DISSERTATION

KINEMATIC DESIGN AND MOTION PLANNING OF FAULT TOLERANT ROBOTS WITH LOCKED JOINT FAILURES

Submitted by
Biyun Xie
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
Summer 2019

Doctoral Committee:

Advisor: Anthony A. Maciejewski

Edwin K P Chong
Ali Pezeshki
Jianguo Zhao

ABSTRACT

KINEMATIC DESIGN AND MOTION PLANNING OF FAULT TOLERANT ROBOTS WITH LOCKED JOINT FAILURES

The problem of kinematic design and motion planning of fault tolerant robots with locked joint failure is studied in this work. In kinematic design, the problem of designing optimally fault tolerant robots for equal joint failure probabilities is first explored. A measure of local fault tolerance for equal joint failure probabilities has previously been defined based on the properties of the singular values of the Jacobian matrix. Based on this measure, one can determine a Jacobian that is optimal. Because these measures are solely based on the singular values of the Jacobian, permutation of the columns does not affect the optimality. Therefore, when one generates a kinematic robot design from this optimal Jacobian, there will be 7! robot designs with the same locally optimal fault tolerant property. This work shows how to analyze and organize the kinematic structure of these 7! designs in terms of their Denavit and Hartenberg (DH) parameters. Furthermore, global fault tolerant measures are defined in order to evaluate the different designs. It is shown that robot designs that are very similar in terms of DH parameters, e.g., robots generated from Jacobians where the columns are in reverse order, can have very different global properties. Finally, a computationally efficient approach to calculate the global pre- and post-failure dexterity measures is presented and used to identify two Pareto optimal robot designs. The workspaces for these optimal designs are also shown.

Then, the problem of designing optimally fault tolerant robots for different joint failure probabilities is considered. A measure of fault tolerance for different joint failure probabilities is defined based on the properties of the singular values of the Jacobian after failures. Using this measure, methods to design optimally fault tolerant robots for an arbitrary set of joint failure probabilities and multiple cases of joint failure probabilities are introduced separately. Given an arbitrary set of joint failure probabilities, the optimal null space that optimizes the fault tolerant measure is derived, and the associated isotropic Jacobians are constructed. The kinematic parameters of the optimally fault tolerant robots are then generated from these Jacobians. One special case, i.e., how to construct the optimal Jacobian of spatial 7R robots for both positioning and orienting is further discussed. For multiple cases of joint failure probabilities, the optimal robot is designed through optimizing the sum of the fault tolerant measures for all the possible joint failure probabilities. This technique is illustrated on planar 3R robots, and it is shown that there exists a family of optimal robots.

After the optimally fault tolerant robots are designed, the problem of planning the optimal trajectory with minimum probability of task failure for a set of point-to-point tasks, after experiencing locked joint failures, is studied. The proposed approach first develops a method to calculate the probability of task failure for an arbitrary trajectory, where the trajectory is divided into small segments, and the probability of task failure of each segment is calculated based on its failure scenarios. Then, a motion planning algorithm is proposed to find the optimal trajectory with minimum probability of task failure. There are two cases. The trajectory in the first case is the optimal trajectory from the start configuration to the intersection of the bounding boxes of all the task points. In the other case, all the configurations along the self-motion manifold of task point 1 need to be checked, and the optimal trajectory is the trajectory with minimum probability of task failure among them. The proposed approach is demonstrated on planar 2R redundant robots, illustrating the effectiveness of the algorithm.

ACKNOWLEDGEMENTS

Thank you, my advisor Prof. Maciejewski! First of all, thank you for accepting me as your PhD student four years ago, and providing me a precious opportunity to study in USA to improve my academic proficiency and experience the different culture, which is an important turning point in my life. During the four years of my PhD study, thank you for always being there to point out the right direction for my research, and provide valuable inspirations for solving the problems. Particularly, thank you for sitting together with me and revising my papers word by word. I have learned quite a lot about scientific writing from this process, such as how to classify references and describe them in a storytelling way, how to organize the paragraphs and sections in a logical way, how to write the captions of figures to help readers understand the figures without reading the paper, and so on. These writing experiences are invaluable for my future academic career. Most importantly, thank you for your generous help and support in my faculty position application, from editing the application materials, providing guidance for my on-campus interview to reviewing the contract. Finding an ideal faculty position is anther important turning point in my life. In these four years, you have taught me how to be a creative researcher, a helpful advisor and an exceptional professor. You gave me the most precious memories at CSU!

Thank you, my parents! You have always loved and supported me unconditionally. Although I have chosen a path that deviates from your expectations, you still respect my decisions, and feel proud of my accomplishments.

I would also like to sincerely thank my PhD committee members, Dr. Edwin K P Chong, Dr. Peter Young, Dr. Ali Pezeshki and Dr. Jianguo Zhao for all of their guidance through my PhD study. Their discussion, ideas, and feedback in my qualify exam, preliminary exam and defense have been absolutely invaluable.

DEDICATION

To my academic advisor, who gives me strength to reach for the stars and chase my dreams!

TABLE OF CONTENTS

[1] Most of this chapter is published in [1].

[2] Most of this chapter is published in [2].

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Applications of Fault Tolerant Robots

In recent years, robots have become increasingly common for a wide range of applications, and the reliability of robots operating in structured and benign environments is quite high. However for dangerous tasks in remote or hazardous environments, where routine maintenance can not be performed, one must plan for the probability of failures. Such applications include space exploration [3–5], underwater exploration [6–8], nuclear waste remediation [9–11], and disaster rescue [12–14]. One may also need to plan for the probability of failures in applications requiring high degrees of safety, such as robotic surgery [15], rehabilitation [16], and human robot interaction [17–20]. Fault tolerant robots are defined as those robots that can still fulfill their remaining assigned tasks after a failure has occurred, without any hardware repair. These types of robots are especially useful in the above two situations.

1.2 Reliability of Robot Systems

The reliability of individual components of the robot can be calculated based on the failure rate data that are available from a database. However, the reliability of robot systems is much more complicated, which depends on many factors [21]. Various robot reliability assessment methods have been developed, such as fault tree analysis [22], reliability block diagram [23] and Markov and Non-Markovian Models [24]. It is suggested that well-designed robots are to be expected to have a useful life of at least 40,000 working hours, mean time to failure (MTTF) of at least 400 hours, and a Mean Time To Repair (MTTR) of no more than eight hours [25].

Based on probability theory, robotics systems will inevitably experience failures. To realize fault tolerance, one must consider the types of failures that could occur, and the effects that they will have on the robotic system, i.e., a Failure Mode and Effects Analysis (FMEA). The potential failure modes of a medical robotic system, such as chemical corrosion, adhesion, cracking/breaking, etc. and their causes and effects is studied in [26], which can be used to improve the design of the arm structure components in order to obtain an optimized model. Two mechanical robots, 3P and 6R robots, are analyzed by using FMEA, and critical failure modes are determined for each robot in [27]. Corrective actions are proposed for critical items to modify robots reliability and reduce their risks.

Because it is impossible to anticipate all possible failures, it is typical to design, plan and control robots to be fault tolerant to the classes of failures that are most likely. The most frequently occurring failures can be categorized as locked-joint failures [28], because many failures do result in a locked joint, but also because other joint failure modes, such as free-swinging joint failures, can be transformed into this failure mode using fail-safe brakes [29].

1.3 Fault Tolerance of Redundant Robots

1.3.1 Overview

Many failures can be represented by the inability to control one of the degrees of freedom. In such cases, in addition to improving the reliability of the components of the robot, one can also employ kinematically redundant robots. These types of robots have more than the minimum number of degrees of freedom (DOF) to achieve their assigned tasks, so their extra DOFs can be used to tolerate different types of joint failures. Many of the previous studies on using kinematically redundant robots to achieve fault tolerance can be roughly divided into three categories, namely, design, motion planning and control.

1.3.2 Fault Tolerant Kinematic Design

In the design category, it has been previously shown that an improperly designed kinematically redundant robot can actually be fault intolerant [30], so that there has been significant effort devoted to identifying fault-tolerant kinematic designs. Researchers have explored the number of DOFs that are necessary and sufficient to guarantee fault tolerance, along with how these joints should be distributed [31]. Other work has assumed a certain amount of redundancy, frequently a single additional DOF, and then developed an optimal kinematic design [32,33]. In [32], researchers applied a genetic algorithm to optimize the joint types and link lengths based on the fault tolerant workspace reachability index. In [33], researchers incorporate fault tolerance and reliability into the design of robot manipulators using fault tree analysis. Other studies have shown that there exist entire classes of designs with desired optimal fault tolerance properties [1, 34–38]. In [34] researchers constructed optimally fault tolerant Jacobians for equal joint failure probabilities to optimize the worst-case minimum singular value after a failure. Based on these Jacobians, families of optimally fault tolerant planar 3R robots [35], spatial positioning 4R robots [36] and spatial positioning and orienting 7R robots [37] were designed. The structural characteristics and global pre- and post-failure dexterity performance of the 7! optimally fault tolerant robots generated in [37] were further analyzed in [1]. In [30], a local measure of fault tolerance for kinematically redundant manipulators, named worst-case relative manipulability, was defined. The optimally fault tolerant Jacobians were constructed based on this fault tolerance measure, and examples of optimally fault-tolerant 7- and 8-DOF mechanisms were presented in [39]. In [38], a class of orthogonal Gough-Stewart platforms was developed that provide optimal fault tolerant manipulability under a single failure.

1.3.3 Fault Tolerant Motion Planning

In the motion planning category, fault tolerant motion planning strategies can be applied to self-repair or recover from joint failures [3, 4, 40–43] or be incorporated in anticipation of failures [37, 44–49]. In the first case, the goal is to adapt the robot's motion strategy after a failure to allow the robot to compensate for the failure and finish its task without any hardware repair. For legged robots, the fault tolerant gaits need to be identified efficiently in response to the damage [40–42], and for manipulators, the motion of the healthy joints needs to be re-planned after a failure to compensate for the motion of the locked joints [3, 4, 43].

In the latter case, the goal is to keep the robot in a configuration that guarantees good performance after a failure. This guarantee takes different forms depending on the type of task being performed. For point-to-point tasks, such as pick and place, the robot should be able to reach all task points even after an arbitrary joint failure. This can be guaranteed by making sure that none of the robot's joints go outside of the ranges of the self-motion manifolds associated with every task point. For a single task point, this range is determined by the bounding box enclosing the self-motion manifolds of that task point. Similarly, for multiple task points, the robot configuration must be constrained to the intersection of all bounding boxes, which provides a set of artificial joint limits to ensure fault tolerance [44].

For path tracking tasks, such as cutting, the robot should be able to complete the entire path after a failure. This can be guaranteed by making sure that the end effector path is within the "fault-tolerant workspace", which is defined as the intersection of the pre-failure and all post-failure workspaces [45, 46, 50]. The size and shape of these workspaces are determined by the artificial joint limits that are applied. In those cases where a relatively large pre-failure workspace is required, it may not be possible to guarantee a sufficient fault-tolerant workspace. For such cases, the reliability map, which is defined based on the joint failure probabilities, can be used to maximize the probability of task completion [47].

For trajectory tracking tasks, such as arc welding and painting, the robot should be able to maintain the required end-effector velocity even after a failure. This can be guaranteed by making sure that for every possible joint failure, some combination of the healthy joints is able to recover the end effector velocity lost from the failed joint. This will be true if every minimum singular value of the post-failure Jacobians is not zero. A worst-case measure of fault tolerance is the smallest of these minimum singular values [37], which is also related to the maximum joint velocity jump [48, 49, 51, 52] and end-effector velocity jump [53, 54] due to failure.

1.3.4 Fault Tolerant Control

In the control category, researchers frequently developed some type of fault detection [55–57], isolation and identification [58, 59] schemes, and then proposed fault tolerant control systems to self-repair or recover from failures. In [60], based on the isolated fault, an inverse dynamics robust controller is reconfigured to include an adaptive term. In [61], a mode-dependent dynamic output feedback controller for wheeled mobile manipulators is designed based on Markovian control theory, which guarantees not only the robust stochastic stability but also a prescribed disturbance attenuation level for the resulting closed-loop system. In [62], a fast finite time active fault tolerant control is developed by combining a robust nonsingular fast terminal sliding mode control with a simple fault diagnosis scheme. In [63], an active fault tolerant control system is developed for reconfigurable manipulator actuator based on local joint information.

1.4 Contribution of This Study

The problem of kinematic design of optimally fault tolerant robots for both equal and unequal joint failure probabilities is studied in this work first. After the optimally fault toelrant robots are designed, the problem of planning an optimal trajectory with minimum probability of task failure is explored. In kinematic design, the main contributions of this work are as follows: (1) analyze the characteristics of the kinematic properties, described by Denavit and Hartenberg (DH) parameters, of the 7! robots generated from an optimally fault tolerant 7R Jacobian, and illustrate the structural correlations between these 7! robots; (2) study the global pre- and post-failure dexterity performance of the 7! robots, and obtain the optimal robot designs. (3) determine the equations for the null space and the canonical form of a Jacobian that optimizes the minimum singular value after a failure for arbitrary joint failure probabilities (4) design classes of robots that are optimal in terms of the resulting minimum singular value after a failure for an arbitrary set of joint failure probabilities, e.g., planar 3R robots, spatial positioning 4R robots and spatial positioning and orienting 7R robots; (5) develop a method to design optimally fault tolerant robots for cases where the joint failure probabilities change, and illustrate this method for planar 3R robots. In motion planning, the main contributions of this work are as follows: (1) identify the failure probability of an arbitrary joint path; (2) propose the method of planning the optimal trajectory with minimum probability of task failure.

1.5 Organization of This Study

The remainder of this dissertation is organized as follows:

Chapter 2 explores the problem of kinematic design of optimally fault tolerant robots for equal joint failure probabilities. An optimally fault tolerant Jacobian for 7R manipulators has previously been constructed based on a measure of local fault tolerance. Because this measure is solely based on the singular values of the Jacobian, permutation of the columns does not affect the optimality. Therefore, when one generates a kinematic robot design from this optimal Jacobian, there will be 7! robot designs with the same locally optimal fault tolerant property. The work described in this chapter shows how to analyze and organize the kinematic structure of these 7! designs in terms of their Denavit and Hartenberg (DH) parameters. Furthermore, global fault tolerant measures are defined in order to evaluate the different designs.

Chapter 3 studies the problem of kinematic design of optimally fault tolerant robots for different joint failure probabilities. A measure of fault tolerance for different joint failure probabilities is defined based on the properties of the singular values of the Jacobian after failures. Using this measure, methods to design optimally fault tolerant robots for an arbitrary set of joint failure probabilities and multiple cases of joint failure probabilities are introduced separately.

Chapter 4 considers the problem of planning the optimal trajectory with minimum probability of task failure for a set of point-to-point tasks, after experiencing locked joint failures. The proposed approach first develops a method to calculate the probability of task failure for an arbitrary trajectory based on the failure scenarios of different regions in the joint space. Then, a motion planning algorithm is proposed to find the optimal trajectory with minimum probability of task failure. The proposed approach is demonstrated on planar 2R redundant robots, illustrating the effectiveness of the algorithm.

Chapter 5 presents the conclusions and future work of this research.

Chapter 2

Kinematic Design of Optimally Fault Tolerant Robots for Equal Joint Failure Probabilities [3]

2.1 Chapter Overview

The work presented in this chapter falls into the category of optimal fault tolerant kinematic design of robots with a single degree of redundancy that are used for fully general spatial positioning and orienting, i.e., 7 DOF manipulators. This chapter builds on the study of [34] and [37] that will be briefly reviewed in the next section. In [34], an optimally fault tolerant Jacobian that is isotropic before failure and possesses the maximal worst-case failure tolerance after failure is developed. From the optimally fault tolerant Jacobian, a family of $7!$ different manipulator kinematics that locally possesses the properties of this Jacobian was generated in [37]. Then the volume of the six-dimensional workspace where the robot has a guaranteed level of fault tolerance was calculated for a few examples. However, there was not an exhaustive analysis of the global properties of the large number of robots with the desired locally optimal design. Nor has there been any taxonomy developed for the classification of robots into similar characteristics that helps to explain their global pre- and post-failure dexterity performance.

These topics are the focus of this chapter. Specifically, the main contributions of this chapter are as follows: (1) the characteristics of the kinematic properties, described by Denavit and Hartenberg (DH) parameters, of the $7!$ robots are analyzed, and used to illustrate the structural correlations between these $7!$ robots; (2) the global pre- and post-failure dexterity performance of the $7!$ robots are studied and the optimal robot designs are obtained.

[3] Most of this chapter is published in [1].

2.2 Background on Optimally Fault Tolerant Kinematic Design

2.2.1 Definition of Optimally Fault Tolerant Jacobians

The Jacobian matrix **J** of a robot is a mapping from the joint angle velocities to the end-effector velocities, which is frequently used to quantify the dexterity of a robot. For an *n* DOF robot working in an *m* dimensional workspace, the Jacobian **J** is an $m \times n$ matrix, written as a collection of columns

$$\mathbf{J}_{m\times n} = \begin{bmatrix} j_1 & j_2 & \cdots & j_n \end{bmatrix} \tag{2.1}$$

where j_i is the contribution of joint *i* to the end-effector velocity. When an arbitrary single joint *f* fails and is locked, the reduced Jacobian ${}^f\mathbf{J}$ can be simply obtained by removing the *f*th column from the original Jacobian to obtain

$${}^f\mathbf{J}_{m\times(n-1)} = \begin{bmatrix} j_1 & j_2 & \cdots & j_{f-1} & j_{f+1} & \cdots & j_n \end{bmatrix}. \tag{2.2}$$

In this work, failure tolerance is defined as the worst-case dexterity after an arbitrary single joint is locked. Based on this definition, a measure of the worst-case failure tolerance is given by

$$\sigma_m^* = \min_{f=1}^{n} {}^f\sigma_m \tag{2.3}$$

where the superscript * indicates a post-failure measure, and ${}^f\sigma_m$ is the minimal singular value of ${}^f\mathbf{J}$.

In [34], an optimally fault tolerant Jacobian is defined as follows: (1) in order to ensure that the robot has optimal dexterity performance before failure, the optimally fault tolerant Jacobian is required to be isotropic, i.e.,

$$\sigma_1 = \sigma_2 = \cdots = \sigma_m = \sigma \tag{2.4}$$

where the σ_i's are the singular values of the original Jacobian; (2) in order to ensure that the robot has optimal fault tolerance after failure, the optimally fault tolerant Jacobian should have the maximum value of the worst-case failure tolerance measure. Under the condition (2.4), the worst-case failure tolerance measure reaches its maximum value when

$$^{1}\sigma_m = \, ^{2}\sigma_m = \, \cdots = \, ^{n}\sigma_m = \, \sigma\sqrt{\frac{n-m}{n}} \tag{2.5}$$

where σ is the singular value of the original Jacobian. This optimally fault tolerant Jacobian requires that each joint contributes equally to the null space, which physically means that the redundancy of the robot is uniformly distributed among all the joints so that a failure at any one joint can be compensated for by the remaining joints.

Using the above definition, the structure of an optimally fault tolerant Jacobian can be identified. For the case of a seven DOF fully spatial manipulator, the canonical optimal Jacobian is a triangular matrix where the i^{th} row is given by:

$$\begin{aligned} j_{ik} &= 0 && \text{where} \quad k < i \\ j_{ik} &= -\sqrt{\tfrac{7-i}{8-i}} && \text{where} \quad k = i \\ j_{ik} &= \sqrt{\tfrac{1}{(7-i)(8-i)}} && \text{where} \quad k > i. \end{aligned} \tag{2.6}$$

Unfortunately, this canonical Jacobian cannot be realized by any manipulator built with only rotational joints. In [34], a physically realizable Jacobian for a rotary joint manipulator that is closest to optimally fault tolerant, was calculated using a numerical approach resulting in:

$$\mathbf{J}^* = \begin{bmatrix} 1 & 0.43 & 0.75 & -0.54 & 0.14 & 0.33 & -0.38 \\ 0 & -0.60 & 0.65 & 0.46 & -0.79 & -0.19 & -0.80 \\ 0 & -0.67 & -0.14 & -0.70 & 0.60 & -0.93 & -0.46 \\ 0 & 0.77 & 0.14 & 0.84 & 0.58 & -0.69 & -0.43 \\ 1 & -0.15 & -0.36 & 0.33 & -0.42 & -0.72 & 0.59 \\ 0 & 0.62 & -0.92 & -0.43 & -0.69 & -0.10 & -0.68 \end{bmatrix}. \tag{2.7}$$

The end-effector position and orientation at this optimally fault tolerant configuration will be referred to as the design location. The next section will show how one can identify all the physically realizable robots that possess this optimal Jacobian.

2.2.2 Generation of Robot Kinematics From Jacobians

Once a Jacobian is identified, the DH parameters of a robot that possesses this Jacobian can be generated by applying the technique developed in [35]. Let v_i and ω_i denote the end-effector linear velocity and orientational velocity, respectively, due to the i^{th} joint velocity. Each column of the Jacobian consists of these two parts, which are calculated as follows,

$$j_i = \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} \hat{z}_{i-1} \times p_{i-1} \\ \hat{z}_{i-1} \end{bmatrix} \forall\, i = 1, ..., n \tag{2.8}$$

where $\hat{z}_{i-1}$ is the unit vector along the i^{th} joint axis, and p_{i-1} is the position vector from the $i-1$ coordinate frame to the hand coordinate frame. Consequently, all joint axes can be obtained from the rotational velocities of the Jacobian. By definition, the x axis of coordinate *i* is the common normal of z_{i-1} and z_i, so all x axes can be obtained after the joint axes are determined. Because all z and x axes are calculated from the Jacobian, the four DH parameters can be obtained according to their definitions.

Permuting the columns of the Jacobian changes the physical parameters of the corresponding robot but does not affect its fault tolerance properties. Therefore, in [37] a family of 7! different robot designs were generated from the Jacobian in (7), and the 6-dimensional fault tolerant workspace volume was defined. However, only three robot designs were evaluated due to the computational complexity of the workspace calculations. The following section performs an analysis on the structure of all these 7! robots. This is followed by an analysis of their global pre- and post-failure capabilities that ultimately can be used to determine the best robot designs.

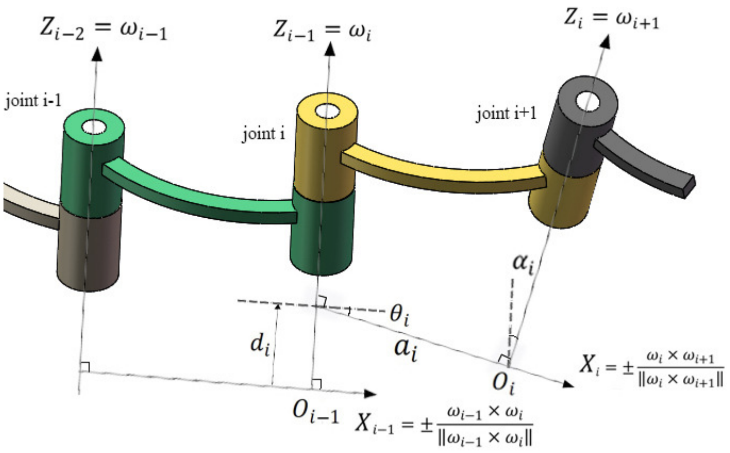
2.3 Characteristics of the Kinematic Parameters of the 7! Optimal Robots

2.3.1 All the Possible Values of the Four DH Parameters

By definition, DH parameter α_i is the twist angle from the z_{i-1} axis to the z_i axis about the x_i axis, and a_i is the link length, i.e., minimum distance from the z_{i-1} axis to the z_i axis along the x_i axis, as shown in Figure 2.1. From the definition it can be seen that the value of α_i and a_i are both determined by the z_{i-1} axis and the z_i axis, which are obtained from two adjacent columns of a Jacobian, and the permutation of the z_{i-1} axis and the z_i axis does not affect the sign of α_i and a_i due to the convention that the x_i axis is chosen to point away from the z_{i-1} axis. Because the order does not matter, there are $C(7,2) = 21$ possible combinations to choose 2 columns from the 7 columns of the optimal Jacobian in (2.7) to generate two adjacent columns, so there are only 21 possible values of α and a in the 7! permutations. In addition, each α has a unique associated a, and vice versa.

Similarly, by definition, DH parameter d_i is the distance from the origin of the $i-1$ coordinate frame to the x_i axis along the z_{i-1} axis, and θ_i is the joint angle from the x_{i-1} axis to the x_i axis about the z_{i-1} axis, as shown in Figure 2.1. From the definition it can be seen that the value of d_i and θ_i are both determined by the x_{i-1} axis and the x_i axis, which are obtained from the z_{i-2}, z_{i-1} and z_i axes. Because d and θ are computed based on three consecutive columns, permuting these columns will change their values. Thus, there are $P(7,3) = 210$ possible permutations to choose

3 columns from the 7 columns of the optimal Jacobian in (2.7), so there are 210 possible values of d and θ in the $7!$ permutations. In addition, each d has a unique associated θ, and vice versa.



$Z_{i-2} = \omega_{i-1}$
$Z_{i-1} = \omega_i$
$Z_i = \omega_{i+1}$
joint i-1
joint i
joint i+1
α_i
θ_i
d_i
a_i
O_i
$X_i = \pm \frac{\omega_i \times \omega_{i+1}}{\|\omega_i \times \omega_{i+1}\|}$
O_{i-1}
$X_{i-1} = \pm \frac{\omega_{i-1} \times \omega_i}{\|\omega_{i-1} \times \omega_i\|}$

Figure 2.1: The four DH parameters for each joint that specify a robot's kinematics can be obtained from the columns of a desired Jacobian, in our case the optimally fault tolerant Jacobian given in (2.7).

In summary, for all the DH parameters of the $7!$ robot designs, there are only 21 pairs of α and a, as shown in Figure 2.2a (with the joint pairs that generate them shown in Figure 2.2c), and they have the following properties:

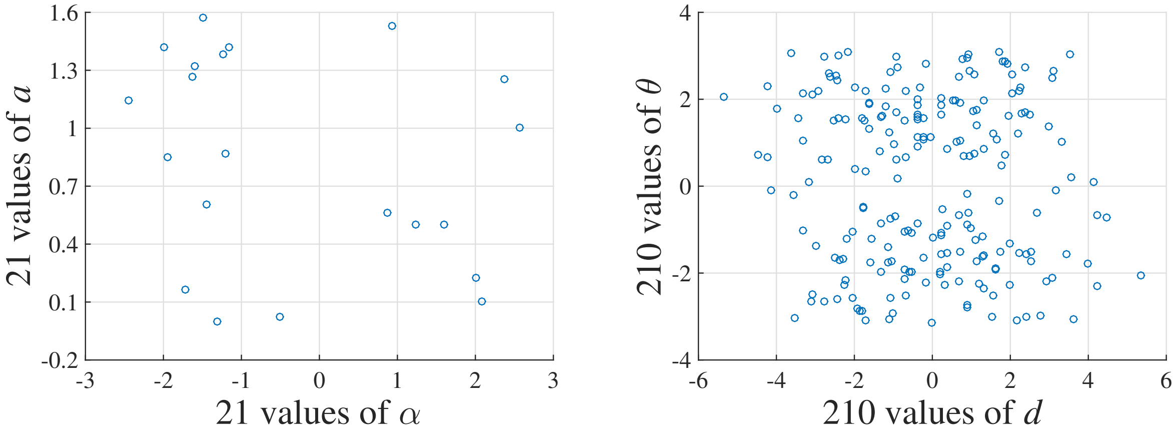
(1) The values of α are not uniformly distributed. In particular, there are no α's near 0, and there is a big gap between -0.50 and 0.86, because if joints i and k are parallel then $\omega_i = \omega_k$ so that j_i and j_k will have similar contributions to the end effector velocity, which is not beneficial for dexterity or fault tolerance. In contrast, there are many α's near $\pm\pi/2$, because if joint i is orthogonal to joint k then $\omega_i \perp \omega_k$ so that j_i and j_k are more likely to maintain the optimal angular separation for fault tolerance.

(2) The values of a vary from 0.00 to 1.57, which makes sense because a can not be greater than 2. This is true because at the optimally fault tolerant configuration all the joints are constrained to lie on a unit sphere that is centered at the end effector, and a is the distance between two joint axes, which can not be larger than the diameter of the unit sphere.

There are 210 pairs of d and θ, as shown in Figure 2.2b, and they have the following properties:

(1) For each pair of d and θ there exists a pair $-d$ and $-\theta$ that is symmetric with respect to the origin. This is because the permutations of two x axes affect the sign of d and θ, but not the value.

(2) In contrast to the link length a, the values of d vary from -5.35 to 5.35, and are not limited to 2. In fact, because d is the distance between the location of the two common normals before and after its joint axis, it can theoretically be from $-\infty$ to ∞, however, d is limited because of the properties of α.



(a)

(b)

j_2	j_3	j_4	j_5	j_6	j_7	
{-1.72, 0.17}	{-1.94, 0.85}	{-1.23, 1.38}	{2.00, 0.23}	{2.37, 1.26}	{0.94, 1.53}	j_1
	{-1.99, 1.42}	{1.24, 0.50}	{-1.49, 1.57}	{2.08, 0.10}	{2.57, 1.00}	j_2
		{-1.16, 1.42}	{-0.50, 0.02}	{-1.32, 0.00}	{-1.21, 0.87}	j_3
			{0.86, 0.56}	{-2.45, 1.14}	{-1.44, 0.60}	j_4
				{-1.60, 1.32}	{1.60, 0.50}	j_5
					{-1.63, 1.27}	j_6

(c)

Figure 2.2: In (a) are all possible 21 pairs of α and a for the 7! optimal fault tolerant robot designs. The joint pairs that generate these values are shown in (c). Note the large number of robots with α near $-\pi/2$. In (b) are all possible 210 pairs of d and θ that are symmetric with respect to the origin.

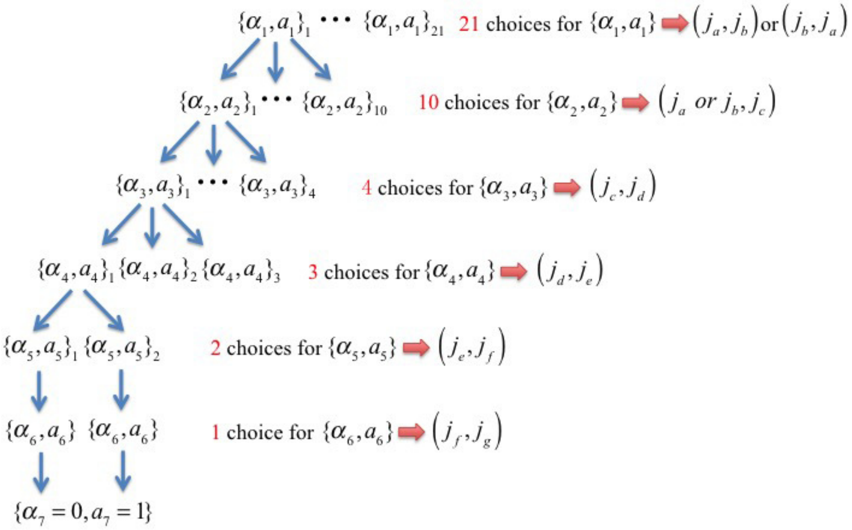
It is important to point out that the DH parameters for a given robot's kinematic structure are not unique. For example, one can take the negative of any column of the Jacobian and the DH parameters will change, however, the robot's kinematic structure and its fault tolerant properties will not. In particular, taking the negative of the i^{th} column of the Jacobian changes the values of α_{i-1} and α_i by adding π (if α_{i-1} or α_i is negative) or subtracting π (if α_{i-1} or α_i is positive). The signs of d_i and θ_i will also change.

2.3.2 Organization of the Seven Sets of DH Parameters for Each Robot

In Subsection 2.3.1, all the possible DH parameters of the 7! robot designs generated from the optimally fault tolerant Jacobian in (2.7) were determined. In this section, we illustrate how the DH parameters for each of the seven joints of a robot can be obtained by selecting from the above possible values.

As stated in Subsection 2.3.1, there are 21 possible values for each individual α_i and a_i. Because selecting one determines the other, we will denote the pair as $\{\alpha_i, a_i\}$ to emphasize that they can not be separated. In addition, it is important to recall that once an $\{\alpha_i, a_i\}$ is selected, this implies that two specific columns of the Jacobian, say j_a and j_b must be adjacent (see Figure 2.2c), but the order does not matter. We will denote this with the ordered pairs (j_a, j_b) and (j_b, j_a). Also, whenever $\{\alpha_{i+1}, a_{i+1}\}$ is selected, note that its associated ordered pair must contain one Jacobian column from the order pair of the previous $\{\alpha_i, a_i\}$.

We now describe the possibilities for determining all seven sets of DH parameters of an optimal robot. The value of $\{\alpha_1, a_1\}$ can be chosen freely from the 21 possible choices. Once the value of $\{\alpha_1, a_1\}$ is determined, the first two columns of the associated Jacobian (j_a, j_b) or (j_b, j_a) are determined. Thus there are now two possibilities for j_2. These two possibilities can be paired with five remaining choices for j_3 so that there are ten possibilities for $\{\alpha_2, a_2\}$. The choices for remaining $\{\alpha_i, a_i\}$'s can be determined in an analogous manner. In particular, the number of possible choices for $\{\alpha_3, a_3\}$ ... $\{\alpha_6, a_6\}$ are 4, 3, 2, and 1, respectively. The value of $\{\alpha_7, a_7\}$ is arbitrary because the end-effector coordinate frame is arbitrary. In [35], α_7 is set to be 0, and a_7 is set to be 1. The organization of all the 7 sets of $\{\alpha_i, a_i\}$ is a tree structure, as shown in Figure 2.3.

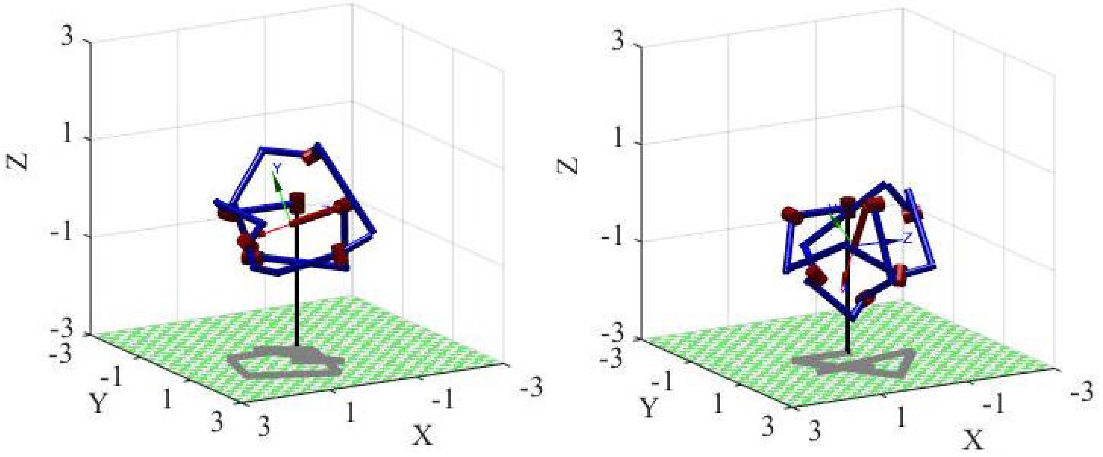


$\{\alpha_1, a_1\}_1 \cdots \{\alpha_1, a_1\}_{21}$
21 choices for $\{\alpha_1, a_1\}$ (j_a, j_b) or (j_b, j_a)
$\{\alpha_2, a_2\}_1 \cdots \{\alpha_2, a_2\}_{10}$
10 choices for $\{\alpha_2, a_2\}$ $(j_a \text{ or } j_b, j_c)$
$\{\alpha_3, a_3\}_1 \cdots \{\alpha_3, a_3\}_4$
4 choices for $\{\alpha_3, a_3\}$ (j_c, j_d)
$\{\alpha_4, a_4\}_1 \{\alpha_4, a_4\}_2 \{\alpha_4, a_4\}_3$
3 choices for $\{\alpha_4, a_4\}$ (j_d, j_e)
$\{\alpha_5, a_5\}_1 \{\alpha_5, a_5\}_2$
2 choices for $\{\alpha_5, a_5\}$ (j_e, j_f)
$\{\alpha_6, a_6\}$ $\{\alpha_6, a_6\}$
1 choice for $\{\alpha_6, a_6\}$ (j_f, j_g)
$\{\alpha_7 = 0, a_7 = 1\}$

Figure 2.3: The tree structure of the 7 sets $\{\alpha_i, a_i\}$ of each robot design

2.3.3 Correlations Between the DH Parameters of Two Reverse Version Designs

For any robot design from the 7! robots whose associated optimally fault tolerant Jacobian is $\mathbf{J}^{\mathrm{l}} = \begin{bmatrix} j_{\mathrm{l}_1} & j_{\mathrm{l}_2} & j_{\mathrm{l}_3} & j_{\mathrm{l}_4} & j_{\mathrm{l}_5} & j_{\mathrm{l}_6} & j_{\mathrm{l}_7} \end{bmatrix}$, there exists a reverse version of this robot whose associated Jacobian is $\mathbf{J}^{\mathrm{r}} = \begin{bmatrix} j_{\mathrm{l}_7} & j_{\mathrm{l}_6} & j_{\mathrm{l}_5} & j_{\mathrm{l}_4} & j_{\mathrm{l}_3} & j_{\mathrm{l}_2} & j_{\mathrm{l}_1} \end{bmatrix}$, where “l” indicates left to right order, and “r” is the reverse. Based on the analysis in Subsection A, these two robot designs have the same α’s and a’s, but they are in reverse order, i.e., $\alpha_1^{\mathrm{l}} = \alpha_6^{\mathrm{r}}$, $a_1^{\mathrm{l}} = a_6^{\mathrm{r}}$, ..., $\alpha_6^{\mathrm{l}} = \alpha_1^{\mathrm{r}}$, $a_6^{\mathrm{l}} = a_1^{\mathrm{r}}$. These two robot designs have d’s and θ’s that are of opposite sign, and they are also in reverse order, i.e., $d_2^{\mathrm{l}} = -d_6^{\mathrm{r}}$, $\theta_2^{\mathrm{l}} = -\theta_6^{\mathrm{r}}$, ..., $d_6^{\mathrm{l}} = -d_2^{\mathrm{r}}$, $\theta_6^{\mathrm{l}} = -\theta_2^{\mathrm{r}}$. Figure 2.4 shows two reverse version robot designs. It can be seen that although these two robots have very similar DH parameters, they have different structures, and it will be shown in the next section that the global properties of these two reverse version robots is quite different, although this is not typically the case.



(a) **(b)**

Figure 2.4: Two reverse version robot designs that have very different global properties. Note how all of the joint axes are tangent to a sphere of unit radius about the end effector position, as is required for an optimally fault tolerant configuration. (This depiction of the robot kinematics is meant to illustrate these properties and not to represent how the robot would be manufactured.)

2.4 Global Pre- and Post-Failure Dexterity of the 7! Robot Designs

2.4.1 Overview

All these 7! robot designs possess the locally optimal fault tolerant Jacobian at a specific optimal configuration, so they have the same optimal local performance with the end effector at the optimal design location. However, it is important to also consider the global performance of a robot design. In [37], a global measure of a robot's 6-dimensional fault tolerant workspace volume was proposed, but only the global performance of three robot designs was studied due to the computational complexity. In this section, the global pre- and post-failure dexterity performance of all 7! robot designs are studied, and based on these results, the optimal robot designs are identified.

2.4.2 Correlations Between Common Pre- and Post-Failure Dexterity Measures

The pre-failure dexterity performance is evaluated by using the following three commonly used measures, i.e., minimal singular value σ_m, the condition number κ, and manipulability w, which are defined as

$$\kappa = \frac{\sigma_1}{\sigma_m} \quad \text{and} \tag{2.9}$$

$$w = \sigma_1 \sigma_2 \cdots \sigma_m. \tag{2.10}$$

We will use the reciprocal of κ to make the measure be between 0 and 1. The post-failure dexterity performance is frequently measured by the worst-case value of the above three dexterity measures after an arbitrary joint is locked, i.e.,

$$\kappa^* = \min_{f=1}^{n} {}^f\kappa, \tag{2.11}$$

$$w^* = \min_{f=1}^{n} {}^f w, \tag{2.12}$$

and σ_m^* is defined as in (2.3), where ${}^f\kappa$ and ${}^f w$ are the condition number and manipulability, respectively, after joint f is locked. In order to eliminate the difference in units between linear velocity and rotational velocity, the first three rows of the Jacobian are normalized by the maximal distance that this robot can reach, before calculating these measures.

In order to study the correlations between these six pre- and post-failure dexterity measures, 10,000 configurations are randomly sampled in the joint space of the robot design generated from the Jacobian in (2.7). The correlation coefficient between each of the measures is shown in Table 2.1. The three pre-failure dexterity measures are highly correlated with each other, especially the inverse condition number and minimal singular value, and this is also true for the three post-failure dexterity measures. However, the correlations between the pre- and post-failure dexterity measures are relatively low. In the sub correlation coefficient matrix of the three pre-failure dexterity measures, the column of σ_m has the largest norm, which means that σ_m is the most representative measure to evaluate the pre-failure dexterity performance of the robot designs. Similarly, σ_m^* is the most representative measure among the three post-failure dexterity measures. Figure 2.5 shows the values of σ_m and σ_m^* at all 10,000 sampled configurations. It can be seen that the configurations with a small σ_m must also have a small σ_m^*, however, the reverse is not true. This indicates that σ_m^* is really a measure of a different property. For the remainder of this work σ_m and σ_m^* are used to evaluate the global pre- and post-failure dexterity performance of the 7! robot designs.

Table 2.1: The correlations between the local pre- and post-failure dexterity measures

	$1/\kappa$	w	σ_m	$1/\kappa^*$	w^*	σ_m^*
$1/\kappa$	1.000	0.748	0.992	0.401	0.439	0.401
w	0.748	1.000	0.773	0.342	0.604	0.357
σ_m	0.992	0.773	1.000	0.398	0.454	0.405
$1/\kappa^*$	0.401	0.342	0.398	1.000	0.817	0.996
w^*	0.439	0.604	0.454	0.817	1.000	0.825
σ_m^*	0.401	0.357	0.405	0.996	0.825	1.000

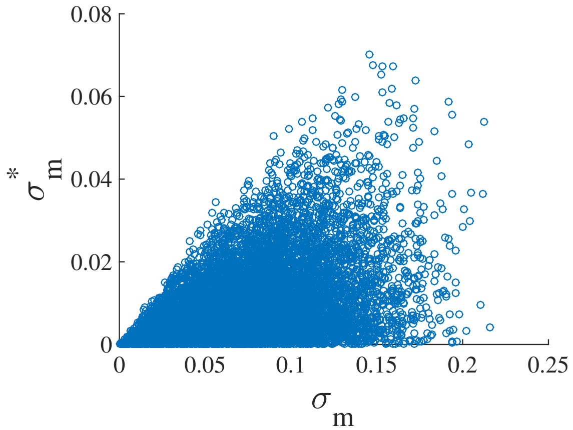


Figure 2.5: The values of σ_m and σ_m^* for 10,000 samples in the joint space for the robot generated from (7). Note that σ_m^* is bounded by σ_m, and can take any value down to zero even for large σ_m.

2.4.3 Correlations Between the Global Pre- and Post-Failure Dexterity in the Joint Space and in the Workspace

There are two ways to approximate the average global performance of a robot, i.e., sampling in the joint space or sampling in the workspace. One can estimate the average global dexterity in the joint space by simply evaluating the dexterity measures at randomly generated configurations and taking the average. That is, the global σ_m and σ_m^* in the joint space are calculated using

$$\bar{\sigma}_m = \frac{\sum\limits_{i=1}^{N} \sigma_m}{N} \quad \text{and} \tag{2.13}$$

$$\bar{\sigma}_m^* = \frac{\sum\limits_{i=1}^{N} \sigma_m^*}{N} \tag{2.14}$$

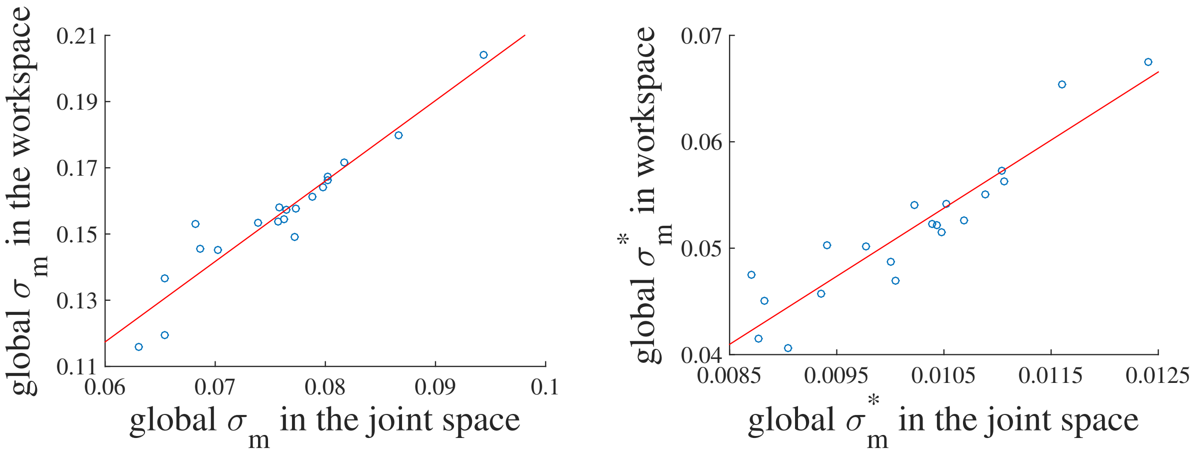
where $\bar{\sigma}_m$ is the average dexterity, $\bar{\sigma}_m^*$ is the average fault tolerance, and N is the number of samples. Clearly the accuracy of the average global measure increases with N, however, so does the computation time. As a compromise between accuracy and computation time, $N = 10,000$ is used when sampling in the joint space. Based on our analysis this results in an error of ≈3% for global σ_m and ≈5% for global σ_m^*.

In contrast to sampling in the joint space, the calculation of the global measure in the workspace is much more difficult and time consuming, because at each location in the workspace, there are multiple configurations that have different local dexterity performance. One can assume that an inverse kinematics routine that optimizes the desired dexterity measure is being used. Therefore, it makes sense to quantify the dexterity measure at a workspace location using the maximum value over all configurations at that location. This requires the following steps: (a) A number of locations (position and orientation) are randomly sampled in the 6-dimensional workspace by the direct sampling method in [37]. (b) All the self-motion manifolds at each location are calculated, and the local σ_m and σ_m^* along these self-motion manifolds are obtained. (c) The maximal σ_m and σ_m^* are saved as the optimal pre- and post-failure dexterity performance at this location. (d)

Finally, the global σ_m and σ_m^* in the workspace can be calculated using (2.13) and (2.14) based on these optimal σ_m and σ_m^* found at each sample location. For sampling in the workspace $N = 5000$ is used due to the greater computational complexity. This results in an error of $\approx 2\%$ for both global σ_m and σ_m^*. It may at first seem strange that a higher accuracy is obtained with a lower number of samples as compared to sampling in the joint space. However, this is due to the fact that an optimization is done at each workspace location to identify the optimal value, which are then averaged.

When robots are applied for a particular task, one is usually more interested in a robot's performance in the workspace. Therefore, robot designers would typically prefer a measure of the workspace dexterity. However, as discussed above, this is much more computationally expensive, especially when it needs to be evaluated for 7! robot designs. If it can be shown that there is a correlation between the measures computed in the joint space and those in the workspace, then the more computationally efficient joint-space computations can be used to identify a smaller number of optimal robot design candidates on which the more computationally expensive workspace analysis can be performed.

To see if joint space measures are correlated to workspace measures, 20 robot designs are randomly chosen from the 7! robots, and their global σ_m and σ_m^* are calculated. Figure 2.6 shows the global σ_m and σ_m^* both in the joint space and in the workspace. There is a relatively strong linear correlation between the joint space measures and the workspace measures. The linear correlation coefficient between the global σ_m in the joint space and in the workspace is 0.74, and the linear correlation coefficient between the global σ_m^* in the joint space and in the workspace is 0.71. This means that one can use the joint-space measures as an approximation for the workspace measures to select optimal robot design candidates.



(a) **(b)**

Figure 2.6: The correlations between global measures in the joint space and in the workspace where (a) is the global σ_m and (b) is the global σ^*_m.

2.4.4 Global Pre- and Post-Failure Dexterity of the 7! Optimal Robots in the Joint Space

Based on the relatively strong linear correlation obtained in the above section, the robot designs with optimal global dexterity performance in the joint space are more likely to have optimal global dexterity performance in the workspace. Therefore, the global pre- and post-failure dexterity performance of all the 7! robots are first calculated in the joint space to find the robot designs with optimal global joint space measures. These robots will be the optimal robot design candidates.

Figure 2.7 shows the global σ_m and σ^*_m of all the 7! robot designs in the joint space. As discussed in Section III, there are some relationships between specific DH parameters and good local pre- and post-failure dexterity measures. Reformatting the data in Figure 2.7 reveals additional correlations with global properties, i.e., that nearly orthogonal joints with small link lengths are more likely to generate robots with good global pre- and post-failure dexterity. There are three joint pairs, out of the 21 possible pairs, that nearly satisfy these conditions, i.e., the $\{\alpha, a\}$ pairs $\{-1.32, 0.00\}$, $\{-1.72, 0.17\}$ and $\{1.60, 0.50\}$. One can classify each of the 7! robot designs into four groups according to how many of these $\{\alpha, a\}$ pairs they contain. The distribution of global pre- and post-failure dexterity performance in the joint space of these groups is shown in

Figure 2.8. It is easy to see that the mean dexterity performance increases as the number of good pairs increases, and even the worst robot in the group with three pairs has relatively good global pre- and post-failure dexterity. However, one must be careful because the order of the DH parameters also matters. Every one of the 7! robot designs has a reverse version that has very similar DH parameters, but in reverse order. The global performance of the two can be quite different, as shown in Figure 2.9 where this difference is plotted for all 7!/2 pairs.

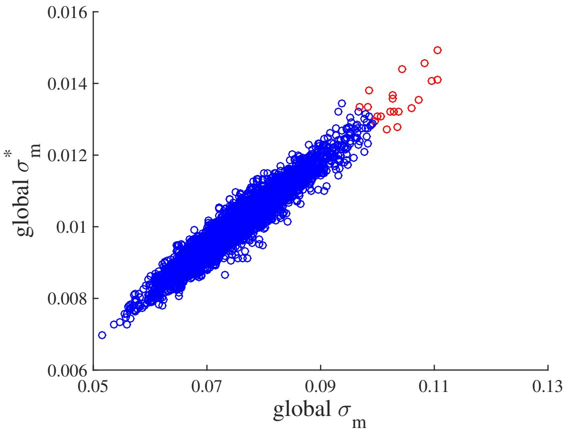


Figure 2.7: The global σ_m and σ_m^* of the 7! robots in the joint space

The red points in Figure 2.7 are the 20 optimal robot design candidates, which have optimal global pre- and post-failure dexterity performance in the joint space. None of the remaining robot designs have both better global pre- and post-failure dexterity than these 20 candidates. Based on the analysis in the above section, it is likely that these 20 candidates have better global performance in the workspace than the remaining robot designs, so only the global performance of these 20 robot designs are calculated in the workspace. These top twenty robot designs are ordered (somewhat arbitrarily) by the sum of their normalized global σ_m and σ_m^*.

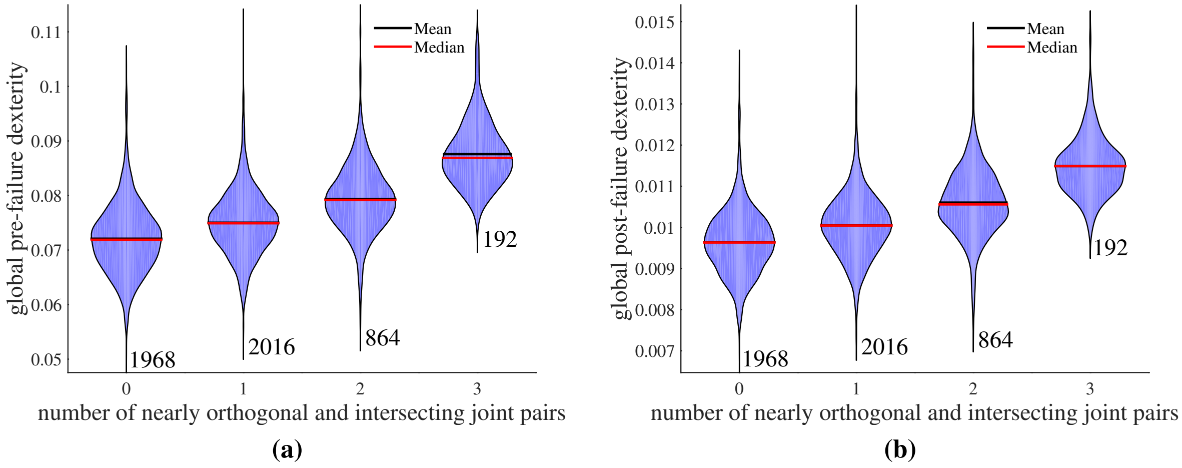


Figure 2.8: Robots that contain a larger number of nearly orthogonal and intersecting joint pairs are more likely to have high global pre- and post-failure dexterity. The 7! robots have been grouped based on the number of such joint pairs, and the distribution of their global pre- and post-failure dexterities are shown in (a) and (b), respectively. The number next to the distribution indicates the numbers of robots in each group.

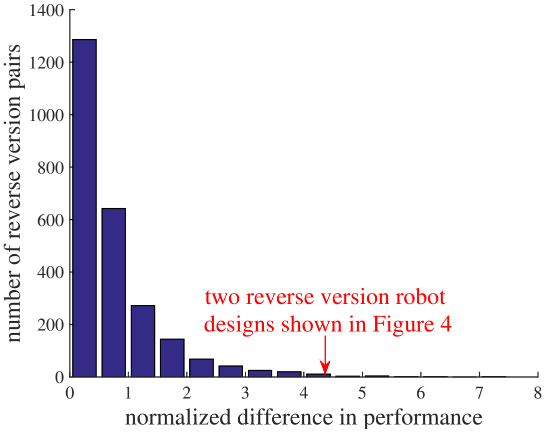
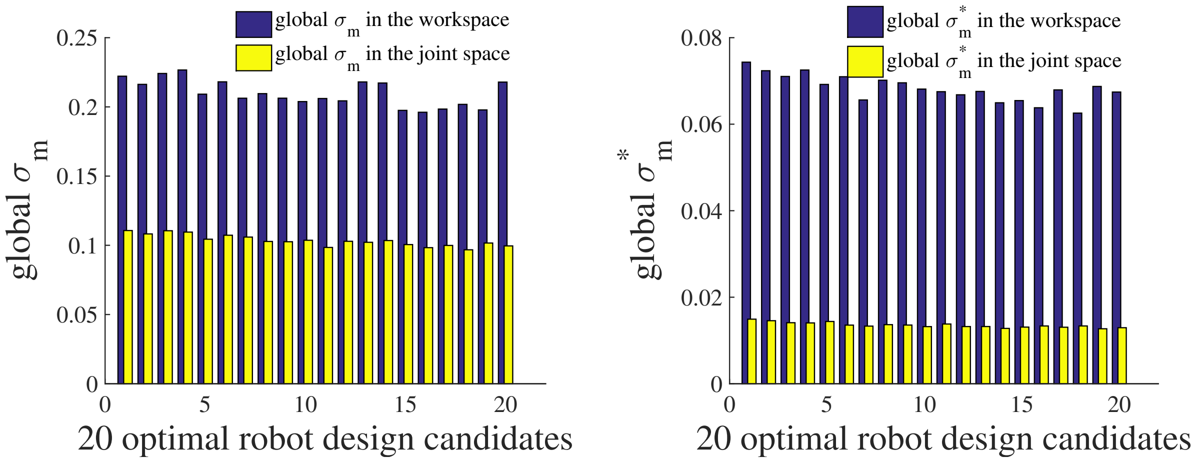


Figure 2.9: The distribution of the difference in performance between a robot design and its reverse version is shown. The difference is computed as the Euclidean distance between the normalized pre- and post-failure measures.

2.4.5 Global Pre- and Post-Failure Dexterity of the 20 Optimal Candidates in the Workspace

The global dexterity performance of the 20 optimal robot design candidates are calculated in the workspace. Both the global joint space measures and the global workspace measures of the 20 candidates are shown in Figure 2.10. Clearly, the global measures in the workspace are much better than those in the joint space, because at each location the configuration with the optimal measures at this location are identified. Figure 2.11 shows the Pareto frontier when the two objective functions are global σ_m and σ_m^*. Among the 20 robot design candidates, Candidate 4, that results from the permutation $\begin{bmatrix} j_7 & j_4 & j_5 & j_1 & j_2 & j_6 & j_3 \end{bmatrix}$, has the best global pre-failure dexterity in the workspace, and Candidate 1, that results from the permutation $\begin{bmatrix} j_6 & j_7 & j_3 & j_5 & j_2 & j_1 & j_4 \end{bmatrix}$, has the best global post-failure dexterity in the workspace, where j_i is the i^{th} column of the optimal Jacobian in (7). These two optimal robot designs are shown in their optimally fault tolerant configurations in Figure 2.12.



(a) (b)

Figure 2.10: The global measures of the 20 optimal robot design candidates in the joint space and in the workspace where (a) is the global σ_m and (b) is the global σ_m^*.

The pre- and post-failure dexterity measures throughout the workspace for the two optimal robot designs are shown in Figure 2.13. Because of the difficulty of representing and visualizing

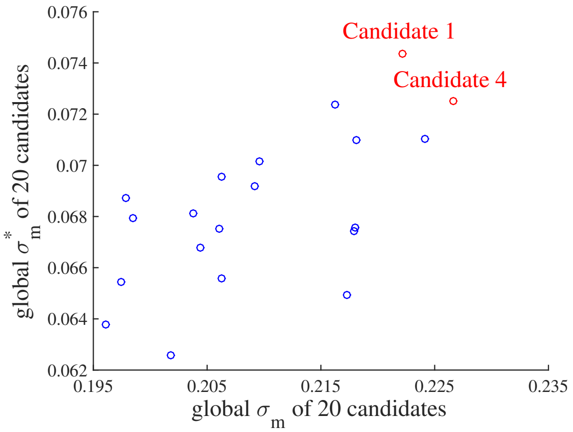


Figure 2.11: The global values of σ_m and σ_m^* computed in the in the workspace are shown for the 20 optimal robot design candidates. Candidates 1 and 4 represent the Pareto solutions to this bi-objective optimization.

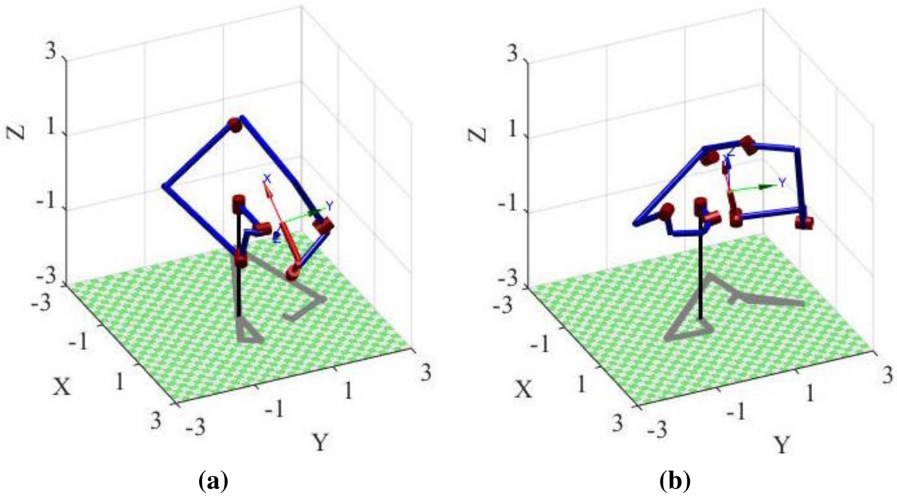
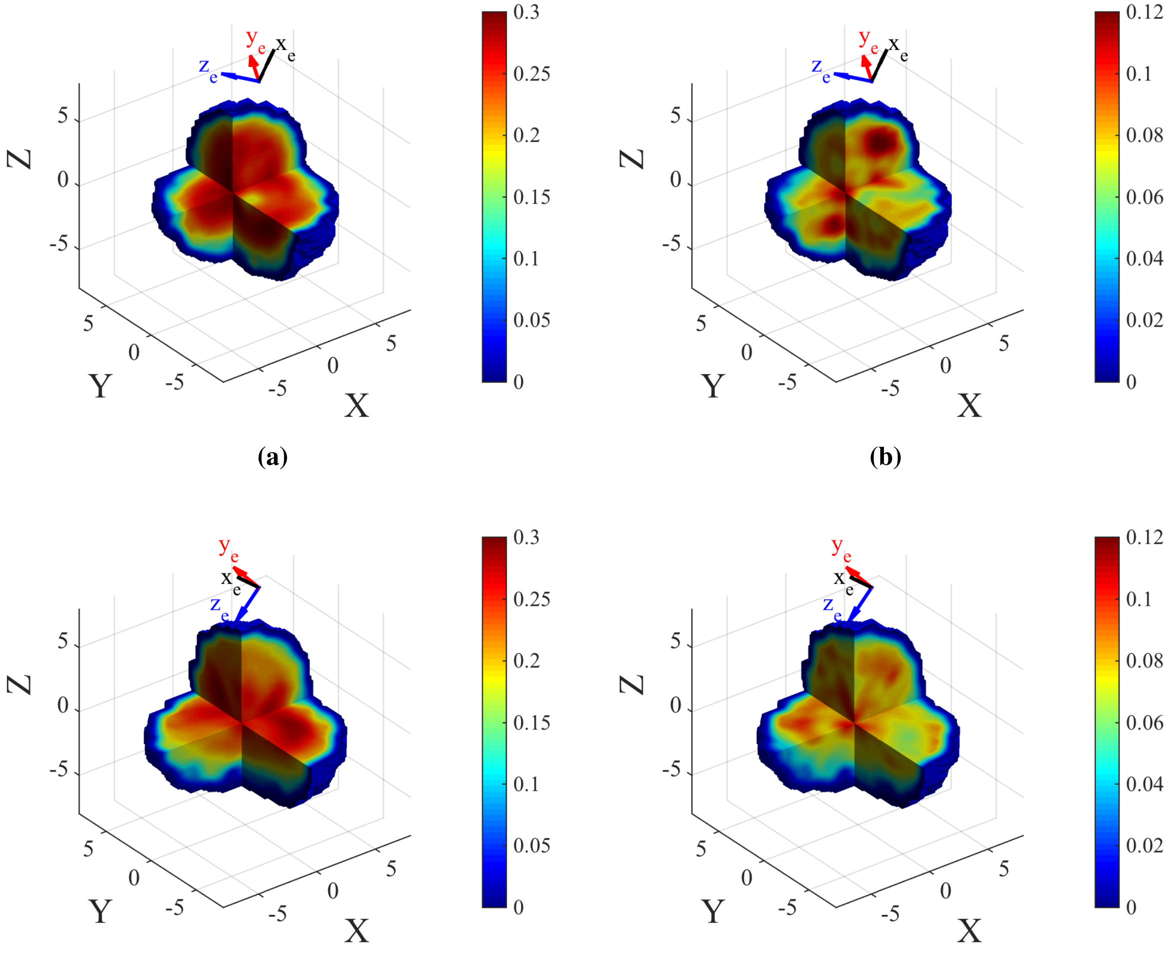


Figure 2.12: The optimal robot designs in their optimal configuration where (a) is the Candidate 1 robot design and (b) is the Candidate 4 robot design.

a 6-dimensional workspace, only the performance in their 3-dimensional position workspace is plotted by constraining the orientation to be that of the optimal design configuration. In order to show the dexterity performance in the interior of the workspace, the 3-dimensional position workspace is shown with multiple cross-sections at the design point. Figure 2.13a and Figure 2.13b are the pre- and post-failure dexterity performance, respectively, of the Candidate 4 robot design, and Figure 2.13c and Figure 2.13d are the pre- and post-failure dexterity performance, respectively, of Candidate 1. From the figure, one can see that neither σ_m nor σ_m^* reaches its maximum value at the center point. This is because at this design location the robots are required to be both isotropic and fault tolerant. At other points, the robots are no longer constrained to be isotropic so that the values of σ_m and σ_m^* can be higher. Note that the volume of high pre-failure dexterity is much more uniform than that of post-failure dexterity. To illustrate the high levels of pre- and post-failure dexterity that can be maintained over a large portion of the workspace, Figure 2.14 shows the configurations of Candidate 1 and 4 robot designs at the location where the fault tolerance measure is 90% of the maximum value.

2.5 Chapter Summary

This chapter explored the structure and global pre- and post-failure dexterity performance of the $7!$ robot designs generated from an optimally fault tolerant Jacobian. It was shown that when describing the kinematic design of a robot in terms of DH parameters for all joints, there are only 21 possible values of α and a for all $7!$ robot designs, and there are 210 possible values of d and θ. In addition, these designs were organized into a tree structure based on the possible choices for $\{\alpha, a\}$ pairs. It was also shown that each of the $7!$ robot designs had a reverse version with very similar DH parameters but potentially very different global properties. Furthermore, the global performance of the $7!$ robot designs was analyzed. It was shown that there is a relatively strong correlation between performance measures computed in the joint space and workspace, so that the computationally efficient joint space calculations could be used to identify the best candidates for optimal designs in the workspace. These candidates were further analyzed in the



(a)

(b)

(c)

(d)

Figure 2.13: The pre- and post-failure dexterity performance in the 3-dimensional position workspace are shown for Candidate robot designs 1 and 4, where both are constrained to be at the orientation of the optimal design point. Pre- and post-failure dexterity performance of Candidate 4 are shown in (a) and (b), respectively, and Candidate 1 in (c) and (d), respectively.

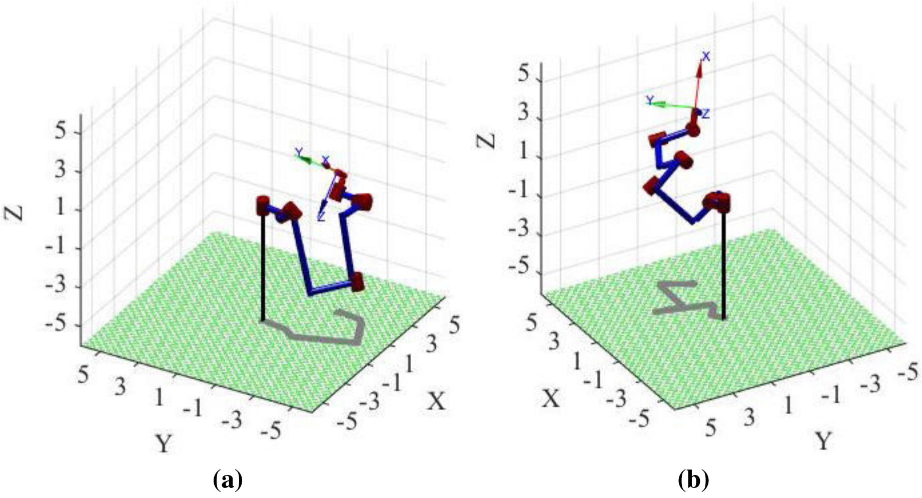


Figure 2.14: The configurations of the optimal robot designs at the locations where the fault tolerance measure is 90% of the maximum value are shown in (a) for Candidate 1 and (b) for Candidate 4.

workspace to determine two Pareto optimal designs in terms of pre- and post-failure dexterity, and the distribution of these measures throughout the workspace were shown.

Chapter 3

Kinematic Design of Optimally Fault Tolerant Robots for Different Joint Failure Probabilities [4]

3.1 Chapter Overview

All of the studies on fault tolerant design [34–37] only consider the case where all joints are equally likely to fail when designing the optimally fault tolerant robots. While this is typically how the robot components are designed, individual variations within the components or environmental factors will change the failure probabilities over the robot's lifetime. Identical components may also have unequal failure probabilities due to the different loads that they are exposed to, which is typically the case for modular robot designs [64]. Thus, when designing an optimally fault tolerant robot, this factor needs to be considered. This is the focus of this chapter. Specifically, the main contributions of this chapter are as follows: (1) determine the equations for the null space and the canonical form of a Jacobian that optimizes the minimum singular value after a failure for arbitrary joint failure probabilities (2) design classes of robots that are optimal in terms of the resulting minimum singular value after a failure for an arbitrary set of joint failure probabilities, e.g., planar 3R robots, spatial positioning 4R robots and spatial positioning and orienting 7R robots; (3) develop a method to design optimally fault tolerant robots for cases where the joint failure probabilities change, and illustrate this method for planar 3R robots. This is in contrast to the work in [30] where unequal joint failure probabilities were first proposed, however, only for the reduced manipulability measure of fault tolerance and not for the minimum singular value. In addition, the kinematic design problem was not explored in [30].

[4] Most of this chapter is published in [2].

3.2 A Definition of Fault Tolerance for Different Joint Failure Probabilities

The local kinematic dexterity of a robot is frequently described by the properties of the Jacobian matrix $\mathbf{J}$, which linearly relates joint velocities to end-effector velocities. The Jacobian $\mathbf{J}$ can be written as a collection of columns

$$\mathbf{J}_{m\times n} = \begin{bmatrix} \mathbf{j}_1 & \mathbf{j}_2 & \cdots & \mathbf{j}_n \end{bmatrix} \tag{3.1}$$

where m is the dimension of the workspace, n is the number of DOFs of the robot, and $\mathbf{j}_i$ is the contribution of joint i to the end-effector velocity. For redundant robots, the DOFs of the robot n is larger than the workspace dimension m, so that the extra DOFs can be utilized to realize fault tolerance.

If an arbitrary single joint fails and is locked, this joint is no longer able to contribute any motion of the end-effector. Therefore, the reduced Jacobian after joint f fails can be easily obtained by removing the fth column from the original Jacobian, i.e.,

$$^{f}\mathbf{J}_{m\times(n-1)} = \begin{bmatrix} \mathbf{j}_1 & \mathbf{j}_2 & \cdots & \mathbf{j}_{f-1} & \mathbf{j}_{f+1} & \cdots & \mathbf{j}_n \end{bmatrix}. \tag{3.2}$$

The local kinematic dexterity of the reduced robot can be quantified by combinations of the singular values of the associated reduced Jacobian, such as their ratio (condition number) or product (manipulability). In this work, the minimal singular value of the reduced Jacobian, denoted as $^{f}\sigma_m$, is used to define the dexterity of the reduced robot after joint f is locked. This is because the minimum singular value tends to denominate the behavior of both the condition number and the manipulability, but also because the minimum singular value is a measure of proximity to a singularity, i.e. a measure of worst-case dexterity over all end-effector motion directions [34]. A larger value of $^{f}\sigma_m$ means that the reduced robot maintains a higher motion ability in the worst-case direction, so that a robot in this configuration is more fault tolerant to the failure of joint f. When

${}^f\sigma_m$ equals 0, the reduced robot is in a singular configuration, and this robot in this configuration is fault intolerant to the failure of joint f.

The definition of fault tolerance in this work is based on all the possible ${}^f\sigma_m$ considering all the possible joint failures. A measure of fault tolerance is defined as

$$\mathcal{F} = w_1\, {}^1\sigma_m + w_2\, {}^2\sigma_m + \cdots + w_n\, {}^n\sigma_m \tag{3.3}$$

where the w_i's are the weighting coefficients, satisfying $\sum\limits_{i=1}^{n} w_i = 1$. In order to weight each reduced Jacobian's minimum singular value with its probability of occurring, the weights are selected as

$$w_i = \frac{p_i}{p_1 + p_2 + \cdots + p_n} \tag{3.4}$$

where p_i is the failure probability of joint i. When all the joints are equally likely to fail, all the weighting coefficients are equal, which implies that all the ${}^f\sigma_m$ are equally important. At the other extreme, i.e., when one of the joints is much more likely to fail than the others, the weighting coefficient of this joint approaches 1, and the weighting coefficients of the other joints approach 0 so that the optimization will focus on this joint's ${}^f\sigma_m$. Based on this definition of fault tolerance, the kinematic design of optimally fault tolerant robots will be studied in the following sections.

3.3 Designing For an Arbitrary Set of Joint Failure Probabilities

3.3.1 Overview

We assume that robot manufacturers have historical data on the failure probabilities of the various physical components of joints used in the design of their robots. If such data is not available, then one can employ the reliability analysis described in [65] using the component reliability data available in [66]. Therefore, these probabilities are known during the kinematic design of robots where the fault tolerance measure defined in (3.3) is maximized. This insures that the robot achieves optimally fault tolerant performance after a locked joint failure. In addition, it is desirable to have optimal dexterity before a failure, so that the robot is required to have an isotropic pre-failure Jacobian, i.e., the singular values are all equal. Therefore, the problem in this section can be stated as finding a Jacobian that satisfies

$$\begin{aligned} &\max\ \mathcal{F} = w_1\,{}^1\sigma_m + w_2\,{}^2\sigma_m + \cdots + w_n\,{}^n\sigma_m \\ &\text{s.t.} \quad \sigma_1 = \sigma_2 = \cdots = \sigma_m = \sigma \end{aligned} \tag{3.5}$$

where σ is the singular value before a failure. The variables in this optimization are the elements of the Jacobian. Because a Jacobian is a function of a robot's kinematic parameters, one can determine the optimal kinematic design from the Jacobian that is the solution to (3.5).

Once an optimally fault tolerant Jacobian is identified for a given set of joint failure probabilities, the method developed in [35] can be applied to generate robot kinematic parameters from this Jacobian, and a family of optimally fault tolerant robots can be obtained. In this section, the optimal null space that maximizes the fault tolerance measure in (3.3) is identified first. Then, the isotropic Jacobian with this optimal null space is constructed.

3.3.2 Finding the Optimally Fault Tolerant Null Vector

For manipulators with one degree of redundancy, the absolute value of the ith component of its Jacobian's null vector, denoted n_i, is equal to the absolute value of the determinant of the Jacobian after joint i fails, which is the product of the singular values after joint i fails, i.e.,

$$|n_i| = |\det({}^i\mathbf{J})| = {}^i\sigma_1\,{}^i\sigma_2\ \cdots\ {}^i\sigma_m. \tag{3.6}$$

The singular values before and after a failure satisfy the following inequality [67]

$$\sigma_1 \geqslant {}^i\sigma_1 \geqslant \sigma_2 \geqslant {}^i\sigma_2 \geqslant \cdots \geqslant \sigma_m \geqslant {}^i\sigma_m. \tag{3.7}$$

Note that the directions of the singular vectors associated with σ_i and ${}^f\sigma_i$ might be quite different. Therefore, although ${}^f\sigma_i > 0$, the robot might still lose the ability to move along the direction of the singular vector associated with σ_i after joint f fails.

For isotropic Jacobians, the singular values before a failure are all equal, i.e., $\sigma_i = \sigma \;\forall\; i$, so from (3.7) it is easy to see that

$$ {}^i\sigma_1 =^i \sigma_2 = \cdots =^i \sigma_{m-1} = \sigma. \tag{3.8}$$

Substituting (3.8) into (3.6), the null space can be obtained as follows

$$|n_i| = |\det({}^i\mathbf{J})| = {}^i\sigma_m \sigma^{m-1}. \tag{3.9}$$

The normalized version of the above equation is given by

$$|\hat{n}_i| = \frac{{}^i\sigma_m \sigma^{m-1}}{\sqrt{\sum\limits_{i=1}^{n}[\det({}^i\mathbf{J})]^2}}. \tag{3.10}$$

By applying the Binet-Cauchy Theorem for an isotropic J

$$\det(\mathbf{J}\mathbf{J}^{\mathrm{T}}) = \sum_{i=1}^{n}[\det({}^i\mathbf{J})]^2 = \sigma^{2m}, \tag{3.11}$$

equation (3.10) becomes

$$|\hat{n}_i| = \frac{{}^i\sigma_m}{\sigma} \tag{3.12}$$

which can be interpreted as the post-failure dexterity divided by the pre-failure dexterity. We can now use this relationship along with the optimization defined by (3.5) to determine the form of an optimal null vector.

Maximizing the objective function defined in (3.5) is equivalent to

$$\begin{aligned} &\max \;\; w_1|\hat{n}_1| + w_2|\hat{n}_2| + \cdots + w_n|\hat{n}_n| \\ &\text{s.t.} \quad |\hat{n}_1|^2 + |\hat{n}_2|^2 + \cdots + |\hat{n}_n|^2 = 1. \end{aligned} \tag{3.13}$$

This constrained optimization problem can be solved by the method of Lagrange multipliers. The Lagrange function is defined as

$$\begin{aligned} L(|\hat{n}|, \lambda) = w_1|\hat{n}_1| + w_2|\hat{n}_2| + \cdots + w_n|\hat{n}_n| - \\ \lambda(|\hat{n}_1|^2 + |\hat{n}_2|^2 + \cdots + |\hat{n}_n|^2 - 1). \end{aligned} \tag{3.14}$$

Solving the set of equations resulting from taking partial derivatives yields

$$|\hat{n}_i| = \frac{w_i}{\sqrt{w_1^2 + w_2^2 + \cdots + w_n^2}}. \tag{3.15}$$

Therefore, when the elements of a Jacobian's null vector are given by (3.15) the objective function reaches its maximum value and ${}^i\sigma_m = \sigma w_i / \sqrt{w_1^2 + w_2^2 + \cdots + w_n^2}$.

There are two additional items that should be pointed out about the above derivation. First, if the constraint of isotropy in (3.5) is replaced by a bound on σ_1, i.e., a simple norm constraint on the Jacobian, the optimal solution will still be the same. This is true because to maximize the objective function, each ${}^i\sigma_m \; \forall \; i$ should be as large as possible. However, the singular values before and after a failure satisfy the inequality in (3.7), which shows that ${}^i\sigma_m$ is bounded by σ_m. Therefore, the minimum singular value before a failure should also be as large as possible. The value of σ_m reaches its maximum when $\sigma_1 = \sigma_2 = \cdots = \sigma_m = \sigma$, i.e., the Jacobian is isotropic, which is the same as the constraint in (3.5). Second, because of isotropy, maximizing the fault tolerance measure defined in (3.3) also maximizes the reduced manipulability [30]. This is true because the determinant in (3.9) is equal to the reduced manipulability. Furthermore, the ratio of the reduced manipulabilities is equal to the ratio of the weighting coefficients.

3.3.3 Constructing Isotropic Jacobians According to the Optimally Fault Tolerant Null Vector

This subsection will discuss how to construct isotropic Jacobians whose associated null vectors satisfy (3.15). An isotropic Jacobian can be decomposed into the following two matrices,

$$\mathbf{J} = \mathbf{D}\mathbf{U}, \tag{3.16}$$

where $\mathbf{D}$ is an $m \times n$ diagonal matrix whose diagonal elements are σ, and $\mathbf{U}$ is an $n \times n$ orthogonal matrix, whose column can be written as

$$\mathbf{u}_i = \begin{bmatrix} \frac{1}{\sigma}\mathbf{j}_i \\ \hat{n}_i \end{bmatrix}. \tag{3.17}$$

The first m rows of $\mathbf{U}$ are equal to the Jacobian divided by σ, which without loss of generality can be considered to be equal to 1, and the last row is equal to the transpose of the null vector. For optimally fault tolerant Jacobians, the elements of the null vector are given by (3.15), where for the following derivation we assume each element to be non-negative.

We now construct an orthogonal $\mathbf{U}$ that satisfies these constraints as follows: (1) The weighting coefficients w_i are sorted in an ascending order, and the columns of the Jacobian are permuted accordingly. (2) All the entries in the first column of the permuted Jacobian are set to 0, except for the first entry that is computed using the constraint that $\|\mathbf{u}_1\| = 1$. (3) All the entries in the second column of the permuted Jacobian are set to 0, except the first and second entry, where the first entry is calculated using constraint that $\mathbf{u}_1 \cdot \mathbf{u}_2 = 0$, and the second entry is calculated using the constraint that $\|\mathbf{u}_2\| = 1$. (4) The remaining columns are obtained in a similar way, using the properties of an orthogonal matrix, i.e., that $\|\mathbf{u}_i\| = 1$ and $\mathbf{u}_i \cdot \mathbf{u}_j = 0 \; \forall i, j$. The element in the pth row and qth column of the permuted Jacobian matrix $\mathbf{J}$ is now in the form

$$
\begin{aligned}
j_{pq} &= 0 && \text{where } p > q \\
j_{pq} &= -\sqrt{\sum_{i=p+1}^{n} w_i^2 / \sum_{i=p}^{n} w_i^2} && \text{where } p = q \\
j_{pq} &= \frac{w_p w_q}{\sqrt{\sum\limits_{i=p}^{n} w_i^2 \sum\limits_{i=p+1}^{n} w_i^2}} && \text{where } p < q
\end{aligned}
\tag{3.18}
$$

Finally, the columns of the Jacobian are permuted back so that the null vector of this isotropic Jacobian is

$$
\hat{\mathbf{n}} = \begin{bmatrix} \frac{w_1}{\sqrt{\sum\limits_{i=1}^{n} w_i^2}} & \frac{w_2}{\sqrt{\sum\limits_{i=1}^{n} w_i^2}} & \cdots & \frac{w_n}{\sqrt{\sum\limits_{i=1}^{n} w_i^2}} \end{bmatrix}^{\mathrm{T}}. \tag{3.19}
$$

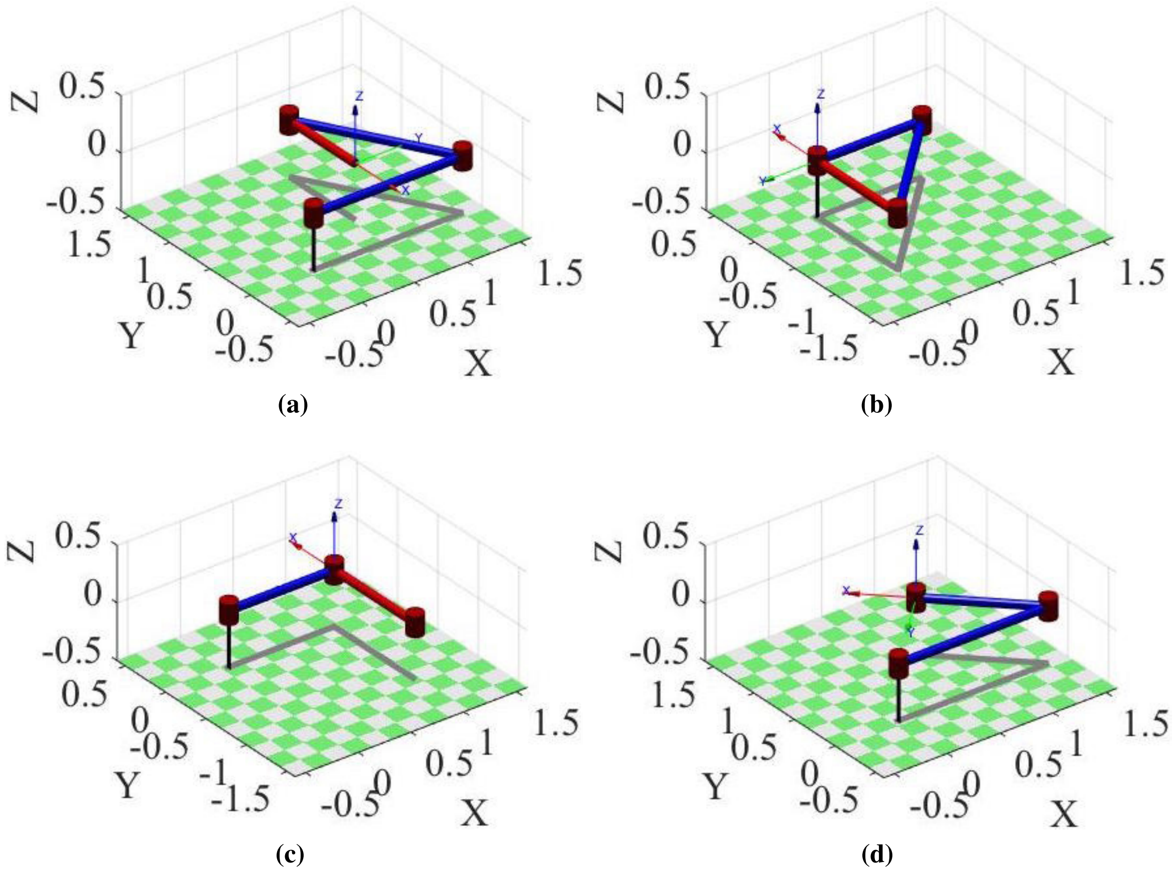
After the optimally fault tolerant Jacobians are constructed, the method developed in [35] can be applied to generate robot kinematic parameters from these Jacobians. It is important to note that the optimally fault tolerant Jacobian is locally optimal and so results in a robot design that is optimal at the design configuration. However, robots that have such configurations are more likely to have better performance throughout the workspace. After an optimally fault tolerant robot design is obtained, its workspace can be further evaluated using the method proposed in [37].

3.3.4 Results for Positioning Robots

(1) planar 3R robots

We consider four illustrative examples for this simple case, i.e., equal probability of joint failure and an impending failure in each joint. For the equal probability of failure case, the weighting coefficient is $\mathbf{w} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \end{bmatrix}$. The optimal robot is shown in Figure 3.1a, where the link length vector $\mathbf{l}$ is given by $\mathbf{l} = \begin{bmatrix} \sqrt{2} & \sqrt{2} & \sqrt{2/3} \end{bmatrix}$. This is the same as the result obtained in [34]. Now, consider the other extreme cases, i.e., one of the joints is known to have an imminent failure. The weighting coefficients for this case are $w_f = 1$ for the fth joint that is failing and $w_i = 0$ for the other two joints. The optimal robots for these cases are shown in Figure 3.1b-Figure 3.1d, for $f = 1$ to 3 respectively, with the corresponding link lengths given by: ${}^1\mathbf{l} = \begin{bmatrix} 1 & \sqrt{2} & 1 \end{bmatrix}$,

$^{2}\mathbf{l} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$ and $^{3}\mathbf{l} = \begin{bmatrix} \sqrt{2} & 1 & 0 \end{bmatrix}$ where we have appended the superscript of the failure joint to the link length vector $\mathbf{l}$. Note that when joint i's probability of failure is one, the end-effector of the robot must be located on this joint axis so that the contribution of joint i to the end-effector movement is 0 and its failure does not affect the end-effector motion.



(a)

(b)

(c)

(d)

Figure 3.1: The optimally fault tolerant planar 3R robots are shown for different joint failure probabilities. The optimal robot when all the joints are equally likely to fail is shown in (a). The optimal robots when each joint is failing are shown in (b)-(d) for joints 1 to 3, respectively. Note that for the robots in (b)-(d), the end-effector is located at the joint axis that is likely to fail. A robot designer can use these extremal cases to determine the desired link lengths based on the relative probability of failure for the different joint.

(2) spatial 4R robots

Similar to the planar 3R robots, we consider five illustrative examples for this case. Note that as described in [36], there exists a family of optimal robots that can realize a given Jacobian, because the linear velocity represented by a column of the Jacobian can be realized by a one-dimensional set of joint axes oriented circularly around the end effector. In [36], the entire family of optimal robot designs were parameterized, however, here we simply select one example design where all the twist angles are selected to be ±90°.

The optimal robots are shown in Figure 3.2, where Figure 3.2a is the case when the joints are equally likely to fail. The three singular values after joint i fails are ${}^i\sigma_1 =^i \sigma_2 = 1.00$, ${}^i\sigma_m = 0.50 \ \forall \ i$, which are also equal to the length of the semi axes of the manipulability ellipsoid after a failure. Figure 3.2b-Figure 3.2e are the cases when joint one to four is failing, respectively. Similarly to the 3R case, the end-effector is located on the joint axis that is failing in Figure 3.2b-Figure 3.2e so that there is no linear motion associated with the failing joint. Therefore, the manipulability ellipsoid after a failure is still a unit sphere, i.e., isotropic, the same as before a failure.

3.3.5 Results for Spatial 7DOF Robots for Positioning and Orienting

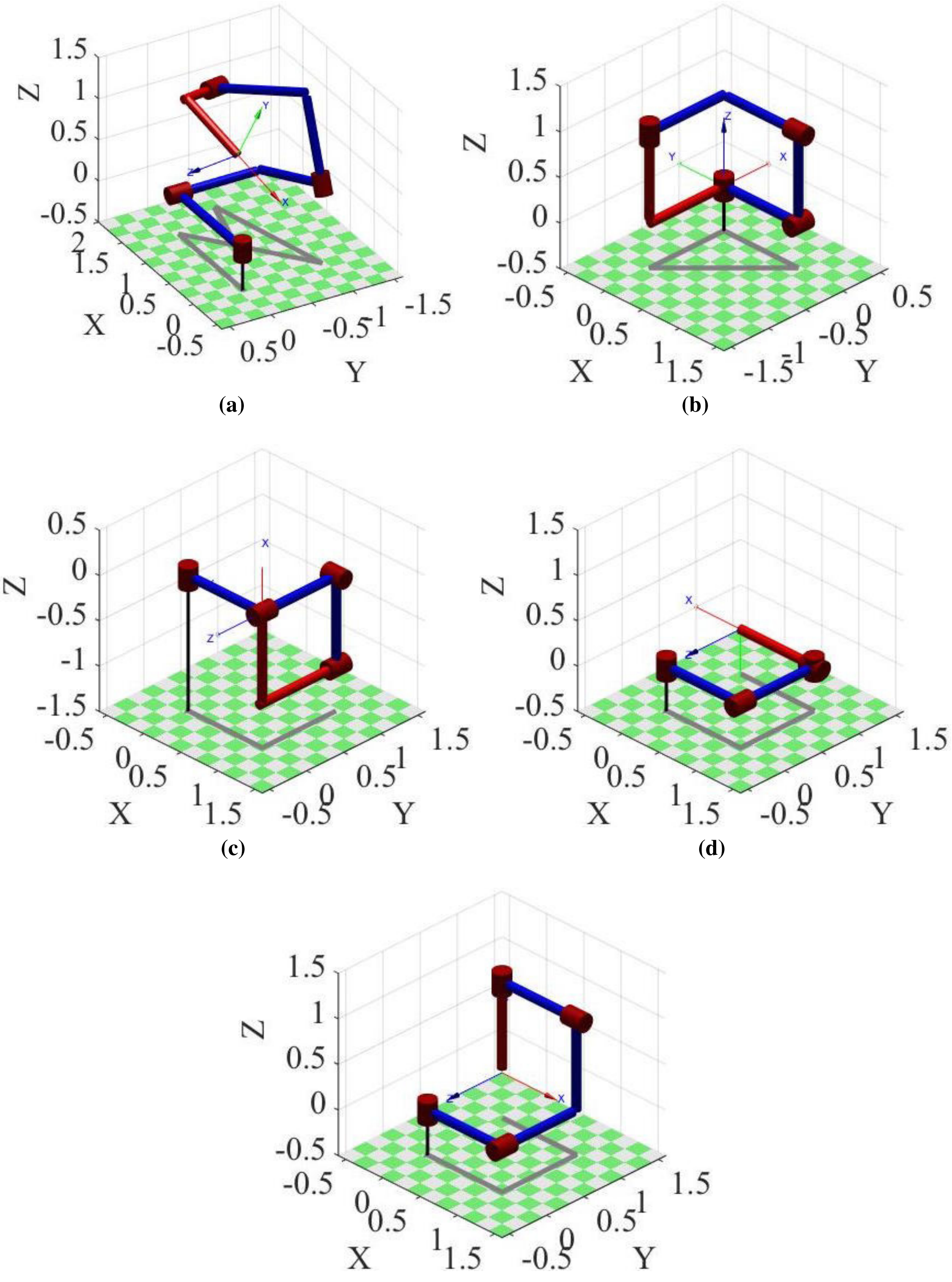
The optimally fault tolerant Jacobians of spatial 7DOF robots described by (3.18) can not be realized by rotational joints. This is true because for a rotational joint i, the associated column of a Jacobian is given by

$$\mathbf{j}_i = \begin{bmatrix} \mathbf{v}_i \\ \boldsymbol{\omega}_i \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{a}}_i \times \mathbf{p}_i \\ \hat{\mathbf{a}}_i \end{bmatrix} \tag{3.20}$$

where $\mathbf{v}_i$ and $\boldsymbol{\omega}_i$ are the linear velocity and angular velocity, respectively, $\hat{\mathbf{a}}_i$ is the unit vector of the rotational joint axis, and $\mathbf{p}_i$ is a position vector from the joint axis to the end-effector. Therefore, $\|\boldsymbol{\omega}_i\| = 1$ and the linear velocity must be orthogonal to the angular velocity, i.e., $\mathbf{v}_i \perp \boldsymbol{\omega}_i$. These constraints are not true for the Jacobian described by (3.18). In this section, we discuss modifications to the above approach so that spatial 7R robots can be designed.

For isotropic spatial 7R Jacobians (3.17) becomes

$$\mathbf{u}_i = \begin{bmatrix} \sqrt{\frac{3}{7}}\mathbf{v}_i \\ \sqrt{\frac{3}{7}}\boldsymbol{\omega}_i \\ \hat{n}_i \end{bmatrix} \tag{3.21}$$



(a)

(b)

(c)

(d)

(e)

Figure 3.2: The optimally fault tolerant spatial 4R robots are shown for different joint failure probabilities. The optimal robot when all the joints are equally likely to fail is shown in (a). The optimal robots when joints 1 to 4 are failing are shown in (b)-(e), respectively. Note that in (b)-(e), the end-effector is located on the joint axis that is failing. A robot designer can use these extremal cases to determine the desired link lengths based on the relative probability of failure for the different joint.

where the factor of $\sqrt{\frac{3}{7}}$ is due to the constraint that $\mathbf{U}$ is orthogonal. To be optimally fault tolerant, the elements of the null vector must satisfy (3.15). In order to enforce the constraints that $\|\boldsymbol{\omega}_i\| = 1$ and $\mathbf{v}_i \perp \boldsymbol{\omega}_i$, one can represent the ith column of the Jacobian as

$$\begin{aligned} \mathbf{j}_i &= \begin{bmatrix} \mathbf{v}_i \\ \hdashline \boldsymbol{\omega}_i \end{bmatrix} \\ &= \begin{bmatrix} \|\mathbf{v}_i\| \begin{bmatrix} \cos\alpha_i \sin\beta_i \cos\gamma_i + \sin\alpha_i \sin\gamma_i \\ \sin\alpha_i \sin\beta_i \cos\gamma_i - \cos\alpha_i \sin\gamma_i \\ -\cos\beta_i \cos\gamma_i \end{bmatrix} \\ \hdashline \begin{bmatrix} \cos\alpha_i \cos\beta_i \\ \sin\alpha_i \cos\beta_i \\ \sin\beta_i \end{bmatrix} \end{bmatrix}. \end{aligned} \tag{3.22}$$

where α_i, β_i and γ_i are the parameters defining joint i. Using (3.15) and the fact that $\mathbf{u}_i$ is a unit vector (because $\mathbf{U}$ is orthogonal) one can show that the norm of the linear velocity is given by

$$\|\mathbf{v}_i\| = \sqrt{\frac{4}{3} - \frac{7w_i^2}{3\sum\limits_{i=1}^{7} w_i^2}}. \tag{3.23}$$

By imposing the constraints that the rows of $\mathbf{U}$ are unit norm and orthogonal one can determine the values of the α_i's, β_i's and γ_i's to identify the optimally fault tolerant 7R manipulator Jacobian

Unfortunately, it is not possible to satisfy all these constraints for an arbitrary set of joint failure probabilities. In particular, the optimal 7R manipulator Jacobian for equal joint failure probabilities has not been identified [34]. However, it is possible to solve the set of nonlinear constraint equations using the Levenberg-Marquardt algorithm to find one set of weighting coefficients, i.e., failure probabilities, that is close to equally likely, $\mathbf{w} = \begin{bmatrix} 0.15 & 0.11 & 0.11 & 0.12 & 0.19 & 0.19 & 0.12 \end{bmatrix}$[5], where the associated 7R manipulator Jacobian is

$$\mathbf{J} = \begin{bmatrix} -0.38 & -1.02 & 0.52 & -0.39 & -0.10 & 0.48 & 0.69 \\ 0.04 & -0.30 & 0.39 & 0.79 & -0.70 & 0.59 & -0.79 \\ -0.91 & -0.16 & -0.84 & 0.57 & 0.52 & 0.41 & 0.01 \\ 0.68 & -0.28 & -0.79 & 0.44 & -0.65 & -0.03 & 0.74 \\ -0.66 & 0.51 & 0.56 & 0.66 & -0.39 & -0.55 & 0.65 \\ -0.31 & 0.81 & -0.23 & -0.61 & -0.66 & 0.83 & 0.17 \end{bmatrix} \tag{3.24}$$

3.3.6 Extension to Multiple Degrees of Redundancy and Joint Failures

(1) multiple degrees of redundancy

The above approach can be extended to the case of robots with multiple degrees of redundancy in a manner analogous to that in [30]. For the Jacobian of a robot with multiple degrees of redundancy, the matrix $\mathbf{U}$ in (3.16) can be decomposed into the following two matrices,

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{bmatrix} \tag{3.25}$$

where $\mathbf{U}_1$ is equal to the Jacobian divided by σ, and $\mathbf{U}_2$ is a matrix of $n - m$ orthonormal n-dimensional vectors in the null space of $\mathbf{J}$. Let $\hat{\mathbf{n}}_i$ be the ith column of $\mathbf{U}_2$, and its norm can be calculate as

$$\|\hat{\mathbf{n}}_i\| = \frac{\det({}^i\mathbf{J})}{\det(\mathbf{J})} = \frac{{}^i\sigma_1\, {}^i\sigma_2\, \cdots\, {}^i\sigma_m}{\sigma_1\, \sigma_2\, \cdots\, \sigma_m}. \tag{3.26}$$

Substituting (3.8) into (3.26), the relationship between the null space and the minimum singular value after a failure is given by

$$\|\hat{\mathbf{n}}_i\| = \frac{{}^i\sigma_m}{\sigma}. \tag{3.27}$$

[5] The elements of w do not add up to 1 due to rounding.

Equation (3.12) can be considered as a special case of this equation, in which each column has only one element. Likewise, the generalization of the constraint in (3.13) becomes

$$\|\hat{\mathbf{n}}_1\|^2 + \|\hat{\mathbf{n}}_2\|^2 + \cdots + \|\hat{\mathbf{n}}_n\|^2 = n - m. \tag{3.28}$$

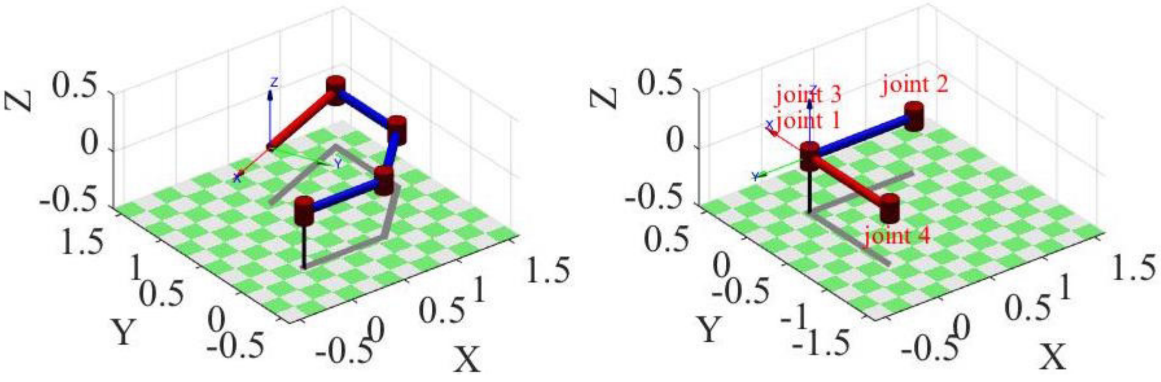
Similarly, applying the method of Lagrange multipliers, the null space of the optimally fault tolerant Jacobian for the robots with multiple degrees of redundancy needs to satisfy

$$\|\hat{\mathbf{n}}_i\| = w_i \sqrt{\frac{n-m}{w_1^2 + w_2^2 + \cdots + w_n^2}} \leqslant 1. \tag{3.29}$$

For the cases where $\|\hat{\mathbf{n}}_i\| > 1$, the value of $\|\hat{\mathbf{n}}_i\|$ is set to be 1, and the norms of the other columns of the null space can be identified according to the weighting coefficients.

(2) multiple joint failures

For the case of multiple joint failures, the minimum singular value after these failures is not only affected by the norm of the null space matrix columns, but also by the angles between them. This is best illustrated through a simple example. Consider the case of a planar 4R manipulator that experiences two simultaneous joint failures. If all joints are equally likely to fail then all possible pairs of joint failures are also equally likely to fail. This means that the optimal design is one where the remaining two columns of the Jacobian are as orthogonal as possible after the locking of any two arbitrary joints. This results in a design like the one shown in Figure 3.3a, which is the same as the optimal planar 4R robot obtained in [30]. The situation of unequal joint failure probabilities, which is not discussed in [30], requires the consideration of all six possible combinations of pairs of joint failures. Consider the extremal case where it is known which pair of joints will fail, e.g., joints one and three. Similar to the one joint failure case, the end-effector must be located on the joint axes of the two joints that are going to fail, as shown in Figure 3.3b.



(a) **(b)**

Figure 3.3: The optimally fault tolerant planar 4R robot designs for two simultaneous joint failures are shown. In (a) all joints are equally likely to fail. In (b) joint 1 and joint 3 are guaranteed to fail, so that their axes are coincident with the end effector.

3.4 Designing For Multiple Cases of Joint Failure Probabilities

3.4.1 Problem Formulation

In the previous section, a method for designing an optimally fault tolerant robot for any set of joint failure probabilities was presented. However, joint failure probabilities are usually changing while performing a task. In particular, one common case is that a robot is designed so that all the joints are equally likely to fail when manufactured, however, during use one of the joints becomes more likely to fail either due to original component variation or the characteristics of the tasks. Unfortunately, one can not guarantee at design time which joint is the one that is more likely to fail, so that all cases should be considered. This scenario is discussed in this section.

When all the joints are equally likely to fail, one would like to optimize $\frac{1}{n}({}^1\sigma_m + {}^2\sigma_m + \cdots + {}^n\sigma_m)$. When joint i is failing, one would like to optimize ${}^i\sigma_m$. To consider all of these cases simultaneously, the objective function is the sum of the equally likely case along with all of the ${}^i\sigma_m$'s. The design problem can be stated as finding the optimal robot kinematic parameters and the $n+1$ different optimal configurations for each component of the objective function that will maximize the overall objective function. It can be mathematically expressed as

$$
\begin{aligned}
\max\ \mathcal{F} = \quad & \tfrac{1}{n}({}^1\sigma_{m,c_0} +^2 \sigma_{m,c_0} + \cdots +^n \sigma_{m,c_0}) + \\
& {}^1\sigma_{m,c_1} +^2 \sigma_{m,c_2} + \cdots +^n \sigma_{m,c_n} \\
\text{s.t.} \quad & l_1 + l_2 + \cdots + l_n = l
\end{aligned}
\tag{3.30}
$$

where the notation ${}^f\sigma_{m,c_i}$, indicates the minimum singular value of the Jacobian at the configuration c_i for a failure in joint f. Here, the $n+1$ different configurations may be at different locations, so the robot would need to adjust the relative position between its base and the task once a joint failure is imminent. In cases where one does not want to change the location of the task, one can impose the additional constraint that all the $f(c_i)$'s are equal where $f()$ denotes the forward kinematics function of the robot.

There are many different ways of solving the above constrained optimization problem. The approach used here is to first discretize the space of kinematic parameters to identify candidate robot designs. For each candidate design we discretize its workspace and find the configuration that optimizes each component of the objective function for every workspace location. This approach makes it easy to determine the optimal configurations with or without the constraint that the $f(c_i)$'s are equal.

3.4.2 Illustrative Example for Planar 3R Robots

The above optimization problem is solved for a simple 3R robot to illustrate the properties of the various robot designs. As described above, the kinematic parameters, i.e, the three link lengths are discretized in the interval [0 1] using an increment of 0.02 under the constraint that the overall length is equal to one. There are no joint limits imposed, so that the resulting circular workspaces is then uniformly sampled using 5000 points. At each workspace location, all the self-motion manifolds are calculated, and each component of the objective function $\mathcal{F}$ in (3.30), i.e., $\frac{1}{3}({}^1\sigma_{m,c_0} +^2 \sigma_{m,c_0} +^3 \sigma_{m,c_0})$, ${}^1\sigma_{m,c_1}$, ${}^2\sigma_{m,c_2}$ and ${}^3\sigma_{m,c_3}$ are evaluated along all the self-motion manifolds. The maximum values of each component of the objective function are saved, and the sum of them is the optimal objective function value at each workspace location. Finally, the maximal objective function value over all the sampled points in the workspace is saved as the optimal objective function value for each set of link lengths.

Figure 3.4 shows the value of the objective function $\mathcal{F}$ in (3.30) for each set of link lengths. These results are shown for the version of the optimization that includes the constraint that all the $f(c_i)$'s are equal. However, it is interesting to point out that if one solves the unconstrained problem, these same configurations will be part of a larger set of equally optimal configurations. Thus one does not lose any optimality by constraining the task to be at the same location before and after a joint failure. From Figure 3.4 one can also see that there is a family of optimal robots that have the same value of the objective function. All these optimal robots have the property that the third link length is equal to 0.4, i.e. $l_1 + l_2 = 0.6$.

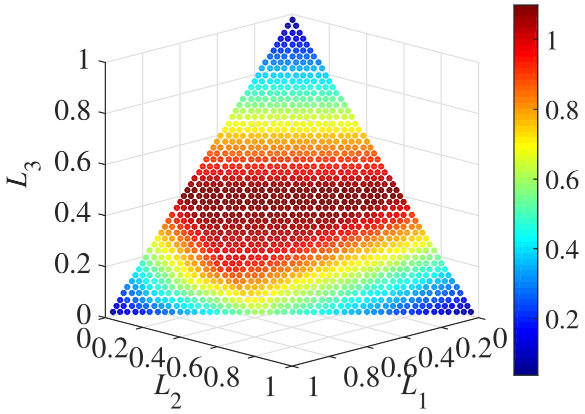
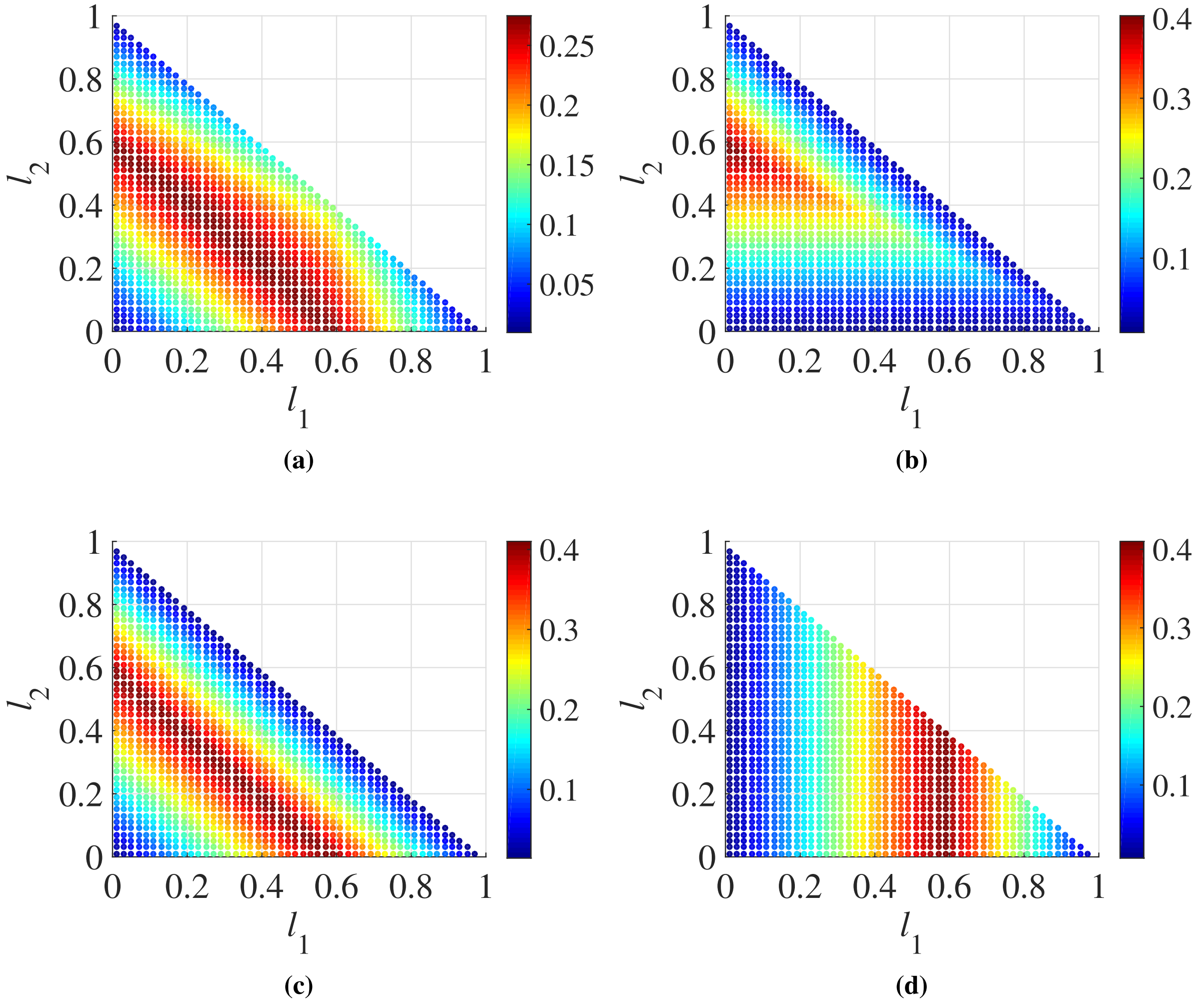


Figure 3.4: The objective function values $\mathcal{F}$ for all possible link lengths, which have been normalized to equal one, is shown. Note that there is a family of optimal robots that have the same value of the objective function $\mathcal{F}$ along the line where $l_3 = 0.4$.

To distinguish between the robots in this optimal family, the values for each component of the objective function $\mathcal{F}$ for all possible link lengths are shown in Figure 3.5. These figures are obtained by the same method as describe above, only replacing the objective function by each component. Because $l_3 = 1 - l_1 - l_2$, the value of each component of the objective function is only plotted as a function of l_1 and l_2. From Figure 3.5, one can see that the optimization of $\mathcal{F}$ also



(a)

(b)

(c)

(d)

Figure 3.5: The value of each component of the objective function $\mathcal{F}$ is plotted for all the sets of link lengths, with the first component, i.e., $\frac{1}{3}({}^1\sigma_m +^2 \sigma_m +^3 \sigma_m)$, shown in (a) and the components for the individual joint failures, i.e., ${}^1\sigma_m$, ${}^2\sigma_m$ and ${}^3\sigma_m$, shown in (b)-(d), respectively. Note that in (a), for the robots with link lengths $l_1 + l_2 < 0.6$, the optimal value of $\frac{1}{3}({}^1\sigma_m +^2 \sigma_m +^3 \sigma_m)$ is a function of the sum $l_1 + l_2$. In (b), the functional dependence of the optimal value of ${}^1\sigma_m$ depends on whether the link lengths are above or below the line defined by $l_1 = 1 - (\sqrt{2}/2 + 1)l_2$. If below, then ${}^1\sigma_m$ is only a function of l_2 and if above it is a function of the sum $l_1 + l_2$. In (c), for all the robots, the optimal value of ${}^2\sigma_m$ is only a function of the sum $l_1 + l_2$. In (d), for all the robots, the optimal value of ${}^3\sigma_m$ is only a function of l_1.

results in the optimization of the equal failure probability case, i.e., $\frac{1}{3}({}^1\sigma_m +^2 \sigma_m +^3 \sigma_m)$, shown in (a) and the joint two failure case, i.e., ${}^2\sigma_m$, shown in (c) individually. Unfortunately, the joint one and three cases, i.e., ${}^1\sigma_m$ and ${}^3\sigma_m$, shown in (b) and (d), respectively, are directly competing with each other. Obviously, one can select the weighting between these two cases based on failure probabilities, but if failures in joint one and three are equally likely, then the optimal link lengths are when $l_1 = l_2 = 0.3$ (with $l_3 = 0.4$). It turns out that the optimal configurations for each of the four components of $\mathcal{F}$ are very close to each other, i.e., optimization of the equal probability and joint two failure cases result in $\theta = [59°, 39°, 115°]$ (shown in Figure 3.6) and for the joint one and three failure cases result in $\theta = [55°, 44°, 111°]$. This is fortunate, because this means that the robot only needs to slightly adjust its configuration when the joint failure probabilities are changing. However, this will not always be true, as shown in Figure 3.7, which depicts the optimal configurations for the robot with $\mathbf{l} = [0.43, 0.29, 0.28]$.

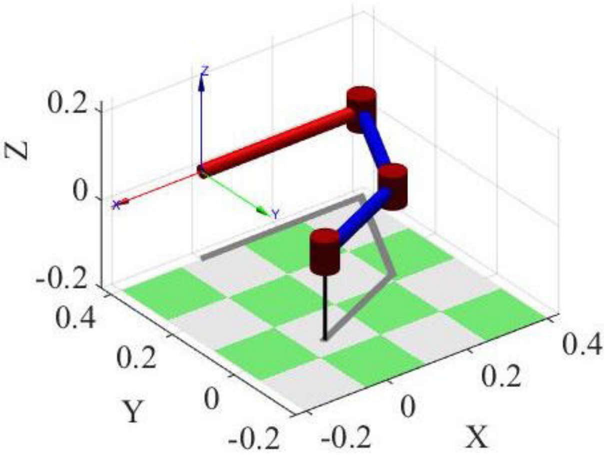


Figure 3.6: The optimally fault tolerant planar 3R robot where $\mathbf{l} = [0.30, 0.30, 0.40]$ is shown at the optimal configuration for the equal failure probability case, i.e., $\frac{1}{3}({}^1\sigma_m +^2 \sigma_m +^3 \sigma_m)$.

3.5 Chapter Summary

This chapter presented methods to design optimally fault tolerant robots for different joint failure probabilities. It was shown that for an arbitrary set of joint failure probabilities, the optimal

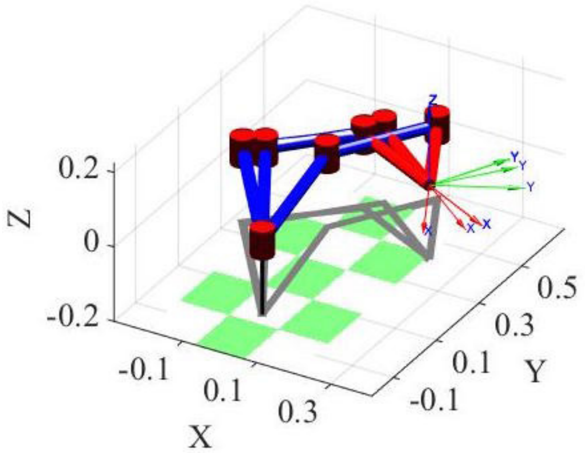


Figure 3.7: The optimal configurations for each component of the objective function are shown for the robot with $\mathbf{l} = [0.43, 0.29, 0.28]$. Note that the configuration for optimizing one of the cases is quite different from the others, which was not the case for the robot depicted in Figure. 3.6.

null space satisfies $|\hat{n}_i| = w_i/\sqrt{w_1^2 + w_2^2 + \cdots + w_n^2}$, where the ratio of each element is equal to the ratio of the failure probabilities of the different joints. Based on the optimal null space, isotropic Jacobians for an arbitrary number of joints are constructed, and the kinematic parameters of the robots are generated from these optimal Jacobians. In addition, a method for designing an optimal spatial 7R manipulator Jacobian is derived, and it is shown that such optimal 7R designs do not exist for all possible joint failure probabilities. An example of such a case is that of equal joint failure probabilities. However, an optimal Jacobian for nearly equal joint failure probabilities is given. Furthermore, a method for designing optimally fault tolerant robots for multiple cases of joint failure probabilities is introduced and the design of optimal planar 3R robots is given as an illustrative example. It was shown that there exists a family of optimal robots whose last link length is equal to 40% of the total link length. This family of robots are also shown to be optimal for the case of equally likely joint failures, as well as failures in joint two. However, the optimization for failures in joints one and three are directly competing, so that one must select a particular design based on the failure probabilities of these joints.

Chapter 4

Maximizing Probability of Task Completion with Locked Joint Failures

4.1 Chapter Overview

As described in Subsection 1.3.3, in the studies of fault tolerant motion planning, the end-effector tasks can be classified into three categories: point-to-point tasks, path tracking tasks and trajectory tracking tasks. The first type of task, i.e., point-to-point tasks, is studied in this chapter. The existing fault tolerant motion planning algorithm for point-to-point tasks is to constraint the robot configuration to the intersection of all bounding boxes enclosing the self-motion manifolds of each task point. However, when there are many task points or task points are far away from each other, there usually does not exist an intersection between the self-motion manifolds of all task points. Furthermore, even though there exists an intersection, the robot may not be able to reach all the task points when restricted to this intersection. In both these two cases, one can not guarantee a fault tolerant joint trajectory. Therefore, similar to the reliability map for path tracking tasks, this chapter studies the fault tolerant motion planning problem for point-to-point tasks from a probability point of view. Specifically, given a starting configuration in the joint space and a set of ordered tasks points in the task space, one needs to plan a trajectory from the starting configuration to these tasks points in the specified order that maximizes the probability of completing the tasks with locked joint failures. The main contributions of this chapter are as follows: (1) the failure probability of an arbitrary joint path is identified; (2) the method of planning the trajectory with minimum probability of task failure is proposed.

4.2 Calculating the Probability of Task Failure for an Arbitrary Joint Trajectory

4.2.1 Self-motion Manifolds

Given a set of joint angles $\boldsymbol{\theta}$, the end-effector position of the robot $\mathbf{x}$ can be easily obtained by the following forward kinematics equation,

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

However, the inverse kinematics of a robot is much more complicated. In particular, for redundant robots, there may exist an infinite number of joint angles for a given end-effector position due to the extra DOFs. The self-motion manifolds of this task point consist of all these joint angles. When the robot moves along these self-motion manifolds, the end-effector of the robot stays at the same position. It is important to note that the dimension of the self-motion manifold is equal to the degree of redundancy.

For one degree of redundancy, arguably the most common way to calculate the self-motion manifolds of a task point is to move along the null vector of the Jacobian with a small step to reach a new configuration on the self-motion manifold, and repeat this operation. This works because the null vector is tangent to the self-motion manifold.

4.2.2 Joint Space Division

To efficiently compute the failure probability of a trajectory, one can first divide the joint space up into regions of different "failure scenarios", that correspond to specific joints failing and whether that failure prevents the robot from reaching the desired task points. The boundaries of these regions correspond to the limits on the ranges of the self-motion manifolds. These boundaries can be easily identified because the corresponding element of the null vector is zero.

It is important to point out that after each task point is reached, the joint space needs to be re-divided by different boundaries. This is true because after one of the task points is reached, one only needs to check the reachability of the remaining task points that have not been reached. Therefore, the joint space is divided by the boundaries of the self-motion manifolds intersections of all the remaining task points that have not been reached. These intersections are updated after each task point is reached. A simple two-dimensional example with two task points is shown in Figure 4.1 to illustrate the concept of regions. The blue and green curves are the self-motion manifolds of task points 1 and 2, respectively. In Figure 4.1a, the robot moves from the start configuration to task point 1, and one needs to check whether the robot is still able to reach both task point 1 and 2 after an arbitrary failure. The dashed lines are the θ_1 boundaries of the self-motion manifolds intersections of task point 1 and 2, and the dash-dotted lines are the θ_2 boundaries. The entire joint space is divided into 9 regions by these boundaries. After task point 1 is reached, the robot moves towards task point 2, and one only needs to check the reachability of task point 2. In Figure 4.1b, the dashed lines are the θ_1 boundaries of the self-motion manifold of task point 2, and the dash-dotted lines are the θ_2 boundaries. The entire joint space is re-divided into 9 regions by these boundaries.

One now needs to categorize each of these regions based on their failure scenarios. For example, consider region 1 in Figure 4.1a where regardless of whether joint 1 or joint 2 fails, task points 1 and 2 are no longer reachable. Because if either joint 1 or joint 2 fails, then the robot can only move either vertically or horizontally, respectively. Thus for region 1 in Figure 4.1a we define two failure scenarios: A, which is when joint 1 fails in this region and B, which is when joint 2 fails in this region. In either scenario, it does not matter whether the other joint fails or not. Now consider region 5 in Figure 4.1a, which is the intersection of the bounding boxes of the two self-motion manifolds. If joint 1 is locked in any position in this region the robot is still able to reach both task points 1 and 2, unless joint 2 is also locked in the subsequent post-failure trajectory. This is also true if joint 2 is locked in any position in this region, unless joint 1 is also locked in the subsequent post-failure trajectory. Therefore, the failure scenarios of region 5 in Figure 4.1a are: C, which is when joint 1 fails in this region, and joint 2 also fails in the subsequent post-failure trajectory and D, which is when joint 2 fails in this region, and joint 1 also fails in the subsequent post-failure

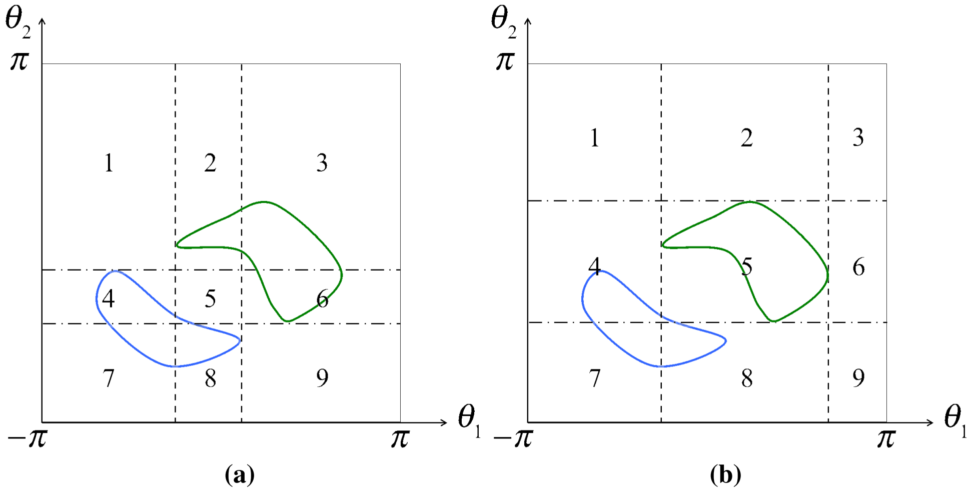


Figure 4.1: The joint space is divided into different regions according to the boundaries of the self-motion manifolds intersections of all the remaining task points that have not been reached. The blue and green curves are the self-motion manifolds of task point 1 and 2, respectively. The robot moves from the start configuration to task point 1, and entire joint space is divided into 9 regions by the boundaries of the self-motion manifolds intersections of task point 1 and 2 in (a). The dashed lines are the θ_1 boundaries of the self-motion manifolds intersections of task point 1 and 2, and the dash-dotted lines are the θ_2 boundaries. After task point 1 is reached, the robot moves towards task point 2, and the entire joint space is re-divided into 9 regions by the boundaries of the self-motion manifold of task point 2 in (b). The dashed lines are the θ_1 boundaries of the self-motion manifold of task point 2, and the dash-dotted lines are the θ_2 boundaries.

trajectory. The failure scenarios of the regions in Figure 4.1b can be identified in a similar way, where only the reachability of task point 2 needs to be considered. It is important to note that the failure scenarios will increase as the number of DOFs increases, however the failure scenarios can be enumerated in an analogous manner.

Table 4.1 lists the failure scenarios of each region both before and after task point 1 has been reached. In each situation, according to the failure scenarios, the regions can be classified into 4 categories, as shown in the table, and the regions belonging to the same category are shown in Figure 4.2 using the same color. Obviously, in Figure 4.2a region 5 has the lowest probability of task failure because it is the intersection of the bounding boxes of the two self-motion manifolds. This is also true in Figure 4.2b because region 5 is the bounding boxes of the self-motion manifold of task point 2. In contrast, the regions belonging to Category 1 both before and after task point 1 has been reached have the highest probability of task failure. Note that higher dimensional joint spaces can be divided and categorized in a similar manner.

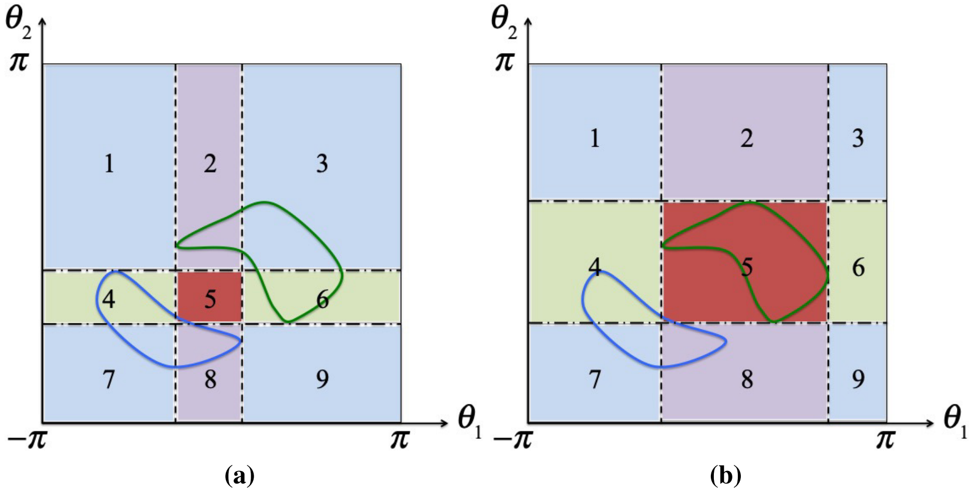


Figure 4.2: The joint space is classified into four categories according to the failure scenarios of each region both before task point 1 has been reached in (a) and after task point 1 has been reached in (b), where each category is shown in a different color.

Table 4.1: The task failure scenarios of the different regions in the joint space

category	regions	failure scenarios	
		task 1 not reached	task 1 reached
1	1, 3, 7, 9	A, B	A, B
2	2, 8	B, C	B, C
3	4, 6	A, D	A, D
4	5	C, D	C, D

Task failure scenarios:
A: joint 1 fails in this region, regardless of whether joint 2 is healthy or locked in the subsequent post-failure trajectory.
B: joint 2 fails in this region, regardless of whether joint 1 is healthy or locked in the subsequent post-failure trajectory.
C: joint 1 fails in this region, and joint 2 also fails in the subsequent post-failure trajectory.
D: joint 2 fails in this region, and joint 1 also fails in the subsequent post-failure trajectory.

4.2.3 Failure Probability Calculation

For an arbitrary joint trajectory, it may go through different regions with different failure scenarios. To calculate its probability of task failure, the trajectory needs to be divided into several pieces according to the regions. For example, the joint trajectory in Figure 4.3a goes through region 1 and 4 when the robot moves from the start configuration to task point 1. After task point 1 is reached, the joint space is re-divided, and the joint trajectory in Figure 4.3b goes through region 4 and 5 when the robot moves towards task points 2. Therefore, the entire trajectory is divided into 4 pieces. Based on the failure scenarios of each region, the probability of task failure is calculated for each piece. The sum of the individual probabilities is the probability of task failure for the entire trajectory.

As shown in Table 4.1, there are a total of four types of failure scenarios for a planar 2R robot, and the probabilities of these four types are derived as follows. For simplicity, it is assumed that the maximum velocity of each joint is equal, and each joint moves along the path in a constant velocity v, which is some percentage of the maximum velocity. As each joint of the robot moves independently, for a straight line joint trajectory, the time needed to finish this trajectory Δt can be

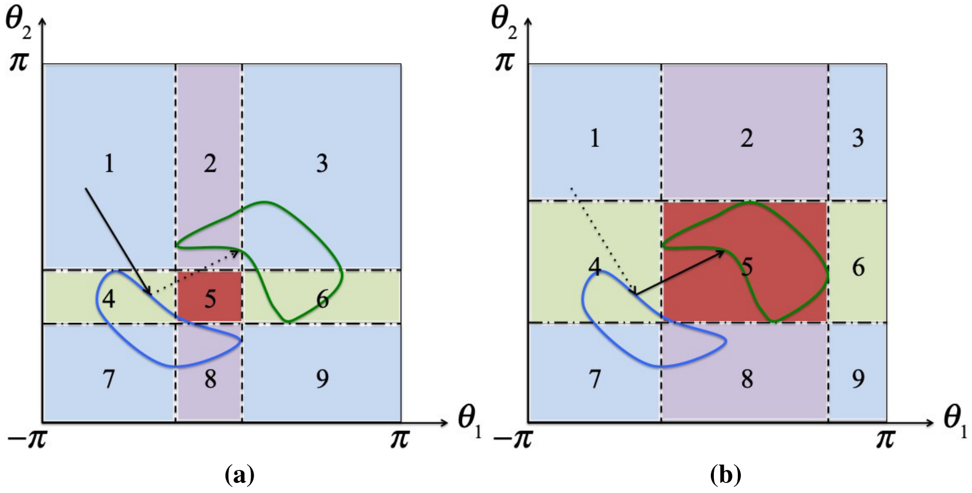


Figure 4.3: The joint trajectory is divided into four pieces according to the regions. When the robot moves from the start configuration to task point 1, the joint trajectory goes through region 1 and 4 in (a). After task point 1 is reached, the joint space is re-divided, and the joint trajectory goes through region 4 and 5 in (b).

calculated as

$$\Delta t = \frac{max(\Delta\theta_1, \Delta\theta_2, ..., \Delta\theta_n)}{v}, \tag{4.2}$$

where $\Delta\theta_i$, $i = 1, 2, ..., n$ is the distance that joint i moves.

For failure scenario A, if joint one fails in any position, then the task can not be completed. Thus, the probability of task failure in scenario A, denoted P_A, is equal to the probability of joint 1 failing in this region, i.e.,

$$P_A = e^{\lambda_1 t_0} - e^{\lambda_1 t_1} = e^{\lambda_1 t_0} - e^{\lambda_1 (t_0 + \Delta t_1)}, \tag{4.3}$$

where λ_1 is the failure rate of joint 1, t_0 is the time when the robot moves into this region, and Δt_1 is the time period spent in the region. Similarly, the probability of task failure in scenario B, denoted P_B, is

$$P_B = e^{\lambda_2 t_0} - e^{\lambda_2 t_1} = e^{\lambda_2 t_0} - e^{\lambda_2 (t_0 + \Delta t_1)}, \tag{4.4}$$

where λ_2 is the failure rate of joint 2. For failure scenario C, the post-failure trajectory to reach the task points depends on the position where joint 1 is locked, i.e., the time period spent in the post-failure trajectory Δt_2 is a function of the time when joint 1 fails. Therefore, the probability of task failure in scenario C, denoted P_C, is

$$P_C = \int_{t_0}^{t_0+\Delta t_1} \lambda_1 e^{-\lambda_1 t} \times \left(e^{\lambda_2 t} - e^{\lambda_2 (t+\Delta t_2)}\right) dt, \tag{4.5}$$

where the term $\lambda_1 e^{-\lambda_1 t}$ is the failure density function of joint 1 at time t, and the second factor is the failure probability of joint 2 failing in the post-failure trajectory when joint 1 fails at time t. Similarly, the probability of task failure in scenario D, denoted P_D, is

$$P_D = \int_{t_0}^{t_0+\Delta t_1} \lambda_2 e^{-\lambda_2 t} \times \left(e^{\lambda_1 t} - e^{\lambda_1 (t+\Delta t_2)}\right) dt. \tag{4.6}$$

The probabilities of failure scenarios A and B can be directly calculated using (4.3) and (4.4) because the value of Δt_1 can be computed. However, the value of Δt_2 depend on when the failure occurs. As a result, to estimate the probability of task failure in scenario C or D, the entire trajectory is divided into small segments where all post-failure trajectories that occur in a segment are considered constant.

Figure 4.4 shows examples of estimating the probabilities of failure scenarios C and D in a small segment shown in red. Note that task point 1 is not reached yet in Figure 4.4, and the failure scenarios of this small segment are failure scenarios C and D, shown in (a) and (b), respectively. In failure scenario C, the new trajectory to reach both task points 1 and 2 after joint 1 fails in this segment is the dashed line in Figure 4.4a. Therefore, the probability of failure scenario C in the segment, denoted ΔP_C, is approximately equal to the probability of joint 1 failing in this segment times the probability of joint 2 failing in the post-failure trajectory, i.e.,

$$\begin{aligned}\Delta P_C &= (e^{-\lambda_1 t_0} - e^{\lambda_1 t_1})(e^{-\lambda_2 t_1} - e^{\lambda_2 t_2}) \\ &= (e^{-\lambda_1 t_0} - e^{\lambda_1 (t_0+\Delta t_1)})(e^{-\lambda_2 (t_0+\Delta t_1)} - e^{\lambda_2 (t_0+\Delta t_1+\Delta t_2)}),\end{aligned} \tag{4.7}$$

where t_0 is the time when the robot moves into this segment, t_1 is the time when the robot moves out of this segment, and t_2 is the time when the robot reaches task point 2. Δt_1 is the time period spent in this segment. Δt_2 is the time period spent in the post failure trajectory, which can be calculated by $\Delta t_2 = \Delta\theta_{2,\text{post}}/v$, where $\Delta\theta_{2,\text{post}}$ is the distance joint 2 moves in the post failure trajectory, as shown in Figure 4.4a.

Similarly, in failure scenario D, the new trajectory to reach both task points 1 and 2 after joint 2 fails in this segment is the dashed line in Figure 4.4b. Therefore, the probability of failure scenario D in the segment, denoted ΔP_D, is approximately equal to the probability of joint 2 failing in this segment times the probability of joint 1 failing in the post-failure trajectory, i.e.,

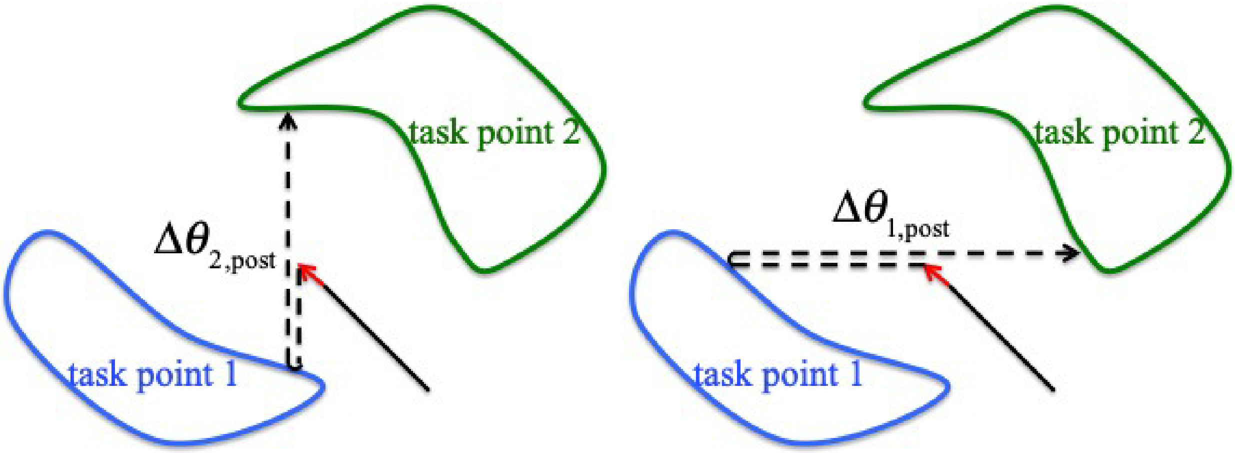
$$\begin{aligned}\Delta P_D &= (e^{-\lambda_2 t_0} - e^{\lambda_2 t_1})(e^{-\lambda_1 t_1} - e^{\lambda_1 t_2}) \\ &= (e^{-\lambda_2 t_0} - e^{\lambda_2 (t_0+\Delta t_1)})(e^{-\lambda_1 (t_0+\Delta t_1)} - e^{\lambda_1 (t_0+\Delta t_1+\Delta t_2)}),\end{aligned} \tag{4.8}$$

where Δt_2 can be calculated by $\Delta t_2 = \Delta\theta_{1,\text{post}}/v$, where $\Delta\theta_{1,\text{post}}$ is the distance joint 1 moves in the post failure trajectory, as shown in Figure 4.4b. For the segment whose path already reaches task point 1, the new trajectories after joint 1 or 2 fails in this segment only need to reach task point 2, as shown in Figure 4.5.

4.3 Planning the Trajectory with Minimum Probability of Task Failure

4.3.1 Overview

Once a method for estimating the probability of task failure is determined, it is desired to find the trajectory with minimum probability of task failure for a given set of task points. For simplicity, it is assumed that only one joint is going to fail during the motion, i.e., only failure scenario A and B will happen. It makes sense because the failure probability in scenario C and D is relatively small compared with the failure probability in scenario A and B, which can be ignored when checking the failure probability of a joint trajectory. Based on this assumption, there are two cases for identifying the optimal trajectory with minimum probability of task failure. The first one is that there exists an intersection (or intersections) between the self-motion manifold


task point 2
$\Delta\theta_{2,\mathrm{post}}$
task point 1
task point 2
$\Delta\theta_{1,\mathrm{post}}$
task point 1

(a) **(b)**

Figure 4.4: The method of how to estimate the probability of failure scenarios C and D are shown. In (a), it is assumed that regardless of where joint 1 fails in the small segment shown in red, the post-failure trajectory is the dashed line with distance $\Delta\theta_{2,\mathrm{post}}$. The probability of failure scenario C in this small segment is equal to the probability of joint 1 failing in this segment times the probability of joint 2 failing in the post-failure trajectory. Similarly, the dashed line in (b) with distance $\Delta\theta_{1,\mathrm{post}}$ is the new trajectory after joint 2 fails in the segment. The probability of failure scenario D in this small segment is equal to the probability of joint 2 failing in this segment times the probability of joint 1 failing in the post-failure trajectory.

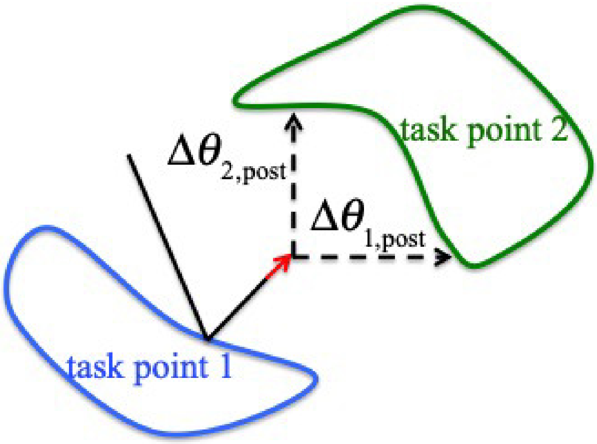

$\Delta\theta_{2,\mathrm{post}}$
task point 2
$\Delta\theta_{1,\mathrm{post}}$
task point 1

Figure 4.5: In the cases where task point 1 is reached, the new trajectories to reach task point 2 after joint 1 or 2 fails in the segment are shown as $\Delta\theta_{2,\mathrm{post}}$ and $\Delta\theta_{1,\mathrm{post}}$, respectively.

bounding boxes of all the task points, and there exists self-motion manifolds of task point i in the intersection(s) between the self-motion manifold bounding boxes of task point i, $i+1$, ..., n, where $i = 1, 2, ..., n-1$. If the self-motion manifolds of the task points do not satisfy the above two conditions in Case 1, then these situations all belong to Case 2.

A simple two-dimensional example with two task points is shown in Figure 4.6 and Figure 4.7 to illustrate these two cases, respectively. In Figure 4.6, the bounding boxes of the self-motion manifolds of task point 1 and 2 intersect each other, and there exists the self-motion manifold of task point 1 in the intersection, which belongs to Case 1. In this case, the optimal trajectory with minimum probability of task failure is the optimal trajectory from the start configuration to the intersection of the bounding boxes. This is true because as long as the robot reaches the intersection, regardless of which joint fails, the robot can always reach task point 1 (recalling that only one joint is going to fail). After task point 1 is reached in the intersection, the robot is already inside the bounding box of the self-motion manifold of task point 2, so the robot can always reach task point 2 after an arbitrary failure. The details on how to find the optimal trajectory from the start configuration to the intersection of the bounding boxes will be discussed in the next subsection.

Figure 4.7 shows two different situations belonging to Case 2. The bounding boxes of the self-motion manifolds of task point 1 and 2 shown in Figure 4.7a intersect each other. However, if the robot is restricted to this intersection, the robot is not able to reach task point 1. In Figure 4.7b, there does not exist an intersection between the bounding boxes of the self-motion manifolds of task point 1 and 2. In this case, to find the optimal trajectory with minimum probability of task failure, all the configurations along the self-motion manifold of task point 1 need to be checked. For each configuration of task point 1, the optimal trajectory from the start configuration to this specific configuration and the optimal trajectory from this specific configuration to the bounding box of the self-motion manifold of task 2 need to be identified, respectively. The total failure probability of this trajectory is equal to the sum of the failure probabilities of these two pieces. The optimal trajectory from start configuration to task point 1 and 2 can be eventually determined by choosing the trajectory with smallest failure probability among all the trajectories. The details on how to find the optimal trajectory from the start configuration to a configuration of task point 1 and the optimal trajectory from this configuration to the bounding box of the self-motion manifold of task 2 will be discussed in Subsection 4.3.3.

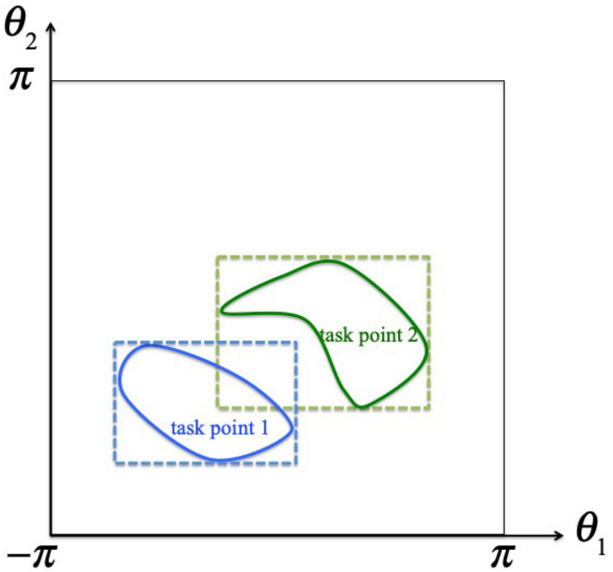


Figure 4.6: An example of Case 1 is shown. There exists an intersection between the self-motion manifold bounding boxes of task point 1 and 2, and there exists self-motion manifold of task point 1 in the intersection.

4.3.2 Planning the Optimal Trajectory in Case 1

In Case 1, the optimal trajectory with minimum probability of task failure is the optimal trajectory from the start configuration to the intersection between the bounding boxes. To illustrate the planning algorithm more clearly, a two-dimensional joint space is used as an example, and the optimal trajectory in a higher dimensional joint space can be identified in an analogous manner. As described in the above section, the joint space is divided into different regions based on the intersection. For a two-dimensional joint space, there are three sub-cases according to the region of the start configuration. Sub-case 1 is the start configuration is already in the intersection, where the robot can pick an arbitrary trajectory in the intersection, because the failure probabilities of all these trajectories are all equal to zero.

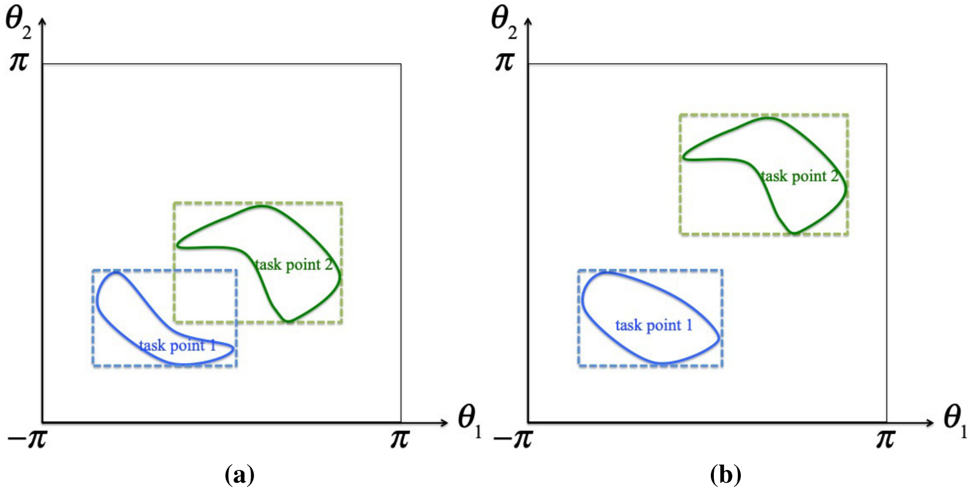


Figure 4.7: Two situations belonging to Case 2 are shown. In (a), there exists an intersection between the self-motion manifold bounding boxes of task point 1 and 2, but there does not exist self-motion manifold of task point 1 in the intersection. In (b), there even does not exist an intersection between the self-motion manifold bounding boxes of task point 1 and 2.

Sub-case 2 is when the start configuration is in a region whose failure scenario is only A or only B, i.e., region 2, 4, 6 or 8 as shown in Figure 4.8. In this sub-case, the optimal trajectories are the trajectories that satisfy $max(\Delta\theta_1, \Delta\theta_2)$ is equal to the θ_1 distance (region 4, 6) or θ_2 distance (region 2, 8) from the start configuration (the red star in Figure 4.8) to the intersection between the bounding boxes, which are the black trajectories in Figure 4.8, where $\Delta\theta_1$ and $\Delta\theta_2$ are the distance of the trajectory in θ_1 and θ_2. One can pick the optimal trajectory from these potential trajectories based on other criterion. Among these optimal trajectories, the red one in Figure 4.8 is the shortest distance trajectory from the start configuration to the intersection between the bounding boxes.

Sub-case 3 is when the start configuration is in a region whose failure scenarios are scenario A and B, i.e., region 1, 3, 7 or 9. To minimize the failure probability, the robot should move to the intersection in θ_1 or θ_2 as soon as possible, so there are two ways to reach the intersection between the bounding boxes, as shown in Figure 4.9. Let $\Delta\theta'_1$ and $\Delta\theta'_2$ be the θ_1 and θ_2 distance from the start configuration to the intersection between the bounding boxes, respectively. The first way is the robot can first reach the intersection in θ_1 by rotating $\Delta\theta'_1$ in joint 1 and $\Delta\theta_2$ in joint 2 simultaneously, and then the robot can reach the intersection between the bounding boxes

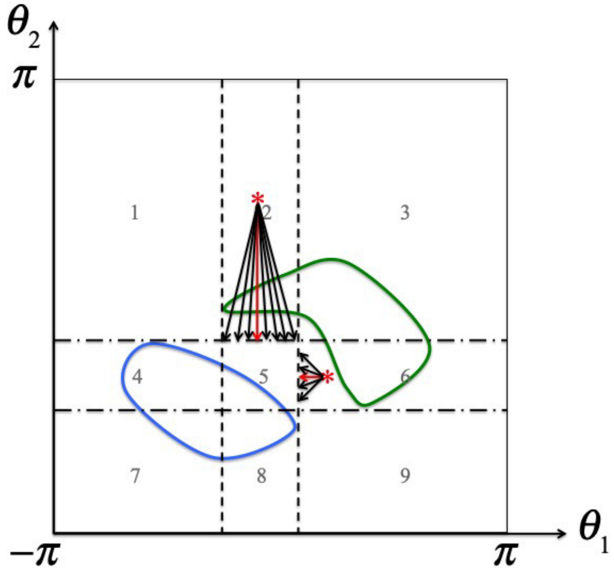


Figure 4.8: The optimal trajectories when the start configuration is in regions whose failure scenarios are scenario B, C or A, D are shown. The red trajectory is the shortest distance trajectory.

by rotating $\Delta\theta_2' - \Delta\theta_2$ ($\Delta\theta_2' \geq \Delta\theta_2$) in joint 2. The other way is the robot can first reach the intersection in θ_2 by rotating $\Delta\theta_1$ in joint 1 and $\Delta\theta_2'$ in joint 2 simultaneously, and then the robot can reach the intersection between the bounding boxes by rotating $\Delta\theta_1' - \Delta\theta_1$ ($\Delta\theta_1' \geq \Delta\theta_1$) in joint 1. The failure probability of the trajectory in the first way can be calculated by

$$\begin{aligned} P_1 =& e^{-\lambda_1 t_0} - e^{-\lambda_1(t_0+\frac{max(\Delta\theta_1',\Delta\theta_2)}{v})} + e^{-\lambda_2 t_0} - e^{-\lambda_2(t_0+\frac{max(\Delta\theta_1',\Delta\theta_2)}{v})} + \\ & e^{-\lambda_2(t_0+\frac{max(\Delta\theta_1',\Delta\theta_2)}{v})} - e^{-\lambda_2(t_0+\frac{max(\Delta\theta_1',\Delta\theta_2)}{v}+\frac{\Delta\theta_2'-\Delta\theta_2}{v})} \\ =& e^{-\lambda_1 t_0} - e^{-\lambda_1(t_0+\frac{max(\Delta\theta_1',\Delta\theta_2)}{v})} + e^{-\lambda_2 t_0} - e^{-\lambda_2(t_0+\frac{max(\Delta\theta_1',\Delta\theta_2)}{v}+\frac{\Delta\theta_2'-\Delta\theta_2}{v})}. \end{aligned} \tag{4.9}$$

Similarly, the failure probability of the trajectory in the second way can be calculated by

$$\begin{aligned} P_2 =& e^{-\lambda_1 t_0} - e^{-\lambda_1(t_0+\frac{max(\Delta\theta_2',\Delta\theta_1)}{v})} + e^{-\lambda_2 t_0} - e^{-\lambda_2(t_0+\frac{max(\Delta\theta_2',\Delta\theta_1)}{v})} + \\ & e^{-\lambda_1(t_0+\frac{max(\Delta\theta_2',\Delta\theta_1)}{v})} - e^{-\lambda_1(t_0+\frac{max(\Delta\theta_2',\Delta\theta_1)}{v}+\frac{\Delta\theta_1'-\Delta\theta_1}{v})} \\ =& e^{-\lambda_1 t_0} - e^{-\lambda_1(t_0+\frac{max(\Delta\theta_2',\Delta\theta_1)}{v}+\frac{\Delta\theta_1'-\Delta\theta_1}{v})} + e^{-\lambda_2 t_0} - e^{-\lambda_2(t_0+\frac{max(\Delta\theta_2',\Delta\theta_1)}{v})}. \end{aligned} \tag{4.10}$$

If $\Delta\theta_1' = \Delta\theta_2'$, i.e., the θ_1 distance from the start configuration to the intersection between the bounding boxes is equal to the θ_2 distance from the start configuration to the intersection between the bounding boxes,

$$P_1 = e^{-\lambda_1 t_0} - e^{-\lambda_1 (t_0 + \frac{\Delta\theta_1'}{v})} + e^{-\lambda_2 t_0} - e^{-\lambda_2 (t_0 + \frac{\Delta\theta_1' + \Delta\theta_2' - \Delta\theta_2}{v})}, \tag{4.11}$$

$$P_2 = e^{-\lambda_1 t_0} - e^{-\lambda_1 (t_0 + \frac{\Delta\theta_2' + \Delta\theta_1' - \Delta\theta_1}{v})} + e^{-\lambda_2 t_0} - e^{-\lambda_2 (t_0 + \frac{\Delta\theta_2'}{v})}. \tag{4.12}$$

When $\Delta\theta_2 = \Delta\theta_2'$, P_1 reaches its minimum

$$P_1^* = e^{-\lambda_1 t_0} - e^{-\lambda_1 (t_0 + \frac{\Delta\theta_1'}{v})} + e^{-\lambda_2 t_0} - e^{-\lambda_2 (t_0 + \frac{\Delta\theta_1'}{v})}, \tag{4.13}$$

and when $\Delta\theta_1 = \Delta\theta_1'$, P_2 reaches its minimum

$$P_2^* = e^{-\lambda_1 t_0} - e^{-\lambda_1 (t_0 + \frac{\Delta\theta_2'}{v})} + e^{-\lambda_2 t_0} - e^{-\lambda_2 (t_0 + \frac{\Delta\theta_2'}{v})}. \tag{4.14}$$

Because $\Delta\theta_1' = \Delta\theta_2'$, $P_1^* = P_2^*$, the optimal trajectory is the straight line from the start configuration to the corner of the intersection between the bounding boxes, which is shown in red in Figure 4.9a. If $\Delta\theta_1' > \Delta\theta_2'$, i.e., the θ_1 distance from the start configuration to the intersection between the bounding boxes is larger than the θ_2 distance from the start configuration to the intersection between the bounding boxes, P_1 and its minimum value are still equal to (4.11) and (4.13), respectively. The failure probability of the second way P_2 depends on the values of $\Delta\theta_2'$ and $\Delta\theta_1$. If $\Delta\theta_1' > \Delta\theta_2' \geq \Delta\theta_1$, P_2 is still equal to (4.12), and when $\Delta\theta_1 = \Delta\theta_2'$, P_2 reaches its minimum

$$P_2^* = e^{-\lambda_1 t_0} - e^{-\lambda_1 (t_0 + \frac{\Delta\theta_1'}{v})} + e^{-\lambda_2 t_0} - e^{-\lambda_2 (t_0 + \frac{\Delta\theta_2'}{v})}. \tag{4.15}$$

If $\Delta\theta_1' > \Delta\theta_1 \geq \Delta\theta_2'$, P_2 becomes

$$P_2 = e^{-\lambda_1 t_0} - e^{-\lambda_1 (t_0 + \frac{\Delta\theta_1'}{v})} + e^{-\lambda_2 t_0} - e^{-\lambda_2 (t_0 + \frac{\Delta\theta_1}{v})}. \tag{4.16}$$

When $\Delta\theta_1 = \Delta\theta_2'$, it reaches the minimum, which is the same as (4.15). Because $\Delta\theta_1' > \Delta\theta_2'$, P_2^* in (4.15) is smaller than P_1^* in (4.13), so the optimal trajectory is the robot first rotates $\Delta\theta_2'$ in both joint 1 and joint 2 to reach the intersection in θ_2, and then it rotates $\Delta\theta_1' - \Delta\theta_2'$ in joint 1 to reach the intersection between the bounding boxes, which is shown in red in Figure 4.9b. This result also makes senses from a physical point of view. The robot tries to reach a closer intersection (intersection in θ_2) first, and then reach the intersection between the bounding boxes. In the first piece, all the trajectories that satisfies $\Delta\theta_1 \leq \Delta\theta_2'$ spend the same amount of time, i.e., the failure probabilities of the trajectories are all the same, so the optimal one is rotating $\Delta\theta_2'$ in both joint 1 and joint 2 to minimize the distance of the trajectory in the second piece. When $\Delta\theta_1' < \Delta\theta_2'$, the optimal trajectory can be obtained in an analogous manner by flipping θ_1 and θ_2, which is shown in red in Figure 4.9c.

4.3.3 Planning the Optimal Trajectory in Case 2

The method to find the optimal trajectory in Case 2 will be discussed in this subsection. In Case 2, all the configurations along the self-motion manifold of task point 1 need to be checked. For each configuration of task point 1, the optimal trajectory from the start configuration to this specific configuration and the optimal trajectory from this specific configuration to the bounding box of the self-motion manifold of task 2 need to be identified, respectively. Applying the method presented in the above subsection, the second piece, i.e., the optimal trajectory from one specific configuration of task point 1 to the bounding box of the self-motion manifold of task 2 can be identified easily. When the method is employed, the start configuration is substituted by the configuration of task point 1, and the intersection between the bounding boxes is substituted by the bounding box of the self-motion manifold of task point 2. The intersection in θ_1 is substituted by the θ_1 range of the self-motion manifold of task point 2, and the intersection in θ_2 is substituted by the θ_2 range of the self-motion manifold of task point 2. The key problem is how to find the optimal trajectory from the start configuration to one configuration of task point 1.

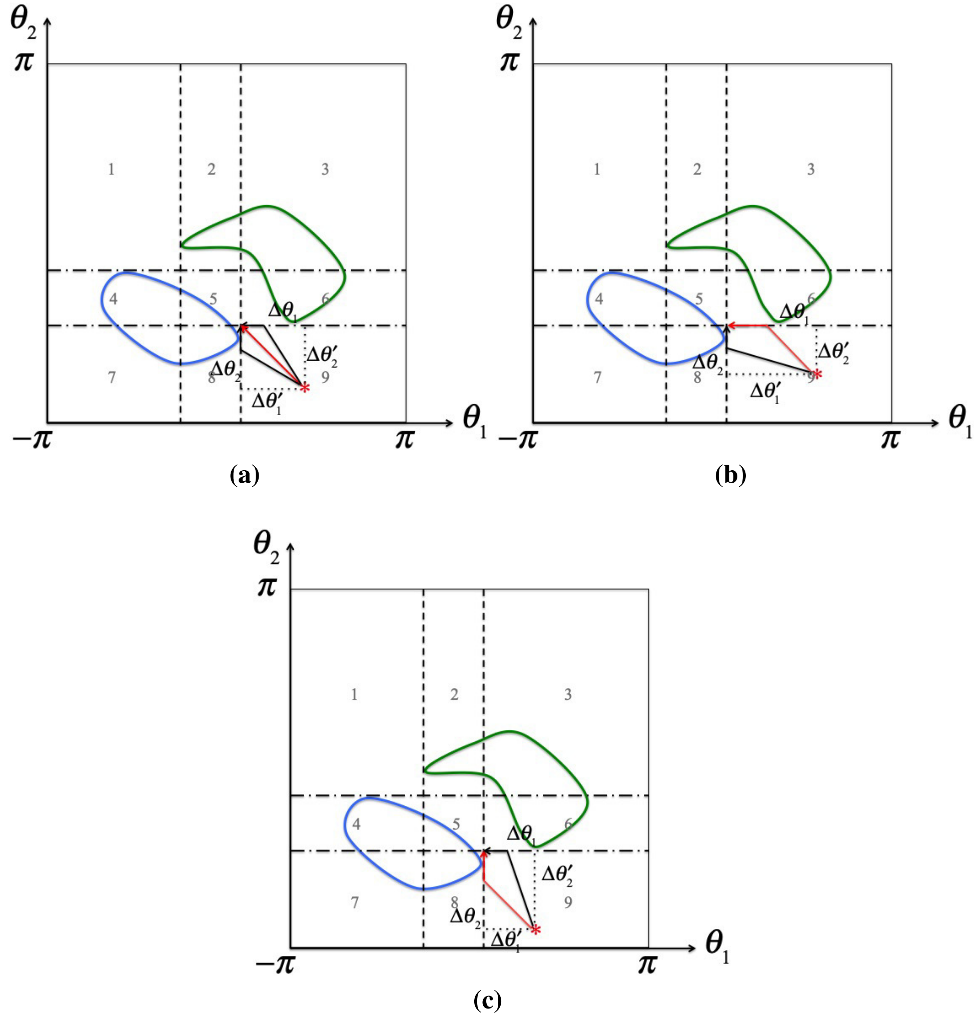


Figure 4.9: The optimal trajectory from the start configuration to the intersection between the bounding boxes is shown. In (a), the distance in θ_1 from the start configuration to the intersection between the bounding boxes $\Delta\theta'_1$ is equal to the distance in θ_2 from the start configuration to the intersection between the bounding boxes $\Delta\theta'_2$. The optimal trajectory is the straight line from the start configuration to the corner of the intersection between the bounding boxes. In (b), when $\Delta\theta'_1 > \Delta\theta'_2$, the optimal trajectory is the robot first rotates $\Delta\theta'_2$ in both joint 1 and joint 2 to reach the intersection in θ_2, and then it rotates $\Delta\theta'_1 - \Delta\theta'_2$ in joint 1 to reach the intersection between the bounding boxes. In (c), when $\Delta\theta'_1 < \Delta\theta'_2$, the optimal trajectory is the robot first rotates $\Delta\theta'_1$ in both joint 1 and joint 2 to reach the intersection in θ_1, and then it rotates $\Delta\theta'_2 - \Delta\theta'_1$ in joint 2 to reach the intersection between the bounding boxes.

To illustrate the planning algorithm more clearly, a two-dimensional joint space is used as an example, and the optimal trajectory in a higher dimensional joint space can be identified in an analogous manner. Similarly to the method presented in the above section, there are two sub-cases according to the relation between the region of the start configuration and the region of the task point 1 configuration. Sub-case 1 is the start configuration and the configuration of task point 1 are in the same region. One potential trajectory is the straight line from the start configuration to the configuration of task point 1. This is the optimal trajectory when the region of the start configuration and the configuration of task point 1 has the fewest number of failure scenarios compared with other regions, as shown in Figure 4.10. In Figure 4.10a, the bounding boxes of the self-motion manifolds of task point 1 and 2 do not intersect in neither θ_1 or θ_2, so the joint space is only divided into one region, whose failure scenarios are A and B. Obviously, the optimal trajectory is the straight line from the start configuration to the configuration of task point 1. In Figure 4.10b, the bounding boxes of the self-motion manifolds of task point 1 and 2 intersect each other in θ_2, so the joint space is divided into three regions. The start configuration and the configuration of task point 1 is in the region with the fewest number of failure scenarios, which is failure scenario A. Therefore, the optimal trajectory is the straight line from the start configuration to the configuration of task point 1.

However, when the adjacent regions have a fewer number of failure scenarios, to minimize the failure probability, the robot may first go into these regions as soon as possible, and then reach task point 1, as shown in Figure 4.11. The robot may go into only one adjacent region to reach task point 1, as shown in Figure 4.11a, and in some situations the robot may need to go through two adjacent regions to reach task point 1, as shown in Figure 4.11b. As described in the above subsection, to minimize the distance of the trajectories in the adjacent regions, the robot will rotate the same amount of degrees in both joint 1 and joint 2 when the robot goes into and outside of the adjacent regions, which spends the same amount of time to go into and outside of the region directly. In this sub-case, all these potential trajectories including the straight-line trajectory from the start configuration to the configuration of task point 1 are identified first, and then the failure

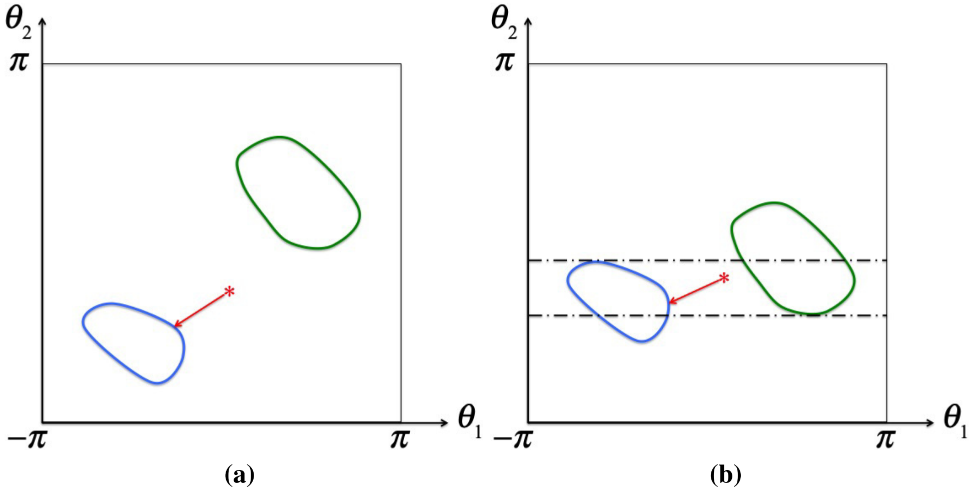


Figure 4.10: The optimal trajectory from the start configuration to the configuration of task point 1 is the straight line when the start configuration and the configuration of task point are in the same region with the fewest number of failure scenarios. In (a), the bounding boxes of the self-motion manifolds of task point 1 and 2 do not intersect in neither θ_1 or θ_2, so the joint space is only divided into one region, whose failure scenarios are A and B. In (b), the bounding boxes of the self-motion manifolds of task point 1 and 2 intersect each other in θ_2, so the joint space is divided into three regions. The start configuration and the configuration of task point 1 is in the region with the fewest number of failure scenarios, which is failure scenario A.

probabilities of all these potential trajectories are calculated. The trajectory with the smallest failure probability is the optimal trajectory from the start configuration to the configuration of task point 1.

To figure out all the potential trajectories, one need to figure out the potential adjacent regions that the robot may go into from the start configuration and go outside of to reach the configuration of task point 1, respectively. These adjacent regions need to satisfy the following four conditions. (1) the adjacent region must be inside the joint space, i.e. the boundaries of the adjacent region is between $-\pi$ and π. (2) The number of the failure scenarios of this adjacent region is smaller than the region of the start configuration and the configuration of task point 1. (3) The distance from the start configuration (or the configuration of task point 1) to this region is smaller than $max(\Delta\theta_1', \Delta\theta_2')$, where $\Delta\theta_1'$ and $\Delta\theta_2'$ are the θ_1 distance and θ_2 distance between the start configuration and the configuration of task point 1, respectively. (4) When the robot rotate the same amount of degrees in both joint 1 and joint 2 to go into the adjacent region from the start configuration (or go outside of the adjacent region to reach the configuration of task point 1), the

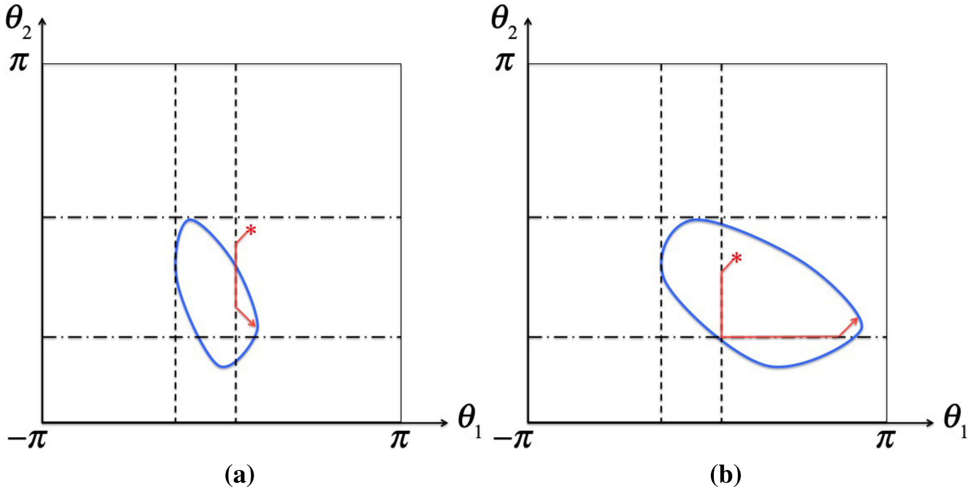


Figure 4.11: When the start configuration and the configuration of task point are in the same region whose failure scenarios are more than its adjacent regions, the robot may need to go into adjacent regions to reach task point 1. In (a), the robot may only go into one adjacent region to reach task point 1. In (b), the robot may need to go through two adjacent regions to reach task point 1.

configuration arriving (or leaving) the adjacent region should be still inside the boundaries of the adjacent region. After the sets of the potential adjacent regions that the robot may go into from the start configuration and go outside of to reach the configuration of task point 1 are identified, respectively, all the potential trajectories from the start configuration to the configuration of task point 1 can be obtained by taking the combinations between these two sets. The valid combination should satisfy that the sum of the distance from the start configuration to its associated adjacent region and the distance from the configuration of task point 1 to its associated adjacent region is smaller than $max(\Delta\theta'_1, \Delta\theta'_2)$; otherwise the failure probability of this potential trajectory will be larger than the straight-line trajectory from the start configuration to the configuration of task point 1. Finally, the optimal trajectory is the trajectory with the smallest failure probability.

Sub-case 2 is the start configuration and the configuration of task point 1 are in different regions. For this sub-case, one potential trajectory is the robot first goes into the region with the fewest number of failure scenarios by following the trajectory with minimum failure probability, and then it goes through the optimal region. Last, the robot goes outside of that region to reach the configuration of task point 1 also by following the trajectory with minimum failure probability.

The first piece, i.e., the trajectory from the start configuration to the region with the fewest number of failure scenarios, can be determined by applying the method described in the above subsection. This is also true for the trajectory from the optimal region to the configuration of task point 1, where the start configuration is substituted by the configuration of task point 1. Last, the configuration that the robot arrives the optimal region from the start configuration and the configuration that the robot leaves the optimal region to reach the configuration of task point 1 are connected by straight line, as shown in Figure 4.12.

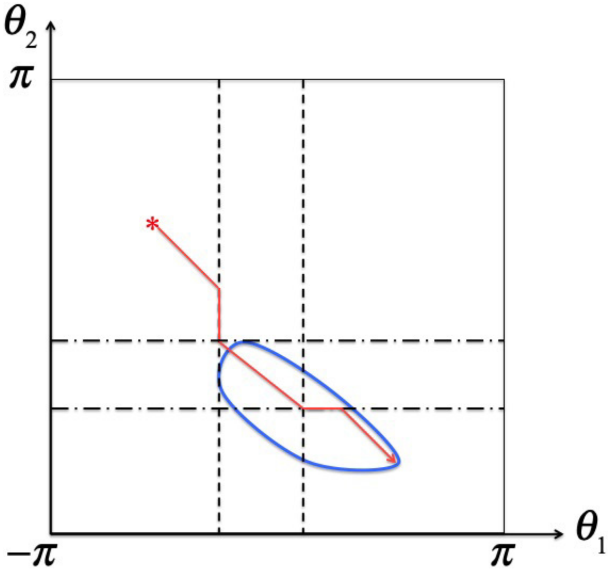


Figure 4.12: The potential trajectory from the start configuration to the configuration of task point 1 when they are in different regions. The robot first goes into the region with the fewest number of failure scenarios by following the trajectory with minimum failure probability, and then it goes through the optimal region in straight line. Last, the robot goes outside of that region to reach the configuration of task point 1 also by following the trajectory with minimum failure probability.

However, when the start configuration and the configuration of task point 1 are in the regions with the same boundaries in θ_1 or θ_2, the robot may not need to go through the regions with the fewest number of failure scenarios. In this situation, the region with the fewest number of failure scenarios among the region of the start configuration, the region of the configuration of task point 1 and the regions between them need to be identified first. The robot goes into this region from the start configuration by following the trajectory with minimum failure probability, then it goes through this region. Last, the robot goes outside of that region to reach the configuration of task point 1 also by following the trajectory with minimum failure probability. Comparing the failure probabilities of the two potential trajectories going through and not going through the optimal region, one can find the optimal trajectory from the start configuration to the configuration of task point 1 when they are in different regions with the same boundaries in θ_1 or θ_2.

An example of the potential trajectories when the start configuration and the configuration of task point 1 are in the regions with the same boundaries in θ_1 is shown in Figure 4.13. The start configuration is in region 3 whose failure scenarios are A and B, and the configuration of task point 1 is in region 6 whose failure scenario is only A. As these two regions are adjacent, the optimal region between them is region 6. If the distance in θ_1 between the start configuration and the configuration of task point 1 denoted as $\Delta\theta_1'$ is smaller than the distance in θ_2 from the star configuration to region 6 denoted as $\Delta\theta_2''$, the robot rotates $\Delta\theta_1'$ in joint 1 and $\Delta\theta_2''$ in joint 2 to go into region 6 from the start configuration, and then rotates $\Delta\theta_2' - \Delta\theta_2''$ in joint 2 to reach the configuration of task point 1, where $\Delta\theta_2'$ is the distance in θ_2 between the start configuration and the configuration of task point 1, as shown in Figure 4.13a. Otherwise, the robot rotates $\Delta\theta_2''$ in both joint 1 and joint 2 to go into region 6, and then rotates $\Delta\theta_1' - \Delta\theta_2''$ in joint 1 and $\Delta\theta_2' - \Delta\theta_2''$ in joint 2 to reach the configuration of task point 1, as shown in Figure 4.13b. This makes sense because the robot first moves as much as needed but less than $\Delta\theta_2''$ in θ_1 to reach region 6, and then reaches the configuration of task point 1 in straight-line trajectory in region 6, because the number of the failure scenarios in region 6 is fewer than that in region 3.

Another example is shown in Figure 4.14. The start configuration is in region 3 whose failure scenarios are A and B, and the configuration of task point 1 is in region 9 whose failure scenarios are also A and B. The region between them is region 6 whose failure scenario is only A. Therefore, the optimal region among them is region 6. The potential trajectory is divided into three pieces. The first piece is the robot moves as much as needed but less than $\Delta\theta_2''$ in θ_1 to reach region 6. The last piece is the robot moves as much as needed but less than $\Delta\theta_2'''$ in θ_1 to leave region 6 and reach the configuration of task point 1, where $\Delta\theta_2'''$ is the distance in θ_2 from the configuration of task point 1 to region 6. The second piece is the straight line connecting the two configurations arriving and leaving region 6.

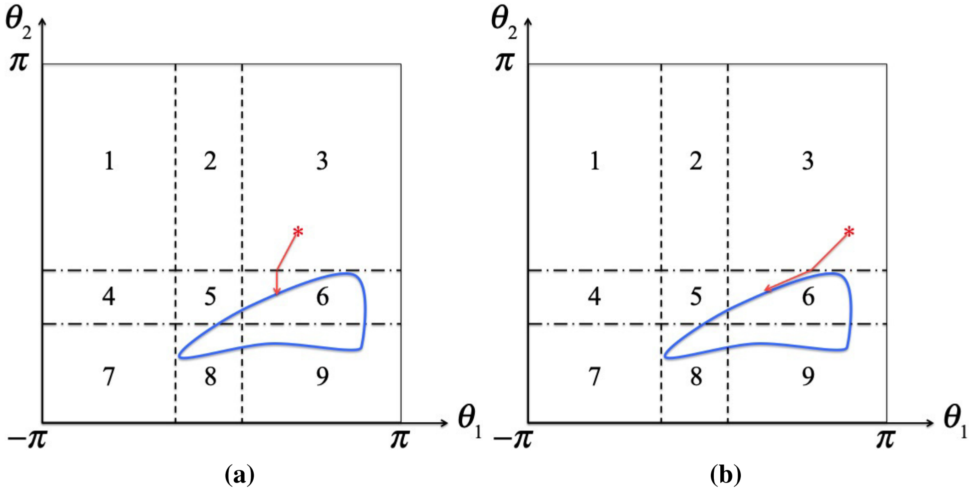
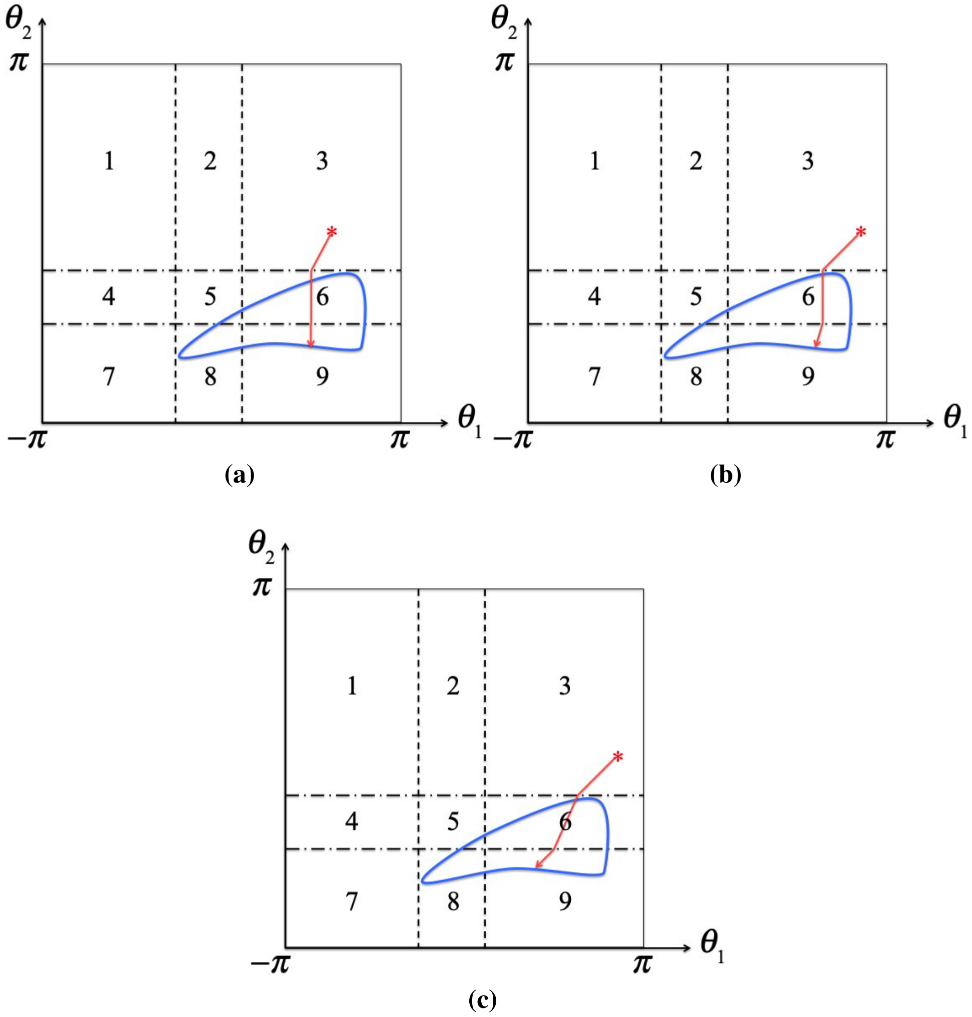


Figure 4.13: An example of the potential trajectory from the start configuration to the configuration of task point 1 when they are in different regions with the same boundaries in θ_1 is shown. In (a), the robot rotates as much as needed in joint 1 and a determined value in joint 2 to go into region 6 from the start configuration, and then only rotates joint 2 to reach the configuration of task point 1. In (b), the robot rotates the maximum value allowed in joint 1 to minimize the failure probability and a determined value in joint 2 to go into region 6, and then rotates both joint 1 and joint 2 to reach the configuration of task point 1.

4.4 Illustrative Example for Planar 2R Redundant Robots

4.4.1 Optimal Trajectory in Case 1

The above algorithm for planning the trajectory with minimum probability of task failure in Case 1, where there exists an intersection between the self-motion manifold bounding boxes of task point 1 and 2, and there exists self-motion manifolds of task point 1 in the intersection, is illustrated on a simple planar 2R robot. Only the x coordinate of the end-effector is controlled, so the robot has one degree of redundancy. The results are shown in Fig. 4.15. The blue circles are the discrete configurations along the self-motion manifold of task point 1, and the green circles are the discrete configurations along the self-motion manifold of task point 2. The intersection between



(a)

(b)

(c)

Figure 4.14: Another example of the potential trajectory from the start configuration to the configuration of task point 1 when they are in different regions with the same boundaries in θ_1 is shown. In (a), the robot rotates as much as needed in joint 1 and a determined value in joint 2 to go into region 6 from the start configuration, and then only rotates joint 2 to go through region 6 and reach the configuration of task point 1. In (b), the robot rotates the maximum value allowed in joint 1 to minimize the failure probability and a determined value in joint 2 to go into region 6, and rotates only joint 2 to go through region 6. Last, the robot rotates as much as needed in joint 1 and a determined value in joint 2 to leave region 6 and reach the configuration of task point 1. In (c), the robot rotates the maximum value allowed in joint 1 to minimize the failure probability and a determined value in joint 2 to go into region 6 and go outside of region 6 to reach the configuration of task point 1, and it rotates both joint 1 and joint 2 to go through region 6.

the bounding boxes of these two self-motion manifolds is the bounding box of the self-motion manifold of task point 1.

In Figure 4.15a, the start configuration is in a region whose failure scenario is only A, so it belongs to sub-case 2. Recall that there are infinite number of optimal trajectories as long as the trajectory satisfy that the θ_2 distance of the trajectory is less than the θ_1 distance from the start configuration to the intersection. The red trajectory is the shortest distance trajectory from the start configuration to the bounding box. In Figure 4.15b, the start configuration is in a region whose failure scenarios are A and B, so it belongs to sub-case 3. The optimal trajectory is the robot rotates the same amount of value in both joint 1 and joint 2 to reach the intersection in θ_1 as soon as possible, and then the robot rotates only joint 2 to reach the intersection of the bounding boxes.

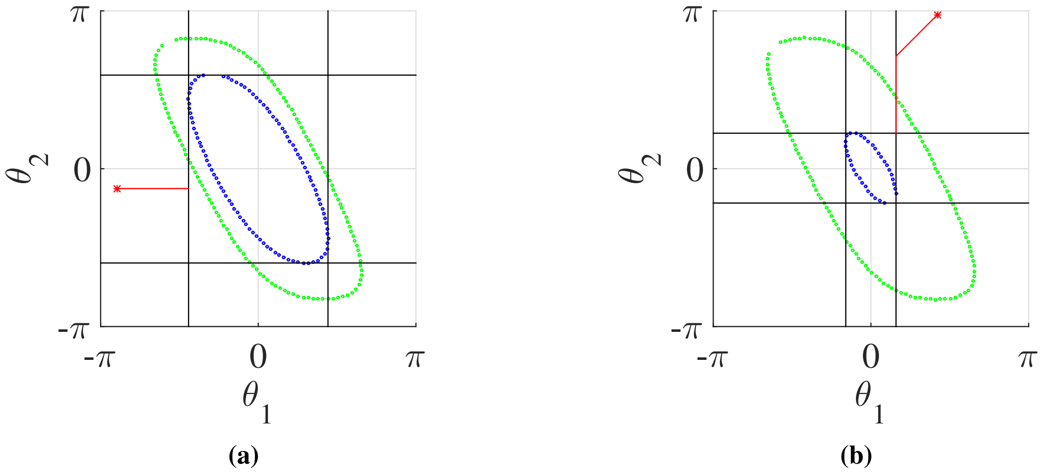


Figure 4.15: Two illustrative examples of planning the optimal trajectories with minimum probability of task failure in Case 1 for a planar 2R robot are shown. (a) is an example of sub-case 2 where the failure scenario of the region of the start configuration is only A, and (b) is an example of sub-case 3 where the failure scenarios of the region of the start configuration are A and B.

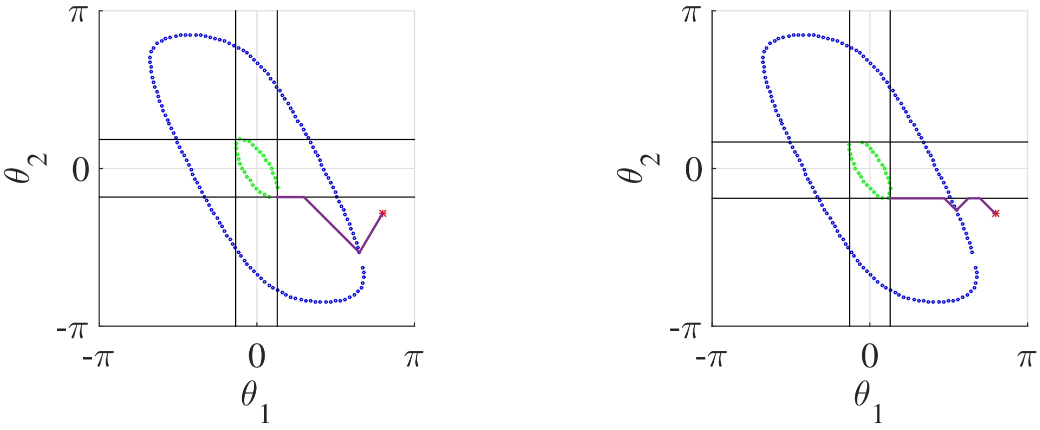
4.4.2 Optimal Trajectory in Case 2

An illustrative example of planning the trajectory with minimum probability of task failure in Case 2 is shown in Figure 4.16, Figure 4.17 and Figure 4.18. In this example, there exists an intersection between the bounding boxes of the self-motion manifolds of task point 1 and 2, which is the bounding box of the self-motion manifold of task point 2. However, there does not exist the self-motion manifold of task point 1 in the intersection, so it belongs to Case 2, and all the configurations along the self-motion manifold of task point 1 need to be checked. In Figure 4.16, the start configuration and the configuration of task point 1 are in the same region, which belongs to sub-case 1. The optimal trajectory from the start configuration to the configuration of task point 1 can be a straight-line trajectory as shown in Figure 4.16a, or the optimal trajectory goes through an adjacent region with fewer number of failure scenarios and then reaches the configuration of task point 1, as shown in Figure 4.16b.

Sub-case 2 is the start configuration and the configuration of task point 1 are in different regions, as shown in Figure 4.17. From Figure 4.17a to Figure 4.17d, the start configuration and the configuration of task point 1 are in different regions with the same boundaries in either θ_1 or θ_2. The optimal trajectory from the start configuration to the configuration of task point 1 may go though the intersection of the bounding boxes, i.e., the region with the fewest number of failure scenarios, as shown in Figure 4.17a and Figure 4.17c, or the optimal trajectory goes from the region of the start configuration to the region of the task point 1 configuration directly, as shown in Figure 4.17b and Figure 4.17d. When the start configuration and the configuration of task point 1 are in different regions with different boundaries in θ_1 and θ_2, the optimal trajectory from the start configuration to the configuration of task point 1 must go through the intersection, as shown in Figure 4.17e. Finally, comparing the failure probabilities of all the trajectories going through different configurations of task point 1, one can obtain the optimal trajectory with minimum probability of task failure, as shown in Figure 4.18.

4.5 Chapter Summary

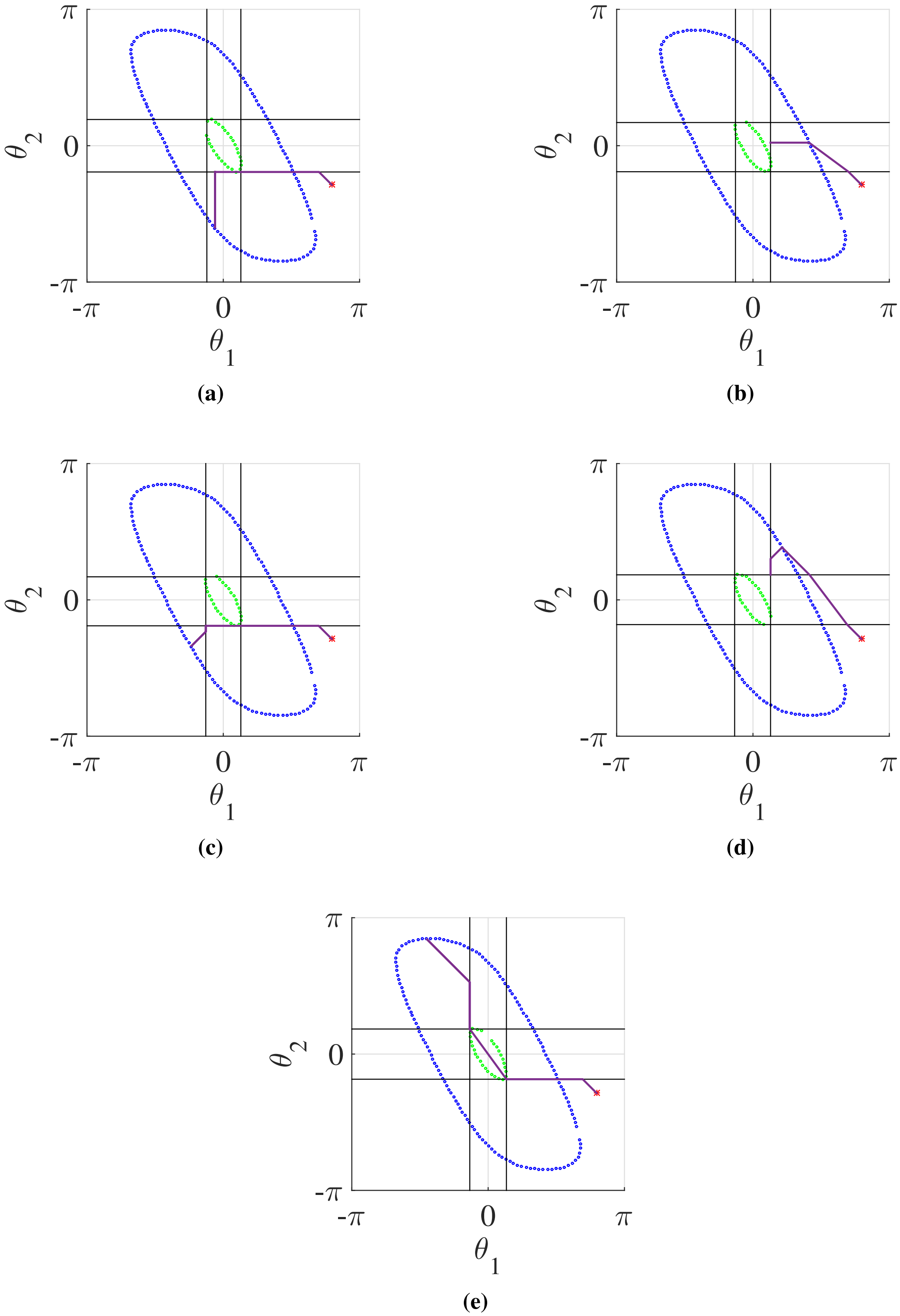
This chapter proposed an algorithm to plan a trajectory with minimum probability of task failure with locked joints for point-to-point tasks. The joint space is first divided into different regions based on the intersection of the self-motion manifold bounding boxes of all the task points, and the failure scenarios of each region are identified. These regions can be further classified



(a) **(b)**

Figure 4.16: When the start configuration and the configurations of task point 1 are in the same region, different optimal trajectories are shown. In (a), the optimal trajectory from the start configuration to the configuration of task point 1 is a straight line, and in (b) the optimal trajectory goes through an adjacent region with fewer number of failure scenarios and then reaches the configuration of task point 1.

into several categories based on their failure scenarios. After each task point is reached, the joint space is re-divided based on the intersection of the self-motion manifold bounding boxes of all the remaining task points. The probability of task failure of an arbitrary trajectory can be calculated by dividing the trajectory into small segments, and the equations to calculate the probability of task failure of each segment are derived based on different failure scenarios. After the method to calculate the probability of task failure of an arbitrary trajectory is developed, an algorithm is proposed to find the optimal trajectory with minimum probability of task failure. There are two cases in finding the optimal trajectory. Case 1 is when there exists an intersection between the self-motion manifold bounding boxes of all the task points, and there exists the self-motion manifold of a task point in the intersection of this task point and its following ones. In this case, the optimal trajectory is the trajectory with minimum probability of task failure from the start configuration to the intersection of the bounding boxes, and there are three sub-cases according to the region of the start configuration. All the other situations belong to Case 2, and in this case all the configurations along the self-motion manifold of task point 1 need to be checked to find



(a)

(b)

(c)

(d)

(e)

Figure 4.17: When the start configuration and the configurations of task point 1 are in different regions, different optimal trajectories are shown. In (a) and (c), the optimal trajectory from the start configuration to the configuration of task point 1 goes through the intersection of the bounding boxes, and in (b) and (d) the optimal trajectory goes from the region of the start configuration to the region of the task point 1 configuration directly.

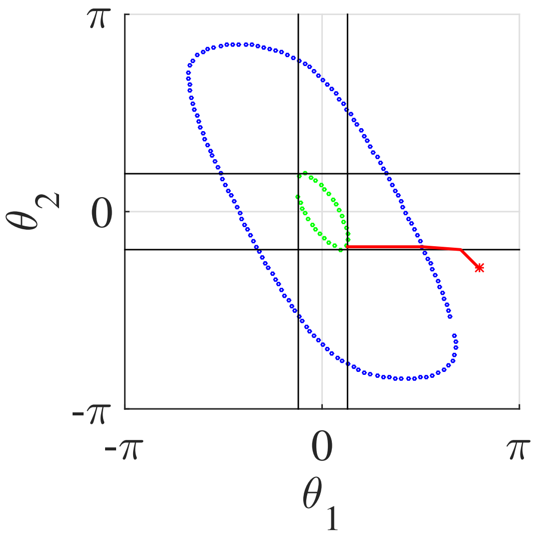


Figure 4.18: The optimal trajectory with minimum probability of task failure is shown.

the trajectory with minimum probability of task failure. Finally, examples of planning optimal trajectories for planar 2R redundant robots are given.

Chapter 5

Conclusions

5.1 Summary

This work focuses on the kinematic design and motion planning of fault tolerant robots with locked joint failures. The problem of kinematic design of optimally fault toelrant robots is first studied in Chapter 2 and 3, and then the problem of planning the optimal trajectory with minimum probability of task failure is studied in Chapter 4. Chapter 2 explored the structure and global pre- and post-failure dexterity performance of the $7!$ robot designs generated from an optimally fault tolerant Jacobian. It was shown that when describing the kinematic design of a robot in terms of DH parameters for all joints, there are only 21 possible values of α and a for all $7!$ robot designs, and there are 210 possible values of d and θ. In addition, these designs were organized into a tree structure based on the possible choices for $\{\alpha, a\}$ pairs. It was also shown that each of the $7!$ robot designs had a reverse version with very similar DH parameters but potentially very different global properties. Furthermore, the global performance of the $7!$ robot designs was analyzed. It was shown that there is a relatively strong correlation between performance measures computed in the joint space and workspace, so that the computationally efficient joint space calculations could be used to identify the best candidates for optimal designs in the workspace. These candidates were further analyzed in the workspace to determine two Pareto optimal designs in terms of pre- and post-failure dexterity, and the distribution of these measures throughout the workspace were shown.

Chapter 3 presented methods to design optimally fault tolerant robots for different joint failure probabilities. It was shown that for an arbitrary set of joint failure probabilities, the optimal null space satisfies $|\hat{n}_i| = w_i/\sqrt{w_1^2 + w_2^2 + \cdots + w_n^2}$, where the ratio of each element is equal to the ratio of the failure probabilities of the different joints. Based on the optimal null space, isotropic Jacobians for an arbitrary number of joints are constructed, and the kinematic parameters of the robots are generated from these optimal Jacobians. In addition, a method for designing an optimal spatial 7R manipulator Jacobian is derived, and it is shown that such optimal 7R designs do not exist for all possible joint failure probabilities. An example of such a case is that of equal joint failure probabilities. However, an optimal Jacobian for nearly equal joint failure probabilities is given. Furthermore, a method for designing optimally fault tolerant robots for multiple cases of joint failure probabilities is introduced and the design of optimal planar 3R robots is given as an illustrative example. It was shown that there exists a family of optimal robots whose last link length is equal to 40% of the total link length. This family of robots are also shown to be optimal for the case of equally likely joint failures, as well as failures in joint two. However, the optimization for failures in joints one and three are directly competing, so that one must select a particular design based on the failure probabilities of these joints.

Chapter 4 proposed an algorithm to plan a trajectory with minimum probability of task failure with locked joints for point-to-point tasks. The joint space is first divided into different regions based on the intersection of the self-motion manifold bounding boxes of all the task points, and the failure scenarios of each region are identified. These regions can be further classified into several categories based on their failure scenarios. After each task point is reached, the joint space is re-divided based on the intersection of the self-motion manifold bounding boxes of all the remaining task points. The probability of task failure of an arbitrary trajectory can be calculated by dividing the trajectory into small segments, and the equations to calculate the probability of task failure of each segment are derived based on different failure scenarios. After the method to calculate the probability of task failure of an arbitrary trajectory is developed, an algorithm is proposed to find the optimal trajectory with minimum probability of task failure. There are two cases in finding the optimal trajectory. Case 1 is when there exists an intersection between the self-motion manifold bounding boxes of all the task points, and there exists the self-motion manifold of a task point in the intersection of this task point and its following ones. In this case, the optimal trajectory is the trajectory with minimum probability of task failure from the start configuration to the intersection of the bounding boxes, and there are three sub-cases according to the region of the start configuration. All the other situations belong to Case 2, and in this case all the configurations along the self-motion manifold of task point 1 need to be checked to find the trajectory with minimum probability of task failure. Finally, examples of planning optimal trajectories for planar 2R redundant robots are given.

5.2 Future Work

There are a wide range of potential issues to be explored as future work. In kinematic design of optimally fault tolerant robots, determining for which sets of joint failure probabilities that optimal 7R designs exist is an open problem. In addition, the characterization of the families of optimal robots with respect to practical considerations such as joint limits and self collision, and how they affect the resulting workspace, is an important design factor. Furthermore, this work can be extended to consider a measure of fault tolerance with respect to a certain set of tasks that one would like to guarantee that the robot can perform after a failure.

In planning the optimal trajectory with minimum probability of task failure, the algorithm will be extended to a higher dimensional joint space, such as planar 3R robots for positioning, spatial 4R robots for positioning, and spatial 7R robots for positioning and orientating. The examples of planning the optimal trajectories for these robots will be given in the future work.

Bibliography

[1] Biyun Xie and Anthony A Maciejewski. Structure and performance analysis of the 7! robots generated from an optimally fault tolerant Jacobian. *IEEE Robotics and Automation Letters*, 2(4):1956–1963, 2017.

[2] Biyun Xie and Anthony A Maciejewski. Kinematic design of optimally fault tolerant robots for different joint failure probabilities. *IEEE Robotics and Automation Letters*, 3(2):827–834, 2018.

[3] Zonggao Mu, Bing Zhang, Wenfu Xu, Bing Li, and Bin Liang. Fault tolerance kinematics and trajectory planning of a 6-DOF space manipulator under a single joint failure. In *IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pages 483–488, 2016.

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

[5] Didier Crestani, Karen Godary-Dejean, and Lionel Lapierre. Enhancing fault tolerance of autonomous mobile robots. *Robotics and Autonomous Systems*, 68:140–155, 2015.

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

[7] Yujia Wang, Philip A Wilson, and Xing Liu. Adaptive neural network-based backstepping fault tolerant control for underwater vehicles with thruster fault. *Ocean Engineering*, 110:15–24, 2015.

[8] Iuliu Vasilescu, Paulina Varshavskaya, Keith Kotay, and Daniela Rus. Autonomous modular optical underwater robot (amour) design, prototype and feasibility study. In *Proceedings*

of the 2005 IEEE International Conference on Robotics and Automation, pages 1603–1609. IEEE, 2005.

[9] James Trevelyan, William R Hamel, and Sung-Chul Kang. Robotics in hazardous applications. In *Springer Handbook of Robotics*, pages 1521–1548. Springer, 2016.

[10] Ian D Walker and Joseph R Cavallaro. Failure mode analysis for a hazardous waste clean-up manipulator. *Reliability Engineering & System Safety*, 53(3):277–290, 1996.

[11] Richard Colbaugh and Mohammad Jamshidi. Robot manipulator control for hazardous waste-handling applications. *Journal of Robotic Systems*, 9(2):215–250, 1992.

[12] 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.

[13] Rachot Phuengsuk and Jackrit Suthakorn. A study on risk assessment for improving reliability of rescue robots. In *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 667–672, 2016.

[14] Takeyuki Kawata, Kazuto Kamiyama, Masaru Kojima, Mitsuhiro Horade, Yasushi Mae, and Tatsuo Arai. Fault-tolerant adaptive gait generation for multi-limbed robot. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3381–3386. IEEE, 2016.

[15] Min Yang Jung, Russell H Taylor, and Peter Kazanzides. Safety design view: A conceptual framework for systematic understanding of safety features of medical robot systems. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1883–1888, 2014.

[16] Baraka Olivier Mushage, Jean Chamberlain Chedjou, and Kyandoghere Kyamakya. Fuzzy neural network and observer-based fault-tolerant adaptive nonlinear control of uncertain 5-dof upper-limb exoskeleton robot for passive rehabilitation. *Nonlinear Dynamics*, 87(3):2021–2037, 2017.

[17] Rachid Alami, Alin Albu-Schäffer, Antonio Bicchi, Rainer Bischoff, Raja Chatila, Alessandro De Luca, Agostino De Santis, Georges Giralt, Jérémie Guiochet, Gerd Hirzinger, et al. Safe and dependable physical human-robot interaction in anthropic domains: State of the art and challenges. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1–16, 2006.

[18] Milos Vasic and Aude Billard. Safety issues in human-robot interactions. In *2013 IEEE International Conference on Robotics and Automation (ICRA)*, pages 197–204, 2013.

[19] Matteo Parigi Polverini, Andrea Maria Zanchettin, and Paolo Rocco. A computationally efficient safety assessment for collaborative robotics applications. *Robotics and Computer-Integrated Manufacturing*, 46:25–37, 2017.

[20] Ons Amri, Majdi Mansouri, Ayman Al-Khazraji, Hazem Nounou, Mohamed Nounou, and Ahmed Ben Hamida. Improved model based fault detection technique and application to humanoid robots. *Mechatronics*, 53:140–151, 2018.

[21] Adrian Kampa. The review of reliability factors related to industrial robots. *Robotics and Automation Engineering Journal*, 3(5):1–5, 2018.

[22] Carlos Carreras and Ian D Walker. Interval methods for fault-tree analysis in robotics. *IEEE Transactions on Reliability*, 50(1):3–11, 2001.

[23] Dean L Schneider, Delbert Tesar, and J Wesley Barnes. Development and testing of a reliability performance index for modular robotic systems. In *Proceedings of Annual Reliability and Maintainability Symposium (RAMS)*, pages 263–271. IEEE, 1994.

[24] Martin L Leuschen, Ian D Walker, and Joseph R Cavallaro. Robot reliability through fuzzy markov models. In *Annual Reliability and Maintainability Symposium. 1998 Proceedings. International Symposium on Product Quality and Integrity*, pages 209–214. IEEE, 1998.

[25] Arthur J Critchlow. *Introduction to robotics*. Macmillan Pub Co, 1985.

[26] Violeta Dumitru and Mirela Cherciu. Application of the FMEA concept to medical robotic system. In *Advanced Engineering Forum*, volume 13, pages 324–331. Trans Tech Publ, 2015.

[27] MH Korayem and A Iravani. Improvement of 3P and 6R mechanical robots reliability and quality applying FMEA and QFD approaches. *Robotics and Computer-Integrated Manufacturing*, 24(3):472–487, 2008.

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

[29] 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.

[30] Rodney G Roberts and Anthony A Maciejewski. A local measure of fault tolerance for kinematically redundant manipulators. *IEEE Transactions on Robotics and Automation*, 12(4):543–552, 1996.

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

[32] Qian Li and Jing Zhao. A universal approach for configuration synthesis of reconfigurable robots based on fault tolerant indices. *Industrial Robot: An International Journal*, 39(1):69–78, 2012.

[33] Ian D Walker and Joseph R Cavallaro. The use of fault trees for the design of robots for hazardous environments. In *Proceedings Annual Reliability and Maintainability Symposium*, pages 229–235, 1996.

[34] Anthony A Maciejewski and Rodney G Roberts. On the existence of an optimally failure tolerant 7R manipulator Jacobian. *Applied Mathematics and Computer Science*, 5(2):343–357, 1995.

[35] Khaled M Ben-Gharbia, Anthony A Maciejewski, and Rodney Roberts. An illustration of generating robots from optimal fault-tolerant Jacobians. In *15th IASTED International Conference on Robotics and Applications*, pages 1–3, 2010.

[36] 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.

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

[38] 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.

[39] Rodney G Roberts, Hyun Geun Yu, and Anthony A Maciejewski. Fundamental limitations on designing optimally fault-tolerant redundant manipulators. *IEEE Transactions on Robotics*, 24(5):1224–1237, 2008.

[40] Hui Du and Feng Gao. Fault tolerance properties and motion planning of a six-legged robot with multiple faults. *Robotica*, 35(6):1397–1414, 2017.

[41] Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, 2015.

[42] Vojtěch Vonásek, Sergej Neumann, David Oertel, and Heinz Wörn. Online motion planning for failure recovery of modular robotic systems. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1905–1910, 2015.

[43] Kene Li and Yunong Zhang. Fault-tolerant motion planning and control of redundant manipulator. *Control Engineering Practice*, 20(3):282–292, 2012.

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

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

[46] Jing Zhao, Kailiang Zhang, and Xuebin Yao. Study on fault tolerant workspace and fault tolerant planning algorithm based on optimal initial position for two spatial coordinating manipulators. *Mechanism and Machine Theory*, 41(5):584–595, 2006.

[47] Hamid Abdi, Anthony A Maciejewski, and Saeid Nahavandi. Reliability maps for probabilistic guarantees of task motion for robotic manipulators. *Advanced Robotics*, 27(2):81–92, 2013.

[48] Zhao Jing and Fang Cheng. On the joint velocity jump during fault tolerant operations for manipulators with multiple degrees of redundancy. *Mechanism and Machine Theory*, 44(6):1201–1210, 2009.

[49] Kenneth N Groom, Anthony A Maciejewski, and Venkataramanan Balakrishnan. Real-time failure-tolerant control of kinematically redundant manipulators. *IEEE Transactions on Robotics and Automation*, 15(6):1109–1115, 1999.

[50] Rodney G Roberts, Rodrigo S Jamisola, and Anthony A Maciejewski. Identifying the failure-tolerant workspace boundaries of a kinematically redundant manipulator. In *2007 IEEE International Conference on Robotics and Automation*, pages 4517–4523, 2007.

[51] Zhao Jing and Li Qian. On the joint velocity jump for redundant robots in the presence of locked-joint failures. *Journal of Mechanical Design*, 130(10):102305, 2008.

[52] Qingxuan Jia, Tong Li, Gang Chen, Hanxu Sun, and Jian Zhang. Trajectory optimization for velocity jumps reduction considering the unexpectedness characteristics of space manipulator joint-locked failure. *International Journal of Aerospace Engineering*, 2016, 2016.

[53] Hamid Abdi and Saeid Nahavandi. Joint velocity redistribution for fault tolerant manipulators. In *2010 IEEE Conference on Robotics Automation and Mechatronics (RAM)*, pages 492–497, 2010.

[54] Hamid Abdi, Saeid Nahavandi, Yakov Frayman, and Anthony A Maciejewski. Optimal mapping of joint faults into healthy joint velocity space for fault-tolerant redundant manipulators. *Robotica*, 30(04):635–648, 2012.

[55] Monica L Visinsky, Joseph R Cavallaro, and Ian D Walker. Expert system framework for fault detection and fault tolerance in robotics. *Computers & electrical engineering*, 20(5):421–435, 1994.

[56] Warren E Dixon, Ian D Walker, Darren M Dawson, and John P Hartranft. Fault detection for robot manipulators with parametric uncertainty: A prediction-error-based approach. *IEEE Transactions on Robotics and Automation*, 16(6):689–699, 2000.

[57] Arun T Vemuri, Marios M Polycarpou, and Sotiris A Diakourtis. Neural network based fault detection in robotic manipulators. *IEEE Transactions on Robotics and Automation*, 14(2):342–348, 1998.

[58] Michael L McIntyre, Warren E Dixon, Darren M Dawson, and Ian D Walker. Fault detection and identification for robot manipulators. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 5, pages 4981–4986. IEEE, 2004.

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

[60] Giuseppe Muscio and Francesco Pierri. A fault tolerant adaptive control for robot manipulators. In *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1697–1702, 2012.

[61] Yu Kang, Zhijun Li, Yifan Dong, and Hongsheng Xi. Markovian-based fault-tolerant control for wheeled mobile manipulators. *IEEE Transactions on Control Systems Technology*, 20(1):266–276, 2012.

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

[63] Bo Zhao and Yuanchun Li. Local joint information based active fault tolerant control for reconfigurable manipulator. *Nonlinear dynamics*, 77(3):859–876, 2014.

[64] Dean L Schneider. Reliability and maintainability of modular robot systems: A roadmap for design. Technical report, Air Force Institute of Technology Wright-Patterson AFB OH, 1993.

[65] David Nicholls. *System reliability toolkit*. Riac, 2005.

[66] David Mahar, William Fields, and John Reade. *Nonelectronic Parts Reliability Data (NPRD-2016)*. Quanterion Solutions Incorporated, 2015.

[67] Robert C Thompson. Principal submatrices ix: Interlacing inequalities for singular values of submatrices. *Linear Algebra and its Applications*, 5(1):1–12, 1972.