Proceedings of the 2007 IEEE/RSJ International
Conference on Intelligent Robots and Systems
San Diego, CA, USA, Oct 29 - Nov 2, 2007

Implementation Issues in Identifying the Failure-Tolerant Workspace Boundaries of a Kinematically Redundant Manipulator

Randy C. Hoover*, Rodney G. Roberts+, and Anthony A. Maciejewski*

*Dept. of Electrical and Computer Eng.
Colorado State University
Fort Collins, CO 80523-1373, USA
Email: {hooverr, aam}@colostate.edu

+Dept. of Electrical and Computer Eng.
Florida A & M - Florida State University
Tallahassee, FL 32310-6046, USA
Email: rroberts@eng.fsu.edu

Abstract— **In addition to possessing a number of other important properties, kinematically redundant manipulators are inherently more tolerant to locked-joint failures than non-redundant manipulators. However, a joint failure can still render a kinematically redundant manipulator useless if the manipulator is poorly designed or controlled. This paper focuses on the implementation issues involved in identifying a region of the workspace for which task completion is guaranteed in the event of a locked-joint failure for a general class of planar 3R manipulators. The existence of such a region, called a *failure-tolerant workspace*, will be guaranteed by imposing a suitable set of artificial joint limits prior to a failure. The authors have developed a Graphical User Interface (GUI) that computes all workspace boundaries of interest for the planar 3R manipulators. This GUI allows the user to explore different robot geometries, and adjust the artificial joint limits, in an attempt to gain further understanding of a manipulator's failure-tolerant workspace.**

I. Introduction

Kinematically redundant manipulators have a number of advantages over non-redundant manipulators including the potential for obstacle avoidance and greater dexterity. In this work, we consider the important advantage of failure tolerance and discuss some of the implementation issues associated with determining the failure-tolerant workspace of a redundant manipulator.

Failures can cause unnecessary delays due to repairs and can even pose a significant danger during task execution. Since a kinematically redundant manipulator has more joints than are required for its specified task, it is possible that the manipulator can still perform its required task even if it suffers a joint failure. Failure tolerance is particularly important for manipulators operating in hazardous or remote environments such as in space exploration [1] and nuclear waste disposal [2]. A number of studies have been dedicated to the assessment [3] and analysis [4], [5] of robot reliability. Other studies related to enhancing a robot's tolerance to failure include work on failure detection [6], layered failure tolerance control [7], failure tolerance by trajectory planning [8], kinematic failure recovery [9], and manipulators specifically designed for failure tolerance [10]. In this article, a motion planning technique is proposed that uses an *a priori* strategy so that joint failures can be gracefully accommodated.

One approach to guaranteeing failure tolerance is to add enough kinematic redundancy to compensate for a locked-joint failure. It was shown in [11] that to insure the failure tolerance of a manipulator operating in an m-dimensional workspace, $m + 2$ joints are required if the joint space is unconstrained. However, adding too much kinematic redundancy is typically undesirable and actually makes the robotic system more likely to fail. It has been shown that by judiciously choosing suitable joint constraints, one can significantly increase the manipulator's failure-tolerant workspace even if the manipulator only has one degree of redundancy. In [12], failure-tolerant workspaces that contain prescribed end-effector locations were identified by using bounding boxes in the configuration space that enclose self-motion manifolds corresponding to selected end-effector locations. In [13], a theoretical framework was developed by the authors in which artificial joint limits are used to determine the boundaries of a failure-tolerant workspace. By imposing a set of artificial joint limits prior to failure, a suitable post-failure workspace can be developed. Once a joint has failed, it is locked in its current position for the remainder of the task. In this case, the artificial joint limits are released, and the remaining healthy joints are allowed to freely move within their physical joint limits.

The current work focuses on the implementation issues involved in computing the failure-tolerant workspace for a general class of planar 3R manipulators based on the theoretical framework developed in [13]. The authors have developed a Graphical User Interface (GUI) that computes all workspace boundaries of interest for the planar 3R manipulators. This GUI allows the user to explore different robot geometries, in an attempt to gain further understanding of a manipulator's failure-tolerant workspace. This GUI is also useful in maximizing the failure-tolerant workspace for existing robot geometries.

The remainder of this paper is organized as follows: In Section II, the failure-tolerant workspace problem is mathematically formulated and the theoretical framework developed in [13] is discussed. In Section III, the identification of the failure-tolerant workspace boundaries is described and algorithms used to determine these boundaries are given. In Section IV, an illustrative example is given outlining the

1-4244-0912-8/07/$25.00 ©2007 IEEE.

implementation algorithms used in the GUI for determining the actual failure-tolerant workspace for the general class of planar 3R manipulators. This includes the extraction of each workspace boundary (both pre- and post-failure) as well as the failure-tolerant workspace. This section also includes a numerical example, with conclusions and future work presented in Section V.

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 $\mathbf{f} : \mathcal{C} \rightarrow \mathcal{W}$. In this work, we will assume that the configuration space $\mathcal{C}$ has the form $\boldsymbol{\mathcal{C}_B} = B_1 \times \cdots \times B_n$. If joint i has no physical joint limits, $B_i = \mathbb{R}$, and if joint i does have physical joint limits, $B_i = [\underline{b}_i, \overline{b}_i]$ where $\underline{b}_i < \overline{b}_i$. Initially we introduce artificial limits for each joint so that the i^{th} joint $q_i \in A_i = [\underline{a}_i, \overline{a}_i]$. If it can be safely assumed that joint i will not fail then we can set $A_i = B_i$. The set $\boldsymbol{\mathcal{C}_A} = A_1 \times \cdots \times A_n$ denotes the region of the configuration space corresponding to the artificial joint limits. The joint space prior to a failure is then simply $\boldsymbol{\mathcal{C}_A}$. 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. Generally there are 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, in spite of the locked joint, become reachable after releasing the artificial joint limits of the non-failed joints. The *failure-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 $\boldsymbol{\mathcal{C}_A}$.

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. The goal then is to determine the failure-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 $\boldsymbol{\mathcal{C}_A}$ and the pre-failure workspace is given by

$$\boldsymbol{\mathcal{W}_0} = \mathbf{f}(\boldsymbol{\mathcal{C}_A}) = \{\mathbf{x} = \mathbf{f}(\mathbf{q}) | \mathbf{q} \in \boldsymbol{\mathcal{C}_A}\}. \tag{1}$$

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

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

Geometrically, one can consider ${}^i\boldsymbol{\mathcal{C}}(\theta_i)$ to be the intersection of the hyperplane given by $q_i = \theta_i$ with the feasible configuration space $\boldsymbol{\mathcal{C}_B}$. Because the artificial joint limits were enforced prior to the failure, we have that $\underline{a}_i \leq \theta_i \leq \overline{a}_i$. 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

$$\boldsymbol{\mathcal{W}_i} = \bigcap_{\underline{a}_i \leq \theta_i \leq \overline{a}_i} \mathbf{f}({}^i\boldsymbol{\mathcal{C}}(\theta_i)). \tag{3}$$

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

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

The 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 conditions for its boundaries. We start with the following definition:

Definition 1: A boundary point of a subset $\boldsymbol{\mathcal{S}}$ of $\mathbb{R}^m$ is a point $\mathbf{x} \in \mathbb{R}^m$ such that every open neighborhood of $\mathbf{x}$ contains at least one point in $\boldsymbol{\mathcal{S}}$ and at least one point not in $\boldsymbol{\mathcal{S}}$.

Whether or not a given end-effector location is in the failure-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

$$\mathbf{f}^{-1}(\mathbf{x}) = \{\mathbf{q} \in \boldsymbol{\mathcal{C}_B} | \mathbf{f}(\mathbf{q}) = \mathbf{x}\}. \tag{5}$$

We can now formally state the characterizing conditions for the failure-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 [13]:

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

$$\boldsymbol{\mathcal{C}_A} \cap \mathbf{f}^{-1}(\mathbf{x}) \neq 0. \tag{6}$$

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

$$A_i \subset P_i[\mathbf{f}^{-1}(\mathbf{x})] \text{ for } i \in \mathbf{F}. \tag{7}$$

where $P_i[\mathbf{f}^{-1}(\mathbf{x})]$ is the projection of the pre-image of workspace location $\mathbf{x}$ onto the i^{th} axis.

III. Identifying the Failure-Tolerant Workspace Boundaries

A. Overview

In the following subsections we present a technique for computing the actual failure-tolerant workspace boundary. Our approach to identifying the failure-tolerant workspace boundary, is to first identify all of the potential workspace boundaries that satisfy Conditions 1 and 2. Using these potential workspace boundaries, the intermediate failure-tolerant workspace boundaries $\boldsymbol{\mathcal{W}_i}$, ($i \in \mathbf{F}$), as well as the final failure-tolerant workspace boundary $\boldsymbol{\mathcal{W}_F}$ can be computed.

B. Potential Boundaries for the Pre-Failure Workspace

The potential workspace boundaries of the pre-failure workspace are characterized by kinematic singularities and joint-limit singularities (the term *semi-singularity* is also used [14]). 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 the manipulator Jacobian. Hence, the kinematic singularities are those configurations for which each element n_{J_i} of $\mathbf{n}_J$ is zero.

Identifying joint-limit singularities is more challenging, 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. It can be shown however that evaluation of the canonical null vector $\mathbf{n}_J$ plays a crucial role in the evaluation of joint-limit singularities as well [13]. In particular, the components of $\mathbf{n}_J$ corresponding to actively constrained joints must have the appropriate sign in order for those joints to move away from their joint limits. More details on single and multiple joints being at their limits as well as multiple degrees of redundancy can be found in [13] and [15]. As an example, for the general class of planar 3R manipulators, there are 27 different cases to consider for joint limit singularities. Steps for determining the kinematic and joint limit singularities for the pre-failure workspace are outlined in the following algorithm:

Algorithm for Identifying the Potential Pre-Failure Workspace Boundaries (Condition 1).

Step 1. Determine all joint values where $\mathbf{n}_J = \mathbf{0}$ and determine their images.
Step 2. Set $q_i = \underline{a}_i$ or $\overline{a}_i$, find where $n_{J_i} = 0$, and determine their images.
Step 3. Set $q_i = \underline{a}_i$ with $q_j = \underline{a}_j$ or $q_i = \overline{a}_i$ with $q_j = \overline{a}_j$, find where n_{J_i} and n_{J_j} are opposite in sign, and determine their images for $i \neq j$.
Step 4. Set $q_i = \underline{a}_i$ with $q_j = \overline{a}_j$ or $q_i = \overline{a}_i$ with $q_j = \underline{a}_j$, find where n_{J_i} and n_{J_j} are of the same sign, and determine their images for $i \neq j$.

C. Potential Boundaries for the Post-Failure Workspace

The potential workspace boundaries of the post-failure workspace can be characterized by the concept of a pre-image and Condition 2. In Section II, we outlined characterizing conditions for determining $\boldsymbol{\mathcal{W}_F}$ based on the pre-images of workspace locations. Because closed form expressions for pre-images are difficult if not impossible to obtain, this approach is not a feasible method for determining the failure-tolerant workspace. Instead, we will use these conditions to identify candidate boundaries of the failure-tolerant workspace.

Condition 1 relates to the pre-failure workspace, however; Condition 2 can be used to identify potential boundaries relating to a locked-joint failure, i.e., the post-failure potential boundaries. 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,

$$\mathbf{f}^{-1}(\mathbf{x}) = \bigcup_{i=1}^{N} \boldsymbol{\mathcal{M}_i}, \tag{8}$$

where $\boldsymbol{\mathcal{M}_i}$ is the i^{th} r-dimensional self-motion manifold in the inverse kinematic pre-image such that $\boldsymbol{\mathcal{M}_i} \cap \boldsymbol{\mathcal{M}_j} = 0$ when $i \neq j$ and N is the number of self-motion manifolds [16].

In the evaluation of the boundary conditions for the areas defined by Condition 2, there are two cases to consider, each with two separate sub-cases [13]:

Case I. The projection of the pre-image for a specific workspace location fails to contain an endpoint of A_i.
(a) The self-motion manifold is tangent to the hyperplane corresponding to the artificial joint limit for joint i, and thus a violation of Condition 2 occurs inside $\boldsymbol{\mathcal{C}_B}$.
(b) The self-motion manifold exits $\boldsymbol{\mathcal{C}_B}$ before the projection of the self-motion onto the i^{th} axis can cover A_i.

Case II. The projection of the pre-image for a specific workspace location becomes disjoint within A_i.
(a) The self-motion manifold exits $\boldsymbol{\mathcal{C}_B}$ at a point where the manifold's projection is in the interior of A_i.
(b) There is a sudden change in the topology of the pre-image due to Condition 2 being violated in the interior of A_i where the failure occurs in the interior of $\boldsymbol{\mathcal{C}_B}$ rather than its boundary.

The steps used in determining the potential post-failure workspace boundaries are outlined in the following algorithm:

Algorithm for Identifying the Potential Post-Failure Workspace Boundaries (Condition 2).

Step 1. For $i \in \mathbf{F}$, find configurations in $\mathbf{q} \in \boldsymbol{\mathcal{C}_B}$ satisfying at least one of the following conditions and determine their images:
(a) $n_{J_i}(\mathbf{q}) = 0$ where $q_i = \underline{a}_i$ or $\overline{a}_i$.
(b) $q_i = \underline{a}_i$ or $\overline{a}_i$ with $q_j = \underline{b}_j$ or $\overline{b}_j$ for some $i \neq j$.
(c) $n_{J_i}(\mathbf{q}) = 0$ for $\underline{a}_i \leq q_i \leq \overline{a}_i$ and $q_j = \underline{b}_j$ or $\overline{b}_j$

for some $i \neq j$.

Step 2. Determine all joint values where $\mathbf{n}_J = \mathbf{0}$ and determine their images.

D. Identifying the True Workspace Boundaries $\mathcal{W}_i$ and $\mathcal{W}_F$

Once the pre- and post-failure potential boundaries have been identified, one can readily test these candidate boundaries to identify the true workspace boundary. Note that if the manipulator is operating in $\mathbb{R}^2$, these boundaries will be 1-dimensional curves; if however the manipulator is operating in $\mathbb{R}^3$, these boundaries will be 2-dimensional surfaces. The goal here is to identify the true pre-failure workspace boundary, and all post-failure workspace boundaries. Once these boundaries have been determined, we formulate how to determine the final failure-tolerant workspace boundary $\mathcal{W}_F$ from these intermediate boundaries.

The true workspace boundary for workspace $\mathcal{W}_i$, ($i \in \mathbf{F}$), is computed by first finding the intersections of all potential boundaries, and then evaluating each of these simple non-intersecting curves(surfaces) to verify if it is a member of the true workspace boundary. Note that this can be efficiently done by evaluating any convenient point along the curve(surface). This is true because if one point on the non-intersecting potential boundary curve(surface) satisfies the membership conditions (outlined below), then the entire curve(surface) must satisfy the membership conditions. This is done simply by selecting two points that are perpendicular to the tangent of the non-intersecting potential boundary curve(surface) but lie on opposite sides. If only one of these points satisfies the membership conditions, then this portion of the potential boundary is a portion of the true workspace boundary by Definition 1. The conditions for membership are as follows:

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

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

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

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

In the above conditions, we remind the reader that $\mathcal{M}_k$ is the k^{th} r-dimensional self-motion manifold in the inverse kinematic pre-image. Using these conditions, the workspace boundaries for $\mathcal{W}_i$, ($i \in \mathbf{F} \cup \{0\}$), can be calculated.

Once the final pre- and post-failure workspace boundaries have been determined, we can easily find $\mathcal{W}_F$ from these intermediate boundaries. The final failure-tolerant workspace boundary $\mathcal{W}_F$ is calculated using the following algorithm.

Algorithm for Identifying the Failure-Tolerant Workspace Boundary $\mathcal{W}_F$

Step 1. Determine the final pre- and post-failure workspace boundaries.

Step 2. Calculate all the intersections of the final pre- and post-failure workspace boundaries.

Step 3. Test each simple non-intersecting curve (surface) for membership in $\mathcal{W}_F$.

Note, membership in $\mathcal{W}_F$ is determined by Definition 1, and each point on the simple non-intersecting curve(surface) passing the membership conditions for all workspace boundaries $\mathcal{W}_i$ with $i \in \mathbf{F} \cup \{0\}$.

IV. Illustrative 3-DOF Example

In this section, an example of the implementation of the algorithms used by the GUI is given. The identification of all workspace boundaries (including the failure-tolerant workspace) is discussed from an implementation point of view. Therefore, it is useful to know how the potential boundaries identified by the algorithms given in Section III are stored and processed. It is also important to note that we are analyzing a general class of planar 3R manipulators with rotary joints. The forward kinematics of the planar 3R manipulator are given by

$$\mathbf{f}(\cdot) = \begin{bmatrix} f_x(\theta_1, \theta_2, \theta_3) \\ f_y(\theta_1, \theta_2, \theta_3) \end{bmatrix} = \begin{bmatrix} l_1 c_1 + l_2 c_{12} + l_3 c_{123} \\ l_1 s_1 + l_2 s_{12} + l_3 s_{123} \end{bmatrix} \quad (9)$$

where l_i is the length of link i, and $c_{ijk} = \cos(\theta_i + \theta_j + \theta_k)$, etc.

All potential boundaries of the pre- and post-failure workspace are stored by case and by joint values in $\hat{\mathcal{W}}_i$ where $\hat{\mathcal{W}}_i$ is denoted as the cell-array for the potential boundaries corresponding to $\mathcal{W}_i$. The term cell-array here is defined as an array of arrays with potentially different dimensions. For example, the pre-failure workspace potential boundaries are stored as $\hat{\mathcal{W}}_0 = \{\text{case}\#, \hat{\Theta}_1, \hat{\Theta}_2, \hat{\Theta}_3\}$, where $\hat{\Theta}_j$ and $\hat{\Theta}_k$ are fixed, and $\hat{\mathbf{\Theta}}_i$ contains a minimum and maximum value for $i \neq j$, $j \neq k$, and $i \neq k$. This results from the fact that for the 3-DOF manipulator, all images that correspond to candidate potential boundaries based on the algorithms outlined in Section III, have two joint variables fixed, while the third is free. For example, consider step 1 of the algorithm for finding the potential pre-failure workspace boundaries. Given the canonical null vector $\mathbf{n}_J$ for the planar 3R manipulators

$$\mathbf{n}_J = \begin{bmatrix} l_2 l_3 \sin(\theta_3) \\ -l_2 l_3 \sin(\theta_3) - l_1 l_3 \sin(\theta_2 + \theta_3) \\ l_1 l_2 \sin(\theta_2) + l_1 l_3 \sin(\theta_2 + \theta_3) \end{bmatrix}, \quad (10)$$

where l_i is the length of link i, step 1 requires $\mathbf{n}_J = 0$. Because $\mathbf{n}_J$ is only a function of θ_2 and θ_3, this leaves θ_1 as the free parameter. It is also obvious that $\mathbf{n}_J = 0$ when $(\theta_2, \theta_3) = (k\pi, l\pi)$ where $k, l \in \mathbb{Z}$. Therefore, if $\underline{a}_i > -\pi$ and $\overline{a}_i < \pi$, $(i = 2, 3)$, this step generates one image (potential boundary segment), and the first element in $\hat{\mathcal{W}}_0$ is $\{1, \hat{\mathbf{\Theta}}_1, 0, 0\}$ where $\hat{\mathbf{\Theta}}_1 = [\underline{a}_1, \overline{a}_1]$. The image is then generated by using (9).

Once the potential boundary segments are stored in $\hat{\mathcal{W}}_i$, the true workspace boundary is determined by first finding the intersections of all potential boundaries, and then evaluating each of these simple non-intersecting curves to verify if

it is a member of the true workspace boundary. Because the potential workspace boundaries are generated by the forward kinematics of the manipulator with θ_j and θ_k fixed, and θ_i free, $i \neq j$, $j \neq k$, and $i \neq k$, this constitutes a set of circular arcs in $\mathbb{R}^2$ with different centers and radii. Therefore, the potential boundaries in question consist of 1-dimensional curves, and the intersection of each of these curves can be easily determined.

Once the intersections in the workspace are calculated, the resulting joint space values generating the intersections need to be calculated. Because the inverse mapping ($\mathbf{q} = \mathbf{f}^{-1}(\mathbf{x})$) is nonlinear, this solution is generally not easy to obtain. More importantly, because we are dealing with redundant manipulators, the number of solutions to the inverse problem may be infinite. However, the forward kinematics are readily available, and two of the joint variables are fixed, therefore, solving for the joint space intersections can be formulated as follows:
Given the end-effector equations in (9), we formulate

$$\mathbf{g}(\theta_i) = \mathbf{f}(\theta_i) - \begin{bmatrix} x \\ y \end{bmatrix}, \tag{11}$$

where θ_i is the free parameter, x and y are the intersection coordinates in the workspace, and θ_j and θ_k are fixed for $i \neq j$, $j \neq k$, and $i \neq k$. We then propose the following:

$$\text{minimize}\{\|\mathbf{g}(\theta_i)\|^2 | \theta_{i_{\min}} \leq \theta_i \leq \theta_{i_{\max}}\}, \tag{12}$$

where $\theta_{i_{\min}} = \underline{a}_i$ or $\underline{b}_i$ and $\theta_{i_{\max}} = \overline{a}_i$ or $\overline{b}_i$ depending on the case/boundary. Because we know an intersection exists at (x, y), the minimum of (12) is zero yielding the θ_i intersection in the joint space[1].

Checking each potential boundary against all other potential boundaries yields all intersections associated with a particular boundary. The intersections are then post-processed to generate the cell-array $\check{\boldsymbol{\mathcal{W}}}_{\boldsymbol{i}} = \{\text{case}\#, \check{\Theta}_1, \check{\Theta}_2, \check{\Theta}_3\}$ that corresponds to all non-intersecting potential boundary curves associated with $\boldsymbol{\mathcal{W}_i}$. In the construction of $\check{\boldsymbol{\mathcal{W}}}_{\boldsymbol{i}}$, $\check{\Theta}_j$ and $\check{\Theta}_k$ are fixed, and $\check{\boldsymbol{\Theta}}_i = [a_0, a_1, \cdots, a_m]$ with $a_0 = \theta_{i_{\min}} < a_1 < \cdots < a_m = \theta_{i_{\max}}$ for $i \neq j$, $j \neq k$, and $i \neq k$. These represent the θ_i values where an intersection has been computed for each case (potential boundary) using (12). Using this formulation, each simple non-intersecting potential boundary curve can be calculated using (9) by letting $\theta_j = \check{\Theta}_j$ and $\theta_k = \check{\Theta}_k$, and $a_p \leq \theta_i \leq a_{p+1}$ for $p = 0, 1, \ldots, m-1$, with $i \neq j$, $j \neq k$, and $i \neq k$. Note all of the information required to generate such curves is contained in $\check{\boldsymbol{\mathcal{W}}}_{\boldsymbol{i}}$.

Once $\check{\boldsymbol{\mathcal{W}}}_{\boldsymbol{i}}$ has been constructed, each simple non-intersecting potential boundary curve needs to be evaluated to see if it is a member of the true workspace boundary. This is done simply by selecting two points $(\mathbf{p}_{\text{u}}, \mathbf{p}_{\text{l}})$ that are perpendicular to the tangent of the potential boundary but lie on opposite sides. If only one of these points satisfies the membership conditions (given in Section III), then this portion of the potential boundary is a portion of the true workspace boundary by Definition 1. Mathematically, this is completed as follows:
First, choose a θ value that lies between a_p and a_{p+1}, for example one might choose, $\theta_{\text{c}} = \frac{a_p + a_{p+1}}{2}$ and define

$$\mathbf{t} = \left[\frac{\partial f_x}{\partial \theta_i}, \frac{\partial f_y}{\partial \theta_i}\right]^T \Bigg|_{\theta_i = \theta_{\text{c}}} \tag{13}$$

to compute the tangent evaluated at $\theta_i = \theta_{\text{c}}$. Next, compute the normal vector $\mathbf{n}$ such that, $\mathbf{n} \perp \mathbf{t}$. Finally, compute $\mathbf{p}_{\text{u}} = \mathbf{f}(\theta_{\text{c}}) + \epsilon\mathbf{n}$, and $\mathbf{p}_{\text{l}} = \mathbf{f}(\theta_{\text{c}}) - \epsilon\mathbf{n}$. Note that for ϵ sufficiently small, the points $(\mathbf{p}_{\text{u}}, \mathbf{p}_{\text{l}})$ become arbitrarily close to the true potential workspace boundary.

Once these upper and lower perpendicular points are calculated, each point needs to be checked to verify if this portion of the potential boundary curve is a member of the true workspace boundary. This is done by evaluating the self-motion manifolds of the manipulator corresponding to this workspace location. As mentioned previously, closed form solutions to the self-motion manifolds are generally impossible to calculate, so instead, we use the fact that 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 $\mathbf{n}_J$ [13]. A method for computing the self-motion manifolds using the null vector $\mathbf{n}_J$ is detailed in [17], an alternative approach is detailed in [18].

Once the self-motion manifolds for given workspace locations $(\mathbf{p}_{\text{u}}, \mathbf{p}_{\text{l}})$ have been computed, the manifolds need to be evaluated to determine if this portion of the potential boundary curve is a member of the true workspace boundary. This evaluation is dependent on the workspace under consideration and the conditions for membership outlined in Section III.

As an example of the GUI's usefulness, consider the planar 3R manipulator whose link lengths are given by $L = [1, 2, 1]^T$ meters, symmetric artificial joint limits of $[\pm 11, \pm 100, \pm 150]$ degrees, and no physical joint limits. Figure 1 shows all pre- and post-failure boundaries $\boldsymbol{\mathcal{W}_i}$ with $i = 0, 1, 2, 3$, as well as the failure-tolerant workspace $\boldsymbol{\mathcal{W}_F}$. As can be seen from the figure, the failure-tolerant workspace is very narrow and almost disjoint. This is due to the fact that the failure-tolerant workspace boundary due to a failure in joint θ_3, i.e., $\boldsymbol{\mathcal{W}_3}$ is very narrow. If however, the artificial joint limits in θ_3 are decreased to $[\pm 11, \pm 100, \pm 100]$ degrees, the failure-tolerant workspace can be significantly increased (Fig. 2(a)). This is counter-intuitive in that decreasing the artificial joint limits actually increases the final failure-tolerant workspace. If we continue to decrease the artificial joint limits in θ_3 however, the failure-tolerant workspace becomes disjoint, i.e., it separates into two different workspaces (Fig. 2(b)). While the total area of both workspaces may be larger than that of Fig. 2(a), this separation is typically undesirable. This is due to the fact that if the manipulator fails in one of the two workspaces, the path needed to reach the remaining workspace cannot be

[1]For the 3-DOF planar example the joint space intersection can be computed in closed form using a change of variables, however the authors use the optimization technique to be more general.

determined *a priori*, and must be outside the failure-tolerant workspace area.

V. Conclusions and Future Work

This paper discussed some of the implementation issues associated with determining the failure-tolerant workspace of a redundant manipulator based on the theoretical framework developed in [13]. We have outlined step-by-step algorithms in computing the pre- and post-failure workspace boundaries, as well as the failure-tolerant workspace boundary $\mathcal{W}_F$. The authors have developed a GUI that is effective in computing these workspace boundaries in an attempt to allow the user to gain a further understanding of different robot geometries. We have illustrated the use of the GUI, and have shown its usefulness in maximizing the failure-tolerant workspace by applying a judicious set of artificial joint limits prior to failure. Future work will focus on optimizing the failure-tolerant workspace subject to constraints in robot geometry and artificial joint limits, as well as computing the area of the failure-tolerant workspace [19], [20].

References

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

[2] R. Colbaugh and M. Jamshidi, "Robot manipulator control for hazardous waste-handling applications," *J. Robot. Syst.*, vol. 9, no. 2, pp. 215–250, 1992.

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

[4] B. S. Dhillon and A. R. M. Fashandi, "Safety and reliability assessment techniques in robotics," *Robotica*, vol. 15, no. 6, pp. 701–708, Nov./Dec. 1997.

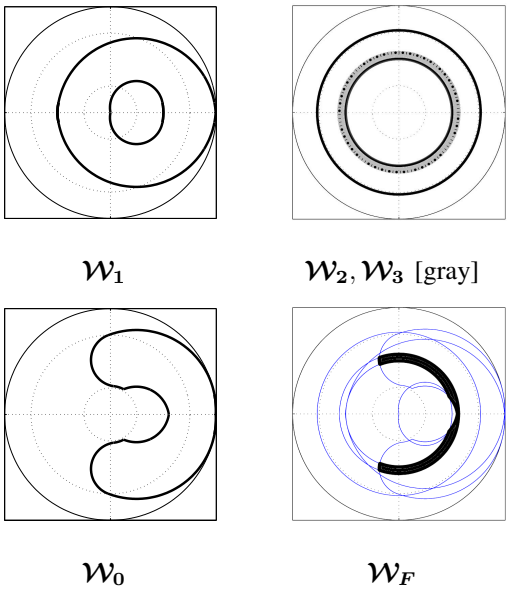

Fig. 1. The pre- and post-failure workspace boundaries, as well as the failure-tolerant workspace boundary $\mathcal{W}_F$ for $L = [1, 2, 1]^T$ meters, and artificial joint limits $[\pm 11, \pm 100, \pm 150]$ degrees. Note that $\mathcal{W}_3$ is contained in $\mathcal{W}_2$.

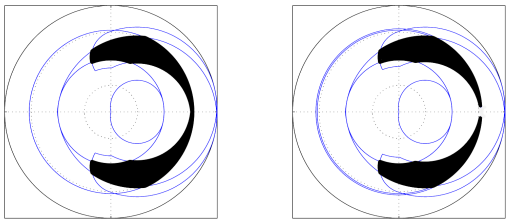

Fig. 2. The failure-tolerant workspace boundary $\mathcal{W}_F$ for $L = [1, 2, 1]^T$ meters, with artificial joint limits (a) $[\pm 11, \pm 100, \pm 100]$ and (b) $[\pm 11, \pm 100, \pm 97]$ degrees.

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

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

[7] Y. Ting, S. Tosunoglu, and D. Tesar, "A control structure for fault-tolerant operation of robotic manipulators," in *IEEE Int. Conf. Robot. Automat.*, Atlanta, GA, May 2-10 1993, pp. 684–690.

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

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

[10] Y. Yi, J. E. McInroy, and Y. Chen, "Fault tolerance of parallel manipulators using space and kinematic redundancy," *IEEE Trans. Robotics*, vol. 22, no. 5, pp. 1017–1021, Oct. 2006.

[11] 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.

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

[13] R. G. Roberts, R. J. Jamisola Jr., and A. A. Maciejewski, "Identifying the failure-tolerant workspace boundaries of a kinematically redundant manipulator," in *IEEE Int. Conf. Robot. Automat.*, Roma, Italy, April 10-14 2007, pp. 4517–4523.

[14] C. Lück, "Self-motion representation and global path planning optimization for redundant manipulators through topology based discretization," *J. Intel. Rob. Syst.*, vol. 19, no. 1, pp. 23–38, May 1997.

[15] R. G. Roberts, "The dexterity and singularities of an underactuated robot," *J. Rob. Syst.*, vol. 18, no. 4, pp. 159–169, 2001.

[16] J. W. Burdick, "On the inverse kinematics of redundant manipulators: Characterization of the self-motion manifolds," in *IEEE Int. Conf. Robot. Automat.*, Scottsdale, AZ, May 14-19 1989, pp. 264–270.

[17] 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.

[18] J. Lenarčič, "Some considerations on the self motion curves of a planar 3R manipulator," *J. Computer Inform. Tech.*, vol. 10, no. 2, pp. 125–131, 2002.

[19] 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.

[20] C. Gosselin, "Determination of the workspace of 6-DOF parallel manipulators," *ASME J. Mech. Design*, vol. 112, no. 3, pp. 331–336, 1990.