DISSERTATION

KINEMATIC DESIGN OF REDUNDANT ROBOTIC MANIPULATORS THAT ARE OPTIMALLY FAULT TOLERANT

Submitted by

Khaled M. Ben-Gharbia

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

Fall 2014

Doctoral Committee:

Advisor: Anthony A. Maciejewski

Edwin K.P. Chong
Rodney G. Roberts
Iuliana Oprea

ABSTRACT

KINEMATIC DESIGN OF REDUNDANT ROBOTIC MANIPULATORS THAT ARE OPTIMALLY FAULT TOLERANT

It is common practice to design a robot's kinematics from the desired properties that are locally specified by a manipulator Jacobian. Conversely, one can determine a manipulator that possesses certain desirable kinematic properties by specifying the required Jacobian. For the case of optimality with respect to fault tolerance, one common definition is that the post-failure Jacobian possesses the largest possible minimum singular value over all possible locked-joint failures. This work considers Jacobians that have been designed to be optimally fault tolerant for 3R and 4R planar manipulators. It also considers 4R spatial positioning manipulators and 7R spatial manipulators. It has been shown in each case that multiple different physical robot kinematic designs can be obtained from (essentially) a single Jacobian that has desirable fault tolerant properties. In the first part of this dissertation, two planar examples, one that is optimal to a single joint failure and the second that is optimal to two joint failures, are analyzed. A mathematical analysis that describes the number of possible planar robot designs for optimally fault-tolerant Jacobians is presented. In the second part, the large family of physical spatial positioning manipulators that can achieve an optimally failure tolerant configuration are parameterized and categorized. The different categories of manipulator designs are then evaluated in terms of their global kinematic properties, with an emphasis on failure tolerance. Several manipulators with a range of desirable kinematic properties are presented and analyzed. In the third part, 7R manipulators that are optimized for fault tolerance for fully general spatial motion are discussed. Two approaches are presented for identifying a physically feasible 7R optimally fault tolerant Jacobian. A technique for calculating both reachable and fault tolerant six-dimensional workspace volumes is presented. Different manipulators are analyzed and compared. In both the planar and spatial cases, the analyses show that there are large variabilities in the global kinematic properties of these designs, despite being generated from the same Jacobian. One can select from these designs to optimize additional application-specific performance criteria.

ACKNOWLEDGEMENTS

First and foremost, I wish to acknowledge my adviser, Dr. Anthony A. Maciejewski, for his constant encouragement and guidance. Dr. Maciejewski was always there whenever I needed help to find the right direction through my research journey. During my Ph.D. studies I have gained a significant amount of knowledge from Dr. Maciejewski, which is invaluable, and I am truly grateful for that. It is an horner to be one of Dr. Maciejewski's students.

I would also like to acknowledge Dr. Rodney G. Roberts from FAMU-FSU. Dr. Roberts has been an integral part in my Ph.D. studies. His advice has been always very important to me and has inspired me. I appreciate his constant support.

I would like to thank my Ph.D. committee members Dr. Edwin Chong and Dr. Iuliana Oprea for their time and help to improve my research work.

I wish to acknowledge parents. Their continues support has been limitless and invaluable. Because of their constant encouragement I was able to overcome all of the difficulties that I faced in this long journey.

Finally, I wish to acknowledge my wonderful wife Haifa Alkahili. Her constant support through the tough times during my Ph.D. studies was very significant. She has influenced and inspired me to reach my goals. I can't imagine how I could finish my earn my Ph.D. degree without having her beside me.

AUTOBIOGRAPHY

I was born in Tripoli-Libya on October 31, 1981. I received my Bachelor of Science degree in Electrical Engineering and my Master of Science degree in Automation and Control Engineering from Tripoli University(formerly known Al-fateh University) in March. 2003 and March. 2008 respectively. I completed my Ph.D. in Electrical Engineering from Colorado State University in September 2014. I have been employed by Wolf Robotics in Fort Collins, Colorado, as a software engineer. My research interests include robotics.

DEDICATION

To my family.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 BACKGROUND

The design and operation of fault-tolerant manipulators is critical for applications in remote and/or hazardous environments where routine maintenance and repair are not possible. Component failures for robots employed in structured and benign environments where regular maintenance can be performed are relatively rare. However, there are many important applications, although less common, where this is not true, e.g., in space exploration [7, 8], underwater exploration [9], and nuclear waste remediation [10]. One recent example is the Fukushima nuclear reactor accident, where robot component failures were not only likely, but inevitable [11, 12]. The failure rates for components in such harsh environments are relatively high [13–15], and maintenance is not possible. Many of these component failures will result in a robot's joint becoming immobilized, i.e., a locked joint failure mode [26]. In addition, component failures that result in other common failure modes, e.g., free-swinging joint failures [17–19], are frequently transformed into the locked joint failure mode by failure recovery mechanisms that employ fail safe brakes [20]. Because of the severe consequences of such failures there has been a great deal of research to improve manipulator reliability [15,21,22], design fault-tolerant robots [23,24], and determine mechanisms for analyzing [25], detecting [26,27], identifying [28–31], and recovering [32–35] from failures.

A large body of work on fault-tolerant manipulators has focused on the properties of kinematically redundant robots, either in serial or parallel form [36–42]. These analyses have been performed both on the local properties associated with the manipulator Jacobian [43–47] as well as the global characteristics such as the resulting workspace following a particular failure [48–52]. (Clearly both local and global kinematic properties are related, e.g., workspace boundaries correspond to singularities in the Jacobian.)

In this work it is assumed that one is given a set of local performance constraints that require a manipulator to function in a configuration that is optimal under normal operation and after an assumed maximum number of joints fail and are locked in position. For example, at a single joint failure, the desired Jacobian matrix must be isotropic, i.e., possess all equal singular values prior to a failure, and have equal minimum singular values for every possible single column being removed. One can then use global characteristics to distinguish between multiple manipulators that meet the local design constraints.

In the first part of this dissertation, it will be shown that there exist multiple different physical planar manipulators that correspond to the same optimally fault tolerant Jacobian. This is due to the fact that permutation of the columns of the Jacobian (or multiplying by ±1) does not affect its fault tolerant properties, however, it does significantly impact the resulting physical manipulator.

The second part of this dissertation deals with the problem of the Jacobian for an optimally fault tolerant, spatial positioning manipulator that possesses four degrees of freedom. It will be shown that for this case, permutation of the columns of the Jacobian (or multiplying by ±1) does not result in a different physical manipulator realization. However, there is a much greater degree of design flexibility.

The third part presents seven degree-of-freedom manipulators for for fully general spatial motion. It will discuss the fact that there isn't an exact optimally fault tolerant Jacobian that could be found, but all solutions converge to the same local minima. All calculated Jacobians are very close to being optimally fault tolerant. Two approaches are presented to identify such a Jacobian. A technique for measuring both reachable and fault tolerant six-dimensional volumes is presented.

Different examples of manipulators are evaluated and compared.

1.2 ORGANIZATION OF THIS STUDY

The remainder of this dissertation is organized as follows:

Chapter 2 defines mathematically a local measure of failure tolerance centered on desirable properties of the manipulator Jacobian. It also shows by using the Gram matrix how one can compute different robot deigns from the same optimally fault tolerant Jacobian.

Chapter 3 presents a procedure for computing the kinematic parameters of a manipulator, described by Denavit and Hartenberg (DH) parameters, from a given Jacobian. This procedure was useful for this work because a Jacobian matrix was defined first and then the manipulator's structure that corresponds to this Jacobian was determined, where one usually computes a Jacobian from the given DH parameters.

Chapter 4 discusses 3R planar manipulators that are designed from the same Jacobian that is optimally fault tolerant to a single locked joint failure. Their global properties was studied. The results showed how these manipulators are different, even though they have the same properties at the local design configuration.

Chapter 5 illustrates how the Gram matrix is used to describe all Jacobians with the same optimal fault tolerance properties. Then, 4R planar manipulators are designed from a Jacobian that is optimally fault tolerant to two locked joint failures. The global properties of these manipulators are analyzed and compared.

Chapter 6 characterizes the set of all 6×4 Jacobian matrices that include an optimally fault tolerant 3×4 spatial positioning sub-Jacobian. This characterization was then used to determine the family of DH parameters that represent physical manipulators with the optimally fault tolerant property. It will be described how one can evaluate a particular robot design (that is generated from the optimal Jacobian) in terms of its global kinematic properties, especially with regard to failure tolerance. The global kinematic properties analyses will be applied on several categories of manipulators. It will be illustrated why permutation of the columns of the Jacobian (or multiplying by ±1) does not result in different physical robots.

Chapter 7 discusses an optimally fault tolerant 6×7 matrix, and how one can identify a feasible Jacobian. It describes how one can evaluate both reachable and fault tolerant six-dimensional volumes. Different examples are presented.

Chapter 8 presents the conclusions of this work.

CHAPTER II

OPTIMALLY FAULT-TOLERANT JACOBIANS[1]

2.1 CHAPTER OVERVIEW

Measuring kinematic behaviors of a manipulator can be made through its Jacobian matrix. In this work, the main interest of a manipulator design is its fault tolerant properties. The chapter will discuss the properties of an optimally fault tolerant Jacobian in Section 2.2. Section 2.3 describes how one can determine different number of robots that can be obtained from a given optimally fault tolerant Jacobian. This section describes also how one can compute the kinematic parameters of a planar manipulator from a given Jacobian.

2.2 DEFINITION OF AN OPTIMALLY FAULT TOLERANT JACOBIAN

The dexterity of manipulators is frequently quantified in terms of the properties of the manipulator Jacobian matrix that relates end-effector velocities to joint angle velocities. The Jacobian will be denoted by the $m \times n$ matrix J where m is the dimension of the task space and n is the number of degrees-of-freedom (DOFs) of the manipulator. For redundant manipulators, $n > m$ and the quantity $n - m$ is the degree of redundancy. The manipulator Jacobian can be written as a collection of columns

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

where j_i represents the end-effector velocity due to the velocity of joint i. For an arbitrary single joint failure at joint f, assuming that the failed joint can be locked, the resulting m by $n - 1$

[1] MOST OF THIS CHAPTER IS PUBLISHED IN [1–6]

Jacobian will be missing the fth column, where f can range from 1 to n. This Jacobian will be denoted by a preceding superscript so that in general

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

The properties of a manipulator Jacobian are perhaps best illustrated through the use of the singular value decomposition (SVD), which can be defined as

$$ J = UDV^T \tag{3} $$

where U is an m by m orthogonal matrix of the output singular vectors, V is an n by n orthogonal matrix of the input singular vectors, and D is a diagonal matrix of the form

$$ D_{m\times n} = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \ddots & \vdots & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & 0 & \vdots & & \vdots \\ 0 & \cdots & 0 & \sigma_m & 0 & \cdots & 0 \end{bmatrix} \tag{4} $$

where the σ_i are the singular values, and are typically ordered from largest to smallest. Most local dexterity measures can be defined in terms of simple combinations of these singular values such as their product (determinant) [53], sum (trace), or ratio (condition number) [54–56]. The most significant of the singular values is σ_m, the minimum singular value, because it is by definition the measure of proximity to a singularity and tends to dominate the behavior of both the manipulability (determinant) and the condition number. The minimum singular value is also a measure of the worst-case dexterity over all possible end-effector motions.

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

$$\mathcal{K} = \min_{f=1}^{n}({}^f\sigma_m). \tag{5}$$

Physically, this corresponds to minimizing the worst-case increase in joint velocity when a joint is locked and the others must accelerate to maintain the desired end effector trajectory. In addition, maximizing $\mathcal{K}$ is equivalent to locally maximizing the distance to the post-failure workspace boundaries [13]. To insure that manipulator performance is optimal prior to a failure, an optimally failure tolerant Jacobian is further defined as having all equal singular values due to the desirable properties of isotropic manipulator configurations [54–56]. Under these conditions, to guarantee that the minimum ${}^f\sigma_m$ is as large as possible they should all be equal. It is easy to show [44] that the worst-case dexterity of an isotropic manipulator that experiences a single joint failure is governed by the inequality

$$\min_{f=1}^{n}({}^f\sigma_m) \leq \sigma\sqrt{\frac{n-m}{n}} \tag{6}$$

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

Using the above definition of an optimally failure tolerant configuration one can identify the structure of the Jacobian required to obtain this property [57].[2] In particular, one can show that the optimally failure tolerant criteria requires that each joint contributes equally to the null space of the Jacobian transformation [45, 47]. Physically, this means that the redundancy of the robot is uniformly distributed among all the joints so that a failure at any one joint can be compensated for

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

by the remaining joints. Therefore, in this work an optimally failure tolerant Jacobian is defined as being isotropic, i.e., $\sigma_i = \sigma$ for all i, and having a maximum worst-case dexterity following a failure, i.e., one for which ${}^f\sigma_m = \sigma\sqrt{\frac{n-m}{n}}$ for all f. The second condition is equivalent to the columns of the Jacobian having equal norms.

After identifying an optimally failure tolerant J, one might be interested in designing the kinematics for a manipulator that would possess these qualities. In the next section, the Gram matrix is used to analyze the different number of manipulator kinematics that can result from a given fault tolerant Jacobian.

2.3 FAULT TOLERANCE AND THE GRAM MATRIX

The Gram matrix,

$$G = J^T J, \tag{7}$$

provides insight into the geometry and fault tolerance of a manipulator design. Here, the Jacobian J can be the positional, orientational, or the manipulator Jacobian. Some care concerning units should be exercised in the case of the manipulator Jacobian or when there is a mixture of revolute and prismatic joints. When a Jacobian is isotropic, the Gram matrix takes on a particularly simple form: if the singular values of J are equal to 1, then $G = J^T J = I - NN^T$ where the $n \times (n-m)$ matrix N consists of $(n-m)$ orthonormal null vectors of J. In the case of a manipulator with a single degree of redundancy, $G = I - \hat{\mathbf{n}}_J \hat{\mathbf{n}}_J^T$, where $\hat{\mathbf{n}}_J$ is the unit length null vector when J is in a non-singular configuration. The requirement for optimal fault tolerance specifies further conditions on the null space matrix N. Specifically, the rows of N must all have the same norm $\sqrt{\frac{n-m}{n}}$ and be spread out in a sense that will be made precise later.

Once an optimal Gram matrix is determined, an obvious and important question is to characterize all the corresponding Jacobians and the kinematic parameters for the corresponding manipulators. Clearly, a simple change in the base frame orientation through rotation and/or reflection will not change the basic robot structure. The difference in this case is simply a pre-multiplication of the Jacobian by an orthogonal matrix. For the sake of discussion, two configurations will be called *equivalent* if their corresponding Jacobians differ only by a pre-multiplication by an orthogonal matrix Q. It can be shown that two full rank n-R Jacobians J and J' are equivalent if and only if $(J')^T J' = J^T J$, i.e., if their Gram matrices are equal.

Two planar n-R manipulators with equivalent Jacobians have essentially the same kinematic parameters, so the corresponding robot configurations can be considered to be the same in that sense. This is because when there is a change in the orientation of the base frame, either through a rotation or a combination of a rotation and reflection, the new Jacobian merely differs from the original by a multiplication by an orthogonal matrix. This is nicely illustrated for a planar 3R manipulator, which has a Jacobian of the form

$$J(\theta_1, \theta_2, \theta_3) = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} - a_3 s_{123} & -a_2 s_{12} - a_3 s_{123} & -a_3 s_{123} \\ a_1 c_1 + a_2 c_{12} + a_3 c_{123} & a_2 c_{12} + a_3 c_{123} & a_3 c_{123} \end{bmatrix}, \tag{8}$$

where the fixed a_i's are the link lengths and the variable θ_i's are the joint angles. They both describe kinematic parameters of a planar manipulator as shown in Figure 1. (The notation s_{ijk} and c_{ijk} indicates $\sin(\theta_i + \theta_j + \theta_k)$ and $\cos(\theta_i + \theta_j + \theta_k)$, respectively.)

If the base frame is changed by a rotation, represented here by a 2×2 rotation matrix $R(\phi)$, the manipulator's Jacobian becomes

$$J'(\theta_1, \theta_2, \theta_3) = R(\phi) J(\theta_1, \theta_2, \theta_3) = J(\theta_1 + \phi, \theta_2, \theta_3), \tag{9}$$

where $R(\phi)$ is the standard rotation matrix corresponding to a counter-clockwise rotation of ϕ radians about the z-axis. The kinematic parameters of the robot corresponding to the new Jacobian $J'(\theta_1, \theta_2, \theta_3)$ are the same as they were for J with the exception that θ_1 is now replaced with $\theta_1 + \phi$.

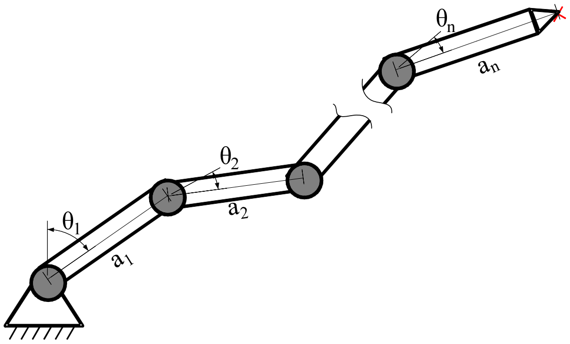


Figure 1: n-R planar kinematic parameters.

Consider now the reflection matrix $F = \text{diag}(-1, 1)$, which corresponds to a reflection about the y-z plane. Then the modified Jacobian resulting from pre-multiplying by F is

$$J'(\theta_1, \theta_2, \theta_3) = FJ(\theta_1, \theta_2, \theta_3) = J(-\theta_1, -\theta_2, -\theta_3). \tag{10}$$

The new kinematic parameters are the same except that the joint angles are the negatives of the original joint angles, giving a left-handed version of the same robot. More generally, any orthogonal matrix can be written in the form $R(\phi)$ or $Q = R(\phi)F$ for a suitable angle ϕ so that pre-multiplying (1) by Q results in the Jacobian

$$QJ(\theta_1, \theta_2, \theta_3) = J(-\theta_1 + \phi, -\theta_2, -\theta_3). \tag{11}$$

Because optimal fault tolerance can be formulated in terms of the Gram matrix, it is desirable to identify the family of kinematic parameters sets that result in optimally fault tolerant configurations. The unique kinematic parameters for a planar 3R robot are easily obtained from (8) by examining the matrix

$$J^* = \begin{bmatrix} j^*{}_1 & j^*{}_2 & j^*{}_3 \end{bmatrix} = \begin{bmatrix} j_1 - j_2 & j_2 - j_3 & j_3 \end{bmatrix}, \tag{12}$$

e.g.,the column norms of this new matrix ($J*$) are equal to the corresponding a_i values, such that

$$a_i^2 = \|{j^*}_i\|^2. \tag{13}$$

This observation generalizes for any planar n-R robot. To find out the values of θ_i, let

$${j^*}_i = \begin{bmatrix} j^*_{1i} & j^*_{2i} \end{bmatrix}^T, \tag{14}$$

then,

$$\theta_i = \text{arctan2}(j^*_{1i}, j^*_{2i}) \tag{15}$$

where arctan2 is the four-quadrant inverse tangent, which returns $\theta_i \in [-\pi, \pi]$.

One could also obtain the values for a_i from the Gram matrix by noting that for $i = 1, 2, \ldots, n-1$,

$$\begin{aligned} a_i^2 &= \|j_i - j_{i+1}\|^2 \\ &= \|j_i\|^2 + \|j_{i+1}\|^2 - 2j_i \cdot j_{i+1} \\ &= g_{ii} + g_{i+1,i+1} - 2g_{i,i+1} \end{aligned} \tag{16}$$

and

$$a_n^2 = \|j_n\|^2 = g_{nn} \tag{17}$$

where $g_{i,i+1}$ is the $(i, i+1)$ element of G. Thus, for planar n-R manipulators, a given Gram matrix G determines a family of equivalent manipulators each with the same set of a_i parameters determined by the square root of a simple linear combination of elements in G.

Another important question is whether one can identify other optimally fault tolerant designs from a given Jacobian that are not equivalent by pre-multiplication by an orthogonal matrix. It is clear from the definition of optimal fault tolerance that rearranging the columns of J or multiplying one or more of the columns of J by -1 will not affect local fault tolerance; however, this will typically result in a very different manipulator. We will say that J and J' are *similar* if one is obtained from the other by permuting and/or multiplying the columns of a Jacobian by -1. In other words, J and J' are similar if $J' = JS$ where S is an $n \times n$ matrix corresponding to the desired signed permutation of the columns of J. For convenience, we will say that J and J' are *nontrivially similar* if $S \neq \pm I$. We are interested in similar Jacobians because they share the same fault tolerance properties but generally correspond to fundamentally different manipulators. The Gram matrix G' corresponding to J' is obtained from the original Gram matrix G simply by applying the same row and column operations that were used to obtain J' from J. Consequently, one can easily obtain the a_i parameters for any similar Jacobian directly from the original G for the case of planar revolute manipulators.

CHAPTER III

COMPUTING DH PARAMETERS FROM A GIVEN JACOBIAN[1]

3.1 CHAPTER OVERVIEW

In the previous chapter, it is shown how one can calculate the kinematic parameters for a planar manipulator from its given Jacobian. In the case of a spatial manipulator, it is more complicated. This chapter will discuss how one can compute the DH parameters of a revolute manipulator from a desired Jacobian. This is the opposite of what is typically done, i.e., DH parameters of a manipulator are typically given, and a Jacobian is then computed from these DH parameters. The analyses of this computation are discussed in Section 3.2.

3.2 DH PARAMETERS FOR A SPATIAL MANIPULATOR

3.2.1 DH Coordinate Frames

The DH notation for specifying a spatial manipulator kinematics is illustrated in Figure 2 with the parameters defined in Table 1 [58]. It is assumed that J is computed for the hand, and is given with respect to the hand coordinate frame.

The ith column of J, that is composed of both the linear partial velocity v_i and rotational partial velocity ω_i, is closely related to the coordinate axes of the $i-1$ coordinate frame of the DH notation [59], i.e.,

$$j_i = \begin{bmatrix} v_i \\ \\ \omega_i \end{bmatrix} = \begin{bmatrix} \hat{z}_{i-1} \times p_{i-1} \\ \\ \hat{z}_{i-1} \end{bmatrix} \forall i = 1, ..., n \tag{18}$$

where p_{i-1} is the position vector from the coordinate frame $i-1$ to the hand coordinate frame.[2]

[1] MOST OF THIS CHAPTER IS PUBLISHED IN [1]

[2] One can apply this section's computations for a planar manipulator, where $v_i = [v_{1i},\ v_{2i},\ 0]$ and $\omega_i = [0,\ 0,\ 1]^T$ conventionally.

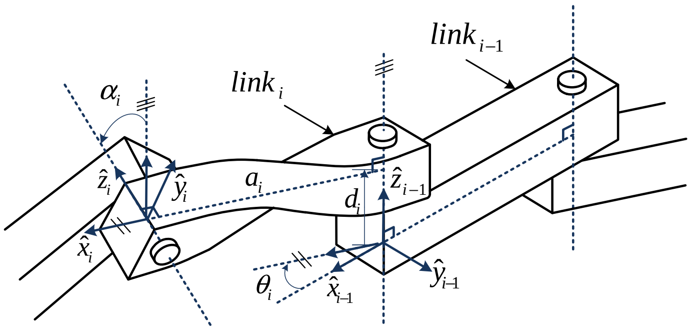


Figure 2: DH parameters for a joint i of an arbitrary manipulator [60].

Table 1: Definition of the Denavit and Hartenberg parameters

α_i	Link twist: the angle between z_{i-1} and z_i around x_i using the right-hand rule.
a_i	Link length: the shortest distance between the joint axes z_{i-1} and z_i.
θ_i	Joint angle: the angle between x_{i-1} and x_i around z_{i-1} using the right-hand rule.
d_i	Joint offset: the distance from the origin of coordinate frame $i-1$ to coordinate frame i along z_{i-1}.

Thus one can easily identify the $\hat{z}_{i-1}$ axis by inspection, i.e., it is equal to ω_i, and compute $\hat{x}_i$ from adjacent columns[3] of J using

$$\hat{x}_i = \pm \frac{\omega_i \times \omega_{i+1}}{\|\omega_i \times \omega_{i+1}\|}. \tag{19}$$

The appropriate sign for $\hat{x}_i$ in (19) is determined based on the convention that $\hat{x}_i$ is pointing away from $\hat{z}_{i-1}$ [59], so that

$$\langle \hat{x}_i,\ p'_i - p'_{i-1} \rangle > 0 \tag{20}$$

[3] To compute the nth coordinate frame assume that $v_{n+1} = [0\ 0\ 0]^T$ and $\omega_{n+1} = [0\ 0\ 1]^T$.

where $\langle \cdot , \cdot \rangle$ denotes the dot product, and p'_{i-1} is defined as

$$p'_{i-1} = \omega_i \times v_i. \tag{21}$$

If $\hat{z}_{i-1}$ and $\hat{z}_i$ are parallel, then $\hat{x}_i$ is given by:

$$\hat{x}_i = \frac{p'_i - p'_{i-1}}{\|p'_i - p'_{i-1}\|}. \tag{22}$$

Finally, the $\hat{y}_i$ axis can be obtained by taking the cross product of $\hat{z}_i$ and $\hat{x}_i$.

The procedure presented here assumes that J is specified with respect to the hand coordinate frame. If the desired J is given with respect to the base coordinate frame, it is easy to transform it to the hand frame by performing the appropriate rotation. To transform J to the hand coordinate frame, the following rotation matrix is used:

$${}^{n}R_0 = \begin{bmatrix} \hat{x}_n & \hat{y}_n & \hat{z}_n \end{bmatrix} \tag{23}$$

where

$$\hat{z}_n = \omega_n, \tag{24}$$

$$\hat{x}_n = \frac{v_n \times \omega_n}{\|v_n \times \omega_n\|}, \tag{25}$$

and

$$\hat{y}_n = \frac{\hat{z}_n \times \hat{x}_n}{\|\hat{z}_n \times \hat{x}_n\|} \tag{26}$$

Hence, the Jacobian with respect to the hand coordinate frame can be obtained by multiplying each of the v_i's and ω_i's by ${}^{n}R_0$.

3.2.2 Computing the DH Parameters

After defining the coordinate frames for each joint, the four DH parameters can be derived using their definitions in Table 1.

3.2.2.1 Link Twist α

The value of α_i can be computed using:

$$\alpha_i = \pm\text{acos}(\langle \hat{z}_{i-1},\ \hat{z}_i \rangle) \tag{27}$$

where acos is the inverse of cosine that returns a value between 0 and π. The sign of α should be the same as the sign of $\langle \hat{x}_i,\ \hat{z}_{i-1} \times \hat{z}_i \rangle$.

3.2.2.2 Link Length a

The value of a_i can be computed using:

$$a_i = \langle \hat{x}_i,\ p'_i - p'_{i-1} \rangle. \tag{28}$$

Because $\hat{x}_i$ is defined to point away from $\hat{z}_{i-1}$, the value of a_i is always positive.

3.2.2.3 Joint Angle θ

The value of θ_i can be computed using

$$\theta_i = \pm\text{acos}(\langle \hat{x}_{i-1},\ \hat{x}_i \rangle). \tag{29}$$

The sign of θ should be the same as the sign of $\langle \hat{z}_{i-1},\ \hat{x}_{i-1} \times \hat{x}_i \rangle$. The value of θ_1 is arbitrary because $\hat{x}_0$ is not defined; thus, it can be assumed to be zero.

3.2.2.4 Link Offset d

The last parameter, d_i, requires that one determine the origins of the coordinate frames $i-1$ and i with respect to the hand coordinate frame, denoted ${}^0O_{i-1}$ and ${}^0O_i$ respectively. One can then apply the definition in Table 1 to compute the distance between these origins along the $\hat{z}_{i-1}$ using:

$$d_i = \langle {}^0O_i -{}^0 O_{i-1},\ \hat{z}_{i-1} \rangle. \tag{30}$$

To determine the origin of any coordinate frame i, one can use the fact that the line L_i parameterized by t_i given by

$$L_i(t_i) = p'_i + t_i\hat{z}_i \tag{31}$$

is known because $\hat{z}_i$ and p'_i are known. One can then use the known value of a_i to determine a linear set of equations relating L_i and L_{i-1} that can be easily solved. That is, the relationship $L_i(t_i) = a_i\hat{x}_i + L_{i-1}(t_{i-1})$ has a unique solution for t_i and t_{i-1} unless the lines L_i and L_{i-1} are parallel, where d_i can be set arbitrarily equal to zero. The value of d_1 is arbitrary because ${}^0O_{-1}$ is not defined; thus, it can be assumed to be zero.

Appendix A presents the MATLAB code that implemented all of the above computations and computes the DH parameters from any given Jacobian.

3.2.3 A Different Solution

Another solution of computing the DH parameters can be considered if one uses always the positive sign of (19), and the condition of (20) is ignored, i.e. the direction of x_i is not forced to be always pointing away from z_{i-1} to z_i. This makes the solution of a_i from (28) (which is equal to the left hand side of (20)) can be either positive or negative comparing to the case in the previous section where it was always positive. However, the resulted value of α_i will be always positive because the condition $\langle\hat{x}_i,\ \hat{z}_{i-1} \times \hat{z}_i\rangle$ now becomes always positive. Note that whenever the direction of $\hat{x}_i$ is changed, both θ_i and θ_{i+1} values are shifted by $\pm\pi$. Recalling that d_i is the shortest distance between the origins of the coordinate frames $i-1$ and i along z_{i-1}, one can see that changing x_i direction will have no effect on d_i.

CHAPTER IV

THREE DOF PLANAR MANIPULATORS [1]

4.1 CHAPTER OVERVIEW

The simplest example of an optimally failure tolerant configuration is a three DOF planar manipulator. This chapter explains how one can design an optimally failure tolerant manipulator for the simple case of a three DOF planar manipulator. Section 4.2 shows an optimally fault tolerant 2 × 3 Jacobian. Section 4.3 discusses how one can determine how many possible different manipulators possess the same locally optimal fault tolerant properties by rearranging the columns of The Jacobian or multiplying one or more of the columns of the Jacobian by -1 (which is discussed in Chapter 2). The enumerated resulting manipulators are then analyzed in terms of their global properties in Section 4.4. Section 4.5 shows a proof of why a particular manipulator has a constant value of the fault tolerant measure $\mathcal{K}$ within a portion of its workspace.

4.2 OPTIMALLY FAULT TOLERANT JACOBIAN

An optimally failure tolerant configuration for a three DOF planar manipulator from [57] is given by the following Jacobian:

$$J = \begin{bmatrix} j_1 & j_2 & j_3 \end{bmatrix} = \begin{bmatrix} -\sqrt{\frac{2}{3}} & \sqrt{\frac{1}{6}} & \sqrt{\frac{1}{6}} \\ 0 & -\sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \end{bmatrix}. \tag{32}$$

The null space at this configuration is given by

$$v_3 = \begin{bmatrix} \sqrt{\frac{1}{3}} \\ \sqrt{\frac{1}{3}} \\ \sqrt{\frac{1}{3}} \end{bmatrix} \tag{33}$$

[1] MOST OF THIS CHAPTER IS PUBLISHED IN [1, 5]

which illustrates that each joint contributes equally to the null space motion, thus distributing the redundancy proportionally to all degrees of freedom. Geometrically, it is easy to see that the three vectors j_1, j_2, and j_3 are all 120° apart which results in a balanced coverage of the planar workspace. If the three possible joint failures are considered, one can show that

$$^{f}\sigma_2 = \sqrt{\frac{1}{3}} \tag{34}$$

for $f = 1$ to 3, which satisfies the optimally failure tolerant criterion. Given this example of an optimally failure tolerant J, one might be interested in designing the kinematics for a manipulator that would possess these qualities. The next section discusses how one can compute all possible different robots that have the same optimally fault tolerant properties at the local design configuration that meets (32).

4.3 DETERMINING OPTIMALLY FAULT TOLERANT MANIPULATORS

For a three column J there are six possible permutations (3!) and eight combinations of different signs (2^3), for a total of 48 possible Jacobian matrices with the desired isotropy and optimal fault-tolerance property. The notation J_{123} will be used to indicate the Jacobian in (32) that is in standard form and indicate column permutations or changes in sign on the subscripts, e.g., a permutation of the second and third columns as J_{132} and a change in the sign of column three as $J_{12(-3)}$.

This section uses equations (12)–(15) that are given in Chapter 2 to compute the kinematic parameters of robots that are optimal that correspond to the all the possible operations of rearranging the columns of (32) or multiplying one or more of its columns by -1. Figs. 3, 4, and 5 illustrate all of the possible robots for all 48 permutations and combinations.

While there are 48 possible permutations and combinations, it is obvious that they do not all

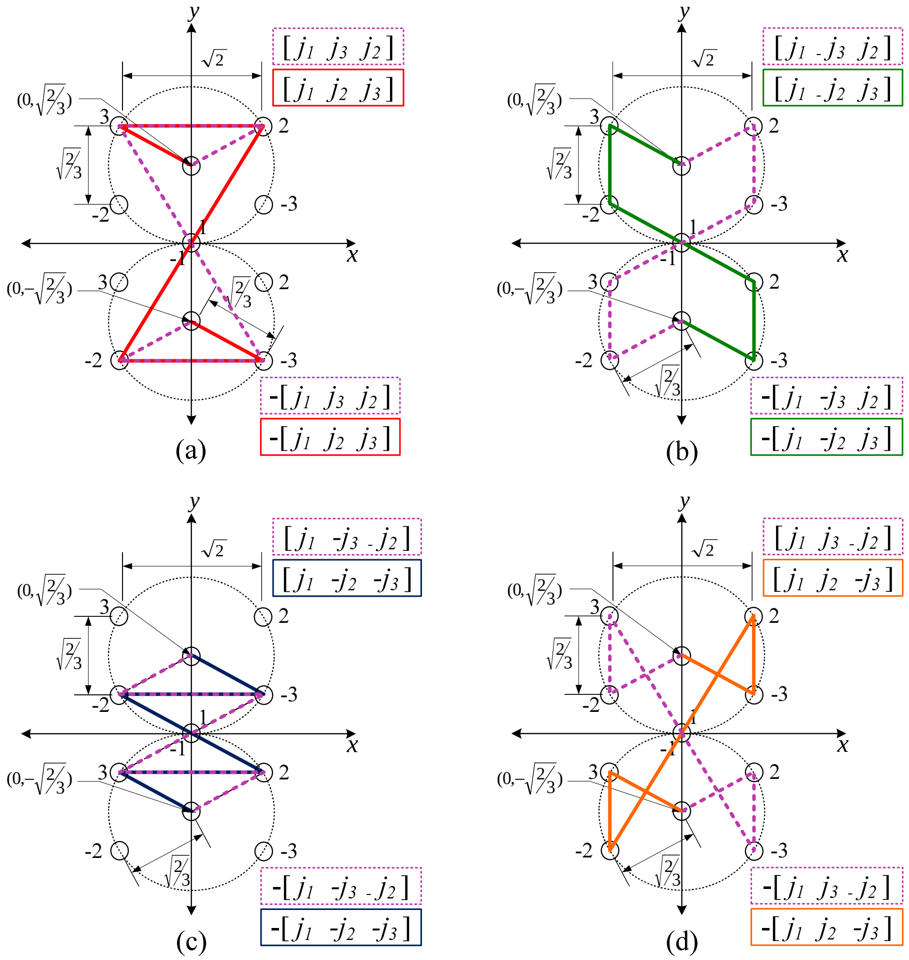
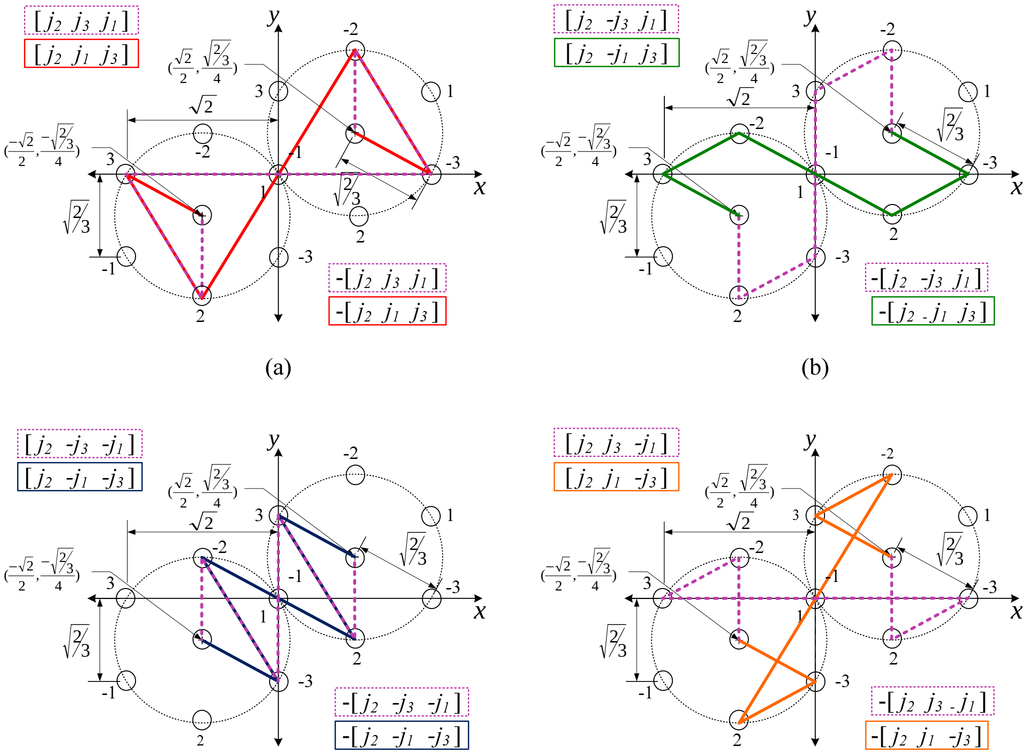


Figure 3: All the robots that are obtained by performing all possible sign combinations and permutations of the 2nd and 3rd columns of the J when j_1 is the first column.

result in different physical robots. In particular, consider the robots indicated with bold lines in Figure 3. These correspond to all eight combinations of sign changes for the Jacobian with the permutation J_{123}. It is easy to see that the four different cases indicated by (a), (b), (c) and (d), represent a single robot each, because the robot below the x-axis is the same as the one above with its first joint angle rotated by 180°. In fact, this is true for all six permutations of the columns so that there are really only four unique robots with a Jacobian that has the desired properties. Note that in each of the Figures 3 to 5, there are two different permutations of the columns that result in mirror versions of the same robot. The DH parameters for the four different robots are given in Table 2 and plotted together in Figure 6, where $L_s = \sqrt{\frac{2}{3}}$ and $L_l = \sqrt{2}$. Note that the value of θ_1



(a)

(b)

(c)

(d)

Figure 4: All the robots that are obtained by performing all possible sign combinations and permutations of the 2nd and 3rd columns of the J when j_2 is the first column.

in all cases is arbitrary because the properties of the Jacobian are independent of θ_1.

Table 2: DH parameters of the four generated robots

	J_{123}		$J_{1(-2)3}$		$J_{1(-2)(-3)}$		$J_{12(-3)}$	
i	θ_i	a_i	θ_i	a_i	θ_i	a_i	θ_i	a_i
1	60	L_l	150	L_s	150	L_s	60	L_l
2	120	L_l	-60	L_s	-150	L_l	-150	L_s
3	150	L_s	-120	L_s	150	L_s	-120	L_s

The next section will discuss how one can distinguish between these robots.

4.4 GLOBAL PROPERTIES ANALYSES

While all four robots in Table 2 have the same desired local behavior at the given configuration, they are quite different in terms of their global properties. For example, even when joint limits are not considered, the workspaces are quite different, e.g., the maximum reach will be either $3L_s$, $2L_s + L_l$, or $L_s + 2L_l$. More importantly, if one is concerned with fault-tolerance, the values of the

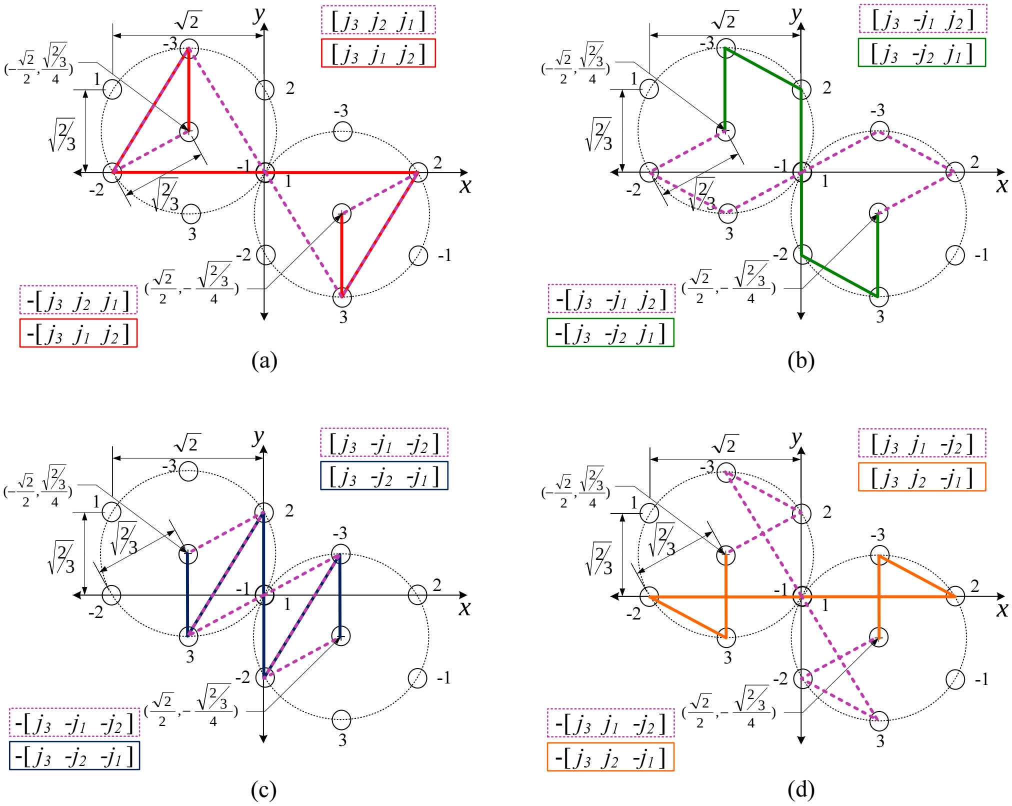


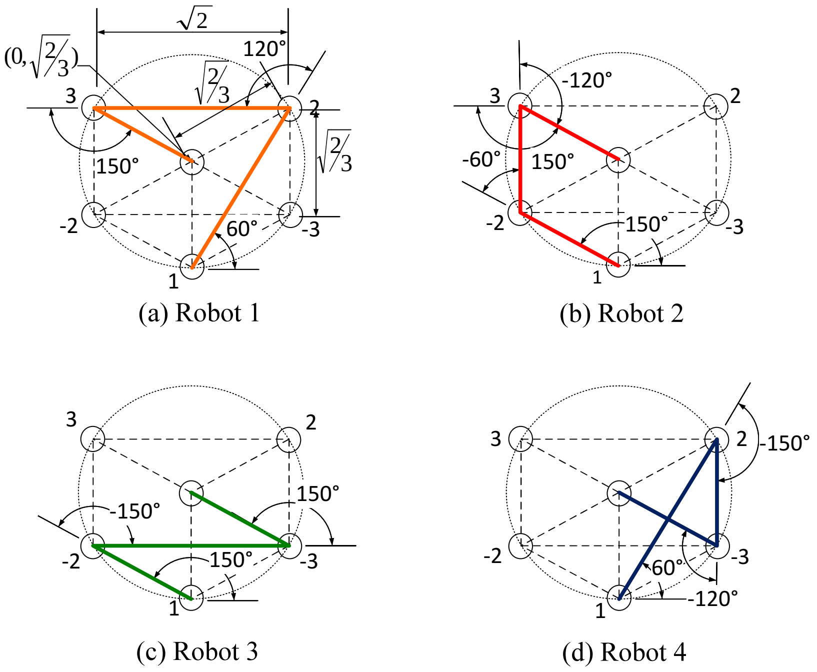
Figure 5: All the robots that are obtained by performing all possible sign combinations and permutations of the 2nd and 3rd columns of J when j_3 is the first column.

proposed fault-tolerance measure vary significantly for these four robot designs.

To determine how the fault tolerance measure $\mathcal{K}$ varies as a robot moves away from the configuration that has the optimal Jacobian, the optimal value of $\mathcal{K}$ were computed for every location within each of the four robot's workspaces.

To see how $\mathcal{K}$ varies with moving away from the local optimal point in each particular robot, an inverse kinematic procedure is used to compute the largest possible $\mathcal{K}$ as one goes away from the base on the x axis until reaching to the singularity point, $\theta_1 = \theta_2 = \theta_3 = 0$. (See Appendix B.)

Because $\mathcal{K}$ is not a function of θ_1, it is sufficient to compute its maximum value as a function of distance from the base of the manipulator. The maximum value of $\mathcal{K}$ is determined by computing all possible robot configurations for each distance, and calculating $\mathcal{K}$ for the Jacobian at that configuration.



(a) Robot 1

(b) Robot 2

(c) Robot 3

(d) Robot 4

Figure 6: A simple three degree-of-freedom planar robot that corresponds to the optimal fault-tolerant Jacobian given by (32) is shown in (d). The three other manipulators that have the same properties of the Jacobian in (32) are shown in (a), (b), and (c).

The results for each of the four robots is shown in Figures 7 to 10 where all three values of ${}^f\sigma_2$ for the configuration with maximum $\mathcal{K}$ are also plotted.

The first interesting point to note is that the manipulator with link lengths (L_l, L_l, L_s) in Figure 7 actually has a configuration with a larger value of $\mathcal{K}$ at the design point that is a distance of $\sqrt{2/3}$ from the base than that of the optimal value of $\mathcal{K} = \sqrt{1/3}$. This is possible because at this configuration the Jacobian is no longer isotropic, however, its non-isotropy is due to a larger maximum singular value, and so may not be considered undesirable. In addition, the value of $\mathcal{K}$ is significantly higher than the optimal value for a significant portion of this manipulator's workspace, making it particularly well suited for applications that require failure tolerance.

In contrast, consider the manipulator with link lengths (L_s, L_s, L_s) in Figure 8. It has a value of $\mathcal{K} = \sqrt{1/3}$ at the the optimal distance as designed, however, this is its peak value of $\mathcal{K}$, and $\mathcal{K}$ is monotonically decreasing away from this point. Thus, in addition to having the smallest workspace,

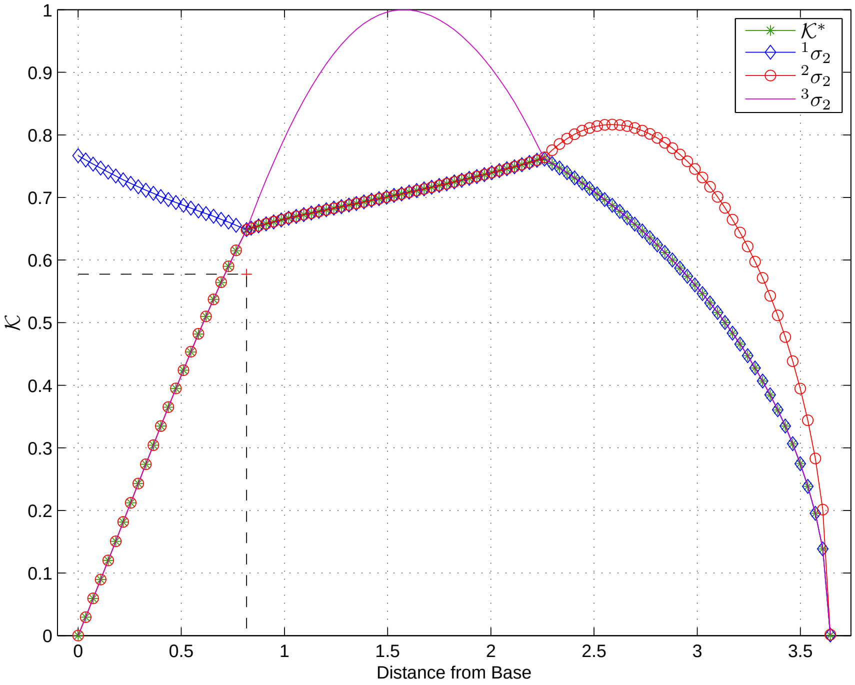


Figure 7: The relationship between $\mathcal{K}$ and the distance from the base for the (L_l, L_l, L_s) robot.

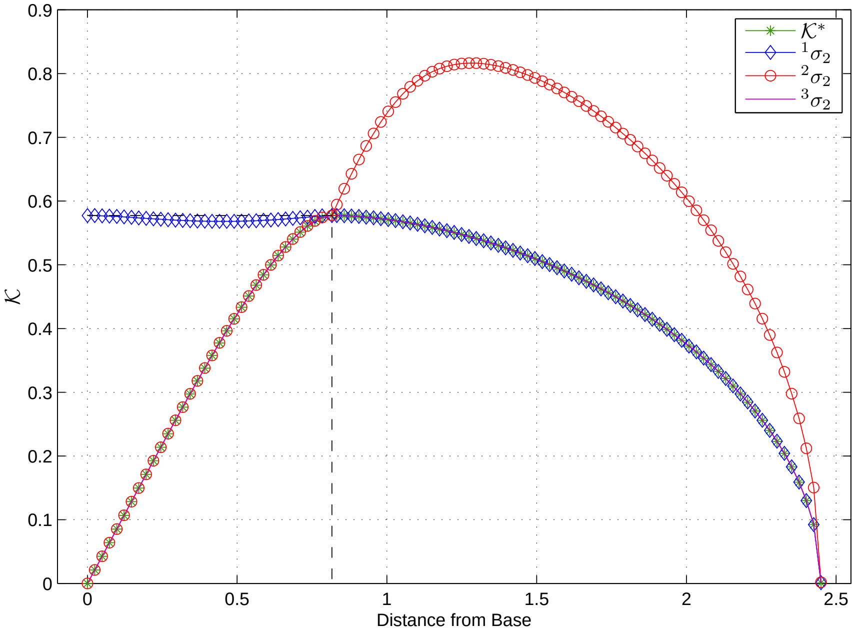


Figure 8: The relationship between $\mathcal{K}$ and the distance from the base for the (L_s, L_s, L_s) robot.

this manipulator has a significantly smaller tolerance to joint failures throughout its workspace.

The characteristics of the two medium length robots, i.e., whose link lengths are (L_s, L_l, L_s) and (L_l, L_s, L_s), fall somewhat inbetween the two extremes just described, but exhibit important

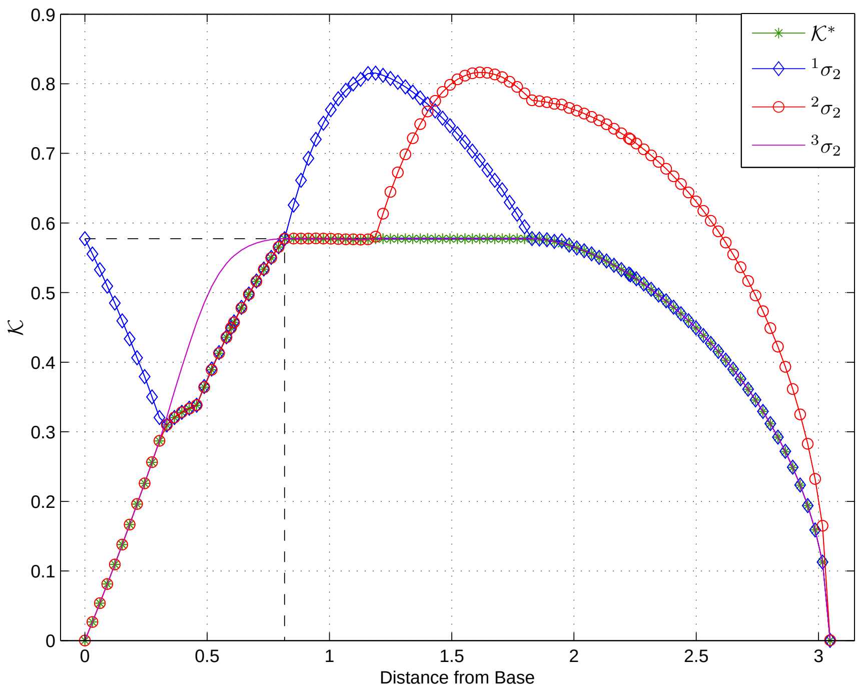


Figure 9: The relationship between $\mathcal{K}$ and the distance from the base for the (L_s, L_l, L_s) robot.

differences. The (L_s, L_l, L_s) robot shown in Figure 9 has a flat region for the maximum value of $\mathcal{K}$ in the middle of its workspace. (See the next section for a proof of why $\mathcal{K}$ is constant in this region.) In contrast, the (L_l, L_s, L_s) robot shown in Figure 10 has a significant dip in the maximum value of $\mathcal{K}$ at a distance near one unit from the base before it returns to a comparable value to that of The (L_s, L_l, L_s) robot.

In summary, even though all four robots are derived from the same optimally fault-tolerant Jacobian, their global properties are quite different, both prior to a failure and afterward. The (L_l, L_l, L_s) robot is arguably the most preferable due to its larger pre-failure workspace and the large value of fault tolerance over a substantial portion of its workspace.

4.5 EXPLANATION OF WHY $\mathcal{K}$ IS CONSTANT FOR (L_S, L_L, L_S) ROBOT IN FIGURE 9

A striking feature of Figure 9 is the flat region of the plot of $\mathcal{K}$ for the (L_s, L_l, L_s) robot. In this chapter, it will be shown that the maximum value of ${}^3\sigma_2$ for (L_s, L_l, L_s) robot is actually constant

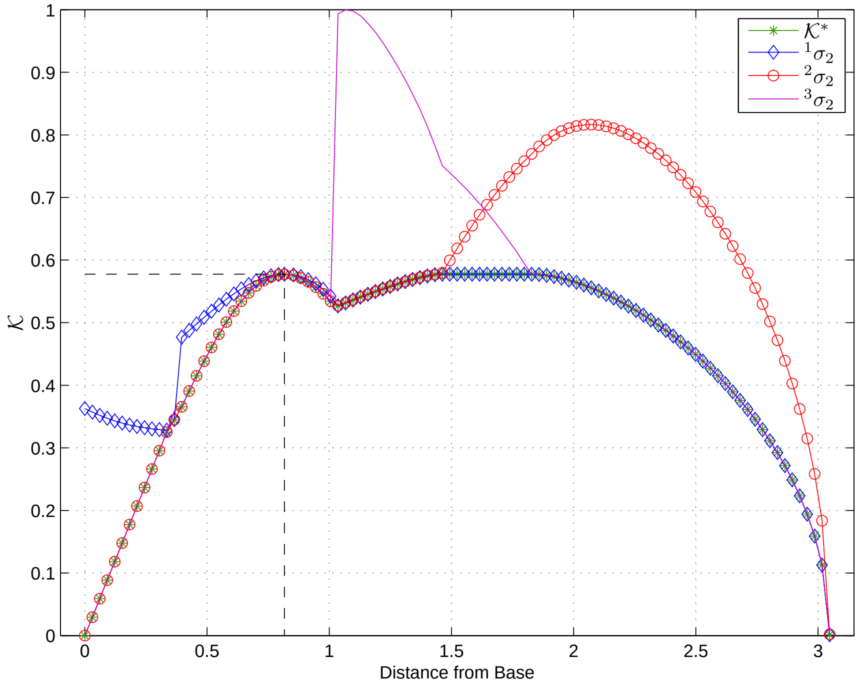


Figure 10: The relationship between $\mathcal{K}$ and the distance from the base for the (L_l, L_s, L_s) robot.

for a range of distances from the base that includes the flat region in Figure 9. In the case of the analysis for (L_s, L_l, L_s) robot in Figure 9, the critical failure happens to be joint 3 over the region where $\mathcal{K}$ is flat. (See Figure 11.)

It can be noted that the singular values of the reduced Jacobian

$$ {}^3J = \begin{bmatrix} -a_1s_1 - a_2s_{12} - a_3s_{123} & -a_2s_{12} - a_3s_{123} \\ a_1c_1 + a_2c_{12} + a_3c_{123} & a_2c_{12} + a_3c_{123} \end{bmatrix} \tag{35} $$

are equal to the square roots of the eigenvalues of

$$ ({}^3J)^T({}^3J) = \begin{bmatrix} \|j_1\|^2 & j_1 \cdot j_2 \\ j_1 \cdot j_2 & \|j_2\|^2 \end{bmatrix}, \tag{36} $$

which are readily given by the characteristic equation of (36). In particular, the minimum singular value ${}^3\sigma_2$ of (35) is given by the relationship

$$ 2({}^3\sigma_2)^2 = \|j_1\|^2 + \|j_2\|^2 - \sqrt{(\|j_1\|^2 - \|j_2\|^2)^2 + 4(j_1 \cdot j_2)^2}. \tag{37} $$

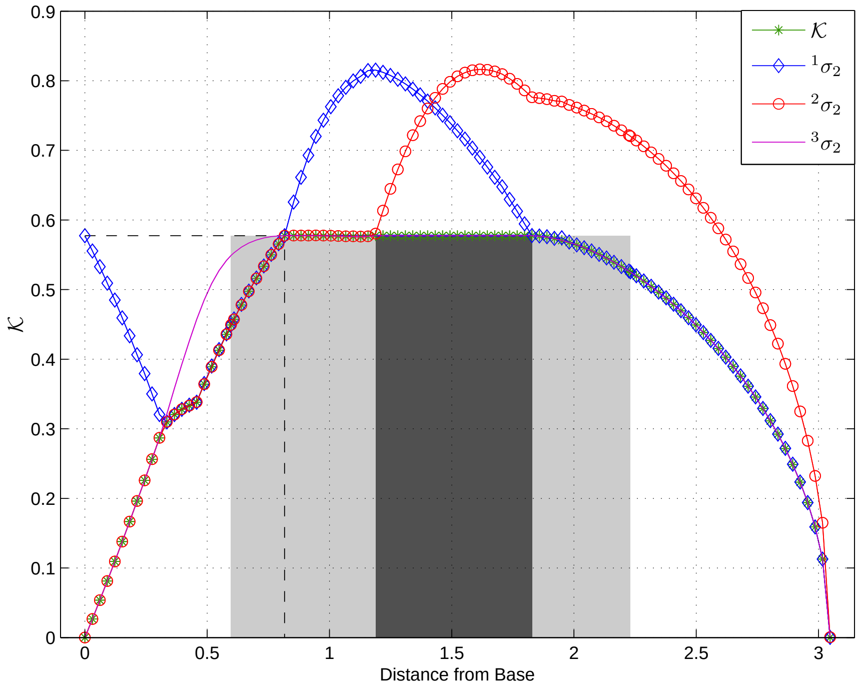


Figure 11: The relationship between $\mathcal{K}$ and the distance from the base for (L_s, L_l, L_s) robot in Table 2. The minimum singular values for all possible failures are shown for the configuration that maximizes $\mathcal{K}$. Note that if only $\mathcal{K}_3$ where being optimized, then it would be constant for a larger range.

From (35), one has that $\|j_1\|$ is equal to the distance d from the base to the end effector and that $\|j_1 - j_2\|$ is equal to the first link length, a_1, which together imply that

$$j_1 \cdot j_2 = \frac{1}{2}[\|j_1\|^2 + \|j_2\|^2 - \|j_1 - j_2\|^2] = \frac{1}{2}[d^2 + \|j_2\|^2 - a_1^2]. \tag{38}$$

The expression for (37) can then be written as

$$2({}^3\sigma_2)^2 = z + d^2 - \sqrt{(z - d^2)^2 + (z + d^2 - a_1^2)^2}, \tag{39}$$

where for convenience the notation $z = \|j_2\|^2$ has introduced. Because the link length a_1 is fixed, for a specified end-effector distance d, (39) is a function of the single variable z, which will be denoted by $g(z)$.

Setting the derivative of

$$g(z) = z + d^2 - \sqrt{(z - d^2)^2 + (z + d^2 - a_1^2)^2} \tag{40}$$

to zero, one obtains a single extremal point whose value depends on whether d is greater than, less than, or equal to $a_1/\sqrt{2}$. This extremal point can be conveniently written as $z^* = \max(d^2, a_1^2 - d^2)$. The case of interest here is when $d > a_1/\sqrt{2}$. In this case, $z^* = d^2$, $g'(z^*) = 0$, and $g''(z) < 0$ for all z. It then follows that $z^* = d^2$ results in a maximum value of $g(z^*) = a_1^2$ regardless of the specific value of d. For the robot to achieve $z = \|j_2\|^2 = d^2$, it is necessary and sufficient that $|a_2 - a_3| \leq d \leq a_2 + a_3$ due to the geometric constraints on $\|j_2\|$ associated with the lengths of the second and third links. In summary, if $d > a_1/\sqrt{2}$ and $|a_2 - a_3| \leq d \leq a_2 + a_3$, then the value of $\mathcal{K}_3(d)$ is equal to $a_1/\sqrt{2}$, where $\mathcal{K}_3(d)$ denotes the maximum value of ${}^3\sigma_2$ over all configurations where the end effector is at a distance d from the base. It is interesting to note that this solution corresponds to the end effector being equally distant from the base and the second joint. Depending on the values of d and the link lengths a_i, these conditions may or may not be possible due to the geometry of the robot. In the case of (L_s, L_l, L_s) robot, $a_1/\sqrt{2} = 1/\sqrt{3} = 0.5774$, $|a_2 - a_3| = 0.5977$, and $a_2 + a_3 = 2.2307$, and it can be concluded that $\mathcal{K}_3 = 0.5774$ over the range $0.5977 \leq d \leq 2.2307$.

CHAPTER V

FOUR DOF PLANAR MANIPULATORS [1]

5.1 CHAPTER OVERVIEW

This chapter discusses 4R planar manipulators that are optimally fault tolerant to any possibility of two locked joints failures at once. 5.2 presents 2×4 optimally fault tolerant Jacobian of a 4R planar manipulator. In the previous chapter we illustrated how one can simply determine how many different manipulators that can be obtained from a given optimally fault tolerant Jacobian, by computing the DH parameters for all possible resulted Jacobians from permuting and multiplying the columns by -1. Here, Section 5.3 will illustrate how one can use the Gram matrix to count the total number of the manipulators. The following section analyzes all these 4R planar manipulators and discusses how they are different in terms of their global properties.

5.2 OPTIMALLY FAULT TOLERANT JACOBIAN

As a further example of an optimally fault tolerant manipulator, consider a planar 4R robot. The requirements for optimal fault tolerance are that the Jacobian is isotropic and that the null space matrix N, which consists of two orthonormal null vectors of J, has the properties that its rows each have a norm of $1/\sqrt{2}$ and that the angles between successive rows are 45°. Any other null space matrix related to N by a row permutation and/or the multiplication of one or more rows by -1 will also result in an optimally fault tolerant Jacobian. The corresponding Jacobian would be given by applying the same operations to the columns of the original Jacobian. An example of

[1] MOST OF THIS CHAPTER IS PUBLISHED IN [2, 5]

a suitable Jacobian [45] is

$$J = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{2} & 0 & -\frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{\sqrt{2}} & \frac{1}{2} \end{bmatrix}. \tag{41}$$

The next section will show how one can compute all different physical robots that can possess the same properties of (41) with discussing firstly the previous example of 3R planar optimally fault tolerant Jacobian.

5.3 EXAMPLES OF MANIPULATORS WITH OPTIMAL FAULT-TOLERANT JACOBIANS

As mentioned earlier in Chapter 2, the restrictions imposed by this definition of fault tolerance limits the number of possible robot geometries. To see this, consider the problem of identifying all planar 3R manipulators with an optimally fault tolerant Jacobian J. When the 2×3 Jacobian J is isotropic with unit singular values, it yields that

$$G = J^T J = I - \hat{\mathbf{n}}_J \hat{\mathbf{n}}_J^T. \tag{42}$$

Fault tolerance requires that the components of $\hat{\mathbf{n}}_J$ have the same magnitude. However, replacing $\hat{\mathbf{n}}_J$ with $-\hat{\mathbf{n}}_J$ does not affect (42) so one only needs to check the four cases $\hat{\mathbf{n}}_J = 1/\sqrt{3}\begin{bmatrix} 1 & \pm 1 & \pm 1 \end{bmatrix}^T$. These four unit null vectors determine four families of non-equivalent Jacobians, each corresponding to one of the four possibilities for $I - \hat{\mathbf{n}}_J \hat{\mathbf{n}}_J^T$, which together identify all Jacobians that are optimally fault tolerant.

The optimally fault tolerant Jacobian given in (32) corresponds to the case when the elements of $\hat{\mathbf{n}}_J$ are all positive and equal. In this case the Gram matrix corresponding to the positional Jacobian is

$$G = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} \end{bmatrix}. \tag{43}$$

The link length parameters for this particular G are then $a_1 = a_2 = \sqrt{\frac{2}{3} + \frac{2}{3} - 2(\frac{-1}{3})} = \sqrt{2}$ and $a_3 = \sqrt{2/3}$. From the family of similar Gram matrices obtained through permutations and multiplications by -1 as described earlier, one can easily deduce that the only possible link length values for an optimally fault tolerant planar 3R manipulator are $L_l = \sqrt{2}$ and $L_s = \sqrt{2/3}$, which are obtained by using off-diagonal elements that equal $\pm\frac{1}{3}$ and diagonal elements equal to $\frac{2}{3}$. Furthermore, the square root of a diagonal value of G is equal to the distance of the end effector from the corresponding joint. In this case, each joint lies on a circle of radius $\sqrt{2/3}$ centered at the end effector with the two possible link lengths $\sqrt{2}$ and $\sqrt{2/3}$, which necessarily place the joints on the vertices of an inscribed hexagon. The four optimally fault tolerant manipulators are already described in the previous chapter by Table 2 and illustrated in Figure 6.

In the case of 4R planar manipulators, the Gram matrix of (41) is

$$G = \begin{bmatrix} \frac{1}{2} & \frac{1}{2\sqrt{2}} & 0 & \frac{-1}{2\sqrt{2}} \\ \frac{1}{2\sqrt{2}} & \frac{1}{2} & \frac{1}{2\sqrt{2}} & 0 \\ 0 & \frac{1}{2\sqrt{2}} & \frac{1}{2} & \frac{1}{2\sqrt{2}} \\ \frac{-1}{2\sqrt{2}} & 0 & \frac{1}{2\sqrt{2}} & \frac{1}{2} \end{bmatrix}. \tag{44}$$

From the diagonal elements of (44) it follows that the joints of the manipulator are located on a circle of radius $1/\sqrt{2}$ centered at the end effector. The link lengths for this particular G are $a_i = \sqrt{1 - \frac{1}{\sqrt{2}}}$ for $i = 1, 2, 3$ and $a_4 = \frac{1}{\sqrt{2}}$. It will be shown below that the four potential link lengths for similar Gram matrices are $L_a = \sqrt{1 - \frac{1}{\sqrt{2}}}$, $L_b = \frac{1}{\sqrt{2}}$, $L_c = 1$, and $L_d = \sqrt{1 + \frac{1}{\sqrt{2}}}$. Consequently, it follows that the joints of an optimally fault tolerant planar 4R manipulator appear on the vertices of an octagon inscribed on a circle of radius $\frac{1}{\sqrt{2}}$ centered at the end effector. The list of all possible manipulators is presented in Table 3 and depicted in Figure 12.

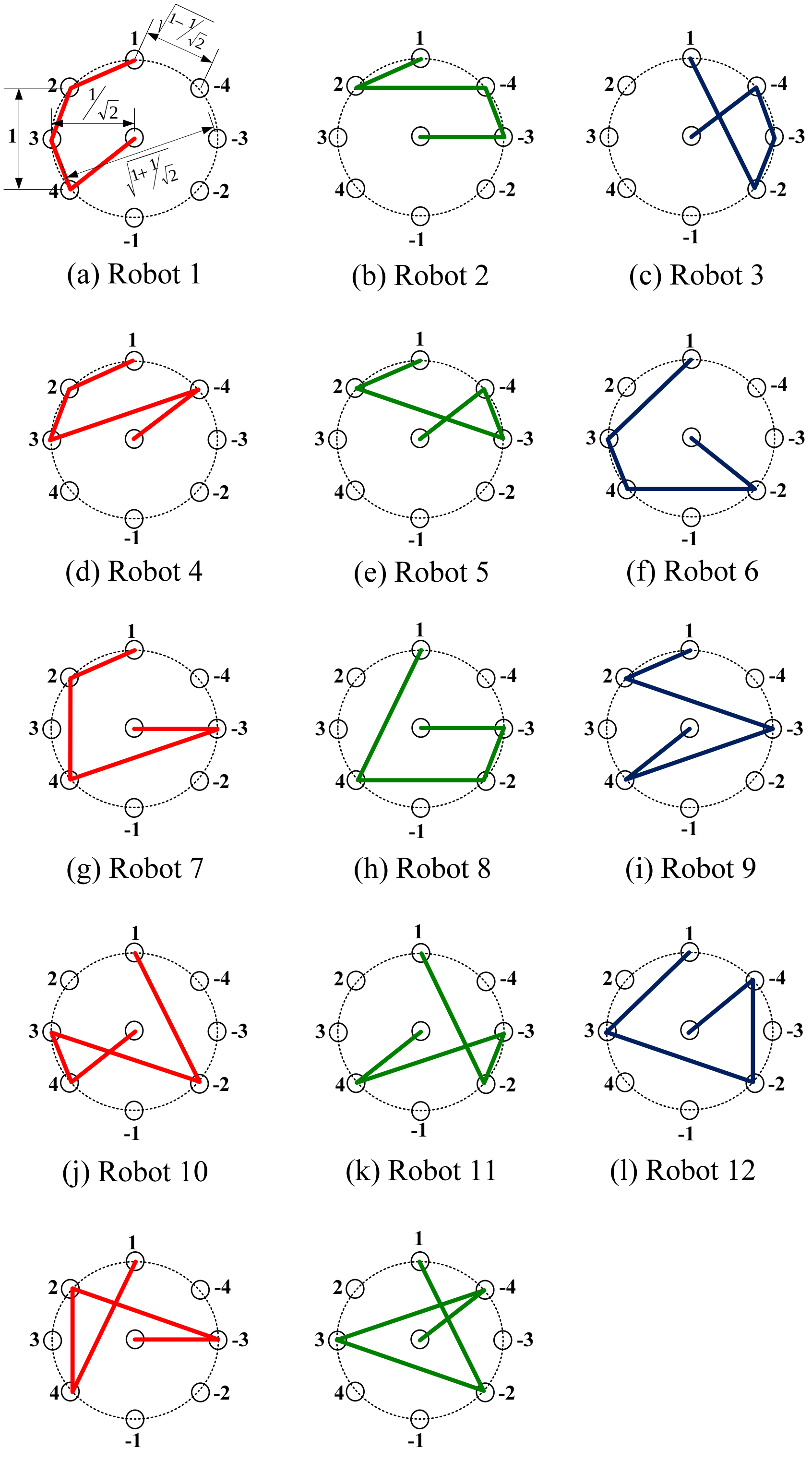
These possible robots resulted from the fact that all possible permutations and multiplications by -1 of the columns of (41) result in the superdiagonal of (44) (i.e., the diagonal above the main diagonal) being in exactly one of three forms: (x,y,z), (x,0,z), or (0,y,0), where each x, y, and z can be either $\pm\frac{1}{2\sqrt{2}}$. Thus, the total number of distinct link lengths is: $2^3 = 8$ for the (x,y,z) case plus $2^2 = 4$ for the (x,0,z) case plus $2^1 = 2$ for the (0,y,0) case resulting in 14 different manipulator designs. Note that not every manipulator with the property that its joints are located in the vertices of this octagon are optimally fault tolerant, but the Gram matrix clearly identifies this requirement for the family of optimally fault tolerant manipulators.

The next section discusses the global fault tolerant behavior of the 4R family of manipulators, and shows how the robots within the same family are still quite different as they act differently beyond the design point.

5.4 ANALYSIS AND COMPARISON OF MANIPULATOR DESIGNS

The fact that there are multiple manipulator designs with the same desired local fault tolerance properties, allows one to use other criteria for selecting a preferred design. In particular, while the robots all share the same local properties at the given configuration, they are quite different in terms of their global properties. To determine how the fault tolerance measure $\mathcal{K}$ varies as a robot moves away from the configuration that has the optimal Jacobian, the optimal value of $\mathcal{K}$ was computed for every location within each of the fourteen robots' workspaces. (See Appendix B.) Because $\mathcal{K}$ is not a function of θ_1, it is sufficient to compute its maximum value as a function of distance from the base of the manipulator.

The fourteen robots in Table 3 have different workspace properties, e.g. Robot 1 has the smallest maximum reach of $3L_a + L_b$ and Robot 14 has the largest at $3L_d + L_b$. Figure 13 illustrates how Robots 1 and 14 are also different in terms of the fault tolerance measure with respect to the distance from the base for the case of two joint failures. (Appendix C includes the plots of the other robots.) Robot 1 has a peak in $\mathcal{K}$ at its optimal value of $\frac{1}{2}\sqrt{1-\frac{1}{\sqrt{2}}}$ at the design point, with $\mathcal{K}$ decreasing relatively rapidly away from this point. In contrast, Robot 14 has a larger value



(a) Robot 1

(b) Robot 2

(c) Robot 3

(d) Robot 4

(e) Robot 5

(f) Robot 6

(g) Robot 7

(h) Robot 8

(i) Robot 9

(j) Robot 10

(k) Robot 11

(l) Robot 12

(m) Robot 13

(n) Robot 14

Figure 12: A simple four degree-of-freedom planar robot that corresponds to the optimal fault-tolerant Jacobian given by (41) is shown in (a). The thirteen other manipulators that have the same properties of the Jacobian in (41) are shown in (b) to (n).

Table 3: The 14 different link length combinations of (41).

Robot	a_1	a_2	a_3	a_4
1	L_a	L_a	L_a	L_b
2	L_a	L_c	L_a	L_b
3	L_d	L_a	L_a	L_b
4	L_a	L_a	L_d	L_b
5	L_a	L_d	L_a	L_b
6	L_c	L_a	L_c	L_b
7	L_a	L_c	L_d	L_b
8	L_d	L_c	L_a	L_b
9	L_a	L_d	L_d	L_b
10	L_d	L_d	L_a	L_b
11	L_d	L_a	L_d	L_b
12	L_c	L_d	L_c	L_b
13	L_d	L_c	L_d	L_b
14	L_d	L_d	L_d	L_b

of $\mathcal{K}$ at the design point than the optimal value of $\mathcal{K}$ which is because of the fact that at this configuration the Jacobian is no longer isotropic. Moreover, the value of $\mathcal{K}$ is significantly higher than the optimal value for a large portion of this manipulator's workspace.

Table 4 presents a comparison of the different global properties for the fourteen robots shown in Figure 12. The fault tolerant workspace percentage column is a measure of the ratio of fault tolerant workspace, in which $\mathcal{K}$ is greater than or equal to the value at the design point, to the total workspace. The average amount of joint motion per meter needed to stay at the configuration with the maximum value of $\mathcal{K}$ throughout the fault tolerant workspace is shown in the last column. Note that Robots 5 and 7 have much larger values for this measure because these robots encounter algorithmic singularities within the workspace that require a significant amount of reconfiguration for the robot to stay at the maximum value of $\mathcal{K}$. Robots 2, 3, 4, 5, 7 and 8 have the fault tolerant workspace separated into two pieces, i.e., there is a region in between where the maximum of $\mathcal{K}$ drops below the optimal value. (Similar to that of the (L_l, L_s, L_s) robot in the 3R case shown in Figure 6.) The amount of this drop varies depending upon the robot, ranging from as small as 0.2% for Robots 4 and 8, to as large as 9% for Robot 3. The number between parentheses is the smaller of the two fault tolerant workspaces, which in all cases includes the design point.

Clearly the maximum reach (or sum of the link lengths) has a dominant effect on the global

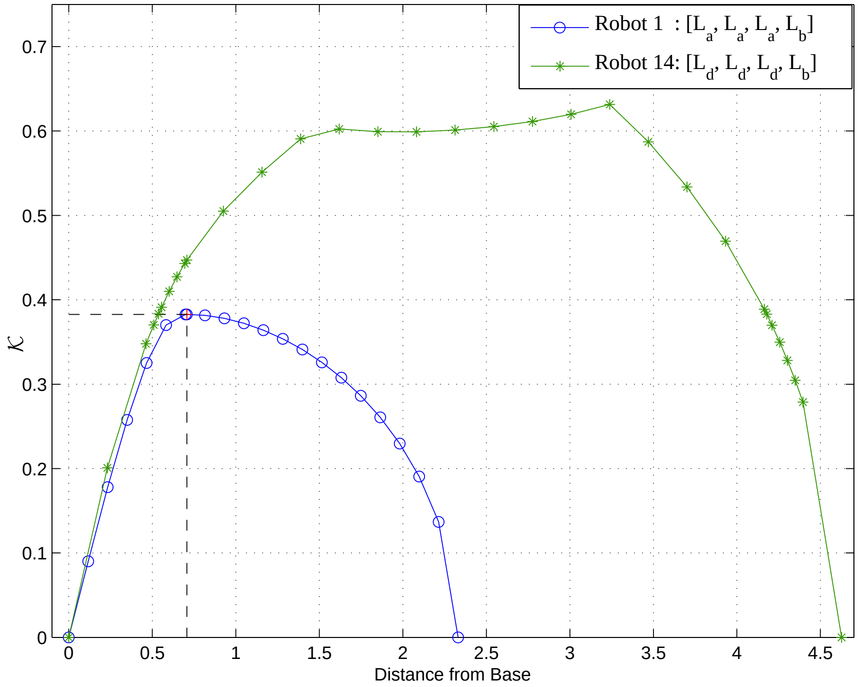


Figure 13: The relationship between $\mathcal{K}$ and the distance from the base for Robots 1 and 14 in Table 3.

fault tolerant properties.[2] However, there are three cases where the same maximum reach can be obtained by multiple different robot designs with significant differences in their global fault tolerant properties. Consider first the case of Robots 9, 10, and 11. Even though their fault tolerant workspace percentages are almost the same, there is a significant difference in the amount of joint motion needed to maintain a fault tolerant configuration. In particular, Robot 11 only moves a total of 177 degrees to traverse the entire fault tolerant region whereas Robots 9 and 10 take 191 and 263 degrees, respectively to do so. This is visually illustrated in Figure 14 where for each robot three different optimal configurations are shown. (The blue one is at the design point, the green one is at a boundary of the fault tolerant workspace, and the red configuration is at the middle.) Furthermore, the joint motion is distributed differently for the three robots, with Robot 9

[2] It is important to note that one can always scale these robot designs to obtain any desired maximum reach. This is why we normalize the fault tolerant workspace results in Table 4 to be a percentage.

Table 4: Fault tolerant workspace analysis of the fourteen robots in Table 3.

Robot	Reach $[m]$	Fault tolerant workspace [%]	Joint motion $[^\circ/m]$
1	2.33	0.00	–
2	2.79	15.14/(0.25)	107.3
3	3.10	9.99/(0.07)	117.0
4	3.10	22.16/(1.31)	98.6
5	3.10	26.96/(0.73)	372.8[1]
6	3.25	47.76	94.1
7	3.56	46.21/(0.04)	190.2[2]
8	3.56	49.06/(0.10)	72.2
9	3.86	57.68	79.7
10	3.86	58.02	109.8
11	3.86	58.78	71.0
12	4.01	73.03	76.5
13	4.32	77.42	74.1
14	4.63	80.23	70.7

[1] Robot 5 has two algorithmic singularities. Choosing the fault tolerant workspace bound before the first algorithmic singularity point gives us the fault tolerant workspace percentage of 6.3%, and joint motion of $84.1^\circ/m$.

[2] Robot 7 has one algorithmic singularity. Similar to Robot 5, if one chooses the fault tolerant workspace bound of Robot 7 before the algorithmic singularity point, the fault tolerant workspace percentage is 30.8%, and joints motion is $70.8^\circ/m$.

requiring much less motion in joint one, which may be desirable due to the large moment of inertia associated with this joint.

Robots 3, 4, and 5 also represent a group with equal reach but different global properties. If one only considers percentage of fault tolerant workspace, then Robot 3 is the worst (at 10%) and Robot 5 is the best (at 27%). However, Robot 5 encounters two algorithmic singularities within this region, which require the robot to reconfigure itself to a new posture in order to maintain $\mathcal{K}$ at its maximum value. This results in excessive joint motion over a very short period of time. If one opts to avoid this reconfiguration and follows the locally optimal value of $\mathcal{K}$, then $\mathcal{K}$ will monotonically decrease and results in a fault tolerant workspace percentage of only 6.3%. This is illustrated in Figure 15. Thus one could argue that Robot 4 is the best design out of the three.

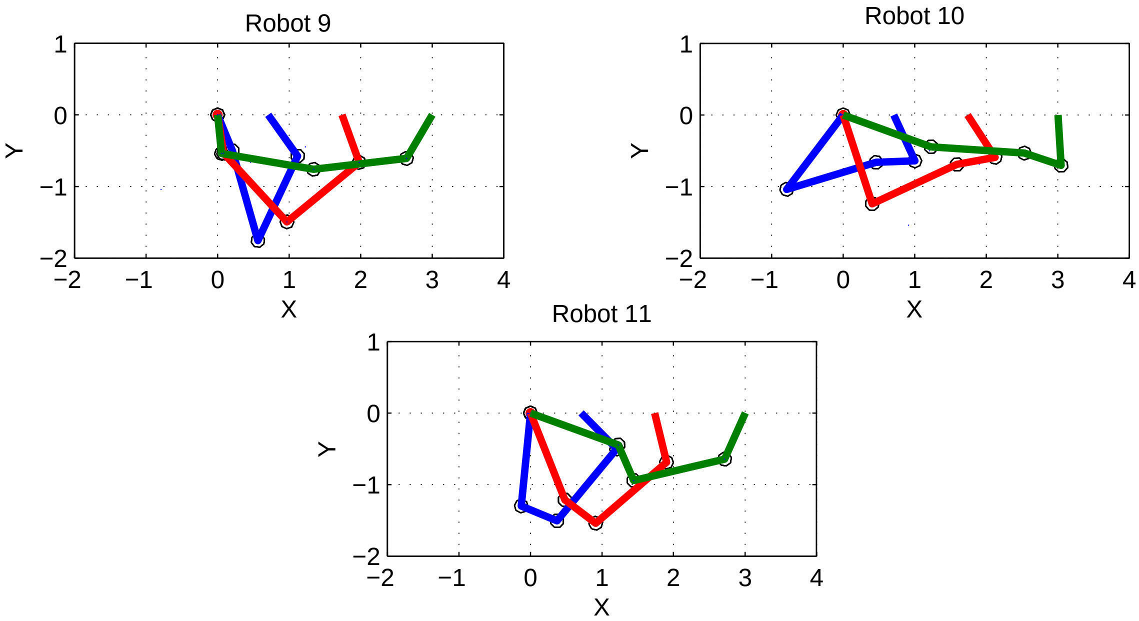


Figure 14: Three different configurations with maximum $\mathcal{K}$ at three different points along the x-axis trajectory for Robots 9, 10, and 11. In all cases the first configuration is from the design point at a distance of $1/\sqrt{2}$ and the last configuration is at the boundary of the fault tolerant workspace at a distance of approximately three.

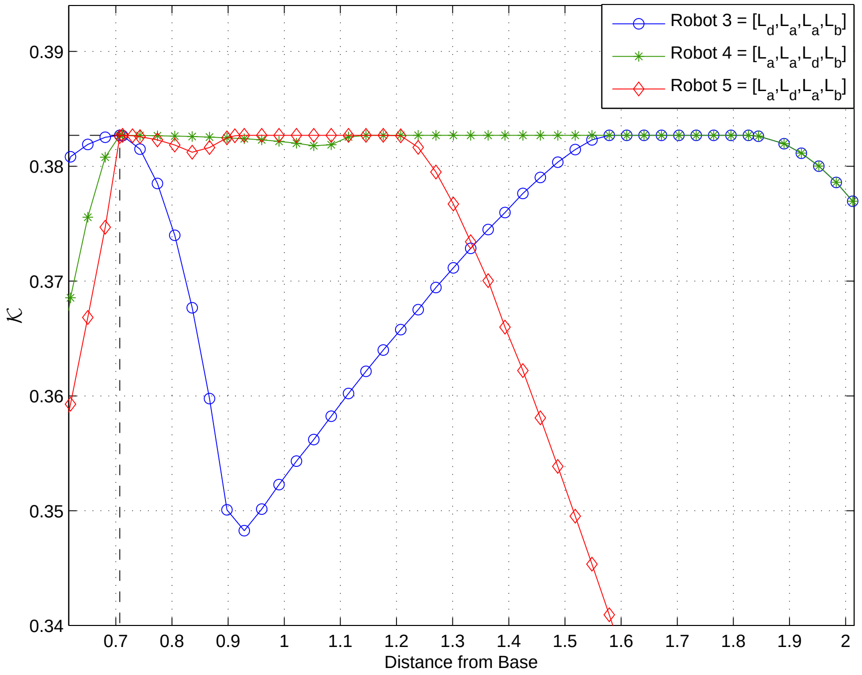


Figure 15: The relationship between $\mathcal{K}$ and the distance from the base for Robots 3, 4, and 5. The plot focuses on the behavior near the design point to highlight the difference in this region. Note that the value of $\mathcal{K}$ for Robot 5 is shown for joint motion that does not include a discontinuity due to an algorithmic singularity. If the discontinuous joint motion is performed, then the $\mathcal{K}$ value for Robot 5 is comparable to that of Robot 4.

CHAPTER VI

SPATIAL POSITIONING MANIPULATORS[1]

6.1 CHAPTER OVERVIEW

This chapter discusses the design of optimally fault-tolerant spatial positioning manipulators. Section 6.3 characterizes the set of all 6×4 Jacobian matrices that include an optimally fault tolerant 3×4 spatial positioning sub-Jacobian. This characterization is then used to determine the family of DH parameters that represent physical manipulators with the optimally fault tolerant property. Section 6.4 describes how one can evaluate a particular robot design (that is generated from the optimal Jacobian) in terms of its global kinematic properties, especially with regard to failure tolerance. The following section illustrates such an analysis on several categories of manipulators. Section 6.6 illustrates why permutation of the columns of the Jacobian (or multiplying by ± 1) does not result in different physical robots.

6.2 OPTIMALLY FAULT TOLERANT JACOBIAN

The general form of a manipulator's Jacobian is given by:

$$J_{6\times 4} = \begin{bmatrix} J_v \\ \\ J_\omega \end{bmatrix} \tag{45}$$

where J_v represents the linear velocity portion of a manipulator Jacobian, and J_ω represents the orientational velocity portion. For the case of a spatial positioning manipulator with four joints,

[1] MOST OF THIS CHAPTER IS PUBLISHED IN [3, 4]

an optimally failure tolerant configuration is given by [57]:

$$J_v = \begin{bmatrix} -\sqrt{\frac{3}{4}} & \sqrt{\frac{1}{12}} & \sqrt{\frac{1}{12}} & \sqrt{\frac{1}{12}} \\ 0 & -\sqrt{\frac{2}{3}} & \sqrt{\frac{1}{6}} & \sqrt{\frac{1}{6}} \\ 0 & 0 & -\sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \end{bmatrix} \tag{46}$$

The null space at this configuration is given by

$$\frac{1}{2}\,[1 \;\; 1 \;\; 1 \;\; 1]^T\,, \tag{47}$$

which illustrates that each joint contributes equally to the null space motion, thus distributing the redundancy proportionally to all DOFs. If the four possible single locked joint failures are considered, one can show that

$${}^f\sigma_3 = \mathcal{K}_{\text{max}} = \frac{1}{2} \tag{48}$$

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

Equation (46) is in fact a canonical form that essentially characterizes all optimally fault tolerant 3×4 Jacobians J_v. Optimal fault tolerance requires that the components of the unit length null vector have the same magnitude. Without loss of generality, these components can be taken to be equal to each other by multiplying columns of J_v by -1 if necessary. Multiplying a column of the Jacobian by -1 corresponds to redefining that axis of rotation or translation to be in the opposite direction, which does not essentially change the manipulator design. Hence, we can always assume that the null space of an optimally fault tolerant J_v is given by the unit length null vector $\hat{n}_{J_v} = \frac{1}{2}\,[1 \;\; 1 \;\; 1 \;\; 1]^T$. With this choice of $\hat{n}_{J_v}$ along with the isotropy condition $\sigma_i = \sigma = 1$, we have that

$$J_v^T J_v = I - \hat{n}_{J_v}\hat{n}_{J_v}^T = \begin{bmatrix} \frac{3}{4} & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4} & \frac{3}{4} & -\frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4} & -\frac{1}{4} & \frac{3}{4} & -\frac{1}{4} \\ -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & \frac{3}{4} \end{bmatrix}, \tag{49}$$

implying that the columns of J_v each have length $\sqrt{\frac{3}{4}}$ and the dot product of any two distinct columns is equal to $-\frac{1}{4}$. Furthermore, since fault tolerance dictates that any three columns of J_v are linearly independent, one can apply the QR factorization to determine a unique orthogonal matrix Q so that QJ_v is upper triangular with negative values along its main diagonal. Applying this orthogonal matrix merely rotates and/or reflects the base coordinate frame so that a rotated/reflected manipulator is obtained. The only 3 × 4 matrix satisfying these conditions, i.e., isotropy with $\sigma = 1$, a null vector with equal components, and an upper triangular form with negative components along the main diagonal, is (46). Hence, any optimally fault tolerant Jacobian J_v can be written as (46) by performing a series of suitable coordinate transformations.

The next section will illustrate how to characterize the set of all 6 × 4 Jacobian matrices that have the linear velocity portion given by J_v in (46). Once all these possible 6 × 4 Jacobians are determined, one will be able to determine the DH parameters for the physical robots.

6.3 CHARACTERIZING FAULT TOLERANT FOUR DOF SPATIAL POSITIONING MANIPULATORS

Our goal in this section is to determine all possible Jacobians of the form of (45) as a primary step to find out the DH parameters for the all corresponding physical robots.

6.3.1 Characterizing Orientational Velocity

The orientational velocity portion, J_ω, is somewhat arbitrary because it does not affect the positional fault tolerance properties.However, one must consider the constraint that each column of J_ω is orthogonal to the corresponding column of J_v. The ith column of J in (45) can be written as

$$j_i = \begin{bmatrix} v_i \\ \\ \omega_i \end{bmatrix}, \tag{50}$$

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

6.3.1.1 *Characterizing ω_1*

Let $\omega_1 = \begin{bmatrix} \omega_{11} & \omega_{21} & \omega_{31} \end{bmatrix}^T$. Because ω_1 and v_1 are orthogonal,

$$\omega_1^T v_1 = \begin{bmatrix} \omega_{11} & \omega_{21} & \omega_{31} \end{bmatrix} \begin{bmatrix} -\sqrt{\frac{3}{4}} \\ 0 \\ 0 \end{bmatrix} = 0, \tag{51}$$

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

$$\omega_1 = \begin{bmatrix} 0 \\ \cos(\beta_1) \\ \sin(\beta_1) \end{bmatrix}, \tag{52}$$

where β_1 can be any value between 0° and 360°.

6.3.1.2 Characterizing ω_2

Let $\omega_2 = \begin{bmatrix} \omega_{12} & \omega_{22} & \omega_{32} \end{bmatrix}^T$. By using the orthogonality condition, the dot product of ω_2 and v_2

$$\omega_2^T v_2 = \begin{bmatrix} \omega_{12} & \omega_{22} & \omega_{32} \end{bmatrix} \begin{bmatrix} \sqrt{\frac{1}{12}} \\ -\sqrt{\frac{2}{3}} \\ 0 \end{bmatrix} = 0, \tag{53}$$

so that

$$\sqrt{\frac{1}{12}}\omega_{12} - \sqrt{\frac{2}{3}}\omega_{22} = 0, \tag{54}$$

which leads to

$$\omega_{22} = \frac{\sqrt{\frac{1}{12}}}{\sqrt{\frac{2}{3}}}\omega_{12} = \sqrt{\frac{1}{8}}\omega_{12}. \tag{55}$$

From the condition that ω_2 is a normalized vector, $\omega_{12}^2 + \omega_{22}^2 + \omega_{32}^2 = 1$, it follows

$$\omega_{32}^2 = 1 - (\omega_{12}^2 + \omega_{22}^2) \tag{56}$$

Substituting (55) in (56) yields

$$\omega_{32} = \pm\sqrt{1 - \frac{9}{8}\omega_{12}^2}. \tag{57}$$

Therefore,

$$\omega_2 = \begin{bmatrix} \omega_{12} \\ \omega_{22} \\ \omega_{32} \end{bmatrix} = \begin{bmatrix} \omega_{12} \\ \sqrt{\frac{1}{8}}\omega_{12} \\ \pm\sqrt{1 - \frac{9}{8}\omega_{12}^2} \end{bmatrix}, \tag{58}$$

where $|\omega_{12}| \leq \frac{2\sqrt{2}}{3}$ as a result of constraining (57) to be a real number. The vector ω_2 rotates in a circle path that lays on a plane that is tangent to the z axis. One can find the plane by rotating the y-z plane about the z axis with an angle θ to make it lay on the plane of ω_2. A vector that rotates on the y-z plane can presented as:

$$\begin{bmatrix} 0 \\ \cos(\beta_2) \\ \sin(\beta_2) \end{bmatrix}, \tag{59}$$

where $\beta_2 \in [0°,\ 360°]$. When a rotation about the z axis by the amount of θ is applied to (59), ω_2 can be written as

$$\omega_2 = \begin{bmatrix} \omega_{12} \\ \sqrt{\frac{1}{8}}\omega_{12} \\ \pm\sqrt{1-\frac{9}{8}\omega_{12}^2} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ \cos(\beta_2) \\ \sin(\beta_2) \end{bmatrix} = \begin{bmatrix} -\sin(\theta)\cos(\beta_2) \\ \cos(\theta)\cos(\beta_2) \\ \sin(\beta_2) \end{bmatrix}. \tag{60}$$

To find the value of θ, one can solve for rows one and two in (60). From row one

$$\omega_{12} = -\sin(\theta)\cos(\beta_2), \tag{61}$$

and from the second row

$$\omega_{12} = \sqrt{8}\cos(\theta)\cos(\beta_2), \tag{62}$$

yields

$$\theta = \arctan(-\sqrt{8}). \tag{63}$$

The value of ω_{12} can be solved as a function of β_2 through the last row, so that

$$1 - \frac{9}{8}\omega_{12}^2 = \sin^2(\beta_2), \tag{64}$$

which can be simplified to

$$\omega_{12} = \frac{\sqrt{8}}{3}\cos(\beta_2) = \frac{2\sqrt{2}}{3}\cos(\beta_2). \tag{65}$$

Consequently, one can write (58) as a function of β_2 so that

$$\omega_2 = \begin{bmatrix} \frac{2\sqrt{2}}{3}\cos(\beta_2) \\ \frac{1}{3}\cos(\beta_2) \\ \sin(\beta_2) \end{bmatrix}. \tag{66}$$

6.3.1.3 Characterizing ω_3

If $\omega_3 = \begin{bmatrix} \omega_{13} & \omega_{23} & \omega_{33} \end{bmatrix}^T$, by using the orthogonality condition with $\mathbf{v}_3$,

$$\omega_{33} = \sqrt{\frac{1}{6}}\omega_{13} + \sqrt{\frac{1}{3}}\omega_{23}. \tag{67}$$

From the unity norm condition of ω_3 one can find that

$$\omega_{23} = \frac{-\sqrt{2}}{8}\omega_{13} \pm \frac{3}{8}\sqrt{\frac{16}{3} - 6\omega_{13}^2}. \tag{68}$$

From the square root part of (68),

$$|\omega_{13}| \leq \frac{2\sqrt{2}}{3}. \tag{69}$$

Similarly to ω_2, ω_3 can be parameterized with a rotation angle, β_3. One can rotate ω_3 by a rotation matrix R to make it lay on the x-y plane, such that

$$\mathrm{R}\omega_3 = \begin{bmatrix} \cos(\beta_3) \\ \sin(\beta_3) \\ 0 \end{bmatrix}, \tag{70}$$

which can be rewritten as

$$\omega_3 = \mathrm{R}^{\mathrm{T}} \begin{bmatrix} \cos(\beta_3) \\ \sin(\beta_3) \\ 0 \end{bmatrix}, \tag{71}$$

where $\beta_3 \in [0°, \ 360°]$, and R is

$$\mathrm{R} = \mathrm{R_x}(70.53°)\mathrm{R_z}(120°)\mathrm{R_y}(90°),^{2} \tag{72}$$

and $\mathrm{R_x}$, $\mathrm{R_y}$, and $\mathrm{R_z}$ are rotations around the x, y, and z axes respectively with the specified amount between the parenthesis. Therefore, (71) can be simplified such that

[2] $2\,\mathrm{acos}(\sqrt{\frac{2}{3}}) = 70.53°$

$$\omega_3 = \begin{bmatrix} 0 & \frac{2\sqrt{2}}{3} & \frac{1}{3} \\ -\frac{\sqrt{3}}{2} & -\frac{1}{6} & \frac{\sqrt{2}}{3} \\ -\frac{1}{2} & \frac{\sqrt{3}}{6} & \sqrt{\frac{2}{3}} \end{bmatrix} \begin{bmatrix} \cos(\beta_3) \\ \sin(\beta_3) \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{2\sqrt{2}}{3}\sin(\beta_3) \\ -\left(\frac{\sqrt{3}}{2}\cos(\beta_3) + \frac{1}{6}\sin(\beta_3)\right) \\ -\frac{1}{2}\cos(\beta_3) + \frac{\sqrt{3}}{6}\sin(\beta_3) \end{bmatrix}. \tag{73}$$

6.3.1.4 Characterizing ω_4

When the 3rd and the 4th columns in (46) are compared, one can note that only the last element's sign of the 3rd column changes. Consequently, applying the same steps of computing ω_3, one can find that the solution of (73) can be used with only alternating the last element sign and replacing β_3 with another variable, β_4, which is also $\in [0°,\ 360°]$. Hence,

$$\omega_4 = \begin{bmatrix} \frac{2\sqrt{2}}{3}\sin(\beta_4) \\ -\left(\frac{\sqrt{3}}{2}\cos(\beta_4) + \frac{1}{6}\sin(\beta_4)\right) \\ -\left(-\frac{1}{2}\cos(\beta_4) + \frac{\sqrt{3}}{6}\sin(\beta_4)\right) \end{bmatrix}. \tag{74}$$

6.3.2 Characterizing DH Parameters

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

$$\alpha_i = f_{\alpha_i}(\beta_i, \beta_{i+1}) \quad \text{and} \tag{75}$$

$$a_i = f_{a_i}(\beta_i, \beta_{i+1}). \tag{76}$$

For example, Figs. 16 and 17 show how the joint twist and length parameters of joint 1, α_1 and a_1, vary as a function of β_1 and β_2. Note that there is considerable flexibility in selecting these two joint parameters, i.e., the twist angle can be set anywhere from 0° to 180° and the magnitude of the

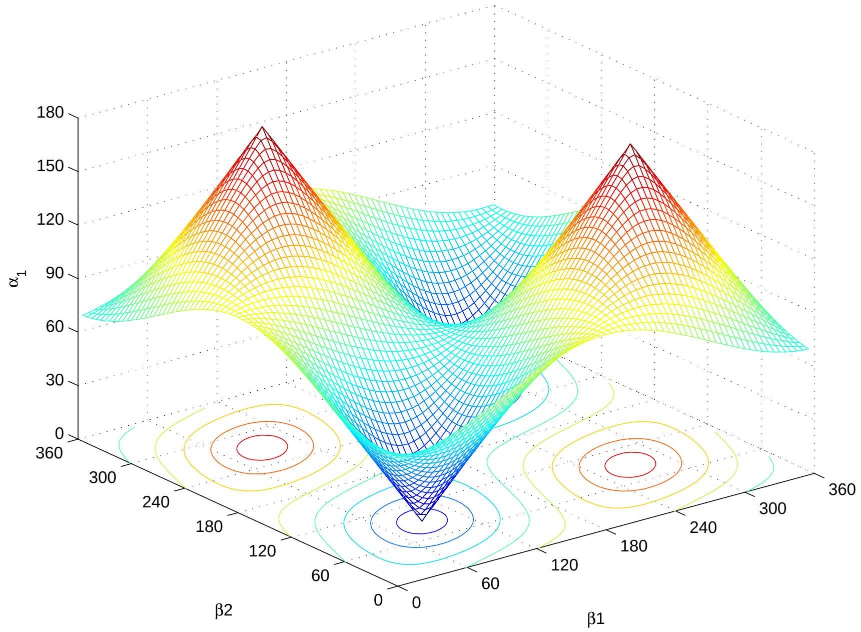


Figure 16: The relationship between the first link twist angle α_1 and the parameters β_1 and β_2.

length can be anywhere from 0 to $\sqrt{3}$. Because the tool, i.e., fifth, coordinate frame is arbitrary, it is assumed to be in the same orientation as the fourth so that

$$\alpha_4 = 0 \quad \text{and} \tag{77}$$

$$a_4 = \sqrt{3}/2. \tag{78}$$

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

$$\theta_i = f_{\theta_i}(\beta_{i-1}, \beta_i, \beta_{i+1}) \quad \text{and} \tag{79}$$

$$d_i = f_{d_i}(\beta_{i-1}, \beta_i, \beta_{i+1}). \tag{80}$$

For the first coordinate frame, θ_1 and d_1 are arbitrary so they can be assumed to be zero because one can select the orientation of the zeroth coordinate frame. At the fourth coordinate frame, the

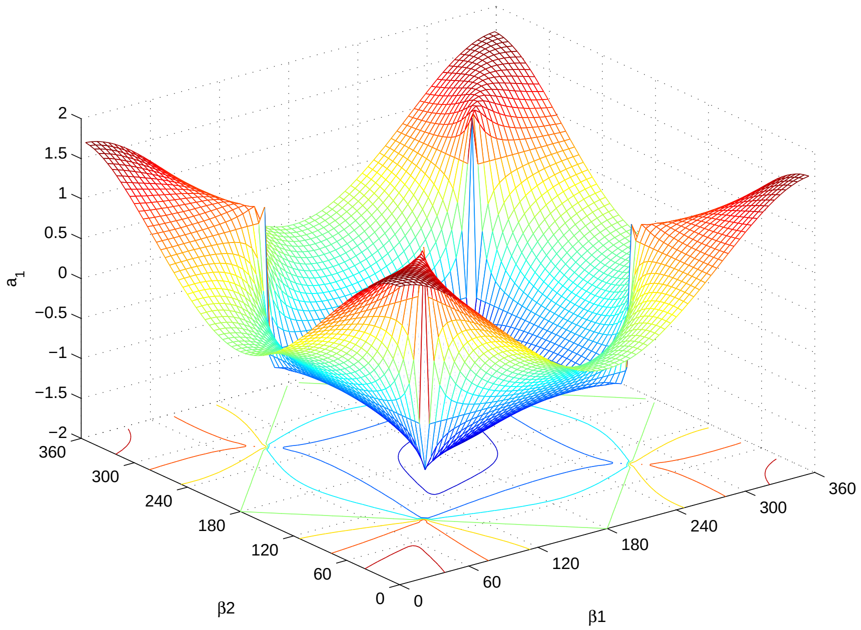


Figure 17: The relationship between first link length a_1 and the parameters β_1 and β_2.

joint parameters are not functions of the fifth coordinate frame, i.e.,

$$\theta_4 = f_{\theta_4}(\beta_3, \beta_4) \quad \text{and} \tag{81}$$

$$d_4 = f_{d_4}(\beta_3, \beta_4), \tag{82}$$

because it is selected to be aligned with the fourth.

The exact values of the DH parameters for a given set of β_i's can be computed using the algorithm that is presented in [1]. Clearly, there is an infinite family of robots that correspond to (46). The next section will discuss how to compute the global failure tolerance properties of the various possible physical robots that can be generated from this single optimal Jacobian.

6.4 COMPUTING GLOBAL FAULT TOLERANCE PROPERTIES

Different combinations of (β_1, β_2, β_3, β_4) correspond to different potential robots (in terms of their DH parameters). While these robots have the same desired optimal local fault tolerant design point, they are quite different in terms of their global properties. Not only is the size of the workspace quite different, but more importantly if one is concerned with fault tolerance, there is considerable difference in how the value of the fault tolerance measure varies away from the design point.

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

6.4.1 Volume Estimation Using Mont Carlo Integration Algorithm

One of the best techniques for computing an estimate for the volume of an unknown three-dimensional shape is to use the Mont Carlo integration algorithm (see Figure 18). First, one million uniformly distributed random configurations are generated in the joint space, where $0 \leq \theta_i < 2\pi$ for all i, that are transformed to the workspace using forward kinematics. Then the maximum reach of the manipulator is estimated by picking the point with the largest norm and using inverse kinematics to drive the robot until its Jacobian is singular and a workspace boundary is reached. A sphere whose radius R is 110% of this maximum reach is used as the boundary for our Monte Carlo integration. Then 10,000 uniformly distributed random points are generated within this sphere and they are determined if they are reachable. Reachability is determined by performing iterative inverse kinematics starting with the joint configuration (from the one million configurations initially generated)[3] whose workspace location is the closest to the Monte Carlo point being evaluated. An

[3] In this work, they are sorted in their positional space, only on x-y plane, to make 10×10 rectangular prisms to speed up the searching through them. One can use a different sorting scheme. However, It was found that the sorting

estimate for the total reachable workspace volume is then given by

$$V_{\mathrm{r}} = \left(\frac{n_{\mathrm{r}}}{10,000}\right)\frac{4}{3}\pi R^3, \tag{83}$$

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

6.4.2 Computing the Maximum Value of $\mathcal{K}$

Determining the maximum value of $\mathcal{K}$ for a workspace location is complicated by the fact that one must consider all of the possible pre-images for this location. (See Appendix B for determining the maximum $\mathcal{K}$ on a manifold.) Even determining the number of manifolds associated with a location, and whether these manifolds are open or closed, is not trivial. For example, consider the case illustrated in Figure 19 that shows all the configurations for one optimal robot design at four distinct locations (identified by A, B, C, and D) that lie on a straight line segment within the workspace. (Points A and D represent the two endpoints of the line segment with B and C being 57% and 80% of the distance from A to D, respectively.) The possible configurations for being at point A are represented by two open manifolds, which then combine into a single closed manifold at point B, which then becomes two disjoint manifolds (as illustrated by point C), with one of them ultimately disappearing at point D. This illustrates that one cannot simply track a locally optimal value of $\mathcal{K}$ because these local optima may disappear, as they do in this example. In order to accurately compute the maximum value of $\mathcal{K}$ for a workspace location, one must make sure to consider all of the possible configurations associated with all of the possible manifolds for that location.

To deal with this situation, several joint configurations are selected (from the one million random configurations initially generated)[4] so that their workspace locations are near the reachable

in 10×10×10 cubes increased the computation time.

[4] In this work, five configurations are used.

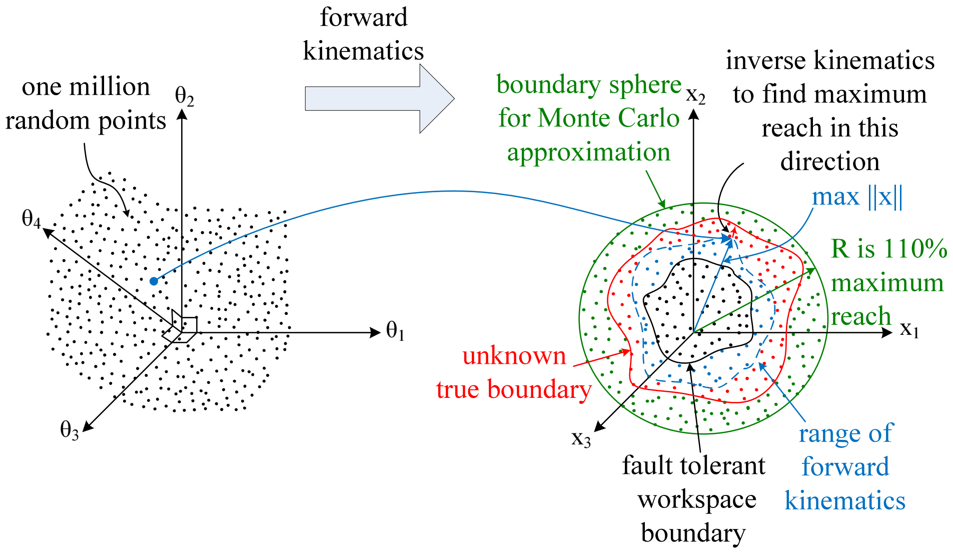


Figure 18: An illustration of how the Monte Carlo integration algorithm is used to compute the volume of both the fault tolerant workspace and the entire reachable workspace.

workspace point whose maximum value of $\mathcal{K}$ is being evaluated. This increases the probability that all self-motion manifolds associated with this workspace location will be represented. (See Figure 20.) Then iterative inverse kinematics is performed to drive each of these configurations exactly to the point being evaluated. Once the exact end effector position is achieved, the entire self-motion manifold is mapped out by stepping along the null vector of the Jacobian. As we compute the configurations along the manifold, each configuration's value of $\mathcal{K}$ is calculated and the maximum is saved. If maximum value of $\mathcal{K}$ over all manifolds associated with this point is greater than $\gamma\ \mathcal{K}_{\max}$, then this reachable workspace point is included in the count for the fault tolerant workspace volume, denoted n_{f}.[5] The fraction of the total workspace that is fault tolerant, denoted $W_{\mathcal{K}}$, can then be easily estimated by the ratio $n_{\mathrm{f}}/n_{\mathrm{r}}$.

The implementation of this section's computations was done in C++ with using Armadillo open source C++ linear algebra library [62]. Depending on the robot structure, the running time of computing the above global properties takes 10-30 minutes. The next section will use this

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

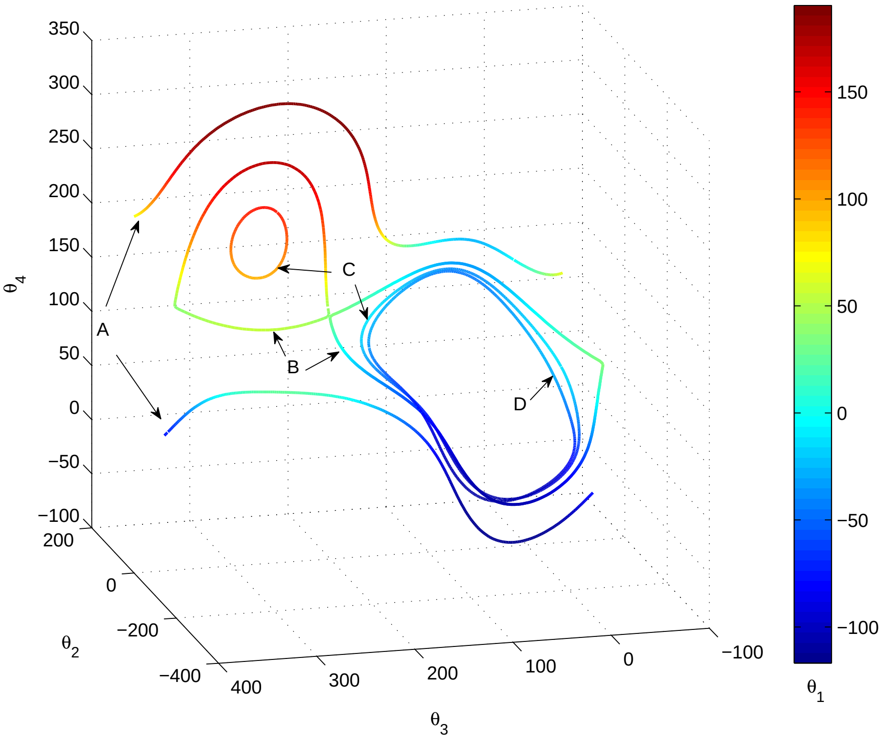


Figure 19: Self-motion manifolds for the robot generated from $(\beta_1,\ \beta_2,\ \beta_3,\ \beta_4) = (200^\circ,\ 220^\circ,\ 130^\circ,\ 90^\circ)$ where the value of θ_1 is indicated using color. The points A, B, C, and D lie on a line segment in the workspace where A is one endpoint located at $[x, y, z] = [-0.04,\ -0.82,\ 0.22]$, D is the other endpoint at $[x, y, z] = [-0.16, -1.79, 0.47]$, and points B and C are located 57% and 80% of the way from A to D, respectively.

measure to evaluate families of robots that have commonly used values for link twists.

6.5 EXAMPLES OF MANIPULATORS WITH COMMON LINK TWIST PARAMETERS

6.5.1 Introduction

When designing a robot's kinematics there are many factors that must be considered. These factors may limit the range of desirable values for the joint parameters. This section uses the example of where one may be interested in limiting the joint twist values, i.e., setting α_i's to $\pm 90^\circ$, 0°, or 180° as is common in many commercial manipulators. (Because α_4 is already set to zero (see (77)), one only needs to consider $i = 1, 2, 3$.)

Recall that the parameter α_i is defined as the angle between the rotation axes of joints i and $i + 1$, which is the same as ω_i and ω_{i+1}, respectively. Therefore, one can use the dot product

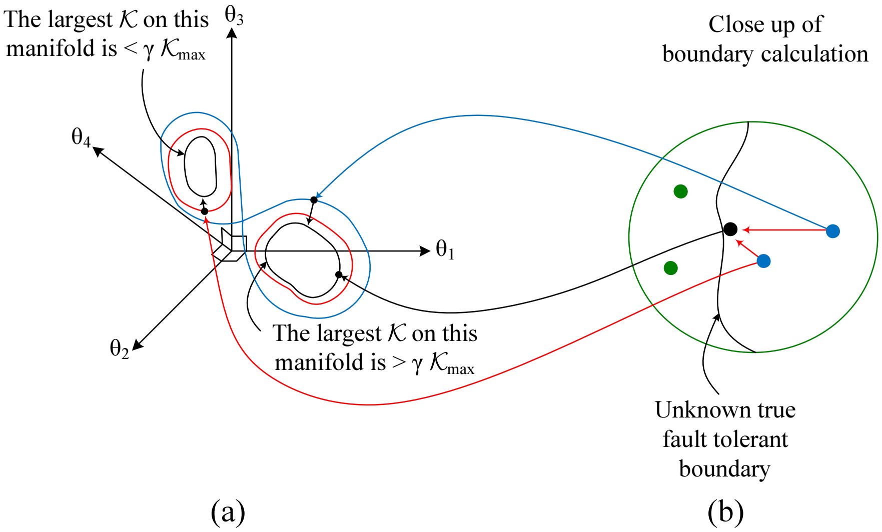

The largest $\mathcal{K}$ on this manifold is < γ $\mathcal{K}_{max}$
θ3
θ4
θ1
θ2
The largest $\mathcal{K}$ on this manifold is > γ $\mathcal{K}_{max}$
Close up of boundary calculation
Unknown true fault tolerant boundary
(a)
(b)

Figure 20: To evaluate the maximum value of $\mathcal{K}$ at a workspace location (shown in black in (b)) multiple randomly generated configurations (two of which are shown in blue) are used in order to evaluate the multiple self-motion manifolds associated with a workspace location. The configuration that corresponds to the largest value of $\mathcal{K}$ for each of these workspace locations is shown in (a). Note that the closest random point may not be associated with the manifold that has the largest value of $\mathcal{K}$.

between the appropriate pair of equations (52), (66), (73), and (74) to determine the values of the β_i's that result in the desired α_i's.

6.5.2 Link twist $\alpha_i = \pm 90°$

To determine the relationship between β_i and β_{i+1} for $\alpha_i = \pm 90°$ one can solve for the case where the dot product between ω_i and ω_{i+1} is equal to zero. In the case of $\alpha_1 = \pm 90°$, using (52) and (66) results in

$$\beta_2 = -\arctan\left(\frac{1}{3\tan(\beta_1)}\right) + k\pi, \tag{84}$$

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

Similarly, one can find the solutions of when α_2 and α_3 are equal to $\pm 90°$ using (66), (73), and

(74) so that

$$\beta_3 = \arctan\left(\frac{\sqrt{3}+3\tan(\beta_2)}{5+\sqrt{3}\tan(\beta_2)}\right) + k\pi \tag{85}$$

and

$$\beta_4 = -\arctan\left(\frac{3+\sqrt{3}\tan(\beta_3)}{5\tan(\beta_3)+\sqrt{3}}\right) + k\pi, \tag{86}$$

where $k = 0, 1$.

Note that when $k = 1$ the direction of ω_{i+1} flips from what it was when $k = 0$ and thus the sign of α_i will change. However, one can add 180° to β_i with $k = 1$, so that both ω_i and ω_{i+1} are flipped, and the sign of α_i will stay the same. Thus there will be two sets of values for β_2 and β_3 that will satisfy (85) and result in $\alpha_2 = 90°$ and two sets of values for β_2 and β_3 that will satisfy (85) and result in $\alpha_2 = -90°$. The same is true for (86).

6.5.3 Link twist $\alpha_i = 0°$ or $180°$

To solve for α_i being equal to 0° or 180°, one can set the dot product of ω_i and ω_{i+1} to 1 or -1, respectively. In contrast to the section above, the solutions of these equations result in discrete values of β_i and β_{i+1}. Table 5 shows the values of β_i and β_{i+1} for all cases of $\alpha_i = 0°$ or $\alpha_i = 180°$. Note that there are two sets of β_i and β_{i+1} for a given desired α_i, except for α_1 where there is only one set of β_1 and β_2. (This is because the other two sets of β_1 and β_2 result in mirror versions of the corresponding robots.)

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

Table 5: Values of β_i and β_{i+1} to make $\alpha_i = 0°$ or 180°

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

6.5.4 Manipulator Categories

The above subsections provide the equations for determining the required β_i and β_{i+1} values to achieve a desired α_i, i.e., solving equation (75) for $\alpha_i = \pm 90°, 0°$, and $180°$. However, because selecting a desired α_i restricts the range of possible values for β_i *and* β_{i+1} it is not possible to arbitrarily select all three of the α_i values. For example, if one selects the value of α_1 and α_3, the values of $\beta_1, \beta_2, \beta_3$, and β_4 are all specified so that the choices for α_2 are limited. This subsection determines if a particular combination of α_i's results in a feasible kinematic design with the desired fault tolerant Jacobian and, if so, how many such designs exist.

Even with restricting α_i to $\pm 90°, 0°$, or $180°$ there still exists a large number of possible robots. In order to analyze them further, they are organized into groups. Because $\alpha_4 = 0$ the total number of different combinations of setting α_i to one of these four values is $4^3 = 64$. These 64 combinations are organized into eight robot groups, based on whether an α_i results in adjacent joint axes being parallel ($\|$) or perpendicular ($\perp$), i.e., whether $\alpha_i = 0°, 180°$ or $\alpha_i = \pm 90°$, respectively. Table 6 enumerates these eight groups along with the total number of robots that they include.

Note that some robot structures are not feasible because they are not physically able to result in the optimally fault tolerant Jacobian given by (46). For example, robots in group 1 have all of their joints parallel so that they result in planar manipulators, which clearly are not capable of the desired Jacobian. Likewise, robots in groups 2 and 3 result in planar substructures that make it physically impossible to achieve the desired Jacobian.

For cases where there are not three joint axes in parallel, one is able to identify multiple feasible robot designs. For groups 4-7, there exist eight unique robots in each group. To determine all of the β_i parameters that result in the specified joint twist values (α_i's) one can use Table 5 and equations (84)-(86). In some cases the inverses of (84) and (85), i.e.,

$$\begin{aligned} \beta_1 &= -\arctan\left(\frac{1}{3\tan(\beta_2)}\right) \quad \text{and} && (87) \\ \beta_2 &= -\arctan\left(\frac{\sqrt{3}-5\tan(\beta_3)}{3-\sqrt{3}\tan(\beta_3)}\right) + k\pi, && (88) \end{aligned}$$

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

This procedure is illustrated with a specific example. Consider robot group 6. Because joint axes 2 and 3 are parallel, i.e., $\alpha_2 = 0^\circ, 180^\circ$, β_2 and β_3 are constrained to the discrete values given in Table 5. Therefore, our strategy is to start with each of these possible values and then evaluate the equations (87) and (86) to determine the required values of β_1 and β_4. Specifically, for $\alpha_2 = 0^\circ$, $(\beta_2, \beta_3) = (30^\circ, 120^\circ)$ or $(210^\circ,\ 300^\circ)$. Then to set α_1 and α_3 to $\pm 90^\circ$, equations (87) and (86) are used respectively. This results in

$$\begin{aligned} (\beta_1, \beta_2, \beta_3, \beta_4) &= (150^\circ, 30^\circ, 120^\circ, 0^\circ) \quad \text{or} && (89) \\ &= (150^\circ, 30^\circ, 120^\circ, 180^\circ) \text{ or} \\ &= (150^\circ, 210^\circ, 300^\circ, 0^\circ) \quad \text{or} \\ &= (150^\circ, 210^\circ, 300^\circ, 180^\circ) \end{aligned}$$

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

Finally, robot group 8 is unique in that it is parameterized by β_1 using equations (84)-(86) and so there are an unlimited number of robots in this group. In the next subsection all of the robots

Table 6: The size of robot groups with all possible combinations of α_i being ±90°, 0°, or 180°

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

Table 7: The best robot configurations among the individual robot groups 4-7 that have one additional $\alpha_i = 0$ in Table 6

Robot Group	$(\alpha_1,\ \alpha_2,\ \alpha_3,\ \alpha_4)$ [degrees]	$(\beta_1,\ \beta_2,\ \beta_3,\ \beta_4)$ [degrees]	$W_{\mathcal{K}}$ [%]	max reach [m]
6	$(-90,\ 0,\ -90,\ 0)$	(150, 30, 120, 180)	67	3.92
4	(0, 90, 0, 0)	(90, 90, 240, 240)	59	5.19
7	(90, 90, 0, 0)	(0, 270, 240, 240)	48	3.93
5	(0, 90, 90, 0)	(90, 90, 240, 330)	30	4.96

from these groups are evaluated to identify an optimal design in terms of its global fault tolerance capabilities.

6.5.5 Global Fault Tolerance Analysis

The global fault tolerant properties were computed for all of the robots represented in Table 6 . For robot groups 4-7, this simply involved evaluating each of the possible robot designs for $W_{\mathcal{K}}$. (Recall that $W_{\mathcal{K}}$ is the percentage of the workspace that has a fault tolerance value that is greater than or equal to $\gamma = 0.9$ of the maximum.) For each group the robot (in terms of its β_i's parameters) that had the largest value of $W_{\mathcal{K}}$ was determined. These results are shown in Table 7 along with a depiction of the resulting robots in the optimal fault tolerant design configuration in Figure 21. Note that best robot from each group vary from a maximum of $W_{\mathcal{K}} = 67\%$ to a minimum of 30%. (Appendix D presents the results of the 32 robots for groups 4-7.)

In contrast, determining the best robot from group 8 required an optimization over the independent variable β_1. However, this resulted in by far the best global fault tolerant robot with a

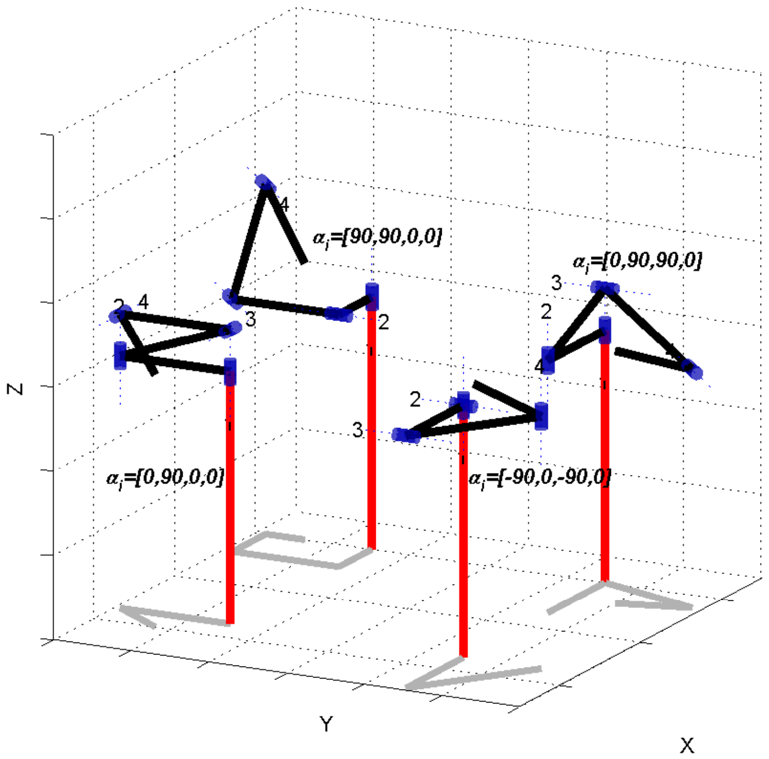


Figure 21: The optimal fault tolerant configurations at the design point of Table 7 robots. (The figure was generated using the Robotics Toolbox described in [61].)

$W_{\mathcal{K}} = 75\%$ for the robot given by $(\beta_1,\ \beta_2,\ \beta_3,\ \beta_4) = (30°,\ 330°,\ 0°,\ 300°)$, which corresponds to

$$J_\omega = \begin{bmatrix} 0 & \sqrt{\frac{2}{3}} & 0 & -\sqrt{\frac{2}{3}} \\ \sqrt{\frac{3}{4}} & \sqrt{\frac{1}{12}} & -\sqrt{\frac{3}{4}} & -\sqrt{\frac{1}{12}} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{bmatrix}. \tag{90}$$

The DH parameters for this robot are given in Table 8 and an image of the robot in its optimal design configuration is shown in Figure 22, where the base coordinate frame has been rotated to

Table 8: The DH parameters of the globally optimal robot that corresponds to $(\beta_1,\ \beta_2,\ \beta_3,\ \beta_4) = (30^\circ,\ 330^\circ,\ 0^\circ,\ 300^\circ)$

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

align the first joint axis with the z-axis, resulting in the following Jacobian

$$J_{6\times4} = \begin{bmatrix} J_v \\ \\ J_\omega \end{bmatrix} = \left[\begin{array}{cccc} -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ \hdashline 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{array}\right]. \tag{91}$$

This globally optimal fault tolerant robot design has a total maximum reach and reachable volume, V_r, of 5.5 m and 560 m^3, respectively. An illustration of both the reachable and the fault tolerant workspace are given in Figure 23 where the three-dimensional volume is shown with multiple cross-sections to better visualize the two different volumes. Clearly one can see that a $W_\mathcal{K} = 75\%$ results in a significant amount of the reachable volume being fault tolerant as well.

6.6 JACOBIAN COLUMN PERMUTATION AND/OR SIGN CHANGE EFFECTS

6.6.1 Introduction

The analysis so far has been based on the positional Jacobian J_v given in (46). The family of manipulator Jacobians with a compatible orientational Jacobian J_ω was characterized based on parameterizing the rotation axes with β_i, $i = 1, 2, 3, 4$ given in equations (52), (66), (73), and (74). There are $4! \times 2^4 = 384$ different versions of (46), where $4! = 24$ represents the total number of permutation operations and 2^4 is the number of all possible cases of alternating the column signs. Performing these operations, particularly permuting the columns, on an arbitrary Jacobian will typically result in a Jacobian that corresponds to a completely different manipulator. However, this section will illustrate that these operations do not collectively change the family of fault-tolerant robots identified using (46). To investigate the effects that these two types of operations have on the resulting manipulator geometry, each will be studied individually.

The notation $J_{v_{1234}}$ will be used to indicate the positional Jacobian in (46), which is in standard form. The Jacobian resulting from permutations and multiplication of columns by -1 will then

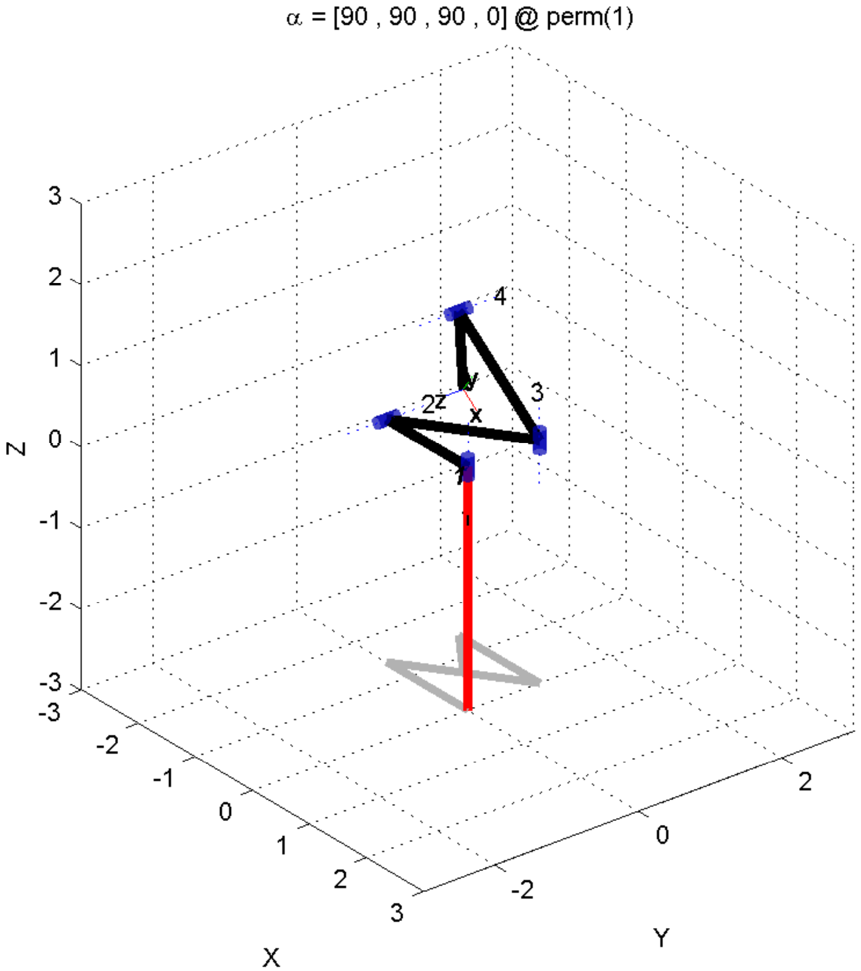


Figure 22: The locally optimal fault tolerant configuration at the design point of the globally optimal robot defined in Table 8. (The figure was generated using the Robotics Toolbox described in [61].))

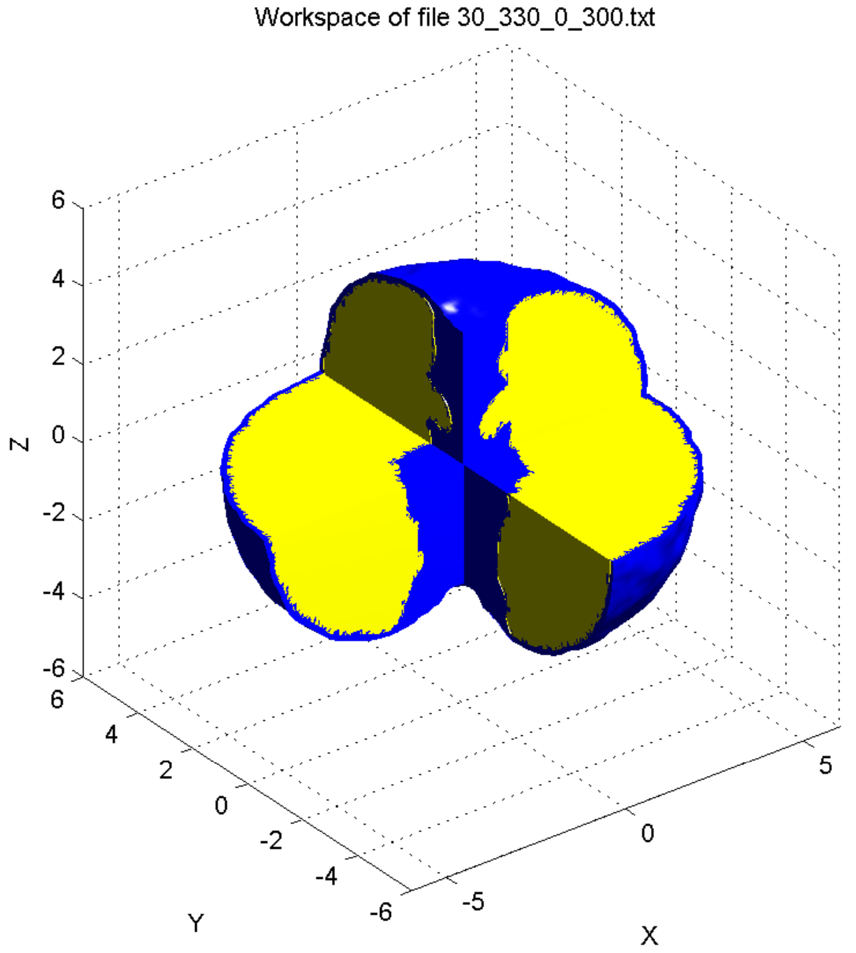


Figure 23: The workspace volumes of the globally optimal robot defined in Table 8. The blue color indicates the total reachable workspace volume, while the yellow color presents the fault tolerant workspace volume.

be denoted by permuting the corresponding indices and inserting appropriate minus signs. For example, the permutation corresponding to swapping the second and third columns will be written as $J_{v_{1324}}$ and a change in the sign of column three will be written as $J_{v_{12(-3)4}}$. A fully spatial Jacobian will be denoted J_{1234} and column permutations and multiplications by -1 will be denoted in the same way for J_{1234}, for example, $J_{13(-2)4}$ will correspond to multiplying the second column of J_{1234} by -1 and swapping it with the third column.

It should be noted that unlike J_v, the orientational Jacobian J_ω depends on $(\beta_1, \beta_2, \beta_3, \beta_4)$.

When it is important to point out this distinction, J_{1234} will be explicitly written as

$$J_{1234}(\beta_1, \beta_2, \beta_3, \beta_4) = \left[\begin{array}{c} J_{v1234} \\ \hdashline J_{\omega_{1234}}(\beta_1, \beta_2, \beta_3, \beta_4) \end{array}\right]. \tag{92}$$

6.6.2 Sign Change Effect

It turns out that multiplying one or more columns of a Jacobian by -1 has no real effect on the geometry of the corresponding robot. Essentially, what happens is that the direction of the corresponding joint axis is reversed. If one multiplies one or more columns of J_v by -1, the operation is equivalent to modifying the corresponding columns of J_ω by adding 180° to the corresponding β_i.

6.6.3 Column Permutations and Tetrahedral Symmetry

Next, the effects of the 24 possible column permutations of (46) are considered. Because of the nice column structure of (46), there is a particularly elegant geometric interpretation for permuting the columns of $J_{v_{1234}}$. This follows by noting that the columns of $J_{v_{1234}}$ correspond to a regular tetrahedron. To make this clearer, let P_i denote the vertex corresponding to v_i, the i-th column of J_v. For example, the first vertex P_1 is given by the first column of J_v and hence lies on the negative x-axis at the coordinate $\begin{bmatrix} -\sqrt{3}/2 & 0 & 0 \end{bmatrix}^T$. The six edges will be denoted as $\overline{P_iP_j}$, $i < j$, so that $\overline{P_1P_2}$ denotes the edge connecting the first and second vertices.

This simple geometric interpretation allows us to use a well-known relationship between relabeling the vertices of a regular tetrahedron and rotating and/or reflecting the tetrahedron, called *tetrahedral symmetry.* Among the 24 permutations or relabelings of the vertices, 12 correspond to rotations of the tetrahedron and the other 12 correspond to reflections or combinations of reflections and rotations, called rotoreflections. Physically, the fact that any relabeling can be achieved through rotations and reflections would require that the distances between any two vertices, i.e., the lengths of the edges, are all equal, which is clearly true for a regular tetrahedron.

Rotations in $\mathbb{R}^3$ can be characterized by using the axis-angle notation where

$$R_{\hat{k}}(\theta) = \hat{k}\hat{k}^T + \cos\theta(I - \hat{k}\hat{k}^T) + \sin\theta S(\hat{k}) \tag{93}$$

represents a counterclockwise rotation of θ radians about the $\hat{k}$ axis, where $\hat{k} = \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix}^T$ is a unit vector and

$$S(\hat{k}) = \begin{bmatrix} 0 & -k_3 & k_2 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix} \tag{94}$$

represents the cross product operation $S(\hat{k})u = \hat{k} \times u$ for any vector u in $\mathbb{R}^3$. A rotoreflection $Q_{\hat{k}}(\theta)$ corresponds to a counterclockwise rotation of θ radians about the unit vector $\hat{k}$ followed by a reflection about the plane through the origin perpendicular to $\hat{k}$ and is given by

$$Q_{\hat{k}}(\theta) = -\hat{k}\hat{k}^T + \cos\theta(I - \hat{k}\hat{k}^T) + \sin\theta S(\hat{k}). \tag{95}$$

An important special case is a pure reflection $F_k = Q_{\hat{k}}(0) = I - 2\hat{k}\hat{k}^T$ about the plane through the origin and perpendicular to $\hat{k}$ (and the case $Q_{\hat{k}}(\pi) = -I$).

The relationship between rotations and column permutations of $J_{v_{1234}}$ is given in Table 9 and can be best visualized using the regular tetrahedron described above. One can see that the axis of rotation taking $J_{v_{1234}}$ to $J_{v_{1342}}$ and $J_{v_{1423}}$ is given by $\hat{k} = \hat{v_1}$ with rotation angles of 120° and 240°, respectively, where $\hat{v_1} = v_1/\|v_1\|$ denotes the normalized version of the first column v_1 of $J_{v_{1423}}$. These two rotations about the axis connecting the origin to the vertex P_1 correspond to the cyclic permutations of vertices P_2, P_3, and P_4, with the first vertex P_1, of course, remaining fixed. The next six rows correspond to rotations about the other axes connecting the origin to the other three vertices. The last three rows of Table 9 are more complicated and correspond to rotations about the line connecting the midpoints of two non-adjacent edges. For example, it can be shown

Table 9: Permuted Jacobians that are related to J_{1234} by a rotation $R_k(\theta) = kk^T + \cos(\theta)[I - kk^T] + \sin(\theta)S(k)$

	Permuted Jacobian	Rotation Axis (k)	θ [degrees]
Identity			
1	J_{1234}	$[-1,\ 0,\ 0]^T$	0
rotation of $\pm 120°$ about $\hat{v_i}$ axis			
2	J_{1423}	$\hat{v_1} = [-1,\ 0,\ 0]^T$	120
3	J_{1342}	$\hat{v_1} = [-1,\ 0,\ 0]^T$	240
4	J_{3241}	$\hat{v_2} = [\frac{1}{3},\ -\frac{2\sqrt{2}}{3},\ 0]^T$	120
5	J_{4213}	$\hat{v_2} = [\frac{1}{3},\ -\frac{2\sqrt{2}}{3},\ 0]^T$	240
6	J_{4132}	$\hat{v_3} = [\frac{1}{3},\ \frac{\sqrt{2}}{3},\ -\sqrt{\frac{2}{3}}]^T$	120
7	J_{2431}	$\hat{v_3} = [\frac{1}{3},\ \frac{\sqrt{2}}{3},\ -\sqrt{\frac{2}{3}}]^T$	240
8	J_{2314}	$\hat{v_4} = [\frac{1}{3},\ \frac{\sqrt{2}}{3},\ \sqrt{\frac{2}{3}}]^T$	120
9	J_{3124}	$\hat{v_4} = [\frac{1}{3},\ \frac{\sqrt{2}}{3},\ \sqrt{\frac{2}{3}}]^T$	240
rotation of 180° about $\frac{\hat{v}_1+\hat{v}_i}{\lVert\hat{v}_1+\hat{v}_i\rVert}$ axis, $i = 2, 3, 4$			
10	J_{2143}	$\frac{\hat{v_1}+\hat{v_2}}{\lVert\hat{v_1}+\hat{v_2}\rVert} = [-\sqrt{\frac{1}{3}},\ -\sqrt{\frac{2}{3}},\ 0]^T$	180
11	J_{3412}	$\frac{\hat{v_1}+\hat{v_3}}{\lVert\hat{v_1}+\hat{v_3}\rVert} = [-\sqrt{\frac{1}{3}},\ \sqrt{\frac{1}{6}},\ -\sqrt{\frac{1}{2}}]^T$	180
12	J_{4321}	$\frac{\hat{v_1}+\hat{v_4}}{\lVert\hat{v_1}+\hat{v_4}\rVert} = [-\sqrt{\frac{1}{3}},\ \sqrt{\frac{1}{6}},\ \sqrt{\frac{1}{2}}]^T$	180

that $J_{v_{2143}}$ is given by rotating 180° about the line connecting the midpoints of the $\overline{P_1P_2}$ and $\overline{P_3P_4}$ edges, which happens to be given by the vector $v_1 + v_2$.

The permuted Jacobians given in Table 10 are obtained from $J_{v_{1234}}$ in a similar but slightly more complicated manner. The first six rows are obtained by a pure reflection about a corresponding plane. For example, $J_{v_{2134}}$ is obtained by reflecting about the plane containing the midpoint of the $\overline{P_1P_2}$ edge and the third and fourth vertices. The corresponding perpendicular to this plane is the unit vector pointing from the second vertex to the first, i.e., $\hat{k} = (v_1 - v_2)/\lVert v_1 - v_2 \rVert$. The remaining six rows are even more complicated and are given by a rotoreflection of ±90° about the axes connecting the midpoints of two non-touching edges. Once again, these three axes are given in the last three lines of Table 9.

The exact nature of the rotations and reflections are not so important as the fact that (1) they are rotations/reflections and (2) that such operations do not change a manipulator's geometry. To see this, one can observe that

$$J_{1423}(\beta_1, \beta_4, \beta_2, \beta_3) = \left[\begin{array}{c} R_{1423} J_{v_{1234}} \\ \hdashline R_{1423} J_{\omega_{1234}}(\beta_1', \beta_2', \beta_3', \beta_4') \end{array}\right], \tag{96}$$

where the β_i's are given by the corresponding equation in (52), (66), (73), or (74), R_{1423} is the rotation matrix taking $J_{v_{1234}}$ to $J_{v_{1423}}$, and β_1', β_2', β_3', and β_4' are suitable β_i parameters for J_{1234}. In other words, given any J_{1423} generated from $J_{v_{1423}}$, there is a suitable rotated version of a corresponding J_{1234}. The β_i parameters will typically be different than the β_i' parameters. Manipulator Jacobians corresponding to other permutations of $J_{v_{1234}}$ are obtained in a similar manner. The key point is that generating the family of manipulator Jacobians from a permuted version of (46) results in a rotated/reflected version of the family that was generated from (46). Hence, the robot geometries do not essentially change. Consequently, one need only consider the unmodified Jacobian described by (46) with J_ω as a function of the β_i's to optimize the whole family of robots. Figure 24 illustrates the kinematic equivalency of these various robots.

Table 10: Permuted Jacobians that are reflections of Table 9 by $Q_k(\theta) = -kk^T + \cos(\theta)[I - kk^T] + \sin(\theta)S(k)$

	Permuted Jacobian	Rotation & Reflection Axis (k)	θ [degrees]
pure reflections $F_k = I - 2kk^T = Q_k(0)$			
1	J_{2134}	$\frac{\hat{v_1}-\hat{v_2}}{\lVert\hat{v_1}-\hat{v_2}\rVert} = [-\sqrt{\frac{2}{3}},\ \sqrt{\frac{1}{3}},\ 0]^T$	0
2	J_{3214}	$\frac{\hat{v_1}-\hat{v_3}}{\lVert\hat{v_1}-\hat{v_3}\rVert} = [-\sqrt{\frac{2}{3}},\ -\sqrt{\frac{1}{12}},\ \frac{1}{2}]^T$	0
3	J_{4231}	$\frac{\hat{v_1}-\hat{v_4}}{\lVert\hat{v_1}-\hat{v_4}\rVert} = [-\sqrt{\frac{2}{3}},\ -\sqrt{\frac{1}{12}},\ -\frac{1}{2}]^T$	0
4	J_{1324}	$\frac{\hat{v_2}-\hat{v_3}}{\lVert\hat{v_2}-\hat{v_3}\rVert} = [0,\ -\sqrt{\frac{2}{3}},\ \sqrt{\frac{1}{3}}]^T$	0
5	J_{1432}	$\frac{\hat{v_2}-\hat{v_4}}{\lVert\hat{v_2}-\hat{v_4}\rVert} = [0,\ -\sqrt{\frac{2}{3}},\ -\sqrt{\frac{1}{3}}]^T$	0
6	J_{1243}	$\frac{\hat{v_3}-\hat{v_4}}{\lVert\hat{v_3}-\hat{v_4}\rVert} = [0,\ 0,\ -1]^T$	0
rotoreflections $Q_k(\theta)$			
7	J_{3421}	$\frac{\hat{v_1}+\hat{v_2}}{\lVert\hat{v_1}+\hat{v_2}\rVert} = [-\sqrt{\frac{1}{3}},\ -\sqrt{\frac{2}{3}},\ 0]^T$	90
8	J_{4312}	$\frac{\hat{v_1}+\hat{v_2}}{\lVert\hat{v_1}+\hat{v_2}\rVert} = [-\sqrt{\frac{1}{3}},\ -\sqrt{\frac{2}{3}},\ 0]^T$	-90
9	J_{4123}	$\frac{\hat{v_1}+\hat{v_3}}{\lVert\hat{v_1}+\hat{v_3}\rVert} = [-\sqrt{\frac{1}{3}},\ \sqrt{\frac{1}{6}},\ -\sqrt{\frac{1}{2}}]^T$	90
10	J_{2341}	$\frac{\hat{v_1}+\hat{v_3}}{\lVert\hat{v_1}+\hat{v_3}\rVert} = [-\sqrt{\frac{1}{3}},\ \sqrt{\frac{1}{6}},\ -\sqrt{\frac{1}{2}}]^T$	-90
11	J_{2413}	$\frac{\hat{v_1}+\hat{v_4}}{\lVert\hat{v_1}+\hat{v_4}\rVert} = [-\sqrt{\frac{1}{3}},\ \sqrt{\frac{1}{6}},\ \sqrt{\frac{1}{2}}]^T$	90
12	J_{3142}	$\frac{\hat{v_1}+\hat{v_4}}{\lVert\hat{v_1}+\hat{v_4}\rVert} = [-\sqrt{\frac{1}{3}},\ \sqrt{\frac{1}{6}},\ \sqrt{\frac{1}{2}}]^T$	-90

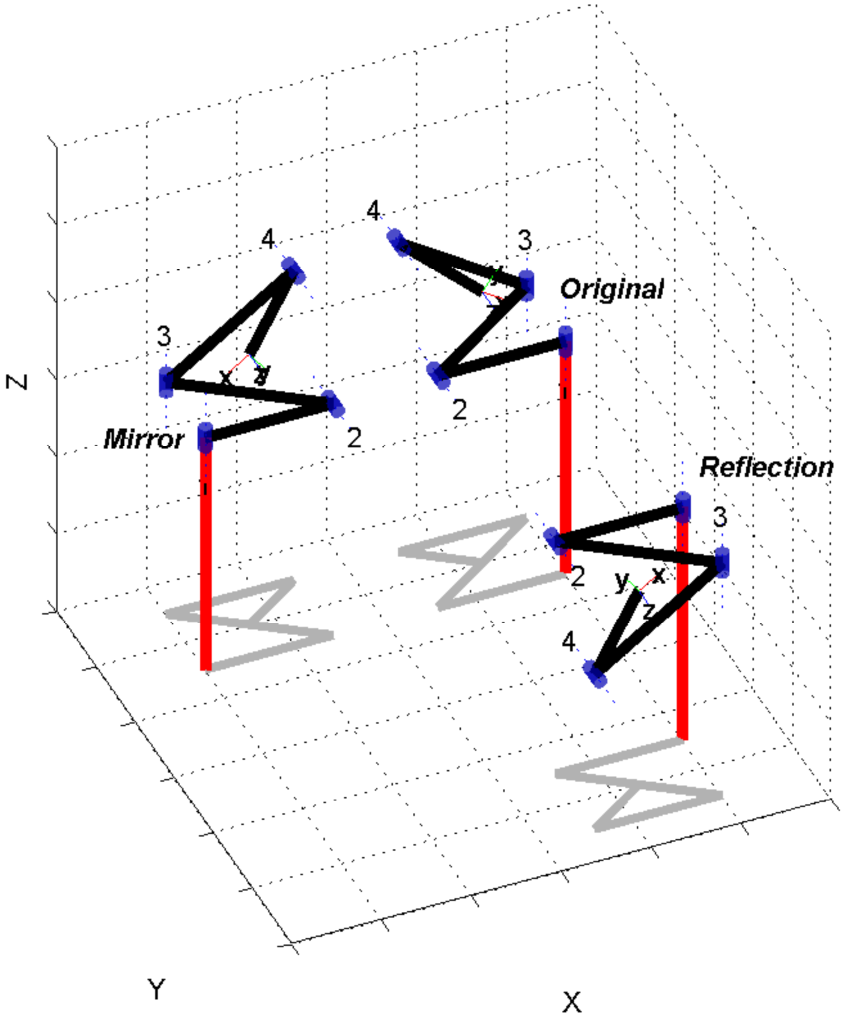


Figure 24: An illustration of kinematically equivalent robots generated from different Jacobians. The robot labeled "original" is generated from J_{1234} and represents all the robots specified in Table 9. The robot labeled "reflection" represents all the robots specified in Table 10. The robot labeled "mirror" represents robots generated from using the alternate solutions for β_1 when its range is not restricted.

CHAPTER VII

SEVEN DOF SPATIAL MANIPULATORS[1]

7.1 CHAPTER OVERVIEW

This chapter discusses the design of optimally fault-tolerant seven-joint fully spatial manipulators. The following section discusses the design of their optimally fault tolerant Jacobians which will be used to determine the the DH parameters. Section 7.3 will discuss how one can evaluate one robot design in terms of its global kinematic. In particular, how one can evaluate a six-dimensional volume that is a compound of orientation volume and the three-dimensional volume. The next section will analyze different examples, and discuss the effect of permuting the columns of designed Jacobians.

7.2 OPTIMALLY FAULT TOLERANT JACOBIAN

It was presented in [57] that for the case of a seven DOF fully spatial manipulator, the canonical structure of an optimally failure tolerant configuration is given by:

$$J = \begin{bmatrix} -\sqrt{\frac{6}{7}} & \sqrt{\frac{1}{42}} & \sqrt{\frac{1}{42}} & \sqrt{\frac{1}{42}} & \sqrt{\frac{1}{42}} & \sqrt{\frac{1}{42}} & \sqrt{\frac{1}{42}} \\ 0 & -\sqrt{\frac{5}{6}} & \sqrt{\frac{1}{30}} & \sqrt{\frac{1}{30}} & \sqrt{\frac{1}{30}} & \sqrt{\frac{1}{30}} & \sqrt{\frac{1}{30}} \\ 0 & 0 & -\sqrt{\frac{4}{5}} & \sqrt{\frac{1}{20}} & \sqrt{\frac{1}{20}} & \sqrt{\frac{1}{20}} & \sqrt{\frac{1}{20}} \\ 0 & 0 & 0 & -\sqrt{\frac{3}{4}} & \sqrt{\frac{1}{12}} & \sqrt{\frac{1}{12}} & \sqrt{\frac{1}{12}} \\ 0 & 0 & 0 & 0 & -\sqrt{\frac{2}{3}} & \sqrt{\frac{1}{6}} & \sqrt{\frac{1}{6}} \\ 0 & 0 & 0 & 0 & 0 & -\sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \end{bmatrix}. \tag{97}$$

Although the canonical form in (97) has the desirable property of fault tolerance, it corresponds to a manipulator possessing three prismatic joints and four joints that are capable of an arbitrary

[1] SOME OF THIS CHAPTER IS PUBLISHED IN [6]

screw motion [57]. The columns of a manipulator Jacobian for a robot consisting of only revolute joints have a more restrictive algebraic structure. In particular, the vector ω_i corresponding to the last three components of a column j_i must have unit length and must be perpendicular to the vector v_i consisting of the first three components of that column. Furthermore, the isotropy condition and the condition that the columns of J have equal norms require that the v_is also have unit norm. Including this additional constraint on the v_is gives a total of 21 constraints corresponding to $\|v_i\| = 1$, $\|\omega_i\| = 1$, and $v_i \cdot \omega_i = 0$ for $i = 1, 2, \ldots, 7$. If an isotropic configuration exists, then it follows that $\sigma = \sqrt{7/3} = 1.5275$. As noted in [57], we were not able to find an isotropic revolute manipulator Jacobian for which (6) achieves its upper bound. Instead, we identify manipulator Jacobians that were close to being ideally fault tolerant.

First, we determined a manipulator Jacobian satisfying the 21 constraints on the columns that was closest to satisfying the isotropy condition in the sense described in [57]. In this case, the objective function was the sum of the squares of the 21 unique constraints given by $JJ^T - n/3I = 0$. This resulted in an optimal manipulator Jacobian

$$J = \begin{bmatrix} 1 & 0.4296 & 0.7495 & -0.5431 & 0.1401 & 0.3298 & -0.3783 \\ 0 & -0.6041 & 0.6479 & 0.4640 & -0.7889 & -0.1853 & -0.8047 \\ 0 & -0.6712 & -0.1357 & -0.6998 & 0.5983 & -0.9257 & -0.4575 \\ 0 & 0.7678 & 0.1449 & 0.8391 & 0.5831 & -0.6882 & -0.4296 \\ 1 & -0.1469 & -0.3607 & 0.3308 & -0.4226 & -0.7184 & 0.5904 \\ 0 & 0.6236 & -0.9214 & -0.4319 & -0.6938 & -0.1014 & -0.6832 \end{bmatrix}. \tag{98}$$

This Jacobian has a $\mathcal{K} = 0.5196$, whereas the optimal value of $\mathcal{K}$ from (6) is 0.5774. Furthermore, while it does minimize the objective function, the value of the objective function was not zero and hence (98) is not isotropic. However, its singular values only range between $\sigma_1 = 1.5829$ and $\sigma_m = \sigma_6 = 1.4726$. This compares reasonably well to the Jacobian in (97) which is isotropic with

$\sigma = 1.5275$.

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

$$J = \begin{bmatrix} -0.2937 & 0.7136 & 0.6495 & -0.9856 & -0.0356 & 0.2941 & -0.3424 \\ -0.4233 & 0.5893 & 0.2041 & 0.1238 & -0.5771 & -0.8564 & 0.9396 \\ 0.8571 & 0.3788 & -0.7325 & -0.1150 & -0.8159 & 0.4243 & 0.0033 \\ 0.4676 & 0.6923 & -0.3226 & 0.1359 & 0.5575 & -0.8910 & -0.6397 \\ 0.7184 & -0.5105 & 0.9463 & 0.1769 & -0.6891 & -0.4063 & -0.2356 \\ 0.5151 & -0.5101 & -0.0224 & -0.9748 & 0.4630 & -0.2025 & 0.7316 \end{bmatrix}, \qquad (99)$$

where ${}^f\sigma_6 = 0.5714$ for all $f = 1, 2, \ldots, 7$ and $n_J = \frac{1}{\sqrt{7}}\,[1\ 1\ 1\ 1\ 1\ 1\ 1]^T$. As a trade off $\sigma_1 = 1.6455$ and $\sigma_m = \sigma_6 = 1.4169$ deviate more than (98) from the isotopic value $\sigma = 1.5275$. Thus, numerical experiments suggest that applying these types operations to a representative solution may identify a finite set of Jacobians that is sufficiently close to a large set of solutions. It should be noted that multiplying columns by -1 only changes the direction of the joint axis; however, permuting the columns results in a significant change in the robots structure. Thus applying $n!$ permutations may result in a family of Jacobians that sufficiently represent a large set of solutions.

In both approaches all calculated solutions converged to their respective optimal values. As pointed out earlier in Chapters 2 and 6, the operation of premultiplying a Jacobian by an orthogonal matrix corresponding to a coordinate frame rotation/reflection and postmultiplying by a signed permutation matrix changes the physical properties of the corresponding robot but does not affect its fault tolerance properties. When comparing two solutions for a given approach, it was found that applying a suitable operation of this type would result in a Jacobian that was close to the other Jacobian.

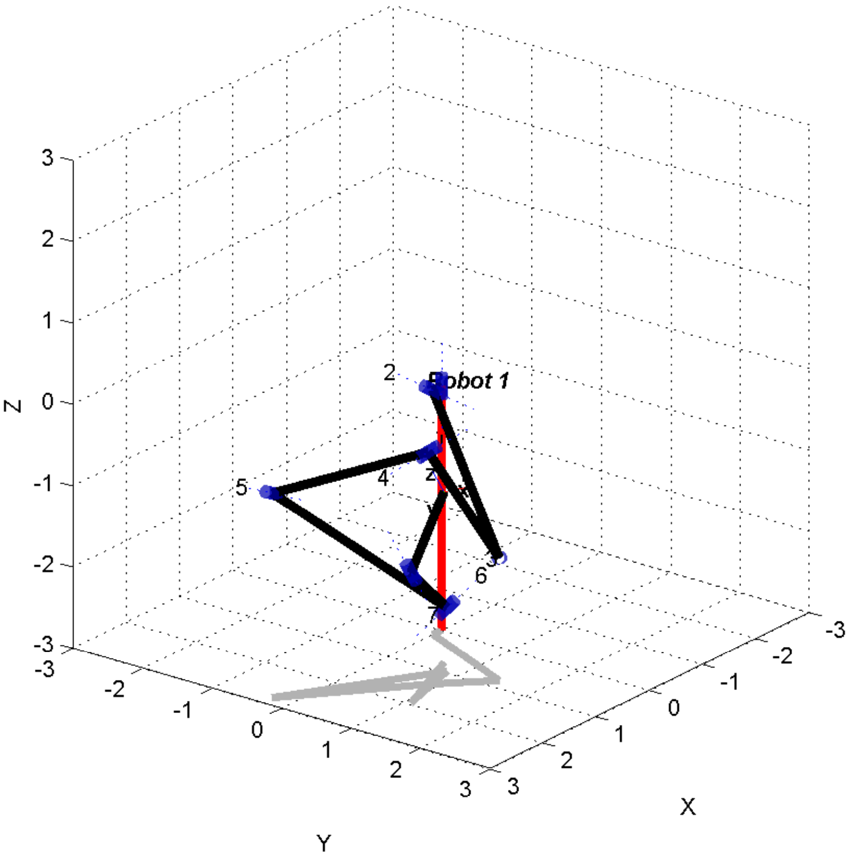


Figure 25: The configuration of a robot that was generated to have the locally optimal fault tolerant Jacobian in (98), referred to as the robot from (98).

Using the technique described in Chapter 3, one can generate the Denavit and Hartenberg parameters for a robot with a prescribed Jacobian. For example, Table 11 illustrates this for the Jacobian in (98) with the robot described by these parameters depicted in Figure 25, which we will refer to as robot from (98).

Table 11: The DH parameters of robot from (98)

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

7.3 COMPUTING GLOBAL FAULT TOLERANCE PROPERTIES

7.3.1 Preliminaries

The above section shows that there are multiple Jacobians, and therefore multiple manipulator designs, that share the same local fault tolerance properties. To distinguish between them, one would select a specific Jacobian and then calculate the corresponding physical robot in order to evaluate its global properties, especially how the fault tolerance measure varies across the workspace. Even though we are designing a fully general spatial manipulator with a six-dimensional workspace consisting of both position and orientation, it is also useful to consider the three-dimensional maximum reachable workspace volume. Specifically, in this work, both the three-dimensional reachable volume and six-dimensional volume of the workspace that has a $\mathcal{K}$ greater than or equal to a given fraction of the maximum, i.e., $\mathcal{K} \geq \gamma\, \mathcal{K}_{\max}$, where $\mathcal{K}_{\max}$ is equal to $\mathcal{K}$ of (98) or (99), and $0 \leq \gamma \leq 1$ is a user defined parameter, are computed. (For all of the results shown in the following examples $\gamma \approx 0.4$ is used.) The most difficult portion of this calculation is computing the six-dimensional volume, which is discussed in the following subsection.

7.3.2 Calculating a Six-Dimensional Volume

The six-dimensional workspace volume, denoted V_{6d}, of a fully spatial robotic manipulator can be decomposed into the product of the reachable workspace volume, denoted V_r and measured in m^3, and orientation volume, denoted V_o and measured in rad^3, within the reachable workspace, in the following way. The six-dimensional workspace volume corresponding to a small volume element of the reachable workspace centered at the three-dimensional cartesian position x is approximately equal to

$$\Delta V_{6d} \approx V_o(x)\Delta V_r(x) \tag{100}$$

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

$$V_{6d} \approx \sum_{i=1}^{N_r} V_o(P_i)\Delta V_r(P_i) = \frac{V_r}{N_r}\sum_{i=1}^{N_r} V_{o_i} \tag{101}$$

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

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

7.3.3 Calculating Orientation Volume

7.3.3.1 Parameterized sampling of quaternions

One simple way to sample orientations is to use spherical polar coordinates to parameterize unit quaternions [63], i.e.,

$$\begin{aligned} s &= \cos(\psi) \\ v_x &= \sin(\psi)\cos(\phi) \\ v_y &= \sin(\psi)\sin(\phi)\cos(\theta) \\ v_z &= \sin(\psi)\sin(\phi)\sin(\theta) \end{aligned} \tag{102}$$

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

$$\sin^2(\psi)\sin(\phi)d\psi d\phi d\theta. \tag{103}$$

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

$$V_{o_i} \approx \pi^3 \frac{1}{N_o} \sum_{j=1}^{N_{o_i}} \sin^2(\psi_j) sin(\phi_j) \tag{104}$$

where ψ_j and ϕ_j are the spherical polar coordinates of achievable orientation j. Note that the maximum orientation volume, denoted $v_{o_{max}}$, is π^2, which can be obtained from integrating (103) over the total range of ψ, ϕ, and θ.

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

7.3.3.2 Uniformly sampling unit quaternions

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

$$V_{o_i} \approx V_{o_{max}} \frac{N_{o_i}}{N_o} = \pi^2 \frac{N_{o_i}}{N_o}. \tag{105}$$

In order to generate N_o uniformly distributed quaternions on the surface of a 3-sphere [64], for each generated $q = [s,\ v_x,\ v_y,\ v_z]$, we select s and v_x as independent random variables uniformly distributed between $[0,\ 1]$ and $[-1,\ 1]$ respectively, under the constraint that $S_1 = s^2 + v_x^2 < 1$. We

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

then compute two different independent uniform variables, v'_y and v'_z, between $[-1,\ 1]$, under the constraint that $S_2 = {v'_y}^2 + {v'_z}^2 < 1$. Then,

$$\begin{aligned} q &= [s,\ v_x,\ v_y,\ v_z] \\ &= \left[s,\ v_x,\ \left(\sqrt{\frac{1-S_1}{S_2}}\right) v'_y,\ \left(\sqrt{\frac{1-S_1}{S_2}}\right) v'_z\right]. \end{aligned} \tag{106}$$

We compared the uniform sampling approach to the parameterized sampling approach using a hundred randomly generated positions in robot from (98)'s workspace and determined that the uniform sampling approached had both slightly higher accuracy and slightly lower computation time, so this approach is used in all orientation volume calculations in this work. Clearly, the accuracy of any Monte Carlo technique is a function of the number of samples, N_o. Based on experimentation, we use $N_o = 1000$ as a compromise between accuracy and computation time.

7.3.4 Workspace Volume Estimation Algorithms

7.3.4.1 Overview

To determine an estimate for the workspace volumes, we develop two algorithms based on Monte Carlo integration. The first, denoted Algorithm A, which is more straightforward, uses direct sampling within the six-dimensional workspace. It is appropriate is one only needs information about six-dimensional volumes. The second, denoted Algorithm B, uses a decomposed sampling technique where a number of samples are associated with each position in order to obtain orientation volume information as a function of position. In both cases, we use a total of 10^7 samples, which we have experimentally determined to be sufficient for the Monte Carlo integration to converge as shown in Figure 26. In Algorithm B, 10^4 samples are used within the three-dimensional position workspace with 10^3 samples used in the orientation space associated with each position sample

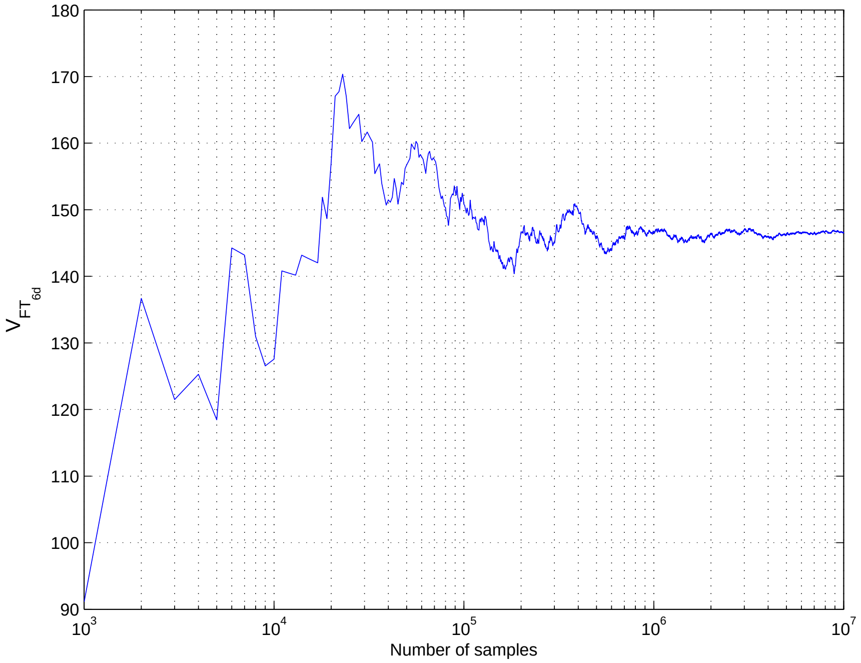


Figure 26: Six-dimensional fault tolerant volume for $\mathcal{K} > 0.2$ for the robot from (98) as a function of sampling rate

point.[3]

7.3.4.2 Algorithm A - Direct sampling in six-dimensional space

The first algorithm that we developed uses direct sampling in the six-dimensional workspace to implement Monte Carlo integration. To make our sampling efficient, we first compute a maximum radius for the reachable workspace. To do this, we generate one million uniformly distributed random configurations in the joint space, where $0 \leq \theta_i < 2\pi$ for all i, that are transformed to the workspace using forward kinematics. Then the maximum reach R_{max} of the manipulator is estimated by picking the point with the largest norm and using inverse kinematics on only the linear velocity portion of the Jacobian to drive the robot to its workspace boundary where its Jacobian is singular. The process for doing this calculation is illustrated in Figure 27. Once R_{max} has been

[3] Both algorithms has been implemented in C++ using Armadillo linear algebra library [62], utilizing the CSU ISTeC Cray HPC System.

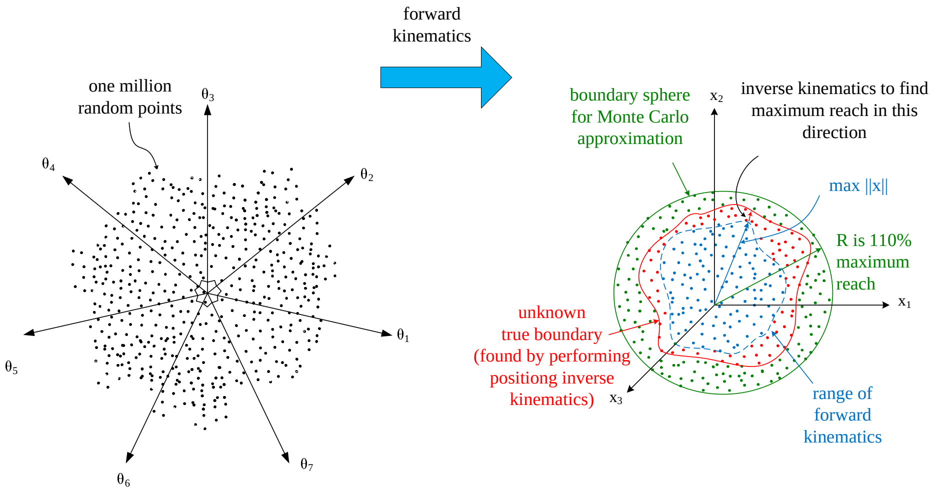

forward kinematics
one million random points
θ3
θ4
θ2
θ1
θ5
θ6
θ7
boundary sphere for Monte Carlo approximation
x2
inverse kinematics to find maximum reach in this direction
max ||x||
R is 110% maximum reach
x1
unknown true boundary (found by performing positiong inverse kinematics)
x3
range of forward kinematics

Figure 27: An illustration of how Monte Carlo integration is used to compute the volume of the reachable workspace. This is done by first determining the maximum reach of the robot by performing forward kinematics on one million random joint configurations and then driving each of these to a workspace boundary. The Monte Carlo integration is then performed within a sphere of R that is 110% of the maximum reach.

determined, we randomly select N samples directly in the six-dimensional workspace, which consists of positions and quaternions that represent orientations. The position part is uniformly sampled within a sphere whose radius R is 110% of R_{max} so that the maximum volume, $V_{r_{max}} = \frac{4}{3}\pi R^3$. The quaternion part is sampled using the uniform sampling approach described in subsection 7.3.3.2. Denote N_{3d} as the number of samples that are reachable for the given position irrespective of orientation. Then the reachable workspace volume V_r can be approximated by

$$\begin{aligned} V_r &\approx \left(\frac{N_{3d}}{N}\right) V_{r_{max}} \\ &\approx \left(\frac{N_{3d}}{N}\right) \frac{4}{3}\pi R^3. \end{aligned} \tag{107}$$

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

$$V_{6d} \approx \left(\frac{N_{6d}}{N}\right) V_{r_{max}} V_{o_{max}}$$

$$\approx \left(\frac{N_{6d}}{N}\right)\frac{4}{3}\pi R^3 \cdot \pi^2. \tag{108}$$

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

To calculate the fault tolerant six-dimensional volume, we need to determine the number of points where $\mathcal{K} \geq \gamma\ \mathcal{K}_{\max}$, which is denoted N_{FT}. Determining whether a point satisfies this condition requires that one check all of the robot configurations in all self-motion manifolds associated with this point. Techniques for doing this are described in Section 7.3.4.4. Once N_{FT} is computed, the six-dimensional volume is approximated by

$$V_{FT} \approx \left(\frac{N_{FT}}{N}\right)\frac{4}{3}\pi R^3 \cdot \pi^2. \tag{109}$$

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

$$V_{FT_{3d}} \approx \left(\frac{N_{FT_{3d}}}{N}\right)\frac{4}{3}\pi R^3 \tag{110}$$

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

7.3.4.3 Algorithm B - Decomposed sampling of six-dimensional space

An alternate approach to directly sampling the six-dimensional workspace volume is to decompose the workspace into two three-dimensional spaces and perform Monte Carlo integration on both. We sample N_p points in the three-dimensional position workspace and for each reachable position we sample N_o points in the orientation space to determine the associated orientation volume, using one of the approaches discussed in Subsection 7.3.3.

Similarly to Algorithm A, we determine R_{max} first in order to define the sampling sphere whose radius R is 110% of R_{max}. One can then apply (107) to calculate the three-dimensional position volume by replacing N with N_p. For each of these reachable points one then still needs to determine the orientation volume using one of the approaches that were discussed in Subsection 7.3.3. Then, one can directly use (101) to calculate the total reachable six-dimensional volume.[4]

Because Algorithm B uses decomposed sampling, $V_{FT_{3d}}$ is easily determined during the process for computing V_{FT}. Figure 28 illustrates the process for computing these two volumes. We first describe how to compute the fault tolerant orientation volume associated with a reachable position P_i, denoted $V_{FT_{oi}}$. For each P_i shown in (a), there is an associated N_{oi} reachable orientations as illustrated in (b). For each reachable position and orientation there are multiple robot configurations which typically occur in multiple self-motion manifolds as illustrated in (c). If any of these configurations have a $\mathcal{K} \geq \gamma\, \mathcal{K}_{\max}$ then this orientation should be included in the sum of all such orientations, denoted $N_{FT_{oi}}$. One can now compute $V_{FT_{oi}}$ using

$$V_{FT_{oi}} \approx \pi^2 \frac{N_{FT_{oi}}}{N_o}, \tag{111}$$

which is analogous to (105). Therefore, one can use (110) to determine $V_{FT_{3d}}$ by replacing N with

[4] If one is not interested in the orientation volume associated with each position, then one can directly compute (108) using $N = N_p N_o$ and $N_{6d} = \sum_{i=1}^{N_{3d}} N_{oi}$.

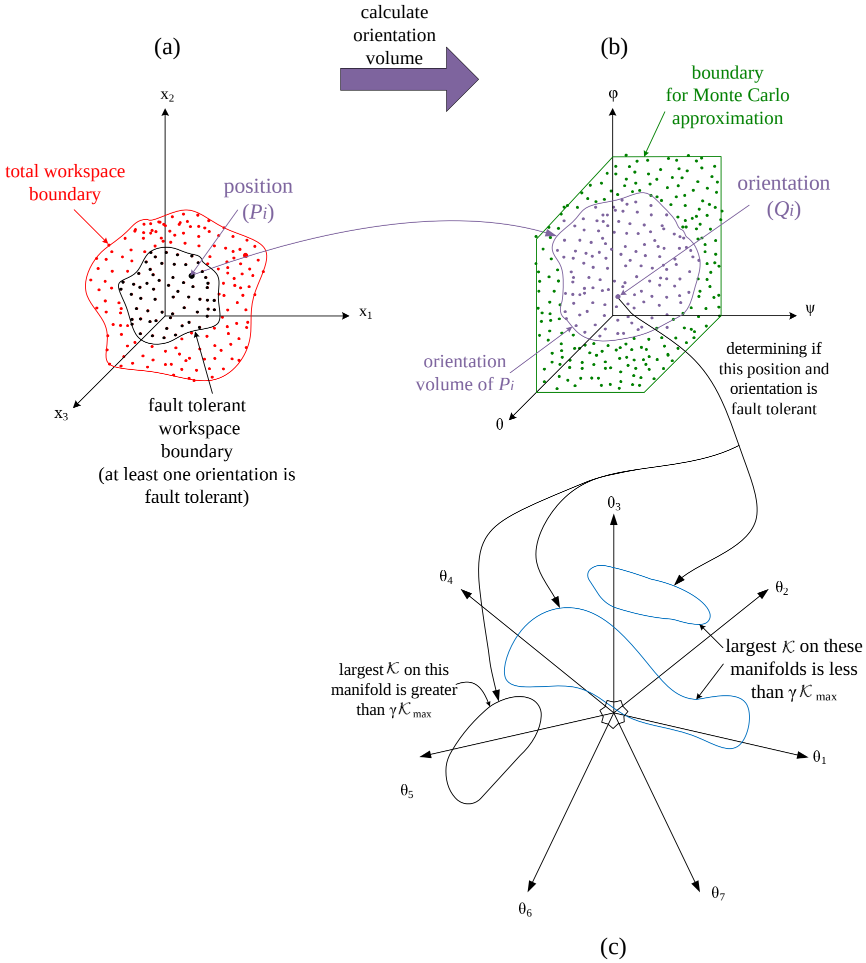


Figure 28: An illustration of how Monte Carlo integration is used to compute the six-dimensional volumes of both the fault-tolerant workspace and the reachable workspace. For each reachable workspace position P_i in (a) we use Monte Carlo integration to evaluate the achievable orientation volume at that P_i as shown in (b). To evaluate the fault tolerance of a P_i and Q_i in (b) one needs to identify the maximum value of $\mathcal{K}$ for all self-motion manifolds associated with that P_i and Q_i, as shown in (c). The three-dimensional fault-tolerant volume contains all of the positions that have at least one fault-tolerant orientation.

N_p, where $N_{FT_{3d}}$ is now the sum of all of reachable positions whose $V_{FT_{oi}} > 0$. Similar to (101) V_{FT} can be approximated by

$$V_{FT} \approx \frac{V_{FT_{3d}}}{N_{FT_{3d}}} \sum_{i=1}^{N_{FT_{3d}}} V_{FT_{o_i}}. \qquad (112)$$

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

7.3.4.4 Maximizing $\mathcal{K}$

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

The first approach is to evaluate every configuration for every self-motion manifold to determine the maximum value of $\mathcal{K}$ at a workspace point (see Figure 28 (c)). This is not easy, because even determining how many self-motion manifolds exist is not trivial. Our approach to identifying all manifolds is to use multiple random configurations whose locations are close to the point that we are evaluating. Recall that we have already computed the forward kinematics mapping of one million samples in the joint space that were used to estimate the maximum reach R_{max}. It is likely that these samples will include all self-motion manifolds when the sampling rate is high enough. However, it is still possible to miss a manifold, especially if it is small, and increasing the number of samples is computationally expensive. Figure 29 shows an example for a typical point where five different self-motion manifolds were identified. They are graphed by stepping along the manifold and plotting the absolute value of the difference from an arbitrarily assigned start configuration. The fact that each plot returns to zero indicates that all five of the self-motion manifolds are closed curves, which is not necessarily true.

This approach is relatively straightforward for an isolated point in the workspace, however, if one is concerned with continuous trajectories of the end-effector in the workspace, the situation becomes more complicated. This is because adjacent points in the workspace may have maximum $\mathcal{K}$ values that are associated with configurations that are not adjacent in the joint space. This means that it is not possible to track the maximum $\mathcal{K}$ trajectory without large discontinuities in joint configuration. These jumps in configuration can be either between self-motion manifolds or within a single manifold.

One way of dealing with this issue is to use the gradient projection technique with $\nabla\mathcal{K}$ as described in [13]. It maximizes $\mathcal{K}$ locally, depending on the starting configuration, as opposed to optimizing $\mathcal{K}$ globally. In this work we always chose the starting configuration to be the optimally fault tolerant design configuration. This approach is much faster than searching for the global optimal across all self-motion manifolds and it bounds the joint velocity, which makes it applicable for real-time implementation. If one is concerned with locally minimal joint velocity then one can use the pseudoinverse solution, once again, starting from the design configuration.

To illustrate the differences between the above techniques for maximizing $\mathcal{K}$ we selected an example trajectory that moves the end effector along a straight line in the y direction through the design point while maintaining constant orientation. Figure 30 presents the maximum value of $\mathcal{K}$ and norm of the joint displacement, $\|\Delta\theta\|$, along this trajectory for the three techniques. Within ±1 m of the design point the $\nabla\mathcal{K}$ tracking approach is essentially the same as searching all self-motion manifolds because the locally optimal point is globally optimal. Beyond ±2 m the globally optimal value of $\mathcal{K}$ can be maintained at a relatively large value, i.e., $\mathcal{K} \approx 0.2$ between -4 m to 7 m, however, it requires many transitions between disjoint self-motion manifolds that result in large joint displacements. These large displacements can be alleviated by using the $\nabla\mathcal{K}$ approach (especially if one does not want to exactly track the local optimal), however, the locally optimal value of $\mathcal{K}$ is relatively small outside of ±2 m. In other words, to obtain larger values of $\mathcal{K}$, one must switch self-motion manifolds. This will require deviation from a purely y velocity trajectory. This is illustrated in Figure 31 that shows the joint displacement required to move between two adjacent locations, e.g., from $y = 1.50$ m and $y = 1.55$ m, with $\mathcal{K}$ at a global maximum. The total joint displacement for this motion is 3.7 rad and results in an end effector motion of 2.7 m from the desired linear trajectory.

Figure 30 illustrated the variation in $\mathcal{K}$ over a specific trajectory. To evaluate how the three

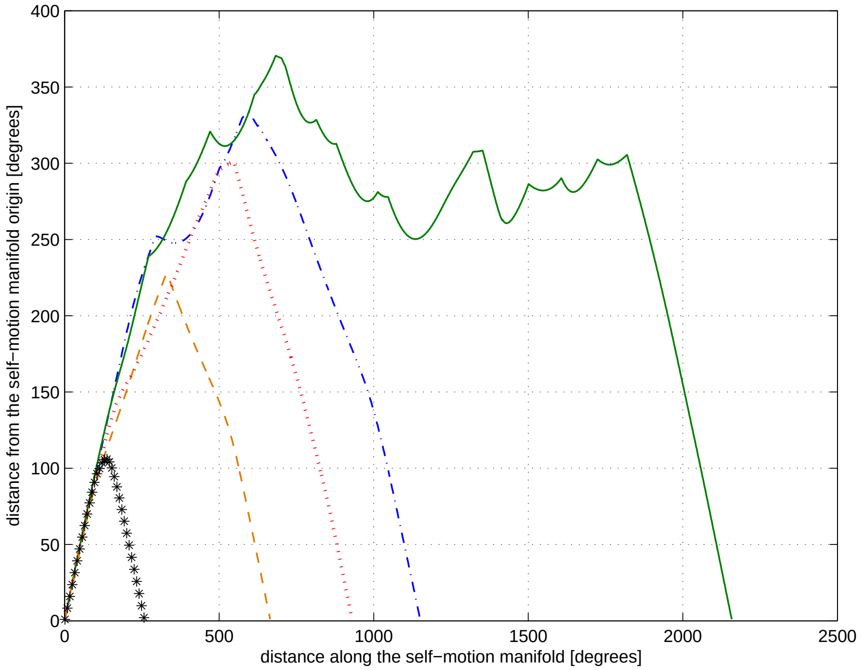


Figure 29: An example of five self-motion manifolds for the robot from (98) where its end effector is located four meters from the design point in the positive y direction. The independent axis is a measure of the size of the manifold (in degrees) with the dependent axis being the distance from an arbitrary origin on the manifold (giving some sense of its shape).

Table 12: Comparison of $\mathcal{K}$ maximization techniques using the robot from (98)

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

techniques behave over the entire workspace we computed both the three-dimensional and six-dimensional fault tolerant volumes where $\mathcal{K} \geq \gamma\ \mathcal{K}_{\max} = 0.2$. The results are shown in Table 12. In the remainder of this work we maximize $\mathcal{K}$ by searching all of the self-motion manifolds.

7.4 EXPLORATION OF DIFFERENT ROBOT DESIGNS

7.4.1 Comparing Robot Designs

Now that we have a way to measure the global workspace volumes of interest, we have a way of comparing various different robot designs that are all locally optimal. In Section 7.2, we discussed two different definitions of "optimal" Jacobians, i.e., those given in (98) and (99), from which the

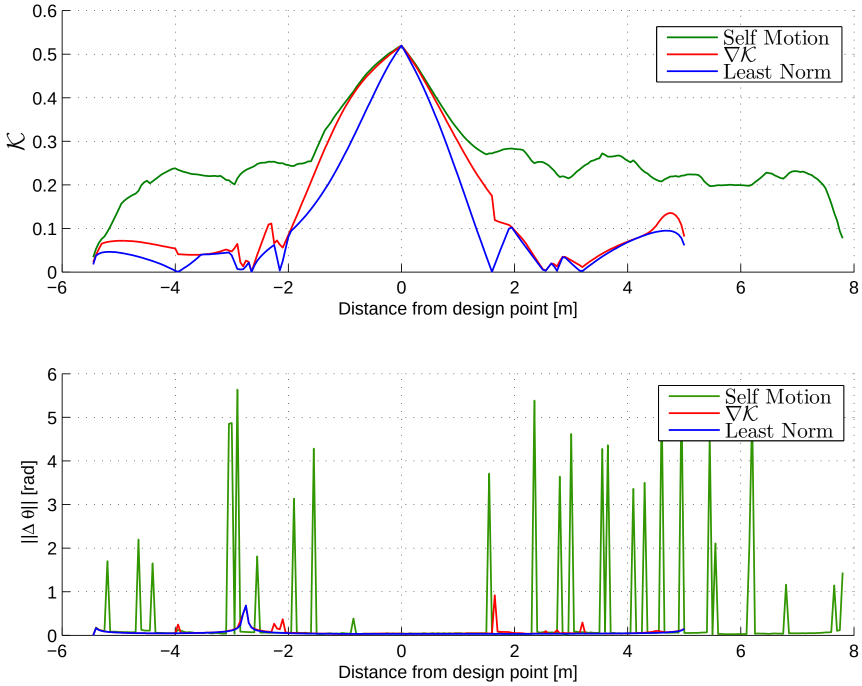


Figure 30: A plot of the maximum value of $\mathcal{K}$ and the rate of joint displacement for robot from (98), for a trajectory along the y-axis away from the design point while keeping the orientation constant. The ability to maintain a large value of $\mathcal{K}$ far from the design point comes at the expense of very large joint motion. In fact, the magnitude of joint change $\|\Delta\theta\|$ curve for the technique that tracks the maximum over all self-motion manifolds is scaled down by a factor of ten.

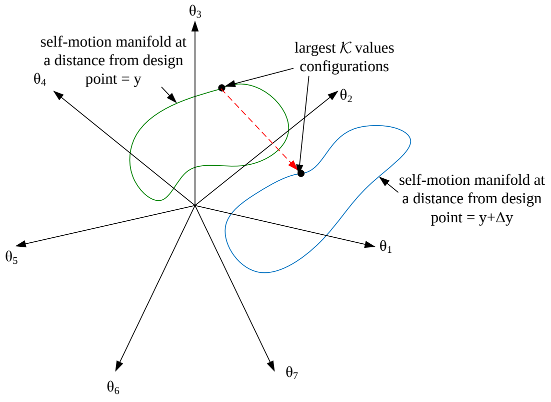


Figure 31: An illustration of the large value of joint motion that can occur when tracking the globally optimal fault-tolerant configuration. This motion is due to a switch between self-motion manifolds. This reconfiguration will also require motion of the end effector.

kinematic parameters of a manipulator can be determined. Regardless of which definition, i.e., Jacobian, we use, there are $7! = 5040$ different permutations that result in an equal number of unique robot designs, which obviously still possess the same locally optimal fault tolerant property.

To illustrate one aspect of the wide variation among all of the robot designs, Figure 32 is a plot of the maximum reach as a function of the permutation index, which is ordered from minimum to maximum value of R_{max}, for both Jacobians given in (98) and (99). It is not surprising that the variation of R_{max} is similar for both Jacobians, because both optimizations require that every column is of equal norm.

There appears to be a rough relationship between the maximum robot reach, R_{max}, and the normalized workspace volumes. We will illustrate this with robot designs that have minimum, maximum, and mid-range values of R_{max}. It turns out that the robots designed directly from (98) and (99) are in the mid-range, with $R_{max} = 9.4$ and $R_{max} = 8.7$, respectively. The minimum and maximum R_{max} robots for (98) result from permutations $[j_1 \;\; j_2 \;\; j_5 \;\; j_7 \;\; j_4 \;\; j_3 \;\; j_6]$ and $[j_2 \;\; j_7 \;\; j_3 \;\; j_5 \;\; j_4 \;\; j_1 \;\; j_6]$, respectively. The DH parameters for robots generated from these Jacobians are given in Table 13 and Table 14, respectively. The minimum and maximum R_{max} robots for (99) result from permutations $[j_5 \;\; j_1 \;\; j_4 \;\; j_6 \;\; j_2 \;\; j_3 \;\; j_7]$ and $[j_3 \;\; j_5 \;\; j_2 \;\; j_4 \;\; j_7 \;\; j_1 \;\; j_6]$, respectively.

Table 15 presents the three- and six-dimensional volumes using percentages out of the maximum to give some intuition about the relative size of these workspaces for these six different robots. Arguably, the best robot design is given by the minimum R_{max} of (98), where the maximum reachable three-dimensional volume is 99% of a sphere of radius R_{max}, indicating that this robot's reachable workspace is almost spherical. The six-dimensional workspace volume is 49% of the maximum six-dimensional volume, i.e., $\frac{4}{3}\pi R_{max}^3 \cdot \pi^2$. In other words, within the spherical workspace, this robot is capable of achieving approximately half of all possible orientations. For relatively high degrees of fault tolerance, i.e., $\mathcal{K} > 0.2$, the robot would be limited to only 27% of the maximum workspace. However, if one is only concerned with a spatial positioning task, the three-dimensional volume with $\mathcal{K} > 0.2$ is quite large, i.e., 81%.

This "best" robot design is shown in its optimal configuration in Figure 33. From Figure 33, one

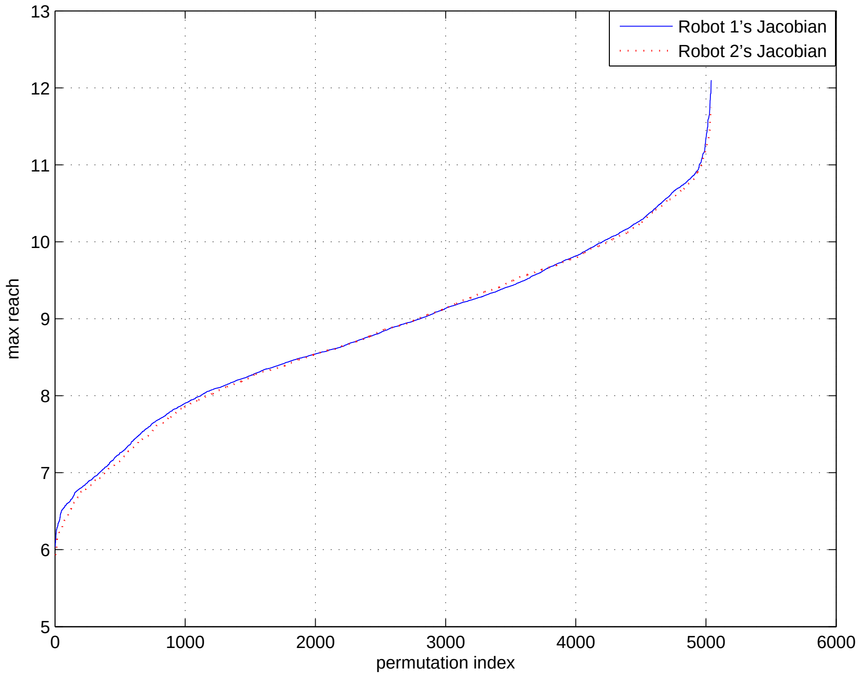


Figure 32: A plot of the maximum reach for all possible robots generated from the 7! column permutations of (98) and (99).

can see that the robot is not folded up on itself like robot from (98) in Figure 25. This is typical of min robot designs because the joints of a robot generated from an optimally fault tolerant Jacobian are constrained to lie on a sphere that is centered at the end effector. Consequently, the robot designs will become increasingly folded up on themselves as their R_{max} increases. Note that the min robot for the Jacobian from (99) is also the best, in the sense of having the largest volumes.

To visualize the six-dimensional volume, we use a three-dimensional volume plot, where for each

Table 13: The DH parameters of the min robot from (98)

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

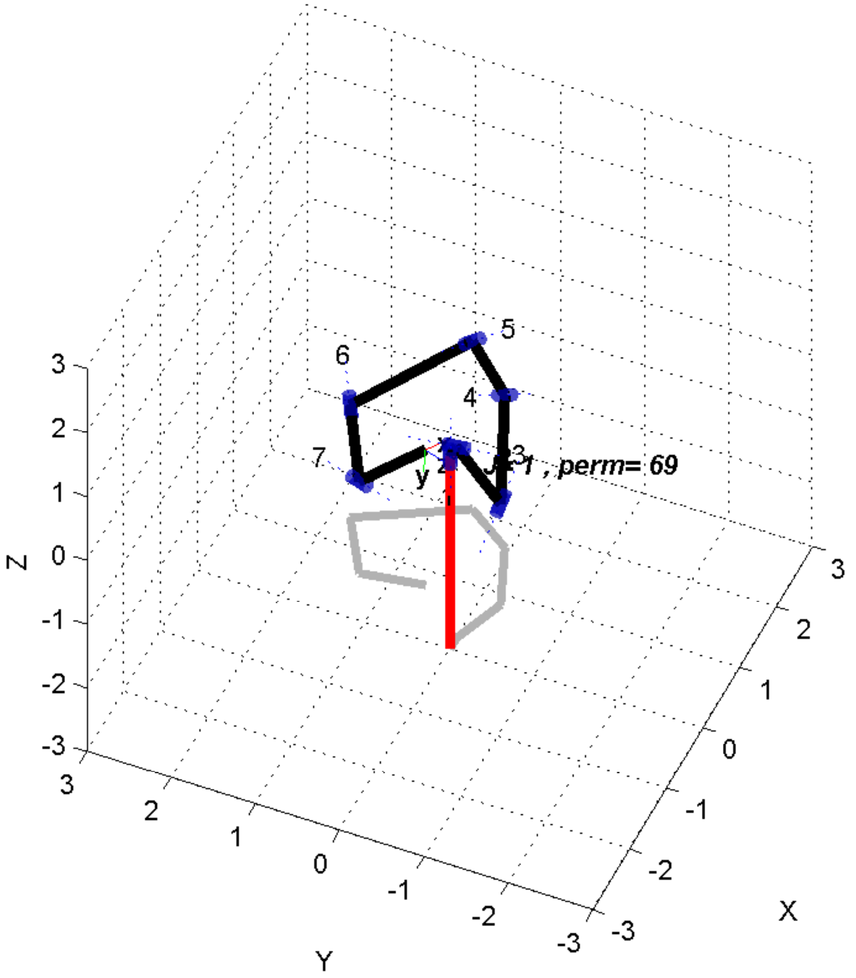


Figure 33: The locally optimal fault tolerant configuration at the design point of the min robot from (98). Note the more evenly distributed links of this robot configuration as compared to that of Figure 25

point we use color to represent the orientation volume at that point. One can visualize the three-dimensional workspace by simply ignoring the color map. Figure 34 illustrates the fault tolerant

Table 14: The DH parameters of the max robot from (98)

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

Table 15: Comparison of robot workspace volumes[5]

		Robots from J of (98)			Robots from J of (99)		
		Max reach R_{max} [m]			Max reach R_{max} [m]		
		min	mid	max	min	mid	max
		6.0	9.4	12.1	5.9	8.7	11.7
Volumes[%]	V_r	99	96	62	97	95	68
	$V_{FT_{3d}}$	81	81	51	81	73	59
	V_{6d}	49	46	26	49	41	28
	V_{FT}	27	24	12	26	19	13

six-dimensional volume for the min robot from (98), where the three-dimensional volume part is shown with multiple cross-sections at the design point to better visualize the interior. The color map represents the orientation volume distribution within the three-dimensional volume, where the orientation volume is represented by its percentage of the maximum value $V_{o_{max}} = \pi^2$. One can see how the largest values of fault tolerant orientation volume are concentrated around the design point. Figure 35 illustrates the total reachable volume, which is similar to the volume where $\mathcal{K} > 0$, because there are large self motion manifolds that make it easy to find a configuration where $\mathcal{K} \neq 0$.

7.4.2 Modifying the Anthropomorphic Arm Design

It is interesting to note that robots with a kinematic design that is similar to a seven degree-of-freedom human arm, e.g., the Mitsubishi PA–10 (the DH parameters are presented in Table 16) are very fault intolerant. This is because the fourth joint, i.e., the elbow, is the only joint that can change the distance from the spherical shoulder to the spherical wrist, so that is critical. Thus if the robot's tool is close to the wrist, the wrist joints cannot significantly compensate for a loss in linear velocity due to motion of the elbow. One way to mitigate this issue, and improve the fault tolerance to a failure in the elbow joint, is to use a bigger tool offset. However, this will still not achieve the optimal fault tolerance of the designs discussed here.

Another alternative is to physically modify the design of an anthropomorphic arm structure by moving the third joint to be together with the fourth joint at the same coordinate frame. Table 17 presents the resulting DH parameters, where the parameters a_2, a_4, and d_7 can be designed to

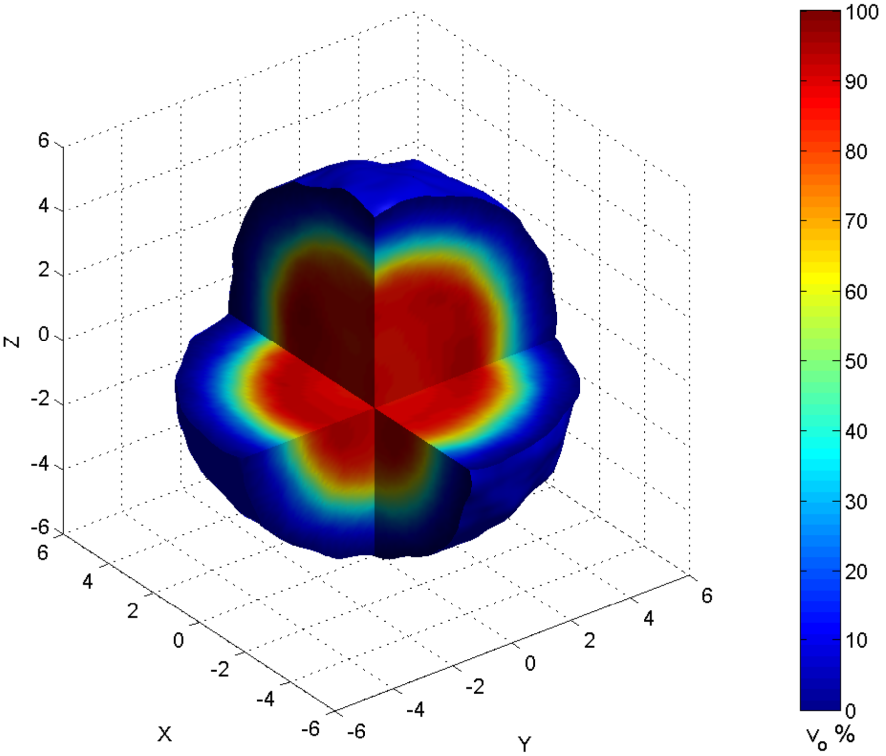


Figure 34: Six-dimensional fault tolerant volume of $\mathcal{K} > 0.2$ for the min robot from (98). The orientation volume is represented by its color within the three-dimensional volume as a percentage of $V_{o_{max}} = \pi^2$.

Table 16: The DH parameters of PA–10 robot

i	α_i [degrees]	a_i [m]	d_i [m]	θ_i [degrees]
1	-90	0	0	θ_1
2	90	0	0	θ_2
3	-90	0	0.450	θ_3
4	90	0	0	θ_4
5	-90	0	0.5	θ_5
6	90	0	0	θ_6
7	0	0	0.08	θ_7

further optimize fault tolerance. Note that moving the third joint in this way interchanged the roles of the d offsets and the a link lengths for some of the joints. However, the robots still share similar physical structure as indicated in Fig. 36.

In this work we chose the Jacobian in (98) for comparison with the modified anthropomorphic arm. In order to obtain a fair comparison with (98), we chose the values of a_2, a_4, and d_7 so that

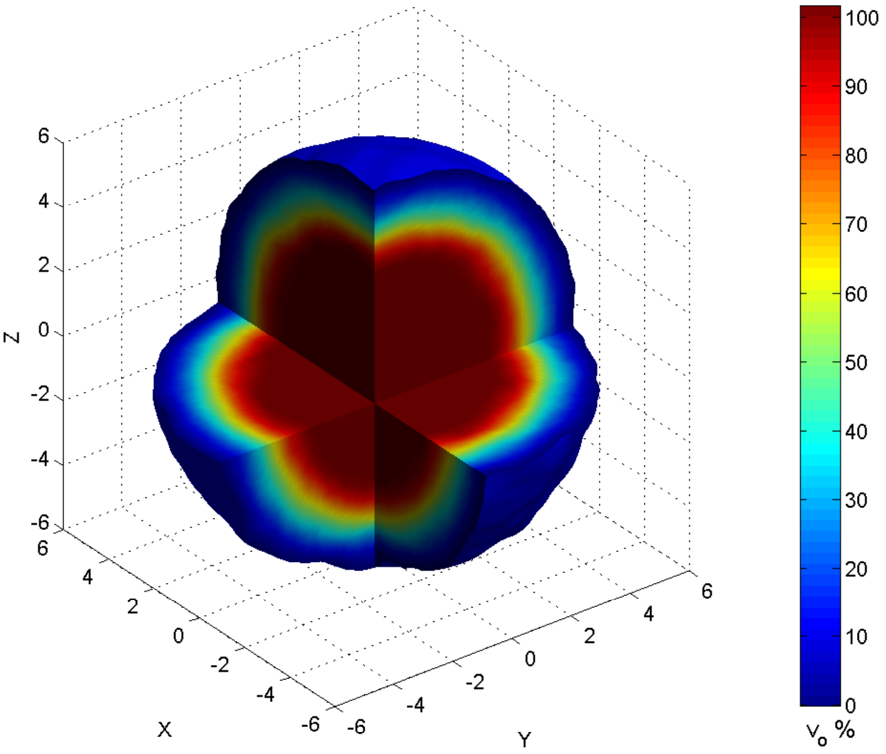


Figure 35: Six-dimensional reachable volume for the min robot from (98). This is very similar to the fault tolerant volume for $\mathcal{K} \neq 0$. The orientation volume is represented by its color within the three-dimensional volume as a percentage of $V_{o_{max}} = \pi^2$.

the average of the singular values

$$\sigma_{ave} = \frac{1}{6} \sum_{i=1}^{6} \sigma_i \tag{113}$$

is close to the average of the singular values of (98). To do this, we first randomly generated one million uniformly distributed points in the joint space. We chose $a_2 = a_4 = a$ as an approximation to the lengths of a human arm, and varied a to be 1, 2, and 3. We then varied d_7 to be in the range of 0 to 100% of a, in 10% increments. Note that when identifying the fault tolerance of a manipulator Jacobian, one should remember that scaling the length parameters essentially changes the weighting between the position and orientation. Otherwise, it might be tempting to think that two (a_2, a_4, d_7) combinations that are proportional to each other would result in the same fault tolerance properties. Thus each of the above 33 selections will generally result in different

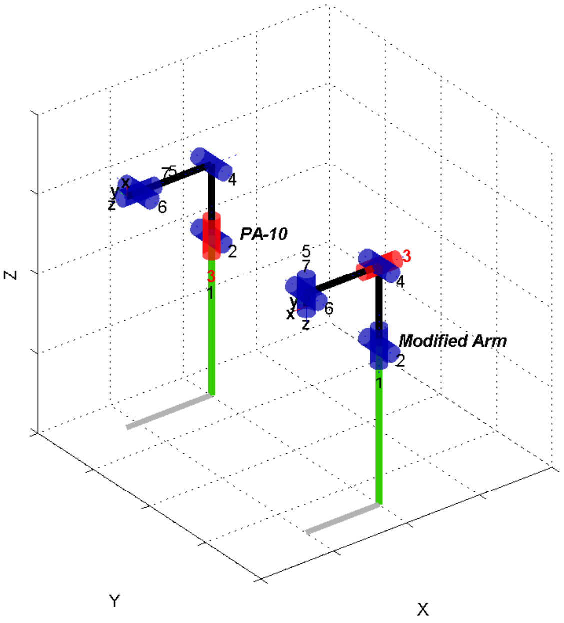


Figure 36: Modifying the PA-10 to be fault tolerant. Note that the third joint is moved from the base frame to the same location as the coordinate frame of the fourth joint. This is a result of considering the joint offsets d_3 and d_5 to be link lengths a_2 and a_4. To help with visualizing this difference, the joint values for the PA-10 are set to $[\ 0^\circ\ ,\ 0^\circ\ ,\ 0^\circ\ ,\ -90^\circ\ ,\ 0^\circ\ ,\ 0^\circ\ ,\ 0^\circ]$, whereas for the modified arm the joint values are $[\ 0^\circ\ ,\ -90^\circ\ ,\ 0^\circ\ ,\ -90^\circ\ ,\ 0^\circ\ ,\ 0^\circ\ ,\ 0^\circ]$. (This graphic was generated using the Robotics Toolbox described in [61].)

Table 17: Kinematics of a modified anthropomorphic arm

i	α_i [degrees]	a_i [m]	d_i [m]	θ_i [degrees]
1	-90	0	0	θ_1
2	90	a_2	0	θ_2
3	-90	0	0	θ_3
4	90	a_4	0	θ_4
5	-90	0	0	θ_5
6	90	0	0	θ_6
7	0	0	d_7	θ_7

fault tolerance properties. For each of the 33 resulting robots, we calculated one million Jacobians corresponding to the one million configurations generated earlier, and evaluated (113). The robot parameters that resulted in a configuration that most closely matched the average singular value of the nominal Jacobian in (98) were $a = 2$ and $d_7 = 1$.

A desirable configuration should have a large $\mathcal{K}$ value and should be close to being isotropic. In the next step we optimized the joint configuration and d_7 value in terms of both $\mathcal{K}$ and the isotropy measure

$$\mathcal{I} = \frac{\sigma_6}{\sigma_1}. \tag{114}$$

Note that $\mathcal{I}$ is the reciprocal of the condition number and is well defined for any nonzero matrix. Unlike the condition number, $\mathcal{I}$ is bounded and takes on the optimal value 1 when the configuration is isotropic and the value 0 at a singularity. In general, larger values of $\mathcal{I}$ correspond to better configurations in the sense of isotropy. In order to take into account both measures, we chose to maximize $\mathcal{K} + \mathcal{I}$. This was accomplished using a nullspace projection method. For each of the one million starting configurations, we used the projection of the gradient of $\mathcal{K} + \mathcal{I}$ onto the nullspace to determine the joint motion that would improve fault tolerance and isotropy:

$$\dot{\theta} = \left(I - J^{+}J\right)\left(\nabla\mathcal{K} + \nabla\mathcal{I}\right). \tag{115}$$

Applying (115) reconfigures the robot along its self-motion manifold to a maximum value for $\mathcal{K}+\mathcal{I}$ subject to the constraint of not moving the end effector.[6] The gradients for $\mathcal{K}$ and $\mathcal{I}$ are found using the method described in [13]. In [13] it was noted that $\mathcal{K}$ can be written as

$$\mathcal{K} = {}^{F}u_6^{T}\ {}^{F}J\ {}^{F}v_6 \tag{116}$$

where $F = \arg\min_{f=1,\ldots,7}\ {}^{f}\sigma_6$ is the index of the most debilitating joint failure and ${}^{F}u_6$ and ${}^{F}v_6$ are respectively the input and output singular vectors corresponding to the minimum singular value of ${}^{F}J$. The gradient of $\mathcal{K}$ is found by taking its partial derivatives, which results in three terms. It is not difficult to show that the first and third terms are zero, i.e.,

$$\left(\frac{\partial\ {}^{F}u_6^{T}}{\partial\theta_i}\right)\ {}^{F}J\ {}^{F}v_6 = {}^{F}u_6^{T}\ {}^{F}J\left(\frac{\partial\ {}^{F}v_6}{\partial\theta_i}\right) = 0 \tag{117}$$

because the partial derivatives of ${}^{F}u_6^{T}\ {}^{F}u_6 = {}^{F}v_6^{T}\ {}^{F}v_6 = 1$ are zero. Thus, we have that the partial

[6] In practice, a position error term is required to make sure that the end effector is not moved.

derivative of $\mathcal{K}$ is given by

$$\frac{\partial \mathcal{K}}{\partial \theta_i} = {}^F u_6^T \left(\frac{\partial\ {}^F J}{\partial \theta_i} \right) {}^F v_6. \tag{118}$$

To obtain the partial derivative of ${}^F J$ with respect to θ_i, we use the fact that for rotary-jointed robots

$$\frac{\partial j_k}{\partial \theta_i} = \begin{cases} \begin{bmatrix} (z_i \times z_k) \times p_k + z_k \times (z_i \times p_k) \\ z_i \times z_k \end{bmatrix} & i < k \\ \begin{bmatrix} z_k \times (z_i \times p_k) \\ 0 \end{bmatrix} & i \geq k \end{cases} \tag{119}$$

where z_l is the axis of rotation of the l-th joint and p_l is the vector from the l-th joint axis to the end effector. The expression for $\partial j_k / \partial \theta_i$ can be simplified to

$$\frac{\partial j_k}{\partial \theta_i} = \begin{cases} \begin{bmatrix} (z_i^T p_k) z_k - (z_i^T z_k) p_k \\ z_i \times z_k \end{bmatrix} & i < k \\ \begin{bmatrix} (z_k^T p_i) z_i - (z_i^T z_k) p_i \\ 0 \end{bmatrix} & i \geq k. \end{cases} \tag{120}$$

The gradient of $\mathcal{K}$,

$$\nabla \mathcal{K} = \left[\frac{\partial \mathcal{K}}{\partial \theta_1}\ \frac{\partial \mathcal{K}}{\partial \theta_2}\ \cdots\ \frac{\partial \mathcal{K}}{\partial \theta_7} \right]^T, \tag{121}$$

is then given by using (118) and (120).

The gradient of $\mathcal{I}$,

$$\nabla \mathcal{I} = \left[\frac{\partial \mathcal{I}}{\partial \theta_1}\ \frac{\partial \mathcal{I}}{\partial \theta_2}\ \cdots\ \frac{\partial \mathcal{I}}{\partial \theta_7} \right]^T, \tag{122}$$

is found in an analogous manner by taking the partial derivatives of

$$\mathcal{I} = \frac{\sigma_6}{\sigma_1} = \frac{u_6^T J v_6}{u_1^T J v_1}, \tag{123}$$

to obtain

$$\frac{\partial \mathcal{I}}{\partial \theta_i} = \frac{\sigma_1 \left(u_6^T \frac{\partial J}{\partial \theta_i} v_6 \right) - \sigma_6 \left(u_1^T \frac{\partial J}{\partial \theta_i} v_1 \right)}{\sigma_1^2} \tag{124}$$

Table 18: Modified anthropomorphic arm with optimized DH parameters

i	α_i [degrees]	a_i [m]	d_i [m]	θ_i [degrees]
1	-90	0	0	0°
2	90	2	0	23°
3	-90	0	0	132°
4	90	2	0	316°
5	-90	0	0	307°
6	90	0	0	273°
7	0	0	1.1	114°

where $\partial J/\partial\theta_i$ is given column wise by (120) and u_i and v_i are the ith columns of the SVD matrices U and V for J, respectively.

We then evaluated 21 robots with $a_2 = a_4 = a = 2$ and varying d_7 from 0 to $a = 2$ in increments of 0.1 by starting with the original one million randomly generated configurations and using the above approach to drive the robot to a configuration that maximized $\mathcal{K} + \mathcal{I}$. We then selected the optimal joint configuration to be the one that corresponds to the maximum value of $\mathcal{K} = 0.3672$, which resulted in

$$\theta = [0^\circ \ 23^\circ \ 132^\circ \ 316^\circ \ 307^\circ \ 273^\circ \ 114^\circ]^T , \tag{125}$$

and $d_7 = 1.1$. Fig. 37 illustrates an image of the robot at the optimized configuration and its DH parameters are given in Table 18.

Figure 38 illustrates that the design that we have found in (125), recalling its $\mathcal{K} = 0.3672$, is the maximum that the robot can achieve along the three orthogonal trajectories from the design point. (See Appendix E for different trajectories that illustrate the same fact.) Table 19 presents a comparison with the min robot from (98). They both have close results except for the six-dimensional fault tolerant volume, where the min robot from (98) is still ranked as the best among all the other robots by its highest percentage value.(See Appendix F to visualize the volumes.)

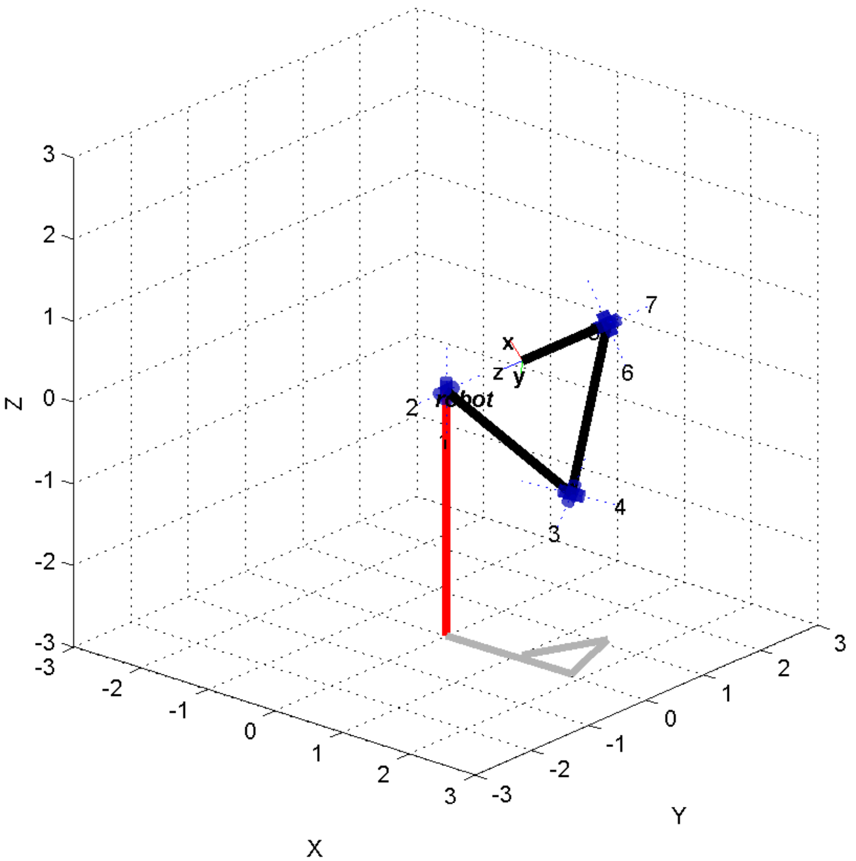


Figure 37: The locally optimal fault tolerant configuration at the design point of the modified anthropomorphic arm. (This graphic was generated using the Robotics Toolbox described in [61].)

Table 19: Comparison between the min robot from (98) and the modified anthropomorphic arm robot

Global measure	the min robot from (98)	the modified anthropomorphic arm robot
R_{max} [m]	6.0	5.1
V_r [%]	99	100
$V_{FT_{3d}}$ [%]	81	78
V_{6d} [%]	49	49
V_{FT} [%]	27	10

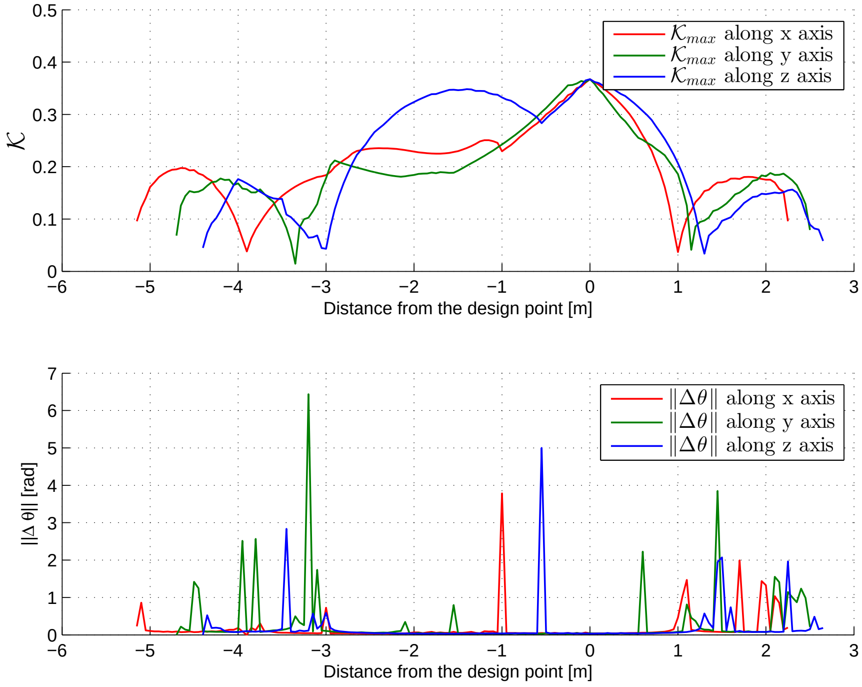


Figure 38: The locally optimal fault tolerant configuration at the design point of the modified anthropomorphic arm robot. (See Appendix E for different trajectories.)

CHAPTER VIII

CONCLUSIONS

8.1 SUMMARY

This work has shown that there are multiple different robot designs that possess the same desired Jacobian that has been selected to have desirable failure tolerance properties at a specific operating point. It has been shown that even though these manipulators all have the same local properties, their global properties can differ significantly, both in terms of pre-failure kinematics as well as post-failure performance. This can provide robot system designers with a great deal of flexibility when considering the different constraints that arise from different applications.

In the first part of this work, a mathematical analysis for planar manipulators has presented, based on the Gram matrix, that allows one to enumerate all of the possible planar manipulators that possess certain desired fault tolerance properties based on the form of a desired Jacobian. This analysis was illustrated on both a 3R manipulator experiencing a single locked joint failure and a 4R manipulator experiencing two joint failures. It was further shown that there are significant differences in the capabilities of the resulting manipulators, both in terms of pre- and post-failure performance. It was shown that some of these differences are related to the fact that the same Jacobian can result in manipulators that vary significantly in workspace area. However, it is quite surprising that major differences in behavior were also found in manipulator designs that were identical in terms of area.

The second part of this work has shown that one can parameterize the infinite family of four-DOF spatial positioning manipulators that correspond to an optimally failure tolerant Jacobian.

It was further shown that with an appropriate parametrization, one does not need to consider all possible permutations (and multiplications of ±1) of the columns of the Jacobian. Then a method for evaluating the global fault tolerant properties of the resulting manipulator designs was presented and used to illustrate how one would optimize the kinematic design for a given family of manipulators to obtain a robot that has a high degree of failure tolerance over its entire workspace, in addition to being locally optimal.

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

8.2 FUTURE DIRECTIONS

In terms of future work, it would be interesting to know how one can characterize self-motion manifolds for spatial manipulators and to know how many there are. Other future work would be to prove if its true that all feasible optimally fault tolerant Jacobians that are for 7R manipulators are really equivalent.

One could also extend this work to 8×6 optimally fault tolerant Jacobians. There are entire families of different Jacobians [57], in addition to their 8! of column permutations. This would give more flexibility to chose for a required design. The performance of manipulator designs may then be compared to 7R manipulators.

Finally, an interesting future development would be to extend the algorithm that was illustrated in Chapter 3 to be applicable for a Jacobian whose manipulator has any number of prismatic joints, since the current algorithm works only for the case of revolute manipulators.

REFERENCES

[1] K. M. Ben-Gharbia, A. A. Maciejewski, and R. G. Roberts, “An illustration of generating robots from optimal fault-tolerant Jacobians,” *15th IASTED Int. Conf. Robot. Applic.*, Cambridge, MA, Nov. 1-3, 2010, pp. 461–468.

[2] K. M. Ben-Gharbia, R. G. Roberts, and A. A. Maciejewski, “Examples of planar robot kinematic designs from optimally fault-tolerant Jacobians,” in *IEEE Int. Conf. on Robot. and Automat.*, Shanghai, China, May 9-13, 2011.

[3] K. M. Ben-Gharbia, A. A. Maciejewski, and R. G. Roberts, “Examples of spatial positioning redundant robotic manipulators that are optimally fault tolerant,” in *IEEE Int. Conf. on Syst., Man, Cybern.*, Anchorage, Alaska, Oct. 9-12, 2011, pp. 1526–1531.

[4] K. M. Ben-Gharbia, A. A. Maciejewski, and R. G. Roberts, “Kinematic design of redundant robotic manipulators for spatial positioning that are optimally fault tolerant,” *IEEE Trans. Robotics*, vol. 29, no. 5, pp. 1300–1307, 2013.

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

[6] K. M. Ben-Gharbia, A. A. Maciejewski, and R. G. Roberts, “An Example of a Seven Joint Manipulator Optimized for Kinematic Fault Tolerance,” *IEEE Int. Conf. on Syst., Man, Cybern.*, Diego, CA, Oct. 5-8, 2014, pp. xxx–xxx.

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

[8] G. Visentin and F. Didot, “Testing space robotics on the Japanese ETS-VII satellite,” ESA Bulletin-European Space Agency, pp. 61–65, Sep. 1999.

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

[10] W. H. McCulloch, “Safety analysis requirements for robotic systems in DOE nuclear facilities,” in *Proc. 2nd Specialty Conf. Robot. Challenging Environ.*, Albuquerque, NM, Jun. 1-6, 1996, pp. 235–240.

[11] S. Kawatsuma, M. Fukushima, and T. Okada, “Emergency response by robots to Fukushima-daiichi accident: Summary and lessons learned,” *Industrial Robot: An International Journal*, vol. 39, no. 5, pp. 428–435, 2012.

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

[13] K. N. Groom, A. A. Maciejewski, and V. Balakrishnan, “Real-time failure-tolerant control of kinematically redundant manipulators,” *IEEE Trans. Robot. Automat.*, vol. 15, no. 6, pp. 1109–1116, Dec. 1999.

[14] Reliability Information Analysis Center, “Nonelectronic parts reliability data,” Defense Technical Information Center / Air Force Research Lab, Rome, NY, USA, no. NPRD-2011, 2011.

[15] S. Cheng and B. S. Dhillon, “Reliability and availability analysis of a robot-safety system,” *Journal of Quality in Maintenance Engineering*, vol. 17, no. 2, pp. 203–232, 2011.

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

[17] Y. Ting, S. Tosunoglu, and B. Fernandez. “Control algorithms for fault-tolerant robots,” *IEEE Int. Conf. Robot. Automat.*, vol. 2, pp. 910–915, May 8–13, 1994.

[18] J. D. English and A. A. Maciejewski, “Fault tolerance for kinematically redundant manipulators: Anticipating free-swinging joint failures,” *IEEE Trans. Robot. Automat.*, vol. 14, no. 4, pp. 566–575, Aug. 1998.

[19] J. D. English and A. A. Maciejewski, “Failure Tolerance through active braking: A kinematic approach,” *Int. J. Robot. Res.*, vol. 20, no. 4, pp. 287–299, Apr. 2001.

[20] P. Nieminen, S. Esque, A. Muhammad, J. Mattila, J. Väyrynen, M. Siuko, and M. Vilenius, “Water hydraulic manipulator for fail safe and fault tolerant remote handling operations at ITER,” *Fusion Engineering and Design*, vol. 84, no. 7, pp. 1420–1424, 2009.

[21] D. L. Schneider, D. Tesar, and J. W. Barnes, “Development & testing of a reliability performance index for modular robotic systems,” in *Proc. Annual Rel. Maintain. Symp.*, Anaheim, CA, Jan. 24-27, 1994, pp. 263–271.

[22] B. S. Dhillon A. R. M. Fashandi and K. L. Liu, “Robot systems reliability and safety: A review,” *J. Quality Maintenance Engineering*, vol. 8, no. 3, pp. 170–212, 2002.

[23] C. J. J. Paredis and P. K. Khosla, “Designing fault-tolerant manipulators: How many degrees of freedom?,” *Int. J. Robot. Res.*, vol. 15, no. 6, pp. 611-628, Dec. 1996.

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

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

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

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

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

[29] M. Anand, T. Selvaraj, S. Kumanan, and J. Janarthanan, "A hybrid fuzzy logic artificial neural network algorithm-based fault detection and isolation for industrial robot manipulators," *Int. J. Manufact. Res.*, vol. 2, no. 3, pp. 279–302, 2007.

[30] D. Brambilla, L. Capisani, A. Ferrara, and P. Pisu, "Fault detection for robot manipulators via second-order sliding modes," *IEEE Trans. Industrial Electronics*, vol. 55, no. 11, pp. 3954–3963, Nov. 2008.

[31] L. Capisani, A. Ferrara, A. F. de Loza, and L. Fridman, "Manipulator fault diagnosis via higher order sliding-mode observers," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 10, pp. 3979–3986, 2012.

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

[33] X. Chen and S. Nof, “Error detection and prediction algorithms: Application in robotics,” *J. Intell. Robot. Syst.; Robotic Systems*, vol. 48, no. 2, pp. 225–252, 2007.

[34] M. Ji and N. Sarkar, “Supervisory fault adaptive control of a mobile robot and its application in sensor-fault accommodation,” *IEEE Trans. Robotics*, vol. 23, no. 1, pp. 174–178, Feb. 2007.

[35] A. De Luca and L. Ferrajoli, “A modified Newton-Euler method for dynamic computations in robot fault detection and control,” in *IEEE Int. Conf. Robot. Automat.*, May 2009, pp. 3359–3364.

[36] J. E. McInroy, J. F. O’Brien, and G. W. Neat, “Precise, fault-tolerant pointing using a Stewart platform,” *IEEE/ASME Trans. Mechatronics*, vol. 4, no. 1, pp. 91–95, Mar. 1999.

[37] M. Hassan and L. Notash, “Optimizing fault tolerance to joint jam in the design of parallel robot manipulators,” *Mech. Mach. Theory*, vol. 42, no. 10, pp. 1401–1417, 2007.

[38] Y. Chen, J. E. McInroy, and Y. Yi, “Optimal, fault-tolerant mappings to achieve secondary goals without compromising primary performance,” *IEEE Trans. Robotics*, vol. 19, no. 4, pp. 680–691, Aug. 2003.

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

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

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

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

[43] C. L. Lewis and A. A. Maciejewski. "Dexterity optimization of kinematically redundant manipulators in the presence of failures," *Computers and Electrical Engineering: An International Journal*, vol. 20, No. 3, pp. 273–288, May 1994.

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

[45] R. G. Roberts and A. A. Maciejewski, "A local measure of fault tolerance for kinematically redundant manipulators," *IEEE Trans. Robotics Automat.*, vol. 12, no. 4, pp. 543–552, Aug. 1996.

[46] R. G. Roberts, "On the local fault tolerance of a kinematically redundant manipulator," *J. Robotic Syst.*, vol. 13, no. 10, pp. 649–661, Oct. 1996.

[47] R. G. Roberts, H. G. Yu, and A. A. Maciejewski, "Fundamental limitations on designing optimally failure-tolerant kinematically redundant manipulators," *IEEE Trans. Robotics*, vol. 24, no. 5, pp. 1124–1237, Oct. 2008.

[48] C. J. J. Paredis and P. K. Khosla, "Fault tolerant task execution through global trajectory planning," *Rel. Eng. Syst. Safety*, vol. 53, pp. 225–235, 1996.

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

[50] R. S. Jamisola, Jr., A. A. Maciejewski, and R. G. Roberts,“Failure-tolerant path planning for kinematically redundant manipulators anticipating locked-joint failures” *IEEE Trans. Robotics*, vol. 22, no. 4, pp. 603–612, Aug. 2006.

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

[52] R. C. Hoover, R. G. Roberts, A. A. Maciejewski, P. S. Naik, and K. M. Ben-Gharbia, “Designing a Failure-Tolerant Workspace for Kinematically Redundant Robots,” accepted to appear in *IEEE Trans. Autom. Sci. Eng.*, 2014.

[53] T. Yoshikawa, “Manipulability of robotic mechanisms,” *Int. J. Robotics Res.*, vol. 4, no. 2, pp. 3–9, 1985.

[54] C. A. Klein and B. E. Blaho, “Dexterity measures for the design and control of kinematically redundant manipulators,” *Int. J. Robot. Res.*, vol. 6, no. 2, pp. 72–83, 1987.

[55] C. Klein and T. A. Miklos, “Spatial robotic isotropy,” *Int. J. Robot. Res.*, vol. 10, no. 4, pp. 426-437, 1991.

[56] K. E. Zanganeh and J. Angeles, “Kinematic isotropy and the optimum design of parallel manipulators,” *Int. J. Robot. Res.*, vol. 16, no. 2, pp. 185-197, Apr. 1997.

[57] A. A. Maciejewski and R. G. Roberts, “On the existence of an optimally failure tolerant 7R manipulator Jacobian,” *Applied Mathematics and Computer Science*, vol. 5, no. 2, pp. 343–357, 1995.

[58] J. Denavit and R. S. Hartenberg, A kinematic notation for lower-pair mechanisms based on matrices, *Trans ASME J. Appl. Mech*, 23, 1955, 215–221.

[59] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence* (New York: McGraw-Hill, 1987).

[60] A. A. Maciejewski and C. A. Klein, "SAM: Animation software for simulating articulated motion," *Computers and Graphics: An International Journal*, vol. 9, no. 4, pp. 383–391, 1985.

[61] P. I. Corke, "A robotics toolbox for MATLAB," *IEEE Robot. Autom. Mag.*, vol. 3, no. 1, pp. 2–32, Mar. 1996.

[62] C. Sanderson, "Armadillo: An Open Source C++ Linear Algebra Library for Fast Prototyping and Computationally Intensive Experiments". NICTA, St Lucia, QLD, Austalia, Tech. Rep., 2010.

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

[64] G. Marsaglia, "Choosing a point from the surface of a sphere," *The Annals of Mathematical Statistics*, vol. 43, no. 2, pp. 645–646, 1972.

[65] C. A. Klein and L.-C. Chu, "Comparison of extended Jacobian and Lagrange multiplier based methods for resolving kinematic redundancy," *Journal of Intelligent and Robotic Systems*, vol. 19, pp. 39–54, 1997.

APPENDIX A

MATLAB CODE OF COMPUTING DH PARAMETERS FROM A JACOBIAN

```
function [alpha a theta d Jc comp_robot R_02n]=J2DHv5(J)% J is n by 6 matrix


pi_prime=[];

n=size(J);

n=n(2);

pii_prime=[];diff=[];


R_02n=zeros(3,3);

if norm(J(4:6,1)-[0 0 1]')<1e-5 | norm(J(4:6,n)-[0 0 1]')>1e-5

    xn=cross(J(1:3,n),J(4:6,n))/norm(cross(J(4:6,n),J(1:3,n)));

    R_02n=[xn,cross(J(4:6,n),xn)/norm(cross(J(4:6,n),xn)),J(4:6,n)];

    J(4:6,:)=R_02n'*J(4:6,:);

    J(1:3,:)=R_02n'*J(1:3,:);

    J(4:6,n)=[0 0 1]';

J;

end


%initial value because we can't use index 0

J=[J [0 0 0 0 0 1]'];

x=[0 0 0]';

O=[0 0 0]';
```

```
alpha=0;

a=0;

theta=0;

d=0;


%note: the indexing i and i+1 is used instead of i-1 and i

for i=1:n

    %%

    v(:,i)=J(1:3,i);

    v(:,i+1)=J(1:3,i+1);

    z(:,i)=J(4:6,i);%==w

    z(:,i+1)=J(4:6,i+1);

    pi_prime=[pi_prime cross(z(:,i),v(:,i))]; %pi'=zi cross (zi cross pi)

    pii_prime=[pii_prime cross(z(:,i+1),v(:,i+1))];%pi+1'

    diff=[diff pii_prime(:,i)-pi_prime(:,i)];

    if norm(cross(z(:,i),z(:,i+1)))>1e-4

        x(:,i+1)=cross(z(:,i),z(:,i+1))/norm(cross(z(:,i),z(:,i+1)));

        if i==1

            x(:,i)=x(:,i+1);%x0 is arbitrary

        end

        if dot(x(:,i+1),diff(:,i))<0;

             x(:,i+1)=-x(:,i+1);

        end

    else

        x(:,i+1)=diff(:,i)/norm(diff(:,i));

    end

    %%

    %1: computing Alpha

    % cos alpha= zi dot zi+1
```

```
s=dot(z(:,i), z(:,i+1));% to solve the problem when the answer is very close ...
    to 1 and getting complex answer of acos
if abs(abs(s)−1)≤1e−4
    if s>0
        s=1;
    elseif s<0
        s=−1;
    end
end
alpha(i+1)=acos(s);
%define the sign of alpha
if dot(x(:,i+1),cross(z(:,i),z(:,i+1)))<0
     alpha(i+1)=−alpha(i+1);
end
%%
%2: computing the Link−Length
% x dot diffrence
a(i+1)=(dot(x(:,i+1),diff(:,i)));
%%
%3: computing Theta
% cos theta= x i+1 dot x i
if i==1
    theta(i+1)=0;%arbitrary value for theta1
else
    theta(i+1)=abs(acos((dot(x(:,i+1),x(:,i)))));
    if theta(i+1)≠0
        if dot(z(:,i),cross(x(:,i),x(:,i+1)))<0
            theta(i+1)=−theta(i+1);
        end
    end
end
%%
%4: computing d
%finding the cross points x_org
%then d =(O i+1 - O i) dot zi

if i==n
    O(:,i+1)=[0 0 0]';
elseif alpha(i+1)==0 |(abs(alpha(i+1))≤3.1416+1e-4 && abs(alpha(i+1))≥3.1416-1e-4)
    O(:,i+1)=O(:,i);
else
    pi_2prime=pi_prime(:,i)+(a(i+1))*x(:,i+1); %shifting the zi axis to zi+1 ...
        to find the corss point
    %we have 3 equations in 2 unkonws; we will choose any 2 so that det(A)≠0

    A{1,i}=[z(1:2,i) -z(1:2,i+1)];
    A{2,i}=[z(1,i) -z(1,i+1);z(3,i) -z(3,i+1)];
    A{3,i}=[z(2:3,i) -z(2:3,i+1)];

    diff_prime{1,i}=pii_prime(1:2,i)-pi_2prime(1:2);
    diff_prime{2,i}=[pii_prime(1,i);pii_prime(3,i)]-[pi_2prime(1);pi_2prime(3)];
    diff_prime{3,i}=pii_prime(2:3,i)-pi_2prime(2:3);
    det_A=abs([det(A{1,i}),det(A{2,i}),det(A{3,i})]);

    [num ind]=max(det_A);%picking the matrix that has the lagrer det
    t=A{ind,i}\diff_prime{ind,i};%same as inv(A)*b
    O(:,i+1)=pii_prime(:,i)+t(2)*z(:,i+1);
end
```

```
    if i==1
        d(i+1)=0;
    else
        d(i+1)=dot((O(:,i+1)-O(:,i)),z(:,i));
    end
end
alpha=alpha(2:n+1);
a=a(2:n+1);
theta=theta(2:n+1);
d=d(2:n+1);

disp('----------------------------------------------------------------------------')
disp(' ')
disp('    alpha        a      theta        d')
disp('    ------------------------------------')
disp([alpha'*180/pi a'    theta'*180/pi   d'])
```

APPENDIX B

SEARCHING FOR THE MAXIMUM $\mathcal{K}$ VALUE ONTO THE NULL SPACE

The maximum value of $\mathcal{K}$ is simply determined after calculating $\mathcal{K}$ for every possible configuration at a position in the workspace, i.e. the self-motion manifold. The exhaustive calculation was in computing the self-motion manifold. In this work, there were one-dimensional null space, as the case in the 3R planar and spatial positioning manipulators, and two-dimensional null space, as in 4R planar manipulators. Section B.1 will illustrate the computations of maximum $\mathcal{K}$ with single locked joint failure for one-dimensional null space case, while the two-dimensional null space case where two locked joint failures are assumed is presented in Section B.2.

B.1 ONE-DIMENSIONAL NULL SPACE

Let P_i be the position the workspace that one want to maximize $\mathcal{K}$ at, and let θ_0 be the initial configuration at P_i. The following steps will explain the algorithm used for optimizing $\mathcal{K}$ by computing the self-motion manifold:

1. Name $\theta = \theta_0$.

2. Compute $J(\theta)$.

3. Calculate $\|n_J\|$ (the null vector of $J(\theta)$.)

4. scale n_J such that $\|n_J\| = \alpha\|n_J\|$, where α is the step size within the null space in radius.

5. Recalculate θ such that $\theta = \theta + n_J$.

6. Recompute $J(\theta)$.

7. Calculate $\dot{\theta}$ using Damped Least Squares (DLS) inverse kinematics (see (130) below).

8. Save the result in θ such that $\theta = \theta + \dot{\theta}$.

9. Compute $J(\theta)$.

10. Compute $\mathcal{K}$ at this configuration, and save it with θ value if it is the maximum.

11. Redo steps 3–10 until $\theta \approx \theta_0$.

The DLS inverse kinematics equation (the is used in step 7) is

$$\dot{\theta} = J^T(JJ^T + \lambda I)^{-1}\dot{x}_c, \tag{126}$$

where $\dot{\theta}$ is the joint velocity, λ is a scalar value that can be optimized to minimize

$$\|\dot{x}_c - J\dot{\theta}\|^2 + \lambda^2\|\dot{\theta}\|^2, \tag{127}$$

I is identity matrix, and $\dot{x}_c$ is the position error feedback signal that is used to match both the velocity and the position, such that

$$\dot{x}_c = \dot{x}_D + K_p(x_D - x_A)/\Delta t, \tag{128}$$

where $\dot{x}_D$ is the desired position velocity, x_D and x_A are the desired and the actual positions respectively, Δt is time step, and $0 \leq K_p \leq 1$ ($K_p = 0.5$ was used.)

In step 3, one way to compute the null vector is to compute the SVD of J (Recalling $J = UDV^T$ from Chapter 2), which is already required for step 10, then n_J will be equal to the last column of V. Note that there should not be a big change in the direction of n_J from the previous iteration. If that happens, an oscillation in the motion will occur between two or more adjacent configurations. To fix that, one can add a comparison condition with the pervious n_J vector and flip the current n_J direction if the angle between them is too large, e.g. $> 90°$. For a 3R planar manipulator, one can also use the canonical null vector form such that

$$n_J = \begin{bmatrix} a_2a_3\sin(\theta_3) \\ -a_2a_3\sin(\theta_3) - a_1a_3\sin(\theta_2+\theta_3) \\ a_1a_2\sin(\theta_2) + a_1a_3\sin(\theta_2+\theta_3) \end{bmatrix}. \tag{129}$$

(Recalling that a_i and θ_i are the kinematic parameters of a planar manipulator from 2.) The following section will present the computation of maximum $\mathcal{K}$ in the two-dimensional null space case.

B.2 TWO-DIMENSIONAL NULL SPACE

Two locked joint failures were assumed in the 4R planar manipulators. Accordingly, one can compute σ_{min} for all possible post-failure Jacobians with two columns being removed; consequently, $\mathcal{K}$ will be the smallest value among all σ_{min}[1]. With two-dimensional space, it is complicated to compute the self-motion manifold for every P_i position in the workspace and compute the maximum value of $\mathcal{K}$. Instead, in this work $\mathcal{K}$ was maximized using its gradient projection ($\nabla\mathcal{K}$) [13] in the null space, such that

$$\dot{\theta} = J^{+}\dot{x} + \left(\sum_{i=r+1}^{n} v_i v_i^T\right)\nabla\mathcal{K}, \tag{130}$$

where J^{+} is the pseudoinverse of J, $\dot{x}$ is the positional velocity, r is the rank of J, n the number J columns, and v_i is the column i of the SVD matrix V. The term $\left(\sum_{i=r+1}^{n} v_i v_i^T\right)$ is the projection operation in the null space.

The following steps illustrate the algorithm used to maximize the $\mathcal{K}$ value, assuming that one has the initial joint values θ_0 at P_i :

1. Name $\theta = \theta_0$.

2. Compute $J(\theta)$.

3. Calculate the null projection $\left(\sum_{i=r+1}^{n} v_i v_i^T\right)$

4. Compute $\nabla\mathcal{K}$.

5. Make $\|\nabla\mathcal{K}\| \leq a$ ($a = 0.5$ was used here.)

[1] Equation (5) in Chapter 2 describes the case with single locked joint failure.

6. Compute a scalar $\alpha = \mathcal{K}' - \mathcal{K}$ ($\mathcal{K}'$ is the next smallest calculated value of $\sigma_{min} > \mathcal{K}$ from all post-failure Jacobians.)

7. $\theta = \theta + \alpha \left(\sum_{i=r+1}^{n} v_i v_i^T \right) \nabla \mathcal{K}$.

8. Recompute $J(\theta)$.

9. Calculate $\dot{\theta}$ using DLS inverse kinematics.

10. Save the result in θ such that $\theta = \theta + \dot{\theta}$.

11. Compute $J(\theta)$.

12. Compute $\mathcal{K}$ at this configuration, and save it if with θ value if it is the maximum.

13. Redo steps 3–12 until either one of the following events happen:

    (a) $\nabla \mathcal{K} \approx 0$.

    (b) maximum iteration would be reached.

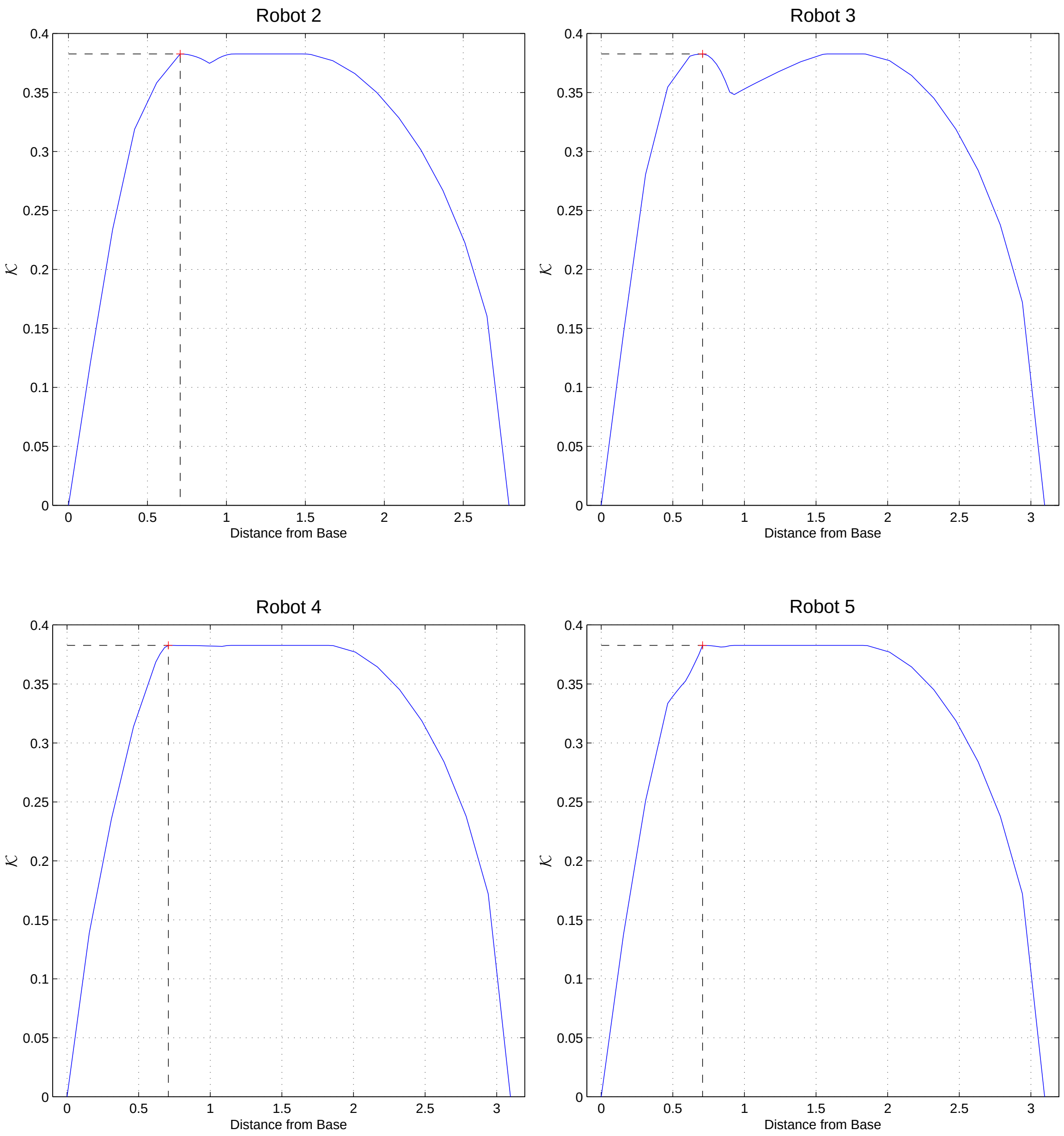
In step 6, α was used to solve the issue when the failure index of $\mathcal{K}$ is not unique. At this point, $\mathcal{K}$ is not differentiable [13].

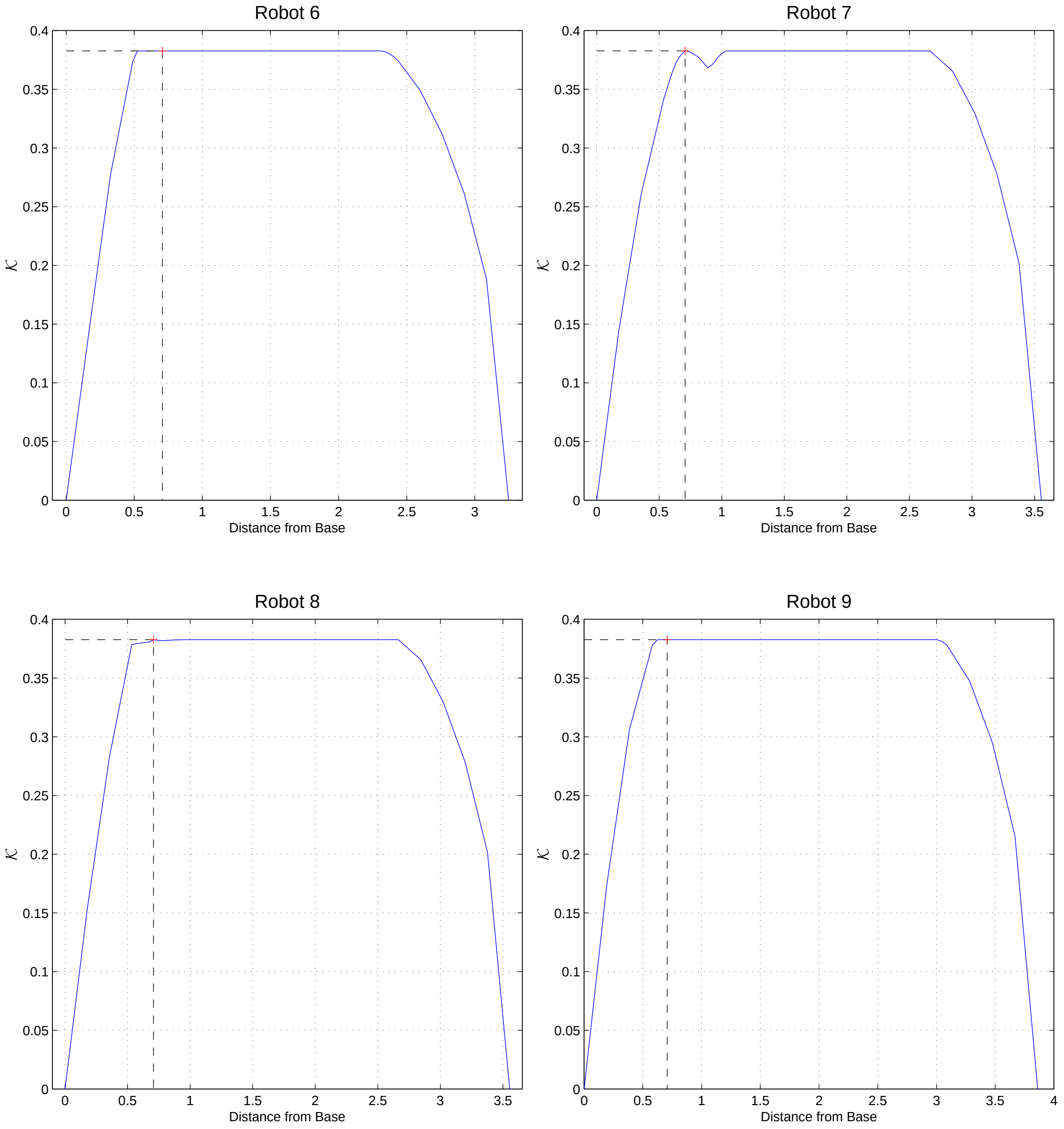
The only issue with this algorithm is that one can not be sure if the maximum value of $\mathcal{K}$ is the global maximum. In this work, most of the time the maximum evaluated $\mathcal{K}$ by performing the above steps was a local maxima. To do more searching for the global value, one can redo the above steps using the best configurations from the previous calculations of the adjacent locations P_{i-1} and P_{i+1}. The DLS inverse kinematics formula can be used first to drive the robot to the current P_i location, and that value will be the new value of θ_0.

Finally, more random θ_0 values were selected and the above steps were repeated in order to increase the probability of being on the global maxima.

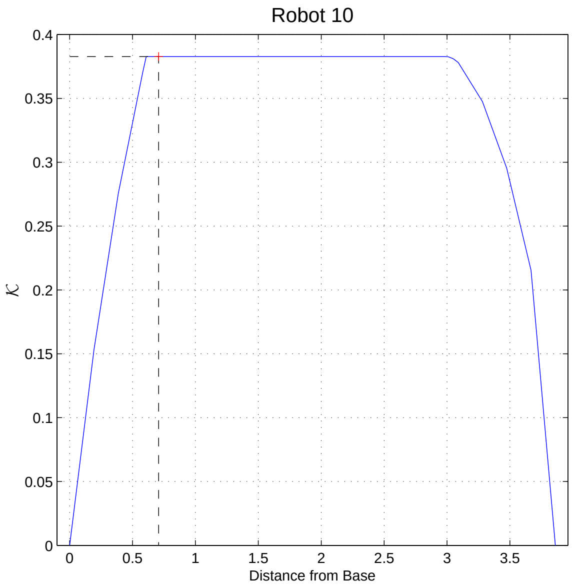
APPENDIX C

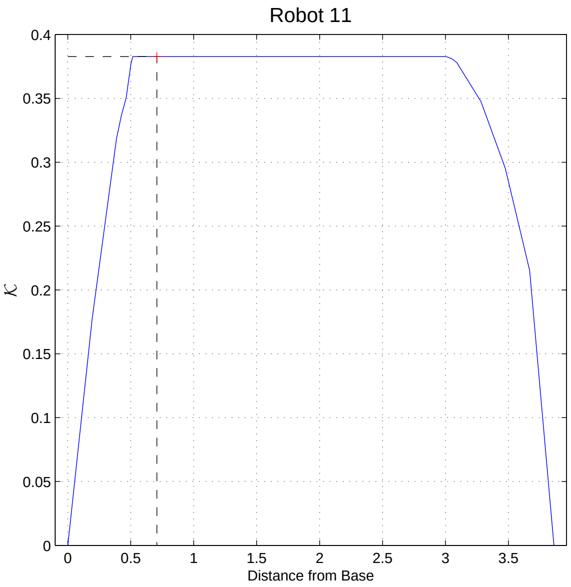
FOUR DOF PLANAR MANIPULATORS IN TABLE 3: MAXIMUM $\mathcal{K}$ PLOTS VERSUS DISTANCE FROM BASE

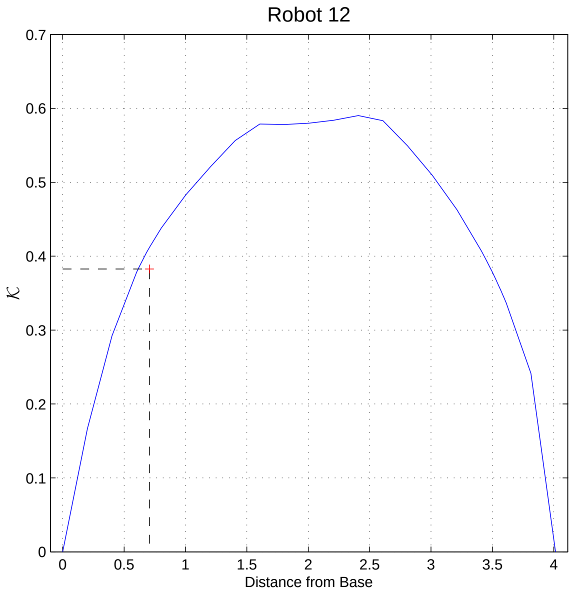


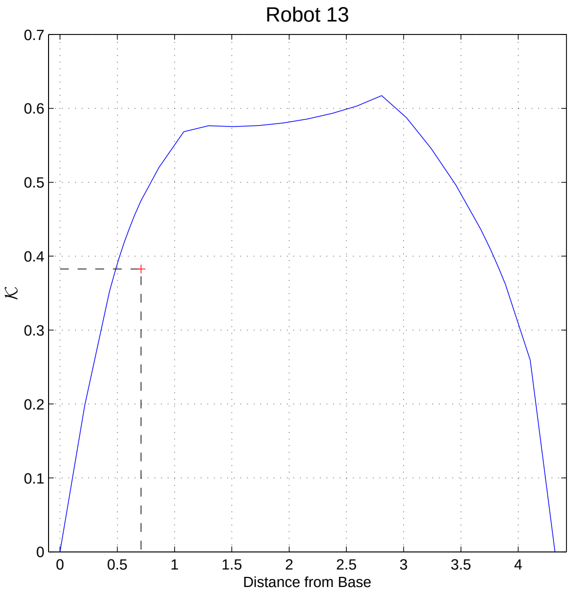


Robot 6
Robot 7
Robot 8
Robot 9
$\mathcal{K}$
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0
0.5
1
1.5
2
2.5
3
3.5
4
Distance from Base


Robot 10


Robot 11


Robot 12


Robot 13

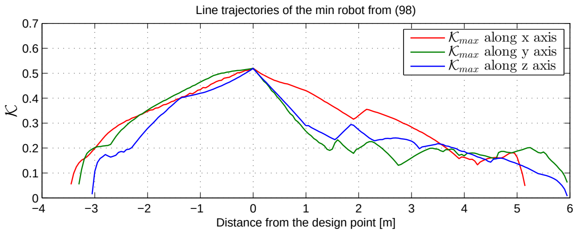
APPENDIX D

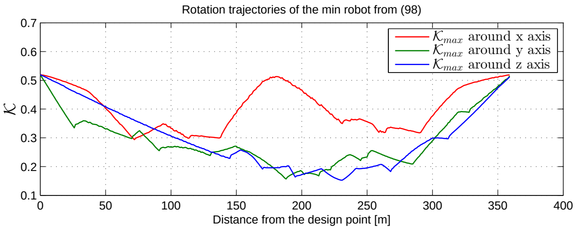
TABLE OF RESULTS FOR ROBOT GROUPS 4-7 IN TABLE 6

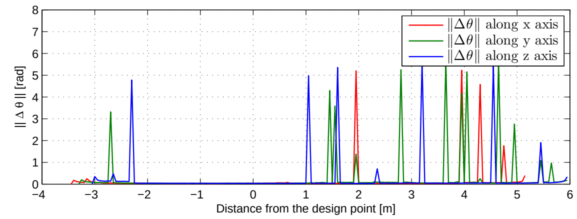
Robot Group	$(\alpha_1,\ \alpha_2,\ \alpha_3,\ \alpha_4)$ [degrees]	$(\beta_1,\ \beta_2,\ \beta_3,\ \beta_4)$ [degrees]	$W_{\mathcal{K}}$ [%]	max reach [m]
4	(0, 90, 0, 0)	(90, 90, 240, 240)	59	5.19
	(0, 90, 0, 0)	(90, 90, 60, 60)	53	3.83
	(0, 90, 180, 0)	(90, 90, 240, 60)	51	4.78
	(180, 90, 0, 0)	(90, 270, 240, 240)	42	3.45
	(180, −90, 0, 0)	(90, 270, 60, 60)	41	4.78
	(0, 90, 180, 0)	(90, 90, 60, 240)	37	3.43
	(180, −90, 180, 0)	(90, 270, 60, 240)	33	4.37
	(180, −90, 180, 0)	(90, 270, 240, 60)	31	3.05
5	(0, 90, 90, 0)	(90, 90, 240, 330)	30	4.96
	(0, 90, −90, 0)	(90, 90, 240, 150)	26	4.61
	(180, −90, −90, 0)	(90, 270, 60, 150)	13	4.55
	(0, −90, −90, 0)	(90, 90, 60, 150)	12	3.28
	(180, −90, 90, 0)	(90, 270, 60, 330)	11	4.20
	(0, −90, 90, 0)	(90, 90, 60, 330)	8	3.97
	(180, 90, 90, 0)	(90, 270, 240, 330)	6	2.90
	(180, 90, −90, 0)	(90, 270, 240, 150)	3	3.59
6	(−90, 0, −90, 0)	(150, 30, 120, 180)	67	3.92
	(−90, 180, −90, 0)	(150, 210, 120, 180)	52	4.88
	(−90, 0, 90, 0)	(150, 210, 300, 0)	48	5.16
	(−90, 0, −90, 0)	(150, 210, 300, 180)	31	4.00
	(−90, 180, −90, 0)	(150, 30, 300, 180)	26	2.41
	(−90, 0, 90, 0)	(150, 30, 120, 0)	4	2.41
	(−90, 180, 90, 0)	(150, 30, 300, 0)	3	3.33
	(−90, 180, 90, 0)	(150, 210, 120, 0)	3	3.41
7	(90, 90, 0, 0)	(0, 270, 240, 240)	48	3.93
	(90, 90, 0, 0)	(0, 90, 240, 240)	40	4.93
	(90, −90, 180, 0)	(0, 270, 240, 60)	32	3.52
	(90, 90, 0, 0)	(0, 90, 60, 60)	21	3.23
	(90, −90, 0, 0)	(0, 270, 60, 60)	21	4.57
	(90, 90, 180, 0)	(0, 90, 240, 60)	19	4.52
	(90, 90, 180, 0)	(0, 90, 60, 240)	12	2.83
	(90, −90, 180, 0)	(0, 270, 60, 240)	6	4.16

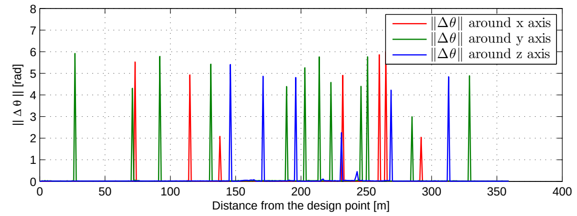
APPENDIX E

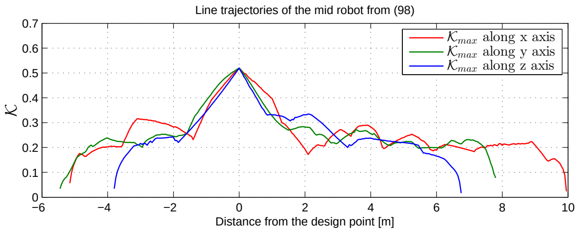
SEVEN DOF SPATIAL MANIPULATORS IN CHAPTER 7: MAXIMUM $\mathcal{K}$ TRJECTORIES

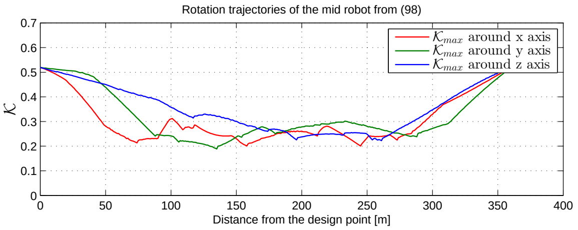


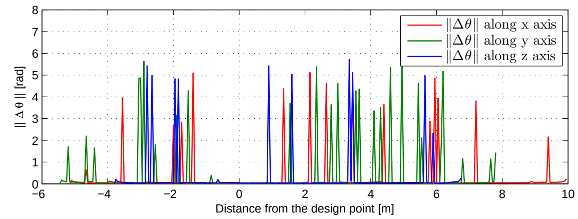


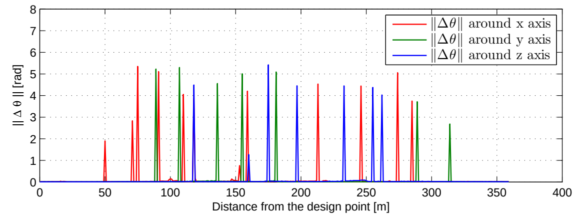




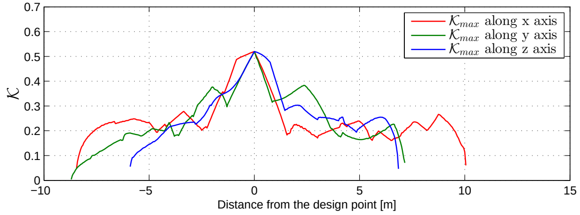


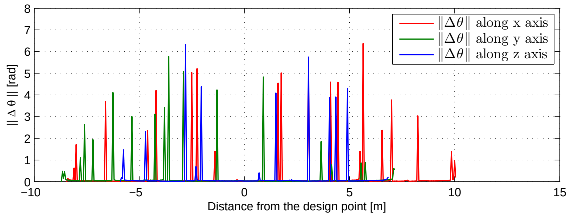




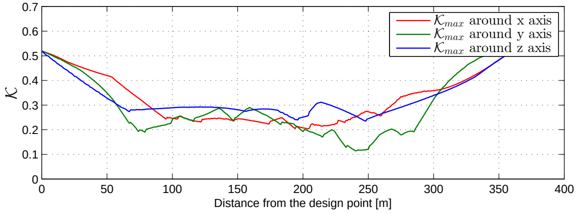


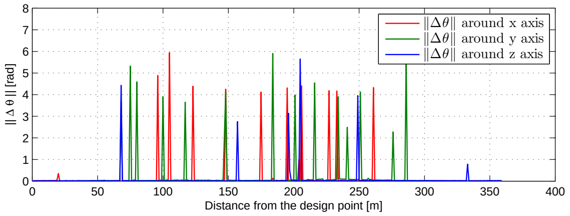
Line trajectories of the max robot from (98)



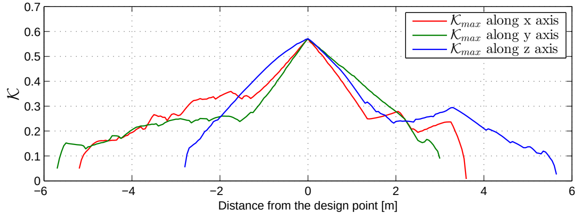


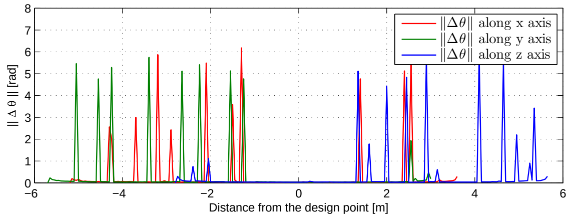
Rotation trajectories of the max robot from (98)



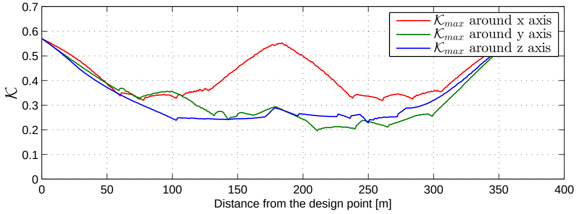


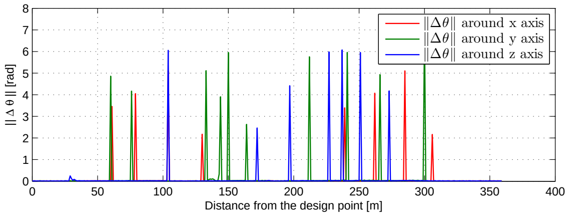
Line trajectories of the min robot from (99)

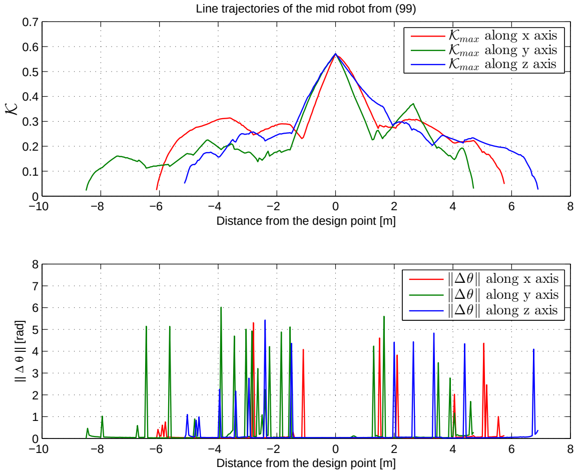


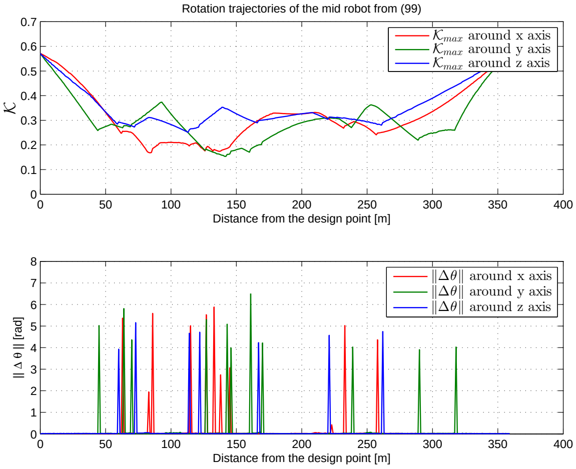


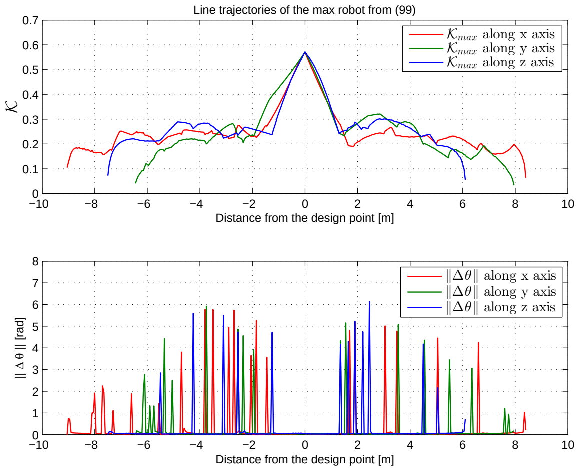
Rotation trajectories of the min robot from (99)

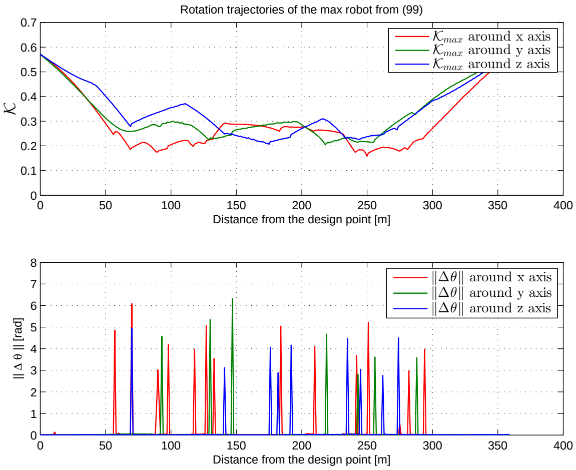


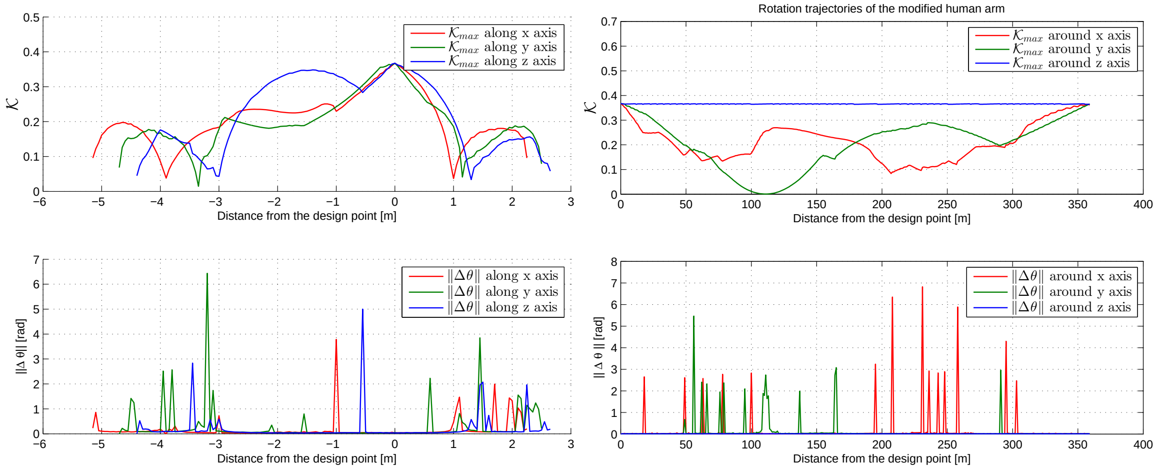



Line trajectories of the mid robot from (99)
$\mathcal{K}$
$\mathcal{K}_{max}$ along x axis
$\mathcal{K}_{max}$ along y axis
$\mathcal{K}_{max}$ along z axis
Distance from the design point [m]
|| Δ θ || [rad]
$\|\Delta\theta\|$ along x axis
$\|\Delta\theta\|$ along y axis
$\|\Delta\theta\|$ along z axis
Distance from the design point [m]


Rotation trajectories of the mid robot from (99)
$\mathcal{K}$
$\mathcal{K}_{max}$ around x axis
$\mathcal{K}_{max}$ around y axis
$\mathcal{K}_{max}$ around z axis
Distance from the design point [m]
|| Δ θ || [rad]
$\|\Delta\theta\|$ around x axis
$\|\Delta\theta\|$ around y axis
$\|\Delta\theta\|$ around z axis
Distance from the design point [m]


Line trajectories of the max robot from (99)
$\mathcal{K}$
$\mathcal{K}_{max}$ along x axis
$\mathcal{K}_{max}$ along y axis
$\mathcal{K}_{max}$ along z axis
Distance from the design point [m]
|| Δ θ || [rad]
$\|\Delta\theta\|$ along x axis
$\|\Delta\theta\|$ along y axis
$\|\Delta\theta\|$ along z axis
Distance from the design point [m]


Rotation trajectories of the max robot from (99)
$\mathcal{K}$
$\mathcal{K}_{max}$ around x axis
$\mathcal{K}_{max}$ around y axis
$\mathcal{K}_{max}$ around z axis
Distance from the design point [m]
|| Δ θ || [rad]
$\|\Delta\theta\|$ around x axis
$\|\Delta\theta\|$ around y axis
$\|\Delta\theta\|$ around z axis
Distance from the design point [m]


Rotation trajectories of the modified human arm
$\mathcal{K}_{max}$ along x axis
$\mathcal{K}_{max}$ along y axis
$\mathcal{K}_{max}$ along z axis
$\mathcal{K}$
Distance from the design point [m]
$\mathcal{K}_{max}$ around x axis
$\mathcal{K}_{max}$ around y axis
$\mathcal{K}_{max}$ around z axis
$\mathcal{K}$
Distance from the design point [m]
$\|\Delta\theta\|$ along x axis
$\|\Delta\theta\|$ along y axis
$\|\Delta\theta\|$ along z axis
||Δ θ|| [rad]
Distance from the design point [m]
$\|\Delta\theta\|$ around x axis
$\|\Delta\theta\|$ around y axis
$\|\Delta\theta\|$ around z axis
|| Δ θ || [rad]
Distance from the design point [m]
0.5
0.4
0.3
0.2
0.1
0
−6
−5
−4
−3
−2
−1
0
1
2
3
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
50
100
150
200
250
300
350
400
7
6
5
4
3
2
1
0
8

APPENDIX F

SEVEN DOF SPATIAL MANIPULATORS IN CHAPTER 7: VOLUME PLOTS

This appendix presents the six-dimensional fault tolerant volume for both cases of $\mathcal{K} > 0.2$ and $\mathcal{K} > 0$ for every robot in Chapter 7. The orientation volume is represented by its distribution (in color map) within the three-dimensional volume. (Its percentage value out of $V_{o_{max}} = \pi^2$ is plotted.) One can recall the discussion of Figures 34 and 35 in Subsection 7.4.1 to help understanding the following Figures. Evaluating $\mathcal{K} > 0.2$ and $\mathcal{K} > 0$ has been done by $\nabla\mathcal{K}$ in Section F.1, while searching over all self-motion manifolds is in Section F.2.

The remainder of the page is intensionally left empty.

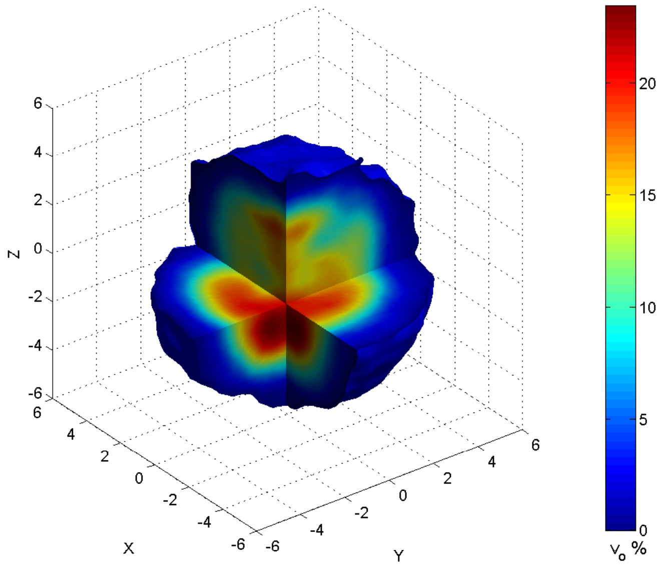


Figure 39: Six-dimensional fault tolerant volume for $\mathcal{K} > 0.2$ for the min robot from (98) with using $\nabla\mathcal{K}$, where $V_{FT} = 3.3\%$ and $V_{FT_{3d}} = 68\%$. Note that the color scale is not up to 100%.

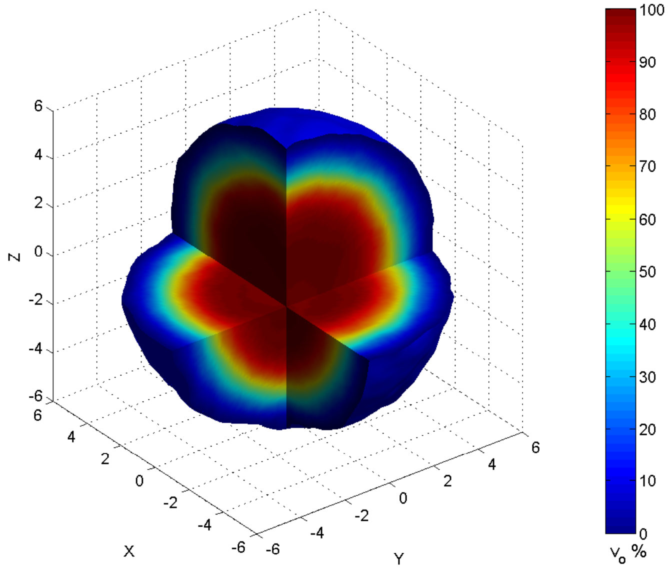


Figure 40: Six-dimensional fault tolerant volume for $\mathcal{K} > 0$ for the min robot from (98) with using $\nabla\mathcal{K}$, where $V_{FT} = 40\%$ and $V_{FT_{3d}} = 97\%$.

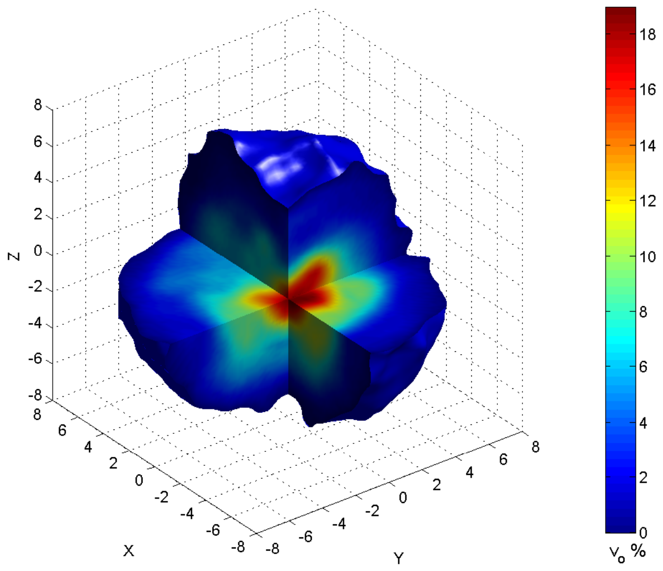


Figure 41: Six-dimensional fault tolerant volume for $\mathcal{K} > 0.2$ for the mid robot from (98) with using $\nabla\mathcal{K}$, where $V_{FT} = 1.5\%$ and $V_{FT_{3d}} = 62\%$. Note that the color scale is not up to 100%.

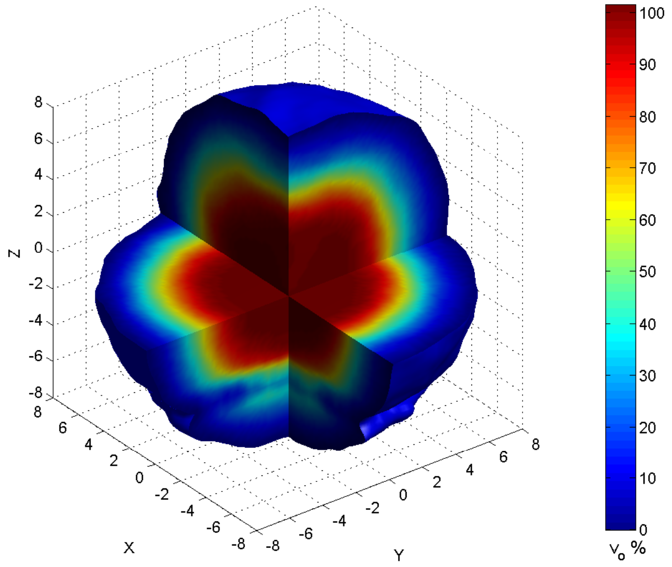


Figure 42: Six-dimensional fault tolerant volume for $\mathcal{K} > 0$ for the mid robot from (98) with using $\nabla\mathcal{K}$, where $V_{FT} = 32\%$ and $V_{FT_{3d}} = 93\%$.

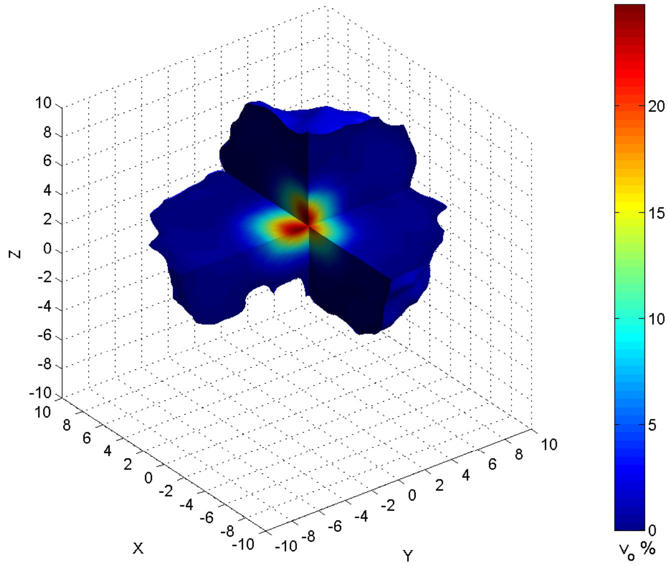


Figure 43: Six-dimensional fault tolerant volume for $\mathcal{K} > 0.2$ for the max robot from (98) with using $\nabla\mathcal{K}$, where $V_{FT} = 0.4\%$ and $V_{FT_{3d}} = 39\%$. Note that the color scale is not up to 100%.

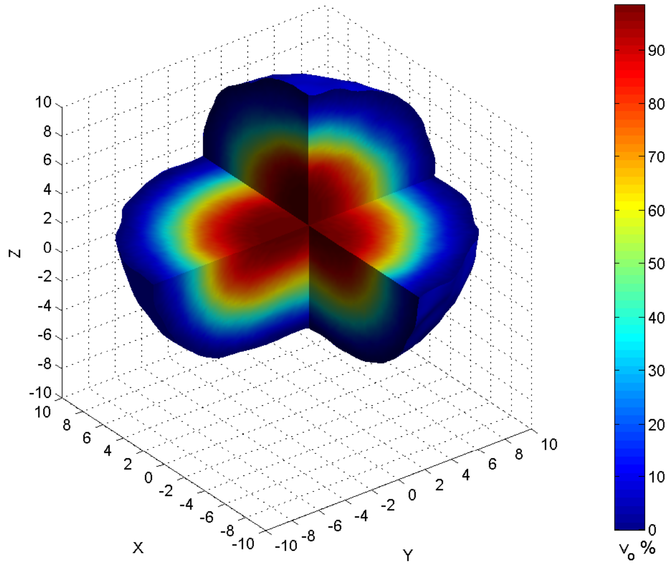


Figure 44: Six-dimensional fault tolerant volume for $\mathcal{K} > 0$ for the max robot from (98) with using $\nabla\mathcal{K}$, where $V_{FT} = 18\%$ and $V_{FT_{3d}} = 60\%$.

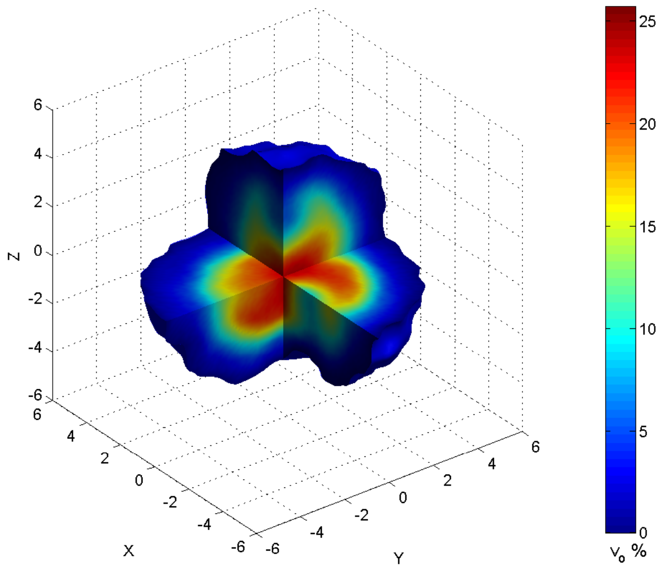


Figure 45: Six-dimensional fault tolerant volume for $\mathcal{K} > 0.2$ for the min robot from (99) with using $\nabla\mathcal{K}$, where $V_{FT} = 3.1\%$ and $V_{FT_{3d}} = 70\%$. Note that the color scale is not up to 100%.

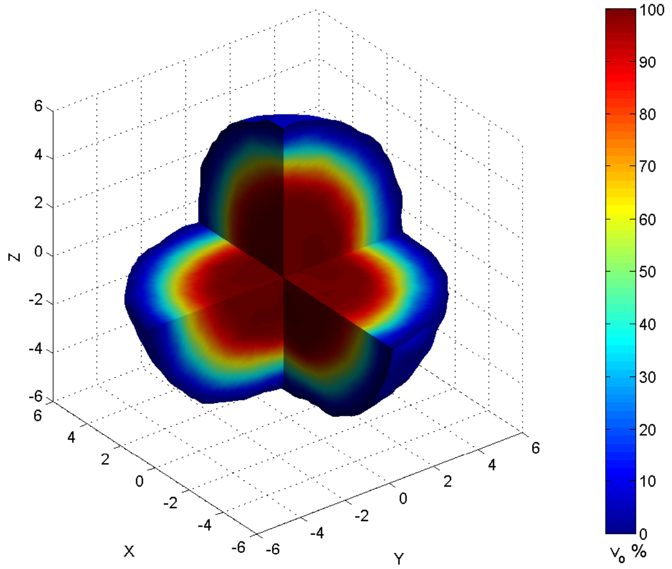


Figure 46: Six-dimensional fault tolerant volume for $\mathcal{K} > 0$ for the min robot from (99) with using $\nabla\mathcal{K}$, where $V_{FT} = 40\%$ and $V_{FT_{3d}} = 95\%$.

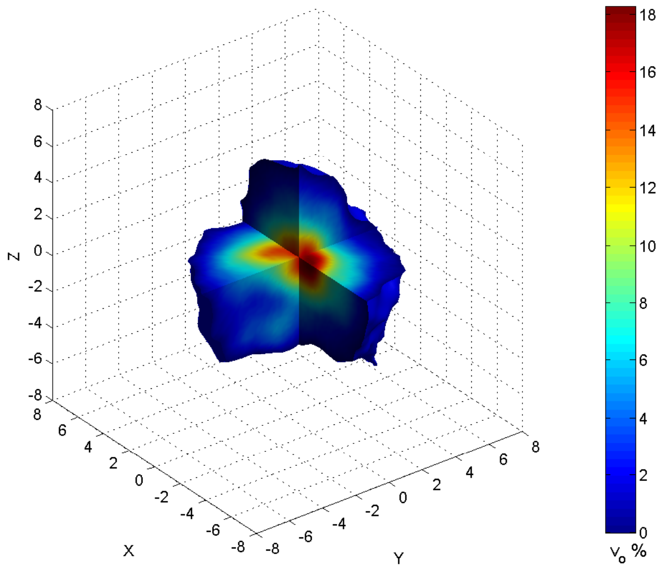


Figure 47: Six-dimensional fault tolerant volume for $\mathcal{K} > 0.2$ for the mid robot from (99) with using $\nabla\mathcal{K}$, where $V_{FT} = 0.7\%$ and $V_{FT_{3d}} = 33\%$. Note that the color scale is not up to 100%.

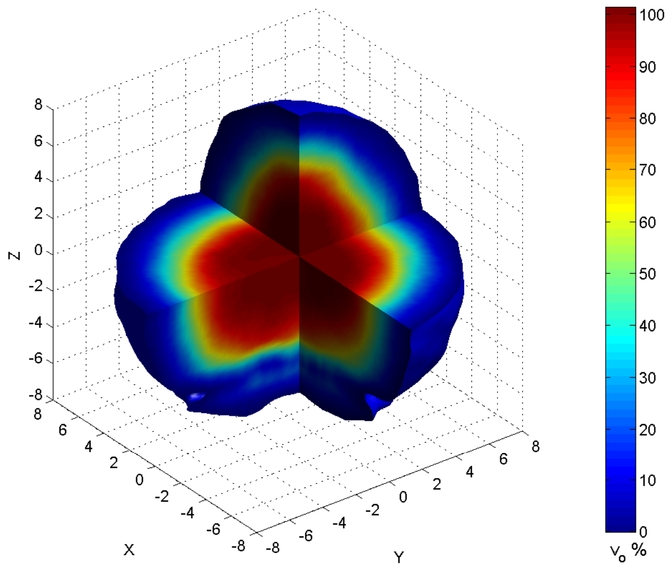


Figure 48: Six-dimensional fault tolerant volume for $\mathcal{K} > 0$ for the mid robot from (99) with using $\nabla\mathcal{K}$, where $V_{FT} = 30\%$ and $V_{FT_{3d}} = 91\%$.

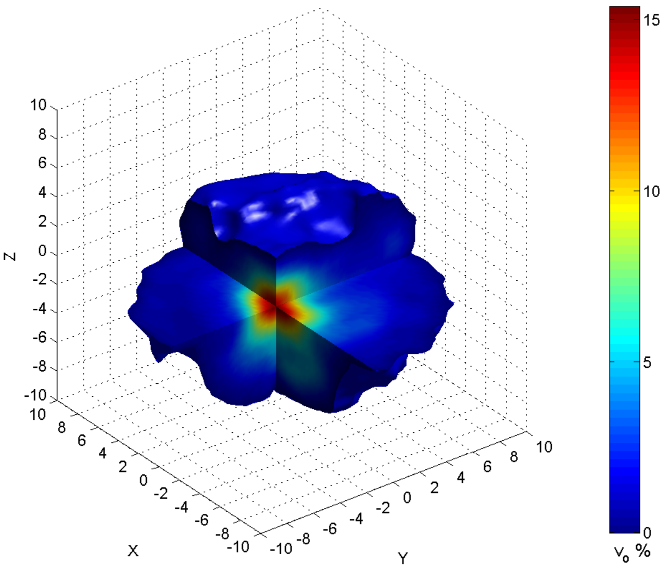


Figure 49: Six-dimensional fault tolerant volume for $\mathcal{K} > 0.2$ for the max robot from (99) with using $\nabla\mathcal{K}$, where $V_{FT} = 0.6\%$ and $V_{FT_{3d}} = 41\%$. Note that the color scale is not up to 100%.

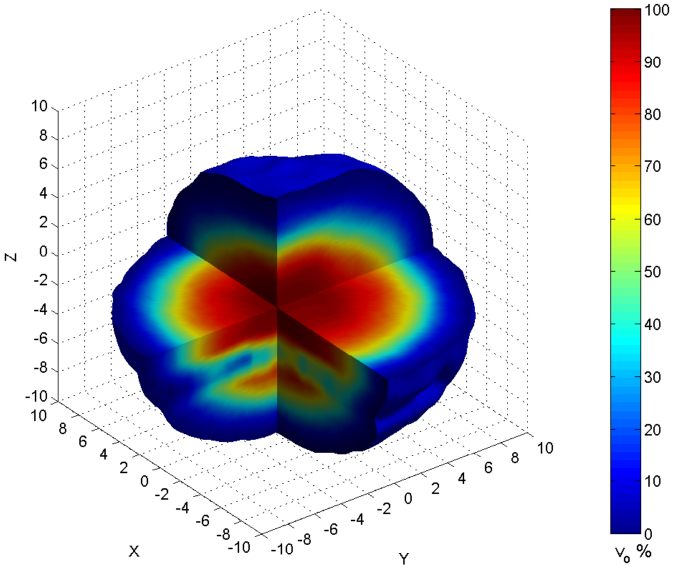


Figure 50: Six-dimensional fault tolerant volume for $\mathcal{K} > 0$ for the max robot from (99) with using $\nabla\mathcal{K}$, where $V_{FT} = 19\%$ and $V_{FT_{3d}} = 66\%$.

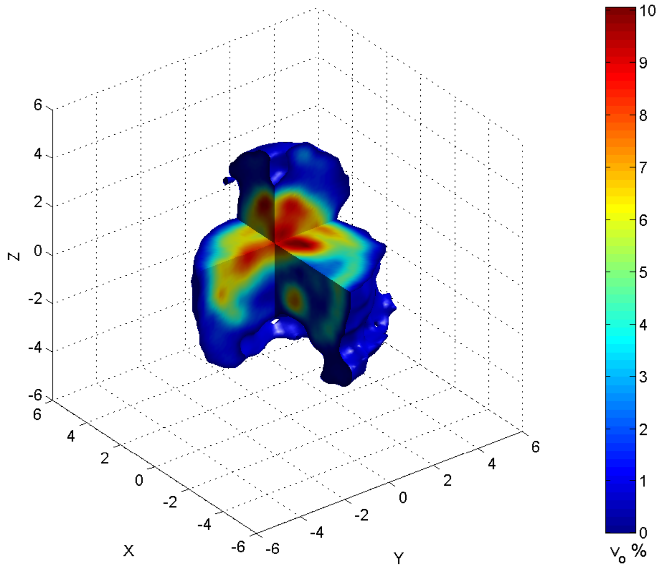


Figure 51: Six-dimensional fault tolerant volume for $\mathcal{K} > 0.2$ for the modified human arm robot with using $\nabla\mathcal{K}$, where $V_{FT} = 1.3\%$ and $V_{FT_{3d}} = 63\%$. Note that the color scale is not up to 100%.

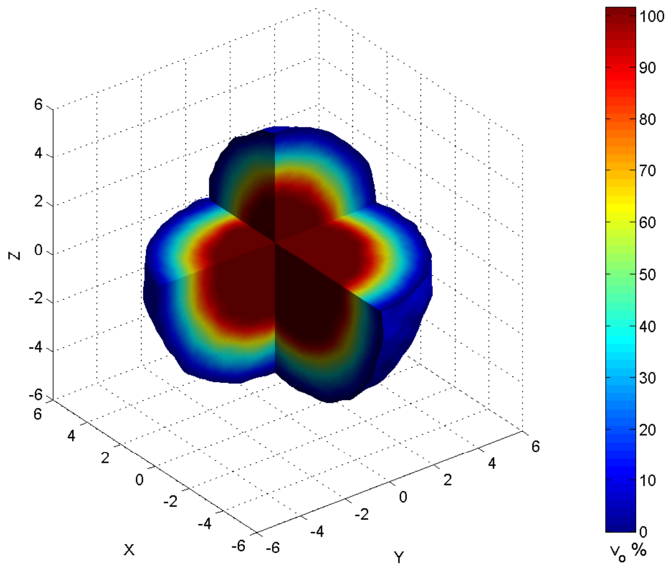


Figure 52: Six-dimensional fault tolerant volume for $\mathcal{K} > 0$ for the modified human arm robot with using $\nabla\mathcal{K}$, where $V_{FT} = 43\%$ and $V_{FT_{3d}} = 100\%$.

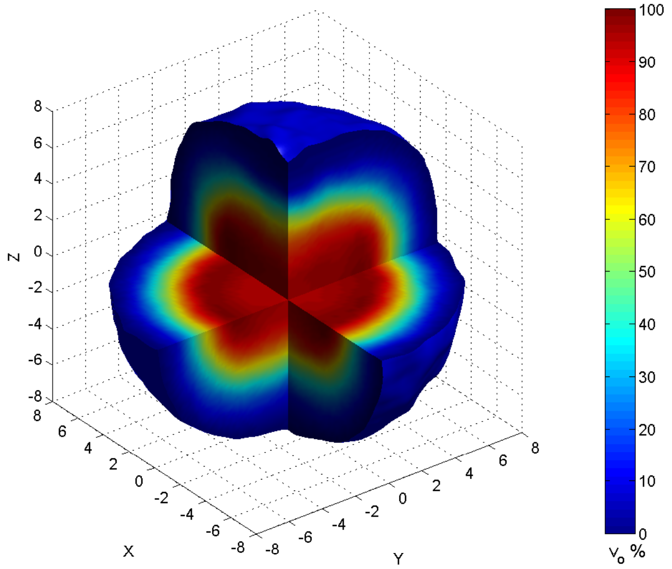


Figure 53: Six-dimensional fault tolerant volume for $\mathcal{K} > 0.2$ for the mid robot from (98) with tracking the maximum $\mathcal{K}$ over all self-motion manifolds, where $V_{FT} = 24\%$ and $V_{FT_{3d}} = 81\%$.

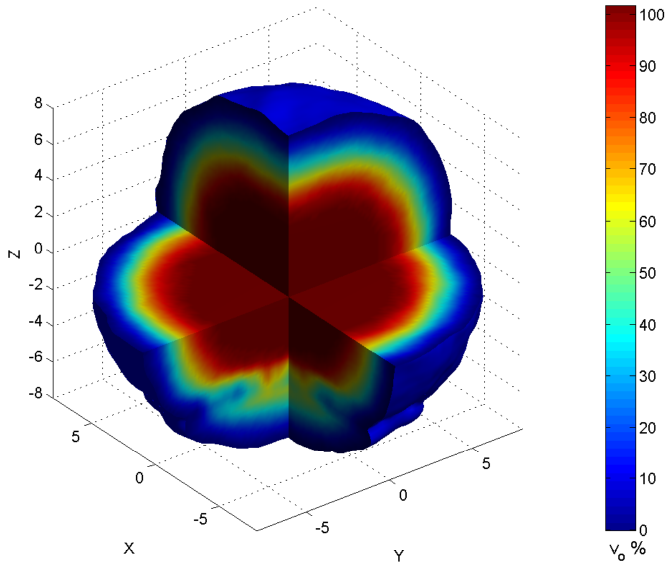


Figure 54: Six-dimensional reachable volume for the mid robot from (98), where $V_r = 49\%$ and $V_{6d} = 99\%$.

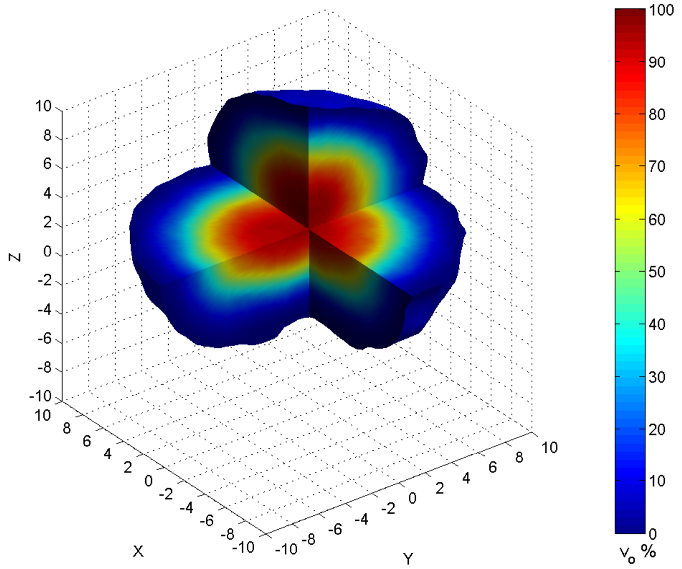


Figure 55: Six-dimensional fault tolerant volume for $\mathcal{K} > 0.2$ for the max robot from (98) with tracking the maximum $\mathcal{K}$ over all self-motion manifolds, where $V_{FT} = 12\%$ and $V_{FT_{3d}} = 51\%$.

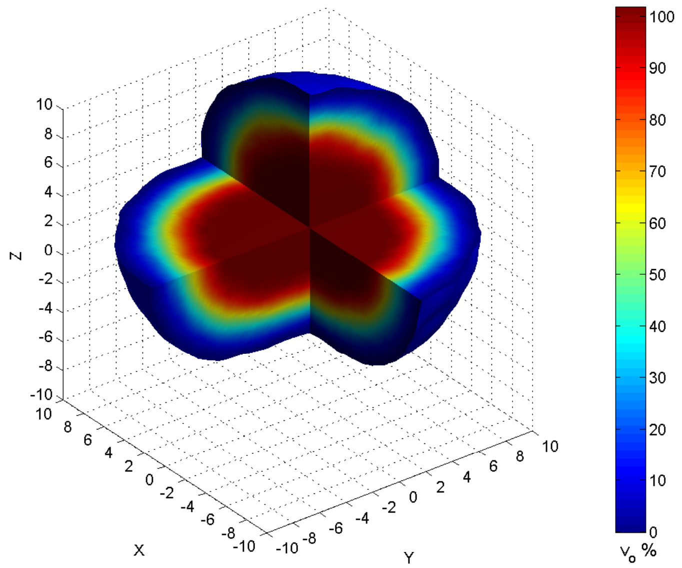


Figure 56: Six-dimensional reachable volume for the max robot from (98), where $V_r = 62\%$ and $V_{6d} = 26\%$.

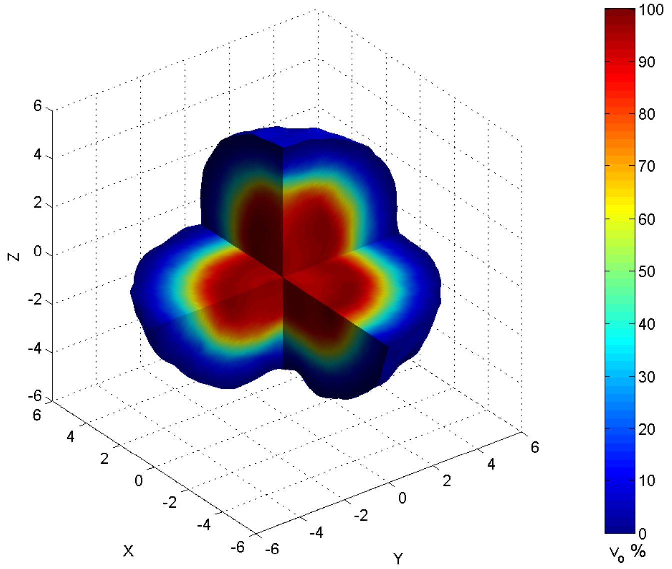


Figure 57: Six-dimensional fault tolerant volume for $\mathcal{K} > 0.2$ for the min robot from (99) with tracking the maximum $\mathcal{K}$ over all self-motion manifolds, where $V_{FT} = 26\%$ and $V_{FT_{3d}} = 81\%$.

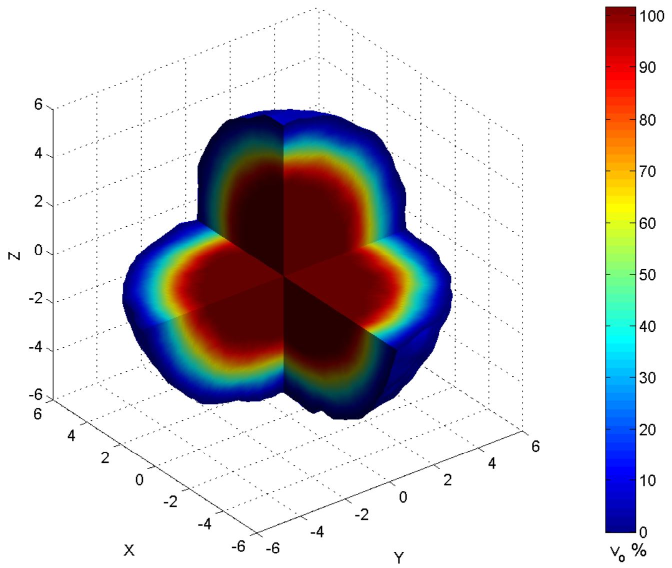


Figure 58: Six-dimensional reachable volume for the min robot from (99), where $V_r = 49\%$ and $V_{6d} = 97\%$.

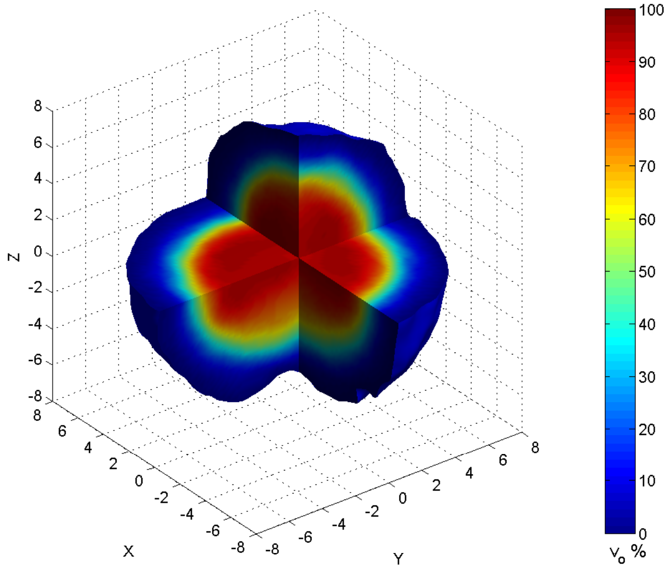


Figure 59: Six-dimensional fault tolerant volume for $\mathcal{K} > 0.2$ for the mid robot from (99) with tracking the maximum $\mathcal{K}$ over all self-motion manifolds, where $V_{FT} = 73\%$ and $V_{FT_{3d}} = 19\%$.

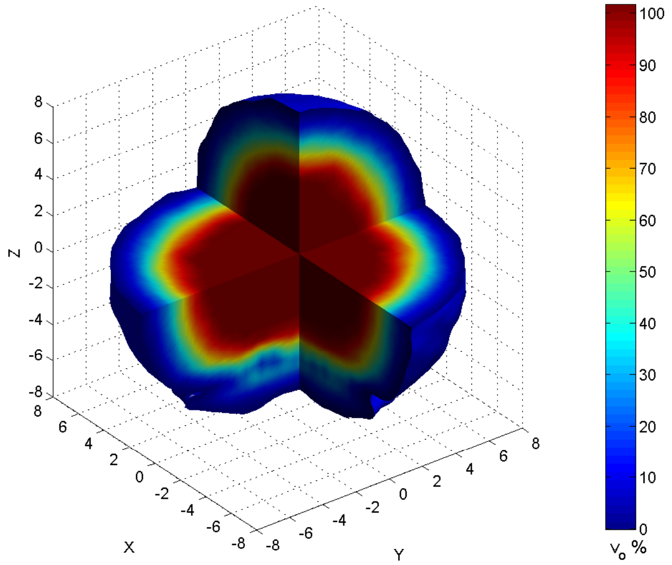


Figure 60: Six-dimensional reachable volume for the mid robot from (99), where $V_r = 41\%$ and $V_{6d} = 95\%$.

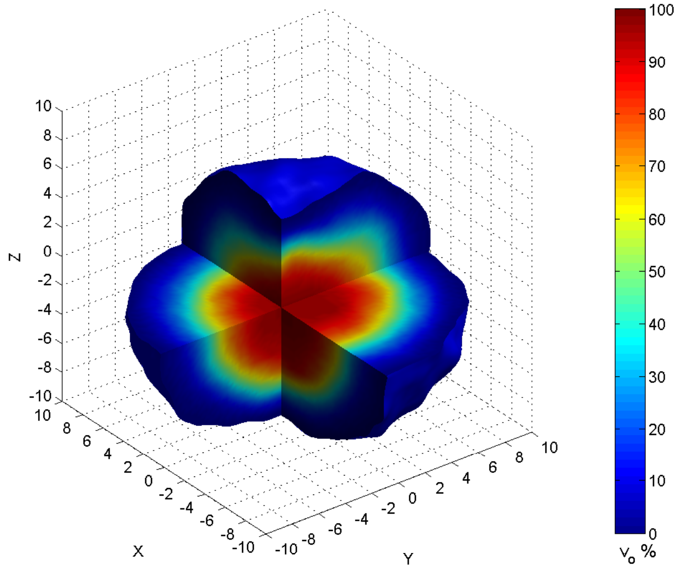


Figure 61: Six-dimensional fault tolerant volume for $\mathcal{K} > 0.2$ for the max robot from (99) with tracking the maximum $\mathcal{K}$ over all self-motion manifolds, where $V_{FT} = 13\%$ and $V_{FT_{3d}} = 59\%$.

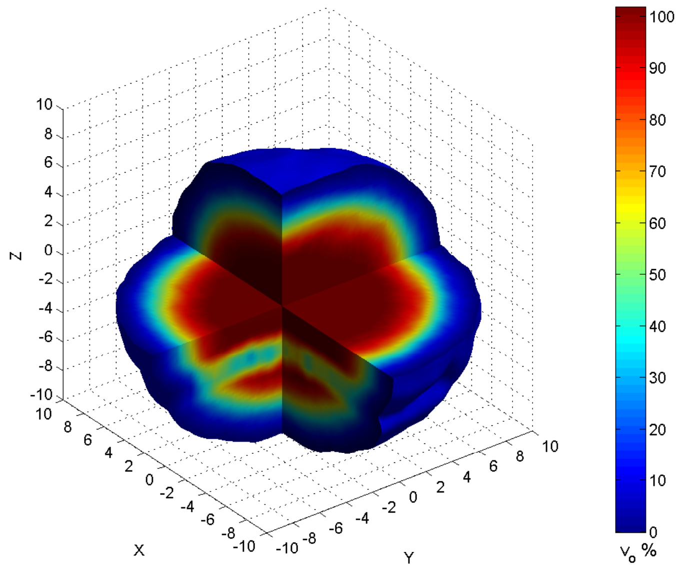


Figure 62: Six-dimensional reachable volume for the max robot from (99), where $V_r = 68\%$ and $V_{6d} = 28\%$.

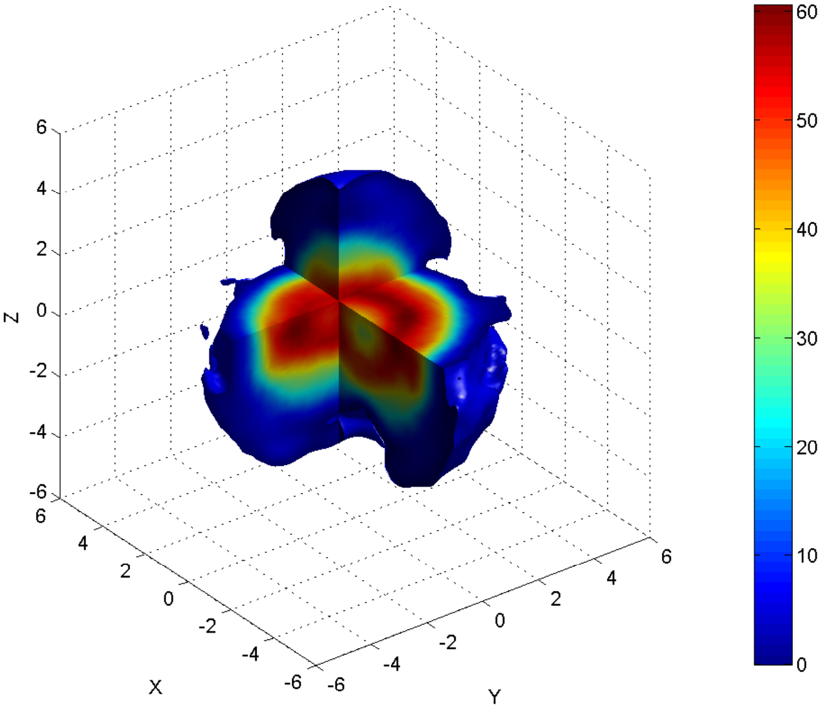


Figure 63: Six-dimensional fault tolerant volume for $\mathcal{K} > 0.2$ for the modified human arm robot with tracking the maximum $\mathcal{K}$ over all self-motion manifolds, where $V_{FT} = 10\%$ and $V_{FT_{3d}} = 78\%$.

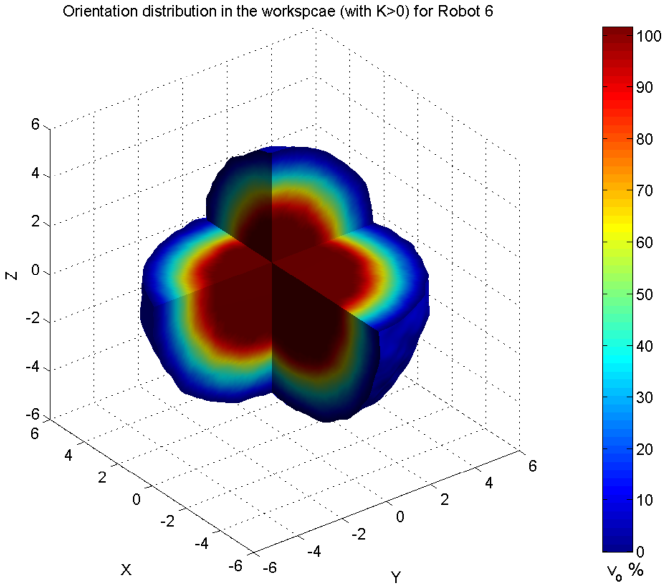

Orientation distribution in the workspcae (with K>0) for Robot 6

Figure 64: Six-dimensional reachable volume for the modified human arm robot, where $V_r = 100\%$ and $V_{6d} = 49\%$.