In Case II(b), illustrated in Fig. 5, Condition 2 is violated in the interior of A_i, where the failure occurs in the interior of $\boldsymbol{\mathcal{C}}_B$ rather than at its boundary. This corresponds to a sudden change in the topological nature of the self-motion manifolds where the pre-image becomes disconnected. In this case, the changes in topology are identified by co-regular surfaces [37] and the candidate boundaries are given by the images of the kinematic singularities of the robot.

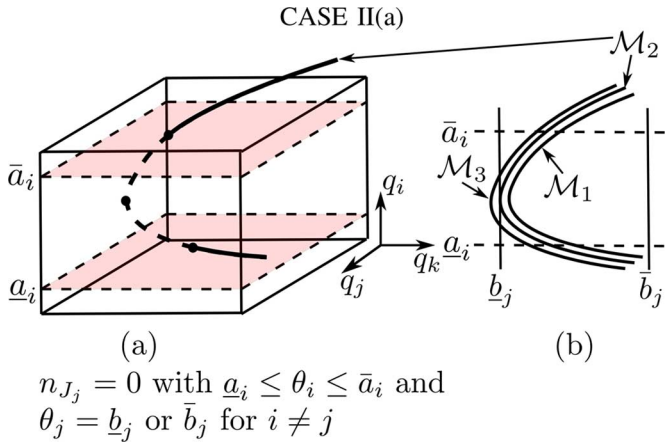


Fig. 4. An example of a self-motion manifold on the verge of violating Condition 2 at an interior point of A_i. The self-motion manifold is tangent to the j-th physical joint limit constraint so that $n_{Jj} = 0$. The corresponding end-effector location may be a boundary point of the fault tolerant workspace as there are nearby self-motion manifolds that appear within $\mathcal{C}_B$ satisfying Condition 2, but there are also self-motion manifolds arbitrarily close by which come out of $\mathcal{C}_B \subset \mathbb{R}^n$, violating Condition 2 in the interior of $A_i = [\underline{a}_i, \bar{a}_i]$.

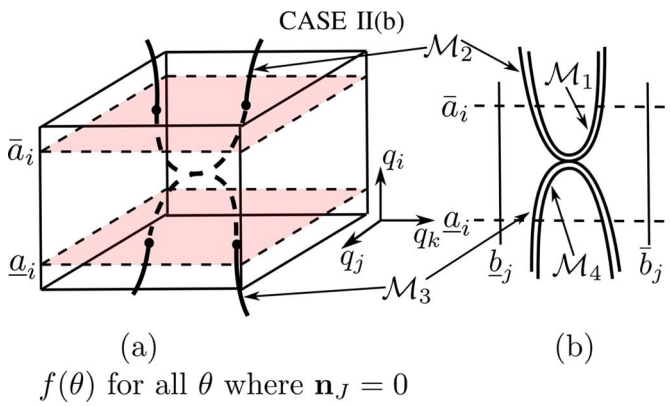


Fig. 5. An example of a portion of the configuration space containing a self-motion manifold that is not co-regular. The intersecting lines represent a set of joint values resulting in the same end-effector location. The configuration at the intersection is a kinematic singularity. Such configurations are associated with a fundamental change in the topology of the configuration space and may indicate a boundary point corresponding to a violation of Condition 2 at an interior point of A_i.

VI. Identifying True Workspace Boundaries

Now that the set of candidate boundaries have been determined, our goal is to extract the true boundary. This is done by first determining if there any intersections between the potential workspace boundaries. These intersections are easily determined by checking to see if any points simultaneously satisfy multiple potential boundary conditions. The potential boundaries are then segmented into simple non-intersecting curves.

Once all potential boundaries are separated into simple non-intersecting curves, each of these curves must be checked to determine if it is a true boundary. This can be efficiently done by selecting any convenient interior point on the curve and computing the tangent at that point. By checking two sufficiently close points that lie on the normal line but on opposite sides of this tangent, one can determine if the curve is a true boundary. This entire process is illustrated in Fig. 6 for a simple planar 3R manipulator.

The conditions for membership in the pre-failure workspace $\boldsymbol{\mathcal{W}_0}$ and a post-failure workspace $\boldsymbol{\mathcal{W}_i}$ are slightly different and are respectively given by:

Condition for membership in $\boldsymbol{\mathcal{W}_0}$

A point $\mathbf{x} \in \boldsymbol{\mathcal{W}_0}$ if and only if there exists a self-motion manifold $\boldsymbol{\mathcal{M}_k}$ corresponding to $\mathbf{x}$ such that for $\mathbf{q} \in \boldsymbol{\mathcal{M}_k}, \underline{a}_i \leq q_i \leq \bar{a}_i$, for $i = 1, \ldots, n$.

Condition for membership in $\boldsymbol{\mathcal{W}_i}$, where $i \in \mathrm{F}$

A point $\mathbf{x} \in \boldsymbol{\mathcal{W}_i}$ if and only if there exists a union of self-motion manifolds $\boldsymbol{\mathcal{M}_k}$ corresponding to $\mathbf{x}$ whose projection onto the i-th axis completely covers the range $[\underline{a}_i, \bar{a}_i]$ with $\underline{b}_j \leq \theta_j \leq \bar{b}_j$ for $j = 1, \ldots, n$ and $j \neq i$.

VII. An Algorithm for Computing the Fault Tolerant Workspace

We summarize the entire procedure for computing a fault tolerant workspace in the following algorithm.

Condition 1

Compute Potential Pre-Failure Workspace Boundaries

1. Kinematic Singularities:
Compute $\mathbf{f}(\boldsymbol{\theta})$ for all $\boldsymbol{\theta}$, where $\mathbf{n}_J = \mathbf{0}$.
2. Joint Limit Singularities:
(a) Single Joint:
Compute $\mathbf{f}(\theta)$ for all $\boldsymbol{\theta}$, where $q_i = \underline{a}_i$ or $\bar{a}_i$, and $n_{J_i} = 0$.
(b) Multiple Joint:
For all $i \neq j$ compute $\mathbf{f}(\boldsymbol{\theta})$ for all $\boldsymbol{\theta}$, where $q_i = \underline{a}_i$ and $q_j = \underline{a}_j$ or $q_i = \bar{a}_i$ and $q_j = \bar{a}_j$, where n_{J_i} and n_{J_j} are opposite in sign
or
$q_i = \underline{a}_i$ and $q_j = \bar{a}_j$ or $q_i = \bar{a}_i$ and $q_j = \underline{a}_j$, where n_{J_i} and n_{J_j} are of the same sign.

Compute the Pre-Failure Workspace Boundary $\boldsymbol{\mathcal{W}_0}$

3. Compute intersections of all curves computed in Steps 1 and 2 to generate simple non-intersecting curves.
4. For every curve generated by Step 3, check the $\boldsymbol{\mathcal{W}_0}$ membership condition of two sufficiently close points that lie on the normal line but on opposite sides of the tangent to any interior point on the curve.

Condition 2

Compute Potential Post-Failure Workspace Boundaries

5. For all $i \in \mathbf{F}$, compute $\mathbf{f}(\boldsymbol{\theta})$ for all $\boldsymbol{\theta}$ where $\mathbf{q} \in \boldsymbol{\mathcal{C}_B}$ and:
(a) Case Ia: $n_{J_i}(\mathbf{q}) = 0$ and $q_i = \underline{a}_i$ or $\bar{a}_i$.
(b) Case Ib: $q_i = \underline{a}_i$ or $\bar{a}_i$ and $q_j = \underline{b}_j$ or $\bar{b}_j$ for $i \neq j$.
(c) Case IIa: $n_{J_j}(\mathbf{q}) = 0$ with $\underline{a}_i \leq q_i \leq \bar{a}_i$ and $q_j = \underline{b}_j$ or $\bar{b}_j$ for $i \neq j$.
(d) Case IIb: Same as Step 1.

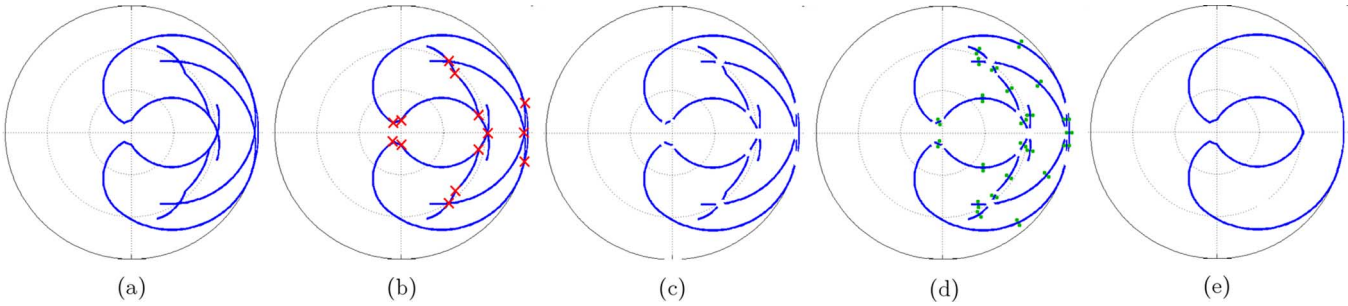


Fig. 6. An example of identifying the true workspace boundaries for a simple planar 3R manipulator. The potential boundaries are identified in (a) with all intersections computed in (b). This results in the disjoint set of segments in (c). A point on both sides of each segment is checked in (d) to determine which segments belong to the true boundary, which is illustrated in (e).

Compute the Post-Failure Workspace Boundaries $\mathcal{W}_i$

6. For all $i \in$ F
- i) **(a)** Compute intersections of all curves computed in Step 5 for a given i to generate simple non-intersecting curves.
- ii) **(b)** For every curve generated in (a), check the $\mathcal{W}_i$ membership condition of two sufficiently close points that lie on the normal line but on opposite sides of the tangent to any interior point on the curve.

Compute the Failure-Tolerant Workspace Boundary $\mathcal{W}_F$

7. Compute intersections of all curves computed in Steps 4 and 6 to generate simple non-intersecting curves.

8. For every curve generated in Step 7, check all $\mathcal{W}_i$ membership conditions for $i \in \mathbf{F} \cup \{0\}$ of two sufficiently close points that lie on the normal line but on opposite sides of the tangent to any interior point on the curve.

It should be noted that extending the above algorithm to spatial manipulators is conceptually easy, however, the associated computations become numerically challenging. Much of the difficulty is due to computing the intersections of potential boundaries, which now consist of (not necessarily spherical) surfaces in $\mathbb{R}^3$.

VIII. Illustrative Examples

A. Preliminaries

This section presents two simple examples to illustrate the application of the algorithm discussed in the previous section to various manipulator design problems. To simplify the visualization of the results, a planar 3-DOF robot is used. The forward kinematics of the planar 3R manipulator are given by

$$\mathbf{f}(\boldsymbol{\theta}) = \begin{bmatrix} l_1\cos(\theta_1) + l_2\cos(\theta_{12}) + l_3\cos(\theta_{123}) \\ l_1\sin(\theta_1) + l_2\sin(\theta_{12}) + l_3\sin(\theta_{123}) \end{bmatrix} \tag{12}$$

where l_i is the length of link i, $\theta_{12} = \theta_1 + \theta_2$, and $\theta_{123} = \theta_1 + \theta_2 + \theta_3$. The canonical null vector for a planar 3R manipulator is given by

$$\mathbf{n}_J = \begin{bmatrix} l_2 l_3 \sin(\theta_3) \\ -l_2 l_3 \sin(\theta_3) - l_1 l_3 \sin(\theta_2 + \theta_3) \\ l_1 l_2 \sin(\theta_2) + l_1 l_3 \sin(\theta_2 + \theta_3) \end{bmatrix}. \tag{13}$$

Note that $\mathbf{n}_J$ is not a function of θ_1.

For the examples presented here, we use the Mitsubishi PA-10-7C redundant manipulator, however, we lock joints 1, 3, 5, and 7 at 0° and then perform the analysis on the resulting planar 3-DOF arm. The link lengths of the PA-10 are $l_1 = 0.45$, $l_2 = 0.5$, and $l_3 = 0.08$ m, with the possibility of extending l_3 by an arbitrary tool offset. The range of the physical joint limits of the PA-10 are $\boldsymbol{\theta} = [\pm 94^\circ, \pm 143^\circ, \pm 180^\circ]^T$, however to accommodate a tool extension the third joint limit is restricted to $\pm 150^\circ$.

B. Tradeoff Between Pre-Failure and Post-Failure Workspaces

1) Problem Definition: In this example, we assume that one is interested in operating at a point that has a locally optimal fault-tolerant configuration, but that one would like to optimize the post-failure workspace as well. It has been previously shown [23] that an equal link length 3-DOF manipulator has an optimally fault tolerant Jacobian at a configuration of $\mathbf{q} = [0^\circ, 90^\circ, 90^\circ]^T$. Therefore, for this example we select a tool offset for the PA-10 that results in $l_3 = 0.45$, i.e., as close to equal link lengths as possible, and then analyze the guaranteed post-failure workspace for a range of artificial limits around the optimal configuration. This results in a classic case of two competing objectives where one must perform a bi-objective optimization to obtain the best possible pre-failure *and* post-failure workspace areas. Because there is an inherent tradeoff between these objectives whose optimal solution is dependent on the application, our goal is to provide flexibility to a designer by providing a set of "optimal" solutions depending on the relative importance of pre- and post-failure performance, i.e., we seek a set of Pareto optimal solutions. We denote the area of $\mathcal{W}_F$ with $\mathcal{A}_F$ and the area of $\mathcal{W}_0$ with $\mathcal{A}_0$ and set as our objective the simultaneous maximization of the normalized versions of these areas, i.e., $\mathcal{A}_F/\mathcal{A}_0$ versus $\mathcal{A}_0/\mathcal{A}_{0_{\max}}$, where

$\mathcal{A}_{0_{\max}}$ is the pre-failure workspace area without any artificial limits.

2) Brute Force Technique for Computing Pareto Optimal Solutions: As the first step to computing a set of Pareto optimal solutions to our bi-objective optimization, we performed a brute force calculation to determine how the pre- and post-failure workspaces, $\boldsymbol{\mathcal{W}_0}$ and $\boldsymbol{\mathcal{W}_F}$, vary as a function of the symmetric artificial joint limits. To this end, we discretized the two-dimensional workspace into a grid and for each grid square we computed the self-motion manifold [35], [38] of all possible configurations for our robot to reach that location. We check whether the self-motion manifold for each grid square satisfies the conditions given in Section V for membership in $\boldsymbol{\mathcal{W}_0}$ and $\boldsymbol{\mathcal{W}_i}$.

A grid square that satisfies the first condition for $\boldsymbol{\mathcal{W}_0}$ and second condition for $\boldsymbol{\mathcal{W}_i}$ for all $i \in \mathbf{F}$ belongs to the failure tolerant workspace $\boldsymbol{\mathcal{W}_F}$. The number of squares satisfying these two conditions become our estimates of $\mathcal{A}_0$ and $\mathcal{A}_F$, respectively. Clearly, the accuracy of these approximate area calculations increases at a higher grid resolution. The results of these calculations for a wide range of artificial joint limits, i.e., $[5^\circ, 5^\circ, 5^\circ] \leq \Delta\theta \leq [90^\circ, 50^\circ, 60^\circ]$ with a step size of 5°, are shown in Fig. 7. A portion of the Pareto optimal points that lie on the Pareto front are indicated with a red '*'. Clearly, these solutions represent different "optimal" values of the artificial joint limits for a range of different applications where the importance of pre-failure workspace area versus post-failure area varies. Fig. 8 illustrates three resulting workspaces from "optimal" configurations on the Pareto front, and clearly allows the designer to visualize the tradeoff between two desirable criteria, i.e., the maximization of pre-failure workspace and the percentage of the pre-failure workspace that can be guaranteed after a failure.

For the sake of illustration, assume that one is equally interested in both of these normalized areas. Therefore, one would be particularly curious about the region indicated in the zoomed view of Fig. 7, where $\mathcal{A}_F/\mathcal{A}_0 \approx \mathcal{A}_0/\mathcal{A}_{0_{\max}} \approx 0.2$. In this region, the boundaries of the workspaces correspond to those shown in Fig. 8(c). We show in the next section how one would exactly determine the equations for these workspace boundaries and thus compute exact values for these areas.

3) Computing Actual Boundary Equations: The brute force technique is a discretized version of the workspace and so the accuracy of the area computation depends on the resolution of the grid. An accurate calculation is possible by integrating over the actual boundary equations of the workspace. The equations for the boundary curves can be found by checking the self-motion manifold for a grid square just inside and outside of each workspace boundary to determine which condition is being violated. The above technique was used for points on the Pareto front and it was found that the boundary curves of $\boldsymbol{\mathcal{W}_0}$ and $\boldsymbol{\mathcal{W}_F}$ do not change for a portion of the Pareto front from $\Delta\theta = [23.3^\circ, 44.2^\circ, 39.9^\circ]$ to $\Delta\theta = [29.9^\circ, 50.6^\circ, 46.8^\circ]$.

This portion of the Pareto front is shown in the zoomed view of Fig. 7. The details of the three points indicated in the zoomed view are shown in Table I. The boundary curves for $\boldsymbol{\mathcal{W}_0}$ and $\boldsymbol{\mathcal{W}_F}$ for a point $\Delta\theta = [24.4^\circ, 45.1^\circ, 40.9^\circ]$ that falls within this range are shown in Figs. 9 and 10, respectively. The details of

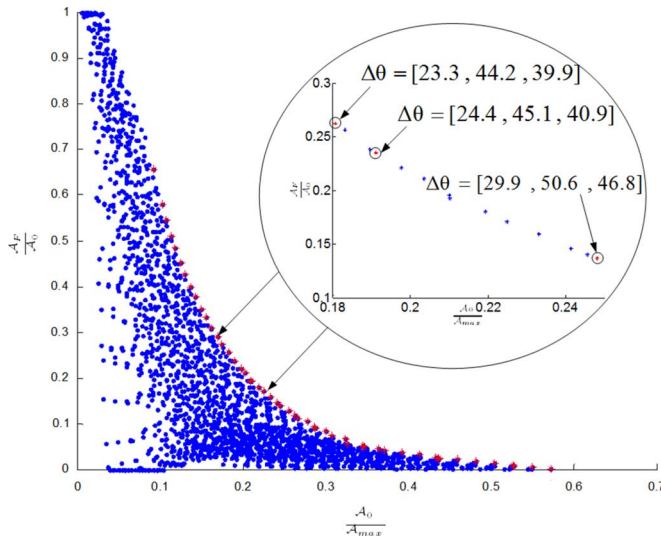


Fig. 7. Brute force analysis of the tradeoff between pre-failure workspace area $\mathcal{A}_0/\mathcal{A}_{0_{\max}}$ and failure tolerant workspace area $\mathcal{A}_F/\mathcal{A}_0$ over a range of artificial limits around the optimal configuration. The points indicated in red represent a set of Pareto optimal solutions, i.e., they represent solutions with the "best" tradeoff between $\mathcal{A}_0/\mathcal{A}_{0_{\max}}$ and $\mathcal{A}_F/\mathcal{A}_0$. The zoomed view of the Pareto front is near the area of interest, i.e., $\mathcal{A}_F/\mathcal{A}_0 \approx \mathcal{A}_0/\mathcal{A}_{0_{\max}} \approx 0.2$, over which the boundary curves for $\boldsymbol{\mathcal{W}_0}$ and $\boldsymbol{\mathcal{W}_F}$ do not change.

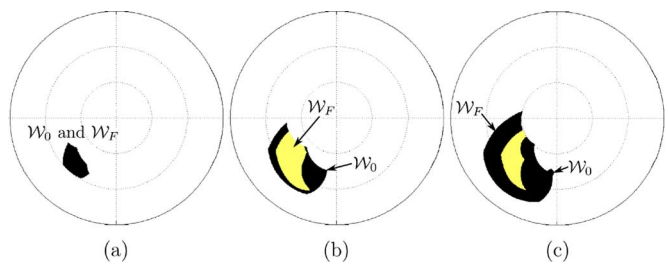


Fig. 8. Workspaces resulting from the three "optimal" configurations shown in Fig. 7, (a) represents $\mathcal{A}_F/\mathcal{A}_0 = 100\%$, (b) represents $\mathcal{A}_F/\mathcal{A}_0 = 50\%$, and (c) represents $\mathcal{A}_F/\mathcal{A}_0 = 20\%$.

these three points are shown in the Table I. The constraint equations of these boundaries are obtained by substituting the corresponding values of θ_i given in Tables II and III in the forward kinematic equation given by (12). The boundary condition satisfied by each curve is also given in the tables. Each curve is a circular arc with two of the joint angles having a constant value whereas the third is variable within a range. The range is determined by the joint angle values of the intersection point. The notation used for expressing a specific joint angle value i for a specific curve j is ${}^j\theta_i$. The notation used at a specific intersection point is as follows. Let j and k be two intersecting boundary curves and we are using the constraint equation of curve j to find the value of its variable joint i at the intersection point. This joint value is denoted by ${}^j_k\theta_i$. Similarly if we use the constraint equation of curve k to find the value of its variable joint i' at the same intersection point, this value is denoted by ${}^k_j\theta'_i$. Physically, these joint values belong to two points that map to the same intersection point in the workspace, but in the configuration space they are two separate points on the same self motion manifold. Examples of the equations for computing the joint values at the intersection points are shown in [39, Appendix].

TABLE I
DETAILS OF THE POINTS ON THE PARETO FRONT SHOWN IN FIG. 7

Point	$\Delta\theta$ (in degrees)	$\mathcal{A}_0$	$\mathcal{A}_F$	$\frac{\mathcal{A}_0}{\mathcal{A}_{0_{max}}}$	$\frac{\mathcal{A}_F}{\mathcal{A}_0}$
$P1$	[23.3°, 44.2°, 39.9°]	0.9408	0.2333	0.1729	0.2480
$P2$	[24.4°, 45.1°, 40.9°]	0.9932	0.2223	0.1826	0.2239
$P3$	[29.9°, 50.6°, 46.8°]	1.3030	0.1600	0.2395	0.1228

4) Computing the Workspace Areas Using Green's Theorem: Once the equations of the workspace boundaries are known, one can integrate over these curves to obtain the area of $\boldsymbol{\mathcal{W}_0}$ and $\boldsymbol{\mathcal{W}_F}$ by using Green's theorem [40], [41], i.e., given a region $D \subset \mathbb{R}^2$ bounded by a closed curve C,

$$g_{\mathcal{A}}(D) = \int\int_D d\mathcal{A} = \frac{1}{2}\int_C (x_1\, dx_2 - x_2\, dx_1) \tag{14}$$

where $g_{\mathcal{A}}(D)$ is the area of the region D. For computing the area of a workspace the closed loop C is a set of adjoining curves that form a closed loop. The workspace area is the sum of the areas under each curve computed in a counterclockwise direction over the loop. Consider a particular curve j with joint index i variable in its constraint equation. For such a curve, the workspace coordinates x_1 and x_2 are given as follows:

$$x_1 = f_1(\theta_i) \tag{15}$$
$$x_2 = f_2(\theta_i) \tag{16}$$

where $\mathbf{f} = [f_1(\theta_i), f_2(\theta_i)]$ is the forward mapping function given by (12) and θ_i is the variable joint. Therefore, Green's equation to compute area $\mathcal{A}_j$ under each curve j is of the form

$$\begin{aligned}\mathcal{A}_j &= \frac{1}{2}\int_{\theta_i} (f_1(\theta_i)\, df_2(\theta_i) - f_2(\theta_i)\, df_1(\theta_i)) \\ &= \frac{1}{2}\int_{\theta_i} \left(\left[f_1(\theta_i)\frac{df_2(\theta_i)}{d\theta_i} - f_2(\theta_i)\frac{df_1(\theta_i)}{d\theta_i} \right] d\theta_i \right).\end{aligned} \tag{17}$$

An example of the computations associated with this integration is given in [39, Appendix].

The equation for the area of $\boldsymbol{\mathcal{W}_0}$ in Fig. 9 is then given by

$$\mathcal{A}_0 = \sum_{j\in\{a,\ldots,h\}} \mathcal{A}_j \tag{18}$$

and the area of $\boldsymbol{\mathcal{W}_F}$ in Fig. 10 is given by

$$\mathcal{A}_F = \sum_{j\in\{p,\ldots,v\}} \mathcal{A}_j \tag{19}$$

where $\mathcal{A}_j$ is the area associated with curve j belonging to either $\boldsymbol{\mathcal{W}_0}$ or $\boldsymbol{\mathcal{W}_F}$, respectively. An example of the computations associated with determining these areas is given in [39, Appendix].

The above area equations allow one to exactly compute the optimal values of $\mathcal{A}_0/\mathcal{A}_{0_{\max}}$ and $\mathcal{A}_F/\mathcal{A}_0$ for different points along the desired Pareto front. In practice, this would be used by an automation engineer who is applying a robot to perform a task in an environment that requires fault tolerance to select an optimal set of artificial joint limits for the robot's controller that would guarantee task completion even after a failure.

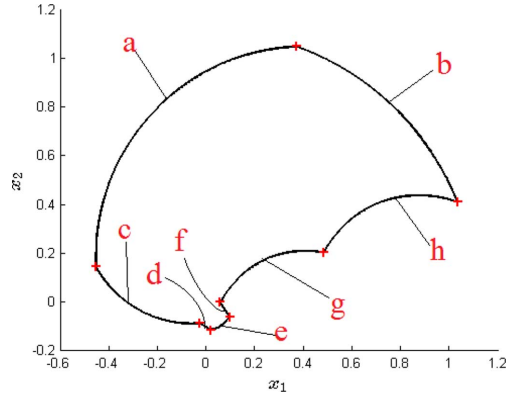


Fig. 9. Pre-failure workspace $\boldsymbol{\mathcal{W}_0}$ for $\Delta\theta_1 = 24.4°, \Delta\theta_2 = 45.1°$ and $\Delta\theta_3 = 40.9°$.

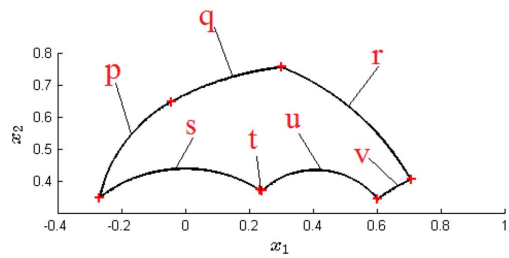


Fig. 10. Failure tolerant workspace $\boldsymbol{\mathcal{W}_F}$ for $\Delta\theta_1 = 24.4°, \Delta\theta_2 = 45.1°$ and $\Delta\theta_3 = 40.9°$.

C. Tool Offset Design for Fault Tolerance

Consider an example where one is not willing to lose any of the pre-failure workspace area by imposing any artificial joint limits, thereby eliminating the possibility of guaranteeing a failure-tolerant workspace. Although one is not willing to lose any pre-failure workspace, one is willing to place three different tool changers within the pre-failure operating space so that once a failure occurs, the tool (and resulting tool offset) can be changed. Under these design constraints, one is interested in determining:

i) An initial tool offset that will guarantee each of the post-failure workspaces ($\boldsymbol{\mathcal{W}_i}$ with $i = 1, 2, 3$) is non-empty.
ii) A tool extension that can be placed in each of these workspaces such that a common place in the workspace can be reached post-failure.
iii) The minimum area achieved for all three post-failure workspaces when the new tool extension is used.

To determine the initial tool offset required to satisfy (i), each of the three workspace areas $\boldsymbol{\mathcal{W}_i}$ with $i = 1, 2, 3$ can be computed as a function of l_3 using the results outlined in Section VII. Because no artificial joint constraints have been imposed on the manipulator pre-failure, the size of $\boldsymbol{\mathcal{W}_2}$ and $\boldsymbol{\mathcal{W}_3}$ are inversely proportional as l_3 varies. As a result, there is only a small range of allowable tool offset values, with the maximum area in each

TABLE II
BOUNDARIES FOR $\mathcal{W}_0$

Curve	θ_1	θ_2	θ_3	Boundary Condition Satisfied
a	$0+\Delta\theta_1$	$\frac{\pi}{2}-\Delta\theta_2$ to $\frac{\pi}{2}+\Delta\theta_2$	$\frac{\pi}{2}-\Delta\theta_3$	$q_1=\bar{a}_1$ and $q_3=\underline{a}_3$
b	$0-\Delta\theta_1$ to $0+\Delta\theta_1$	$\frac{\pi}{2}-\Delta\theta_2$	$\frac{\pi}{2}-\Delta\theta_3$	$q_2=\underline{a}_2$ and $q_3=\underline{a}_3$
c	$0+\Delta\theta_1$	$\frac{\pi}{2}+\Delta\theta_2$	$\frac{\pi}{2}-\Delta\theta_3$ to ${}^{c}_{d}\theta_3$	$q_1=\bar{a}_1$ and $q_2=\bar{a}_2$
d	$0-\Delta\theta_1$	$\frac{\pi}{2}+\Delta\theta_2$	${}^{d}_{c}\theta_3$ to $\frac{\pi}{2}+\Delta\theta_3$	$q_1=\underline{a}_1$ and $q_2=\bar{a}_2$
e	$0-\Delta\theta_1$ to $0+\Delta\theta_1$	$\frac{\pi}{2}+\Delta\theta_2$	$\frac{\pi}{2}+\Delta\theta_3$	$q_2=\bar{a}_2$ and $q_3=\bar{a}_3$
f	$0+\Delta\theta_1$	${}^{f}_{g}\theta_2$ to $\frac{\pi}{2}+\Delta\theta_2$	$\frac{\pi}{2}+\Delta\theta_3$	$q_1=\bar{a}_1$ and $q_3=\bar{a}_3$
g	$0-\Delta\theta_1$	$\frac{\pi}{2}-\Delta\theta_2$ to ${}^{g}_{f}\theta_2$	$\frac{\pi}{2}+\Delta\theta_3$	$q_1=\underline{a}_1$ and $q_3=\bar{a}_3$
h	$0-\Delta\theta_1$	$\frac{\pi}{2}-\Delta\theta_2$	$\frac{\pi}{2}-\Delta\theta_3$ to $\frac{\pi}{2}+\Delta\theta_3$	$q_1=\underline{a}_1$ and $q_2=\underline{a}_2$

TABLE III
BOUNDARIES FOR $\mathcal{W}_F$

Curve	θ_1	θ_2	θ_3	Boundary Condition satisfied
p	$0-\Delta\theta_1$	$\bar{b}_2$	0 to $\frac{\pi}{2}-\Delta\theta_3$	$q_1=\underline{a}_1$ and $q_2=\bar{b}_2$
q	$0-\Delta\theta_1$	${}^{q}_{r}\theta_2$ to $\bar{b}_2$	0	$n_{J_1}(\mathbf{q})=0$ and $q_1=\underline{a}_1$
r	${}^{r}_{q}\theta_1$ to ${}^{r}_{v}\theta_1$	$\frac{\pi}{2}+\Delta\theta_2$	$\arctan\left(\frac{l_1\cos\Delta\theta_2}{l_1\sin\Delta\theta_2-l_2}\right)$	$n_{J_2}(\mathbf{q})=0$ and $q_2=\bar{a}_2$
s	$\underline{b}_1$ to $0-\Delta\theta_1$	$\bar{b}_2$	$\frac{\pi}{2}-\Delta\theta_3$	$q_2=\bar{b}_2$ and $q_3=\underline{a}_3$
t	$\underline{b}_1$	${}^{t}_{u}\theta_2$ to $\bar{b}_2$	$\frac{\pi}{2}-\Delta\theta_3$	$q_1=\underline{b}_1$ and $q_3=\underline{a}_3$
u	$0+\Delta\theta_1$	${}^{u}_{v}\theta_2$ to ${}^{u}_{t}\theta_2$	$\bar{b}_3$	$q_1=\bar{a}_1$ and $q_3=\bar{b}_3$
v	$0-\Delta\theta_1$	$\frac{\pi}{2}-\Delta\theta_2$	${}^{v}_{r}\theta_3$ to ${}^{v}_{u}\theta_3$	$q_1=\underline{a}_1$ and $q_2=\underline{a}_2$

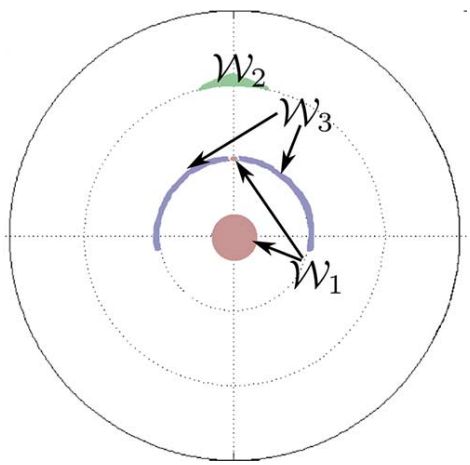


Fig. 11. Post-failure workspaces $\mathcal{W}_{1,2,3}$ resulting from an initial tool offset of $l_3 = 0.38$ meters and no artificial joint constraints. A tool changer in each of these three workspaces, allowing the manipulator to perform a tool change (thereby extending its reach) post failure.

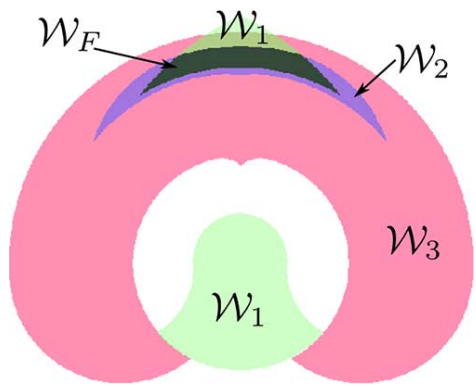


Fig. 12. Post-failure workspaces $\mathcal{W}_{1,2,3}$, as well as the failure-tolerant workspace $\mathcal{W}_F$ after a tool change is made post-failure extending the manipulators reach.

achieved when $l_3 = 0.38$ meters. The resulting workspace areas with this tool offset are shown in Fig. 11. One would then place a tool changer in each of these three workspaces, allowing the manipulator to perform a tool change post failure.

To determine the tool extension required in each workspace so that a common place within the workspace is reachable post-failure (ii), we rely on the techniques outlined in Section VII for computing the post-failure workspace boundaries, as well as the failure tolerant workspace boundary as a function of tool-offset. For the current manipulator, this analysis results in possible tool extensions given by $l_3 = 0.55, 0.50$, and 0.10 meters to be placed in each of the workspaces $\boldsymbol{\mathcal{W}_i}$ with $i = 1, 2, 3$ shown in Fig. 11. Using these tool extensions, the resulting post-failure workspaces, as well as the failure-tolerant workspace are illustrated in Fig. 12. Note that these workspaces are computed assuming that the manipulator can fail anywhere within the physical joint limits and a tool change is made post-failure extending the manipulators reach.

Finally, to determine the minimum usable operating space achievable for all three workspaces post-failure (i.e., after the new tool extension is used) (iii), each of the post-failure workspaces can computed using the algorithm outlined in Section VII for a range of potential failure configurations. For the current manipulator, the minimum operating space for $\boldsymbol{\mathcal{W}_i}$ with $i = 1, 2, 3$ occurs near the physical joint boundaries, i.e.,

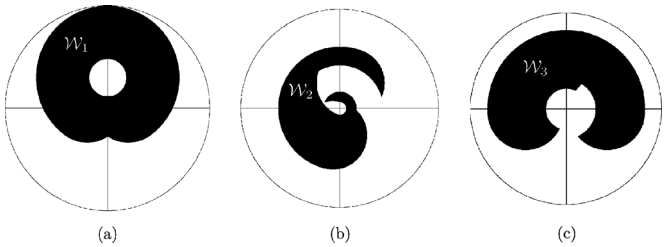


Fig. 13. Minimum operating space achievable for all three workspaces post-failure after the new tool extension is used. The minimum workspaces occur with joints locked at (a) $\mathbf{q} = 0^\circ$ with $l_3 = 0.55$ meters for $\mathcal{W}_1$, (b) 130° with $l_3 = 0.50$ meters for $\mathcal{W}_2$, and (c) 150° with $l_3 = 0.10$ meters for $\mathcal{W}_3$.

with joints locked at $\theta_1 = 0^\circ$ for $\boldsymbol{\mathcal{W}_1}, \boldsymbol{\theta_2} = 130^\circ$ for $\boldsymbol{\mathcal{W}_2}$, and $\theta_3 = 150^\circ$ for $\boldsymbol{\mathcal{W}_3}$. The resulting workspaces for each of these locked configurations are shown in Fig. 13 with $l_3 = 0.55, 0.50$, and 0.10 meters for $\boldsymbol{\mathcal{W}_1}, \boldsymbol{\mathcal{W}_2}$, and $\boldsymbol{\mathcal{W}_3}$, respectively.

IX. Conclusion

This paper presented a technique for designing a desired operating workspace for a kinematically redundant manipulator that can be guaranteed in the presence of an arbitrary single locked joint failure. By applying a judicious set of artificial joint constraints, it was shown that the boundaries of the failure tolerant workspace can be computed by evaluating certain conditions within the manipulator's configuration space. Based upon these conditions, an algorithm is developed to compute the pre- and post-failure workspace boundaries, as well as the failure tolerant workspace boundary $\boldsymbol{\mathcal{W}_F}$. Two design examples based upon the Mitsubishi PA-10 were presented to illustrate the application of the proposed algorithm to various workspace design problems.

In this work, we studied an example using a single degree of redundancy for a two-dimensional task space. Even though the presented approach is theoretically applicable to an arbitrary number of degrees of freedom, increasing either the dimension of the task space or the degree of redundancy increases the complexity of the calculations. Our future work will consider computationally efficient and numerically stable techniques for computing workspace boundaries that are surfaces (as opposed to curves) and self-motion manifolds that are described by a null space with more than one dimension.

References

[1] S. Soylu, B. J. Buckham, and R. P. Podhorodeski, "Redundancy resolution for underwater mobile manipulators," *Ocean Eng.*, vol. 37, no. 2–3, pp. 325–343, 2010.

[2] 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.

[3] L. Cragg and H. Hu, "Application of mobile agents to robust teleoperation of internet robots in nuclear decommissioning," in *Proc. IEEE Int. Conf. Ind. Tech.*, Maribor, Slovenia, Dec. 10–12, 2003, vol. 2, pp. 1214–1219.

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

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

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

[7] 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.

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

[9] I. Eski, S. Erkaya, S. Savas, and S. Yildirim, "Fault detection on robot manipulators using artificial neural networks," *Robot. Comput.-Integr. Manuf.*, vol. 27, no. 1, pp. 115–123, 2011.

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

[11] M. McIntyre, W. Dixon, D. Dawson, and I. Walker, "Fault detection and identification for robot manipulators," *IEEE Trans. Robot.*, vol. 21, no. 5, pp. 1028–1034, Oct. 2005.

[12] C. W. de Silva and K. Wong, "Online fault identification and fault-tolerant control of a multi-module manipulator," *Int. J. Robot. Autom.*, vol. 25, no. 3, pp. 217–228, 2010.

[13] N. Patarinsky-Robson, J. M. McCarthy, and I. Y. Tumer, "Failure recovery planning for an arm mounted on an exploratory rover," *IEEE Trans. Robot.*, vol. 25, no. 6, pp. 1448–1453, Dec. 2009.

[14] J. Park, W.-K. Chung, and Y. Youm, "Failure recoverability by exploiting kinematic redundancy," in *Proc. 5th Int. Workshop on Robot Human Commun*, Tsukuba, Japan, Nov. 11–14, 1996, pp. 298–305.

[15] Y. Ting, S. Tosunoglu, and R. Freeman, "Torque redistribution method for fault recovery in redundant serial manipulators," in *Proc. IEEE Int. Conf. Robot. Autom.*, San Diego, CA, USA, May 1994, vol. 2, pp. 1396–1401.

[16] 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.

[17] R. G. Roberts, H. G. Yu, and A. A. Maciejewski, "Designing optimally fault-tolerant manipulators based on relative manipulability indices," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1224–1237, Oct. 2008.

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

[19] C. Ukidve, J. McInroy, and F. Jafari, "Using redundancy to optimize manipulability of Stewart platforms," *IEEE Trans. Mechatronics*, vol. 13, no. 4, pp. 475–479, Aug. 2008.

[20] 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.

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

[22] A. A. Maciejewski, "Fault tolerant properties of kinematically redundant manipulators," in *Proc. IEEE Int. Conf. Robot. Autom.*, Cincinnati, OH, USA, May 13–18, 1990, pp. 638–642.

[23] 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. 543–553, Aug. 1996.

[24] 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.

[25] 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.

[26] Z. Jing, Y. Xuebin, and Z. Lei, "The optimization of initial posture with avoidance of the sudden change in joint velocity for fault tolerant operations of two coordinating redundant manipulators," *Mech. Mach. Theory*, vol. 40, no. 6, pp. 659–668, 2005.

[27] R. S. Jamisola, 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.

[28] S. K. Ralph and D. K. Pai, "Computing fault tolerant motions for a robot manipulator," in *Proc. IEEE Int. Conf. Robot. Autom.*, Detroit, MI, USA, May 10–15, 1999, pp. 486–493.

[29] C. J. J. Paredis and P. K. Khosla, "Fault tolerant task execution through global trajectory planning," *Reliability Eng. Syst. Safety*, vol. 53, no. 3, pp. 225–235, Sep. 1996.

[30] H. Abdi and S. Nahavandi, "Task completion with partially-failed manipulators," in *Proc. IEEE Conf. Robot. Autom. Mechatronics (RAM)*, 2010, Jun. 2010, pp. 269–274.

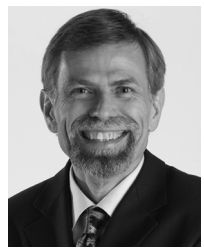
[31] C. J. J. Paredis and P. K. Khosla, "Designing fault-tolerant manipulators: How many degrees of freedom?," *Int. J. Robot. Res.*, vol. 15, no. 6, pp. 611–628, Dec. 1996.
[32] C. L. Lewis and A. A. Maciejewski, "Fault tolerant operation of kinematically redundant manipulators for locked joint failures," *IEEE Trans. Robot. Autom.*, vol. 13, no. 4, pp. 622–629, Aug. 1997.
[33] R. C. Hoover, R. G. Roberts, and A. A. Maciejewski, "Implementation issues in identifying the failure-tolerant workspace boundaries of a kinematically redundant manipulator," in *Proc. IEEE Int. Conf. Intell. Robot. Syst.*, San Diego, CA, USA, Oct. 29–Nov. 2 2007, pp. 3528–3533.
[34] 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.*, Roma, Italy, Apr. 10–14, 2007, pp. 4517–4523.
[35] C. Lück, "Self-motion representation and global path planning optimization for redundant manipulators through topology based discretization," *J. Intel. Robot. Syst.*, vol. 19, no. 1, pp. 23–38, May 1997.
[36] R. G. Roberts, "The dexterity and singularities of an underactuated robot," *J. Robot. Syst.*, vol. 18, no. 4, pp. 159–169, 2001.
[37] J. W. Burdick, "On the inverse kinematics of redundant manipulators: Characterization of the self-motion manifolds," in *Proc. IEEE Int. Conf. Robot. Autom.*, Scottsdale, AZ, USA, May 14–19, 1989, pp. 264–270.
[38] J. Lenarčič, "Some considerations on the self motion curves of a planar 3R manipulator," *J. Comput. Inform. Tech.*, vol. 10, no. 2, pp. 125–131, 2002.
[39] P. S. Naik, A. A. Maciejewski, R. G. Roberts, R. C. Hoover, and K. M. Ben-Gharbia, "An example of computing the failure-tolerant workspace area for a planar kinematically redundant robot," in *Proc. 9th IEEE Int. Conf. Autom. Sci. Eng.*, Madison, WI, USA, Aug. 17–21, 2013, pp. 312–317.
[40] J. P. Merlet, C. M. Gosselin, and N. Mauly, "Workspaces of planar parallel manipulators," *Mech. Mach. Theory*, vol. 33, no. 1/2, pp. 7–20, 1998.
[41] C. Gosselin, "Determination of the workspace of 6-DOF parallel manipulators," *ASME J. Mech. Design*, vol. 112, no. 3, pp. 331–336, 1990.

Randy C. Hoover (S'06–M'09) received the B.S. degree in electrical engineering and the M.S. degree in measurement and control engineering from Idaho State University, Pocatello, ID, USA, in 2002 and 2004, respectively, and the Ph.D. degree in electrical engineering from Colorado State University, Fort Collins, CO, USA, in 2009.

He was a National Science Foundation Fellow from 2003 to 2004. He is currently an Assistant Professor at the South Dakota School of Mines and Technology, where he directs the Computer Vision and Unmanned Systems Lab. His research interests are in the areas of computer vision, robotics, and control theory.

Rodney G. Roberts (SM'02) received the B.S. degree in electrical engineering and the B.S. degree in mathematics from 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 at Wright Patterson Air Force Base in 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 research interests are in the areas of robotics and image processing.

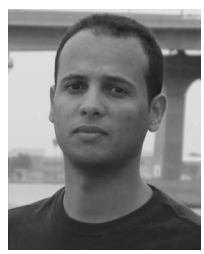


Anthony A. Maciejewski (F'05) received the B.S. E.E., M.S., and Ph.D. degrees from The Ohio State University, Columbus, OH, USA, in 1982, 1984, and 1987, respectively.

From 1988 to 2001, he was a Professor of Electrical and Computer Engineering at Purdue University, West Lafayette, IN, USA. He is currently the Department Head of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO, USA. A complete vita is available at: http://www.engr.colostate.edu/~aam.

Priya S. Naik (M'12) received the B.E. degree in electronics and telecommunication engineering from University of Pune, Pune, India, in 2009, and the M.S. degree in electrical engineering from the Colorado State University, Fort Collins, CO, USA.

She is currently working with Cummins Inc., Columbus, IN, USA. Her research interests are in the areas of robotics, controls and artificial intelligence.



Khaled M. Ben-Gharbia (S'11) received the B.S. and M.S. degrees in electrical and electronic engineering from University of Tripoli (formerly Al-Fateh University), Tripoli, Libya, in 2003 and 2008, respectively. He is currently working towards the Ph.D. degree in electrical engineering at Colorado State University, Fort Collins, CO, USA.