DISSERTATION

DESIGN AND CONTROL OF KINEMATICALLY REDUNDANT ROBOTS FOR MAXIMIZING FAILURE-TOLERANT WORKSPACES

Submitted by

Ashraf M. Bader

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Spring 2021

Doctoral Committee:

Advisor: Anthony A. Maciejewski

Juliana Oprea
Ali Pezeshki
Peter Young

ABSTRACT

DESIGN AND CONTROL OF KINEMATICALLY REDUNDANT ROBOTS FOR MAXIMIZING FAILURE-TOLERANT WORKSPACES

Kinematically redundant robots have extra degrees of freedom so that they can tolerate a joint failure and still complete an assigned task. Previous work has defined the "failure-tolerant workspace" as the workspace that is guaranteed to be reachable both before and after an arbitrary locked-joint failure. One mechanism for maximizing this workspace is to employ optimal artificial joint limits prior to a failure. This dissertation presents two techniques for determining these optimal artificial joint limits. The first technique is based on the gradient ascent method. The proposed technique is able to deal with the discontinuities of the gradient that are due to changes in the boundaries of the failure tolerant workspace. This technique is illustrated using two examples of three degree-of-freedom planar serial robots. The first example is an equal link length robot where the optimal artificial joint limits are computed exactly. In the second example, both the link lengths and artificial joint limits are determined, resulting in a robot design that has more than twice the failure-tolerant area of previously published locally optimal designs. The second technique presented in this dissertation is a novel hybrid technique for estimating the failure-tolerant workspace size for robots of arbitrary kinematic structure and any number of degrees of freedom performing tasks in a 6D workspace. The method presented combines an algorithm for computing self-motion manifold ranges to estimate workspace envelopes and Monte-Carlo integration to estimate orientation volumes to create a computationally efficient algorithm. This algorithm is then combined with the coordinate ascent optimization technique to determine optimal artificial joint limits that maximize the size of the failure-tolerant workspace of a given robot. This approach is illustrated on multiple examples of robots that perform tasks in 3D planar and 6D spatial workspaces.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, Prof. Anthony Maciejewski, for his guidance, encouragement, and advice. This work would not be possible without his continuous guidance and support.

I am also grateful to my committee members Prof. Juliana Oprea, Prof. Ali Pezeshki, and Prof. Peter Young for their extended discussions and valuable suggestions which have contributed greatly to the improvement of the dissertation.

To my family, my deepest gratitude to my parents for their continuous support that helped me to overcome all difficulties. Finally, special thanks to Malak, my wonderful wife, for her patience and great encouragement in this journey and for giving me all the support I needed. My lovely children, Salma, Badreddin, and Sama, have been the relief of my work burden and the joy of my life.

DEDICATION

I would like to dedicate this thesis to my family.

TABLE OF CONTENTS

[1]This chapter has been published in [1]

[2]This chapter has been published in [2]

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Failure-tolerant robots are frequently used for applications in hazardous and remote environments where performing routine maintenance and repair are not possible. Example applications include handling radioactive materials [3, 4], inspecting nuclear reactors [5, 6], gathering samples near potentially active volcanoes [7], and exploring space [8, 9] or the deep sea [10]. There have been studies on assessing a robot's reliability using fault trees [11, 12]. Other studies, focused on enhancing a robot's tolerance to failure, include faults diagnosis [13, 14], detection and isolation [8, 15], and identification [16], e.g., identifying the locked position of the failed joint [17]. Most previous studies have assumed a locked-joint failure model, where the joint is locked due to the failure itself or because brakes are employed when some other failure occurs, e.g., for a free-swinging joint failure where the actuator torque is lost [18, 19].

Because kinematically redundant robots have the ability to perform their task even if one or more joints fail due to the extra degree(s) of freedom, many failure-tolerance techniques have been developed for optimizing the local and/or global properties of a redundant robot after a locked joint failure. The local properties are frequently quantified by the singular values of the redundant robot's Jacobian [20, 21]. Optimizing these measures can be used to design failure-tolerant robots [20–22] and determine optimal configurations during their control [23]. Global measures of fault tolerance refer to the robot's workspace, e.g., where the reachability of critical task locations can be guaranteed because they have large self-motion manifolds [24] or identifying an entire region that is reachable before and after joint failure, referred to as the failure-tolerant workspace ($\boldsymbol{\mathcal{W}_F}$) [25, 26].

One approach to guarantee the existence of a $\boldsymbol{\mathcal{W}_F}$ is increasing the degree of redundancy (DOR). Unfortunately, it has been shown that for every single locked-joint failure, at least two DORs are required to guarantee the existence of a $\boldsymbol{\mathcal{W}_F}$ [27], if one does not restrict the configurations of the robot prior to a failure. For manipulators with only one DOR, one way to guarantee the

existence of a $\mathcal{W}_F$ is applying the method of artificial joint limits [28]. This method guarantees a $\mathcal{W}_F$ by judiciously using artificial joint limits that are applied prior to a failure and released after a locked-joint failure occurs. To evaluate the workspace size before and after a joint failure, a number of methods have been developed to determine the workspace boundaries [29] and compute the workspace size by generating symbolic equations of these boundaries [26] or estimating the workspace sizes, e.g., using Monte-Carlo integration [22, 30–32]. The optimization of $\mathcal{W}_F$, for planar 3R and 4R robots, has been discussed for one and two arbitrary locked joint failures, respectively [33].

This dissertation presents a technique for maximizing $\mathcal{W}_F$ after an arbitrary single locked-joint failure for arbitrary planar robots. In this work, the optimal artificial joint limits are determined by using the gradient ascent method on the symbolic area equations derived from applying Green's theorem using the $\mathcal{W}_F$ boundaries. One contribution of this work is resolving the discontinuity in the gradients that is caused when $\mathcal{W}_F$ boundary curves change. The proposed approach is able to duplicate previous results to a higher precision and determine an entirely new $\mathcal{W}_F$ with significantly higher area.

Computing $\mathcal{W}_F$ for high-dimensional workspaces is problematic because voxelization methods suffer from exponential growth in the dimension of the workspace and there is no known way to symbolically express the boundaries of such workspaces. The second technique presented in this dissertation addresses these limitations by developing a hybrid approach that uses a combination of discretization in two dimensions, Monte-Carlo integration for estimating orientation volumes, and an efficient numerical procedure for computing workspace envelopes. This method is applicable to fully general six-dimensional workspaces for robots with an arbitrary number of degrees of freedom (DOFs). This approach for efficiently computing $\mathcal{W}_F$ is then used to determine optimal artificial joint limits that maximize $\mathcal{W}_F$ by applying a coordinate ascent method.

This dissertation is organized as follows. In chapter 2, maximizing the failure-tolerant workspace area for planar redundant robots is discussed [1]. This chapter presents a gradient-based method to maximize the failure-tolerant workspace area along with a technique for resolving issues caused by

changes in the workspaces boundary curves. The proposed technique is illustrated using two examples. Chapter 3 presents a novel hybrid technique for estimating the failure-tolerant workspace size [2]. The proposed technique is applicable for any serial robot with arbitrary kinematics where the robot's joints are either revolute or prismatic. The method presented combines an algorithm for computing self-motion manifold ranges to estimate workspace envelopes and Monte-Carlo integration to estimate orientation volumes to create a computationally efficient algorithm. This algorithm is then combined with the coordinate ascent optimization technique to determine optimal artificial joint limits that maximize the size of the failure-tolerant workspace of a given robot. This approach is illustrated on multiple examples of robots that perform tasks in 3D planar and 6D spatial workspaces. Finally, the conclusions of this dissertation are presented in chapter 4.

Chapter 2

Maximizing The Failure-Tolerant Workspace Area For Planar Redundant Robots[3]

2.1 Introduction

Kinematically redundant robots have more than the minimum number of joints required to execute a specific task. These robots have the ability to tolerate joint failures and still complete the assigned task. Robots with this ability are useful when deployed for tasks in hazardous and/or remote environments, for example, when inspecting nuclear reactors [34], gathering samples near active volcanoes [7], or exploring space [9] or the deep sea [10].

Because of the importance of these applications, many different aspects of failure tolerance for robots have been studied. For example, fault trees have been used to assess their reliability [11, 12]. Other work has focused on fault diagnosis [13, 14], detection and isolation [15], identification [16, 17], and control strategies for recovery from failures [35]. The most common failure mode assumed by the majority of previous work is that of a locked-joint failure, due to the failure itself or to brakes being employed [18, 19, 36].

Many fault-tolerance techniques have been developed for optimizing the local and/or global properties of a redundant robot after a locked joint failure. The local properties are frequently quantified by the singular values of the post-failure Jacobian to determine measures of local fault tolerance [37]. Optimization of these measures can be used to design the kinematics of fault-tolerant robots [20–22] and to determine optimal configurations during their control [23]. In contrast, global measures focus more on the effect of failures on a robot's workspace. For example, focusing on completion of a point-to-point task [38, 39] or identifying the boundaries of its failure-

[3]This chapter has been published in [1]

tolerant workspace ($\mathcal{W}_F$), i.e., the workspace that is reachable after any arbitrary locked-joint failure [26].

One approach to guarantee the existence of $\mathcal{W}_F$ is increasing the degree of redundancy (DOR). Unfortunately, it has been shown that for every single locked-joint failure, at least two DORs are required to guarantee the existence of $\mathcal{W}_F$ [27], if one does not restrict the configurations of the robot prior to a failure. However, by judiciously applying artificial joint limits that are released after a failure, one can guarantee the existence of $\mathcal{W}_F$ with only a single DOR [28]. Previous researchers have computed the area of $\mathcal{W}_F$ using Monte-Carlo integration [27] or Green's theorem [26], and tried to maximize its size for planar 3R and 4R robots [33]. Our work and [33] have the same goal of maximizing the area of $\mathcal{W}_F$, however, they differ in two fundamental ways. First, by symbolically computing the gradient we are able to guarantee an accurate locally optimal solution. Second, by properly dealing with the discontinuities of the gradients, instead of applying restrictive assumptions, we are able to identify significantly better solutions.

This current work presents a technique for exactly determining the artificial joint limits that maximize $\mathcal{W}_F$ using the gradient ascent method on the area equations determined from Green's theorem. Our contributions include identifying and resolving the discontinuous change in the gradient of the area due to changes in the boundary equations of $\mathcal{W}_F$. We apply this computationally efficient technique to exactly determine the maximum area of $\mathcal{W}_F$ for planar 3R serial robots of arbitrary link lengths. In the next section, we review the mathematical definition of $\mathcal{W}_F$ and summarize how its boundaries are identified. In Section 2.3, we formally define our optimization problem and then describe our method for solving it in Section 2.4. We then illustrate our technique on two examples of planar 3R robots in Section 2.5. Finally, we present our conclusions in Section 2.6.

2.2 Background on Computing Failure- Tolerant Workspace

2.2.1 Overview

In this section, the definition of a failure-tolerant workspace and the identification of its boundaries are summarized as previously discussed in [26]. In that work, artificial joint limits were imposed prior to a failure in order to increase the size of the post-failure workspace. After a locked-joint failure, the artificial limits on the remaining working joints are released. Even though the artificial limits decrease the pre-failure workspace, if chosen appropriately, a post-failure workspace can be guaranteed. In this work we show how to compute the optimal artificial joint limits that maximize the size of the workspace reachable both before and after a failure.

2.2.2 Definition of Failure-Tolerant Workspace

The forward kinematic function, $\mathbf{x} = \mathbf{f}(\mathbf{q})$, maps a robot configuration $\mathbf{q}$ in the joint space $\mathcal{C}$, i.e., $\mathbf{q} \in \boldsymbol{\mathcal{C}} \subset \mathbb{R}^n$, to a location $\mathbf{x}$ in the workspace $\mathcal{W}$, i.e., $\mathbf{x} \in \boldsymbol{\mathcal{W}} \subset \mathbb{R}^m$ where n is the number of joints and m is the dimension of workspace. For kinematically redundant robots, $n > m$ and $n - m$ is the DOR. In this work, it is assumed that the robots have one DOR, i.e., $n - m = 1$.

Prior to a failure, each joint i, denoted q_i, is bounded by lower $\underline{a}_i$ and upper $\overline{a}_i$ artificial limits where $\overline{a}_i \geq \underline{a}_i$ and $\overline{a}_i, \underline{a}_i \in [-\pi, \pi]$. If we denote the range of q_i when bounded by the artificial joint limits as $A_i = [\underline{a}_i, \overline{a}_i]$ then the pre-failure configuration space is $\boldsymbol{\mathcal{C}_A} = A_1 \times \cdots \times A_n$. Therefore, the pre-failure workspace denoted $\boldsymbol{\mathcal{W}_0}$ is given by

$$\boldsymbol{\mathcal{W}_0} = \mathbf{f}(\boldsymbol{\mathcal{C}_A}) = \{\mathbf{x} = \mathbf{f}(\mathbf{q}) \mid \mathbf{q} \in \boldsymbol{\mathcal{C}_A}\}. \tag{2.1}$$

After joint i fails and is locked at $q_i = \theta_i$ where $\underline{a}_i \leq \theta_i \leq \overline{a}_i$, the artificial limits are released on the remaining joints. In this work, without loss of generality, we assume that the physical limits on all joints are $-\pi$ to π. This yields a reduced configuration space that is given by

$${}^i\boldsymbol{\mathcal{C}}(\theta_i) = \{\mathbf{q} \in \boldsymbol{\mathcal{C}} \mid q_i = \theta_i\}. \tag{2.2}$$

Therefore, the post-failure workspace $\boldsymbol{\mathcal{W}_i}$, which is the guaranteed workspace after a failure in joint i between the artificial limits, i.e., $\underline{a}_i \leq \theta_i \leq \overline{a}_i$, is given by

$$\boldsymbol{\mathcal{W}_i} = \bigcap_{\underline{a}_i \leq \theta_i \leq \overline{a}_i} \mathbf{f}({}^i\boldsymbol{\mathcal{C}}(\theta_i)). \tag{2.3}$$

Finally, the failure-tolerant workspace $\boldsymbol{\mathcal{W}_F}$, which is the workspace that is guaranteed to be reachable both before and after an arbitrary single locked-joint failure, is given by

$$\boldsymbol{\mathcal{W}_F} = \bigcap_{i \in \mathbf{F} \cup \{0\}} \boldsymbol{\mathcal{W}_i} \tag{2.4}$$

where $\mathbf{F} \subset \{1, 2, \ldots, n\}$ is the set of all joint indices for joints that are likely to fail.

2.2.3 Identification of Failure-Tolerant Workspace Boundaries

The authors in [26] have identified two conditions to determine if a workspace location $\mathbf{x}$ belongs to $\boldsymbol{\mathcal{W}_F}$. Both conditions are based on the pre-image of $\mathbf{x}$, denoted $\mathbf{f}^{-1}(\mathbf{x})$, where

$$\mathbf{f}^{-1}(\mathbf{x}) = \{\mathbf{q} \in \boldsymbol{\mathcal{C}} \mid \mathbf{f}(\mathbf{q}) = \mathbf{x}\}, \tag{2.5}$$

i.e., the set of configurations that correspond to $\mathbf{x}$. Condition 1 is that $\mathbf{x}$ be reachable prior to a failure, i.e., $\mathbf{x} \in \boldsymbol{\mathcal{W}_0}$, so that

$$\boldsymbol{\mathcal{C}}_A \cap \mathbf{f}^{-1}(\mathbf{x}) \neq \emptyset. \tag{2.6}$$

Condition 2 is that $\mathbf{x}$ is reachable after a failure, i.e., $\mathbf{x} \in \boldsymbol{\mathcal{W}_i}$, so that

$$A_i \subset P_i[\mathbf{f}^{-1}(\mathbf{x})] \text{ for } i \in \mathbf{F} \tag{2.7}$$

where P_i is the projection onto the i-th joint axis, i.e., the range of q_i for all $\mathbf{q}$ that satisfy $\mathbf{x} = \mathbf{f}(\mathbf{q})$. Condition 2 states that when joint i is locked at any angle within A_i, the end effector can still reach

$\mathbf{x}$ because that angle is contained in the i-th component of $\mathbf{f}^{-1}(\mathbf{x})$. If both conditions are satisfied for $\mathbf{x}$, then $\mathbf{x} \in \boldsymbol{\mathcal{W}_F}$.

Fortunately, it is relatively straightforward to identify sets of equations that can determine all potential boundaries of $\boldsymbol{\mathcal{W}_F}$. The potential boundaries of $\boldsymbol{\mathcal{W}_0}$ are located at $\mathbf{f}(\mathbf{q})$ where the configuration $\mathbf{q} \in \boldsymbol{\mathcal{C}_A}$ is a kinematic singularity or when one or more joints are at an artificial joint limit, i.e., $q_i = \underline{a}_i$ or $q_i = \overline{a}_i$. The potential boundaries of $\boldsymbol{\mathcal{W}_i}$ are the workspace locations where the end effector is on the verge of violating condition 2. This can occur in two ways, i.e., either the projection of the pre-image for this workspace location becomes disjoint within A_i or it fails to contain an endpoint of A_i. The first way will occur at $\mathbf{f}(\mathbf{q})$ where $\mathbf{q} \in {}^i\boldsymbol{\mathcal{C}}(\theta_i)$ is a kinematic singularity. The second situation can be identified by computing the null vector associated with the robot's Jacobian. In particular, let $\mathbf{n}(\mathbf{q})$ represent the null vector of the robot at configuration $\mathbf{q}$ and $\mathbf{n}_i$ be the i-th element of $\mathbf{n}$. Then the potential boundaries of $\boldsymbol{\mathcal{W}_i}$ occur when $n_i = 0$ and $q_i = \underline{a}_i$ or $\overline{a}_i$ [26].

Once all potential boundaries of $\boldsymbol{\mathcal{W}_0}$ and all $\boldsymbol{\mathcal{W}_i}$'s are determined, one can use the two conditions to identify the actual boundaries of $\boldsymbol{\mathcal{W}_F}$. To do this, one must first divide up all boundaries into simple non-intersecting pieces. For each separated non-intersecting potential boundary, two workspace locations on opposite sides of the boundary are evaluated. If they are both reachable before and after a failure, then this is not an actual boundary of $\boldsymbol{\mathcal{W}_F}$. After the actual boundaries of $\boldsymbol{\mathcal{W}_F}$ are identified, these boundaries can be used to write an equation for the size of the failure tolerant workspace by using the Gauss-Ostrogradsky's theorem. In this work, we illustrate this on maximizing the area of $\boldsymbol{\mathcal{W}_F}$ for a planar manipulator, denoted $\mathcal{A}_{\mathcal{W}_F}$, by using Green's theorem.

2.3 Problem Statement

The problem solved in this work is maximizing the area of the failure-tolerant workspace $\mathcal{A}_{\mathcal{W}_F}$ for any planar 3R robot with joint variables denoted by θ. All possible 3R robots can be defined by their link lengths, l_1, l_2, and l_3. However, the link lengths are not independent because having the same ratio of link lengths results in a geometrically similar, i.e., scaled version, robot. Therefore,

we normalize all robots to be of total length 3 m, i.e., we add the constraint that $l_1 + l_2 + l_3 = 3$ m. In addition to the link lengths design parameter one can also select the artificial joint limits $\overline{a}_i$ and $\underline{a}_i$ that are used while the robot is in operation. However, $\underline{a}_1$ and $\overline{a}_1$ are not independent, because $\mathcal{A}_{\mathcal{W}_F}$ is rotationally invariant to q_1, i.e., only the difference between $\underline{a}_1$ and $\overline{a}_1$ affects its value. Therefore, we add the constraint that $\underline{a}_1 = -\overline{a}_1$ in the optimization.

If one denotes $\boldsymbol{L} = \{l_1, l_2, l_3\}$ and $\boldsymbol{A} = \{\underline{a}_1, \overline{a}_1, \underline{a}_2, \overline{a}_2, \underline{a}_3, \overline{a}_3\}$, then the optimization problem for designing a 3R robot with a maximum failure tolerant area can be formulated as:

$$\begin{aligned} \underset{\boldsymbol{L},\boldsymbol{A}}{\text{maximize}} \quad & \mathcal{A}_{\mathcal{W}_F} \\ \text{subject to} \quad & l_1 + l_2 + l_3 = 3 \\ & \underline{a}_1 = -\overline{a}_1. \end{aligned}$$

For the case where one is given a specific 3R robot, the above optimization problem would be solved for $\boldsymbol{A}$ as the only decision variable. The above optimization can be problematic because the gradient of $\mathcal{A}_{\mathcal{W}_F}$ with respect to $\boldsymbol{L}$ and $\boldsymbol{A}$ is discontinuous. Our proposed approach to computing this optimization is discussed in the next section.

2.4 Technique

2.4.1 Overview

In this section, we use the gradient ascent method [40] to maximize $\mathcal{A}_{\mathcal{W}_F}$. To do this, we first compute $\mathcal{A}_{\mathcal{W}_F}$ for a given $\boldsymbol{L}$ and $\boldsymbol{A}$ by applying Green's theorem on the boundary curves of $\boldsymbol{\mathcal{W}_F}$. The technique for identifying the boundary curves of $\boldsymbol{\mathcal{W}_F}$ was summarized in subsection 2.2.3. Then a step along the gradient of $\mathcal{A}_{\mathcal{W}_F}$ with respect to $\boldsymbol{L}$ and $\boldsymbol{A}$ is used to determine a new $\boldsymbol{L}$ and $\boldsymbol{A}$ and the process is repeated until the local maximum is achieved. However, there are two issues that one must consider when computing the gradients. First, there will be a change in the number of boundary curves. Second, and more serious, is when the gradients become discontinuous. In

both cases it is important to consider the critical points where these changes occur. These issues will be discussed in the following subsections.

2.4.2 Green's Theorem

The equation for the area of $\boldsymbol{\mathcal{W}_F}$, i.e., $\mathcal{A}_{\mathcal{W}_F}$, is determined by applying Green's theorem to the boundary curves of $\boldsymbol{\mathcal{W}_F}$. The area of a region D, $D \subset \mathbb{R}^2$, bounded by a closed curve C, denoted $g_{\mathcal{A}}(D)$, can be computed using

$$g_{\mathcal{A}}(D) = \int\int_D d\mathcal{A} = \frac{1}{2}\int_C (x_1\, dx_2 - x_2\, dx_1)\,. \tag{2.8}$$

To compute $\mathcal{A}_{\mathcal{W}_F}$, let curve j be a boundary curve of $\boldsymbol{\mathcal{W}_F}$ and ${}^j\mathbf{x} = [x_1, x_2]$ be its workspace coordinates, i.e.,

$${}^j\mathbf{x} = \mathbf{f}({}^j\boldsymbol{\theta}), \tag{2.9}$$

where $\mathbf{f} = [f_1({}^j\theta_i),\ f_2({}^j\theta_i)]$ is the forward kinematic function and ${}^j\theta_i$ is the variable joint i in the curve j.

Let ${}^j\boldsymbol{\theta}$ be the vector that includes the variable ${}^j\theta_i$ and where the ${}^j\theta_k$'s, $k \neq i$, are specified. The center of curve j, denoted ${}^jC = ({}^jc_{x_1}, {}^jc_{x_2})$, is the workspace location of joint axis i and the radius, denoted ${}^jR$, is the distance from joint axis i to the end effector. The end points of curve j are identified by the intersection of curve j with the two adjoining boundary curves, denoted ${}^j\underline{\mathbf{x}}$ and ${}^j\overline{\mathbf{x}}$. The notation ${}^j\underline{\theta}_i$ and ${}^j\overline{\theta}_i$ will be used for the values of ${}^j\theta_i$ at these endpoints, i.e., ${}^j\underline{\theta}_i \leq {}^j\theta_i \leq {}^j\overline{\theta}_i$,

$$\begin{aligned} {}^j\underline{\boldsymbol{\theta}} &= \mathbf{f}^{-1}({}^j\underline{\mathbf{x}}) \\ {}^j\overline{\boldsymbol{\theta}} &= \mathbf{f}^{-1}({}^j\overline{\mathbf{x}}) \end{aligned} \tag{2.10}$$

with ${}^j\boldsymbol{\theta}_k$ where $k \neq i$ are specified.

Once ${}^j\underline{\theta}_i$ and ${}^j\overline{\theta}_i$ are determined, then the area under the curve j is

$$\mathcal{A}_j = \frac{1}{2}\int_{{}^j\underline{\theta}_i}^{{}^j\overline{\theta}_i} \left(\left[f_1({}^j\theta_i)\frac{df_2({}^j\theta_i)}{d{}^j\theta_i} - f_2({}^j\theta_i)\frac{df_1({}^j\theta_i)}{d{}^j\theta_i} \right] d{}^j\theta_i \right). \tag{2.11}$$

The failure-tolerant workspace area is the summation of the areas under the boundary curves, i.e.,

$$\mathcal{A}_{\mathcal{W}_F} = \sum_{j=1}^{N} \mathcal{A}_j \tag{2.12}$$

where the number of boundary curves N varies.

2.4.3 Gradient Method

Once $\mathcal{A}_{\mathcal{W}_F}(\boldsymbol{L}, \boldsymbol{A})$ is computed for an initial $(\boldsymbol{L}, \boldsymbol{A})^0$ using (2.12), the gradient ascent method is used to compute the next value of $(\boldsymbol{L}, \boldsymbol{A})^{(k+1)}$ to increase the area using

$$(\boldsymbol{L}, \boldsymbol{A})^{(k+1)} = (\boldsymbol{L}, \boldsymbol{A})^{(k)} + \alpha \nabla \mathcal{A}_{\mathcal{W}_F}((\boldsymbol{L}, \boldsymbol{A})^{(k)}) \tag{2.13}$$

where $\nabla \mathcal{A}_{\mathcal{W}_F}$ is the gradient of $\mathcal{A}_{\mathcal{W}_F}$, i.e.,

$$\nabla \mathcal{A}_{\mathcal{W}_F}(\boldsymbol{L}, \boldsymbol{A}) = \left(\frac{\partial \mathcal{A}_{\mathcal{W}_F}(\boldsymbol{L}, \boldsymbol{A})}{\partial \boldsymbol{L}}, \frac{\partial \mathcal{A}_{\mathcal{W}_F}(\boldsymbol{L}, \boldsymbol{A})}{\partial \boldsymbol{A}} \right) \tag{2.14}$$

and α is a positive step size. A constant value of α, i.e., $\alpha = 0.02$, is used until a step results in a decrease in the computed area, $\mathcal{A}_{\mathcal{W}_F}$. This will be due to one of two causes, i.e., either the gradient is near zero or a change in the gradient equation has occurred due to changes in both N and the boundary curves that compose $\boldsymbol{\mathcal{W}_F}$. At this point a binary chop on α is performed to identify the exact location of the gradient being zero or discontinuous.

We illustrate these issues with a simple example for a planar 3R robot with equal link lengths, i.e., $\boldsymbol{L} = \{1, 1, 1\}$, and with artificial joint limits of $\boldsymbol{A} = \{ -30°, 30°, -120°, 60°, -130°, 130°\}$. The boundary curves and two additional near boundary curves of $\boldsymbol{\mathcal{W}_F}$ for this value of $\boldsymbol{A}$ are shown in Fig. 2.1 with the joint values, i.e., ${}^j\boldsymbol{\theta}$, for each curve given in Table 2.1. The dashed

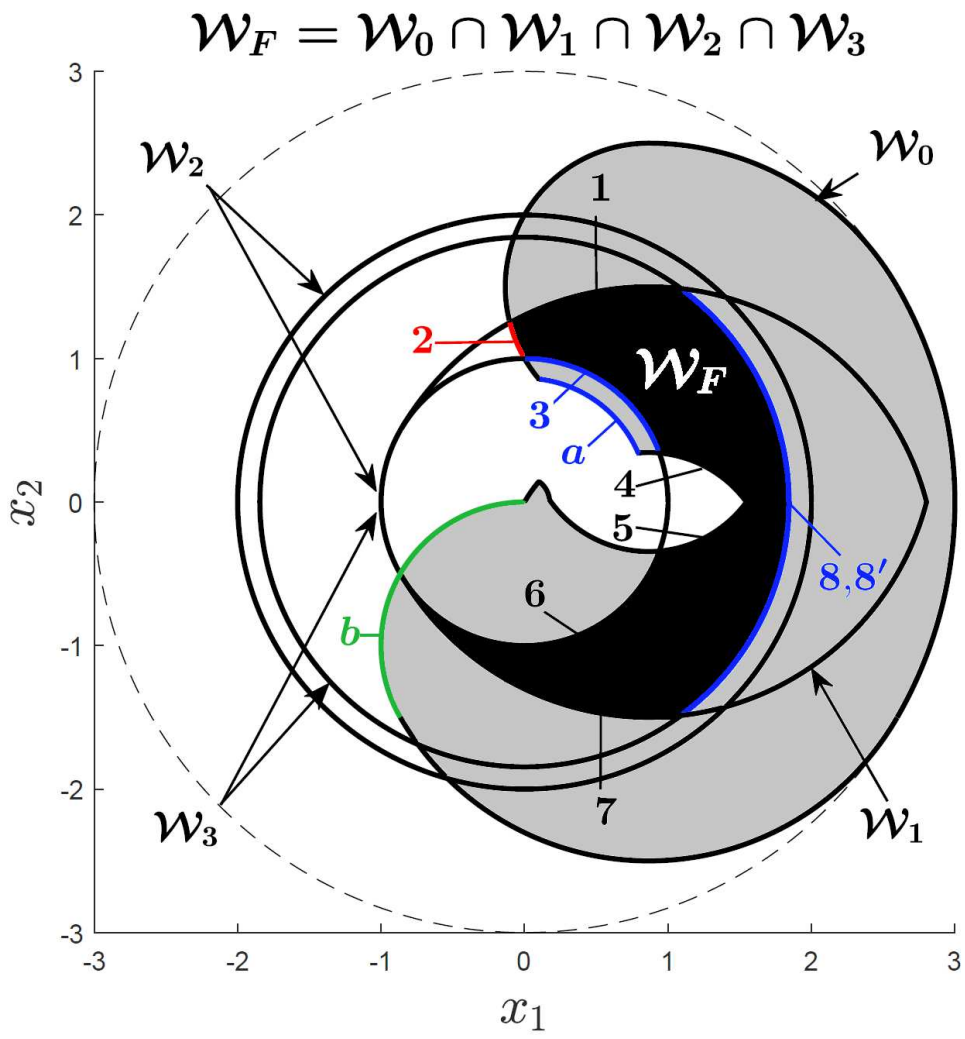


Figure 2.1: A simple example that illustrates some of the issues with computing the gradient of the area of the guaranteed failure-tolerant workspace. Depending on how A changes, boundary curve 2, shown in red, may disappear. Likewise, curve b, shown in green, may appear. Also, the boundary curve 3, shown in blue, could be replaced by curve a. This particular value of A also results in identical boundary curves, i.e., 8 and $8'$.

circle in the figure has a radius of 3 m, i.e., the maximum reach of the robot when there are no constraints on the joint angles.

There are two issues associated with maximizing $\mathcal{A}_{\mathcal{W}_F}$ that need to be considered:

Issue 1: Changes in the number of boundary curves N.

While the size of $\boldsymbol{\mathcal{W}_F}$ is increasing, the value of N may change due to either the disappearance of existing boundary curves or the generation of new ones. As the values of $\boldsymbol{A}$ are changed to increase $\mathcal{A}_{\mathcal{W}_F}$, the boundary curves will change, for example, if $\overline{a}_1$ and $\overline{a}_2$ are increased then curve

Table 2.1: Boundary Curves of $\boldsymbol{\mathcal{W}_F}$ for Fig. 2.1

Curve	θ_1	θ_2	θ_3	Workspace
1	$\underline{a}_1$	[1.98, 2.59]	0	$\boldsymbol{\mathcal{W}_1}$
2	$\overline{a}_1$	$\overline{a}_2$	[1.82, 2.09]	$\boldsymbol{\mathcal{W}_0}$
3	[0.35, π/2]	0 (π)	π (0)	$\boldsymbol{\mathcal{W}_2}$ ($\boldsymbol{\mathcal{W}_3}$)
4	$\underline{a}_1$	[0.02, 0.87]	$\overline{a}_3$	$\boldsymbol{\mathcal{W}_0}$
5	$\overline{a}_1$	[-0.87, -0.02]	$\underline{a}_3$	$\boldsymbol{\mathcal{W}_0}$
6	[-2.61, -0.35]	0 (π)	π (0)	$\boldsymbol{\mathcal{W}_2}$ ($\boldsymbol{\mathcal{W}_3}$)
7	$\overline{a}_1$	[-π, -1.98]	0	$\boldsymbol{\mathcal{W}_1}$
8	[-0.93, 0.93]	$\frac{-\underline{a}_3}{2}$	$\underline{a}_3$	$\boldsymbol{\mathcal{W}_3}$
$8'$	[-0.93, 0.93]	$\frac{-\overline{a}_3}{2}$	$\overline{a}_3$	$\boldsymbol{\mathcal{W}_3}$
a	[-0.52, 0.52]	$\overline{a}_2$	$\overline{a}_3$	$\boldsymbol{\mathcal{W}_0}$
b	$\underline{a}_1$	$\underline{a}_2$	[-2.09 , 0]	$\boldsymbol{\mathcal{W}_0}$

2 will disappear because curves 1 and 3 will intersect before reaching curve 2. The point at which this occurs can be easily identified due to ${}^2\underline{\mathbf{x}} = {}^2\overline{\mathbf{x}}$ and the summation in (2.12) needs to be updated appropriately. Likewise, if one increases the $\underline{a}_1$ and $\underline{a}_2$ values then the new boundary curve b will appear between curves 6 and 7. Once again, the summation in (2.12) needs to be updated appropriately. Fortunately, in both of these cases of disappearing and appearing boundary curves, the gradients are continuous. However, there are cases where boundary curves can overlap and switch back and forth, which results in a discontinuous gradient. This is discussed next.

Issue 2: Switching between identical boundary curves.

For this simple example, curve 6 consists of two identical boundary curves due to the specific value of $\boldsymbol{L}$. This is also true for curve 3. If one is only interested in optimizing $\mathcal{A}_{\mathcal{W}_F}$ by adjusting the artificial joint limits, then $\boldsymbol{L}$ will stay constant and the two boundary curves will stay identical. This is not true for other identical boundary curves. For example, as the area was increased, curve a changed continuously from being a boundary curve, to being identical to curve 3, and then no longer being a boundary. For cases like this, the gradient does change discontinuously, due to the different equations for the different boundary curves, however, the gradients for the boundary curves are both still in the same direction, so this does not cause a problem. Unfortunately, this is not always the case.

Consider the case of curves 8 and $8'$. The gradient ascent algorithm will cause these curves to switch back and forth as the boundary curve, because their area equations result in gradients that oppose each other. This occurs whenever $\overline{a}_3 \neq -\underline{a}_3$, which numerically will always be true. To deal with this situation, one must enforce the constraint that $\overline{a}_3 = -\underline{a}_3$ so that the curves remain identical. Doing so makes the area equations, and gradients, for curve 8 and $8'$ identical. This greatly improves the convergence of the gradient ascent algorithm. It should be noted that there can be switching between multiple boundary curves, as will be illustrated in section 2.5.

2.5 Examples of Maximizing Failure-Tolerant Workspace

2.5.1 Overview

This section presents two illustrative examples that use the technique presented above. Example 1 represents the case where one has a robot with a given set of link lengths and one would like to know the optimal values for the artificial joint limits that maximize the failure-tolerant workspace. Without loss of generality, we select a robot with link lengths $\boldsymbol{L} = \{1,\ 1,\ 1\}$ to illustrate this case. Example 2 represents the case where one is designing a robot and so can select both the link lengths and the artificial joint limits. For example 2, we show what happens when the gradient ascent optimization to maximize the failure-tolerant workspace area is performed from different starting designs. We first start the optimization with the final values from example 1, and show that it converges to a locally optimal design. We then perform the optimization from multiple random starting designs and show the designs with a significantly larger failure-tolerant workspace area.

In both examples, the first step is computing the failure-tolerant area, $\mathcal{A}_{\mathcal{W}_F}$, which requires identifying the boundary curves. We briefly review the algorithm for computing these boundaries for a 3R planar robot, with the complete details in [23]. The forward kinematics for planar 3R manipulators are given by

$$\mathbf{f}(\boldsymbol{\theta}) = \begin{bmatrix} l_1 \cos(\theta_1) + l_2 \cos(\theta_{12}) + l_3 \cos(\theta_{123}) \\ l_1 \sin(\theta_1) + l_2 \sin(\theta_{12}) + l_3 \sin(\theta_{123}) \end{bmatrix} \tag{2.15}$$

where $\theta_{12} = \theta_1 + \theta_2$, and $\theta_{123} = \theta_1 + \theta_2 + \theta_3$ and the associated null vector is given by

$$\mathbf{n} = \begin{bmatrix} l_2 l_3 \sin(\theta_3) \\ -l_2 l_3 \sin(\theta_3) - l_1 l_3 \sin(\theta_{23}) \\ l_1 l_2 \sin(\theta_2) + l_1 l_3 \sin(\theta_{23}) \end{bmatrix}. \tag{2.16}$$

The boundary curves for $\boldsymbol{\mathcal{W}_0}$ occur at the kinematic singularities and the joint limit singularities. The curves associated with the kinematic singularities are given by the values of $\mathbf{f}(\boldsymbol{\theta})$ for all $\boldsymbol{\theta}$ where $\mathbf{n} = \mathbf{0}$. The curves associated with the joint limit singularities are given by the values of $\mathbf{f}(\boldsymbol{\theta})$ for the values of θ that satisfy certain conditions when joints are at their limits. If a single joint i is at its limit, then the condition is given by

$$n_i \;=\; 0 \;\text{ and }\; \theta_i \;=\; \underline{a}_i \text{ or } \overline{a}_i.$$

If two joints, i and j, are at their limits then the conditions are given by: n_i and n_j are opposite in sign and

$$\begin{aligned} \theta_i \;&=\; \underline{a}_i \text{ and } \theta_j \;=\; \underline{a}_j \text{ or} \\ \theta_i \;&=\; \overline{a}_i \text{ and } \theta_j \;=\; \overline{a}_j, \end{aligned}$$

or n_i and n_j are of the same sign and

$$\begin{aligned} \theta_i \;&=\; \underline{a}_i \text{ and } \theta_j \;=\; \overline{a}_j \text{ or} \\ \theta_i \;&=\; \overline{a}_i \text{ and } \theta_j \;=\; \underline{a}_j, \end{aligned}$$

Table 2.2: Boundary Curves of $\boldsymbol{\mathcal{W}_F}$ for Fig. 2.2(a)

Curve	θ_1	θ_2	θ_3	Workspace
1	$\underline{a}_1$	$[{}^1\underline{\theta}_2, {}^1\overline{\theta}_2]$	0	$\boldsymbol{\mathcal{W}_1}$
2	$\overline{a}_1$	$\overline{a}_2$	$[{}^2\underline{\theta}_3, {}^2\overline{\theta}_3]$	$\boldsymbol{\mathcal{W}_0}$
3	$[{}^3\underline{\theta}_1, {}^3\overline{\theta}_1]$	π	0	$\boldsymbol{\mathcal{W}_3}$
4	$\underline{a}_1$	$[{}^4\underline{\theta}_2, {}^4\overline{\theta}_2]$	$\overline{a}_3$	$\boldsymbol{\mathcal{W}_0}$
5	$\underline{a}_1$	$\underline{a}_2$	$[{}^5\underline{\theta}_3, {}^5\overline{\theta}_3]$	$\boldsymbol{\mathcal{W}_0}$
6	$[{}^6\underline{\theta}_1, {}^6\overline{\theta}_1]$	$\frac{-\underline{a}_3}{2}$	$\underline{a}_3$	$\boldsymbol{\mathcal{W}_3}$

where $i \neq j$. The boundary curves for $\boldsymbol{\mathcal{W}_i}$, when there are no joint limits, are given by the kinematic singularities, i.e., the same equations as for $\boldsymbol{\mathcal{W}_0}$, and by the values of $\mathbf{f}(\boldsymbol{\theta})$ for all $\boldsymbol{\theta}$ where

$$n_i \;=\; 0 \text{ and } \theta_i \;=\; \underline{a}_i \text{ or } \overline{a}_i.$$

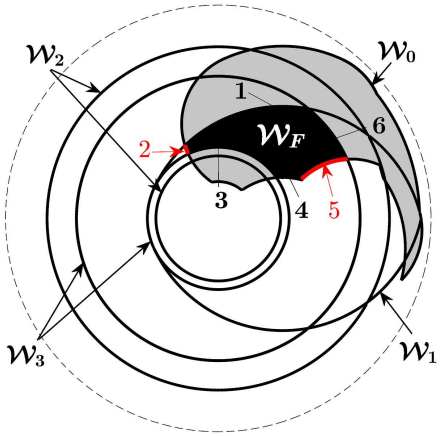
2.5.2 Example 1: Optimization of Artificial Joint Limits

For example 1, one is interested in determining the optimal artificial joint limits for a robot with link lengths $\boldsymbol{L} = \{1, 1, 1\}$. In Fig. 2.2 (a), we show an initial starting point for the gradient ascent algorithm where $\boldsymbol{A} = \{-25°, 25°, 40°, 90°, -60°, 120°\}$ and $N = 6$. These six boundary curves are given in Table 2.2 where each curve has one variable joint and the other two joints are specified. These curves are identified by the algorithm described in subsection 2.2.3.

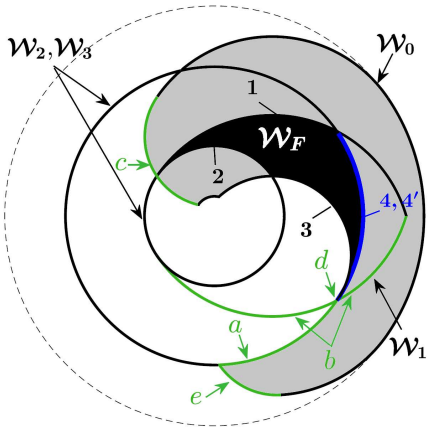
The first step is to determine the symbolic expression for $\mathcal{A}_{\mathcal{W}_F}$. This requires determining the area under each curve, $\mathcal{A}_j$. We illustrate this process using curve 1. To compute the area under curve 1, i.e., $\mathcal{A}_1$, we need to determine the symbolic equation for the center and radius of curve 1. The center of curve 1, i.e., ${}^1C = ({}^1c_{x_1}, {}^1c_{x_2})$ is the workspace location of its variable joint, i.e., joint 2, and is given by

$$({}^1c_{x_1}, {}^1c_{x_2}) = (l_1 \cos(\underline{a}_1), l_1 \sin(\underline{a}_1)) \tag{2.17}$$

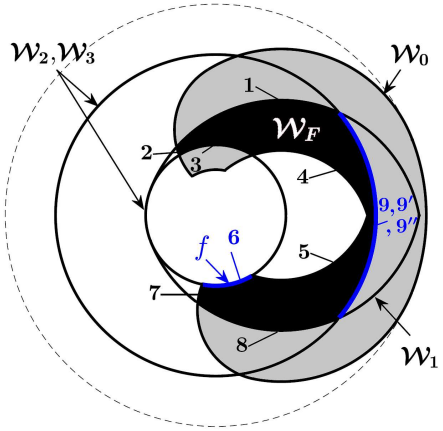
and the radius, i.e., ${}^1R$, is the distance from the joint 2 axis to the end effector, and is given by



$\mathcal{A}_{\mathcal{W}_F}$=1.3383 m^2 (a)



$\mathcal{A}_{\mathcal{W}_F}$=1.8197 m^2 (b)



$\mathcal{A}_{\mathcal{W}_F}$=3.3960 m^2 (c)

Figure 2.2: This figure depicts an example of optimizing $\mathcal{A}_{\mathcal{W}_F}$ for a robot with $\boldsymbol{L} = \{1, 1, 1\}$ that illustrates appearing, disappearing, and identical boundary curves. The gradient ascent algorithm starts with the initial artificial joint limits shown in (a) and as it progresses to (b), the red boundary curves disappear. As it continues on to (c) the green boundary curves appear. The most challenging part of the gradient ascent algorithm is when there are multiple identical boundary curves that have opposing gradients, which occurs frequently, i.e., the blue boundary curves in (b) and (c). If appropriate constraints are employed, the gradient ascent algorithm will converge to the optimal value of $\boldsymbol{A}$ as shown in Fig. 2.3.

$$ {}^1R = \sqrt{(l_2 + l_3\cos({}^1\theta_3))^2 + (l_3\sin({}^1\theta_3))^2}. \tag{2.18} $$

The center-radius form of the circle equation is then used to determine the intersection points of curve 1 with curves 6 and 2, i.e., ${}^1\underline{\mathbf{x}}$ and ${}^1\overline{\mathbf{x}}$, respectively. For example, ${}^1\underline{\mathbf{x}}$ is the intersection point of curve 1 with curve 6 and it can be determined by simultaneously solving the following two equations using the centers and radiuses of curves 1 and 6:

$$ \begin{aligned} {}^1R^2 = &\quad ({}^1\underline{\mathbf{x}}_1 - {}^1C_{x_1})^2 + ({}^1\underline{\mathbf{x}}_2 - {}^1C_{x_2})^2 \\ {}^6R^2 = &\quad ({}^1\underline{\mathbf{x}}_1 - {}^6C_{x_1})^2 + ({}^1\underline{\mathbf{x}}_2 - {}^6C_{x_2})^2. \end{aligned} \tag{2.19} $$

This will result in two solutions, however, one can identify the correct symbolic solution by substituting the given ${}^1\theta_1 = \underline{a}_1 = -25^\circ$ and ${}^1\theta_3 = 0^\circ$. An analogous set of equations for the intersection of curve 1 and 2 can be solved for ${}^1\overline{\mathbf{x}}$. Once the symbolic equations for ${}^1\underline{\mathbf{x}}$ and ${}^1\overline{\mathbf{x}}$ are known, equation (2.10) can be applied to compute the limits on the integral in Green's theorem for the variable joint 2, i.e., $[{}^1\underline{\theta}_2, {}^1\overline{\theta}_2]$, so that (2.11) becomes

$$ \mathcal{A}_1 = \frac{1}{2}\int_{{}^1\underline{\theta}_2}^{{}^1\overline{\theta}_2} \left(\left[f_1({}^1\theta_2)\frac{df_2({}^1\theta_2)}{d{}^1\theta_2} - f_2({}^1\theta_2)\frac{df_1({}^1\theta_2)}{d{}^1\theta_2} \right] d{}^1\theta_2 \right) \tag{2.20} $$

where

$$ \begin{aligned} f_1({}^1\theta_2) &= l_1\cos(\underline{a}_1) + l_2\cos(\underline{a}_1 + {}^1\theta_2) + l_3\cos(\underline{a}_1 + \\ &\quad {}^1\theta_2 + 0^\circ) \\ f_2({}^1\theta_2) &= l_1\sin(\underline{a}_1) + l_2\sin(\underline{a}_1 + {}^1\theta_2) + l_3\sin(\underline{a}_1 + \\ &\quad {}^1\theta_2 + 0^\circ) \\ df_1({}^1\theta_2) &= -l_2\sin(\underline{a}_1 + {}^1\theta_2) - l_3\sin(\underline{a}_1 + {}^1\theta_2 + 0^\circ) \\ df_2({}^1\theta_2) &= l_2\cos(\underline{a}_1 + {}^1\theta_2) + l_3\cos(\underline{a}_1 + {}^1\theta_2 + 0^\circ). \end{aligned} \tag{2.21} $$

$$
\begin{aligned}
{}^1\underline{\theta}_2 = {} & \overline{a}_1 + \pi - \cos^{-1}((\cos(\overline{a}_1) - 4\cos(\underline{a}_3/2)\cos(\overline{a}_1) + \cos(\overline{a}_1)\cos(0^\circ)^2 - 4\cos(\underline{a}_3/2)^2\cos(\overline{a}_1) - \\
& \sin(\overline{a}_1)\,(8\cos(\underline{a}_3/2) + 4\cos(0^\circ) - 2\cos(0^\circ)^2 - 4\cos(0^\circ)^3 - \cos(0^\circ)^4 + 16\cos(\underline{a}_3/2)\cos(0^\circ) - 8 \\
& \cos(\underline{a}_3/2)^2 - 32\cos(\underline{a}_3/2)^3 - 16\cos(\underline{a}_3/2)^4 + 8\cos(\underline{a}_3/2)\cos(0^\circ)^2 + 16\cos(\underline{a}_3/2)^2\cos(0^\circ) + 8 \\
& \cos(\underline{a}_3/2)^2\cos(0^\circ)^2 + 3)^{(1/2)} + 2\cos(\overline{a}_1)\cos(0^\circ))/(2\,(\cos(0^\circ) + 1))).
\end{aligned}
$$

$$
\begin{aligned}
{}^1\overline{\theta}_2 = {} & \overline{a}_1 + \cos^{-1}(-(\cos(\overline{a}_1 + \overline{a}_2 - 0^\circ) + \cos(\overline{a}_1 + \overline{a}_2 \text{ - } 0^\circ)/4 + \cos(\overline{a}_1 + \overline{a}_2 + 0^\circ)/4 - \cos(3\overline{a}_1 + 2 \\
& \overline{a}_2) + (7\cos(\overline{a}_1 + \overline{a}_2))/2 + \cos(\overline{a}_1 + \overline{a}_2 + 0^\circ) - \cos(\overline{a}_1 - \overline{a}_2) + \cos(\overline{a}_1 + 2\,\overline{a}_2) - \cos(3\,\overline{a}_1 + \overline{a}_2) + 2 \\
& \sin(\overline{a}_1)\,(6\cos(2\,\overline{a}_1) - 2\cos(2\,\overline{a}_2) - 2\cos(4\,\overline{a}_1) + \cos(0^\circ)/2 - \cos(0^\circ) - \cos(0^\circ)/8 - 4\cos(2 \\
& \overline{a}_1 + \overline{a}_2 + 0^\circ) + 4\cos(2\,\overline{a}_1 - \overline{a}_2) + 4\cos(2\,\overline{a}_1 + 2\,\overline{a}_2) - 2\cos(4\,\overline{a}_1 + 2\,\overline{a}_2) - 4\cos(2\,\overline{a}_1 - 0^\circ) - \cos(2 \\
& \overline{a}_1 - 0^\circ) - \cos(2\,\overline{a}_1 + 0^\circ) + 4\cos(\overline{a}_2 + 0^\circ) - 4\cos(2\,\overline{a}_1 + \overline{a}_2 - 0^\circ) - \cos(2\,\overline{a}_1 + \overline{a}_2 - 0^\circ) - \cos(2 \\
& \overline{a}_1 + \overline{a}_2 + 0^\circ) - 6\cos(\overline{a}_2) + 9\cos(0^\circ) + 6\cos(2\,\overline{a}_1 + \overline{a}_2) - 4\cos(4\,\overline{a}_1 + \overline{a}_2) + 4\cos(\overline{a}_2 - 0^\circ) - 4 \\
& \cos(2\,\overline{a}_1 + 0^\circ) + \cos(\overline{a}_2 - 0^\circ) + \cos(\overline{a}_2 + 0^\circ) - 19/8)^{(1/2)} + \sin(\overline{a}_1 + \overline{a}_2)\,(6\cos(2\,\overline{a}_1) - 2\cos(2 \\
& \overline{a}_2) - 2\cos(4\overline{a}_1) + \cos(0^\circ)/2 - \cos(0^\circ) - \cos(\,0^\circ)/8 - 4\cos(2\,\overline{a}_1 + \overline{a}_2 + 0^\circ) + 4\cos(2\,\overline{a}_1 - \overline{a}_2) + 4 \\
& \cos(2\,\overline{a}_1 + 2\,\overline{a}_2) - 2\cos(4\,\overline{a}_1 + 2\,\overline{a}_2) - 4\cos(2\,\overline{a}_1 - 0^\circ) - \cos(2\,\overline{a}_1 - 0^\circ) - \cos(2\,\overline{a}_1 + 0^\circ) + 4 \\
& \cos(\overline{a}_2 + 0^\circ) - 4\cos(2\,\overline{a}_1 + \overline{a}_2 - 0^\circ) - \cos(2\,\overline{a}_1 + \overline{a}_2 - 0^\circ) - \cos(2\,\overline{a}_1 + \overline{a}_2 + 0^\circ) - 6\cos(\overline{a}_2) + 9 \\
& \cos(0^\circ) + 6\cos(2\,\overline{a}_1 + \overline{a}_2) - 4\cos(4\,\overline{a}_1 + \overline{a}_2) + 4\cos(\overline{a}_2 - 0^\circ) - 4\cos(2\,\overline{a}_1 + 0^\circ) + \cos(\overline{a}_2 - \\
& 0^\circ) + \cos(\overline{a}_2 + 0^\circ) - 19/8)^{(1/2)})/(4\cos(2\,\overline{a}_1) + 2\cos(2\,\overline{a}_1 + \overline{a}_2 + 0^\circ) + 2\cos(2\,\overline{a}_1 - 0^\circ) - 2 \\
& \cos(\overline{a}_2 + 0^\circ) + 2\cos(2\,\overline{a}_1 + \overline{a}_2 - 0^\circ) - 4\cos(\overline{a}_2) - 6\cos(0^\circ) + 4\cos(2\,\overline{a}_1 + \overline{a}_2) - 2\cos(\overline{a}_2 - 0^\circ) + 2 \\
& \cos(2\,\overline{a}_1 + 0^\circ) - 6)).
\end{aligned}
\tag{2.22}
$$

The actual integration of (2.20) must be performed symbolically, and the result is lengthy (See Appendix A.2). To evaluate (2.12), the process is repeated for curves 2-6 that are shown in Fig. 2.2(a) resulting in $\mathcal{A}_{\mathcal{W}_F}$. Once the equation of $\mathcal{A}_{\mathcal{W}_F}$ is obtained, the gradient is the partial derivative of $\mathcal{A}_{\mathcal{W}_F}$ with respect to $\boldsymbol{L}$ and $\boldsymbol{A}$ as given in (2.14)[4]. Once again, this must be done symbolically, which results in very long equations, i.e., hundreds of pages, that are available in

[4]In this example only $\boldsymbol{A}$ is variable.

the supplementary material. We should note that, despite the fact that generating these expressions takes several minutes (using MATLAB Symbolic Math Toolbox), they can be evaluated in seconds so that the gradient ascent algorithm is computationally efficient.

As we move along the gradient to increase the failure-tolerant workspace area, the red boundary curves 2 and 5 eventually disappear so that $N = 4$. As we continue to follow the gradient, we arrive at the situation in Fig. 2.2(b), that illustrates a critical point, as curve a just becomes a boundary curve when $\boldsymbol{A} = \{-34.2149^\circ, 34.2149^\circ, -51.1152^\circ,$
$111.1152^\circ, -60^\circ, 111.1152^\circ\}$. This is followed by curves b, c, d, and e, each being added as new boundary curves until $N = 9$ as we approach the situation shown in Fig. 2.2(c). In Fig. 2.2(c) we arrive at a different type of critical point, when $\boldsymbol{A} = \{\ -21.2854^\circ,\ 21.2854^\circ,\ -79.8991^\circ,$ $100.1008^\circ,\ -100.1008^\circ,\ 100.1008^\circ\ \}$. At this point, curves 6 and f become identical, however, curve f simply takes the place of curve 6 in the area calculations. The more complicated case of identical boundary curves occurs when the gradients of two (or more) boundaries have competing gradients as discussed in subsection 2.4.3. This occurs three times, i.e., with the curves labeled 4 and $4'$ in Fig. 2.2(b); curves 9, $9'$, and $9''$ in Fig. 2.2(c); and curves 9, $9'$, $9''$, and $9'''$ in Fig. 2.3. Each time this occurs, one must include a constraint so that both curves are simultaneously considered. The three additional constraints are: $\overline{a}_2 = \overline{a}_3$, $-\underline{a}_3 = \overline{a}_3$, and $-\underline{a}_2 = \overline{a}_2$. With these three constraints being applied, the gradient ascent algorithm ultimately converges to the optimal solution where the gradient is zero[5], as shown in Fig. 2.3 with the boundary curves given in Table 2.3. This solution is consistent with the approximated value of $\mathcal{A}_{\mathcal{W}_F}$ in [33].

We now summarize the computations required in this example. First the symbolic expressions for $\mathcal{A}_{\mathcal{W}_F}$ and $\nabla\mathcal{A}_{\mathcal{W}_F}\ (\boldsymbol{A})$ are obtained for the boundary curves shown in Fig. 2(a). Then (2.13) is repeatedly applied, to increase the area until boundary curve 2 disappears. At this point new symbolic expressions for $\mathcal{A}_{\mathcal{W}_F}$ and $\nabla\mathcal{A}_{\mathcal{W}_F}(\boldsymbol{A})$ are employed and (2.13) continues to be applied. This process occurs multiple times as curve 5 disappears and curves a, b, c, d, and e are added,

[5]The critical points, i.e., where the gradients are zero, are evaluated to make sure that they are local maxima and not points of inflection.

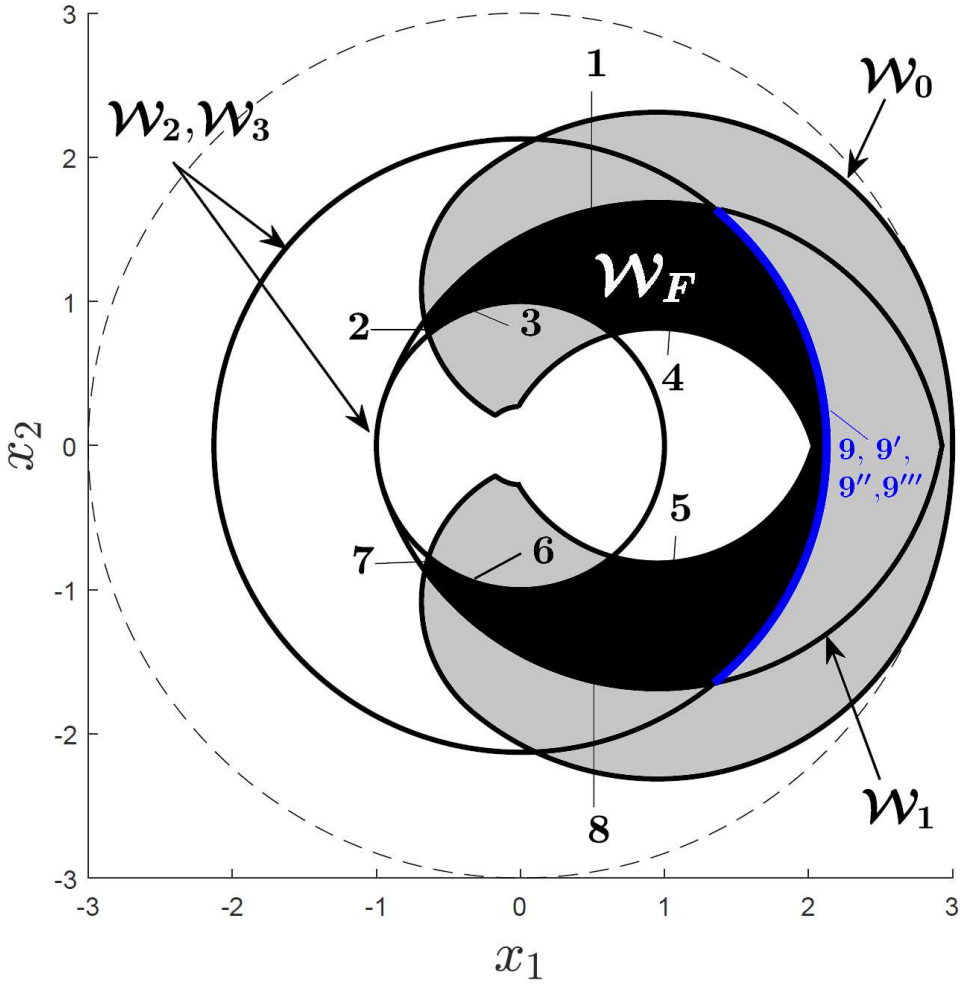


Figure 2.3: This depicts the optimal value of $\mathcal{A}_{\mathcal{W}_F} = 3.5621$ m^2 for a robot with link lengths $\boldsymbol{L} = \{1, 1, 1\}$ that occurs when $\boldsymbol{A} = \{-18.2074°, 18.2074°, -111.3415°, 111.3415°, -111.3415°, -111.3415°\}$ and where $\nabla\mathcal{A}_{\mathcal{W}_F}(\boldsymbol{A}) = 0$. Note that there are four identical boundary curves at the optimal solution, i.e., the blue curves labeled 9, 9′, 9″, and 9‴.

with each time new symbolic expressions for $\mathcal{A}_{\mathcal{W}_F}$ and $\nabla\mathcal{A}_{\mathcal{W}_F}(\boldsymbol{A})$ being computed. When curves 4 and 4' become identical, they have competing gradients so that the constraint $\overline{a}_2 = \overline{a}_3$ must be applied in order to make the two area equations, and their corresponding gradients, to be equal. This allows (2.13) to continue to be applied without chattering. Likewise, the additional constraints of $-\underline{a}_3 = \overline{a}_3$, and $-\underline{a}_2 = \overline{a}_2$ must be applied when curves 9, 9′, and 9″ coincide and curves 9, 9′, 9″, and 9‴ coincide, respectively. All of the symbolic expressions for $\mathcal{A}_{\mathcal{W}_F}$ and $\nabla\mathcal{A}_{\mathcal{W}_F}(\boldsymbol{A})$ are quite complicated, and so are included in the supplementary material.

Table 2.3: Boundary Curves of $\boldsymbol{\mathcal{W}_F}$ for Fig. 2.3

Curve	θ_1	θ_2	θ_3	Workspace
1	$\underline{a}_1$	$[{}^1\underline{\theta}_2, {}^1\overline{\theta}_2]$	0	$\boldsymbol{\mathcal{W}_1}$
2	$\overline{a}_1$	$\overline{a}_2$	$[{}^2\underline{\theta}_3, {}^2\overline{\theta}_3]$	$\boldsymbol{\mathcal{W}_0}$
3	$[{}^3\underline{\theta}_1, {}^3\overline{\theta}_1]$	0 (π)	π (0)	$\boldsymbol{\mathcal{W}_2}$ ($\boldsymbol{\mathcal{W}_3}$)
4	$\underline{a}_1$	$[{}^4\underline{\theta}_2, {}^4\overline{\theta}_2]$	$\overline{a}_3$	$\boldsymbol{\mathcal{W}_0}$
5	$\overline{a}_1$	$[{}^5\underline{\theta}_2, {}^5\overline{\theta}_2]$	$\underline{a}_3$	$\boldsymbol{\mathcal{W}_0}$
6	$[{}^6\underline{\theta}_1, {}^6\overline{\theta}_1]$	0 (π)	π (0)	$\boldsymbol{\mathcal{W}_2}$ ($\boldsymbol{\mathcal{W}_3}$)
7	$\underline{a}_1$	$\underline{a}_2$	$[{}^7\underline{\theta}_3, {}^7\overline{\theta}_3]$	$\boldsymbol{\mathcal{W}_0}$
8	$\overline{a}_1$	$[{}^8\underline{\theta}_2, {}^8\overline{\theta}_2]$	0°	$\boldsymbol{\mathcal{W}_1}$
9	$[{}^9\underline{\theta}_1, {}^9\overline{\theta}_1]$	$\overline{a}_2$	$\frac{-\overline{a}_2}{2}$	$\boldsymbol{\mathcal{W}_2}$
$9'$	$[{}^{9'}\underline{\theta}_1, {}^{9'}\overline{\theta}_1]$	$\frac{-\overline{a}_3}{2}$	$\overline{a}_3$	$\boldsymbol{\mathcal{W}_3}$
$9''$	$[{}^{9''}\underline{\theta}_1, {}^{9''}\overline{\theta}_1]$	$\frac{-\underline{a}_3}{2}$	$\underline{a}_3$	$\boldsymbol{\mathcal{W}_3}$
$9'''$	$[{}^{9'''}\underline{\theta}_1, {}^{9'''}\overline{\theta}_1]$	$\underline{a}_2$	$\frac{-\underline{a}_2}{2}$	$\boldsymbol{\mathcal{W}_2}$

2.5.3 Example 2: Optimal Robot Design

For example 2, one is interested in determining both the optimal link lengths and the artificial joint limits. We will perform the gradient ascent optimization from two different starting values, the first of which is that shown in Fig. 2.3. At this starting point we have the three constraints discussed above in example 1, however, because the link lengths are now variable, one must add a fourth constraint of $l_1 = l_3$. This keeps $\boldsymbol{\mathcal{W}_2} = \boldsymbol{\mathcal{W}_3}$, and prevents their identical boundary curves, shown in blue in Fig. 2.4, from having opposing gradients. With these constraints imposed, the algorithm converges to the optimal solution where $\boldsymbol{\mathcal{W}_0} = \boldsymbol{\mathcal{W}_1}$, as shown in Fig. 2.4 and Table 2.4. This locally optimal value of $\mathcal{A}_{\mathcal{W}_F}$ is approximately 3.5% larger than that shown in [33], due to an approximately 18% difference in l_1. However, this is not the globally optimal value of $\mathcal{A}_{\mathcal{W}_F}$.

To identify the optimal value of $\mathcal{A}_{\mathcal{W}_F}$ one must start from multiple random initial values. If one starts at $\boldsymbol{L} = \{0.7, 1.2, 1.1\}$ and $\boldsymbol{A} = \{-130^\circ, 130^\circ, -60^\circ, 150^\circ, 90^\circ, 120^\circ\}$, then the gradient ascent algorithm will converge to a optimal solution as shown in Fig. 2.5 and Table 2.5, where $\overline{a}_1 = -\underline{a}_1 = 180^\circ$ so that all of the workspaces are annuluses. However, there are multiple constraints that need to be employed as the optimization progresses, i.e., we use $\underline{a}_3 = \overline{a}_3$, $l_2 = l_3$, and $\overline{a}_3 = \cos^{-1}\left((17l_2^2 - 36l_2 + 18)/l_2^2\right)$. It is important to note that there is an infinite set of optimal

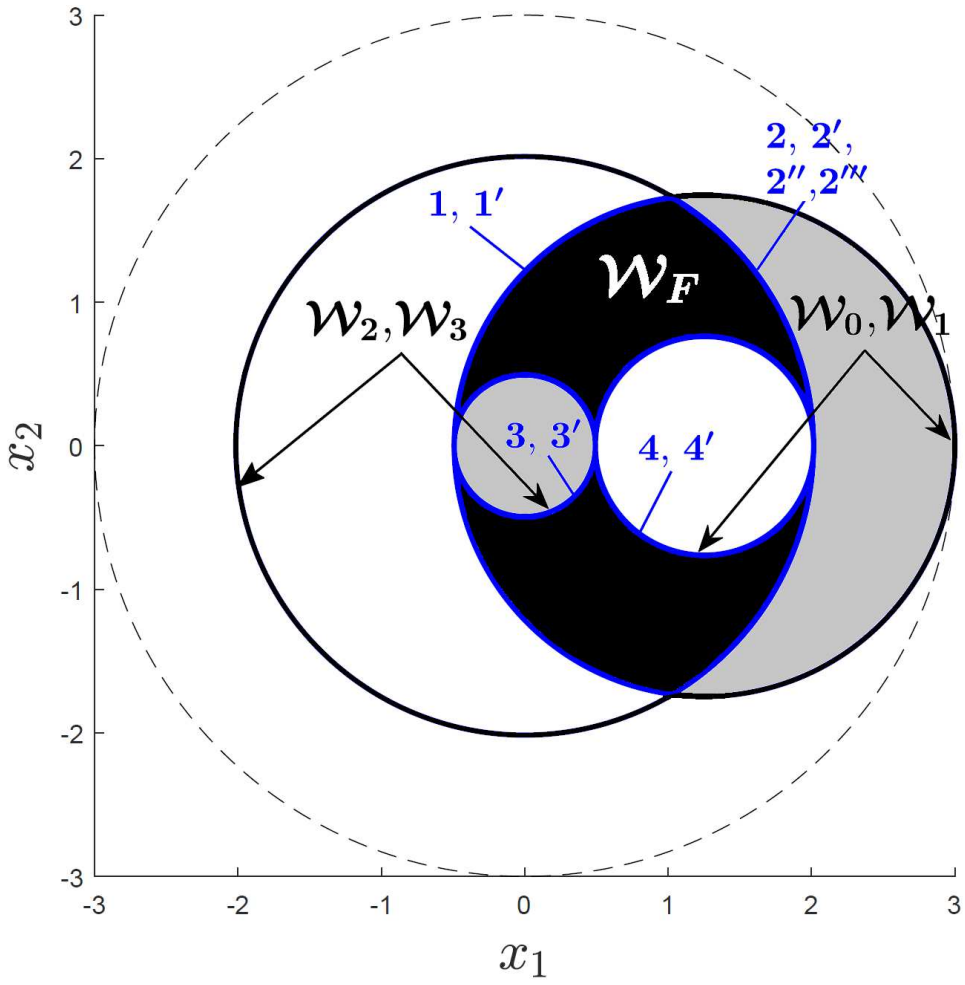


Figure 2.4: This figure illustrates the locally optimal value of $\mathcal{A}_{\mathcal{W}_F} = 3.8435$ m^2, i.e., $\nabla\mathcal{A}_{\mathcal{W}_F}(\boldsymbol{L}, \boldsymbol{A}) = 0$, if one attempts to use the gradient ascent optimization with the starting values from Fig. 2.3. The value of $\boldsymbol{L}$ and $\boldsymbol{A}$ at the locally optimal $\mathcal{A}_{\mathcal{W}_F}$ are $\boldsymbol{L} = \{1.2538, 0.4923, 1.2538\}$ and $\boldsymbol{A} = \{-0^\circ, 0^\circ, -180^\circ, 180^\circ, -180^\circ, 180^\circ\}$. The blue curves indicate multiple identical boundary curves.

solutions that can be characterized by the relationships between the link lengths and artificial joint limits. In particular, the link lengths must satisfy the following relationship:

$$\begin{aligned} l_1 &= 0.5, \\ 1 \leq l_2 &\leq 1.5, \\ l_3 &= 2.5 - l_2 \end{aligned} \tag{2.23}$$

and the artificial limits of joint 2 must satisfy the following inequalities when $\underline{a}_3 > 0$:

Table 2.4: Boundary Curves of $\boldsymbol{\mathcal{W}_F}$ for Fig. 2.4

Curve	θ_1	θ_2	θ_3	Workspace
1	$\overline{a}_1$	$[{}^1\underline{\theta}_2, {}^1\overline{\theta}_2]$	0	$\boldsymbol{\mathcal{W}_0}$
$1'$	$\overline{a}_1$	$[{}^{1'}\underline{\theta}_2, {}^{1'}\overline{\theta}_2]$	0	$\boldsymbol{\mathcal{W}_1}$
2	$[{}^2\underline{\theta}_1, {}^2\overline{\theta}_1]$	$\underline{a}_2$	π	$\boldsymbol{\mathcal{W}_2}$
$2'$	$[{}^{2'}\underline{\theta}_1, {}^{2'}\overline{\theta}_1]$	$\overline{a}_2$	π	$\boldsymbol{\mathcal{W}_2}$
$2''$	$[{}^{2''}\underline{\theta}_1, {}^{2''}\overline{\theta}_1]$	π	$\underline{a}_3$	$\boldsymbol{\mathcal{W}_3}$
$2'''$	$[{}^{2'''}\underline{\theta}_1, {}^{2'''}\overline{\theta}_1]$	π	$\overline{a}_3$	$\boldsymbol{\mathcal{W}_3}$
3	$[{}^3\underline{\theta}_1, {}^3\overline{\theta}_1] = 2\pi$	0	π	$\boldsymbol{\mathcal{W}_2}$
$3'$	$[{}^{3'}\underline{\theta}_1, {}^{3'}\overline{\theta}_1] = 2\pi$	π	0	$\boldsymbol{\mathcal{W}_3}$
4	$\overline{a}_1$	$[{}^4\underline{\theta}_2, {}^4\overline{\theta}_2] = 2\pi$	π	$\boldsymbol{\mathcal{W}_0}$
$4'$	$\overline{a}_1$	$[{}^{4'}\underline{\theta}_2, {}^{4'}\overline{\theta}_2] = 2\pi$	π	$\boldsymbol{\mathcal{W}_1}$

$$-\pi \leq \underline{a}_2 \leq -\cos^{-1}(\beta)$$
$$\pi \geq \overline{a}_2 \geq -\cos^{-1}(\beta) + \pi$$

otherwise, i.e., $\underline{a}_3 < 0$:

$$\begin{aligned} -\pi \leq \underline{a}_2 \leq \cos^{-1}(\beta) - \pi \\ \pi \geq \overline{a}_2 \geq \cos^{-1}(\beta) \end{aligned} \tag{2.24}$$

where $\beta = (5l_2 - 4)/3l_2$, i.e., $38.9424^\circ \leq \cos^{-1}(\beta) \leq 70.5288^\circ$, and

$$\underline{a}_3 = \overline{a}_3 = \pi \pm \cos^{-1}\left(\frac{2l_2^2 - 5l_2 + 4}{-2l_2^2 + 5l_2}\right). \tag{2.25}$$

Thus one has a choice in both l_2 and the artificial joint limits for joint 2. However, in practice, one would pick $l_2 = 1.25$ and the values of $\underline{a}_2$ and $\overline{a}_2$ with minimum absolute values as in Fig. 2.5 because this will maximize the areas of both $\boldsymbol{\mathcal{W}_1}$ and $\boldsymbol{\mathcal{W}_2}$ without affecting $\boldsymbol{\mathcal{W}_F}$.

2.6 Summary

This paper presented a technique for maximizing the area of a failure-tolerant workspace after an arbitrary single locked-joint failure for planar 3R robots of arbitrary link lengths. The technique

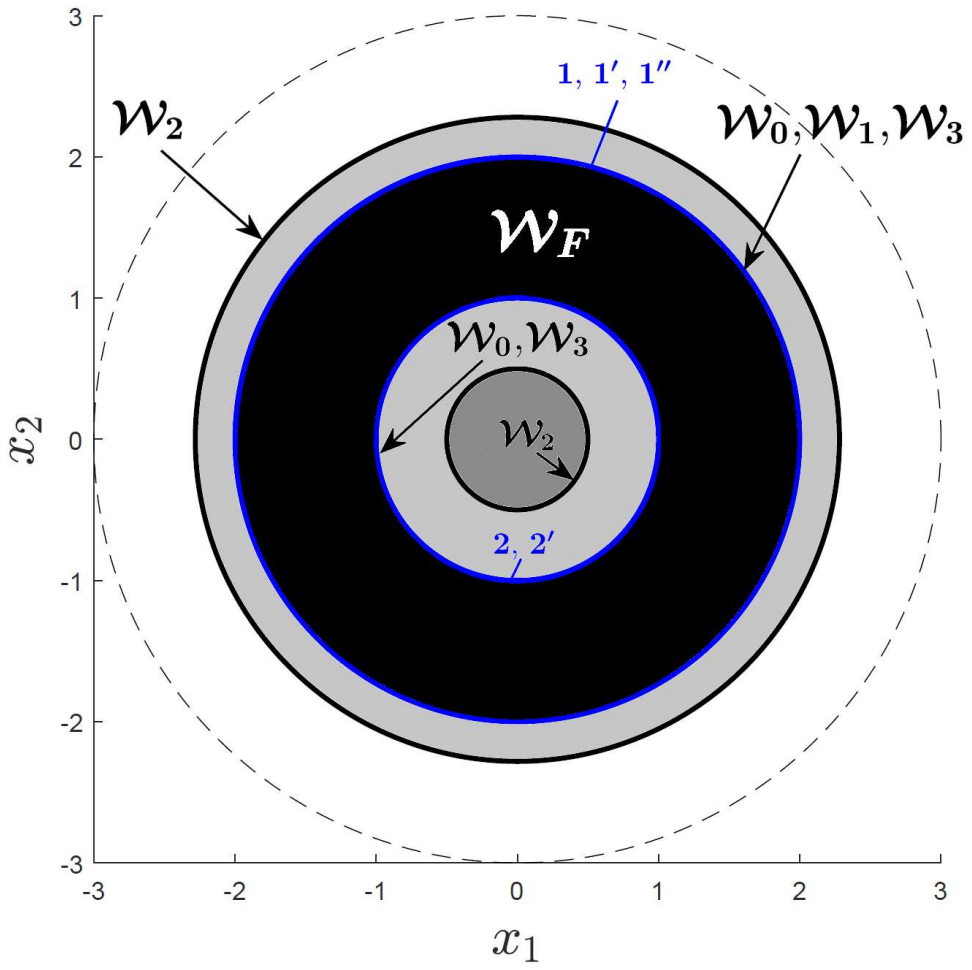


Figure 2.5: This figure shows a significantly larger optimal value of $\mathcal{A}_{\mathcal{W}_F} = 9.4247$ m^2, i.e., $\nabla \mathcal{A}_{\mathcal{W}_F}(\boldsymbol{L}, \boldsymbol{A}) = 0$, when the proposed algorithm is started at random values of $\boldsymbol{L}$ and $\boldsymbol{A}$. The optimal solution converges to $\boldsymbol{L} = \{0.5,\ 1.25,\ 1.25\}$ and $\boldsymbol{A} = \{-180°,\ 180°,\ -53.1301°,\ 126.8698°,\ 106.2602°,\ 106.2602°\}$. The blue curves are identical boundary curves of $\boldsymbol{\mathcal{W}_F}$.

is based on the gradient ascent method and can be used to determine the optimal artificial joint limits for an existing robot, or to design a robot with optimal link lengths and artificial joint limits. It was shown how to deal with the numerical issues associated with gradient-based techniques due to discontinuities of the gradient caused by changes in the boundary curves of the failure-tolerant workspace. The technique was illustrated on two examples, and resulted in a more accurate computation of a previous result, as well the identification of a previously unknown optimal failure-tolerant workspace area.

Table 2.5: Boundary Curves of $\boldsymbol{\mathcal{W}_F}$ for Fig. 2.5

Curve	θ_1	θ_2	θ_3	Workspace
1	$[{}^1\underline{\theta}_1, {}^1\overline{\theta}_1] = 2\pi$	$\underline{a}_2$	$\underline{a}_3$	$\boldsymbol{\mathcal{W}_0}$
$1'$	$[{}^{1'}\underline{\theta}_1, {}^{1'}\overline{\theta}_1] = 2\pi$	π	0	$\boldsymbol{\mathcal{W}_1}$
$1''$	$[{}^{1''}\underline{\theta}_1, {}^{1''}\overline{\theta}_1] = 2\pi$	$\underline{a}_2$	$\overline{a}_3$	$\boldsymbol{\mathcal{W}_3}$
2	$[{}^2\underline{\theta}_1, {}^2\overline{\theta}_1] = 2\pi$	$\overline{a}_2$	$\overline{a}_3$	$\boldsymbol{\mathcal{W}_0}$
$2'$	$[{}^{2'}\underline{\theta}_1, {}^{2'}\overline{\theta}_1] = 2\pi$	$\overline{a}_2$	$\underline{a}_3$	$\boldsymbol{\mathcal{W}_3}$

Chapter 3

A Hybrid Approach for Estimating the Failure-Tolerant Workspace Size of Kinematically Redundant Robots[6]

3.1 Introduction

Failure-tolerant robots are frequently required for applications in hazardous and remote environments where performing routine maintenance and repair are not possible. Example applications include handling radioactive materials [3], inspecting nuclear reactors [6], gathering samples near potentially active volcanoes [7], and exploring space [9] or the deep sea [10]. There have been studies on assessing a robot's reliability using fault trees [12]. Other studies focused on enhancing a robot's tolerance to failure, including fault diagnosis [14], detection and isolation [15], and identification [16], e.g., identifying the locked position of the failed joint [17]. Most previous studies have assumed a locked-joint failure model, where the joint is locked due to the failure itself or because brakes are employed.

Because kinematically redundant robots have the ability to perform their tasks even if one or more joints fail, many failure-tolerance techniques have been developed for optimizing the local and/or global fault-tolerant measures of a redundant robot. Locally, optimizing the singular values of the Jacobian matrix for redundant robot is frequently used to design failure-tolerant robots [21, 22] and determine optimal configurations during their control [23]. Global measures refer to the robot's workspace, e.g., where the reachability of critical task locations can be guaranteed because they have large self-motion manifolds [24]. It also refers to identifying an entire region that is

[6]This chapter has been published in [2]

reachable before and after an arbitrary single locked-joint failure, referred to as the failure-tolerant workspace ($\mathcal{W}_F$) [25], [26].

A number of methods have been developed to compute the workspace size before or after a joint failure. One approach is to determine expressions for the workspace boundaries [29] and then integrate between these boundaries to determine the size of the workspace [26]. Other approaches include estimating the workspace size numerically, e.g., using Monte-Carlo integration [22, 30] or voxelization methods on three- and six-dimensional workspaces [41].

One study has shown that in the worst case, one can guarantee the existence of $\mathcal{W}_F$ for a single locked-joint failure by having two degrees of redundancy (DOR) [27]. However, by restricting the joints prior to a failure, i.e., employing artificial joint limits that are released after the failure, one can guarantee a $\mathcal{W}_F$ with only a single DOR [28]. Determining the artificial joint limits that maximize $\mathcal{W}_F$ for planar 3R and 4R robots after one and two arbitrary locked-joint failures, respectively has been discussed in [33]. In our previous work [1], we maximized $\mathcal{W}_F$ for planar 3R robots by using the gradient ascent method on the symbolic expressions of the $\mathcal{W}_F$ area and its gradient with respect to the artificial joint limits. All previous approaches to maximizing $\mathcal{W}_F$ have been limited to two-dimensional workspaces due to the computational complexity.

Computing $\mathcal{W}_F$ for high-dimensional workspaces is problematic because voxelization methods suffer from exponential growth in the dimension of the workspace and there is no known way to symbolically express the boundaries of such workspaces. This work addresses these limitations by developing a hybrid approach that uses a combination of discretization in two dimensions, Monte-Carlo integration for estimating orientation volumes, and an efficient numerical procedure for computing workspace envelopes. This method is applicable to fully general six-dimensional workspaces for robots with an arbitrary number of degrees of freedom (DOFs). We then use our approach for efficiently computing $\mathcal{W}_F$ to determine optimal artificial joint limits that maximize $\mathcal{W}_F$ by applying a coordinate ascent method.

The remainder of this work is organized as follows. In the next section, we review a mathematical definition of the failure-tolerant workspace. In Section 3.3, we formulate our method of

estimating the failure-tolerant workspace size. In Section 3.4, we combine our method with an optimization technique to identify artificial joint limits that maximize the failure-tolerant workspace size. We then present two illustrative examples in Section 3.5. Finally, the conclusions are presented in Section 3.6.

3.2 Background on Computing Failure-Tolerant Workspace

3.2.1 Overview

In this section, the definition of a failure-tolerant workspace and the conditions to check if a workspace location $\mathbf{x}$ belongs to $\boldsymbol{\mathcal{W}_F}$ are reviewed as discussed in [1, 26, 28]. In those works, the concept of artificial joint limits was used to obtain a $\boldsymbol{\mathcal{W}_F}$. Artificial joint limits are software-imposed limits on the range of the joints prior to any failure. Their purpose is to prevent the robot from failing in a configuration that would result in a very restrictive workspace, i.e., one that does not guarantee any $\boldsymbol{\mathcal{W}_F}$. Once a joint failure does occur, the software limits on the remaining joints are released so that only the physical joint limits affect their range of motion. Applying artificial limits will typically decrease the pre-failure workspace, however, if chosen appropriately, it guarantees a post-failure workspace. In this work, the proposed technique is applicable for any serial robot with arbitrary kinematics where it is assumed that its joints, either revolute or prismatic, have no physical limits and one would like to be failure tolerant to a single arbitrary locked-joint failure. The robot must be kinematically redundant, i.e., $n > m$ where n is the number of joints, m is the workspace dimension, and the DOR is equal to $n - m$.

3.2.2 Definition of Failure-Tolerant Workspace

The generalized m-dimensional vector $\mathbf{x}$ represents the position and/or orientation of the robot's end-effector in the workspace, $\mathbf{x} \in \boldsymbol{\mathcal{W}} \subset \mathbb{R}^m$, and the n-dimensional vector $\boldsymbol{\theta}$ represents the robot joint angles in the configuration space, $\boldsymbol{\mathcal{C}}$, $\boldsymbol{\theta} \in \boldsymbol{\mathcal{C}} \subset \mathbb{R}^n$. The forward kinematic function, denoted $\mathbf{f}$, maps joint angles to workspace position and/or orientation,

$$\mathbf{x} = \mathbf{f}(\boldsymbol{\theta}). \tag{3.1}$$

Prior to failure, the angle of joint i, denoted θ_i, is bounded by lower $\underline{a}_i$ and upper $\overline{a}_i$ artificial joint limits where $\underline{a}_i$ and $\overline{a}_i \in [-2\pi,\ 2\pi]$, and $i = \{1, 2, \ldots, n\}$. The range of joint angles between $\underline{a}_i$ and $\overline{a}_i$ is defined as $A_i = [\underline{a}_i,\ \overline{a}_i]$. This yields in the pre-failure configuration space $\boldsymbol{\mathcal{C}_A} = A_1 \times \cdots \times A_n$. Mapping $\boldsymbol{\mathcal{C}_A}$ into the workspace result in the pre-failure workspace,

$$\boldsymbol{\mathcal{W}_0} = \mathbf{f}(\boldsymbol{\mathcal{C}_A}) = \{\mathbf{x} = \mathbf{f}(\boldsymbol{\theta}) \mid \boldsymbol{\theta} \in \boldsymbol{\mathcal{C}_A}\}. \tag{3.2}$$

After joint i fails and is locked at $\theta_i = q_i$ where $\underline{a}_i \leq q_i \leq \overline{a}_i$, the artificial limits are released on the remaining working joints. This results in the reduced configuration space that is a hyperplane at $\theta_i = q_i$ in the configuration space $\boldsymbol{\mathcal{C}}$,

$${}^{i}\boldsymbol{\mathcal{C}}(q_i) = \{\boldsymbol{\theta} \in \boldsymbol{\mathcal{C}} \mid \theta_i = q_i\}. \tag{3.3}$$

The guaranteed reachable workspace after joint i fails at q_i between $\underline{a}_i \leq q_i \leq \overline{a}_i$ is the post-failure workspace, denoted $\boldsymbol{\mathcal{W}_i}$, and given by

$$\boldsymbol{\mathcal{W}_i} = \bigcap_{\underline{a}_i \leq q_i \leq \overline{a}_i} \mathbf{f}({}^{i}\boldsymbol{\mathcal{C}}(q_i)). \tag{3.4}$$

The guaranteed reachable workspace both before and after an arbitrary single locked-joint failure is the failure-tolerant workspace,

$$\boldsymbol{\mathcal{W}_F} = \bigcap_{i \in \mathbf{F} \cup \{0\}} \boldsymbol{\mathcal{W}_i} \tag{3.5}$$

where $\mathbf{F} \subset \{1, 2, \ldots, n\}$ is the failure index for the joints that are prone to failures.

3.2.3 Identification of a Failure-Tolerant Workspace Point

Previous work [26] has identified two conditions for determining if a workspace location $\mathbf{x}$ belongs to $\boldsymbol{\mathcal{W}_F}$. The pre-image of $\mathbf{x}$, denoted $\mathbf{f}^{-1}(\mathbf{x})$, that is given by

$$\mathbf{f}^{-1}(\mathbf{x}) = \{\boldsymbol{\theta} \in \boldsymbol{\mathcal{C}} \mid \mathbf{f}(\boldsymbol{\theta}) = \mathbf{x}\} \tag{3.6}$$

is used to formulate both conditions. Condition 1 is that $\mathbf{x}$ be reachable prior to a failure, i.e., $\mathbf{x} \in \boldsymbol{\mathcal{W}_0}$, so that

$$\boldsymbol{\mathcal{C}}_A \cap \mathbf{f}^{-1}(\mathbf{x}) \neq \emptyset. \tag{3.7}$$

The above condition states that there must be an intersection between the pre-failure configuration space, $\boldsymbol{\mathcal{C}}_A$, and the pre-image of $\mathbf{x}$, $\mathbf{f}^{-1}(\mathbf{x})$, for at least one configuration. Condition 2 is that $\mathbf{x}$ is reachable after a failure, i.e., $\mathbf{x} \in \boldsymbol{\mathcal{W}_i}$ for $i \in \mathbf{F}$, so that

$$A_i \subset P_i[\mathbf{f}^{-1}(\mathbf{x})] \text{ for } i \in \mathbf{F} \tag{3.8}$$

where P_i is the projection onto the i-th joint axis, i.e., the range of θ_i for all $\boldsymbol{\theta}$ that satisfies $\mathbf{x} = \mathbf{f}(\boldsymbol{\theta})$. Condition 2 means that after joint i is locked at $\theta_i = q_i$ where $\underline{a}_i \leq q_i \leq \overline{a}_i$, the workspace point $\mathbf{x}$ can still be reached because q_i is contained in the i-th component of $\mathbf{f}^{-1}(\mathbf{x})$.

3.3 Formulation for Estimating the Failure-Tolerant Workspace

3.3.1 Calculation of a Three-Dimensional Volume Element

The proposed technique is first illustrated for the workspace of general spatial 4 DOF robots, because they are the simplest spatial redundant robots possible, i.e., robots with only one DOR and that do not include orientation of the end effector. Assume that joint 1, i.e., the base joint, is a revolute joint[7] where Fig. 3.1 is the three-dimensional (3D) workspace of that robot. The pre-image of a workspace point $\mathbf{x}$, i.e., the set of configurations that correspond to $\mathbf{x}$, is given by (3.6). If one rotates $\mathbf{x}$ about the rotation axis of joint 1, i.e., the z-axis in Fig. 3.1, by $\beta \in [-\pi,\ \pi]$, then the pre-image of the rotated $\mathbf{x}$, denoted $\mathbf{x}'$, is computed by evaluating (3.6) for $\mathbf{x}'$ where $\mathbf{x}' = \boldsymbol{R}_z\mathbf{x}$, and

[7] If the first joint is prismatic, an analogous procedure can be performed.

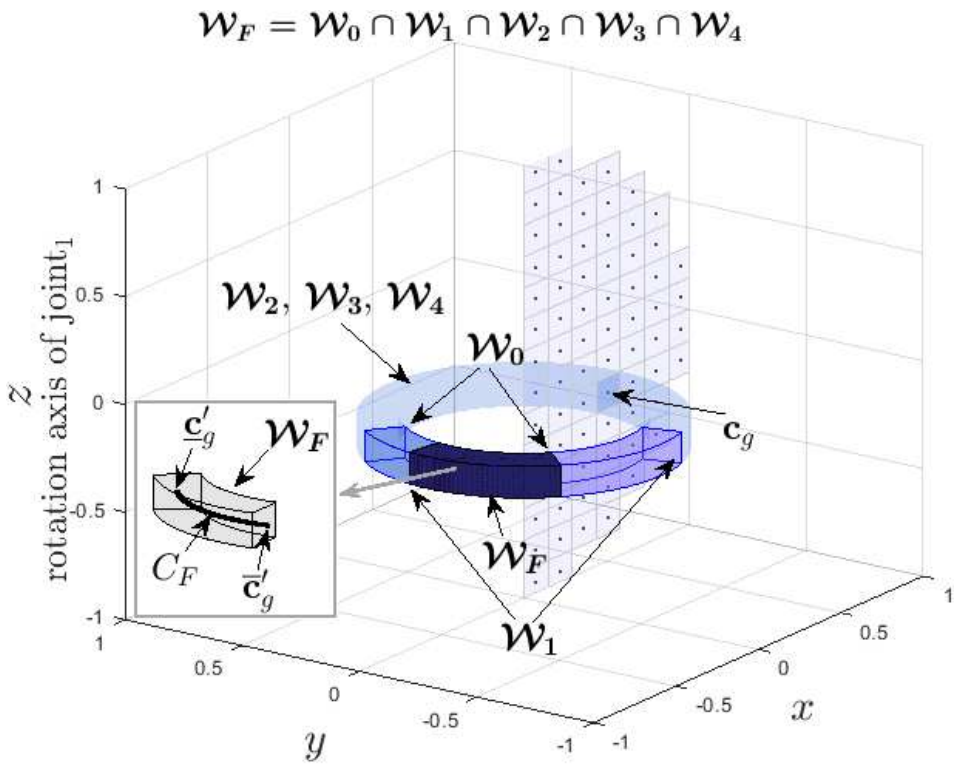


Figure 3.1: A simple example that illustrates how the 3D failure-tolerant workspace volume of a 4 DOF spatial robot is estimated. This figure shows the reachable grid centers in the half-plane. The pre-image of each grid center, $\mathbf{c}_g$, is computed to determine the rotation angles, $\underline{\beta}_F$ and $\overline{\beta}_F$, at the endpoints, $\underline{\mathbf{c}}'_g$ and $\overline{\mathbf{c}}'_g$, of arc C_F where $C_F \subset \boldsymbol{\mathcal{W}_F}$. The volume of the failure-tolerant workspace of one grid is the integration under the grid area over the arc length of C_F.

$$\boldsymbol{R}_z = \begin{bmatrix} \cos(\beta) & -\sin(\beta) & 0 \\ \sin(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.9}$$

The pre-image of $\mathbf{x}'$ is identical to that of $\mathbf{x}$ except that every configuration's joint one value is related by $\theta'_1 = \theta_1 + \beta$. This simple relationship means that one does not have to compute the pre-images for the entire workspace, i.e., one of the dimensions can be easily inferred.

In a 3D workspace, one can discretize a half-plane into equal-square grids where the normal of the half-plane is perpendicular to the rotation axis of joint 1. Without loss of generality, consider the half-plane where $x \geq 0$ and $y = 0$ as shown in Fig.3.1. A grid center is denoted $\mathbf{c}_g$ and $\mathbf{c}'_g$ denotes a grid center that has been rotated about the z-axis by an angle β.

After computing the pre-image for a reachable $\mathbf{c}_g$, one can determine the rotation angle β_F, where $\mathbf{c}'_g$ belongs to $\boldsymbol{\mathcal{W}_F}$, i.e., when the two conditions in Section 3.2.3 are satisfied. Moreover, one can determine a range of rotation angles, $\beta_F = [\underline{\beta}_F,\ \overline{\beta}_F]$, for a set of points that form a circular arc, denoted C_F, where the endpoints of C_F are denoted $\underline{\mathbf{c}}'_g$ and $\overline{\mathbf{c}}'_g$. The computation of the rotation angles, $\underline{\beta}_F$ and $\overline{\beta}_F$, is discussed in detail later in this section.

To compute the volume of a single volume element, denoted v_F, one needs to integrate the grid area over the arc length C_F, i.e.,

$$v_F \approx \int_{\underline{\beta}_F}^{\overline{\beta}_F} r\ \Delta g\ d\beta_F, \tag{3.10}$$

where r is the shortest distance from $\mathbf{c}_g$ to the joint 1 axis, i.e., the radius of C_F, and Δg is the grid area. It is important to note that the pre-image of a grid center may consist of a union of disjoint self-motion manifolds (SMM). The ranges of $[\underline{\beta}_F,\ \overline{\beta}_F]$ for these disjoint SMMs may, or may not, overlap. If two ranges overlap, then they are replaced with the union of those two ranges, until no overlapping ranges remain. Let $\mathbf{B}$ denote the set of non-overlapping ranges of $[\underline{\beta}_F,\ \overline{\beta}_F]$ for a given grid center. Therefore, to compute the volume associated with a grid center one must compute (3.10) for each range in $\mathbf{B}$. Finally, the failure-tolerant workspace volume, denoted $\mathcal{V}_F$, is the summation of all the volumes for all grids, i.e.,

$$\mathcal{V}_F \approx \sum_{i=1}^{N_{c_g}} \sum_{j=1}^{|\mathbf{B}|} v_F(i,j), \tag{3.11}$$

where N_{c_g} is the number of reachable $\mathbf{c}_g$.

3.3.2 Calculation of a Six-Dimensional Volume Element

To extend the 3D workspace volume calculations to a 6D workspace (position and orientation) hypervolume, one can estimate the orientation volume over C_F using Monte-Carlo integration and then modify the integrand in (3.10). To estimate the orientation volume, one can use the technique described in [22, 42] for uniform random sampling of orientations at a workspace point. If one denotes the total number of randomly selected orientations as N_o, then an estimate for the failure-

tolerant orientation volume, $\mathcal{V}_{F_o}$, at a failure-tolerant workspace point is given by

$$\mathcal{V}_{F_o} \approx \frac{N_F}{N_o}\pi^2 \tag{3.12}$$

where N_F is the number of failure-tolerant orientations and π^2 is the maximum orientation volume when $N_F = N_o$.

A 6D hypervolume element is the hypervolume along a single C_F where every point in C_F has the same number of failure-tolerant orientations, i.e., N_F is constant. Therefore, one can compute the 6D hypervolume by simply including the estimate of the orientation volume within the integrand of (3.10), which results in

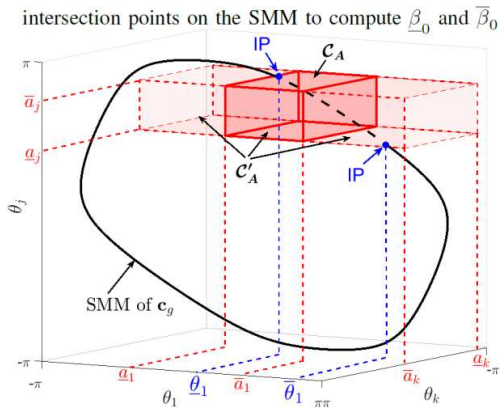
$$v_F \approx \int_{\underline{\beta}_F}^{\overline{\beta}_F} (r\ \Delta g) \left(\frac{N_F}{N_o}\pi^2\right) d\beta_F. \tag{3.13}$$

Then, to compute an estimate of the entire 6D hypervolume, one can apply (3.11). Note that the number of elements in the set $\mathbf{B}$ will likely be much larger because a new range is created for every change in N_F.

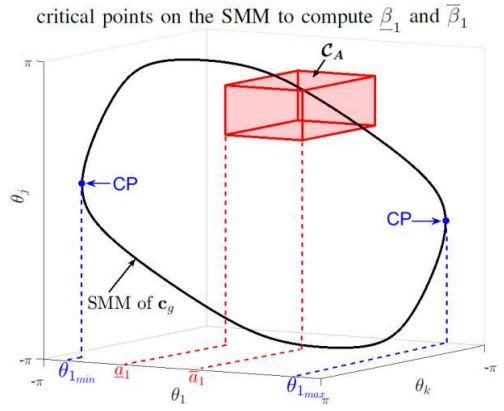
3.3.3 Computation of the Rotation Angle Range

The Rotation Angle Range of C_F

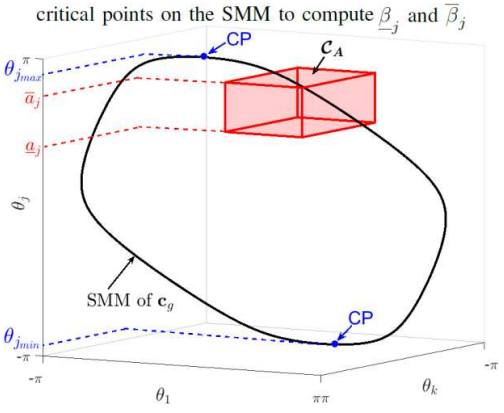
The rotation angle range, $[\underline{\beta}_F,\ \overline{\beta}_F]$, for a single SMM of $\mathbf{c}_g$ is defined by the limits of where $\mathbf{c}'_g$ is reachable both before and after a failure, i.e., when the pre-image of $\mathbf{c}'_g$ satisfies condition 1 and condition 2, respectively, for all failures. Our approach to determining these limits is to compute these ranges for all possible failures and then take the intersection of these ranges, along with the pre-failure range. Prior to a failure, while the artificial joint limits are in place, the pre-failure rotation angle range is denoted $[\underline{\beta}_0,\ \overline{\beta}_0]$ and the arc of reachable points is denoted C_0, where $C_0 \subset \boldsymbol{\mathcal{W}_0}$. After joint i is locked, the rotation angle range is denoted $[\underline{\beta}_i,\ \overline{\beta}_i]$, and the arc of reachable points is denoted C_i, where $C_i \subset \boldsymbol{\mathcal{W}_i}$ and $i = \{1, 2, \ldots, n\}$. Therefore, the range of rotation angles, $[\underline{\beta}_F,\ \overline{\beta}_F]$, for a single SMM of $\mathbf{c}_g$ for the guaranteed set of reachable points, C_F, both



(a)



(b)



(c)

Figure 3.2: This figure shows the intersection points (IP) and critical points (CP) that are located on a SMM of $\mathbf{c}_g$, i.e., a 3D or 6D grid center. The IPs, shown in blue in (a), are determined to compute the pre-failure rotation angle range of $C_0 \subset \boldsymbol{\mathcal{W}_0}$. The dashed portion of the SMM is the intersection with the red-dashed hyperbox, $\boldsymbol{\mathcal{C}'_A}$. In (a), the IPs are the minimum, $\underline{\theta}_1$, and maximum, $\overline{\theta}_1$, values of θ_1 where $\boldsymbol{\theta} \in \boldsymbol{\mathcal{C}'_A}$. The CPs are shown in blue in (b) and (c). These CPs are used to compute the post-failure rotation angle ranges of $C_i \subset \boldsymbol{\mathcal{W}_i}$. In (b) and (c), the CPs are the minimum, $\theta_{i_{min}}$, and maximum, $\theta_{i_{max}}$, values of θ_i over the entire SMM, i.e., where the null space has no component of θ_i.

before and after an arbitrary single locked-joint failure are given by

$$\left[\underline{\beta}_F,\ \overline{\beta}_F\right] = \bigcap_{i=0}^{n} \left[\underline{\beta}_i,\ \overline{\beta}_i\right] \tag{3.14}$$

and C_F is given by

$$C_F = \bigcap_{i=0}^{n} C_i. \tag{3.15}$$

It is important to note that $[\underline{\beta}_F,\ \overline{\beta}_F]$ in (3.14) is computed for each SMM of its associated $\mathbf{c}_g$. Therefore, these ranges of $[\underline{\beta}_F,\ \overline{\beta}_F]$ should be combined into non-overlapping ranges before being included in the set $\mathbf{B}$ when computing (3.11). Furthermore, for a 6D $\mathbf{c}_g$, one must perform these calculations for every one of the N_o orientations at this position. To compute $[\underline{\beta}_i,\ \overline{\beta}_i]$, one needs to identify the intersection points of the SMM of $\mathbf{c}_g$ with the planes associated with artificial joint limits, as well as the critical points of an SMM with respect to each θ_i, which is illustrated in Fig.3.2 and discussed next.

The Rotation Angle Range of C_0

To determine if the range $[\underline{\beta}_0,\ \overline{\beta}_0]$ is nonempty, one needs to find at least one β that satisfies condition 1, i.e., there is at least one configuration in common between $\boldsymbol{\mathcal{C}_A}$ and a SMM of $\mathbf{c}'_g$. To compute $[\underline{\beta}_0,\ \overline{\beta}_0]$ of $\mathbf{c}_g$, one needs to determine the intersection points of the SMM with $\boldsymbol{\mathcal{C}'_A}$, where $\boldsymbol{\mathcal{C}'_A} = [-\pi,\ \pi] \times A_2 \times A_3 \times \ldots \times A_n$, i.e., $\boldsymbol{\mathcal{C}_A}$ with the artificial joint limits on θ_1 released. These points are indicated in blue in Fig. 3.2(a) with the portions of the SMM inside of $\boldsymbol{\mathcal{C}'_A}$ shown with a dashed line. Denote the θ_1 values of these intersection points as $\underline{\theta}_1$ and $\overline{\theta}_1$, whose values are given by

$$\begin{aligned} \underline{\theta}_1 &= \min \theta_1 \ \forall\, \boldsymbol{\theta}\ \in \{\boldsymbol{\theta} \mid \mathbf{f}(\boldsymbol{\theta}) = \mathbf{c}_g, \boldsymbol{\theta} \in \boldsymbol{\mathcal{C}'}_A\} \\ \overline{\theta}_1 &= \max \theta_1 \ \forall\, \boldsymbol{\theta}\ \in \{\boldsymbol{\theta} \mid \mathbf{f}(\boldsymbol{\theta}) = \mathbf{c}_g, \boldsymbol{\theta} \in \boldsymbol{\mathcal{C}'}_A\}. \end{aligned} \tag{3.16}$$

Once $\underline{\theta}_1$ and $\overline{\theta}_1$ are computed, the rotation angle range, $[\underline{\beta}_0,\ \overline{\beta}_0]$, of C_0 is determined by

$$\left[\underline{\beta}_0,\ \overline{\beta}_0\right] = \left[\underline{a}_1 - \overline{\theta}_1,\ \overline{a}_1 - \underline{\theta}_1\right]. \tag{3.17}$$

In practice, to solve (3.17), one needs to compute $\underline{\theta}_1$ and $\overline{\theta}_1$ in (3.16) as follows. The end effector must be at $\mathbf{c}_g$, i.e., $\mathbf{f}(\boldsymbol{\theta}) = \mathbf{c}_g$, while computing the extremal values of θ_1 with $\boldsymbol{\theta} \in \boldsymbol{\mathcal{C}'_A}$, i.e., $\underline{\theta}_1$ or $\overline{\theta}_1$. This can be done by applying the following inverse-kinematic equation

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}^{+}\dot{\mathbf{x}} + \left(\sum_{i=1}^{n-m} \hat{\mathbf{n}}_{J_i} \cdot \hat{\mathbf{n}}_{J_i}^{\mathsf{T}} \right) \mathbf{z} \tag{3.18}$$

where $\boldsymbol{J}^{+}$ is the n by m pseudoinverse of the manipulator Jacobian, the m-dimensional vector $\dot{\mathbf{x}}$ is the end-effector velocity that drives the robot to $\mathbf{c}_g$ and maintains it there, the $n-m$ vectors $\hat{\mathbf{n}}_{J_i}$ represent a basis for the $n-m$-dimensional null space of $\boldsymbol{J}$, and n-dimensional vector $\mathbf{z}$ represents the desired joint velocity to achieve the secondary goals.

One can start with a random configuration in $\boldsymbol{\mathcal{C}'_A}$ and then use (3.18) to drive the manipulator to $\mathbf{c}_g$ while maintaining the joints within $\mathcal{C}'_A$ by using the method described in [43]. Once the end effector reaches $\mathbf{c}_g$ with $\boldsymbol{\theta} \in \mathcal{C}'_A$, one can determine the extremal values of θ_1 by setting the first component in vector $\mathbf{z}$ to $z_1 = \pm 1$. By projecting this modified version of $\mathbf{z}$ onto the null space of $\boldsymbol{J}$, this allows one to increase (or decrease) the value of θ_1 while staying on the SMM and satisfying any hard joint constraints to stay within $\mathcal{C}'_A$.

This process terminates when one of three conditions occur. First, there may be no constraints on θ_1, i.e., it can rotate 2π while staying within $\mathcal{C}'_A$ and keeping the end effector at $\mathbf{c}_g$. Second, the limits of the SMM are reached, i.e., one is at a critical point and the null space has no component of θ_1. Finally, one may reach a point where there is no further motion of θ_1 that does not violate one or more constraints of $\boldsymbol{\theta} \in \boldsymbol{\mathcal{C}'_A}$. For high-dimensional SMMs, i.e, robots with high DOR, these conditions may be encountered prior to a globally extremal value of θ_1, i.e., they are locally extremal values. However, for robots with a single DOR, one can always traverse the entire SMM by simply stepping along $\hat{\mathbf{n}}_J$, which is tangent to the SMM, to find the globally extremal values.

The Rotation Angle Range of C_i

The grid center $\mathbf{c}_g$ rotated by β, i.e., $\mathbf{c}'_g$, is guaranteed to be reachable after joint i is locked at an arbitrary angle, $\theta_i = q_i$, where $\underline{a}_i \leq q_i \leq \overline{a}_i$ and $i = \{1, 2, \ldots, n\}$, if $[\underline{a}_i,\ \overline{a}_i]$ is contained in the

projection of the pre-image of $\mathbf{c}'_g$ onto the ith axis, i.e., condition 2 is satisfied. The post-failure rotation angle range, $[\underline{\beta}_i,\ \overline{\beta}_i]$, of C_i can be determined by computing the critical points on a SMM for $\mathbf{c}_g$ that are indicated in blue in Fig. 3.2(b) and (c). Let the range of θ_i between the two critical points be denoted $[\theta_{i_{min}},\ \theta_{i_{max}}]$ and the union of all these ranges over all SMMs be denoted, Θ_i, i.e.,

$$\Theta_i = \bigcup_{\#\text{ of SMMs}} [\theta_{i_{min}},\ \theta_{i_{max}}]. \tag{3.19}$$

Let the range in Θ_i that contains A_i be denoted $[\Theta_{imin},\ \Theta_{imax}]$, i.e.,

$$[\Theta_{i_{min}},\ \Theta_{i_{max}}] \in \Theta_i \text{ and } A_i \subset [\Theta_{i_{min}},\ \Theta_{i_{max}}]. \tag{3.20}$$

First consider the computation of the post-failure rotation angle range of C_1, i.e., $[\underline{\beta}_1,\ \overline{\beta}_1]$. As described in Section 3.3.1, the pre-image of $\mathbf{c}_g$ is identical to that of $\mathbf{c}'_g$ for any β except for a β offset in θ_1. Therefore the range $[\underline{\beta}_1,\ \overline{\beta}_1]$ is determined when condition 2 is on the verge of being violated, i.e., A_1 becomes outside the range of $[\Theta_{1_{min}},\ \Theta_{1_{max}}]$, which is given by

$$\left[\underline{\beta}_1,\ \overline{\beta}_1\right] = \left[\overline{a}_1 - \Theta_{1_{max}},\ \underline{a}_1 - \Theta_{1_{min}}\right]. \tag{3.21}$$

Next, the rotation angle ranges, $[\underline{\beta}_j,\ \overline{\beta}_j]$, of C_j, where $j = \{2, 3, \ldots, n\}$, are computed as follows. The values of $\theta_{j_{min}}$ and $\theta_{j_{max}}$ in a SMM of $\mathbf{c}'_g$ are identical to those in $\mathbf{c}_g$. Therefore, once condition 2 is satisfied, i.e., $A_j \subset [\Theta_{j_{min}},\ \Theta_{j_{max}}]$, for $\mathbf{c}_g$, then the rotation angle range of C_j is given by

$$\left[\underline{\beta}_j,\ \overline{\beta}_j\right] = [-\pi,\ \pi] \tag{3.22}$$

because here it is assumed, without loss of generality, that there are no physical joint limits. As before, (3.18) is used to compute $\theta_{i_{min}}$ and $\theta_{i_{max}}$ where $\mathbf{z} = \pm\hat{\mathbf{e}}_i$ and $i = \{1, 2, \ldots, n\}$, i.e., the standard basis vectors.

3.4 Optimizing the Failure-Tolerant Workspace

The proposed technique for estimating the size of the failure-tolerant workspace, $\mathcal{W}_{\boldsymbol{F}}$, can be employed in an optimization method to maximize the size of $\mathcal{W}_{\boldsymbol{F}}$. This optimization results in the optimal set of artificial joint limits, denoted $\boldsymbol{A} = \{\underline{a}_1, \overline{a}_1, \underline{a}_2, \overline{a}_2, \ldots, \underline{a}_n, \overline{a}_n\}$, that should be imposed during the control of a kinematically redundant robot before the occurrence of an arbitrary single locked-joint failure and then released after that failure.

Let the unit of measure for the failure-tolerant workspace size be denoted $\mathcal{S}_F$ where $\mathcal{S}_F \in \mathbb{R}_{\geq 0}$. This general measure, $\mathcal{S}_F$, may represent any combination of linear and rotational components, e.g., the failure-tolerant workspace size for planar robots that perform a task defined as a 2D position or 3D position and orientation, and spatial robots that perform 3D position or 6D position and orientation tasks. In cases where $\mathcal{S}_F$ is a combination of different units, e.g., meters and rads for position and orientation, a suitable normalization factor should be employed to make sure that $\mathcal{S}_F$ is a meaningful measure.

In all cases, the value of $\mathcal{S}_F$ is rotationally invariant to θ_1. Therefore, the artificial limits of joint 1, $\underline{a}_1$ and $\overline{a}_1$, are not independent, so that only the difference between them affects the value of $\mathcal{S}_F$. Thus in the following optimization procedure the constraint of $\underline{a}_1 = -\overline{a}_1$ is imposed.

The optimization problem of maximizing $\mathcal{S}_F$ is solved using the coordinate ascent optimization method. Coordinate ascent search is an iterative process that attempts to increase the value of $\mathcal{S}_F$ by performing a one-dimensional search along one variable at a time, i.e., the $\underline{a}_i$s and $\overline{a}_i$s. Fortunately, $\mathcal{S}_F$ is periodic with a period of 2π in every $\underline{a}_i$ and $\overline{a}_i$ so the solution space is bounded. The algorithm converges to a local maximum when there is no change in any of the $\underline{a}_i$s or $\overline{a}_i$s that increases $\mathcal{S}_F$. The iterative process should be applied to multiple random initial $\boldsymbol{A}$s to find the global maximum of $\mathcal{S}_F$.

3.5 Examples

3.5.1 Overview

This section presents two examples of maximizing $\mathcal{S}_F$ by determining the optimal set of artificial joint limits, $\boldsymbol{A}$. Example 1 illustrates the case of planar robots with revolute joints that perform tasks in a 3D workspace (2D position and 1D orientation). Example 2 illustrates the case of spatial robots with revolute joints that perform tasks in a 6D workspace (3D position and 3D orientation). To perform a comparison between the results in both examples, $\mathcal{S}_F$ is normalized by the robot's $\mathcal{S}_{0_{\max}}$, which is the size of the original workspace (denoted $\boldsymbol{\mathcal{W}_{0_{max}}}$), of that robot when no artificial joint constraints are applied. Equation (3.11) is used to estimate $\mathcal{S}_{0_{\max}}$. For position-only workspace elements, (3.10) is evaluated with integration limits of $-\pi$ to π. For position and orientation workspace elements, (3.13) is evaluated with integration limits of $-\pi$ to π where N_F is replaced by the number of reachable orientations, denoted $N_{o_{\max}}$.

It is clear that the size of Δg and the number of randomly sampled orientations, N_o, in (3.10) and (3.13) affect the accuracy of estimating $\mathcal{S}_F$, as well as the computational cost. One would like to automatically determine an appropriate size for Δg based on the size of the robot. In the planar case, we set Δg to be 4% of the robot's maximum reach. This results in 25 samples for the length of the arm, so we also select $N_o = 25$ samples for the 1D orientation in the 3D planar case. In the 6D workspace case, Δg is the area of a square grid, whose length we select to also be 4% of the robot's maximum reach and $N_o = 200$ for the 3D orientation in the 6D workspaces.

3.5.2 Planar Robots

For the planar robot case, where the tasks are 2D position and 1D orientation, we analyze two kinematically redundant robots of 4 DOF and 5 DOF, i.e., DORs of 1 and 2. Both robots have equal link lengths, l_i, where $l_i = 1$ m and $i = \{1, 2, \ldots, n\}$. For the planar case, grid centers are selected on the discretized line segment from $x = 0$ to $x = n$ m. One can apply (3.13), after appropriately adjusting the integrand, to estimate the 3D volume element and then (3.11) is used to estimate $\mathcal{S}_F$, which in this case is the 3D failure-tolerant workspace volume. In (3.13), $\Delta g = 0.04n$ m is the

length of the line segment, i.e., 4% of the robot's maximum reach. The orientation range for the planar case is 2π so that π^2 in (3.13) is replaced by 2π. Note that the units of $\mathcal{S}_F$ and $\mathcal{S}_{0_{\max}}$ are m^2 rad, and that comparisons with other robot's workspaces should be in the same units.

After performing the coordinate ascent optimization method, the maximum $\mathcal{S}_F$ of the 4 DOF planar robot is approximately 9.5% of its $\mathcal{S}_{0_{\max}}$ as illustrated in Fig. 3.3. The three-dimensional workspace is given in (a) with the orientation angle shown on an axis that is orthogonal to the (x, y) position. In addition, color is used to show the percentage of total orientation angles that are reachable at that (x, y) position. Fig. 3.3(b) shows the top view of the workspaces for $\boldsymbol{\mathcal{W}_F}$ and $\boldsymbol{\mathcal{W}_{0_{\max}}}$ in order to more clearly see which (x, y) positions are reachable for at least one orientation, where once again, the color indicates the percentage of total orientations that are reachable at that position. The same robot has $\mathcal{S}_F = 0$ when no artificial joint limits are applied. By increasing the DOR to 2, i.e., a 5 DOF planar robot, the maximized $\mathcal{S}_F$ has increased dramatically to 31% as shown in Fig. 3.4. One impact of increasing the DOR to 2 is that there is now a region where the robot can reach 100% of the orientations, i.e., the yellow region, as compared to a maximum of 47% of the orientations in Fig. 3.3.

3.5.3 Spatial Robot

To compute the maximum $\mathcal{S}_F$ for the case of spatial robots where the task is 3D position and 3D orientation, two spatial robots of 7 DOF and 8 DOF, i.e., spatial robots with DORs of 1 and 2, are analyzed. The kinematic parameters of both robots are given in Tables 3.1 and 3.2 where α_i is the link twist, l_i is the link length, and d_i is the link displacement. The fourth kinematic parameter is the joint angle, θ_i, that is bounded by the lower, $\underline{a}_i$, and upper, $\overline{a}_i$, artificial limits that are the decision variables of the optimization. These two robots are designed to be failure-tolerant robots based on local measures, i.e., an optimally fault-tolerant Jacobian [44, 45].

We performed the coordinate ascent optimization on the 7 DOF spatial robot and obtained a maximum $\mathcal{S}_F$ of 6.5% of its $\mathcal{S}_{0_{\max}}$. The optimal $\boldsymbol{A}$ is given in Table 3.1 and the workspaces are shown in Fig.3.5. This robot would only reach 1% of its $\mathcal{S}_{0_{\max}}$ if artificial joint limits were not

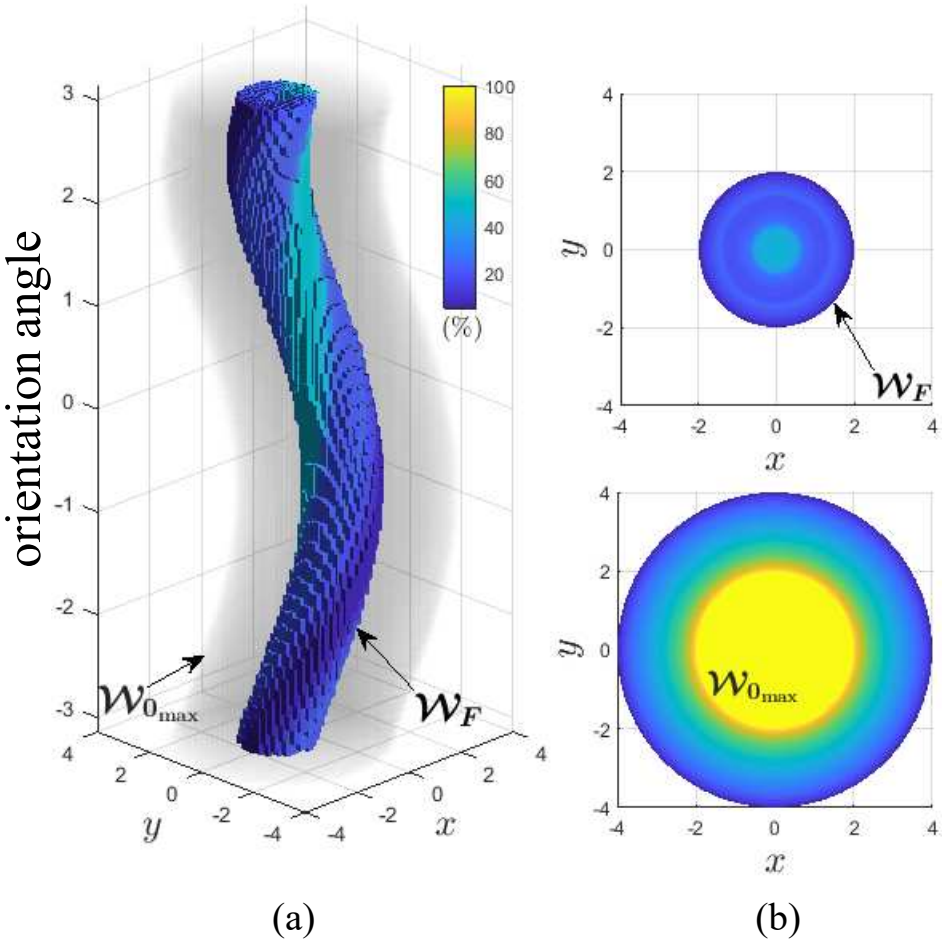


Figure 3.3: This figure shows the 3D view (a) and the top view (b) of $\mathcal{W}_F$ and $\mathcal{W}_{0_{\max}}$ for an equal link length 4 DOF planar robot performing a task in a 3D planar workspace, i.e., orientation along with 2D position. The maximum $\mathcal{S}_F$ of the 4 DOF planar robot is 16.5 m^2 rad at the optimal set of artificial joint limits, $\boldsymbol{A} = \{-180°, 180°, 90°, 143°, 90°, 143° -180°, 180°\}$ where the original workspace size is $\mathcal{S}_{0_{\max}} = 178$ m^2 rad.

used. By increasing the DOR to 2, i.e., an 8 DOF spatial robot, the algorithm converged to the optimal $\boldsymbol{A}$ that is given in Table 3.2. The maximum $\mathcal{S}_F$ of the 8 DOF robot has increased to 21% of its $\mathcal{S}_{0_{\max}}$ as shown in Fig. 3.6.

For the 6D workspaces, the units for $\mathcal{S}_F$ and $\mathcal{S}_{0_{\max}}$ are m^3 rad^3. The two figures, 3.5 and 3.6, show the 6D failure-tolerant workspace, $\mathcal{W}_F$, and the 6D original robot workspace, $\mathcal{W}_{0_{\max}}$, using orthogonal cross sections to show more of the internal structure of the workspaces. In a manner analogous to Figs. 3.3 and 3.4, we use color to show the percentage of the orientations that are reachable at that (x, y, z) position, however, in this case the range of all possible orientations

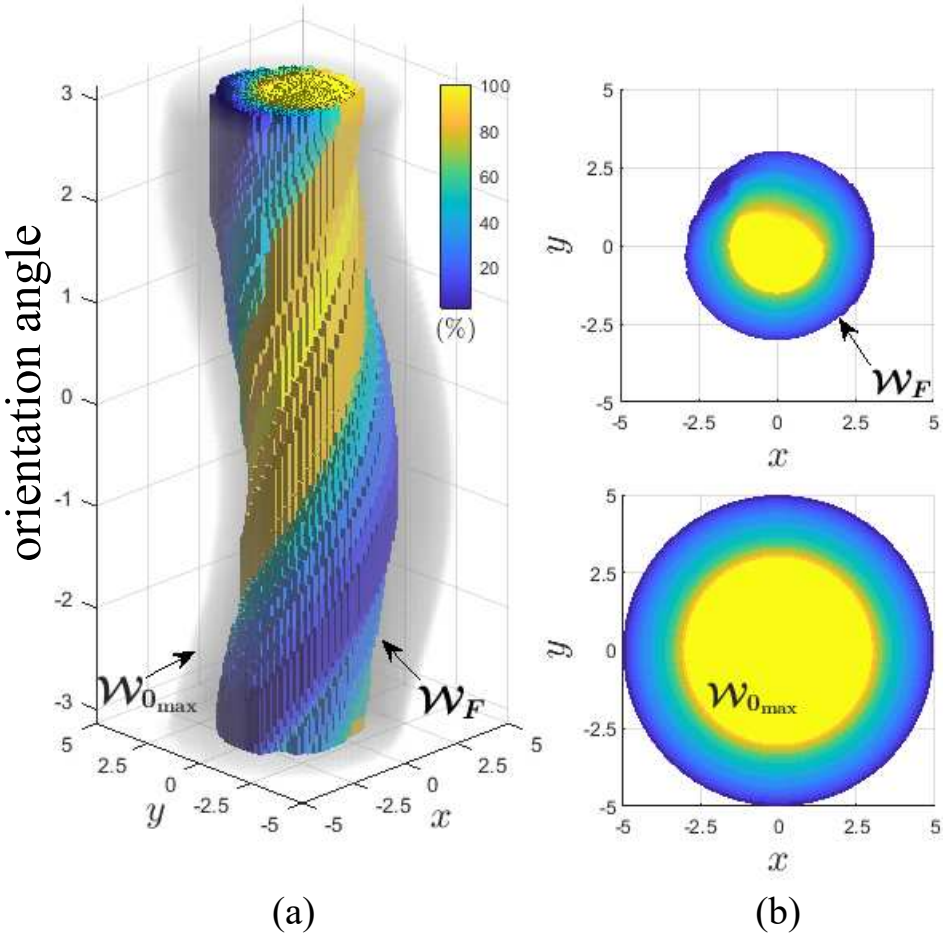


Figure 3.4: This figure shows the 3D view (a) and the top view (b) of $\mathcal{W}_F$ and $\mathcal{W}_{0_{\max}}$ for an equal link length 5 DOF planar robot performing a task in a 3D planar workspace, i.e., orientation along with 2D position. The maximum $\mathcal{S}_F$ of the 5 DOF planar robot is 98 m^2 rad where $\mathcal{S}_{0max} = 317$ m^2 rad at the optimal set of artificial joint limits, $\boldsymbol{A} = \{-111°, 111°, -150°, 25°, -150°, 25°, -150°, 150°, 130°, 275°\}$. Note that $\mathcal{W}_F$ in (b) does not consist of concentric circles, as in Fig. 3.3, because the artificial limits of joint 1 are not $[-180°, 180°]$.

is a three-dimensional volume. These percentages are estimated using the ratio N_F/N_o for $\mathcal{W}_F$ and $N_{o_{\max}}/N_o$ for $\mathcal{W}_{0_{\max}}$. The outer gray boundary that surrounds $\mathcal{W}_F$ is the outer boundary of $\mathcal{W}_{0_{\max}}$.

3.5.4 Evaluation

The technique presented here was compared to the current best approach, which uses a numerical technique to determine the boundary curves of $\mathcal{W}_F$ along with Green's theorem to compute the area inside these boundaries [1]. Because the approach in [1] is only applicable to 3 DOF

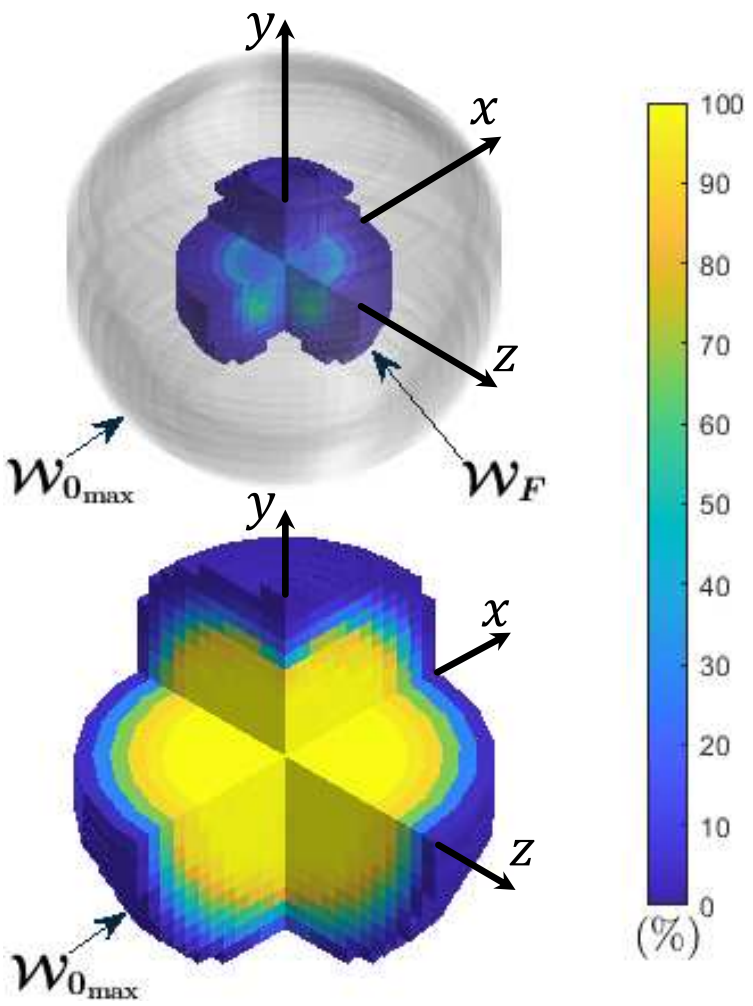


Figure 3.5: This figure shows the maximum $\mathcal{S}_F$ of a 7 DOF robot (kinematic parameters are given in Table 3.1) performing tasks in a 6D workspace. The maximum $\mathcal{S}_F = 892$ m^3 rad^3 is at the optimal $\boldsymbol{A}$ (given in Table 3.1) with $\mathcal{S}_{0_{\max}} = 13{,}274$ m^3 rad^3.

Table 3.1: The kinematic parameters of the 7 DOF spatial robot

$Link_i$	α_i[degrees]	l_i[meters]	d_i[meters]	θ_i[degrees]	
				$\underline{a}_i$	$\overline{a}_i$
1	-98°	0.17	0	-74°	74°
2	-114°	1.42	1.67	-180°	180°
3	-66°	1.42	-0.69	-180°	180°
4	50°	0.56	-1.77	-113°	86°
5	-92°	1.32	2.42	-165°	146°
6	-93°	1.27	-0.38	-64°	98°
7	0°	1	0.95	-180°	180°

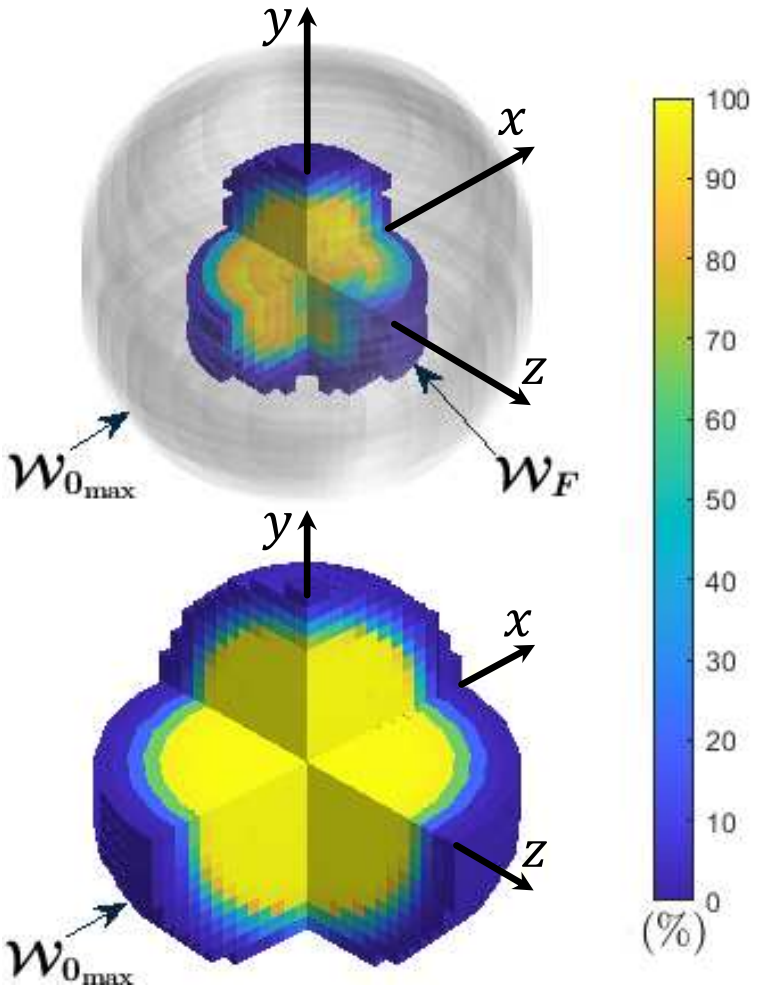


Figure 3.6: This figure shows the maximum $\mathcal{S}_F$ of an 8 DOF robots (kinematic parameters are given in Table 3.2) performing tasks in a 6D workspace. The maximum $\mathcal{S}_F$ = 4,450 m^3 rad^3 is at the optimal $\boldsymbol{A}$ (given in Table 3.2) with $\mathcal{S}_{0_{\max}}$ = 20,920 m^3 rad^3.

Table 3.2: The kinematic parameters of the 8 DOF spatial robot

$Link_i$	α_i[degrees]	l_i[meters]	d_i[meters]	θ_i[degrees]	
				$\underline{a}_i$	$\overline{a}_i$
1	71°	-0.89	0	-60°	60°
2	16°	1.41	-2.52	-180°	180°
3	63°	-1.17	2.35	-180°	180°
4	125°	-1.46	1.09	-3°	340°
5	82°	-0.66	-1.83	-180°	180°
6	46°	1.56	0.61	-204°	58°
7	108°	-0.67	1.56	-180°	180°
8	0°	1	1.21	170°	346°

Table 3.3: Accuracy and computational time comparison.

Method	Number of samples	Accuracy (%)	Time [min]
Uniform sampling	$\pi\ 100^2$	99.3%	361
Random sampling	$\pi\ 100^2$	99.5%	368
Hybrid approach	100	99.7%	7

planar robots with 1 DOR, we use the results for a robot with three links of equal length that has an optimal $\boldsymbol{\mathcal{W}_F}$ area, $\mathcal{S}_F$, equal to 3.56 m^2. The computation time required by this approach was seven minutes, where 99% of that time was spent determining the boundary curves.

In Table 3.3, we compare the accuracy and computation time for uniform and random sampling along with our hybrid approach to the exact result of 3.56 m^2. It is clear that the hybrid approach presented here is able to achieve accuracy that is comparable to the exact approach in the same amount of time, and has the advantage of being applicable to any robot or any workspace dimension. In addition, one can see that both of the sampling approaches suffer from exponential growth, i.e., 100^m where m is the dimension of the workspace, so they are infeasible for a three-dimensional workspace that considers both position and orientation.

In Tables 3.4 and 3.5, we illustrate the computational expense of our approach as a function of sample resolution, dimension of workspace (m), and a robot's DOF (n). From the first row of Table 3.4 one can see that the computation time increase linearly with the number of sample points, i.e., increasing the sample resolution by four from 25 to 100 results in a four-fold increase in the computation time. The relationships between m and n and the computation time are much more complicated and interrelated. In general, the computation time grows slowly with an increase in n for a given m, i.e., going up by a factor of 2 or 3. This means that computing $\mathcal{S}_F$ for robots with higher DORs is computationally feasible. The increase in computational expense is more affected by an increase in m, but less than an order of magnitude per additional dimension. It should also be noted that the variation in computation time for a given m and n is significantly affected by the resulting size of $\mathcal{S}_F$, e.g., see rows two and five of Table 3.4.

Table 3.4: Execution time for position workspaces

Workspace dimension $[m]$	DOF $[n]$	Position samples	Normalized $\mathcal{S}_F$	Time [min]
2D	3	25 (100)	12.6%	1.8 (7)
2D	4	100	26% (43%)	13.5 (17.5)
2D	5	100	52%	23.5
3D	4	1250	19%	28
3D	5	1250	26% (78%)	48 (105)

Table 3.5: Execution time for position and orientation workspaces

Workspace dimension $[m]$	DOF $[n]$	Position samples	Orientation samples	Normalized $\mathcal{S}_F$	Time [min]
3D	4	100	100	9.5%	150
3D	5	100	100	31%	420
6D	7	1250	200	6.5%	2250
6D	8	1250	200	21%	5650

3.6 Conclusion

This work presents a general hybrid technique for estimating the failure-tolerant workspace size of any serial robot with arbitrary kinematics. The method presented combines an algorithm for computing self-motion manifold ranges to estimate workspace envelopes and Monte-Carlo integration to estimate orientation volumes to create a computationally efficient algorithm that can be applied to high degree-of-freedom robots with any degree of redundancy. Because the proposed algorithm is computationally tractable, it can be combined with an optimization technique, like coordinate ascent, to determine optimal artificial joint limits that maximize the size of the failure-tolerant workspace of a given kinematically redundant robot. Therefore, researchers can evaluate or modify an existing design to make the robot more fault tolerant. This approach is illustrated on multiple examples of robots that perform tasks in workspaces that are 3D planar (2D position and 1D orientation) and 6D spatial (3D position and 3D orientation). It is shown how an increase in the degree of redundancy from one to two can dramatically increase the size of the failure-tolerant workspace by more than a factor of three.

Chapter 4

Conclusions

4.1 Summary

This dissertation presented a technique for maximizing the area of a failure-tolerant workspace after an arbitrary single locked-joint failure for planar 3R robots of arbitrary link lengths. The technique is based on the gradient ascent method and can be used to determine the optimal artificial joint limits for an existing robot, or to design a robot with optimal link lengths and artificial joint limits. It was shown how to deal with the numerical issues associated with gradient-based techniques due to discontinuities of the gradient caused by changes in the boundary curves of the failure-tolerant workspace. The technique was illustrated on two examples and resulted in a more accurate computation of a previous result, as well the identification of a previously unknown optimal failure-tolerant workspace area.

This dissertation also addressed the limitations of conventional methods for computing the size of failure-tolerant workspaces. This work resulted in a general hybrid technique for estimating the failure-tolerant workspace size of any serial robot with arbitrary kinematics. The method presented combines an algorithm for computing self-motion manifold ranges to estimate workspace envelopes and Monte-Carlo integration to estimate orientation volumes to create a computationally efficient algorithm that can be applied to high degree-of-freedom robots with any degree of redundancy. Because the proposed algorithm is computationally tractable, it can be combined with an optimization technique, like coordinate ascent, to determine optimal artificial joint limits that maximize the size of the failure-tolerant workspace of a given kinematically redundant robot. Therefore, researchers can evaluate or modify an existing design to make the robot more fault tolerant. This approach was illustrated on multiple examples of robots that perform tasks in workspaces that are 3D planar (2D position and 1D orientation) and 6D spatial (3D position and

3D orientation). It was shown how an increase in the degree of redundancy from one to two can dramatically increase the size of the failure-tolerant workspace by more than a factor of three.

4.2 Future Directions

The work presented in this dissertation can be extended to design optimal failure-tolerant robots with higher degrees-of-freedom that perform tasks in 6D workspaces. This would require not only the artificial joint limits being variables but also the DH parameters of the robot. Also, this study has focused on rotary joints whereas it would be interesting to achieve optimal results for robots with prismatic joints as well. Finally, considering multiple simultaneous locked joint failures would be an interesting research extension.

Bibliography

[1] Ashraf M Bader and Anthony A Maciejewski. Maximizing the failure-tolerant workspace area for planar redundant robots. *Mechanism and Machine Theory*, 143:103635, 2020.

[2] Ashraf M Bader and Anthony A Maciejewski. A hybrid approach for estimating the failure-tolerant workspace size of kinematically redundant robots. *IEEE Robotics and Automation Letters*, 6(2):303–310, 2020.

[3] Janko Petereit, Jürgen Beyerer, Tamim Asfour, Sascha Gentes, Björn Hein, Uwe D Hanebeck, Frank Kirchner, Rüdiger Dillmann, Hans Heinrich Götting, Martin Weiser, et al. Robdekon: Robotic systems for decontamination in hazardous environments. In *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 249–255, 2019.

[4] James P Trevelyan, Sung-Chul Kang, and William R Hamel. *Robotics in hazardous applications*. Springer, 2008.

[5] Keiji Nagatani, Seiga Kiribayashi, Yoshito Okada, Kazuki Otake, Kazuya Yoshida, Satoshi Tadokoro, Takeshi Nishimura, Tomoaki Yoshida, Eiji Koyanagi, Mineo Fukushima, and S Kawatsuma. Emergency response to the nuclear accident at the Fukushima Daiichi Nuclear Power Plants using mobile rescue robots. *Journal of Field Robotics*, 30(1):44–63, 2013.

[6] Roberto Guzman, Roman Navarro, Juan Ferre, and Miguel Moreno. Rescuer: Development of a modular chemical, biological, radiological, and nuclear robot for intervention, sampling, and situation awareness. *Journal of Field Robotics*, 33(7):931–945, 2016.

[7] Giovanni Muscato, Filippo Bonaccorso, Luciano Cantelli, Domenico Longo, and C Donato Melita. Volcanic environments: Robots for exploration and measurement. *IEEE Robotics & Automation Magazine*, 19(1):40–49, 2012.

[8] FL Ni, MH Jin, HL Wang, H Liu, and G Hirzinger. Joint fault-tolerant design of the chinese space robotic arm. In *IEEE International Conference on Information Acquisition*, pages 528–533, 2006.

[9] Zonggao Mu, Liang Han, Wenfu Xu, Bing Li, and Bin Liang. Kinematic analysis and fault-tolerant trajectory planning of space manipulator under a single joint failure. *Robotics and Biomimetics*, 3(1):16, 2016.

[10] Serdar Soylu, Bradley J Buckham, and Ron P Podhorodeski. Redundancy resolution for underwater mobile manipulators. *Ocean Engineering*, 37(2-3):325–343, 2010.

[11] BM Harpel, Joanne Bechta Dugan, Ian D Walker, and Joseph R Cavallaro. Analysis of robots for hazardous environments. *Annual Reliability and Maintainability Symposium.*, pages 111–116, 1997.

[12] Thomas A Ferguson and Lixuan Lu. Fault tree analysis for an inspection robot in a nuclear power plant. In *IOP Conference Series: Materials Science and Engineering*, volume 235, page 012003, 2017.

[13] Fabrizio Caccavale, Alessandro Marino, Giuseppe Muscio, and Francesco Pierri. Discrete-time framework for fault diagnosis in robotic manipulators. *IEEE Transactions on Control Systems Technology*, 21(5):1858–1873, 2013.

[14] Fengyu Xu, Zhong Yang, JinLong Hu, Guoping Jiang, and GuoHong Dai. Fault diagnosis of a selective compliance assembly robot arm manipulator based on the end joint motion analysis: Threshold algorithm and experiments. *Transactions of the Institute of Measurement and Control*, 40(5):1691–1700, 2018.

[15] Amaneh Salmani Rezazadeh, Hamid Reza Koofigar, and Saeed Hosseinnia. Adaptive fault detection and isolation for a class of robot manipulators with time-varying perturbation. *Journal of Mechanical Science and Technology*, 29(11):4901–4911, 2015.

[16] Michael L McIntyre, Warren E Dixon, Darren M Dawson, and Ian D Walker. Fault identification for robot manipulators. *IEEE Transactions on Robotics*, 21(5):1028–1034, 2005.

[17] Haitao Chang, Panfeng Huang, Ming Wang, and Zhenyu Lu. Locked-joint failure identification for free-floating space robots. In *IEEE International Conference on Information and Automation (ICIA)*, pages 170–175, 2014.

[18] CHEN Gang, GUO Wen, JIA Qingxuan, WANG Xuan, and FU Yingzhuo. Failure treatment strategy and fault-tolerant path planning of a space manipulator with free-swinging joint failure. *Chinese Journal of Aeronautics*, 31(12):2290–2305, 2018.

[19] James D English and Anthony A Maciejewski. Failure tolerance through active braking: A kinematic approach. *The International Journal of Robotics Research*, 20(4):287–299, 2001.

[20] Frank L Hammond III. Synthesis of k th order fault-tolerant kinematically redundant manipulator designs using relative kinematic isotropy. *International Journal of Adaptive and Innovative Systems*, 2(1):73–96, 2014.

[21] Chinmay S Ukidve, John E McInroy, and Farhad Jafari. Using redundancy to optimize manipulability of Stewart platforms. *IEEE/ASME Transactions on Mechatronics*, 13(4):475–479, 2008.

[22] Khaled M Ben-Gharbia, Anthony A Maciejewski, and Rodney G Roberts. Kinematic design of manipulators with seven revolute joints optimized for fault tolerance. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(10):1364–1373, 2016.

[23] Yu She, Wenfu Xu, Haijun Su, Bin Liang, and Hongliang Shi. Fault-tolerant analysis and control of SSRMS-type manipulators with single-joint failure. *Acta Astronautica*, 120:270–286, 2016.

[24] Ahmad A Almarkhi and Anthony A Maciejewski. Maximizing the size of self-motion manifolds to improve robot fault tolerance. *IEEE Robotics and Automation Letters*, 4(3):2653–2660, 2019.

[25] Christiaan JJ Paredis, WK Frederick Au, and Pradeep K Khosla. Kinematic design of fault tolerant manipulators. *Computers & electrical engineering*, 20(3):211–220, 1994.

[26] Randy C Hoover, Rodney G Roberts, Anthony A Maciejewski, Priya S Naik, and Khaled M Ben-Gharbia. Designing a failure-tolerant workspace for kinematically redundant robots. *IEEE Transactions on Automation Science and Engineering*, 12(4):1421–1432, 2015.

[27] Christiaan JJ Paredis and Pradeep K Khosla. Designing fault-tolerant manipulators: How many degrees of freedom? *The International Journal of Robotics Research*, 15(6):611–628, 1996.

[28] Christopher L Lewis and Anthony A Maciejewski. Fault tolerant operation of kinematically redundant manipulators for locked joint failures. *IEEE Transactions on Robotics and Automation*, 13(4):622–629, 1997.

[29] Karim Abdel-Malek, Frederick Adkins, Harn-Jou Yeh, and Edward Haug. On the determination of boundaries to manipulator workspaces. *Robotics and Computer-Integrated Manufacturing*, 13(1):63–72, 1997.

[30] Yi Cao, Ke Lu, Xiujuan Li, and Yi Zang. Accurate numerical methods for computing 2d and 3d robot workspace. *International Journal of Advanced Robotic Systems*, 8(6):76, 2011.

[31] Lars I Hatledal, Filippo Sanfilippo, Yingguang Chu, and Houxiang Zhang. A voxel-based numerical method for computing and visualising the workspace of offshore cranes. In *International Conference on Offshore Mechanics and Arctic Engineering*, volume 56475, page V001T01A012. American Society of Mechanical Engineers, 2015.

[32] Zhiyuan Zhao, Shuai He, Yaping Zhao, Ce Xu, Qingwen Wu, and Zhenbang Xu. Workspace analysis for a 9-dof hyper-redundant manipulator based on an improved monte carlo method and voxel algorithm. In *IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 637–642, 2018.

[33] Rodrigo S Jamisola. Optimization of failure-tolerant workspaces for redundant manipulators. *Philippine Science Letters*, 3:66–75, 2010.

[34] Christian Ducros, Gérard Hauser, Najib Mahjoubi, Philippe Girones, Laurence Boisset, Antoine Sorin, Eric Jonquet, Jean Michel Falciola, and Albert Benhamou. Rica: A tracked robot for sampling and radiological characterization in the nuclear field. *Journal of Field Robotics*, 34(3):583–599, 2017.

[35] Mien Van, Shuzhi Sam Ge, and Hongliang Ren. Finite time fault tolerant control for robot manipulators using time delay estimation and continuous nonsingular fast terminal sliding mode control. *IEEE Transactions on Cybernetics*, 47(7):1681–1693, 2017.

[36] Monica L Visinsky, Joseph R Cavallaro, and Ian D Walker. Robot fault detection and fault tolerance: A survey. *Reliability Engineering and System Safety*, 46(2):139–158, 1994.

[37] Anthony A Maciejewski. Fault tolerant properties of kinematically redundant manipulators. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 638–642, 1990.

[38] Rodrigo S Jamisola, Anthony A Maciejewski, and Rodney G Roberts. Failure-tolerant path planning for kinematically redundant manipulators anticipating locked-joint failures. *IEEE Transactions on Robotics*, 22(4):603–612, 2006.

[39] Hamid Abdi and Saeid Nahavandi. Task completion with partially-failed manipulators. In *IEEE Conference on Robotics Automation and Mechatronics*, pages 269–274, 2010.

[40] Edwin KP Chong and Stanislaw H Zak. *An introduction to optimization*, volume 76. John Wiley & Sons, 2013.

[41] Nikolaus Vahrenkamp and Tamim Asfour. Representing the robot's workspace through constrained manipulability analysis. *Autonomous Robots*, 38(1):17–30, 2015.

[42] Chao Chen and Daniel Jackson. Parameterization and evaluation of robotic orientation workspace: A geometric treatment. *IEEE Transactions on Robotics*, 27(4):656–663, 2011.

[43] Fabrizio Flacco, Alessandro De Luca, and Oussama Khatib. Control of redundant robots under hard joint constraints: Saturation in the null space. *IEEE Transactions on Robotics*, 31(3):637–654, 2015.

[44] Khaled M Ben-Gharbia, Anthony A Maciejewski, and Rodney G Roberts. An example of a seven joint manipulator optimized for kinematic fault tolerance. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 802–807, 2014.

[45] Khaled M Ben-Gharbia, Anthony A Maciejewski, and Rodney G Roberts. A kinematic analysis and evaluation of planar robots designed from optimally fault-tolerant Jacobians. *IEEE Transactions on Robotics*, 30(2):516–524, 2014.

[46] Daniela Bittnerova. Alternative method for calculations of volumes by using parameterizations surfaces areas. In *AIP Conference Proceedings*, volume 1570, pages 3–10, 2013.

Appendix A

A.1 Extension to Spatial Positioning 4R Robots

The technique presented in Section 4 can be extended to the case of 4R spatial positioning robots to maximize the failure-tolerant workspace volume $\mathcal{V}_{\mathcal{W}_F}$. Green's theorem is a special case of the more general Gauss-Ostrogradsky's theorem that can be applied to compute the volume under surfaces [46]. In this case, the boundary surfaces are all portions of a torus. Let surface k be a boundary of $\boldsymbol{\mathcal{W}_F}$ with coordinates

$$\begin{aligned} x_1 &= f_1({}^k\theta_i, {}^k\theta_j) \\ x_2 &= f_2({}^k\theta_i, {}^k\theta_j) \\ x_3 &= f_3({}^k\theta_i, {}^k\theta_j) \end{aligned} \tag{A.1}$$

where ${}^k\theta_i$ and ${}^k\theta_j$ are the two variable joints i and j for the surface k and the remaining two joints are at constant joint angles. The volume under surface k can be computed using

$$\mathcal{V}_{\mathcal{W}_F} = \frac{1}{3} \int_{{}^k\theta_i} \int_{{}^k\theta_j} \left(x_1 \, dx_2 \, dx_3 + x_2 \, dx_1 \, dx_3 + x_3 \, dx_1 \, dx_2 \right). \tag{A.2}$$

The failure-tolerant workspace volume is the summation of the volumes under the boundary surfaces

$$\mathcal{V}_{\mathcal{W}_F} = \sum_{k=1}^{M} \mathcal{V}_k, \tag{A.3}$$

where M is the number of boundary surfaces for $\boldsymbol{\mathcal{W}_F}$. Once the equations for $\mathcal{V}_{\mathcal{W}_F}$ are computed, then the gradient can be determined and one can apply (2.13) to maximize the volume.

A.2 Symbolic Equation for $\mathcal{A}_1$

$\mathcal{A}_1 = 2\pi$ - $2\cos^{-1}(\cos(\underline{a}_3/2)\cos(\overline{a}_1) - \cos(\overline{a}_1)/2 + \sin(\overline{a}_1)\,(\cos(\underline{a}_3/2)/2 - \cos(2\,\underline{a}_3)/8 - \cos((3\,\underline{a}_3)/2)/2 + 1/8)^{(1/2)} + (\cos(\underline{a}_3)\cos(\overline{a}_1))/2) - 2\cos^{-1}((78\cos(2\,\overline{a}_1) + \cos(2\,\overline{a}_2) + 2\cos(4\,\overline{a}_1) + 2$

$$\cos(3\,\overline{a}_2) - 2\cos(2\,\overline{a}_1 - \overline{a}_2) - \cos(2\,\overline{a}_1 - 2\,\overline{a}_2) - 11\cos(2\,\overline{a}_1 + 3\,\overline{a}_2) - \cos(2\,\overline{a}_1 + 4\,\overline{a}_2) + 18\cos(4\,\overline{a}_1 + 2\,\overline{a}_2) - \cos(6\,\overline{a}_1 + 2\,\overline{a}_2) - 121\cos(\overline{a}_2) + 96\cos(2\,\overline{a}_1 + \overline{a}_2) + 72\sin(\overline{a}_1)\sin(\overline{a}_1 - \overline{a}_2) + 8\sin(\overline{a}_1)\sin(\overline{a}_1 - 2\,\overline{a}_2) + 44\sin(\overline{a}_1)\sin(\overline{a}_1 + 2\,\overline{a}_2) - 20\sin(3\,\overline{a}_1)\sin(\overline{a}_1 + \overline{a}_2) + 4\sin(\overline{a}_1)\sin(\overline{a}_1 + 3\,\overline{a}_2) - 72\sin(\overline{a}_1)\sin(3\,\overline{a}_1 + \overline{a}_2) + 4\sin(\overline{a}_1)\sin(5\,\overline{a}_1 + \overline{a}_2) - 22\sin(3\,\overline{a}_1 + 2\,\overline{a}_2)\sin(\overline{a}_1 + \overline{a}_2) - 4\sin(3\,\overline{a}_1 + 3\,\overline{a}_2)\sin(\overline{a}_1 + \overline{a}_2) + 4\sin(5\,\overline{a}_1 + 2\,\overline{a}_2)\sin(\overline{a}_1 + \overline{a}_2) + 2\sin(5\,\overline{a}_1 + 3\,\overline{a}_2)\sin(\overline{a}_1 + \overline{a}_2) - 80\cos(\overline{a}_2)^2 - 16\cos(\overline{a}_2)^3 + 100\sin(\overline{a}_1)^2 - 36\sin(\overline{a}_2)^2 + 160\sin(\overline{a}_1)^4 - 32\cos(2\,\overline{a}_1 + \overline{a}_2)^2 + 112\cos(2\,\overline{a}_1)\cos(\overline{a}_2) - 64\cos(2\,\overline{a}_1)\cos(2\,\overline{a}_1 + \overline{a}_2) - 16\cos(\overline{a}_2)\cos(2\,\overline{a}_1 + \overline{a}_2)^2 + 32\cos(\overline{a}_2)^2\cos(2\,\overline{a}_1 + \overline{a}_2) - 4\sin(3\,\overline{a}_1 - \overline{a}_2)\sin(\overline{a}_1) - 44\sin(3\,\overline{a}_1 + 2\,\overline{a}_2)\sin(\overline{a}_1) - 8\sin(3\,\overline{a}_1 + 3\,\overline{a}_2)\sin(\overline{a}_1) + 8\sin(5\,\overline{a}_1 + 2\,\overline{a}_2)\sin(\overline{a}_1) + 4\sin(5\,\overline{a}_1 + 3\,\overline{a}_2)\sin(\overline{a}_1) - 32\cos(2\,\overline{a}_1)^2 + 36\cos(\overline{a}_1 + \overline{a}_2)\cos(\overline{a}_1) + 32\cos(2\,\overline{a}_1)\cos(\overline{a}_2)^2 - 16\cos(2\,\overline{a}_1)^2\cos(\overline{a}_2) + 244\sin(\overline{a}_1 + \overline{a}_2)\sin(\overline{a}_1) + 76\sin(\overline{a}_1 + \overline{a}_2)^2 + 112\cos(\overline{a}_2)\cos(2\,\overline{a}_1 + \overline{a}_2) - 24\cos(\overline{a}_1 + \overline{a}_2)\sin(\overline{a}_1)\,(-(\cos(2\,\overline{a}_1) - \cos(\overline{a}_2) + \cos(2\,\overline{a}_1 + \overline{a}_2) - 1)\,(\cos(2\,\overline{a}_1) - \cos(\overline{a}_2) + \cos(2\,\overline{a}_1 + \overline{a}_2) + 3))^{(1/2)} + 4\sin(\overline{a}_1 + \overline{a}_2)\cos(\overline{a}_1 - \overline{a}_2)\,(-(\cos(2\,\overline{a}_1) - \cos(\overline{a}_2) + \cos(2\,\overline{a}_1 + \overline{a}_2) - 1)\,(\cos(2\,\overline{a}_1) - \cos(\overline{a}_2) + \cos(2\,\overline{a}_1 + \overline{a}_2) + 3))^{(1/2)} - 4\sin(\overline{a}_1 + \overline{a}_2)\cos(\overline{a}_1 + 2\,\overline{a}_2)\,(-(\cos(2\,\overline{a}_1) - \cos(\overline{a}_2) + \cos(2\,\overline{a}_1 + \overline{a}_2) - 1)\,(\cos(2\,\overline{a}_1) - \cos(\overline{a}_2) + \cos(2\,\overline{a}_1 + \overline{a}_2) + 3))^{(1/2)} + 4\sin(\overline{a}_1 + \overline{a}_2)\cos(3\,\overline{a}_1 + \overline{a}_2)\,(-(\cos(2\,\overline{a}_1) - \cos(\overline{a}_2) + \cos(2\,\overline{a}_1 + \overline{a}_2) - 1)\,(\cos(2\,\overline{a}_1) - \cos(\overline{a}_2) + \cos(2\,\overline{a}_1 + \overline{a}_2) + 3))^{(1/2)} + 48\cos(\overline{a}_1 + \overline{a}_2)\cos(\overline{a}_1)\cos(\overline{a}_2) - 48\cos(\overline{a}_1 + \overline{a}_2)\cos(\overline{a}_1)\cos(2\,\overline{a}_1 + \overline{a}_2) + 8\sin(\overline{a}_1)\cos(\overline{a}_1 - \overline{a}_2)\,(-(\cos(2\,\overline{a}_1) - \cos(\overline{a}_2) + \cos(2\,\overline{a}_1 + \overline{a}_2) - 1)\,(\cos(2\,\overline{a}_1) - \cos(\overline{a}_2) + \cos(2\,\overline{a}_1 + \overline{a}_2) + 3))^{(1/2)} - 8\sin(\overline{a}_1)\cos(\overline{a}_1 + 2\,\overline{a}_2)\,(-(\cos(2\,\overline{a}_1) - \cos(\overline{a}_2) + \cos(2\,\overline{a}_1 + \overline{a}_2) - 1)\,(\cos(2\,\overline{a}_1) - \cos(\overline{a}_2) + \cos(2\,\overline{a}_1 + \overline{a}_2) + 3))^{(1/2)} + 8\sin(\overline{a}_1)\cos(3\,\overline{a}_1 + \overline{a}_2)\,(-(\cos(2\,\overline{a}_1) - \cos(\overline{a}_2) + \cos(2\,\overline{a}_1 + \overline{a}_2) - 1)\,(\cos(2\,\overline{a}_1) - \cos(\overline{a}_2) + \cos(2\,\overline{a}_1 + \overline{a}_2) + 3))^{(1/2)} + 4\cos(3\,\overline{a}_1 + 2\,\overline{a}_2)\sin(\overline{a}_1 + \overline{a}_2)\,(-(\cos(2\,\overline{a}_1) - \cos(\overline{a}_2) + \cos(2\,\overline{a}_1 + \overline{a}_2) - 1)\,(\cos(2\,\overline{a}_1) - \cos(\overline{a}_2) + \cos(2\,\overline{a}_1 + \overline{a}_2) + 3))^{(1/2)} - 48\cos(2\,\overline{a}_1)\cos(\overline{a}_1 + \overline{a}_2)\cos(\overline{a}_1) + 16\cos(\overline{a}_1 + \overline{a}_2)\cos(\overline{a}_1)\cos(\overline{a}_2)^2 + 16\cos(\overline{a}_1 + \overline{a}_2)\cos(\overline{a}_1)\cos(2\,\overline{a}_1 + \overline{a}_2)^2 - 12\cos(\overline{a}_1 + \overline{a}_2)\sin(\overline{a}_1 + \overline{a}_2)\,(-(\cos(2\,\overline{a}_1) - \cos(\overline{a}_2) + \cos(2\,\overline{a}_1 + \overline{a}_2) - 1)\,(\cos(2\,\overline{a}_1) - \cos(\overline{a}_2) + \cos(2\,\overline{a}_1 + \overline{a}_2) + 3))^{(1/2)} + 8\cos(3\,\overline{a}_1 + 2\,\overline{a}_2)\sin(\overline{a}_1)\,(-(\cos(2\,\overline{a}_1) - \cos(\overline{a}_2) + \cos(2\,\overline{a}_1 + \overline{a}_2) - 1)\,(\cos(2\,\overline{a}_1) - \cos(\overline{a}_2) + \cos(2\,\overline{a}_1 + \overline{a}_2) + 3))^{(1/2)} - 32\cos(2\,\overline{a}_1)\cos(\overline{a}_2)\cos(2\,\overline{a}_1 + \overline{a}_2) + 16\cos(2\,\overline{a}_1)^2\cos(\overline{a}_1 + \overline{a}_2)\cos(\overline{a}_1) - 32\cos(\overline{a}_1 + \overline{a}_2)\cos(\overline{a}_1)\cos(\overline{a}_2)\cos(2\,\overline{a}_1 + \overline{a}_2) - 32\cos(2\,\overline{a}_1)\cos(\overline{a}_1 + \overline{a}_2)\cos(\overline{a}_1)\cos(\overline{a}_2) + 32\cos(2\,\overline{a}_1)\cos(\overline{a}_1 + \overline{a}_2)\cos(\overline{a}_1)\cos(2\,\overline{a}_1 + \overline{a}_2) - 72)/(8\cos(\overline{a}_1 + \overline{a}_2)\,(2\cos(2$$

$\overline{a}_1)-2\cos(\overline{a}_2)+2\cos(2\,\overline{a}_1+\overline{a}_2)-3)^2))-(\cos(\overline{a}_1)\,((4-(2\cos(\underline{a}_3/2)\cos(\overline{a}_1)-\cos(\overline{a}_1)+2\sin(\overline{a}_1)$
$(\cos(\underline{a}_3/2)/2-\cos(2\,\underline{a}_3)/8-\cos((3\,\underline{a}_3)/2)/2+1/8)^{(1/2)}+\cos(\underline{a}_3)\cos(\overline{a}_1))^2)^{(1/2)}-(4-(78$
$\cos(2\,\overline{a}_1)+\cos(2\,\overline{a}_2)+2\cos(4\,\overline{a}_1)+2\cos(3\,\overline{a}_2)-2\cos(2\,\overline{a}_1-\overline{a}_2)-\cos(2\,\overline{a}_1-2\,\overline{a}_2)-11$
$\cos(2\,\overline{a}_1+3\,\overline{a}_2)-\cos(2\,\overline{a}_1+4\,\overline{a}_2)+18\cos(4\,\overline{a}_1+2\,\overline{a}_2)-\cos(6\,\overline{a}_1+2\,\overline{a}_2)-121\cos(\overline{a}_2)+96$
$\cos(2\,\overline{a}_1+\overline{a}_2)+72\sin(\overline{a}_1)\sin(\overline{a}_1-\overline{a}_2)+8\sin(\overline{a}_1)\sin(\overline{a}_1-2\,\overline{a}_2)+44\sin(\overline{a}_1)\sin(\overline{a}_1+2$
$\overline{a}_2)-20\sin(3\,\overline{a}_1)\sin(\overline{a}_1+\overline{a}_2)+4\sin(\overline{a}_1)\sin(\overline{a}_1+3\,\overline{a}_2)-72\sin(\overline{a}_1)\sin(3\,\overline{a}_1+\overline{a}_2)+4\sin(\overline{a}_1)$
$\sin(5\,\overline{a}_1+\overline{a}_2)-22\sin(3\,\overline{a}_1+2\,\overline{a}_2)\sin(\overline{a}_1+\overline{a}_2)-4\sin(3\,\overline{a}_1+3\,\overline{a}_2)\sin(\overline{a}_1+\overline{a}_2)+4\sin(5$
$\overline{a}_1+2\,\overline{a}_2)\sin(\overline{a}_1+\overline{a}_2)+2\sin(5\,\overline{a}_1+3\,\overline{a}_2)\sin(\overline{a}_1+\overline{a}_2)-80\cos(\overline{a}_2)^2-16\cos(\overline{a}_2)^3+100$
$\sin(\overline{a}_1)^2-36\sin(\overline{a}_2)^2+160\sin(\overline{a}_1)^4-32\cos(2\,\overline{a}_1+\overline{a}_2)^2+112\cos(2\,\overline{a}_1)\cos(\overline{a}_2)-64\cos(2$
$\overline{a}_1)\cos(2\,\overline{a}_1+\overline{a}_2)-16\cos(\overline{a}_2)\cos(2\,\overline{a}_1+\overline{a}_2)^2+32\cos(\overline{a}_2)^2\cos(2\,\overline{a}_1+\overline{a}_2)-4\sin(3\,\overline{a}_1-\overline{a}_2)$
$\sin(\overline{a}_1)-44\sin(3\,\overline{a}_1+2\,\overline{a}_2)\sin(\overline{a}_1)-8\sin(3\,\overline{a}_1+3\,\overline{a}_2)\sin(\overline{a}_1)+8\sin(5\,\overline{a}_1+2\,\overline{a}_2)\sin(\overline{a}_1)+4$
$\sin(5\,\overline{a}_1+3\,\overline{a}_2)\sin(\overline{a}_1)-32\cos(2\,\overline{a}_1)^2+36\cos(\overline{a}_1+\overline{a}_2)\cos(\overline{a}_1)+32\cos(2\,\overline{a}_1)\cos(\overline{a}_2)^2-16$
$\cos(2\,\overline{a}_1)^2\cos(\overline{a}_2)+244\sin(\overline{a}_1+\overline{a}_2)\sin(\overline{a}_1)+76\sin(\overline{a}_1+\overline{a}_2)^2+112\cos(\overline{a}_2)\cos(2\,\overline{a}_1+\overline{a}_2)-24$
$\cos(\overline{a}_1+\overline{a}_2)\sin(\overline{a}_1)\,(-(\cos(2\,\overline{a}_1)-\cos(\overline{a}_2)+\cos(2\,\overline{a}_1+\overline{a}_2)-1)\,(\cos(2\,\overline{a}_1)-\cos(\overline{a}_2)+\cos(2$
$\overline{a}_1+\overline{a}_2)+3))^{(1/2)}+4\sin(\overline{a}_1+\overline{a}_2)\cos(\overline{a}_1-\overline{a}_2)\,(-(\cos(2\,\overline{a}_1)-\cos(\overline{a}_2)+\cos(2\,\overline{a}_1+\overline{a}_2)-1)\,(\cos(2$
$\overline{a}_1)-\cos(\overline{a}_2)+\cos(2\,\overline{a}_1+\overline{a}_2)+3))^{(1/2)}-4\sin(\overline{a}_1+\overline{a}_2)\cos(\overline{a}_1+2\,\overline{a}_2)\,(-(\cos(2\,\overline{a}_1)-\cos(\overline{a}_2)+\cos(2$
$\overline{a}_1+\overline{a}_2)-1)\,(\cos(2\,\overline{a}_1)-\cos(\overline{a}_2)+\cos(2\,\overline{a}_1+\overline{a}_2)+3))^{(1/2)}+4\sin(\overline{a}_1+\overline{a}_2)\cos(3\,\overline{a}_1+\overline{a}_2)$
$(-(\cos(2\,\overline{a}_1)-\cos(\overline{a}_2)+\cos(2\,\overline{a}_1+\overline{a}_2)-1)\,(\cos(2\,\overline{a}_1)-\cos(\overline{a}_2)+\cos(2\,\overline{a}_1+\overline{a}_2)+3))^{(1/2)}+48$
$\cos(\overline{a}_1+\overline{a}_2)\cos(\overline{a}_1)\cos(\overline{a}_2)-48\cos(\overline{a}_1+\overline{a}_2)\cos(\overline{a}_1)\cos(2\,\overline{a}_1+\overline{a}_2)+8\sin(\overline{a}_1)\cos(\overline{a}_1-\overline{a}_2)$
$(-(\cos(2\,\overline{a}_1)-\cos(\overline{a}_2)+\cos(2\,\overline{a}_1+\overline{a}_2)-1)\,(\cos(2\,\overline{a}_1)-\cos(\overline{a}_2)+\cos(2\,\overline{a}_1+\overline{a}_2)+3))^{(1/2)}-8$
$\sin(\overline{a}_1)\cos(\overline{a}_1+2\,\overline{a}_2)\,(-(\cos(2\,\overline{a}_1)-\cos(\overline{a}_2)+\cos(2\,\overline{a}_1+\overline{a}_2)-1)\,(\cos(2\,\overline{a}_1)-\cos(\overline{a}_2)+\cos(2$
$\overline{a}_1+\overline{a}_2)+3))^{(1/2)}+8\sin(\overline{a}_1)\cos(3\,\overline{a}_1+\overline{a}_2)\,(-(\cos(2\,\overline{a}_1)-\cos(\overline{a}_2)+\cos(2\,\overline{a}_1+\overline{a}_2)-1)$
$(\cos(2\,\overline{a}_1)-\cos(\overline{a}_2)+\cos(2\,\overline{a}_1+\overline{a}_2)+3))^{(1/2)}+4\cos(3\,\overline{a}_1+2\,\overline{a}_2)\sin(\overline{a}_1+\overline{a}_2)\,(-(\cos(2$
$\overline{a}_1)-\cos(\overline{a}_2)+\cos(2\,\overline{a}_1+\overline{a}_2)-1)\,(\cos(2\,\overline{a}_1)-\cos(\overline{a}_2)+\cos(2\,\overline{a}_1+\overline{a}_2)+3))^{(1/2)}-48\cos(2\,\overline{a}_1)$
$\cos(\overline{a}_1+\overline{a}_2)\cos(\overline{a}_1)+16\cos(\overline{a}_1+\overline{a}_2)\cos(\overline{a}_1)\cos(\overline{a}_2)^2+16\cos(\overline{a}_1+\overline{a}_2)\cos(\overline{a}_1)\cos(2\,\overline{a}_1+\overline{a}_2)^2-12$
$\cos(\overline{a}_1+\overline{a}_2)\sin(\overline{a}_1+\overline{a}_2)\,(-(\cos(2\,\overline{a}_1)-\cos(\overline{a}_2)+\cos(2\,\overline{a}_1+\overline{a}_2)-1)\,(\cos(2\,\overline{a}_1)-\cos(\overline{a}_2)+\cos(2$
$\overline{a}_1+\overline{a}_2)+3))^{(1/2)}+8\cos(3\,\overline{a}_1+2\,\overline{a}_2)\sin(\overline{a}_1)\,(-(\cos(2\,\overline{a}_1)-\cos(\overline{a}_2)+\cos(2\,\overline{a}_1+\overline{a}_2)-1)\,(\cos(2$

$\overline{a}_1) - \cos(\overline{a}_2) + \cos(2\ \overline{a}_1 + \overline{a}_2) + 3))^{(1/2)} - 32\cos(2\ \overline{a}_1)\cos(\overline{a}_2)\cos(2\ \overline{a}_1 + \overline{a}_2) + 16\cos(2\ \overline{a}_1)^2$ $\cos(\overline{a}_1 + \overline{a}_2)\cos(\overline{a}_1) - 32\cos(\overline{a}_1 + \overline{a}_2)\cos(\overline{a}_1)\cos(\overline{a}_2)\cos(2\ \overline{a}_1 + \overline{a}_2) - 32\cos(2\ \overline{a}_1)\cos(\overline{a}_1 + \overline{a}_2)$ $\cos(\overline{a}_1)\cos(\overline{a}_2) + 32\cos(2\ \overline{a}_1)\cos(\overline{a}_1 + \overline{a}_2)\cos(\overline{a}_1)\cos(2\ \overline{a}_1 + \overline{a}_2) - 72)^2/(16\cos(\overline{a}_1 + \overline{a}_2)^2\ (2$ $\cos(2\ \overline{a}_1) - 2\cos(\overline{a}_2) + 2\cos(2\ \overline{a}_1 + \overline{a}_2) - 3)^4))^{(1/2)}))/2 - (\sin(\overline{a}_1)\ (2\cos(\underline{a}_3/2)\cos(\overline{a}_1) + 2$ $\sin(\overline{a}_1)\ (\cos(\underline{a}_3/2)/2 - \cos(2\ \underline{a}_3)/8 - \cos((3\ \underline{a}_3)/2)/2 + 1/8)^{(1/2)} + \cos(\underline{a}_3)\cos(\overline{a}_1) + (39\cos(2$ $\overline{a}_1) + \cos(2\ \overline{a}_2)/2 + \cos(4\ \overline{a}_1) + \cos(3\ \overline{a}_2) - \cos(2\ \overline{a}_1 - \overline{a}_2) - \cos(2\ \overline{a}_1 - 2\ \overline{a}_2)/2 - (11\cos(2\ \overline{a}_1 + 3$ $\overline{a}_2))/2 - \cos(2\ \overline{a}_1 + 4\ \overline{a}_2)/2 + 9\cos(4\ \overline{a}_1 + 2\ \overline{a}_2) - \cos(6\ \overline{a}_1 + 2\ \overline{a}_2)/2 - (121\cos(\overline{a}_2))/2 + 48\cos(2$ $\overline{a}_1 + \overline{a}_2) + 36\sin(\overline{a}_1)\sin(\overline{a}_1 - \overline{a}_2) + 4\sin(\overline{a}_1)\sin(\overline{a}_1 - 2\ \overline{a}_2) + 22\sin(\overline{a}_1)\sin(\overline{a}_1 + 2\ \overline{a}_2) - 10\sin(3$ $\overline{a}_1)\sin(\overline{a}_1 + \overline{a}_2) + 2\sin(\overline{a}_1)\sin(\overline{a}_1 + 3\ \overline{a}_2) - 36\sin(\overline{a}_1)\sin(3\ \overline{a}_1 + \overline{a}_2) + 2\sin(\overline{a}_1)\sin(5\ \overline{a}_1 + \overline{a}_2) - 11$ $\sin(3\ \overline{a}_1 + 2\ \overline{a}_2)\sin(\overline{a}_1 + \overline{a}_2) - 2\sin(3\ \overline{a}_1 + 3\ \overline{a}_2)\sin(\overline{a}_1 + \overline{a}_2) + 2\sin(5\ \overline{a}_1 + 2\ \overline{a}_2)\sin(\overline{a}_1 + \overline{a}_2) + \sin(5$ $\overline{a}_1 + 3\ \overline{a}_2)\sin(\overline{a}_1 + \overline{a}_2) - 40\cos(\overline{a}_2)^2 - 8\cos(\overline{a}_2)^3 + 50\sin(\overline{a}_1)^2 - 18\sin(\overline{a}_2)^2 + 80\sin(\overline{a}_1)^4 - 16\cos(2$ $\overline{a}_1 + \overline{a}_2)^2 + 56\cos(2\ \overline{a}_1)\cos(\overline{a}_2) - 32\cos(2\ \overline{a}_1)\cos(2\ \overline{a}_1 + \overline{a}_2) - 8\cos(\overline{a}_2)\cos(2\ \overline{a}_1 + \overline{a}_2)^2 + 16\cos(\overline{a}_2)^2$ $\cos(2\ \overline{a}_1 + \overline{a}_2) - 2\sin(3\ \overline{a}_1 - \overline{a}_2)\sin(\overline{a}_1) - 22\sin(3\ \overline{a}_1 + 2\ \overline{a}_2)\sin(\overline{a}_1) - 4\sin(3\ \overline{a}_1 + 3\ \overline{a}_2)\sin(\overline{a}_1) + 4$ $\sin(5\ \overline{a}_1 + 2\ \overline{a}_2)\sin(\overline{a}_1) + 2\sin(5\ \overline{a}_1 + 3\ \overline{a}_2)\sin(\overline{a}_1) - 16\cos(2\ \overline{a}_1)^2 + 16\cos(2\ \overline{a}_1)\cos(\overline{a}_2)^2 - 8$ $\cos(2\ \overline{a}_1)^2\cos(\overline{a}_2) + 122\sin(\overline{a}_1 + \overline{a}_2)\sin(\overline{a}_1) + 38\sin(\overline{a}_1 + \overline{a}_2)^2 + 56\cos(\overline{a}_2)\cos(2\ \overline{a}_1 + \overline{a}_2) - 12$ $\cos(\overline{a}_1 + \overline{a}_2)\sin(\overline{a}_1)\ (-(\cos(2\ \overline{a}_1) - \cos(\overline{a}_2) + \cos(2\ \overline{a}_1 + \overline{a}_2) - 1)\ (\cos(2\ \overline{a}_1) - \cos(\overline{a}_2) + \cos(2$ $\overline{a}_1 + \overline{a}_2) + 3))^{(1/2)} + 2\sin(\overline{a}_1 + \overline{a}_2)\cos(\overline{a}_1 - \overline{a}_2)\ (-(\cos(2\ \overline{a}_1) - \cos(\overline{a}_2) + \cos(2\ \overline{a}_1 + \overline{a}_2) - 1)\ (\cos(2$ $\overline{a}_1) - \cos(\overline{a}_2) + \cos(2\ \overline{a}_1 + \overline{a}_2) + 3))^{(1/2)} - 2\sin(\overline{a}_1 + \overline{a}_2)\cos(\overline{a}_1 + 2\ \overline{a}_2)\ (-(\cos(2\ \overline{a}_1) - \cos(\overline{a}_2) + \cos(2$ $\overline{a}_1 + \overline{a}_2) - 1)\ (\cos(2\ \overline{a}_1) - \cos(\overline{a}_2) + \cos(2\ \overline{a}_1 + \overline{a}_2) + 3))^{(1/2)} + 2\sin(\overline{a}_1 + \overline{a}_2)\cos(3\ \overline{a}_1 + \overline{a}_2)$ $(-(\cos(2\ \overline{a}_1) - \cos(\overline{a}_2) + \cos(2\ \overline{a}_1 + \overline{a}_2) - 1)\ (\cos(2\ \overline{a}_1) - \cos(\overline{a}_2) + \cos(2\ \overline{a}_1 + \overline{a}_2) + 3))^{(1/2)} + 4$ $\sin(\overline{a}_1)\cos(\overline{a}_1 - \overline{a}_2)\ (-(\cos(2\ \overline{a}_1) - \cos(\overline{a}_2) + \cos(2\ \overline{a}_1 + \overline{a}_2) - 1)\ (\cos(2\ \overline{a}_1) - \cos(\overline{a}_2) + \cos(2$ $\overline{a}_1 + \overline{a}_2) + 3))^{(1/2)} - 4\sin(\overline{a}_1)\cos(\overline{a}_1 + 2\ \overline{a}_2)\ (-(\cos(2\ \overline{a}_1) - \cos(\overline{a}_2) + \cos(2\ \overline{a}_1 + \overline{a}_2) - 1)\ (\cos(2$ $\overline{a}_1) - \cos(\overline{a}_2) + \cos(2\ \overline{a}_1 + \overline{a}_2) + 3))^{(1/2)} + 4\sin(\overline{a}_1)\cos(3\ \overline{a}_1 + \overline{a}_2)\ (-(\cos(2\ \overline{a}_1) - \cos(\overline{a}_2) + \cos(2$ $\overline{a}_1 + \overline{a}_2) - 1)\ (\cos(2\ \overline{a}_1) - \cos(\overline{a}_2) + \cos(2\ \overline{a}_1 + \overline{a}_2) + 3))^{(1/2)} + 2\cos(3\ \overline{a}_1 + 2\ \overline{a}_2)\sin(\overline{a}_1 + \overline{a}_2)$ $(-(\cos(2\ \overline{a}_1) - \cos(\overline{a}_2) + \cos(2\ \overline{a}_1 + \overline{a}_2) - 1)\ (\cos(2\ \overline{a}_1) - \cos(\overline{a}_2) + \cos(2\ \overline{a}_1 + \overline{a}_2) + 3))^{(1/2)} - 6$ $\cos(\overline{a}_1 + \overline{a}_2)\sin(\overline{a}_1 + \overline{a}_2)\ (-(\cos(2\ \overline{a}_1) - \cos(\overline{a}_2) + \cos(2\ \overline{a}_1 + \overline{a}_2) - 1)\ (\cos(2\ \overline{a}_1) - \cos(\overline{a}_2) + \cos(2$ $\overline{a}_1 + \overline{a}_2) + 3))^{(1/2)} + 4\cos(3\ \overline{a}_1 + 2\ \overline{a}_2)\sin(\overline{a}_1)\ (-(\cos(2\ \overline{a}_1) - \cos(\overline{a}_2) + \cos(2\ \overline{a}_1 + \overline{a}_2) - 1)$

$(\cos(2\,\overline{a}_1) - \cos(\overline{a}_2) + \cos(2\,\overline{a}_1 + \overline{a}_2) + 3))^{(1/2)} - 16\,\cos(2\,\overline{a}_1)\,\cos(\overline{a}_2)\,\cos(2\,\overline{a}_1 + \overline{a}_2) - 36)/(2\,\cos(\overline{a}_1 + \overline{a}_2)\,(2\,\cos(2\,\overline{a}_1) - 2\,\cos(\overline{a}_2) + 2\,\cos(2\,\overline{a}_1 + \overline{a}_2) - 3)^2)))/2.$

(A.4)