

DISSERTATION

CHARACTERIZING THE SELF-MOTION MANIFOLDS OF REDUNDANT ROBOTS OF  
ARBITRARY KINEMATIC STRUCTURES

Submitted by

Ahmad A Almarkhi

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Spring 2022

Doctoral Committee:

Advisor: Anthony A Maciejewski

Edwin Chong

Juliana Oprea

Jianguo Zhao

Copyright by Ahmad A Almarkhi 2022

All Rights Reserved

## ABSTRACT

### CHARACTERIZING THE SELF-MOTION MANIFOLDS OF REDUNDANT ROBOTS OF ARBITRARY KINEMATIC STRUCTURES

Robot fault tolerance measures can be classified into two categories: 1) Local measures that are based on the singular value decomposition (SVD) of the robot Jacobian, and 2) Global measures that are suitable to quantify the fault tolerance more effectively in pick-and-place applications. One can use the size of the self-motion manifold of a robot as a global fault-tolerance measure. The size of the self-motion manifold at a certain end-effector location can be simply the sum of the range of the joint angles of a robot at that location. This work employs the fact that the largest self-motion manifolds occur due to merging two (or more) previously disjoint manifolds. The connection of previously disjoint manifolds occur in special configurations in the joint space called *singularities*.

Singularities (singular configurations) occur when two or more of the robot joint axes become aligned and are linearly dependent. A significant amount of research has been performed on identifying the robot singularities but was all based on symbolically solving for when the robot Jacobian is not of full rank. In this work, an algorithm was proposed that is based on the gradient of the singular values of the robot Jacobian. This algorithm is not limited to any Degree of Freedom (DoF) nor any specific robot kinematic structure and any rank of singularity.

Based on the robot singularities, one can search for the largest self-motion manifold near robot singularities. The measure of the size of the self-motion manifold was chosen to eliminate the effect of the self-motion manifold's topology and dimension. Because the SVD at singularities is indistinct, one can employ Givens rotations to define the physically meaningful singular directions, i.e., the directions where the robot is not able to move. This approach has been extensively implemented on a 4-DoF robot, different 7-DoF robot, and an 8-DoF robot.

The global fault-tolerance measure might be further optimized by changing the kinematic structure of a robot. This may allow one to determine a globally fault-tolerant robot, i.e., a robot with  $2\pi$  range for all of its joint angles at certain end-effector location, i.e., a location that is the most suitable for pick-and-place tasks.

## ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude for my advisor, Prof. Anthony A. Maciejewski, and express my deep thanks for his ultimate help and support. This would not have happened without your great, persistent guidance and support. I would like, also, to express my thanks to my Ph.D. committee members - Prof. Edwin Chong, Prof. Iuliana Oprea, and Prof. Jianguo Zhao for the effort they put in my research and the support they provided.

I would like to express my spacial appreciation for Prof. Maciejewski and Prof. Chong who made an exceptional effort to help my late friend, Fateh ElSherif.

I thank the Libyan Ministry of Higher Education for funding my study in partial.

I also would like to acknowledge the ECE department staff, and especially, Katya and Alauna. Thank you both for you great support.

Lastly, I would like to express my highest appreciation to my family, My late Mother who wanted to see me a doctor; To my Father, who was the best friend and supporter in this journey; To my Brothers and my only Sister for their unconditional support. To my beloved wife, Eman, thanks for your support, patience, and positivity all the way.

## DEDICATION

*To my wife, Eman and my kids - Balsam, Yahya, and Elias*

## TABLE OF CONTENTS

ABSTRACT . . . . .	ii
ACKNOWLEDGEMENTS . . . . .	iv
DEDICATION . . . . .	v
LIST OF TABLES . . . . .	viii
LIST OF FIGURES . . . . .	ix
Chapter 1    Introduction . . . . .	1
Chapter 2    TWO . . . . .	3
2.1        INTRODUCTION . . . . .	3
2.2        SINGULARITY ANALYSIS . . . . .	5
2.2.1    Background . . . . .	5
2.2.2    Identifying Robot Singularities . . . . .	5
2.2.3    Identifying Singular Directions . . . . .	12
2.3        CASE STUDIES . . . . .	14
2.3.1    Introduction . . . . .	14
2.3.2    4-DoF Regional Robot . . . . .	14
2.3.3    Mitsubishi PA-10 Robot . . . . .	16
2.4        CONCLUSIONS . . . . .	21
Chapter 3    THREE . . . . .	22
3.1        INTRODUCTION . . . . .	22
3.2        BACKGROUND . . . . .	24
3.2.1    Self-Motion Manifolds . . . . .	24
3.2.2    Size of Self-Motion Manifolds . . . . .	25
3.3        IDENTIFYING LARGER SELF-MOTION MANIFOLDS . . . . .	27
3.4        PA-10 ROBOT CASE STUDY . . . . .	33
3.4.1    PA-10 background . . . . .	33
3.4.2    The largest self-motion manifold . . . . .	35
3.4.3    Evaluation . . . . .	39
3.4.4    Example Use Case . . . . .	41
3.5        CONCLUSIONS . . . . .	43
Chapter 4    FOUR . . . . .	44
4.1        INTRODUCTION . . . . .	44
4.2        BACKGROUND ON SELF-MOTION MANIFOLDS . . . . .	46
4.2.1    Preliminaries . . . . .	46
4.2.2    Size of Self-motion Manifolds . . . . .	50
4.3        GENERATING OPTIMALLY FAULT TOLERANT ROBOT DESIGNS . . . . .	51
4.4        RESULTS . . . . .	55
4.4.1    Four-DoF Robots . . . . .	55

4.4.2	Seven-DoF Robots . . . . .	58
4.4.3	Eight-DoF Robots . . . . .	60
4.4.4	Discussion . . . . .	63
4.5	CONCLUSIONS AND FUTURE WORK . . . . .	64
Chapter 5	Conclusions and Future Works . . . . .	65
5.1	CONCLUSIONS: . . . . .	65
5.2	FUTURE PLAN: . . . . .	66
Bibliography	. . . . .	67

## LIST OF TABLES

2.1	DH parameters of an example 4-DoF manipulator . . . . .	15
	17	
2.2	DH parameters of the PA-10 robot . . . . .	17
2.4	PA-10 robot's rank-2 singular configurations . . . . .	19
2.5	PA-10 robot's rank-3 singular configurations . . . . .	20
3.1	The DH Parameters of the PA-10 Robot . . . . .	34
3.2	Efficiency Comparison between the Proposed Algorithms and the Random Approach .	39
3.3	Robot's Home and Task Configurations . . . . .	42
4.1	DH Parameters of the Baseline 4-DoF Robot . . . . .	55
4.2	The DH Parameters of an Example Optimal 4-DoF Robot . . . . .	56
4.3	The DH Parameters of the PA-10 Robot in Maximal Configurations . . . . .	58
4.4	The DH Parameters of an Example Optimal 7-DoF Robot . . . . .	60
4.5	The DH Parameters of the Baseline 8-DoF Robot . . . . .	61
4.6	The DH Parameters of the Optimal 8-DoF Robot . . . . .	63

## LIST OF FIGURES

2.1	The behavior of two nearly-equal singular values. . . . .	8
2.2	The behavior of three nearly-equal singular values. . . . .	10
2.3	Singular value behavior after applying the proposed algorithm. . . . .	12
2.4	A 7-DoF robot in the same rank-3 singularity. . . . .	13
2.5	The rank-1 singularities of the 4-DoF robot. . . . .	15
2.6	The 4-DoF robot in different rank-1 singularities. . . . .	16
2.7	The PA-10 robot in rank-1 singularities. . . . .	18
2.8	The PA-10 robot in rank-2 singularities. . . . .	19
2.9	The PA-10 robot in rank-3 singularities. . . . .	20
3.1	Sketches for different self-motion manifolds. . . . .	26
3.2	Sketch for rank-1 and rank-2 singularities that occur of previously-disjointed manifolds. . . . .	32
3.3	3-D projections of various PA-10 self-motion manifolds. . . . .	35
3.4	Various 3-D projections of different PA-10 self-motion manifolds characterized by $\theta_4$ . . . . .	37
3.5	The optimal self-motion manifold of the PA-10 robot. . . . .	38
3.6	Comparison between self-motion manifold sizes found by using a random approach and approaches based on the local dexterity measures. . . . .	40
3.7	The PA-10 robot in different configurations. . . . .	41
3.8	A joint-six failure simulation of the PA-10 robot. . . . .	42
4.1	The topology of one-dimensional self-motion manifold. . . . .	49
4.2	The baseline 4-DoF robot and its largest self-motion manifold. . . . .	56
4.3	Kinematically optimal fault-tolerant 4-DoF robot. . . . .	57
4.4	The largest self-motion manifold of the baseline 7-DoF robot. . . . .	59
4.5	Kinematically optimal fault-tolerant 7-DoF robot. . . . .	61
4.6	The baseline 8-DoF robot and its largest self-motion manifold. . . . .	62
4.7	kinematically optimal fault-tolerant 8-DoF robot. . . . .	63

# Chapter 1

## Introduction

The design of fault-tolerant robots has been increasingly important, especially for critical applications. Such critical applications are commonly employed where maintenance is too costly or the robots are inaccessible, e.g., search and rescue [1] and disaster recovery [2]. One typical approach for insuring high-availability, and thus fault tolerance, is redundancy. Redundancy has many categories/levels including, structural redundancy, e.g., duplicating parts that most commonly fail [3], functional redundancy, i.e., a human intervention is required to correct a fault situation, and kinematic redundancy, where a robot is designed with more degrees of freedom (DoFs) than the minimum required for a certain task.

Many aspects of fault tolerance have been extensively studied, for example, how to quantify fault tolerance, resulting in two different categories, i.e., local and global fault tolerance measures. Local fault tolerance measures are generally based on the singular value decomposition of the instantaneous Jacobian of the robot. These local measures include the minimum singular value [4], the condition number [5], and the robot manipulability [6]. On the other hand, global fault-tolerance measures are useful for pick-and-place tasks where there is a need to identify a fault-tolerant location in the robot's workspace. In this study, we suggest the size of the robot's self-motion manifold at certain end-effector location to be the global measure for the robot at that workspace location.

The rest of this dissertation is organized as follows:

Chapter 2 proposes an algorithm to identify the singular configurations of any rank for any kinematic-structure robots. The algorithm employs the gradient of the singular values of the robot's Jacobian. It also, deals with the not-uncommon cases when the singular value values are indistinct. In this chapter, we also suggest a way to define the singular directions.

Chapter 3 uses the algorithm for identifying robot singularities to search for the largest self-motion manifold by employing the fact that the largest self-motion manifolds occur near singu-

larities. In this chapter, we also suggest a measure of the "size" of the self-motion manifold that is based on the sum of the joint angle ranges of a robot. This measure quantifies the global fault tolerance of a robot regardless of the topology of that self-motion manifold and the number and the rank of the singularities that occur on the self-motion manifold.

Chapter 4 presents the conclusions of this work and a direction for the future works.

## Chapter 2

# Singularity Analysis for Redundant Manipulators of Arbitrary Kinematic Structure<sup>1</sup>

### 2.1 INTRODUCTION

A robot singular configuration is a configuration in which the robot's end effector loses the ability to move in one (or more) direction(s), i.e., singular direction(s). Such singular configurations are usually called *singularities* [8]. Robot singularities are also called *critical points* [9] or *special configurations* [10]. At a singularity, there is no joint velocity that can result in an end-effector velocity in a singular direction(s). Singularities result from having the corresponding Jacobian ( $\mathbf{J}$ ) columns be linearly dependent. The singular value decomposition (SVD) of  $\mathbf{J}$  can reveal immediate information about singularities. At a singularity, one (or more) singular value(s) of the robot Jacobian are zero. Robot singularities can offer mechanical advantages [11], however, they require more sophisticated inverse kinematics solutions [12].

Identifying robot singularities has been extensively studied. For non-redundant manipulators, where  $\mathbf{J}$  is square, singularities can be found by symbolically solving for conditions when the determinant of  $\mathbf{J}$  equals zero ( $|\mathbf{J}| = 0$ ) [13]. For redundant manipulators, where  $|\mathbf{J}|$  does not exist, the conditions that make  $|\mathbf{J}\mathbf{J}^\top| = 0$  can be computed, but this is usually difficult to solve. In this case, one viable approach is to solve for conditions that make all the  $6 \times 6$  sub-Jacobians singular, i.e., the determinants of all sub-Jacobians equal zero [14], but this also becomes infeasible for robots with a large number of degrees of freedom (DoF). For example, an 8-DoF manipulator requires computing the determinants of 28 sub-Jacobians. In addition, these techniques typically lack the ability to provide information about the singular vector(s) associated with a singularity.

---

<sup>1</sup>This chapter was published in [7]

To more easily identify singularities and find singular directions, [15] suggested using the fact that at a singularity, there must be a screw reciprocal to all screws that represent the columns of the robot Jacobian. This technique has been used to identify the rank-1 singularity conditions and the singular directions for 7-DoF manipulators [16]. The reciprocity-based methodology has also been used to find the rank-1 singularities of an 8-DoF manipulator [17]. In addition, it was extended to identify the rank-2 singularities of a 7-DoF manipulator [18]. This technique shows its merit of being relatively easy and extendable, but it is highly dependent on selecting a reference frame that simplifies the computation of  $\mathbf{J}$ . Building on the reciprocity-based approach, researchers have suggested further simplifications of the Jacobian by performing elementary transformations on the Jacobian before solving for the singularity conditions as in [19]. This approach has been further employed in performing singularity avoidance for manipulators with non-spherical wrists [20]. All these techniques work well for simple classes of kinematically redundant manipulators and for rank-1 singularities. Such manipulators have their successive joint axes either perpendicular or parallel, which makes computing  $\mathbf{J}$  relatively easy.

In this paper we suggest a technique to find the singularities of a manipulator with an arbitrary degree of redundancy and arbitrary kinematic structure. This can be achieved by driving a certain singular value,  $\sigma_i$ , of  $\mathbf{J}$  to zero by following the gradient descent of that singular value, i.e.,  $-\nabla\sigma_i$ . The complexity of this technique is independent of the rank of the singularity. In addition, we present an algorithm to identify the singular directions at high-rank singularities.

The rest of this paper is organized as follows. Methodologies to identify robot singularities and their corresponding singular directions are presented in section 2.2. In section 2.3, the results of applying the methodologies to a 4-DoF robot and a 7-DoF robot are discussed. Finally, the conclusions of this work are presented in section 2.4.

## 2.2 SINGULARITY ANALYSIS

### 2.2.1 Background

The forward kinematics of an  $n$ -DoF robot that is acting in an  $m$ -dimensional workspace can be written as

$$\dot{\boldsymbol{x}} = \boldsymbol{J}\dot{\boldsymbol{\theta}} \quad (2.1)$$

where  $\dot{\boldsymbol{x}}$  is an  $m \times 1$  vector representing the end-effector velocity,  $\boldsymbol{J}$  is the  $m \times n$  robot Jacobian, and  $\dot{\boldsymbol{\theta}}$  is an  $n \times 1$  vector that represents the joint angle rates. For redundant robots,  $n > m$ , where  $n - m$  is the degree of redundancy. For a redundant manipulator,  $\boldsymbol{J}$  is not a square matrix, and thus not invertible, however, an inverse kinematics solution can be found using

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}^+ \dot{\boldsymbol{x}} + \boldsymbol{n}_J \quad (2.2)$$

where  $\boldsymbol{J}^+$  is the pseudoinverse of  $\boldsymbol{J}$  and  $\boldsymbol{n}_J$  is an arbitrary vector in the null space of the Jacobian.

The singular value decomposition of  $\boldsymbol{J}$  can be represented as

$$\boldsymbol{J} = \sum_{i=1}^r \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top \quad (2.3)$$

where  $r$  is the rank of  $\boldsymbol{J}$ , the  $\sigma_i$ 's are the ordered singular values, i.e.,  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_m \geq 0$ , the unit vectors  $\boldsymbol{u}_i$  represent the output singular vectors, and  $\boldsymbol{v}_i$  are the input singular vectors. For a robot at a rank- $n$  singularity, there are  $n$  singular values,  $\sigma_i$ 's, that become zero. Thus, employing a technique that minimizes singular values of  $\boldsymbol{J}$  can be used to identify robot singular configurations.

### 2.2.2 Identifying Robot Singularities

In this section, we explain how to employ the gradient descent of a singular value of  $\boldsymbol{J}$  to drive a robot of an arbitrary kinematic structure to a singularity. This technique is not limited by the rank of the singularity. The singular value  $\sigma_i$  in (4.4) can be expressed as

$$\sigma_i = \mathbf{u}_i^\top \mathbf{J} \mathbf{v}_i. \quad (2.4)$$

Differentiating (4.9) with respect to time results in

$$\dot{\sigma}_i = \dot{\mathbf{u}}_i^\top \mathbf{J} \mathbf{v}_i + \mathbf{u}_i^\top \dot{\mathbf{J}} \mathbf{v}_i + \mathbf{u}_i^\top \mathbf{J} \dot{\mathbf{v}}_i. \quad (2.5)$$

One can note that  $\mathbf{u}_i^\top \mathbf{u}_j$  and  $\mathbf{v}_i^\top \mathbf{v}_j$  are zero for  $i \neq j$  and that the derivative of a unit vector is orthogonal to that vector. So, (4.10) can be further simplified to [21]

$$\dot{\sigma}_i = \mathbf{u}_i^\top \dot{\mathbf{J}} \mathbf{v}_i. \quad (2.6)$$

The partial derivative of  $\sigma_i$  with respect to some  $\theta_k$  can be expressed as

$$\frac{\partial \sigma_i}{\partial \theta_k} = \mathbf{u}_i^\top \frac{\partial \mathbf{J}}{\partial \theta_k} \mathbf{v}_i \quad (2.7)$$

where

$$\frac{\partial \mathbf{J}}{\partial \theta_k} = \left[ \frac{\partial \mathbf{j}_1}{\partial \theta_k}, \frac{\partial \mathbf{j}_2}{\partial \theta_k}, \dots, \frac{\partial \mathbf{j}_n}{\partial \theta_k} \right]. \quad (2.8)$$

The partial derivative of the  $i^{\text{th}}$  column of the Jacobian is given by [22], [23]

$$\frac{\partial \mathbf{j}_i}{\partial \theta_k} = \begin{cases} \begin{bmatrix} (\mathbf{z}_k^\top \mathbf{p}_i) \mathbf{z}_i - (\mathbf{z}_k^\top \mathbf{z}_i) \mathbf{p}_k \\ \mathbf{z}_k \times \mathbf{z}_i \end{bmatrix}, & k < i \\ \begin{bmatrix} (\mathbf{z}_i^\top \mathbf{p}_k) \mathbf{z}_k - (\mathbf{z}_k^\top \mathbf{z}_i) \mathbf{p}_k \\ \mathbf{0} \end{bmatrix}, & k \geq i \end{cases} \quad (2.9)$$

Then, the gradient of  $\sigma_i$  for any  $\mathbf{J}$  can be simply computed from (4.12), (4.13), and (4.14), as

$$\nabla \sigma_i = \left[ \frac{\partial \sigma_i}{\partial \theta_1}, \frac{\partial \sigma_i}{\partial \theta_2}, \dots, \frac{\partial \sigma_i}{\partial \theta_n} \right]. \quad (2.10)$$

Now that one can compute  $\nabla\sigma_i$ , it is possible to employ the gradient descent technique to locate a minima for any singular value  $\sigma_i$ . In the following, we explain an algorithm to find rank-1 and higher rank singularities.

### Identifying Rank-1 Singularities

For rank-1 singularities, one can employ the general equation

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \alpha_k \nabla\sigma_i^{(k)} \quad (2.11)$$

where,  $\boldsymbol{\theta}^{(k+1)}$  is a vector that represents the new joint angles of a robot, the vector  $\boldsymbol{\theta}^{(k)}$  is the current joint angles,  $\alpha_k$  is an adaptive step size, and  $\nabla\sigma_i^{(k)}$  is the gradient of  $\sigma_i$  ( $\sigma_i = \sigma_m$  for rank-1 singularities). In order to identify all rank-1 robot singular configurations, one can start by generating random configurations that span the robot joint space. Then, from each random configuration, one can move the robot along the gradient descent of  $\sigma_m$  as in (2.11). For faster convergence to a singularity, one can use the steepest descent method, in which  $\alpha_k$  needs to be adaptive, i.e., it is chosen at each iteration to achieve a maximum decrease in  $\sigma_m$ . This can be done by conducting a one-dimensional search along the  $-\nabla\sigma_m^{(k)}$  direction until a minimizer,  $\boldsymbol{\theta}^{(k+1)}$ , is found.

### Identifying Rank-2 Singularities

A robot is said to be in a rank-2 singularity if  $\epsilon > \sigma_{m-1} \geq \sigma_m$ , where  $\epsilon$  is a small threshold (virtually zero). To identify rank-2 singularities, one can start with a population of random joint configurations and employ (2.11) by moving along the  $-\nabla\sigma_{m-1}^{(k)}$  direction until the  $\sigma_{m-1} < \epsilon$  condition is satisfied. However, it is not uncommon for an undesirable behavior to occur, that results from having the two singular values  $\sigma_{m-1}$  and  $\sigma_m$  become nearly equal before they reach zero, i.e.,  $\sigma_{m-1} \approx \sigma_m > \epsilon$ . In this case, the two singular values are not distinct, which means that their corresponding singular vectors are ill-defined. In other words, any singular vectors ( $\mathbf{u}$  and  $\mathbf{v}$ ) in the  $\{\mathbf{u}_{m-1}, \mathbf{u}_m\}$  and  $\{\mathbf{v}_{m-1}, \mathbf{v}_m\}$  subspaces are valid for solving (4.11). Figure 2.1, shows the

behavior of the algorithm when the two smaller singular values become nearly equal through the process of driving a robot into a rank-2 singularity. In this case,  $\sigma_5 \approx \sigma_6$  (at around iteration 600), which makes them indistinct and their corresponding singular vectors ill-defined. The direction of  $\nabla\sigma_5$  can completely change direction from one iteration to another, which affects the rate of convergence.



**Figure 2.1:** In subplot (a), the evolution of  $\sigma_5$  and  $\sigma_6$  is shown as the standard gradient descent algorithm is employed. The singular value  $\sigma_5$  is minimized until  $\sigma_5 \approx \sigma_6$  at around iteration 600. When they become nearly equal, the angle between the gradients in successive iterations becomes large. These angles are plotted in (b), where the change in the angles reaches  $180^\circ$ . It is clear that the convergence requires a long time (about 3000 iterations) due to the large change in the gradient direction. In this case, the convergence time is approximately 40 seconds. The threshold,  $\epsilon = 10^{-6}$ , is indicated with a red horizontal line.

To overcome this unwanted effect, one can start with moving the robot along the  $-\nabla\sigma_{m-1}^{(k)}$  direction until  $\sigma_5$  and  $\sigma_6$  become very close in value. Then, a combination between the two gradients is computed. Because the singular value decomposition is not unique in these cases, any singular vectors  $\mathbf{u}$  and  $\mathbf{v}$  in the subspace associated with the equal singular values are valid. One can rotate the singular subspace such that the angle between  $\nabla\sigma_5^{(k)}$  and  $\nabla\sigma_6^{(k)}$  is minimized, i.e.,

$$\begin{aligned}
\mathbf{u}_{5(new)} &= \mathbf{u}_5 \cos \phi + \mathbf{u}_6 \sin \phi \\
\mathbf{u}_{6(new)} &= \mathbf{u}_6 \cos \phi - \mathbf{u}_5 \sin \phi \\
\mathbf{v}_{5(new)} &= \mathbf{v}_5 \cos \phi + \mathbf{v}_6 \sin \phi \\
\mathbf{v}_{6(new)} &= \mathbf{v}_6 \cos \phi - \mathbf{v}_5 \sin \phi
\end{aligned} \tag{2.12}$$

where  $\phi$  is the angle of rotation. It should be noted that the angle between the gradients of  $\sigma_5$  and  $\sigma_6$  can vary from 0 to  $\pi$  based on the angle of rotation  $\phi$ . A suitable selection of the rotation angle for the singular subspaces is crucial in minimizing the change in the gradient direction from one iteration to another. Once the  $\nabla\sigma_5^{(k)}$  and  $\nabla\sigma_6^{(k)}$  that have the minimum angle between them are computed, a combination that minimizes  $\sigma_5$  can be found

$$\nabla\sigma^{(k)} = \gamma\nabla\sigma_5^{(k)} + (1 - \gamma)\nabla\sigma_6^{(k)} \tag{2.13}$$

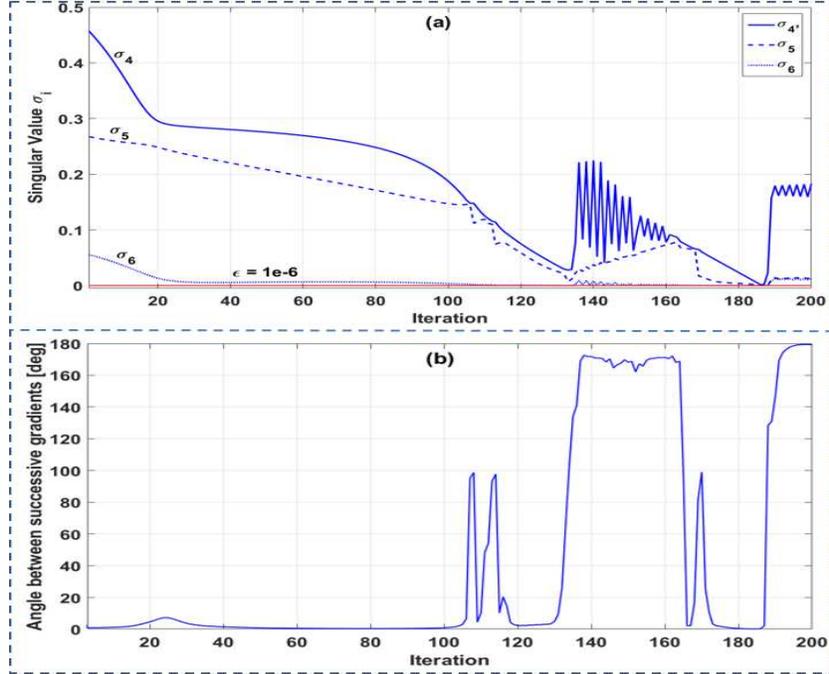
where  $\nabla\sigma^{(k)}$  is the desired gradient and  $\gamma$  is a positive scalar where  $0 \leq \gamma \leq 1$ . This linear search will minimize the change in the gradient direction from one iteration to another. After  $\nabla\sigma^{(k)}$  is computed, the steepest descent method is applied to find an optimal value of  $\alpha_k$  in (2.11) that minimizes  $\sigma_5$ . An analogous process can be employed for identifying higher rank singularities.

### Identifying High-rank Singularities

To identify high-rank singularities, i.e., where three or more singular values become zero, one can employ a similar approach to that applied for identifying rank-2 singularities. For a robot in a singular configuration,  $\mathbf{J}$  is of rank  $r$  if  $\sigma_i = 0$  for  $i > r$ , which also means the robot is in a rank- $(m - r)$  singularity. To find high-rank singularities, one can move the robot by iteratively solving (2.11) until a desired  $\sigma_i$  reaches zero. While moving along the  $-\nabla\sigma_i$  direction, it is possible that  $\sigma_i$  and  $\sigma_{i+1}$  become nearly equal. In this case, the procedure in the previous section can be applied.

In some cases, more than two singular values become nearly equal but larger than the threshold, i.e.,  $\sigma_i \approx \sigma_{i+1} \approx \dots \approx \sigma_m > \epsilon$ . For the purpose of illustration, we will consider the case where a robot is being driven to a rank-3 singularity when the situation  $\sigma_4 \approx \sigma_5 \approx \sigma_6 > \epsilon$  occurs, as

illustrated in Figure 2.2. One can note that around iteration 186 in Figure 2.2 all three singular values became very close in value.



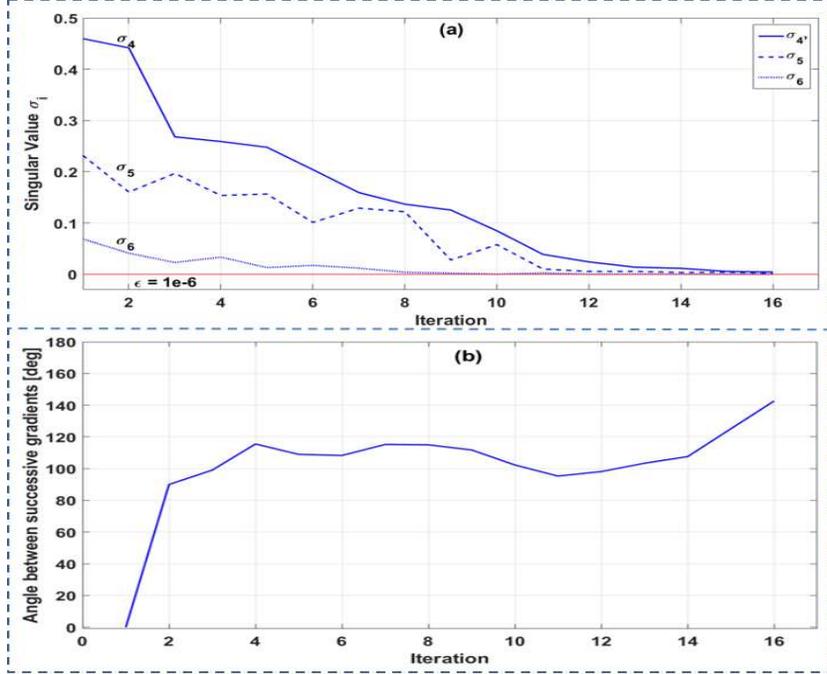
**Figure 2.2:** In (a), the singular value  $\sigma_4$  is minimized until iteration 106, where ( $\sigma_4 \approx \sigma_5$ ). At iterations 135 and 186 the singular values  $\sigma_4$ ,  $\sigma_5$ , and  $\sigma_6$  become nearly equal. Subfigure (b) shows the change in the angle between the gradients in successive iterations. The change of angle reaches  $170^\circ$  when the three singular values become nearly equal. In this case, the singular values  $\sigma_4$ ,  $\sigma_5$ , and  $\sigma_6$  never converge to zero. The threshold,  $\epsilon = 10^{-6}$ , is indicated with a red horizontal line.

At this point, the angle between the gradient of  $\sigma_4$  in successive iterations (the angles between  $\nabla\sigma_4^{(k-1)}$  and  $\nabla\sigma_4^{(k)}$ ) became  $170^\circ$ , which resulted from having the singular values indistinct and their corresponding singular vectors ill-defined. This also contributed to an unwanted increase in  $\sigma_4$  because the gradients switched direction. In this case, one can rotate their corresponding singular subspaces to find a suitable rotation that minimizes the sum of the angles between the three gradients. The solution is to minimize an objective function  $H$ , where

$$H = \sum_{(i=r+1)}^m \sum_{(j=i+1)}^m \theta_{i,j} \quad (2.14)$$

and  $\theta_{i,j}$  is the angle between  $\nabla\sigma_i$  and  $\nabla\sigma_j$ . The rotation of the singular subspaces can be done by iteratively employing (2.12). That is, because the singular subspaces are three-dimensional (or higher), one can iteratively rotate one plane at a time, i.e., in this case,  $\{\mathbf{u}_4, \mathbf{u}_5\}$  and  $\{\mathbf{v}_4, \mathbf{v}_5\}$ , then  $\{\mathbf{u}_5, \mathbf{u}_6\}$  and  $\{\mathbf{v}_5, \mathbf{v}_6\}$ , and so on. This iterative rotation should be done until the sum of the angles between all gradients is minimized. After finding the gradients of the singular values, one can use (2.13) to compute a combination between the first two gradients,  $\nabla\sigma_4$  and  $\nabla\sigma_5$ , that minimize  $\sigma_4$ . Then, using (2.13) again to compute a combination between the resulting gradient and  $\nabla\sigma_6$  that minimizes  $\sigma_4$ . This approach guarantees achieving a minimum amount of gradient direction change and thus a shorter convergence time.

This process can continue until the algorithm cannot converge to any higher rank singularities. This algorithm, along with an adaptive step size, was applied to the same robot that resulted in Figure 2.2 and the results are shown in Figure 2.3. It is clear that the convergence is faster and the change in the gradient angle is smaller. The average convergence time was improved from 40 seconds to less than 2 seconds when the proposed technique is employed.



**Figure 2.3:** This figure shows the behavior of the singular values when the proposed algorithm is applied to a robot to find a rank-3 singularity. This is the same robot as shown in Figure 2.2. Subfigure (a) shows the values of the three smaller singular values,  $\sigma_4$ ,  $\sigma_5$ , and  $\sigma_6$ , while applying the algorithm with an adaptive step size  $\alpha_k$ . In subfigure (b), the angles between the gradients in successive iterations are shown. The angles average around  $120^\circ$ . The algorithm converges in 16 iterations. The convergence time in this case is less than two seconds. The threshold,  $\epsilon = 10^{-6}$ , is indicated with a red horizontal line.

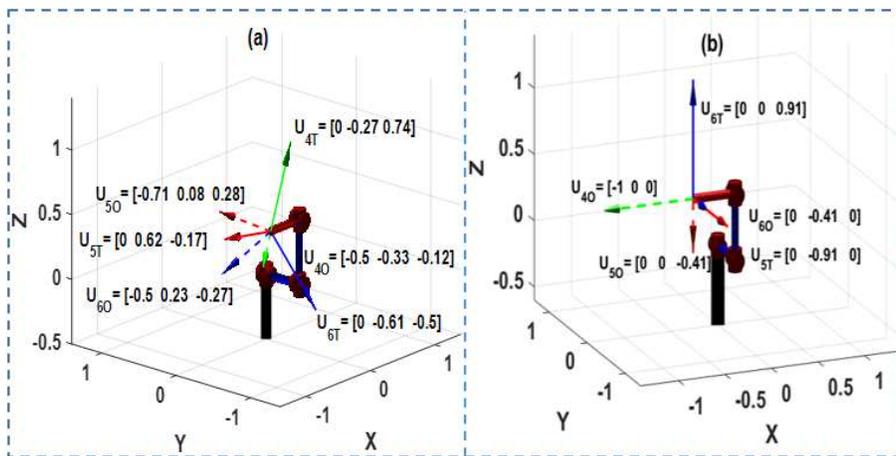
If one applies the above algorithm to an initial population of random configurations then one can identify all the singular configurations of various ranks. It is then possible to analyze the resulting singularities to determine the singularity conditions for the robot. Some singularity conditions depend on the values of a few joints while other joints can take any value. One may observe that a singularity can be satisfied by an infinite number of joint configurations. In the next section, we discuss a mathematical approach to identify singular directions associated with the singularities that are physically meaningful.

### 2.2.3 Identifying Singular Directions

For a robot Jacobian  $\mathbf{J}$  with rank  $r$ , the last  $m - r$  output singular vectors, i.e.,  $\mathbf{u}_i$ 's, span the directions of lost end-effector motion. For spatial manipulators, these singular vectors are

6-dimensional and represent a simultaneous translational and rotational velocity. For rank-1 singularities, there is only one unique singular direction,  $\mathbf{u}_m$ , and it is easy to visualize. At higher rank singularities, the singular value decomposition is not unique. Thus, the singular vectors corresponding to the zero singular values ( $\sigma_{r+1}, \sigma_{r+2}, \dots, \sigma_m$ ) are ill defined and will likely not be well aligned with the world (or task) frame of the robot. However, one can apply Givens rotations to these vectors in order to identify an intuitive representation for the lost end effector motion.

Consider Figure 2.4, that shows a 7-DoF robot in a rank-3 singularity, where both subfigures correspond to the same robot in the same singular configuration.



**Figure 2.4:** A 7-DoF robot in the same rank-3 singularity is presented.<sup>2</sup>In subfigure (a), the three SVD-generated singular directions are indicated. In subfigure (b), the singular directions are plotted after they are properly rotated. The singular directions,  $\mathbf{u}_4$ ,  $\mathbf{u}_5$ , and  $\mathbf{u}_6$  are represented by green, red, and blue respectively. Dotted arrows represent rotational velocity and solid arrows represent translational velocity.

The original singular vectors, identified by employing the singular value decomposition, can be rotated to a more intuitive set that is aligned with the world coordinate frame as follows:

<sup>2</sup>Figure 4, 6, 7, 8, and 9 are produced using the Robotics Toolbox [24].

$$[\mathbf{u}_4 \mathbf{u}_5 \mathbf{u}_6] = \begin{bmatrix} 0 & 0 & 0 \\ -0.27 & 0.62 & -0.61 \\ 0.74 & -0.17 & -0.50 \\ -0.50 & -0.71 & -0.50 \\ -0.33 & 0.08 & 0.23 \\ -0.12 & 0.28 & -0.27 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 0 \\ 0 & -0.91 & 0 \\ 0 & 0 & 0.91 \\ -1 & 0 & 0 \\ 0 & 0 & -0.41 \\ 0 & -0.41 & 0 \end{bmatrix}$$

Using the rotated subspace above (shown in Figure 2.4(b)), one can easily determine that there is no joint velocity that can generate any rotational velocity around the  $-X$  direction (green). In addition, the robot cannot have a simultaneous velocity with the illustrated components of  $-Y$  translational motion with  $-Z$  rotational motion (red). Likewise, there are no joint rates that can generate a simultaneous velocity with the illustrated components of  $Y$  translational motion with  $-Z$  rotational motion (blue). In the next section, we illustrate the results of applying the proposed algorithms to robots of different kinematic structures.

## 2.3 CASE STUDIES

### 2.3.1 Introduction

Our algorithm to identify robot singularities is neither limited by the kinematic structure of the robot, nor by the rank of the singularity. We employed it on several 4-DoF regional and 7-DoF spatial robots and present the results of one 4-DoF and one 7-DoF robot here. For each robot we start with 10,000 random configurations in the joint space. From each point, we apply the gradient-based algorithm to find rank-1 and all higher rank singularities. The resulting singular vectors are then rotated to the most intuitive representation of the lost directions of motion.

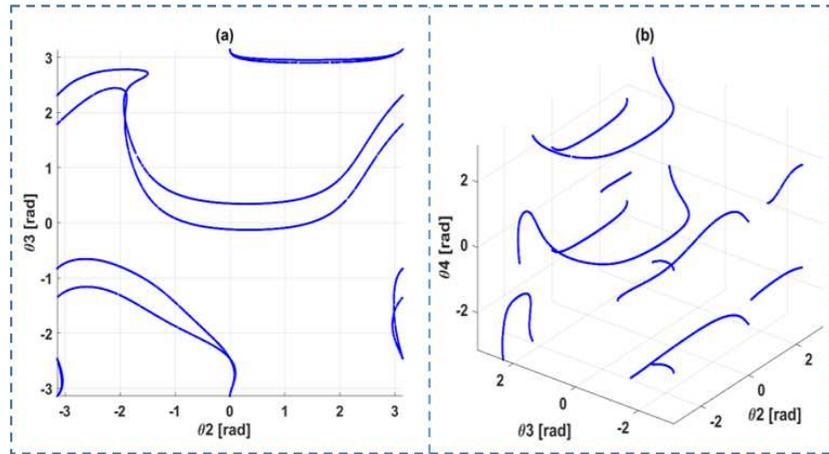
### 2.3.2 4-DoF Regional Robot

We study a locally optimal fault-tolerant 4-DoF robot presented in [25] with the DH parameters given in Table 4.1. This robot is a spatial positioning (regional) manipulator, i.e., it has a 3-dimensional workspace. Our focus is on identifying the singular configurations of this manipulator.

**Table 2.1:** DH parameters of an example 4-DoF manipulator

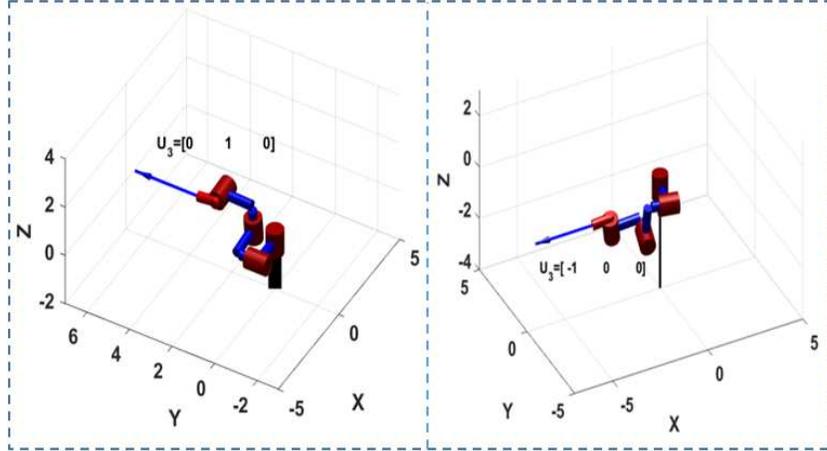
Link <sub><i>i</i></sub>	$\alpha_i$ [radians]	$a_i$ [m]	$d_i$ [m]	$\theta_i$ [radians]
1	$\pi/2$	$\sqrt{2}$	0	$\theta_1$
2	$-\pi/2$	$\sqrt{2}$	1	$\theta_2$
3	$\pi/2$	$\sqrt{2}$	-1	$\theta_3$
4	0	$\sqrt{3/2}$	1/2	$\theta_4$

It was found that this robot has only rank-1 singularities as presented in Figure 2.5. These singularities are arranged along continuous manifolds in the joint space. This means that the robot can continuously move while staying in a singular configuration. Note that these are not the same as the self-motion manifolds.



**Figure 2.5:** This figure illustrates the 4-DoF robot rank-1 singularities. In (a) the singular configurations are shown using a 2-D projection in  $[\theta_2, \theta_3]$ . In (b) a 3-D projection in  $[\theta_2, \theta_3, \theta_4]$  is shown.

Because the robot does not have any high-rank singularities, the presentation of the singular directions is straightforward. The singular direction,  $\mathbf{u}_3$ , gives the actual direction of the loss of the end-effector velocity. Figure 2.6 shows the 4-DoF robot in two different rank-1 singularities along with the singular directions with respect to the world frame.



**Figure 2.6:** This figure shows the 4-DoF robot in two different rank-1 singular configurations. In (a) the robot is shown in the configuration  $\theta = [-3.11, 3.02, -1.53, 0.15]$  rad. The singular direction corresponding to the singularity is  $\mathbf{u}_3 = [0, 1, 0]^T$ , which indicates that the singular direction is aligned with the  $Y$  direction. In (b) the robot is shown in the configuration  $\theta = [-1.76, -1.85, -1.39, 0.02]$  rad. The singular direction is  $\mathbf{u}_3 = [-1, 0, 0]^T$ , which indicates that the singular direction is aligned with the  $-X$  direction.

### 2.3.3 Mitsubishi PA-10 Robot

#### Introduction

The Mitsubishi PA-10 is a 7-DoF manipulator with a 6-dimensional work space. Its kinematic structure is similar to the human arm with three spherical joints in the shoulder, one joint in the elbow, and three spherical joints in the wrist. The DH parameters of the PA-10 are listed in Table 4.3.

**Table 2.3:** PA-10 robot's rank-1 singular configurations<sup>3</sup>

i	$\theta_1$	$\theta_2$	$\theta_3$	$\theta_4$	$\theta_5$	$\theta_6$	$\theta_7$
1	$x$	$\pm\pi, 0$	$\pm\pi/2$	$x$	$x$	$x$	$x$
2	$x$	$\pm\pi, 0$	$x$	$x$	$x$	$\pm\pi, 0$	$x$
3	$x$	$x$	$x$	$\pm\pi, 0$	$x$	$x$	$x$
4	$x$	$x$	$x$	$x$	$\pm\pi/2$	$\pm\pi, 0$	$x$

**Table 2.2:** DH parameters of the PA-10 robot

Link <sub><i>i</i></sub>	$\alpha_i$ [radians]	$a_i$ [m]	$d_i$ [m]	$\theta_i$ [radians]
1	$-\pi/2$	0	0	$\theta_1$
2	$\pi/2$	0	0	$\theta_2$
3	$-\pi/2$	0	0.45	$\theta_3$
4	$\pi/2$	0	0	$\theta_4$
5	$-\pi/2$	0	0	$\theta_5$
6	$\pi/2$	0	0.45	$\theta_6$
7	0	0	0	$\theta_7$

The singularity analysis on the PA-10 resulted in rank-1, rank-2, and rank-3 singularities being identified. We were able to find singularity conditions for each singularity rank. The singular directions,  $\mathbf{u}_i$ 's, were also identified and appropriately rotated. In all figures, loss of directional velocity is indicated with a solid arrow, while the loss of rotational velocity is indicated with a dotted arrow. Blue arrows indicate  $\mathbf{u}_6$ , red arrows indicate  $\mathbf{u}_5$ , and green arrows indicate  $\mathbf{u}_4$ .

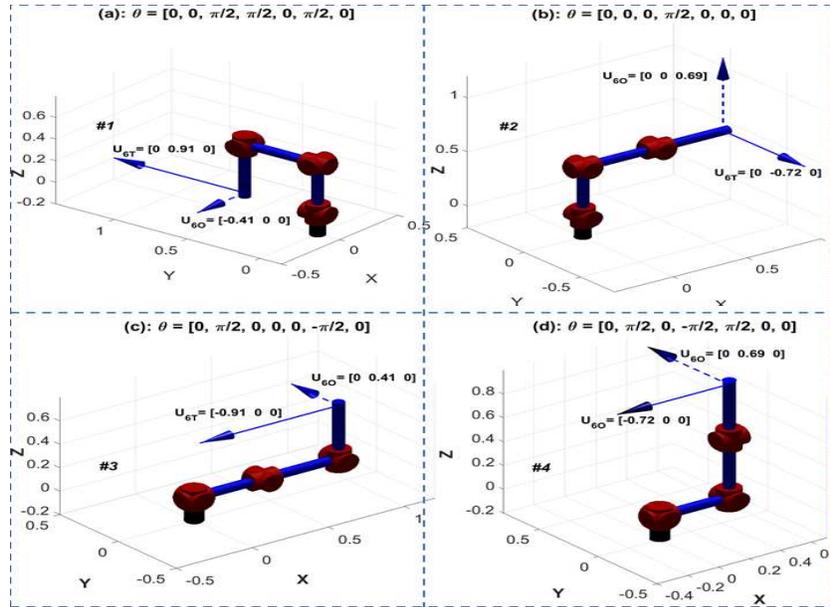
### Rank-1 Singularities

All rank-1 singularities are summarized in Table 2.3. Joint 4 is critical in that the robot will be singular if  $\theta_4$  is equal to  $\pm\pi$  or 0. One can observe that joint 4 is the only joint that can change the distance between the shoulder and the wrist.

---

<sup>3</sup>In all tables, “ $x$ ” means the angle value does not matter.

The robot singular directions that indicate the loss of the end-effector velocity are all shown in Figure 2.7. Because these are rank-1 singularities, their corresponding singular directions are well defined.



**Figure 2.7:** The PA-10 robot is shown in rank-1 singular configurations. The singular direction,  $u_6$ , is also plotted for each singularity. The singularity conditions, 1, 2, 3, and 4, in Table 2.3 are satisfied in subfigures (a), (b), (c), and (d), respectively.

## Rank-2 Singularities

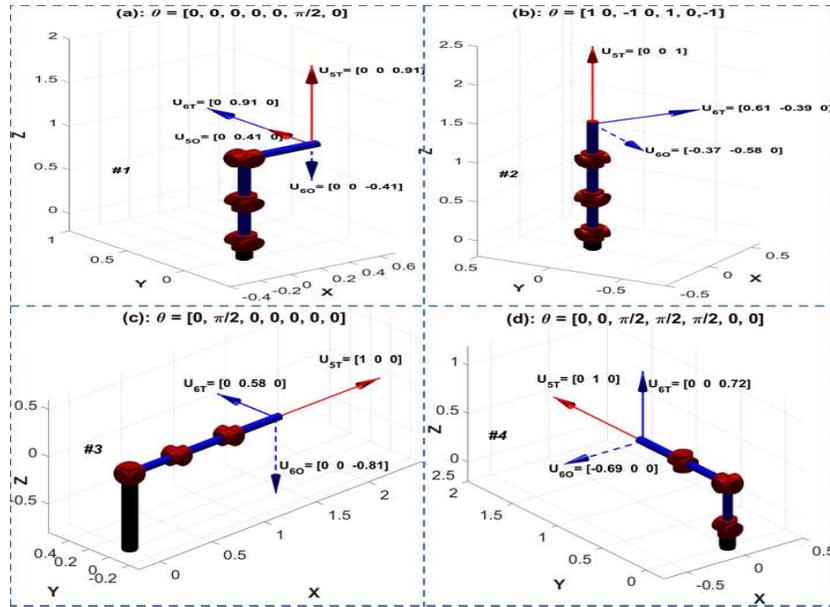
The PA-10's rank-2 singularity conditions are shown in Table 2.4. The common feature between these conditions is that they do not depend on the value of  $\theta_1$  or  $\theta_7$ .

We employed Givens rotation to make sure that the two singular vectors,  $u_5$  and  $u_6$ , are rotated to represent the most intuitive set of singular directions.

**Table 2.4:** PA-10 robot's rank-2 singular configurations

i	$\theta_1$	$\theta_2$	$\theta_3$	$\theta_4$	$\theta_5$	$\theta_6$	$\theta_7$
1	$x$	$\pm\pi, 0$	$\pm\pi, 0$	$\pm\pi, 0$	$x$	$x$	$x$
2	$x$	$\pm\pi, 0$	$x$	$\pm\pi, 0$	$x$	$\pm\pi, 0$	$x$
3	$x$	$x$	$x$	$\pm\pi, 0$	$\pm\pi, 0$	$\pm\pi, 0$	$x$
4	$x$	$\pm\pi, 0$	$\pm\pi/2$	$x$	$\pm\pi/2$	$\pm\pi, 0$	$x$

Figure 2.8 shows the four singular conditions listed in Table 2.4.



**Figure 2.8:** The PA-10 robot is shown in rank-2 singular configurations. The singular directions  $u_5$  and  $u_6$  are plotted for each singularity condition. The singularity conditions, 1, 2, 3, and 4, in Table 2.4 are represented in the subfigures (a), (b), (c), and (d), respectively.

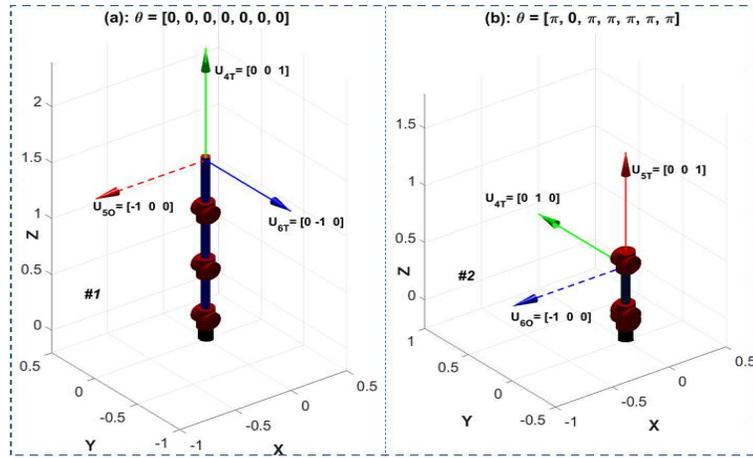
### Rank-3 Singularities

It was found that the PA-10 robot can have rank-3 singularities by aligning the axes of joints 1, 3, 5, and 7 so that their columns of the Jacobian are linearly dependent. The conditions for these rank-3 singularity configurations are listed in Table 2.5.

**Table 2.5:** PA-10 robot's rank-3 singular configurations

$i$	$\theta_1$	$\theta_2$	$\theta_3$	$\theta_4$	$\theta_5$	$\theta_6$	$\theta_7$
1	$x$	$\pm\pi, 0$	$x$				

As before, we have used Givens rotations to make the singular vectors,  $\mathbf{u}_4$ ,  $\mathbf{u}_5$ ,  $\mathbf{u}_6$  as intuitive as possible. Figure 2.9 shows the manipulator in two different rank-3 singular configurations.



**Figure 2.9:** The PA-10 robot is shown in rank-3 singular configurations. The singular directions,  $\mathbf{u}_4$ ,  $\mathbf{u}_5$ , and  $\mathbf{u}_6$  are indicated in green, red, and blue respectively. In (a), the robot is in configuration  $\theta = [0, 0, 0, 0, 0, 0, 0]$  and in (b) the robot is in  $\theta = [\pi, 0, \pi, \pi, \pi, \pi, \pi]$ .

One can observe that the rank-3 singularities occur for the PA-10 when it is completely stretched out (workspace boundary singularity) or when it is folded back on itself.

In general, our approach for identifying robot singularities does not consider physical joint limits. One can employ our technique to find any robot singular configuration but one must exclude infeasible angles due to mechanical limits.

## 2.4 CONCLUSIONS

This work has proposed a procedure based on computing the gradient of a singular value to drive a robot into a singular configuration. This algorithm is able to: (1) identify the singularities of any rank for any robot and (2) deal with ill-defined singular vectors when their corresponding singular values are equal. A second algorithm was presented to obtain the most intuitive representation of the singular vectors associated with configurations that correspond to high-rank singularities. Both algorithms are applicable to robots with an arbitrary number of degrees of freedom and of arbitrary kinematic structure. These algorithms were illustrated on a 4-DoF redundant positioning robot and on a 7-DoF redundant PA-10 robot.

## Chapter 3

# Maximizing the Size of Self-Motion Manifolds to Improve Robot Fault Tolerance<sup>4</sup>

### 3.1 INTRODUCTION

Fault tolerance has been a critical factor in the design and operation of robotic systems that are meant to operate in harsh environments. Due to the mission-critical nature of some robotic applications, failure could result in catastrophic loss of life and/or property. Robots used in search and rescue operations launched after disasters are good examples of when reliability is important [1]. Previous work has shown that the availability of robots in such harsh environments is as low as 50% [27]. Because certain failures can put the entire mission in jeopardy [2], work has been done to redesign rescue robots to make them fault tolerant [28].

Many different aspects of fault tolerance have been considered, such as fault detection, identification, and analysis, as surveyed in [29]. Researchers have also looked at fault-tolerant control of actuators, for example, in automated underwater vehicles [30] [31]. Fault-tolerant control for multirobot systems with undetected failures is discussed in [32]. In all cases, fault tolerance requires redundancy at some level. Categories of redundancy include: *structural redundancy*, e.g., duplicating parts that are most susceptible to failures [3]; *functional redundancy*, i.e., a human operator intervenes to assess faults and implement a work-around; *analytical redundancy*, e.g., when a tachometer signal is integrated to recover from a failed position sensor [29]; and *kinematic redundancy*, i.e., a robot is designed to have more degrees of freedom (DoFs) than the minimum required to complete a task in order to compensate for the lost joint(s). The work presented here focuses on the study of kinematic redundancy.

---

<sup>4</sup>This chapter was published in [26]

To quantify the impact of incorporating kinematic redundancy, researchers have classified the measures of fault tolerance into two categories, i.e., local and global. Quantitative local fault tolerance measures are typically based on the singular value decomposition of the robot’s Jacobian matrix. Such measures include the *minimum singular value* [4], the *condition number* [5], and the *robot manipulability* [6]. The kinematic redundancy is used to configure the robot so that it optimizes the fault tolerance measure. Techniques for doing so frequently involve the *gradient of the minimum singular value* [23].

Global fault-tolerance measures, that typically define reachable workspaces, are useful for pick-and-place tasks. One such measure [4] can be used to identify the best fault-tolerant location for these types of tasks. It quantifies the size of the workspace where the robot can operate before a failure and still return to the desired location after a failure. This can be guaranteed if the robot is operated within specific joint limits determined from the range of the robot’s self-motion manifold. The problem becomes more challenging if the aim is to design a fault-tolerant workspace that is reachable for any trajectory both before and after a failure. A technique for computing the boundaries of such a workspace is presented in [33]. One can use these global measures to evaluate and select the optimal kinematic parameters when designing redundant robots with the same number of DoFs [34].

This work focuses on improving the global fault tolerance of a robot by maximizing the size of a pre-failure workspace while guaranteeing the reachability of a critical task location. This is done by identifying “large” self-motion manifolds, where the metric for the size depends on the ranges of each of the joints, i.e., their bounding box. It is shown that such large self-motion manifolds can be found by searching near high-rank singular configurations because these configurations represent connections of two or more previously disjoint manifolds.

The rest of this paper is organized as follows. An overview of the terminology and background concepts used is presented in section II. In section III, the approach used to analyze a robot design in order to identify its best fault tolerant location(s) is explained. This approach is then illustrated

on a common existing redundant robot design in section IV. Finally, the conclusions of this work are presented in section V.

## 3.2 BACKGROUND

### 3.2.1 Self-Motion Manifolds

The forward kinematics of a robot is represented as

$$\boldsymbol{x} = f(\boldsymbol{\theta}) \quad (3.1)$$

where  $\boldsymbol{x}$  is an  $m$ -dimensional vector representing the end-effector location (position and orientation) and  $\boldsymbol{\theta}$  is an  $n$ -dimensional vector representing the joint angles. For redundant robots,  $n > m$ , where  $n - m$  is the degree of redundancy. In this case, the self-motion manifold(s) is (are) the set of all solutions that result from solving the inverse-kinematic problem represented by

$$\boldsymbol{\theta} = f^{-1}(\boldsymbol{x}). \quad (3.2)$$

The upper limit on the number of self-motion manifolds for redundant spherical, positional, and spatial manipulators is 2, 4, and 16, respectively [35]. The relationship between the robot's joint velocity and its end-effector velocity is represented by

$$\dot{\boldsymbol{x}} = \boldsymbol{J}\dot{\boldsymbol{\theta}} \quad (3.3)$$

where  $\boldsymbol{J}$  is the  $m \times n$  Jacobian. At the velocity level, self motion corresponds to:

$$\boldsymbol{J}\dot{\boldsymbol{\theta}} = \mathbf{0}. \quad (3.4)$$

For the case where  $n - m = 1$  and the robot is in a non-singular configuration, the null space is one dimensional. In this case, the null space will be represented by the unit vector  $\hat{\boldsymbol{n}}_J$ , which is tangent

to a self-motion manifold associated with this location. One can use  $\hat{\mathbf{n}}_J$  to map out the self-motion manifold(s) by integrating how it evolves under the constraint of maintaining a fixed desired end-effector location,  $\mathbf{x}_d$ . Numerically, this can be done by identifying an initial configuration  $\boldsymbol{\theta}_0$  where  $\mathbf{x}_d = f(\boldsymbol{\theta}_0)$ , and repeatedly solving

$$\Delta\boldsymbol{\theta} = \gamma\hat{\mathbf{n}}_J + \mathbf{J}^+\Delta\mathbf{x}_e \quad (3.5)$$

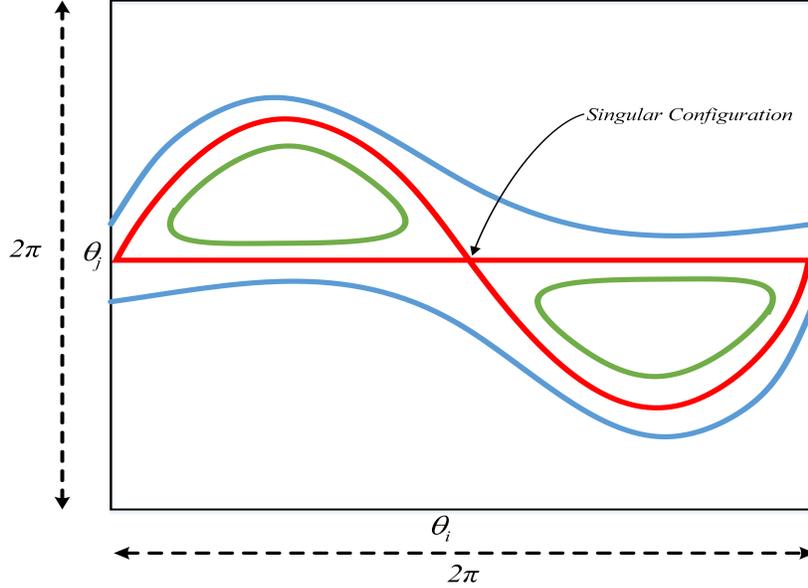
where  $\Delta\boldsymbol{\theta}$  is the change in the joint angles,  $\gamma$  is a real positive scalar that represents the step size along the manifold, and  $\mathbf{J}^+\Delta\mathbf{x}_e$  is an error correction term where  $\mathbf{J}^+$  is the pseudoinverse of the Jacobian matrix and  $\Delta\mathbf{x}_e$  is the end-effector error, i.e., the difference between  $f(\boldsymbol{\theta} + \Delta\boldsymbol{\theta})$  and  $\mathbf{x}_d$ .<sup>5</sup> If there are multiple self-motion manifolds, this procedure must be performed on each of them with an appropriate initial  $\boldsymbol{\theta}_0$ . The characteristics of the individual manifolds can be significantly different in terms of their shape and size.

### 3.2.2 Size of Self-Motion Manifolds

To determine the length of a one-dimensional self-motion manifold, one only needs to sum up the number of times that (4.6) is solved to traverse the entire manifold. To identify when one has returned to the initial configuration, one must be careful to consider the case when one or more of the joints has rotated by  $2\pi$ . More so than length, the range through which each joint angle moves is a useful measure of the robot's fault tolerance at the location associated with this manifold. These ranges define a bounding box, the volume of which has been used as one measure of fault tolerance [4]. Unfortunately, many common 7-DoF robot designs that are like a human arm, have self-motion manifolds that have a zero range for a particular joint, i.e., the elbow joint. Therefore, throughout this work we always use the sum of all joint angle ranges for all self-motion manifolds associated with a location as a measure of fault tolerance.

---

<sup>5</sup>The case where  $\boldsymbol{\theta}$  results in a singular configuration and null space is multi-dimensional will be discussed in Section 3.3



**Figure 3.1:** An illustrative example of commonly occurring self-motion manifolds for high-dimensional manipulators is shown projected into the two-dimensional space for joints  $\theta_i$  and  $\theta_j$ . The single one-dimensional manifold shown in red is the connection of two previously disjoint manifolds. At the center of the figure, the intersection is a singular configuration where the null space is two-dimensional. Note that this is a true intersection and not simply due to the projection onto the  $\theta_i - \theta_j$  plane. If one perturbs the end-effector location from the one associated with the red manifold, the resulting manifolds can be quite different depending on the direction of the perturbation. In blue, the one red manifold splits into two open manifolds and in green into two closed manifolds, where open refers to the fact that  $\theta_i$  can take on any value.

Fortunately, for higher degrees of redundancy (where  $n - m > 1$ ) that result in higher dimensional self-motion manifolds, computing an estimate of a bounding box is more tractable than computing areas, volumes, or hypervolumes of manifolds. For these cases, a bounding box on the joint angle ranges can be computed by modifying (4.6) to

$$\Delta\theta = \gamma N_J \hat{e}_i + J^+ \Delta x_e \quad (3.6)$$

where  $N_J$  is a projection onto the  $(n - m)$ -dimensional null space of the Jacobian and  $\hat{e}_i$  is a unit vector along the  $i$ th joint angle, where  $1 \leq i \leq n$  [4]. By repeatedly solving (4.7), for  $i = 1$  to  $n$ , one can find an approximation of joint-angle ranges that can be used to compute the self-

motion manifold size<sup>6</sup>. The iteration defined by (4.7) should be terminated when either joint angle  $i$  traverses  $2\pi$  or the projection of  $\hat{e}_i$  onto the null space becomes zero. In the latter case, this may be a local minimum so that this measure is a lower bound on the range of joint  $i$ .

As illustrated in Fig. 3.1, singularities play a critical role in the size and shape of self-motion manifolds. At singularities, two (or more) self-motion manifolds can touch and become one manifold, or one manifold can tear apart. This means that larger manifolds tend to include one or more singularities and so it is natural to search for large manifolds near singularities.

### 3.3 IDENTIFYING LARGER SELF-MOTION MANIFOLDS

As discussed above, the larger (thus the more fault-tolerant) self-motion manifolds exist near singularities, so that one should employ a technique for identifying singular configurations. There are many techniques for doing so, e.g., symbolically solving for when the determinant of  $\mathbf{J}$  becomes zero [13] or using reciprocity-based resolution [36]. However, here we employ a technique based on the gradients of the singular values [23] because of its ability to identify high-rank singularities. The singular value decomposition of  $\mathbf{J}$  can be defined as

$$\mathbf{J} = \mathbf{U}\mathbf{D}\mathbf{V}^\top \quad (3.7)$$

where  $\mathbf{U}$  is an  $m \times m$  orthogonal matrix of the output singular vectors,  $\mathbf{V}$  is an  $n \times n$  orthogonal matrix of the input singular vectors, and  $\mathbf{D}$  is an  $m \times n$  diagonal matrix where its diagonal elements are the ordered singular values, i.e.,  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_m \geq 0$ . It can be rewritten as a summation in terms of the singular vectors

$$\mathbf{J} = \sum_{i=1}^r \sigma_i \hat{\mathbf{u}}_i \hat{\mathbf{v}}_i^\top \quad (3.8)$$

where the vectors  $\hat{\mathbf{u}}_i$  and  $\hat{\mathbf{v}}_i$  represent the output and input singular vectors, respectively. For nonsingular  $\mathbf{J}$ , the value of  $\sigma_m$  represents the distance to a singularity. The rank of  $\mathbf{J}$ , denoted  $r$ , is less than  $m$  if the robot is singular (i.e.,  $\sigma_i = 0$  for  $i > r$ ). In this case, the value of  $\sigma_r$  is the

---

<sup>6</sup>This paper will focus only on the 7-DoF case, but future work will expand to higher DoF robots.

distance to the next higher-rank singularity. One can drive the robot towards the nearest singularity by moving the robot along the gradient of  $\sigma_m$  until it reaches zero. From (4.4), it is easy to see that any singular value  $\sigma_i$  can be written as

$$\sigma_i = \hat{\mathbf{u}}_i^\top \mathbf{J} \hat{\mathbf{v}}_i. \quad (3.9)$$

By differentiating (4.9) with respect to time, one obtains

$$\dot{\sigma}_i = \dot{\hat{\mathbf{u}}}_i^\top \mathbf{J} \hat{\mathbf{v}}_i + \hat{\mathbf{u}}_i^\top \dot{\mathbf{J}} \hat{\mathbf{v}}_i + \hat{\mathbf{u}}_i^\top \mathbf{J} \dot{\hat{\mathbf{v}}}_i \quad (3.10)$$

that can be simplified to

$$\dot{\sigma}_i = \hat{\mathbf{u}}_i^\top \dot{\mathbf{J}} \hat{\mathbf{v}}_i. \quad (3.11)$$

The partial derivative of  $\sigma_i$  with respect to  $\theta_k$  can be written as

$$\frac{\partial \sigma_i}{\partial \theta_k} = \hat{\mathbf{u}}_i^\top \frac{\partial \mathbf{J}}{\partial \theta_k} \hat{\mathbf{v}}_i \quad (3.12)$$

where

$$\frac{\partial \mathbf{J}}{\partial \theta_k} = \left[ \frac{\partial \mathbf{j}_1}{\partial \theta_k}, \frac{\partial \mathbf{j}_2}{\partial \theta_k}, \dots, \frac{\partial \mathbf{j}_n}{\partial \theta_k} \right]. \quad (3.13)$$

The partial derivative of the  $i^{\text{th}}$  column of the Jacobian is given by [23], [22]

$$\frac{\partial \mathbf{j}_i}{\partial \theta_k} = \begin{cases} \begin{bmatrix} (\mathbf{z}_k^\top \mathbf{p}_i) \mathbf{z}_i - (\mathbf{z}_k^\top \mathbf{z}_i) \mathbf{p}_k \\ \mathbf{z}_k \times \mathbf{z}_i \end{bmatrix}, & k < i \\ \begin{bmatrix} (\mathbf{z}_i^\top \mathbf{p}_k) \mathbf{z}_k - (\mathbf{z}_k^\top \mathbf{z}_i) \mathbf{p}_k \\ \mathbf{0} \end{bmatrix}, & k \geq i \end{cases} \quad (3.14)$$

Utilizing (4.12), (4.13), and (4.14), one can easily compute the gradient of any singular value of  $\mathbf{J}$  as

$$\nabla\sigma_i = \left[ \frac{\partial\sigma_i}{\partial\theta_1}, \frac{\partial\sigma_i}{\partial\theta_2}, \dots, \frac{\partial\sigma_i}{\partial\theta_n} \right]. \quad (3.15)$$

Moving along this gradient allows one to increase or decrease any desired singular value.

Finding the largest self-motion manifold(s) of any robot is a two-step process. The first step is to find all singular configurations, including high-rank singularities. In this step, we employ the gradient descent technique to minimize a desired singular value

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \alpha \nabla\sigma_i \quad (3.16)$$

where  $\boldsymbol{\theta}^{(k)}$  is the current joint configuration,  $\boldsymbol{\theta}^{(k+1)}$  is the next joint configuration, and  $\alpha$  is a positive scalar, often referred to as the step size. To identify rank-1 singularities, we start with generating random configurations in the joint space. Then, starting from each random configuration, we move the robot along the negative direction of the gradient of  $\sigma_m$  until the value of  $\sigma_m$  approaches zero. The set of final configurations that satisfy the condition of  $\sigma_m \leq \epsilon$  are the rank-1 singularities of the robot, where  $\epsilon$  is a user-defined threshold.

To identify the rank-2 singularities, we start from the same random configurations used to identify rank-1 singularities, but the robot is first moved along the negative direction of  $\nabla\sigma_{m-1}$  until the value of  $\sigma_{m-1}$  approaches zero, i.e.,  $\sigma_m \leq \sigma_{m-1} \leq \epsilon$ . In some cases  $\sigma_{m-1}$  and  $\sigma_m$  become nearly equal before  $\sigma_{m-1}$  approaches zero. This means that the singular vectors  $\hat{\mathbf{u}}_{m-1}$  and  $\hat{\mathbf{u}}_m$  (as well as  $\hat{\mathbf{v}}_{m-1}$  and  $\hat{\mathbf{v}}_m$ ) are ill-defined. That is, any vectors in the subspaces  $\{\hat{\mathbf{u}}_{m-1}, \hat{\mathbf{u}}_m\}$  and  $\{\hat{\mathbf{v}}_{m-1}, \hat{\mathbf{v}}_m\}$  are valid singular vectors for the gradient computation in (4.11) and (4.12). In such cases, we rotate the  $\{\hat{\mathbf{u}}_{m-1}, \hat{\mathbf{u}}_m\}$  and  $\{\hat{\mathbf{v}}_{m-1}, \hat{\mathbf{v}}_m\}$  subspaces so that the angle between  $\nabla\sigma_m$  and  $\nabla\sigma_{m-1}$  is minimized. (The angle between the gradients of  $\sigma_m$  and  $\sigma_{m-1}$  may vary from 0 to  $\pi$ .) We then reduce  $\sigma_{m-1}$  by moving along a negative direction of a linear combination of the gradients of  $\sigma_m$  and  $\sigma_{m-1}$ . We optimize the linear combination by doing two one-dimensional searches. The first search is along  $\beta\nabla\sigma_m + (1 - \beta)\nabla\sigma_{m-1}$  to determine the optimal value of  $\beta$ , where  $0 \leq \beta \leq 1$ , that minimizes  $\sigma_{m-1}$ . The second, is to determine the optimal value of the adaptive step size  $\alpha$  along the negative direction of the computed combination. Those configurations that

converge to where  $\sigma_m \leq \sigma_{m-1} \leq \epsilon$  are rank-2 singularities. If the process does not converge, then a rank-2 singularity does not exist near this configuration. An analogous procedure is used to find rank-3 (and higher-rank) singularities. There are two cases where the singular vectors are ill-defined, i.e., when  $\sigma_{m-2}$  and  $\sigma_{m-1}$  are nearly equal or if the three singular values  $\sigma_{m-2}$ ,  $\sigma_{m-1}$ , and  $\sigma_m$  are all nearly equal. In the first case, the optimization described above can be performed for  $\sigma_{m-2}$  and  $\sigma_{m-1}$  to minimize  $\sigma_{m-2}$ . In the second case, one needs to search for a suitable rotation for the  $\{\hat{\mathbf{u}}_{m-2}, \hat{\mathbf{u}}_{m-1}, \hat{\mathbf{u}}_m\}$  and  $\{\hat{\mathbf{v}}_{m-2}, \hat{\mathbf{v}}_{m-1}, \hat{\mathbf{v}}_m\}$  subspaces to minimize the sum of the angles between the gradients of the three singular values. Then, one must find a suitable combination of the gradients and a step size that minimize  $\sigma_{m-2}$ . An analogous process is repeated for higher-rank singularities until there is no possible joint motion that will reduce  $\sigma_r$  while keeping  $\sigma_m \leq \sigma_{m-1} \leq \dots \leq \sigma_{r+1} \leq \epsilon$ , i.e., there are no rank- $(m - r)$  singularities. The pseudocode for performing this procedure is given in Algorithm 1.

The second step is to compute all the self-motion manifolds that include these singular configurations. However, one would like to reduce the number of these configurations, in order to reduce the amount of computations that result in very similar manifolds. Therefore, if two singular configurations are close to each other, then only one of them will have its self-motion manifold computed. Finding the self-motion manifolds for each singularity configuration can be done by starting the robot in that singular configuration and then repeatedly solving (4.6) until an entire self-motion manifold is computed.

Before applying (4.6), one must first compute the end-effector location associated with this singular configuration. Then, from the singular configuration, the robot is moved in each of the  $(n - r)$  directions of the null space. Mathematically,

$$\Delta\theta = \gamma\hat{\mathbf{v}}_i + \mathbf{J}^+\Delta\mathbf{x}_e \quad (3.17)$$

---

<sup>7</sup>The value of  $N$  should be large enough to sufficiently span the joint space. An analysis of this user-defined parameter is provided in subsection IV(c) below.

---

**Algorithm 1** Identify all-rank singular configurations

---

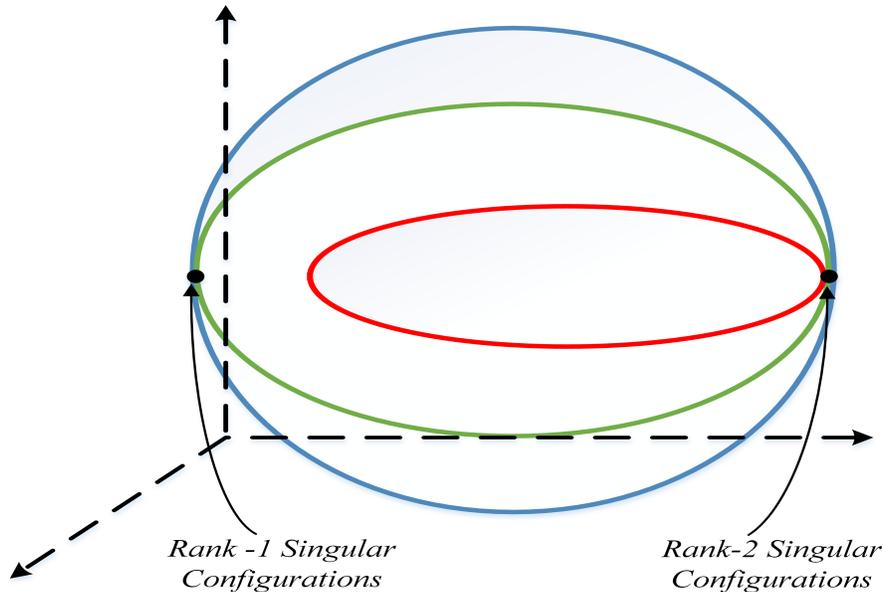
```
1: select N random joint-space configurations7
2: for  $i = 6$  to 1 do {for each workspace dimension}
3:   for  $j = 1$  to  $N$  do {for each random configuration}
4:     select  $j^{\text{th}}$  joint-space configuration  $\theta$ 
5:     compute  $\sigma_i$  {robot Jacobian's  $i^{\text{th}}$  singular value}
6:     while  $\sigma_i \geq \epsilon$  do
7:       if ( $i = 6$ ) then {for rank-one singularities}
8:          $\nabla\sigma = \nabla\sigma_6$ 
9:       else {for high-rank singularities}
10:        for all  $\sigma_k$  where  $\sigma_i \approx \sigma_k$  do {where  $k < i$ }
11:          rotate  $\mathbf{U}$  and  $\mathbf{V}$  subspace associated with  $\sigma_i$  and  $\sigma_k$ 's {to minimize the angles
12:            between  $\nabla\sigma_i, \nabla\sigma_{i+1}, \dots, \nabla\sigma_k$ }
13:          compute  $\nabla\sigma$  {optimal linear combination of the gradients of the singular values}
14:        end for
15:      end if
16:      compute  $\alpha$  {adaptive linear search along  $\nabla\sigma$ }
17:      update  $\theta$  { $\theta^{(k+1)} = \theta^{(k)} - \alpha\nabla\sigma$ }
18:      if ( $\sigma_i$  did not decrease) then
19:        go to 23 {local minimum of  $\sigma_i$ }
20:      end if
21:    end while
22:    save  $\theta$  and singularity rank
23:  end for
```

---

for  $i = r$  to  $n$ , where these  $\hat{v}_i$  are the singular vectors that span the  $(n - r)$ -dimensional null space of the Jacobian at the singular configuration. This guarantees that all of the one-dimensional self-motion manifolds that touch at this configuration can be computed (see Fig. 3.1), because away from the singular configuration the null space becomes one dimensional and well defined. Therefore one can repeatedly solve (4.6) until the robot returns to the initial starting configuration.

However, it is common for self-motion manifolds to include multiple singular configurations (see Fig. 3.2). If while solving (4.6) the null space becomes multi-dimensional, i.e., another singularity is encountered, then one should be careful to select a null-space vector from this higher dimensional subspace that is as close as possible to the one used to enter the singularity. For computational efficiency, one should check any new starting configuration with all previously com-

puted manifolds to prevent redundant computations. The pseudocode for performing the second step is given in Algorithm 2.



**Figure 3.2:** This sketch illustrates the case where three previously disjoint manifolds (red, green, and blue) touch and become one manifold, i.e., a rank-2 singularity. Also, two of these manifolds (green and blue) touch elsewhere creating a rank-1 singularity. Many variations of different-rank singularities can exist on a single manifold.

In general, an end-effector location will have multiple disjoint self-motion manifolds, i.e., the robot cannot move from one manifold to another without changing the end-effector location. To compute the “fault tolerance” of the end-effector location associated with a singular configuration, one needs to compute the “bounding box” of all the self-motion manifolds, i.e., the ranges for each of the joints while staying at this end-effector location. Because of multiple disjoint self-motion manifolds one needs to decide if it is important to reconfigure the arm from one manifold to another without moving the end effector. If it is not, then one can simply take the union of all the joint-angle ranges for each manifold. If it is, one can only take the union for those manifolds that touch. One measure of the size of the bounding box is simply the summation of all these joint-angle ranges. The above procedure for identifying the largest self-motion manifold is illustrated for the well-known 7-DoF Mitsubishi PA-10 robot in the next section.

---

**Algorithm 2** Compute all self-motion manifolds with singularities

---

```
1: import N singular configurations and their ranks {from algorithm 1}
   {remove duplicate singularities}
2:  $\forall i, j \leq N, i \neq j$ 
3: if  $\theta(i) \approx \theta(j)$  then
4:   delete  $\theta(j)$ 
5: end if
   {compute all self-motion manifolds SMMs}
6: for each singularity rank do
7:   for all singular configurations  $\theta$  do {of each rank}
8:     if  $\theta$  does not exist on a previously computed manifold then
9:       compute  $x_e$  {the end-effector location}
10:      find the  $n - r$  configurations near  $\theta$  that satisfy the  $n - r$  singular directions at  $x_e$ 
11:      for  $k = 1$  to  $n - r$  do
12:        if  $k^{th}$  configuration does not exist on a computed manifold then
13:          start at the  $k^{th}$  configuration
14:          while not back to starting configuration do
15:            compute the robot Jacobian ( $J$ )
16:            compute the null vector ( $\hat{n}_J$ )
17:            compute  $\Delta\theta$  { $\Delta\theta = \gamma\hat{n}_J + J^+\Delta x_e$ }
18:            update joint angles { $\theta_{new} = \theta_{old} + \Delta\theta$ }
19:          end while
20:        end if
21:      end for
22:    end if
23:  end for
24: end for
```

---

## 3.4 PA-10 ROBOT CASE STUDY

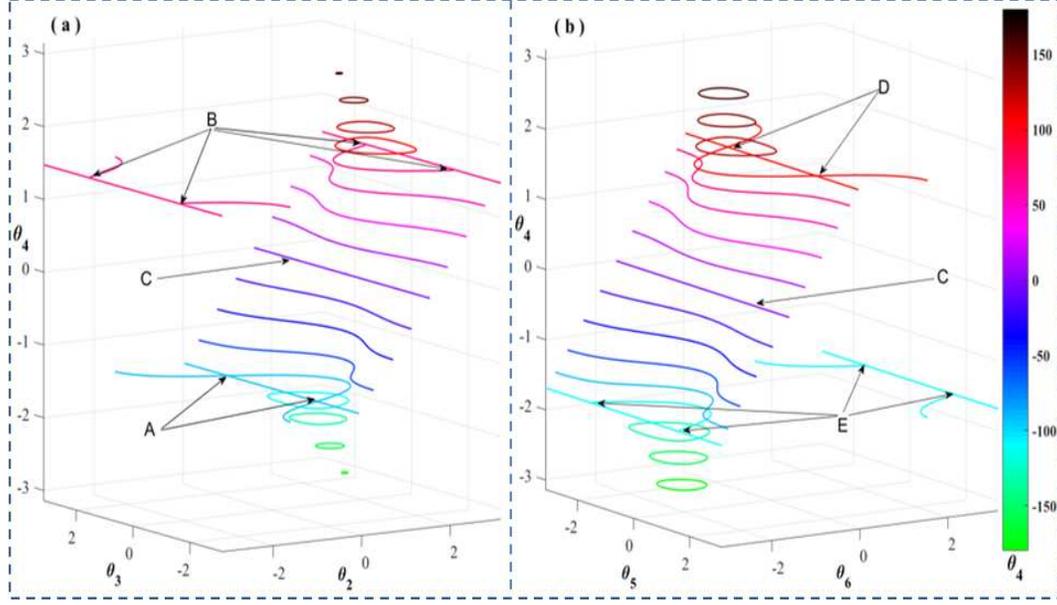
### 3.4.1 PA-10 background

The Mitsubishi PA-10 is used as an illustrative example because it has a well-known, commonly occurring 7-DoF kinematic structure. We first explain the behavior of the self-motion manifolds of the PA-10 when the end-effector location is changed. Then, the algorithms described in the previous section are used to identify the largest self-motion manifolds and how they relate to fault tolerance. The original DH parameters for the PA-10 are given in Table 4.3 [37], with the end effector positioned so that the last link's displacement,  $d_7$ , is equal to  $d_3$ .

**Table 3.1:** The DH Parameters of the PA-10 Robot

$Link_i$	$\alpha_i[rad]$	$a_i[m]$	$d_i[m]$	$\theta_i[rad]$
1	$-\pi/2$	0	0	$\theta_1$
2	$\pi/2$	0	0	$\theta_2$
3	$-\pi/2$	0	0.45	$\theta_3$
4	$\pi/2$	0	0	$\theta_4$
5	$-\pi/2$	0	0.50	$\theta_5$
6	$\pi/2$	0	0	$\theta_6$
7	0	0	0.45	$\theta_7$

Because the elbow joint, i.e.,  $\theta_4$ , is the only one that can change the distance between the shoulder and the wrist, there can be no component of  $\theta_4$  during self motion. Therefore, it is possible to categorize self-motion manifolds based on the value of  $\theta_4$ . This is illustrated in Fig 3.3 where a single manifold for each end-effector location is shown with  $\theta_4$  ranging from  $-\pi$  to  $\pi$ . All of the singularities shown in this figure are rank-1 singularities.



**Figure 3.3:** This figure shows 3-D projections of the PA-10 self-motion manifolds generated by changing  $\theta_4$  from  $-\pi$  to  $+\pi$ , where the value of  $\theta_4$  is indicated using color. Subfigure (a) is a projection in  $[\theta_2, \theta_3, \theta_4]$  and (b) is a projection in  $[\theta_6, \theta_5, \theta_4]$ . Singularities occur at (A) where  $\theta_2 = 0, \theta_3 = \pm\pi/2$ , (B) where  $\theta_2 = \pm\pi, \theta_3 = \pm\pi/2$ , (C) where  $\theta_4 = 0$ , (D) where  $\theta_6 = 0, \theta_5 = \pm\pi/2$ , and (E) where  $\theta_6 = \pm\pi, \theta_3 = \pm\pi/2$ .

The self-motion manifolds in Fig. 3.3 exhibit all of the properties shown in Fig. 3.1, i.e., there are both open and closed manifolds with manifolds connecting/separating at singularities. It is also clear that the largest manifolds are those that include singularities.

The singularities of the PA-10 have been well studied [36], [18]. However, it is important to note that Algorithm 1 above can be applied to any arbitrary robot structure and its computational complexity does not change for high-rank singularities, which are important for identifying the largest self-motion manifold.

### 3.4.2 The largest self-motion manifold

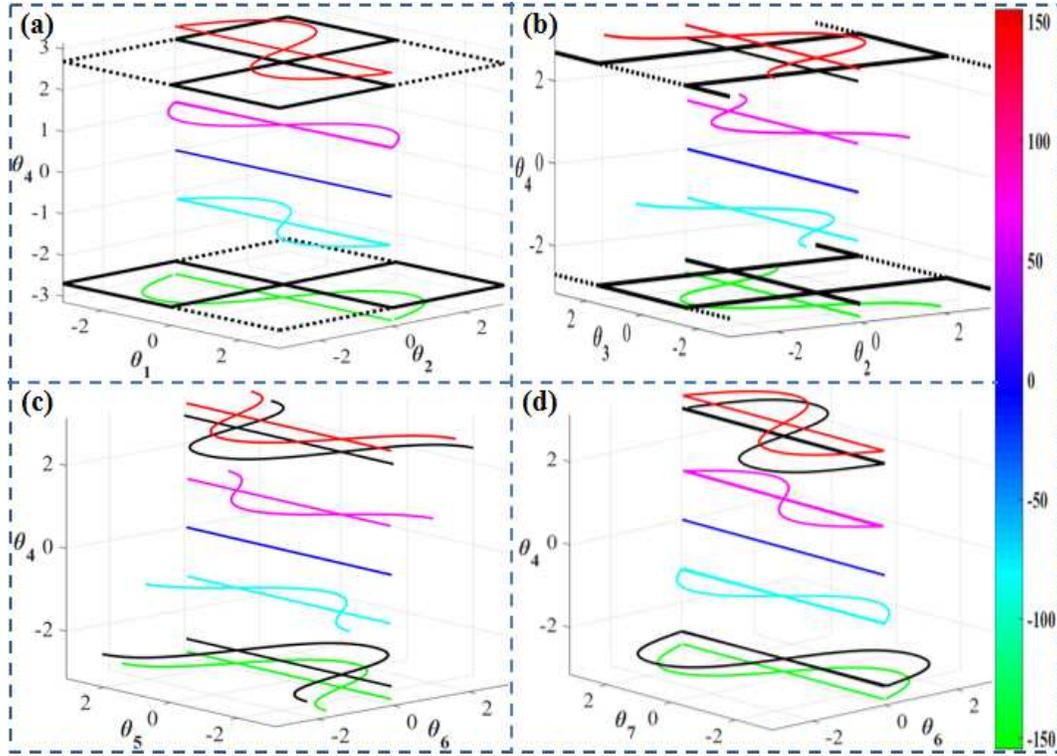
Once all the singularities are identified, Algorithm 2 is able to compute the size of self-motion manifolds that include these singularities. It identified the largest self-motion manifold to be 35.90

$rad$ <sup>8</sup>, where the ranges of  $\theta_1, \theta_2, \theta_3, \theta_5$ , and  $\theta_7$  are  $2\pi rad$ , the range of  $\theta_6$  is  $4.48 rad (\pm 2.24 rad)$ <sup>9</sup>, and the range of  $\theta_4$  is zero, where its value is either  $+2.69$  or  $-2.69 rad$ . To help understand why Algorithm 2 identified this as the optimal solution, consider Fig. 3.4 that shows several self-motion manifolds and their associated singularities characterized by varying  $\theta_4$  from  $-\pi$  to  $\pi$ , analogous to Fig. 3.3. The black manifolds, where  $\theta_4 = \pm 2.69 rad$ , are clearly the largest self-motion manifolds. The nearby manifolds shown in green and red, where  $\theta_4 = -3.00 rad$  and  $\theta_4 = 3.00 rad$ , respectively, are shown to illustrate the behavior at these singularities. The subfigures (a)-(d) are all different projections of the same manifolds, i.e., any color corresponds to only a single manifold. These projections have been selected to illustrate which joint angles have an unrestricted range, i.e.,  $\theta_1, \theta_2, \theta_3, \theta_5$ , and  $\theta_7$ , whereas  $\theta_6$  has a range of  $\pm 2.24 rad$ , which is clearly shown in (c) and (d).

---

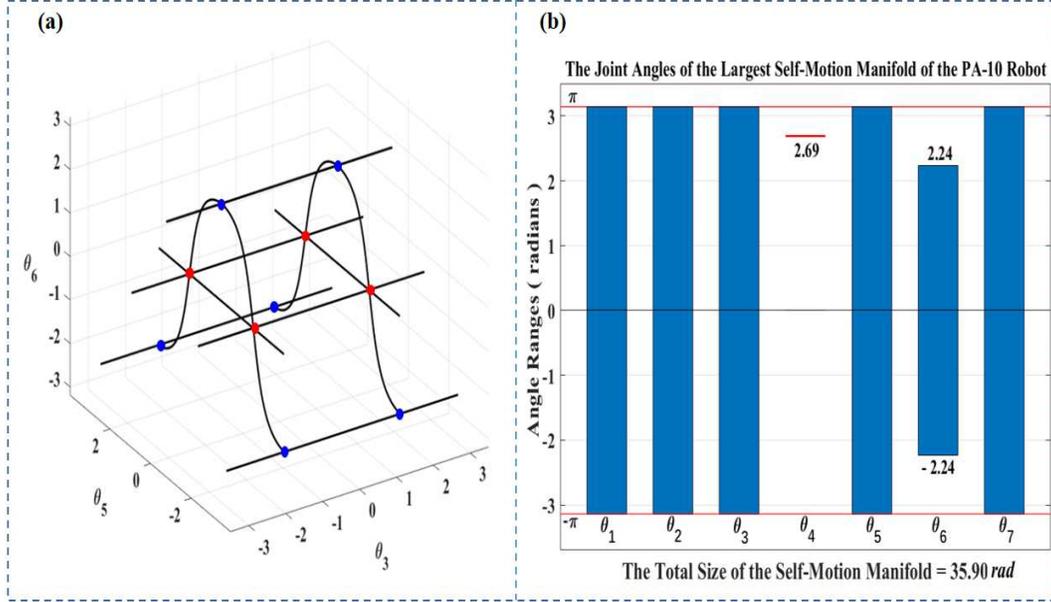
<sup>8</sup>This is the theoretical maximum size where the actual value would include any joint limits on  $\theta_2$  and  $\theta_6$  that are due to self collision.

<sup>9</sup>If one wanted to modify the link offsets of the PA-10 to make  $d_3 = d_5$ , then the size of the largest self-motion manifold could be increased so that  $\theta_6$  would also have a range of  $2\pi$ . This would also change the optimal value of  $\theta_4$  to 0.



**Figure 3.4:** The subfigures (a)-(d) show different 3-D projections of the PA-10 self-motion manifolds characterized by  $\theta_4$ , where any color corresponds to only a single manifold. The optimal self-motion manifolds identified by Algorithm 2 are shown in black where  $\theta_4 = \pm 2.69 \text{ rad}$ . The dotted lines are used to better show parts of the manifold that did not exist on nearby manifolds. The red and green manifolds where  $\theta_4 = 3.00 \text{ rad}$  and  $\theta_4 = -3.00 \text{ rad}$ , respectively, are shown to illustrate the behavior at a singularity.

One should note that the two black manifolds correspond to the same end-effector location, however, the robot cannot move from one manifold to the other without changing this location. The rank of the various singularities on the black manifold is not clear from Fig. 3.4 because they all appear to be of rank 1 due to the projections used. If one looks at the projection in  $\theta_3, \theta_5$ , and  $\theta_6$  space, as in Fig. 3.5(a), then it becomes clear that the black manifold contains four rank-1 and four rank-2 singularities, shown in blue and red respectively. (The four blue dots, rank-1 singularities, at the lower part of Fig. 3.5(a) only represent two singularities, i.e., the dots at  $\theta_5 = \pi$  are the same as those at  $\theta_5 = -\pi$ ).



**Figure 3.5:** In (a) the largest optimal self-motion manifold is shown projected into the  $\theta_3$ ,  $\theta_5$ , and  $\theta_6$  subspace. From this projection one can see that this manifold contains four rank-1 and four rank-2 singularities, shown in blue and red respectively. Note that the two blue singularities at the bottom of the figure, i.e., where  $\theta_6 = -2.24 \text{ rad}$ , are shown twice at both  $\theta_5 = \pm\pi$ . The rank-2 singularities occur when  $\theta = [0, 0, \frac{\pm\pi}{2}, \theta_4, \frac{\pm\pi}{2}, 0, 0]$ , where in this case  $\theta_4 = 2.69 \text{ rad}$ . The ranges of each of the joints is shown in (b) where  $\theta_1, \theta_2, \theta_3, \theta_5$ , and  $\theta_7$  are  $2\pi$ , the range of  $\theta_6$  is  $4.48 \text{ rad}$  ( $\pm 2.24$ ), and the value of  $\theta_4$  is  $2.69 \text{ rad}$ , where its range is zero.

Fig. 3.5(b) shows the ranges of the seven joint angles of the PA-10 while operating on the largest self-motion manifold. There are two separate, equal-sized manifolds at this location. The other manifold has identical joint-angle ranges, except that  $\theta_4 = -2.69 \text{ rad}$ . The configurations with  $\theta_4 = \pm 2.69 \text{ rad}$  are special because they make the axis between the shoulder and the wrist horizontal, and the rotation around this axis can configure the robot into four rank-1 singularities and four rank-2 singularities. Operating the robot slightly away from these special values of  $\theta_4$  will not allow it to reach all the rank-1 singularities. This will not dramatically change the size of the self-motion manifold, however the robot will lose some ability to reconfigure that is offered by the rank-1 singularities.

### 3.4.3 Evaluation

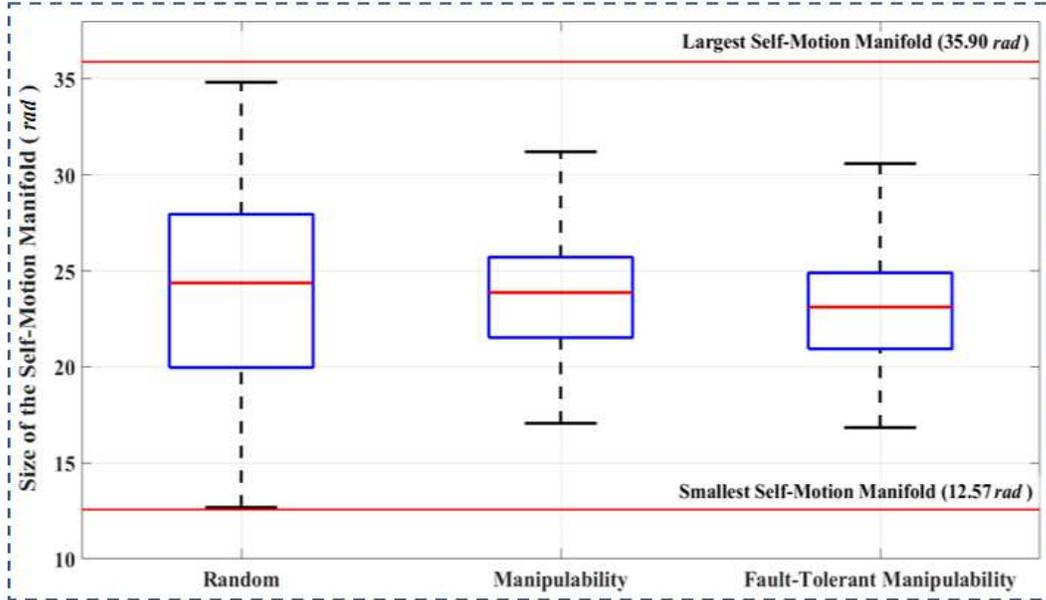
We first compare the proposed approach for identifying the largest self-motion manifold with a straightforward evaluation of self-motion manifolds generated at random configurations. Table 3.2 shows a comparison of the largest self-motion manifold identified by both techniques as a function of the number of random configurations  $N$ .

**Table 3.2:** Efficiency Comparison between the Proposed Algorithms and the Random Approach

No. of Random Configurations	Size of the Largest Self-Motion Manifold (Radians)	
	Random Approach	Proposed Algorithms
10	27.89	28.36
20	33.78	34.09
100	34.33	34.43
200	33.95	34.78
1,000	34.82	35.80
2,000	35.09	35.90
10,000	35.52	35.90

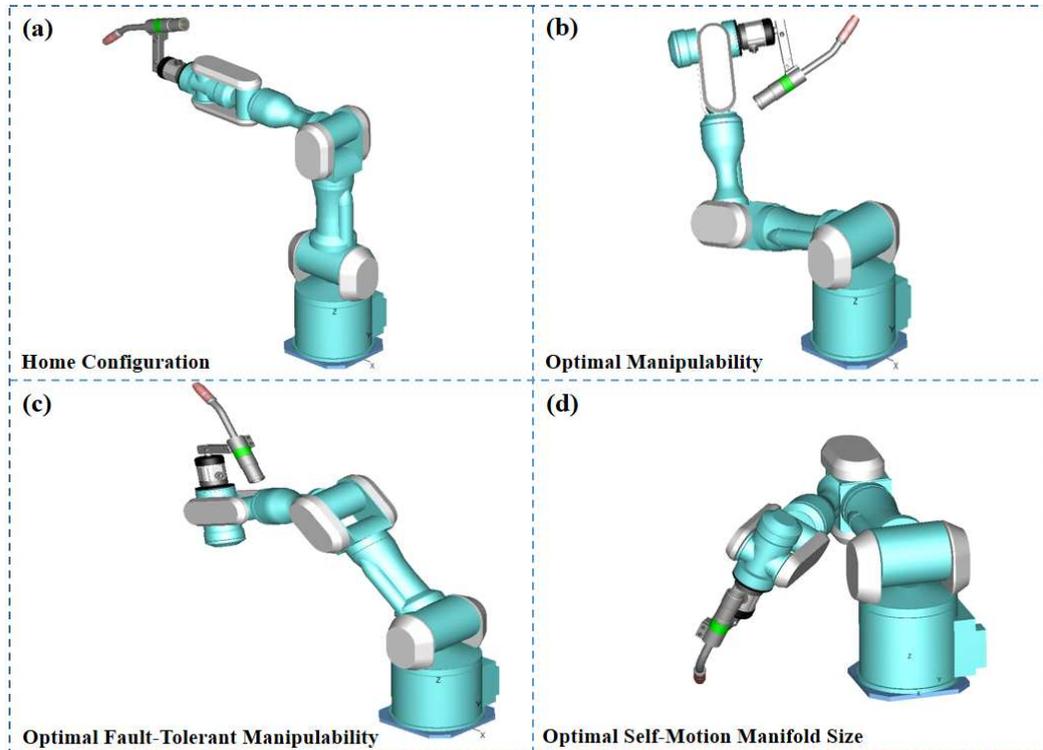
The data shows that the proposed approach converges to the maximum self-motion manifold size of 35.90 *rad* at  $N = 2000$ . This was verified using ten different populations of random configurations. It should also be noted that our proposed approach outperformed the random approach for any value of  $N$ .

Next, we compared our proposed algorithm to an approach that attempts to identify large self-motion manifolds using classical local dexterity [38] and fault-tolerance measures [6]. Fig. 3.6 shows the distribution of 1,000 self-motion manifold sizes computed from joint space configurations selected in three different ways. The left distribution is from the  $N = 1,000$  entry in Table 3.2, which is generated randomly. The middle and the right distributions are for the 1,000 configura-



**Figure 3.6:** This figure shows a comparison of the distributions of self-motion manifold sizes computed for 1,000 random configurations (left) with those found for 1,000 configurations with the best local dexterity measure (middle) and fault-tolerant measure (right). The best 1000 manipulability and fault-tolerant manipulability configurations were selected from 10,000 random configurations. The minimum and the maximum sizes in each distribution are indicated with black horizontal lines and the red line is the median. The largest (35.90 rad) and smallest (12.57 rad) self-motion manifold sizes for this robot are also indicated.

tions with the best manipulability [38] and fault-tolerant manipulability [6], respectively, selected from 10,000 random configurations. Note that the distribution of the self-motion manifold sizes generated from traditional local measures of dexterity and fault tolerance are outperformed by the random approach. This indicates that there is no correlation between classical local measures and self-motion manifold size. In addition, none of these techniques is able to identify the largest self-motion manifold of the robot. However, it is possible to use our proposed approach to optimize both global fault tolerance, i.e., largest self-motion manifold size, and any desired local measure of dexterity or fault tolerance. For example, Fig. 3.7 (d) shows the PA-10 robot in one configuration on the largest self-motion manifold that minimizes the condition number of the Jacobian, i.e.,  $\sigma_1/\sigma_6 = 13.86$ . This illustrates that one does not have to operate near a singular configuration in order to obtain the benefits of a fault-tolerant location with a large self-motion manifold.



**Figure 3.7:** The PA-10 robot (with an arc-welding tool attached) is shown operating at multiple locations. Footnotemark[6] Subfigure (a) shows it in the home location, (b) in the best robot manipulability, (c) in the best fault-tolerant manipulability, and (d) in a location with the largest self-motion manifold. The self-motion manifold size associated with the best manipulability configuration is  $23.67 \text{ rad}$ , and for the fault-tolerant manipulability is  $22.45 \text{ rad}$ .

### 3.4.4 Example Use Case

We now present a simple example use case where the performance of the proposed technique is compared to existing approaches [38] [6]. Assume that a robot will be employed in a remote environment where repair is not feasible and one is designing the workspace to determine where a critical task should be placed. The home configuration of the robot is where all tasks start, and one would like to be able to reach the goal location even after any single locked-joint failure. The start location is shown in Fig. 3.7(a). We select the goal location using the technique presented here, i.e., the location with the largest self-motion manifold, Fig. 3.7(d), and compare it to goal

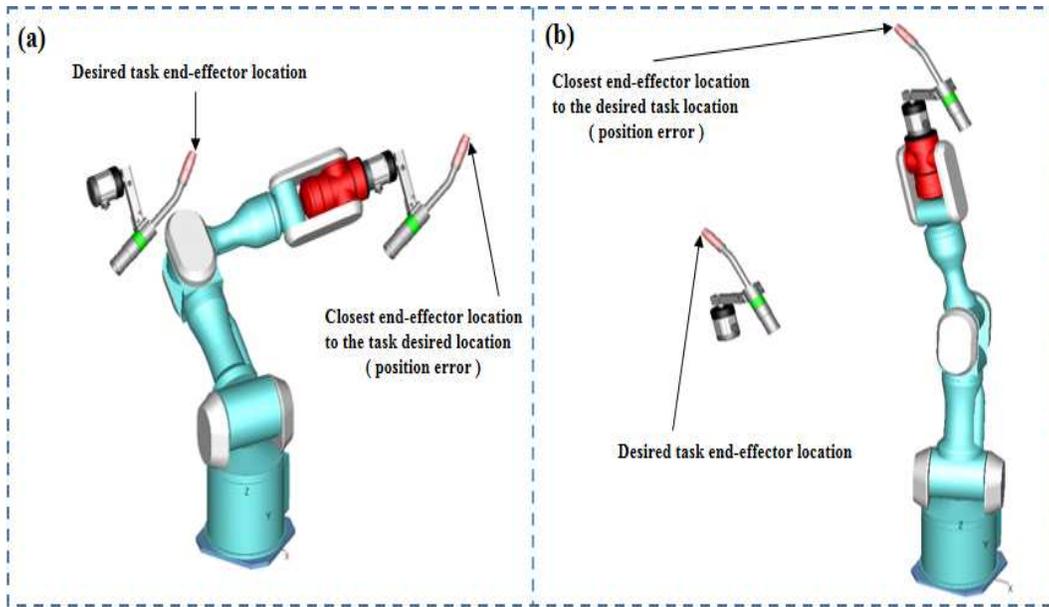
---

<sup>6</sup>Fig. 3.7 and Fig. 3.8 were generated by using the Workspace 5 software package from WAT Solutions, ([www.watsolutions.com](http://www.watsolutions.com)).

**Table 3.3:** Robot’s Home and Task Configurations

Start and Optimal Goals	Joint Configurations [rad]
Task Starting Location	$[0, 0, \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, 0, 0]$
Manipulability	$[0, -1.8, -2.8, -1.3, -2.3, 1.5, 0]$
Fault-Tolerant Manipulability	$[0, -0.8, 3.0, 1.2, -2.9, 1.6, 0]$
Largest Self-motion Manifold	$[0, -\frac{\pi}{2}, \frac{\pi}{2}, -2.69, 1.14, \frac{\pi}{2}, 0]$

locations with globally optimal local measures, i.e., manipulability, Fig. 3.7(b), and fault-tolerant manipulability, Fig. 3.7(c). The joint values for all these configurations are given in Table 3.3.



**Figure 3.8:** The PA-10 robot is shown with joint six (in red) failed at  $\theta_6 = 0$  while trying to perform a task where the goal location is either optimal manipulability in (a) or fault-tolerant manipulability in (b). An arc-welding tool is shown in the desired location to show the difference in position from the closest possible location for the damaged robots. The position errors in (a) and (b) are  $[-0.66, -0.44, 0.02]m$  and  $[-0.56, -0.14, -0.79]m$ , respectively.

We then simulate a joint failure in each of the joints and perform inverse kinematics on the remaining six working joints and attempt to reach the goal locations. In all cases, the robot is able to reach location Fig. 3.7(d), however, there are several joint failures that prevent the robot from reaching the goal locations shown in Fig. 3.7(b) and (c). One example is shown in Fig. 3.8, where

<sup>6</sup>Fig. 3.7 and Fig. 3.8 were generated by using the Workspace 5 software package from WAT Solutions, ([www.watsolutions.com](http://www.watsolutions.com)).

the robot is not able to reach the desired goal location due to a failure of joint six at  $\theta_6 = 0$  that occurred at the start location. In both cases, this joint failure results in the desired goal location being outside the workspace of the damaged robots. Therefore, the best they can do is get to the closest location that is at the boundary of their new workspace. In Fig. 3.8 we select the “closest” configuration by minimizing the orientation error, so that all error is in the position of the tool. This illustrates the merit of operating a robot on the largest self-motion manifold for mission-critical tasks.

### 3.5 CONCLUSIONS

This work uses a measure of fault tolerance that is based on the size of self-motion manifolds. Because singularities occur at the connection of self-motion manifolds, they can be used to identify where larger manifolds exist. We developed algorithms that use this fact to; (1) first identify all ranks of singularities and then, (2) search in the proximity of these singularities to identify large self-motion manifolds. A unique feature of Algorithm 1 is that it can efficiently identify high-rank singularities for arbitrary robot structures. To do this it must track multiple singular values that are nearly equal, where their gradients are not well defined. Algorithm 2 also must deal with the ill-conditioned nature of singular vectors that occur at singular configurations. The efficacy of these algorithms is illustrated on a commonly occurring 7 DoF kinematic structure (Mistubishi PA-10). In addition to identifying the largest self-motion manifold, it provided information that allows one to modify the kinematics to obtain an even larger manifold. It was also able to identify joints that are fault intolerant, so that one could explore alternate designs.

## Chapter 4

# An Algorithm to Design Redundant Manipulators of Optimally Fault-Tolerant Kinematic Structure<sup>10</sup>

One measure of the global fault tolerance of a redundant robot is the size of its self-motion manifold. If this size is defined as the range of its joint angles, then the optimal self-motion manifold size for an  $n$ -degree-of-freedom (DoF) robot is  $n \times 2\pi$ , which is not typical for existing robot designs. This paper presents a novel two-step algorithm to optimize the kinematic structure of a redundant manipulator to have an optimal self-motion manifold size. The algorithm exploits the fact that singularities occur on large self-motion manifolds by optimizing the robots kinematic parameters around a singularity. Because a gradient for the self-motion manifold size does not exist, the kinematic parameter optimization uses a coordinate-ascent procedure. The algorithm was used to design 4-DoF, 7-DoF, and 8-DoF manipulators to illustrate its efficacy at generating optimally fault-tolerant robots of any kinematic structure.

### 4.1 INTRODUCTION

Fault tolerance has been very important to the design and operation of manipulators for mission critical applications, where maintenance and repair are not feasible and a failure could result in a catastrophe. Failures are less likely for robots used in controlled environments where maintenance is relatively easy. However, reliability is critical for robots used in search and rescue operations [1]. Previous work has shown that the robot failure rates in severe environments are high [40] [41] and robot availability is as low as 50% [27]. To address some of these issues, researchers have studied the fault-tolerant control of actuators, e.g., in automated underwater vehicles [30] [31]. Also, Fault-tolerant control for multirobot systems with undetected failures was discussed in [32]. Because an

---

<sup>10</sup>This chapter was published in [39]

entire critical mission can be jeopardized due to an unreparable failure [2], redesigning robots to make them more fault tolerant is an important area of research [28].

Several aspects of robot fault-tolerance have been considered, such as fault detection, identification, and analysis, as surveyed in [29]. The most commonly occurring failure modes are the locked-joint failure [42], which will be considered in this work, and the free-swinging joint failure [43]. The latter mode is often transformed into the locked-joint mode by activation of fail-safe brakes [44]. Failure tolerance, necessarily, requires some level of redundancy. It can be achieved by duplicating parts that are more likely to fail (structural redundancy) [3], by human intervention to assess and overcome faults (functional redundancy), by analyzing working sensors to recover lost sensor information, e.g., integrating a tachometer signal to recover position (analytical redundancy) [29], or by designing a robot with more degree of freedom (DoFs) than the minimum needed to execute a certain task (kinematic redundancy). In this paper we focus on kinematic redundancy.

Quantifying measures of fault-tolerance for kinematically redundant robots has been extensively studied, focusing on two types, i.e., local and global measures. The local fault-tolerance measures are commonly based on the singular value decomposition (SVD) of the Jacobian matrix of a failed robot. These measures include the minimum singular value [4], the robot manipulability [6], and the condition number [5]. These local properties can be optimized by utilizing the kinematic redundancy. For example, the gradient of a singular value can be used to reconfigure a robot to satisfy a desired local fault-tolerance measure [23]. In addition, the local measures are widely used to design and control kinematically redundant fault-tolerant robots [45–47].

Global fault-tolerance measures typically quantify the fault-tolerant workspace. This makes these measures more suitable for pick-and-place tasks. For these types of tasks, a global measure can be used to identify the optimal fault-tolerant workspace locations [4], i.e., locations that are reachable both before and after a failure occurs. This is assured by limiting the robot to operate within software-imposed joint limits that are determined from the boundaries of the robot's self-motion manifold.

Designing a fault-tolerant workspace that is reachable for any trajectory both pre- and post-failure is more difficult. A procedure for computing the boundaries of the fault-tolerant workspace was presented in [33]. The fault-tolerant workspace can be maximized by determining the optimal artificial joint limits for a robot. This has been done by employing the gradient of the fault-tolerant workspace size as a function of the joint limits [48]. These global properties can be used to assess and optimize kinematic parameters of a robot by modifying its structure to improve its fault tolerance [34].

In our previous work [26], it was shown how one can determine the most fault-tolerant location for a given robot by identifying its largest self-motion manifold, where the manifold size depends on the ranges of all joint angles. In this work, we suggest a systematic procedure to generate robots of optimal kinematic structure by maximizing their self-motion manifold size. Because the largest self-motion manifold of a given manipulator is usually not optimal, i.e., not all joints span a  $2\pi$  range on the manifold, we present an efficient algorithm to optimize the kinematic structure of a robot to achieve the theoretically optimal self-motion manifold size.

The rest of the paper is organized in the following manner. A definition of a global fault tolerance suitable for a wide range of robotic applications is described in Section II. In Section III, we present a new optimization algorithm that can design an optimal kinematically fault-tolerant robot from any given baseline robot. In the following section, we illustrate the results of applying our algorithm on a 4-DoF, a 7-DoF, and an 8-DoF robot. Finally, we present our conclusions in Section V.

## 4.2 BACKGROUND ON SELF-MOTION MANIFOLDS

11

### 4.2.1 Preliminaries

Generally, the forward kinematics of a robot is a function of its joint angles

---

<sup>11</sup>Much of the material in this section is adapted from [26] and is included here for completeness.

$$\mathbf{x} = f(\boldsymbol{\theta}) \quad (4.1)$$

where  $\mathbf{x}$  is an  $m$ -dimensional vector representing the end-effector location and  $\boldsymbol{\theta}$  is an  $n$ -dimensional vector representing the joint angles. The inverse kinematics of a robot can be represented as

$$\boldsymbol{\theta} = f^{-1}(\mathbf{x}). \quad (4.2)$$

For non-redundant manipulators, the inverse-kinematic solution is a limited number of distinct solutions, but for redundant robots, i.e.,  $n > m$ , where  $n - m$  is the degree of redundancy, the inverse-kinematic solution for a certain desired end-effector location  $\mathbf{x}_d$  can be a number of continuous sets of solutions in the joint space. Each continuous set of solutions is a self-motion manifold. The upper bound on the number of the self-motion manifolds for redundant spherical, positional, and spatial manipulators is 2, 4, and 16, respectively [35]. At the velocity level, the forward kinematics of a robot can be rewritten as

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\boldsymbol{\theta}} \quad (4.3)$$

where  $\dot{\mathbf{x}}$  is the end-effector velocity,  $\mathbf{J}$  is the  $m \times n$  robot Jacobian matrix, and  $\dot{\boldsymbol{\theta}}$  represents the joint velocities.

The singular value decomposition (SVD) of  $\mathbf{J}$  can be written as

$$\mathbf{J} = \sum_{i=1}^r \sigma_i \hat{\mathbf{u}}_i \hat{\mathbf{v}}_i^{\top} \quad (4.4)$$

where  $\sigma_i$ 's are the ordered singular values, i.e.,  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{r-1} \geq \sigma_r \geq \dots \geq \sigma_m \geq 0$ , the vectors  $\hat{\mathbf{u}}_i$  and  $\hat{\mathbf{v}}_i$  represent the output and input singular vectors, respectively, and  $r$  denotes the rank of  $\mathbf{J}$ , where  $r < m$  for a singular robot ( $\sigma_i = 0$  for  $i > r$ ). The self-motion manifold(s) of a robot can be computed by solving

$$\mathbf{J}\dot{\boldsymbol{\theta}} = \mathbf{0} \quad (4.5)$$

for all possible  $\dot{\theta}$  values. We are not interested in the trivial solution  $\dot{\theta} = \mathbf{0}$ .

Typically, redundant manipulators have  $(n - m)$ -dimensional self-motion manifolds.<sup>12</sup> At singular configurations where self-motion manifolds connect, the number of manifolds that are connecting is one greater than the rank of the singularity. In the case that  $n - m = 1$ , the one-dimensional self-motion manifolds associated with a desired end-effector location  $\mathbf{x}_d$  can be computed by solving

$$\Delta\theta = \gamma\hat{\mathbf{v}}_n + \mathbf{J}^+\Delta\mathbf{x}_e \quad (4.6)$$

where  $\Delta\theta$  is the change in the joint angles,  $\gamma$  is a real positive scalar that represents the step size along the manifold,  $\hat{\mathbf{v}}_n$  is the  $n^{th}$  input singular vector that represents the one-dimensional null vector of the robot's Jacobian, and  $\mathbf{J}^+\Delta\mathbf{x}_e$  is an error correction term where  $\mathbf{J}^+$  is the pseudoinverse of the Jacobian matrix<sup>13</sup> and  $\Delta\mathbf{x}_e$  is the end-effector error, i.e., the difference between  $f(\theta)$  and the desired location,  $\mathbf{x}_d$ .

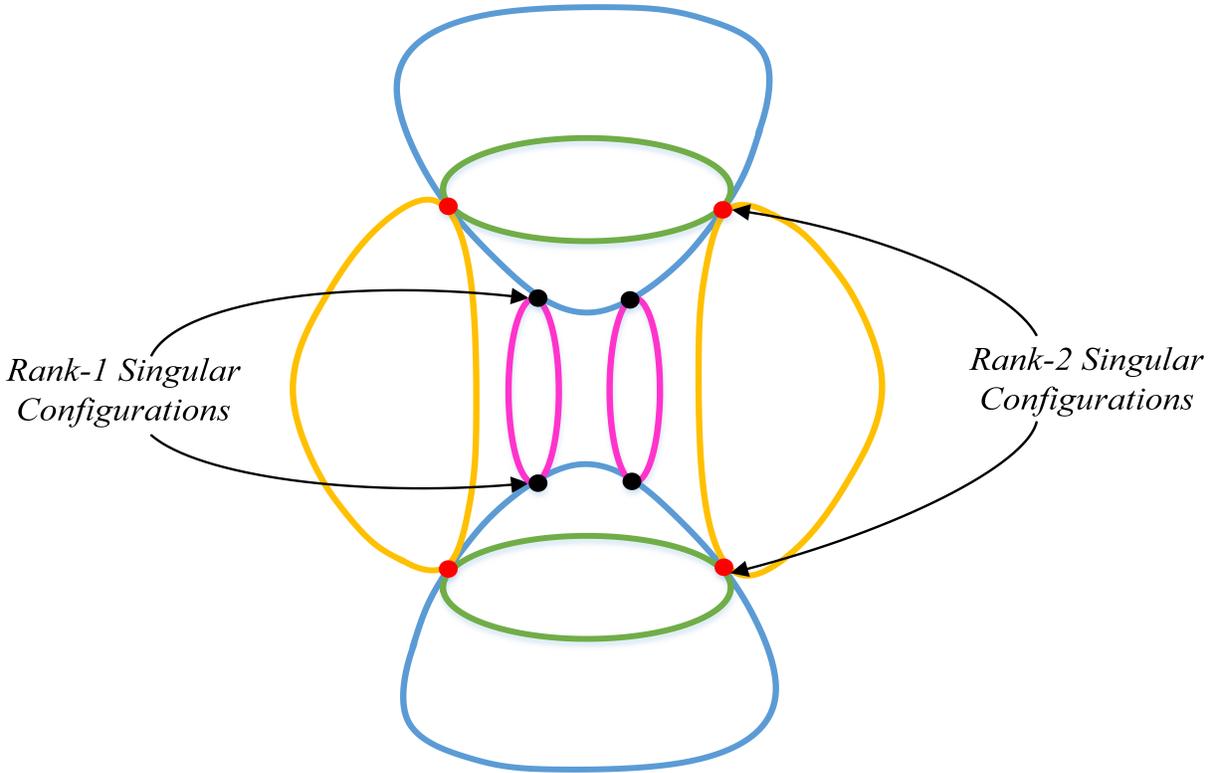
The topology of self-motion manifolds of high-DoF redundant robots can be very complicated. That is not only because visualizing all dimensions of the manifold(s) on one plot is infeasible, but also because it is common for a self-motion manifold to contain multiple singularities of different ranks. In Fig. 4.1, we show an abstract sketch of an actual one-dimensional self-motion manifold for a 7-DoF robot. This is considered one manifold because it is one set of continuous joint-space solutions for a specific workspace location. One can note that singularities (indicated in black and red for rank-1 and rank-2 singularities, respectively) are where the one-dimensional self-motion manifold branches out in one or more additional directions. The topology is more complicated for multi-dimensional self-motion manifolds. For instance, one can observe that for an 8-DoF robot, the self-motion manifold of a typical end-effector location is two-dimensional. This means that a

---

<sup>12</sup>At singular configurations (singularities) associated with workspace boundaries, a self-motion manifold may be of a lower dimension.

<sup>13</sup>Rather than exactly computing  $\mathbf{J}^+$ , we employ the Damped Least Squares (DLS) technique to efficiently compute the inverse kinematics in a numerically stable manner. DLS is able to deal with singular configurations as well as the ill-conditioned transition between singular and non-singular configurations [?].

rank one singularity is now a line, not a point. The topology is even more complicated for higher degree-of-redundancy manipulators.



**Figure 4.1:** This sketch illustrates the topology of a one-dimensional self-motion manifold comprised of multiple previously disjoint manifolds (for the Mitsubishi PA-10 7-DoF robot). The connection points between the previously disjoint manifolds are the singular configurations. One can observe that rank-1 singularities (in black) occur when two one-dimensional manifolds touch, while Rank-2 singularities (in red) occur when three one-dimensional manifolds touch. Note that visualizing the actual self-motion manifolds is challenging because there is typically no one projection that can show all self-motion manifolds distinctly.

For redundant manipulators where  $n - m > 1$ , a self-motion manifold associated with a typical end-effector location is  $(n - m)$ -dimensional, likewise, the null space of the Jacobian matrix is  $(n - m)$ -dimensional. In these cases, computing a self-motion manifold is challenging. Fortunately, one can efficiently find an estimate of a bounding box on the ranges of all joint angles. This estimate can be found by modifying (4.6) to

$$\Delta\theta = \gamma N_J \hat{e}_i + J^+ \Delta x_e \quad (4.7)$$

where  $N_J$  is a projection operator onto the  $(n - m)$ -dimensional null space of the Jacobian matrix and  $\hat{e}_i$  is a basis vector along the  $i^{th}$  joint angle, where  $1 \leq i \leq n$  [4]. The null space projection is given by

$$N_J = \sum_{i=r+1}^n \hat{v}_i \hat{v}_i^\top. \quad (4.8)$$

One can approximate the joint-angle ranges at a specific task location by iteratively solving (4.7) for  $i = 1$  to  $n$  with  $\pm \hat{e}_i$ . These joint ranges can be used to compute the size of the self-motion manifold. One can terminate the iterations while solving (4.7) when either a joint  $i$  spans a  $2\pi$  range or the projection of  $\hat{e}_i$  onto the null space becomes zero. In this case, this could be a local minimum so that this measure is a lower bound on the range of joint  $i$ .

#### 4.2.2 Size of Self-motion Manifolds

The size of a self-motion manifold can be measured differently for different applications. To distinguish between self-motion manifolds of different shapes and sizes, [?] suggested plotting the angular distance along a self-motion manifold versus the angular distance from the origin of the manifold. However, this is not practical for manifolds with complicated topologies as well as for multi-dimensional self-motion manifolds. By iteratively solving either (4.6) or (4.7) for a specific task location, one can determine the ranges of all joint angles over the manifolds. These ranges represent a bounding box, where its volume can be used as a global fault-tolerance measure [4] for that location. Because it is not uncommon for some robot joints to have a zero range over a self-motion manifold, we compute the self-motion manifold size by summing up the ranges of all joint angles over that manifold. In general, a redundant robot has multiple self-motion manifolds associated with a workspace location. Therefore, one needs to consider the union of all angle ranges on these manifolds. At an optimally fault tolerant task location the self-motion manifold size for an  $n$ -DoF robot must be  $n2\pi$ , i.e., each joint spans a  $2\pi$  range.

## 4.3 GENERATING OPTIMALLY FAULT TOLERANT ROBOT DESIGNS

As described above, large self-motion manifolds occur at singular configurations because previously disjoint manifolds are combined.<sup>14</sup> We exploit this observation to develop a procedure for identifying robot kinematic designs that have optimally fault tolerant self-motion manifolds. Generally, this is an iterative procedure where we first drive the robot to a singularity, evaluate its self motion manifold, and then adjust the kinematic parameters, i.e., Denavit and Hartenberg (DH) parameters, to increase the size of the manifold.

At a singularity, one or more singular values of the robot's Jacobian matrix,  $\mathbf{J}$ , are zero. Therefore, to drive a robot into a singularity we employ a gradient-descent technique on a singular value to find the robot's singular joint configuration. From (??), one can express any  $\sigma_i$  as

$$\sigma_i = \hat{\mathbf{u}}_i^\top \mathbf{J} \hat{\mathbf{v}}_i. \quad (4.9)$$

By differentiating (4.9) with respect to time, one obtains

$$\dot{\sigma}_i = \dot{\hat{\mathbf{u}}}_i^\top \mathbf{J} \hat{\mathbf{v}}_i + \hat{\mathbf{u}}_i^\top \dot{\mathbf{J}} \hat{\mathbf{v}}_i + \hat{\mathbf{u}}_i^\top \mathbf{J} \dot{\hat{\mathbf{v}}}_i \quad (4.10)$$

where the first and the third terms vanish due to the fact that the first order change in a singular vector is orthogonal to the vector itself. Thus, (4.10) can be further simplified to

$$\dot{\sigma}_i = \hat{\mathbf{u}}_i^\top \dot{\mathbf{J}} \hat{\mathbf{v}}_i. \quad (4.11)$$

The partial derivative of  $\sigma_i$  with respect to any joint angle  $\theta_k$  can be written as

$$\frac{\partial \sigma_i}{\partial \theta_k} = \hat{\mathbf{u}}_i^\top \frac{\partial \mathbf{J}}{\partial \theta_k} \hat{\mathbf{v}}_i \quad (4.12)$$

---

<sup>14</sup>Typically, self-motion manifolds with internal singular configurations will be larger than average. Manifolds that include reach singularities will be smaller than average

where

$$\frac{\partial \mathbf{J}}{\partial \theta_k} = \left[ \frac{\partial \mathbf{j}_1}{\partial \theta_k}, \frac{\partial \mathbf{j}_2}{\partial \theta_k}, \dots, \frac{\partial \mathbf{j}_n}{\partial \theta_k} \right]. \quad (4.13)$$

The partial derivative of the  $i^{\text{th}}$  column of the Jacobian matrix is given by [23], [22]

$$\frac{\partial \mathbf{j}_i}{\partial \theta_k} = \begin{cases} \begin{bmatrix} (\mathbf{z}_k^\top \mathbf{p}_i) \mathbf{z}_i - (\mathbf{z}_k^\top \mathbf{z}_i) \mathbf{p}_k \\ \mathbf{z}_k \times \mathbf{z}_i \end{bmatrix}, & k < i \\ \begin{bmatrix} (\mathbf{z}_i^\top \mathbf{p}_k) \mathbf{z}_k - (\mathbf{z}_i^\top \mathbf{z}_i) \mathbf{p}_k \\ \mathbf{0} \end{bmatrix}, & k \geq i \end{cases} \quad (4.14)$$

where  $\mathbf{z}_i$  is the axis of rotation of the  $i^{\text{th}}$  robot joint and  $\mathbf{p}_i$  is the position vector from the joint origin to the end-effector. Using (4.12), (4.13), and (4.14) one can compute the gradient of any  $\sigma_i$  of the Jacobian matrix as

$$\nabla \sigma_i = \left[ \frac{\partial \sigma_i}{\partial \theta_1}, \frac{\partial \sigma_i}{\partial \theta_2}, \dots, \frac{\partial \sigma_i}{\partial \theta_n} \right]. \quad (4.15)$$

Moving in the negative direction of this gradient allows one to decrease any desired singular value.

Maximizing a self-motion manifold's size of a robot can be done by employing a two-step iterative procedure. The first step is to drive the robot to a singular configuration from a random starting configuration  $\boldsymbol{\theta}^{(0)}$ . In this step, one can use the gradient descent of a singular value  $\sigma_i$  until it reaches zero, i.e.,

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \delta_k \nabla \sigma_i \quad (4.16)$$

where  $\boldsymbol{\theta}^{(k)}$  is the current joint configuration,  $\boldsymbol{\theta}^{(k+1)}$  is the next joint configuration, and  $\delta_k$  is an adaptive step size. One can iteratively solve (4.16) to guarantee that the robot converges to a desired singularity. Once the robot is at a singularity, we compute the size of the self-motion manifolds for the current end-effector location.

The second step is to optimize the robot's kinematic structure so that the self-motion manifold size is maximized to an optimal value. A gradient for the self-motion manifold size does not exist,

however, one can employ a gradient-free optimization technique. The optimization in this step can be formulated as

$$\mathbf{p}^* = \arg \max_{\mathbf{p} \in \mathbf{R}^l} f(\mathbf{p}) \quad (4.17)$$

where  $\mathbf{p}$  is an  $l$ -dimensional vector of the robot's DH parameters, except for the joint angles, namely the link lengths ( $\mathbf{a}$ ), the link displacements ( $\mathbf{d}$ ), and the link twists ( $\boldsymbol{\alpha}$ ), where  $\mathbf{p} = [\mathbf{a}^\top : \mathbf{d}^\top : \boldsymbol{\alpha}^\top]^\top$  and  $l = 3n$ , where  $n$  is the number of DoF. The function  $f(\mathbf{p})$  is a nonlinear function representing the self-motion manifold size. In this step, one can perform a *coordinate-ascent* procedure along all the DH parameters that are subject to optimization.

The coordinate ascent can be done by sequentially changing (increasing or decreasing) a DH parameter,  $\mathbf{p}_i$ . Once a  $\mathbf{p}_i$  value is slightly changed, the robot needs to be driven back to a singularity using (4.16) and then the change in the self-motion manifold size is evaluated. If the size increases, one can keep updating the same  $\mathbf{p}_i$ . If the size decreases or does not change, one should step back by retrieving the last good  $\mathbf{p}_i$  value and start changing the next DH parameter,  $\mathbf{p}_{i+1}$ . This process can be formulated as

$$\mathbf{p}_i^{(k+1)} = \mathbf{p}_i^{(k)} + \beta_k \mathbf{p}_i^{(k)} \quad (4.18)$$

where  $\mathbf{p}_i^{(k+1)}$  is the next value of a DH parameter,  $\mathbf{p}_i^{(k)}$  is the current value of the DH parameter, and  $\beta^{(k)}$  is a user-defined step size, where  $\beta_k$  can be positive or negative. These two steps should be performed alternately on all DH parameters until one either obtains an optimal robot, or a sweep of all elements of  $\mathbf{p}$  results in no improvement of  $f(\mathbf{p})$ . One should be aware that there could be a local maxima where there is no change to any DH parameter that will improve the self-motion manifold size. The pseudocode to implement this procedure is given in Algorithm 3.

There are several comments that should be pointed out about the behavior of this algorithm. First, it is important to note that the rank of the singularity being used can affect the kinematic structure of the resulting optimal robot. This means that optimizing around a rank-1 singularity could generate completely different optimal robots from the ones generated by optimizing around

---

**Algorithm 3** Generate Kinematically Fault-tolerant Robots

---

```
1: start with DH parameters of a baseline  $n$ -joint redundant robot.  $\{\mathbf{p} = [\mathbf{a} : \mathbf{d} : \boldsymbol{\alpha}]\}$ 
2: initialize  $\boldsymbol{\theta}^{(0)}$  {random joint-space configuration}
3: find  $\boldsymbol{\theta}^{(sing)}$  {drive the robot to a singularity}
4: compute  $\mathbf{S}_{init}$  {initial self-motion manifold size}
5:  $\mathbf{S}_{large} = \mathbf{S}_{init}$  {save initial self-motion manifold size}
6: for  $i = 1$  to  $n \times 3$  do {for DH parameters:  $[\mathbf{a} : \mathbf{d} : \boldsymbol{\alpha}]$ }
7:    $\mathbf{S}_{begin} = \mathbf{S}_{large}$  {save  $\mathbf{S}_{large}$  beginning of sweep}
8:    $\mathbf{dh}_{org} = \mathbf{p}_i$  {save  $\mathbf{p}_i$  in  $\mathbf{dh}_{org}$ }
9:    $\mathbf{p}_i^{(k+1)} = \mathbf{p}_i^{(k)} + \beta_k \mathbf{p}_i^{(k)}$  {perturb DH parameter  $\mathbf{p}_i$ }
10:  find  $\boldsymbol{\theta}^{(sing)}$  {drive the robot to a singularity}
11:  compute  $\mathbf{S}_{new}$  {the new self-motion manifold size}
12:  while  $\mathbf{S}_{new} > \mathbf{S}_{large}$  do
13:     $\mathbf{S}_{large} = \mathbf{S}_{new}$  {update  $\mathbf{S}_{large}$  with new value}
14:     $\mathbf{dh}_{org} = \mathbf{p}_i$  {save  $\mathbf{p}_i$  in  $\mathbf{dh}_{org}$ }
15:     $\mathbf{p}_i^{(k+1)} = \mathbf{p}_i^{(k)} + \beta_k \mathbf{p}_i^{(k)}$  {perturb parameter  $\mathbf{p}_i$ }
16:    find  $\boldsymbol{\theta}^{sing}$  {drive the robot to a singularity}
17:    compute  $\mathbf{S}_{new}$ 
18:    if  $(\mathbf{S}_{new} \approx n \times 2\pi)$  then {robot is optimal}
19:      save the optimal robot  $\{[\mathbf{a} : \mathbf{d} : \boldsymbol{\alpha} : \boldsymbol{\theta}^{(sing)}]\}$ 
20:    end if
21:  end while
22:   $\mathbf{p}_i = \mathbf{dh}_{org}$  {reset to the last good DH parameters}
23: end for
24: if  $(\mathbf{S}_{new} \leq \mathbf{S}_{begin})$  then {sweep didn't improve  $\mathbf{S}_{begin}$ }
25:   go to 2 {select different starting joint configurations}
26: else {do another sweep}
27:   go to 6
28: end if
```

---

a singularity of higher rank.<sup>15</sup> Obviously, different starting joint configurations,  $\boldsymbol{\theta}^{(0)}$ s, may result in different optimal robots. Furthermore, using the same starting configuration, singularity rank, and step size may have drastically different results for a positive step size versus a negative step size. Finally, the algorithm may converge to an optimal robot from the first coordinate-ascent sweep through the DH parameters, however, in some cases, it may take several sweeps through the

---

<sup>15</sup>Using the gradient descent of a singular value to drive a robot to a high-rank singularity may result in unwanted behavior when two or more singular values become nearly equal. In this case, a special procedure should be employed to guarantee efficient convergence [7].

DH parameters to converge. The results of applying the above procedure for generating optimal redundant robots are illustrated for 4-DoF, 7-DoF, and 8-DoF robots in the next section.

## 4.4 RESULTS

### 4.4.1 Four-DoF Robots

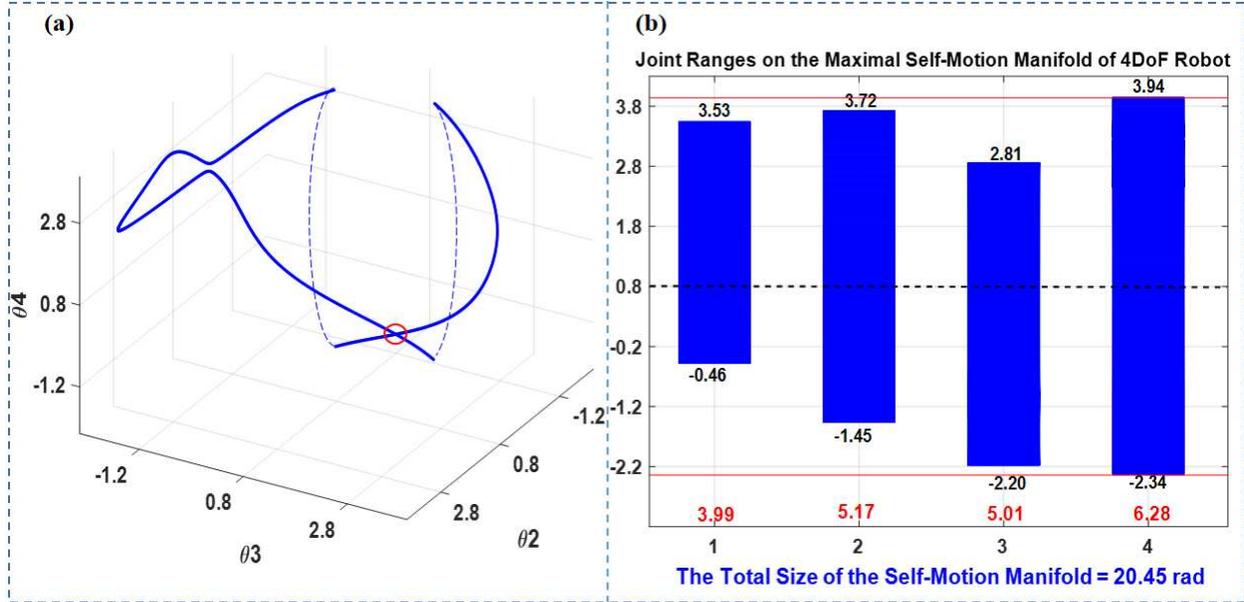
The algorithm is first used to generate optimally fault-tolerant 4-DoF spatial positioning robots. We start with a baseline robot that was designed to be globally optimal with respect to a local fault-tolerance measure [25]. The DH parameters of this robot are listed in Table 4.1.

**Table 4.1:** DH Parameters of the Baseline 4-DoF Robot

$Link_i$	$\alpha_i$ [degrees]	$a_i$ [meters]	$d_i$ [meters]	$\theta_i$ [degrees]
1	90	1.41	0	0
2	-90	1.41	1	180
3	90	1.41	-1	180
4	0	1.22	0.50	145

We first evaluated the global fault-tolerance of this robot, i.e., its largest self-motion manifold size. Fig. 4.2(a) shows the maximal self-motion manifold in  $\theta_2$ ,  $\theta_3$ , and  $\theta_4$  projection, where the singular configuration on the manifold is indicated in red. In subfigure (b), we show the angle ranges of the robot's joints on this self-motion manifold. It is clear that only  $\theta_4$  has an optimal range of  $2\pi$ . The rest of the joints  $\theta_1$ ,  $\theta_2$ , and  $\theta_3$  are not optimal.

By employing the optimization procedure explained in Section III, we were able to modify the kinematics of the baseline robot to generate many robots that have an optimal self-motion manifold at a particular workspace location. Because this is a 4-DoF robot, an optimal self-motion manifold size is 25.13 radians, i.e.,  $4 \times 2\pi$ . The DH parameters of an example optimal robot are listed in Table 4.2.



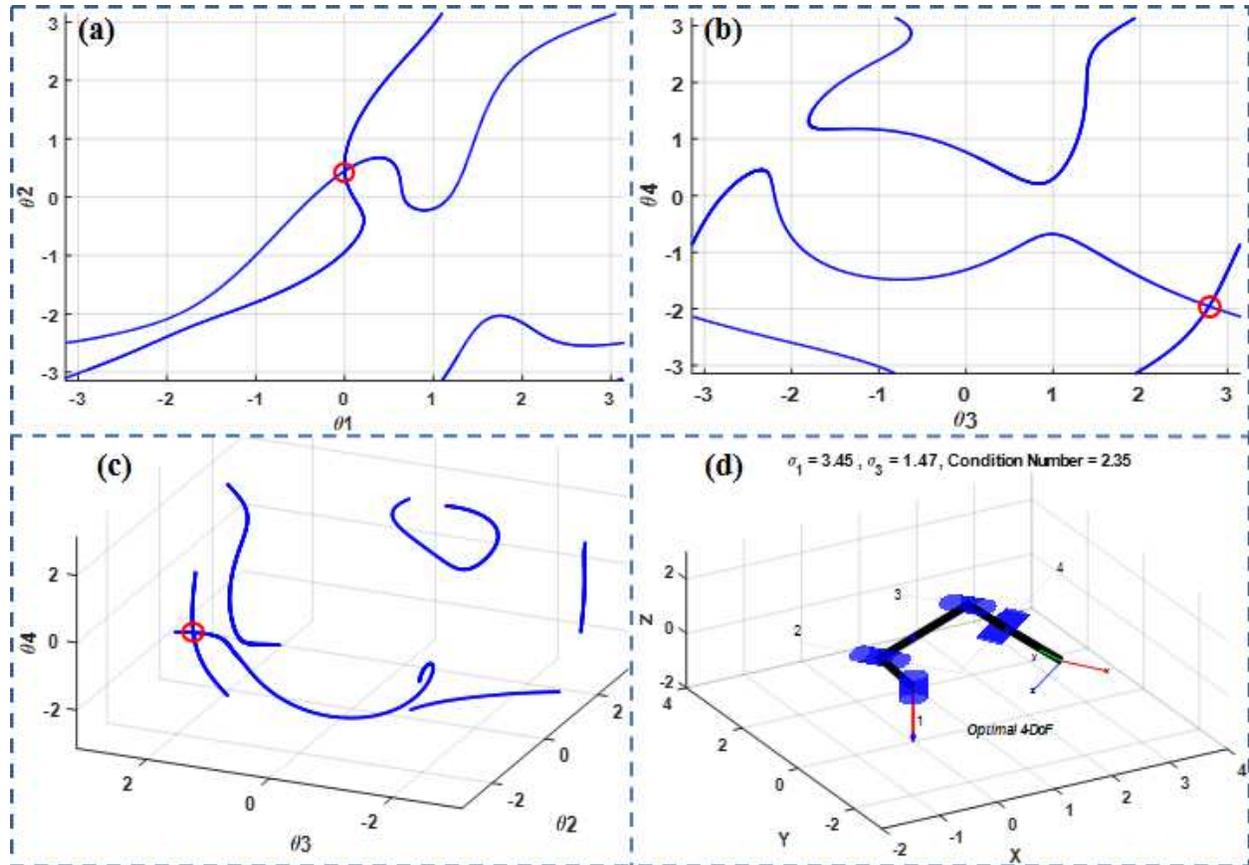
**Figure 4.2:** In this figure, the maximal self-motion manifold of the baseline 4-DoF robot is illustrated. In (a) this manifold is shown projected into the  $\theta_2$ ,  $\theta_3$ , and  $\theta_4$  subspace. From this projection one can see that this manifold contains only one rank-1 singularity, shown in red. Note that this robot has one continuous self-motion manifold where the dotted blue lines show the continuity of  $\theta_4$ . The ranges of each of the joints is shown in (b) where only  $\theta_4$  has a range of  $2\pi$ . Joints 1, 2, and 3 have ranges of 3.99, 5.17, and 5.01 radians, respectively, as indicated in red. The total size of this self-motion manifold is 20.45 radians.

**Table 4.2:** The DH Parameters of an Example Optimal 4-DoF Robot

$Link_i$	$\alpha_i$ [degrees]	$a_i$ [meters]	$d_i$ [meters]	$\theta_i$ [degrees]
1	130.52	1.41	0	0
2	-90	1.77	1	25.87
3	90	1.84	-1	159.78
4	0	1.52	0.55	-111.90

Comparing the optimal robot in Table 4.2 with the baseline robot in Table 4.1, one can observe that the algorithm has changed the first twist angle,  $\alpha_1$  from  $90^\circ$  to  $130.52^\circ$ . The other notable change was in the link lengths  $a_2$ ,  $a_3$ , and  $a_4$  as well as the last link displacement,  $d_4$ . Note that there are multiple possible values for  $\theta$ , i.e., any joint configuration on the self-motion manifold associated with this workspace location. The value shown in the table corresponds to the singula-

riety on this manifold that was identified by the algorithm. The robot at this workspace location has only one self-motion manifold, as shown in Fig. 4.3.



**Figure 4.3:** In this figure, the optimal 4-DoF robot and its optimal self-motion manifold are illustrated. Subfigures (a) and (b) show projections in  $\theta_1, \theta_2$  and  $\theta_3, \theta_4$ , respectively, where the manifold is continuous and all joints span a  $2\pi$  range. Thus, it is easy to see that the total size of the optimal self-motion manifold is 25.13 radians. In (c), the optimal self-motion manifold is presented in the same  $\theta_2, \theta_3$ , and  $\theta_4$  projection as Fig. 2(a). In (d), the robot is shown in a configuration on the self-motion manifold where a local dexterity measure, i.e., the condition number, is best ( $\sigma_1/\sigma_3 = 2.35$ ). The rank-1 singularity on the manifold is marked with a red circle in (a), (b), and (c).

It is easy to see from subfigures (a) and (b) that the range of all joints is  $2\pi$ . A rank-1 singularity that occurs on this self-motion manifold is indicated with a red circle. Subfigure (c) shows the same configuration-space projection as in Fig. 2(a) to illustrate how different the self-motion manifolds are for these two robots. Note that on this optimal self-motion manifold, one can elect to operate the manipulator in a configuration that optimizes an additional preferred dexterity measure, e.g.,

the condition number ( $\sigma_{max}/\sigma_{min}$ ). In subfigure (d), we show the robot in a configuration with the best condition number on this self-motion manifold, where  $\sigma_{max}/\sigma_{min} = 2.35$ .

#### 4.4.2 Seven-DoF Robots

To redundantly operate in a six-dimensional workspace consisting of both position and orientation, one needs a manipulator of at least 7 DoFs. A common 7-DoF redundant robot is the Mitsubishi PA-10, which has a kinematic design that is similar to the human arm. Unfortunately, because the arm has a three-joint spherical shoulder and a three-joint spherical wrist that are connected by a single rotational elbow joint, the PA-10 is fault intolerant with respect to the elbow, i.e., joint 4. Therefore, joint 4 has a zero range on the self-motion manifold(s) of any workspace location. We show that this kinematically fault-intolerant structure can be used as a baseline to generate optimal 7-DoF robots.

The DH parameters of the PA-10 robot are given in Table 4.3, with the last link displacement,  $d_7$  set equal to  $d_3$ . The robot at the joint configurations that are given in Table 4.4 has a maximal

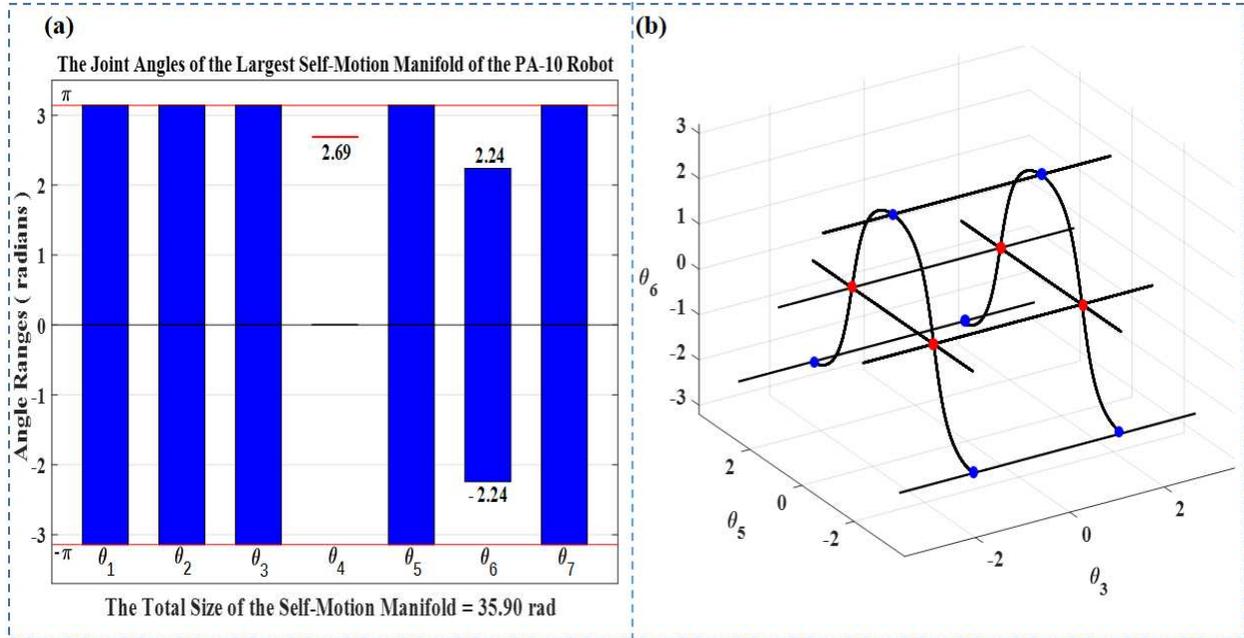
**Table 4.3:** The DH Parameters of the PA-10 Robot in Maximal Configurations

$Link_i$	$\alpha_i$ [degrees]	$a_i$ [meters]	$d_i$ [meters]	$\theta_i$ [degrees]
1	-90	0	0	0
2	90	0	0	0
3	-90	0	0.45	$\pm 90$
4	90	0	0	154.16
5	-90	0	0.5	$\pm 90$
6	90	0	0	0
7	0	0	0.45	0

self-motion manifold with size of 35.90 radians. The angles  $\theta_3$  and  $\theta_5$  can be  $\pm 90$  for the same end-effector location.<sup>16</sup> At this workspace location, the PA-10 has two large self-motion manifolds that are identical in terms of their joint ranges. Joint 4 has a range of zero on both self-motion manifolds with a constant value of  $\pm 154.16^\circ$  for up-elbow and down-elbow configurations, respectively. A

---

<sup>16</sup>For this workspace location, there are up-elbow and down-elbow configurations where the robot cannot move from one configuration to the other without changing the end-effector location.



**Figure 4.4:** The maximal self-motion manifold of the PA-10 robot is shown in this figure. In (a) the ranges of each of the joints is shown where  $\theta_1, \theta_2, \theta_3, \theta_5$ , and  $\theta_7$  are  $2\pi$ , the range of  $\theta_6$  is 4.48 radians ( $\pm 2.24$ ), and the range of  $\theta_4$  is zero. In (b) the one dimensional self-motion manifold is shown projected into the  $\theta_3, \theta_5$ , and  $\theta_6$  subspace. From this projection one can see that this manifold contains four rank-1 and four rank-2 singularities, shown in blue and red respectively. Note that the two blue singularities at the bottom of the figure, i.e., where  $\theta_6 = -2.24$  radians, are shown twice at both  $\theta_5 = \pm\pi$ . The rank-2 singularities occur when  $\theta = [0, 0, \frac{\pm\pi}{2}, \theta_4, \frac{\pm\pi}{2}, 0, 0]$ , where in this case  $\theta_4 = 2.69$  radians, ( $154.16^\circ$ ).

bar plot of the joint ranges is presented in Fig. 4.4(a).<sup>17</sup> A projection of one of the manifolds in  $\theta_3, \theta_5$ , and  $\theta_6$  space is presented in Fig. 4.4(b). It is clear from the figure that  $\theta_4$  and  $\theta_6$  have non-optimal ranges.

We employed our algorithm on this fault-intolerant baseline 7-DoF manipulator to generate robots with optimal kinematic structures that have self-motion manifolds of optimal size, i.e.,  $7 \times 2\pi = 43.98$  radians. In Table 4.4, we present the DH parameters of an example optimal 7-DoF robot. By comparing Table 4.3 and Table 4.4, one can observe that the algorithm has significantly changed the DH parameters to generate an optimal robot. The algorithm has modified all link lengths except  $a_2$  and  $a_3$ , displacements,  $d_3, d_4, d_6$ , and  $d_7$ , and twists  $\alpha_1, \alpha_4, \alpha_6$ , and  $\alpha_7$ .

<sup>17</sup>Fig. 4.4 appears in our previous work [26] where we identified the largest self-motion manifold of the PA-10 robot.

**Table 4.4:** The DH Parameters of an Example Optimal 7-DoF Robot

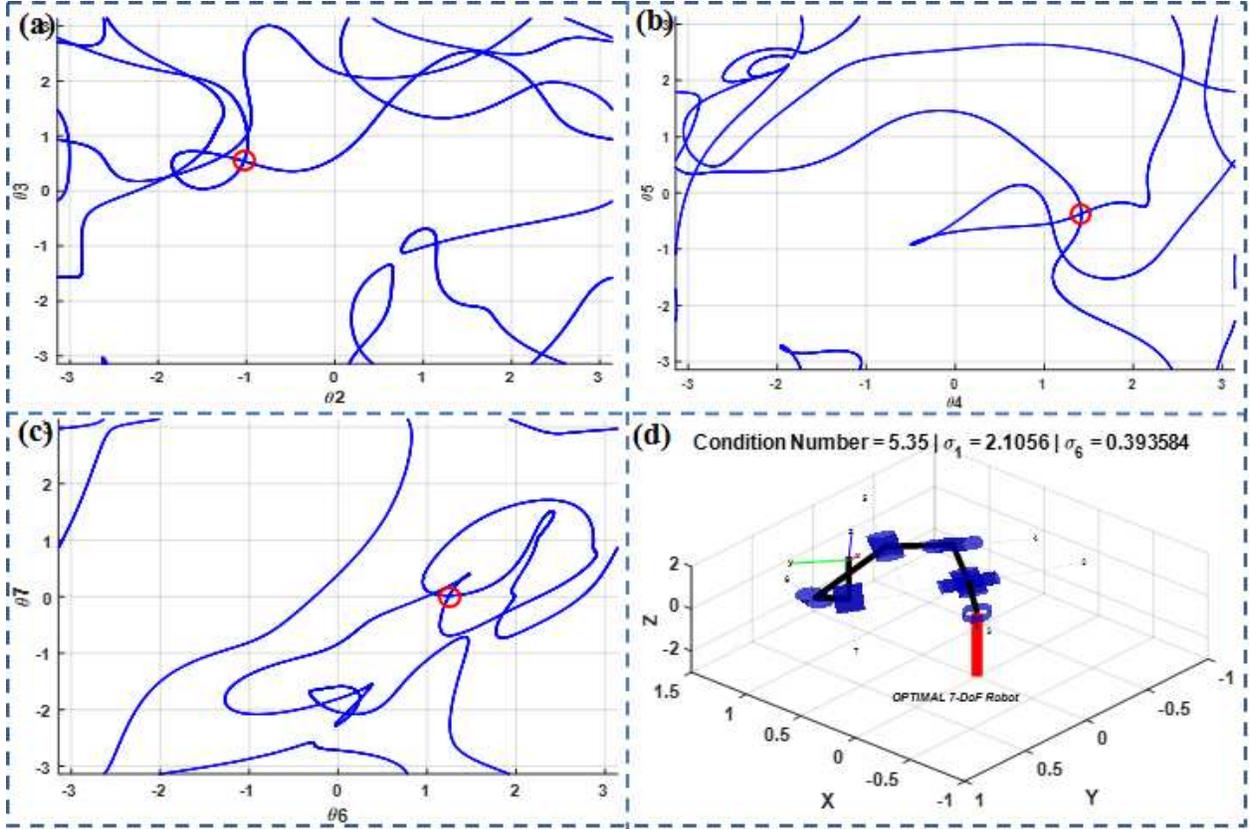
$Link_i$	$\alpha_i$ [degrees]	$a_i$ [meters]	$d_i$ [meters]	$\theta_i$ [degrees]
1	-99.24	0.43	0	0
2	90	0	0	-72.25
3	-90	0	0.54	36.81
4	117	0.26	0.32	81.10
5	-90	0.54	0.50	-21.11
6	118.14	0.18	0.16	71.50
7	7.22	0.16	0.50	1.01

In addition, the algorithm automatically generates singular joint configurations. In Fig. 4.5, we present the optimal self-motion manifold of this robot.

The fault-tolerant 7-DoF robot presented in Table 4.4 has only one continuous self-motion manifold at the optimal end-effector location. From Fig 4.5, one can note that all robot joints have a range of  $2\pi$ . This manifold only contains one singularity, marked with a red circle, that is of rank 1, while all other intersections that appear in the figure are due to the projections.

### 4.4.3 Eight-DoF Robots

To show the merit of using our algorithm to optimize the kinematic structure of high-DoF robots, we illustrate its efficacy on an 8-DoF manipulator. We arbitrarily used an 8-DoF robot that has a 3-joint shoulder, a 2-joint elbow, and a 3-joint wrist. The DH parameters of this baseline robot are given in Table 4.5.

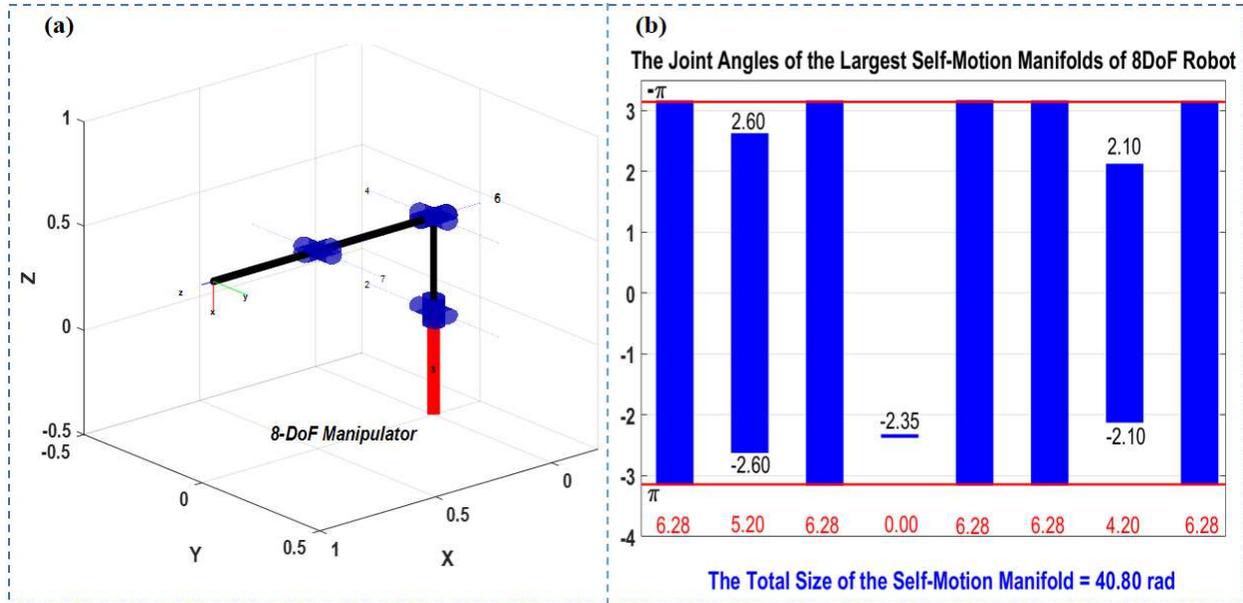


**Figure 4.5:** The optimal self-motion manifold of the fault-tolerant 7-DoF robot is shown in this figure. A projection in  $\theta_2, \theta_3$  is shown in (a),  $\theta_3, \theta_4$  in (b), and  $\theta_6, \theta_7$  in (c). The only singularity (which is of rank 1) on this manifold is indicated with a red circle. In (d) the optimal 7-DoF robot is presented in a configuration where its condition number is minimal, i.e., 5.35.

**Table 4.5:** The DH Parameters of the Baseline 8-DoF Robot

$Link_i$	$\alpha_i$ [degrees]	$a_i$ [meters]	$d_i$ [meters]	$\theta_i$ [degrees]
1	-90	0	0	85.34
2	90	0	0	0
3	-90	0	0.54	-90
4	90	0	0	-134.50
5	-90	0	0	134.78
6	90	0	0.50	-44.78
7	-90	0	0	0
8	0	0	0.45	-12.86

This eight DoF robot at this joint configuration has the maximal self-motion manifold size of 40.80 radians. The robot and its largest self-motion manifold are both shown in Fig. 4.6.

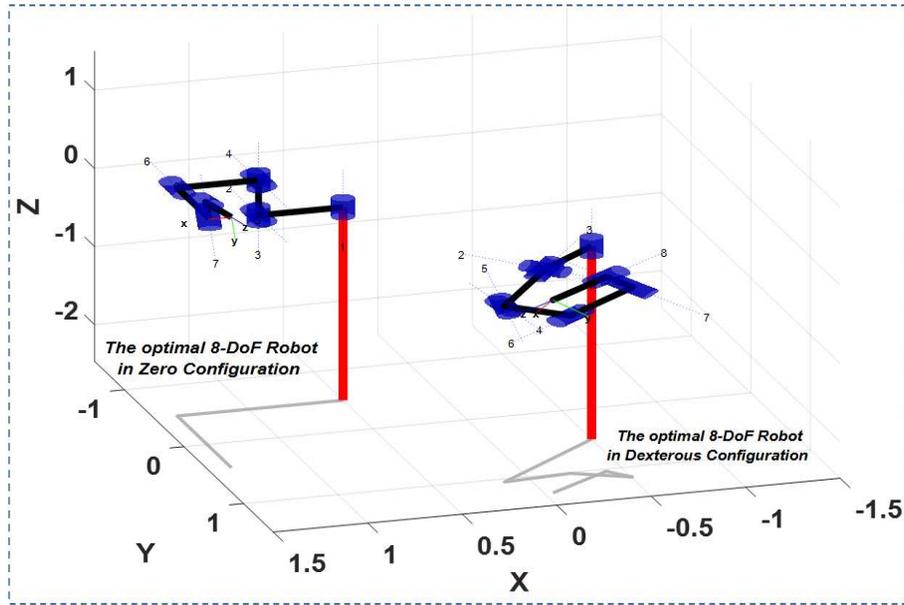


**Figure 4.6:** In (a) the 8-DoF robot is illustrated in a configuration where  $\theta = [0, 0, 0, \frac{\pi}{2}, 0, 0, 0, 0]$  to show its structure. The joint ranges on the largest self-motion manifold are shown as a bar plot in (b) where  $\theta_1, \theta_3, \theta_5, \theta_6,$  and  $\theta_8$  have a  $2\pi$  range. The range of  $\theta_2$  is 5.20 radians ( $\pm 2.60$ ), the range of  $\theta_7$  is 4.20 radians ( $\pm 2.10$ ), while  $\theta_4$  has a zero range with a fixed value at  $\theta_4 = -2.35$  radians.

One should note that in this case, the self-motion manifolds are typically two dimensional, but are of a higher dimension at singularities. In this case, computing a self-motion manifold is challenging, but one can employ (4.7) to determine an estimate of the ranges of all joints. The optimal self-motion manifold size is  $8 \times 2\pi = 50.27$  radians. In Table 4.6, we list an example 8-DoF optimal robot that was generated by employing Algorithm 3. This robot is shown in Fig. 4.7 in the zero configuration and in a relatively dexterous configuration where the condition number is 7.18. One can immediately observe that the algorithm has introduced a single change in link twist parameters by changing  $\alpha_6$  from 90 to 105.30 degrees. Also, two link lengths,  $a_1$  and  $a_5$  were introduced as well as other minor changes in the link displacement parameters.

**Table 4.6:** The DH Parameters of the Optimal 8-DoF Robot

$Link_i$	$\alpha_i$ [degrees]	$a_i$ [meters]	$d_i$ [meters]	$\theta_i$ [degrees]
1	-90	0.45	0	0
2	90	0	0	-83.60
3	-90	0	0.45	180
4	90	0	0	90
5	-90	0.43	0	-168
6	105.30	0	0.52	-90
7	-90	0	0.17	0
8	0	0	0.45	-29.05



**Figure 4.7:** The kinematic design of an optimally fault-tolerant 8-DoF robot is shown in this figure. The robot is presented in the zero configuration to illustrate how various joints are connected (left). The robot is shown in a dexterous configuration where the condition number of the Jacobian matrix is 7.18 (right).

#### 4.4.4 Discussion

Now that we have robots of different DoFs that are optimal in terms of their global fault-tolerance measure, i.e., their self-motion manifold size, we evaluate the quality of these self-motion manifolds with respect to a local dexterity measure. This is done by evaluating the condition number ( $\sigma_{max}/\sigma_{min}$ ) of the Jacobian matrix for configurations on the self-motion manifold and estimating how much of the manifold can meet a certain threshold. For example, for the 4-DoF robot in Fig. 4.3, over 90% of the self-motion manifolds have condition numbers less than 10, and

for the 7-DoF robot in Fig. 4.5 it is over 20%. This illustrates that robot designers have significant flexibility in satisfying multiple criteria, i.e., having a robot that is both optimally fault tolerant and meets pre-failure dexterity design specifications.

## 4.5 CONCLUSIONS AND FUTURE WORK

The goal of the work described here is to identify redundant robot kinematic designs that possess optimally fault tolerant locations of operation within their workspace. If the definition of fault tolerance is reachability of a task location after any arbitrary locked-joint failure occurs, then this corresponds to identifying robots that possess self-motion manifolds that span  $2\pi$  in every joint angle for that task location. We exploit the fact that such large self-motion manifolds are more likely when they contain singular configurations, because singularities occur when previously disjoint manifolds are connected. Our novel algorithm for identifying optimal kinematic designs alternates between driving the robot to a singular configuration and modifying the kinematic parameters by using a coordinate-ascent algorithm to increase the self-motion manifold size.

The premise in this work is that a robot designer already has a baseline robot kinematics in mind, and one would like to improve the fault tolerance of that design, without changing it radically. Remarkably, the algorithm was able to find multiple optimal designs from 4-DoF, 7-DoF, and 8-DoF baseline robots. In our future work, we will be investigating ways to evaluate and characterize all of these optimally fault-tolerant robots and determine whether they can also be optimized for additional desirable objectives. In addition, it is important to point out that the proposed algorithm results in an optimal fault tolerant configuration, which is appropriate for critical workspace locations involved in pick-and-place tasks. For other types of tasks, one may want to guarantee a specified high-level of fault tolerance over a given workspace volume.

# Chapter 5

## Conclusions and Future Works

### 5.1 CONCLUSIONS:

This work has presented a systematic approach to identify the largest self-motion manifold of a robot of any arbitrary kinematic structure. In part, the work suggested a new algorithm to identify the singularities of the a robot then find the optimal self-motion manifold. This algorithm is distinguished from previous research by its ability to identify singular configurations of a robot regardless of its kinematic structure, while previous works highly depend on the simplicity of the robot kinematics and fail to extend to higher DoF robots of general structure. Also, the presented algorithm, unlike other approaches, does not fail to identify high-rank singular configurations, where singularity conditions become very complicated. In this work, we use the sum of the range of the joint angles of a robot as a measure of the fault tolerance. Furthermore, this work employs Givens rotation to identify the physically meaningful singular directions that are not necessarily obvious due to the SVD being indistinct at singular configurations.

The second part of this work focuses on optimizing the kinematic structure of a given robot to maximize the largest self-motion manifold. In this part, we employed the coordinate-ascent technique to minimally change a given baseline robot to maximize its largest self-motion manifold size. The motivation behind using the coordinate-ascent technique is that the self-motion manifold size is not a differentiable function of the DH parameters. This technique allows changing one DH parameters at a time and observe the effect of such change on the size of the largest self-motion manifold. This optimization has been performed near a singular configuration where large self-motion manifolds are most likely to be found. Using this algorithm, we were able to identify several optimal robots of different degree of freedoms, i.e., 4-DoF, 7-DoF, and 8-DoF.

## 5.2 FUTURE PLAN:

After being able to find the largest self-motion manifold of a robot, designing an algorithm to design optimally fault-tolerant robots, and being able to find multiple optimal robots, we think that the natural questions needs to be addressed in future are:

1. Among all optimal robots from the same DoFs, is there a way to classify these robots into subsets/groups that share the same kinematic properties even if they are slightly different in terms of their structure? We think, it would be really interesting to know if such classes of optimal robots can be found.
2. Optimal robots of higher degree of redundancy, where their self-motion manifolds are or two (or more) dimensional, offer the ability to optimize other local fault-tolerance measures (like the condition number). We think of developing an algorithm that can have the ability to design optimal robots with respect to their global fault-tolerance measure, i.e., self-motion manifold size, and their local measure, i.e., the condition number on such manifold. In this case, we are also interested in visualizing these high-dimensional self-motion manifolds.

# Bibliography

- [1] Fumitoshi Matsuno and Satoshi Tadokoro. Rescue robots and systems in Japan. In *IEEE Int. Conf. on Robotics and Biomimetics*, pages 12–20, 2004.
- [2] Keiji Nagatani, Seiga Kiribayashi, Yoshito Okada, Kazuki Otake, Kazuya Yoshida, Satoshi Tadokoro, Takeshi Nishimura, Tomoaki Yoshida, Eiji Koyanagi, Mineo Fukushima, and Shinji Kawatsuma. Emergency response to the nuclear accident at the Fukushima Daiichi nuclear power plants using mobile rescue robots. *Journal of Field Robotics*, 30(1):44–63, 2013.
- [3] Vittorio Monteverde and Sabri Tosunoglu. Effect of kinematic structure and dual actuation on fault tolerance of robot manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 2902–2907, 1997.
- [4] Christopher L Lewis and Anthony A Maciejewski. Fault tolerant operation of kinematically redundant manipulators for locked joint failures. *IEEE Trans. on Robotics and Automation*, 13(4):622–629, 1997.
- [5] Hamid Abdi and Saeid Nahavandi. Minimum reconfiguration for fault tolerant manipulators. In *Proceedings of the 34th Annual Mechanisms and Robotics Conference, Parts A and B*, pages 1345–1350, 2010.
- [6] Rodney G Roberts and Anthony A Maciejewski. A local measure of fault tolerance for kinematically redundant manipulators. *IEEE Trans. on Robotics and Automation*, 12(4):543–552, 1996.
- [7] Ahmad Almarkhi and Anthony Maciejewski. Singularity analysis for redundant manipulators of arbitrary kinematic structures. In *International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, pages 42–49, 2019.

- [8] Daniel R Baker and Charles W Wampler. On the inverse kinematics of redundant manipulators. *The International Journal of Robotics Research*, 7(2):3–21, 1988.
- [9] Joel W Burdick. On the inverse kinematics of redundant manipulators: Characterization of the self-motion manifolds. *International Conference on Robotics and Automation*, 1(2):264–270, 1989.
- [10] Kenneth H Hunt. Special configurations of robot-arms via screw theory. *Robotica*, 4(3):171–179, 1986.
- [11] J Kieffer and J Lenarcic. On the exploitation of mechanical advantage near robot singularities. *Informatica*, 18(3):315–323, 1994.
- [12] Yoshihiko Nakamura and Hideo Hanafusa. Inverse kinematic solutions with singularity robustness for robot manipulator control. *Journal of Dynamic Systems, Measurement, and Control*, 108(3):163–171, 1986.
- [13] KJ Waldron, Shih-Liang Wang, and SJ Bolin. A study of the Jacobian matrix of serial manipulators. *Journal of Mechanisms, Transmissions, and Automation in Design*, 107(2):230–237, 1985.
- [14] FL Litvin, Zhang Yi, V Parenti Castelli, and C Innocenti. Singularities, configurations, and displacement functions for manipulators. *The International Journal of Robotics Research*, 5(2):52–65, 1986.
- [15] K Sugimoto, J Duffy, and KH Hunt. Special configurations of spatial mechanisms and robot arms. *Mechanism and Machine Theory*, 17(2):119–132, 1982.
- [16] Roger Boudreau and Ron P Podhorodeski. Singularity analysis of a kinematically simple class of 7-jointed revolute manipulators. *Transactions of the Canadian Society for Mechanical Engineering*, 34(1):105–117, 2010.

- [17] Scott B Nokleby and Ron P Podhorodeski. Identifying the 1-DOF-loss velocity-degenerate (singular) configurations of an 8-joint manipulator. *Transactions of the Canadian Society for Mechanical Engineering*, 28(2A):109–124, 2004.
- [18] Scott B Nokleby and Ron P Podhorodeski. Identifying multi-DOF-loss velocity degeneracies in kinematically-redundant manipulators. *Mechanism and Machine Theory*, 39(2):201–213, 2004.
- [19] Wenfu Xu, Jintao Zhang, Huihuan Qian, Yongquan Chen, and Yangsheng Xu. Identifying the singularity conditions of Canadarm2 based on elementary Jacobian transformation. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 795–800, 2013.
- [20] Wenfu Xu, Jintao Zhang, Bin Liang, and Bing Li. Singularity analysis and avoidance for robot manipulators with nonspherical wrists. *IEEE Transactions on Industrial Electronics*, 63(1):277–290, 2016.
- [21] Anthony Alexander Maciejewski. *The analysis and control of robotic manipulators operating at or near kinematically singular configurations*. PhD thesis, The Ohio State University, 1988.
- [22] Charles A Klein and Li-Chung Chu. Comparison of extended Jacobian and Lagrange multiplier based methods for resolving kinematic redundancy. *Journal of Intelligent & Robotic Systems*, 19(1):39–54, 1997.
- [23] Kenneth N Groom, Anthony A Maciejewski, and Venkataramanan Balakrishnan. Real-time failure-tolerant control of kinematically redundant manipulators. *IEEE Trans. on Robotics and Automation*, 15(6):1109–1115, 1999.
- [24] Peter Corke. *Robotics, vision and control: Fundamental algorithms in MATLAB®*, volume 118. Springer, 2017.

- [25] Khaled M Ben-Gharbia, Anthony A Maciejewski, and Rodney G Roberts. Kinematic design of redundant robotic manipulators for spatial positioning that are optimally fault tolerant. *IEEE Transactions on Robotics*, 29(5):1300–1307, 2013.
- [26] Ahmad A Almarkhi and Anthony A Maciejewski. Maximizing the size of self-motion manifolds to improve robot fault tolerance. *IEEE Robotics and Automation Letters*, 4(3):2653–2660, 2019.
- [27] Jennifer Carlson and Robin R Murphy. How UGVs physically fail in the field. *IEEE Trans. on Robotics*, 21(3):423–437, 2005.
- [28] Keiji Nagatani, Seiga Kiribayashi, Yoshito Okada, Satoshi Tadokoro, Takeshi Nishimura, Tomoaki Yoshida, Eiji Koyanagi, and Yasushi Hada. Redesign of rescue mobile robot Quince. In *IEEE Int. Symposium on Safety, Security, and Rescue Robotics*, pages 13–18, 2011.
- [29] Monica L Visinsky, Joseph R Cavallaro, and Ian D Walker. Robotic fault detection and fault tolerance: A survey. *Reliability Engineering & System Safety*, 46(2):139–158, 1994.
- [30] Tarun Kanti Podder, Gianluca Antonelli, and Nilanjan Sarkar. Fault tolerant control of an autonomous underwater vehicle under thruster redundancy: Simulations and experiments. In *IEEE Int. Conf. on Robotics and Automation*, pages 1251–1256, 2000.
- [31] N Ranganathan, Minesh I Patel, and R Sathyamurthy. An intelligent system for failure detection and control in an autonomous underwater vehicle. *IEEE Trans. on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 31(6):762–767, 2001.
- [32] Hyongju Park and Seth A Hutchinson. Fault-tolerant rendezvous of multirobot systems. *IEEE Trans. on Robotics*, 33(3):565–582, 2017.
- [33] Randy C Hoover, Rodney G Roberts, Anthony A Maciejewski, Priya S Naik, and Khaled M Ben-Gharbia. Designing a failure-tolerant workspace for kinematically redundant robots. *IEEE Trans. on Automation Science and Engineering*, 12(4):1421–1432, 2015.

- [34] Khaled M Ben-Gharbia, Anthony A Maciejewski, and Rodney G Roberts. Modifying the kinematic structure of an anthropomorphic arm to improve fault tolerance. In *IEEE Int. Conf. on Robotics and Automation*, pages 1455–1460, 2015.
- [35] Joel Wakeman Burdick. *Kinematic analysis and design of redundant robot manipulators*. PhD thesis, Department of Computer Science, Stanford University, 1988.
- [36] Scott B Nokleby and Ron P Podhorodeski. Reciprocity-based resolution of velocity degeneracies (singularities) for redundant manipulators. *Mechanism and Machine Theory*, 36(3):397–409, 2001.
- [37] J. Denavit and R. S. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *ASME Journal of Applied Mechanics*, 22(4):215–221, 1955.
- [38] Tsuneo Yoshikawa. Manipulability of robotic mechanisms. *The Int. Journal of Robotics Research*, 4(2):3–9, 1985.
- [39] A. A. Almarkhi, A. A. Maciejewski, and E. K. P. Chong. An algorithm to design redundant manipulators of optimally fault-tolerant kinematic structure. *IEEE Robotics and Automation Letters*, 5(3):4727–4734, 2020.
- [40] Shen Cheng and BS Dhillon. Reliability and availability analysis of a robot-safety system. *Journal of Quality in Maintenance Engineering*, 2011.
- [41] David Mahar, William Fields, John Reade, Peter Zarubin, and Scott McCombie. Nonelectronic parts reliability data. *Reliability Information Analysis Center*, 2011.
- [42] Yung Ting, Sabri Tosunoglu, and Benito Fernandez. Control algorithms for fault-tolerant robots. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 910–915. IEEE, 1994.

- [43] James D English and Anthony A Maciejewski. Fault tolerance for kinematically redundant manipulators: Anticipating free-swinging joint failures. *IEEE Transactions on Robotics and Automation*, 14(4):566–575, 1998.
- [44] Peetu Nieminen, Salvador Esque, Ali Muhammad, Jouni Mattila, Jukka Väyrynen, Mikko Siuko, and Matti Vilenius. Water hydraulic manipulator for fail safe and fault tolerant remote handling operations at iter. *Fusion Engineering and Design*, 84(7-11):1420–1424, 2009.
- [45] FL Hammond. Synthesis of  $k$  th order fault-tolerant kinematically redundant manipulator designs using relative kinematic isotropy. *International Journal of Adaptive and Innovative Systems*, 2(1):73–96, 2014.
- [46] Chinmay S Ukidve, John E McInroy, and Farhad Jafari. Using redundancy to optimize manipulability of stewart platforms. *IEEE/ASME Transactions on Mechatronics*, 13(4):475–479, 2008.
- [47] Yu She, Wenfu Xu, Haijun Su, Bin Liang, and Hongliang Shi. Fault-tolerant analysis and control of ssrms-type manipulators with single-joint failure. *Acta Astronautica*, 120:270–286, 2016.
- [48] Ashraf M Bader and Anthony A Maciejewski. Maximizing the failure-tolerant workspace area for planar redundant robots. *Mechanism and Machine Theory*, 143:103635, 2020.