Kinematic Design and Optimal Synthesis of Parallel Manipulators Using Monte Carlo Method

A Thesis
Submitted for the Degree of
Master of Science in Engineering
in the **Faculty of Engineering**

by

Arkadeep Narayan Chaudhury

Mechanical Engineering
Indian Institute of Science
Bangalore – 560 012 (INDIA)

December 2017

DEDICATED TO

My parents Nabendu Narayan Chaudhury and Banani Chaudhury,
My brother Abhishek Narayan Chaudhury

And to all my teachers, friends and well wishers

CERTIFICATE

I hereby certify that the content embodied in this thesis titled **Kinematic Design and Optimal Synthesis of Parallel Manipulators Using Monte Carlo Method** has been carried out by Mr. Arkadeep Narayan Chaudhury at the Indian Institute of Science, Bangalore under my supervision and that it has not been submitted elsewhere for the award of any degree or diploma.

Signature of the Thesis Supervisor: ....

Professor Ashitava Ghosal
Dept. of Mechanical Engineering
Indian Institute of Science, Bangalore

DECLARATION

I hereby declare that the content embodied in this thesis titled **Kinematic Design and Optimal Synthesis of Parallel Manipulators Using Monte Carlo Method** is the research carried out by me at the Department of Mechanical Engineering, Indian Institute of Science, Bangalore under the supervision Prof. Ashitava Ghosal, Department of Mechanical Engineering, IISc. In keeping with the general practice in reporting scientific observations, due acknowledgment has been made wherever the work described is based on the findings of other investigations.

Signature of the Author: ....

Arkadeep Narayan Chaudhury
Dept. of Mechanical Engineering
Indian Institute of Science, Bangalore

Acknowledgements

I take this opportunity to thank my adviser Prof. Ashitava Ghosal for his support and invaluable guidance throughout the course of my thesis work. I admire his patience in guiding me through the hurdles I faced during my research. I have learned many things from him and without his support through various lows during my studentship, I would not have come this far. As I graduate from IISc and go for a PhD, I hope to use my learning from him and have an enjoyable experience.

I would also like to acknowledge Professors G. K. Ananthasuresh, B. Gurumoorthy, Dibakar Sen and C. S. Jog for their help regarding various aspects of the research presented in the thesis. Prof. Ananthasuresh helped me by pointing me towards relevant literature that eventually formed the basis of my work on kinematic design of multi-fingered hands. Prof. Gurumoorthy helped me in formulating the equation of rolling between two surfaces. Though the work presented in the thesis is an approximate formulation, work on the exact formulation is currently underway. Prof. Sen arranged for me to use the Polhemus Liberty electromagnetic tracking device, which led to the experimental results documented as a verification of our method of obtaining the human hand workspace. Prof. Jog, though not directly, gave me various hints regarding some analytical formulations in the thesis and has always been an inspiration.

I would also like to thank the Department of Mechanical Engineering, IISc for making my stay at the institute fruitful and enjoyable. In particular, I would like to thank Mrs. Somavathi Devi. Without her help, all the official activities would been very stressful and time consuming.

A special thanks goes to my lab mates Ashith, Puneet, Shounak, Midhun, Ravi, Ashwin Bose, Sagar and Ashwin K. P. for making my stay at the Robotics lab enjoyable. My camaraderie with Ashith goes back to 2014 when I was am intern at the Robotics lab, and worked with him for a month on a small part of his PhD problem. I would cherish the various stimulating discussions with Shounak early on my stay at the lab which helped me take a lot of decisions, that proved to be helpful during my stay here. Thanks to Ashwin K. P. for making my last few months especially fruitful and interesting by involving me in a part of his current work.

A special thanks goes to my friends Santanu, Surya, Prasad and Atanu of the Combustion and

Spray lab for the refreshing weekly badminton sessions, and to my friends Dipanjan, Pabitra, Aritra and Tanmay for making my stay at IISc enjoyable.

I extend my sincere gratitude to my parents and my brother for the immense psychological support, without which, I would not be here today. Without their moral support, I would have been quite lost.

Finally, I am indebted to IISc for offering me the opportunity to pursue my master's degree in its intellectually stimulating academic atmosphere, and orienting me in the general direction towards a life in academia.

Arkadeep Narayan Chaudhury
Department of Mechanical Engineering
Indian Institute of Science
Bangalore.
June, 2018

Abstract

Parallel manipulators are widely used for a variety of tasks where increased accuracy and large load carrying capacities are required. In this work, we propose a Monte Carlo based method to represent and quantify the workspace of the manipulator and demonstrate the use of gradient based optimization techniques to synthesize the manipulator for a required workspace.

First, we define the “well-conditioned workspace” of the manipulator and show how the Monte Carlo method in conjunction with techniques from computational geometry can be used to represent and quantify the workspace of the manipulator. We also discuss the advantages and limitations of this method over the current state-of-the-art. Next, we demonstrate the technique through two examples of obtaining the workspaces of multi-fingered hands. In the first example, we propose a six degrees of freedom approximation of the human three fingered grasp and obtain the workspace of the same. We conduct experiments with human subjects and show that our formulation can obtain envelopes of the actual human hand workspaces with a maximum error of 2 cm. Additionally, we show that the human hand workspace is the largest when the cross section area of the grasped object is roughly equal to the palm area and the hand workspace is always larger when rolling is allowed at the object-finger contact point. In the second example, we obtain the workspace of the well known Stanford-JPL hand.

In the second part of the work, we show that the Monte Carlo method in conjunction with gradient based optimization techniques can be effectively used for optimal design of parallel manipulators for a required well conditioned workspace. We demonstrate this by two examples—in the first example, we consider a planar 2 degrees of freedom 5 bar manipulator and discuss 4 different shapes of the manipulator workspace due to different constraints on the link dimensions and how the choice of constraints affect the problem of optimal design for a prescribed workspace. In the second problem, we discuss the optimal synthesis of a Stewart platform in a special configuration defined by 6 geometric parameters. We consider two assembly modes of the manipulator and compare the workspaces obtained in each case. For both of the examples, we obtain the optimal manipulator dimensions for a given workspace and discuss the sensitivity of the manipulator workspace to perturbations in the design constraints.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Parallel manipulators are widely used for a variety of tasks where increased accuracy and large load carrying capacities are required. The most well-known spatial in-parallel manipulator, the Stewart-Gough platform has been used extensively for tire testing, aircraft simulators, machine tools and many other applications. Other platform type in-parallel manipulators, with three degrees-of-freedom such as the 3-UPU and 3-RPS manipulators have been proposed as a parallel wrist [1] for orienting an object and for tracking the sun for concentrated solar plants [2]–[4]. Hybrid parallel manipulators have been proposed as a model of multi-fingered hands (see, for example, the Stanford-JPL hand by Salisbury and Craig [5] and the Utah-MIT hand by Jacobsen et al. [6] and the 3 fingered hand by Borras and Dollar [7]). Planar multi-degree-of-freedom, closed-loop mechanisms such as the 3-RRR or a 5-bar mechanism have been used for precision manipulation in a plane, haptic devices (see Phantom range of haptic devices by Sensable [8] etc.). In most of these instances, the stress has been to obtain the solutions to the direct and inverse kinematics problems (see, for example, the pioneering works by Wen and Liang [9] and Raghavan and Roth [10] and the references contained therein), perform singularity analysis (see, for example, Bandyopadhyay and Ghosal[11], [12], [13] and the references contained therein), derive and numerically solve the dynamic equations of motion (see, for example, the comprehensive review by Dasgupta and Mruthyunjaya [14] and the references contained therein) and for control (see, for example, the works by Hatip and Ozgoren [15], Narasimhan [16], Wang et al. [17], Wen et al. [9] etc. and the references contained therein). Unlike the extensively studied planar four-bar and other one-degree-of-freedom planar mechanisms, there is relatively less literature on the design of parallel manipulators and multi-degree-of-freedom planar or spatial closed-loop mechanisms for a given set of objectives. In this thesis, we focus on the problem of representing and quantifying the *well-conditioned workspace* of a closed-loop mechanism or parallel manipulator and attempt the optimal dimensional synthesis of the manipulator, for a specified workspace, subject to specified joint limits.

Optimization of parallel manipulators and closed-loop mechanisms in terms of dimensional synthesis for the largest specified workspace and (or) highest end-effector accuracy is a continuing area of research. To address the problem to our capacity, we first disscuss the current state-of-the-art in obtaining and representing the workspace of a manipulator in section 1.2 and then the current state-of-the-art in optimal synthesis of parallel manipulators in section 1.3.

1.2 Workspace of a manipulator

The workspace of a manipulator, $\mathcal{W}_T$, in the most general sense, is the set of the positions and orientations that the end effector of the manipulator can reach ([18])[1]. Obtaining the workspace of any manipulator, is one of the most fundamental steps in the design of a manipulator. The workspace of the end-effector of a serial or a parallel 6 degree-of-freedom manipulator involves three quantities representing position and three quantities representing orientation of the end-effector. The total workspace of the manipulator is represented by taking the position and orientation workspace together. It is difficult, and often not very useful to visualize or define a volume of the total workspace and it is common in literature to seek quantities like *constant orientation workspace*, $\mathcal{W}_P$, or *constant position workspace*, $\mathcal{W}_O$, by independently seeking either of $\mathcal{W}_P$ or $\mathcal{W}_O$ ([20], [21]). Apart from these, for some manipulators, like the planar 3R robot, the planar 3-R$\underline{\text{R}}$R or the planar 3-R$\underline{\text{P}}$R robot, it is common to seek the *dexterous workspace*, which is the set of all locations of the end effector where all orientations of the end effector are possible. In the following sections, we describe techniques of obtaining the workspace of a manipulator.

1.2.1 Analytical methods

Analytical methods are arguably the best possible way for obtaining the workspace of a manipulator. Since these methods do not involve iteration, the analytical expressions can be used to get insight into the effects of the dimensions of the manipulators. For simple manipulators like the 4 R or 5 R planar parallel manipulators, many serial manipulators like the 2R, 3R, the SCARA, etc. the maximal workspace can be easily obtained in it's exact form. Simple formulations exist for dexterous manipulation workspace of the 3R serial chain [18], planar 3-R$\underline{\text{R}}$R or the planar 3-R$\underline{\text{P}}$R robot (see [19] and [22]). For more complex spatial manipulators, the concept of maximal workspace is not very useful due to the possible inclusion of different types

---

[1] This has been termed as the *maximal workspace* by Merlet [19]

of singularities in the workspace. Therefore analytical formulations of "*safe working zones*" excluding the singularities, put forth by Bandyopadhyay and co-workers (see [23] and [24]) for spatial manipulators are useful tools for design.

1.2.2 Interval analysis based methods

Another popular method of workspace evaluation involves discretization of the workspace of the manipulator by nodes, grids or boxes. After discretization, certain parameters are tested at the discrete node to classify them as nodes within or outside the workspace (see [19] for more details). A subsequent improvement on this technique is the use of interval analysis techniques for evaluate the workspace. Significant works have been done on this theme by Merlet and co-workers such as Chablat et al.[25], Hao and Merlet [26], Merlet [27], and most recently by Caro et al. [28] and Bohigas et al. [29]. Workspace evaluation using interval analysis based methods have some disadvantages regarding computational performance and are not always suitably applied to manipulators with higher degrees of freedom. These aspects are discussed in detail in chapter 3.

1.2.3 Monte Carlo method

Monte Carlo method involves statistical trials which can be efficiently used to evaluate integration problems in n-D. Rastegar and co-workers (see [30], [31]) were the first to formulate and solve the problem of obtaining the workspace of n-link serial manipulators. Similar works for obtaining the workspace boundaries using Monte Carlo method exist (see e.g. the works by Alciatore et al. [32]). Most of the aforementioned works involve simple formulations for serial manipulators which is inapplicable to the problem of evaluating the workspace of parallel manipulators due to the possibility of inclusions of singularities in the workspace. Tsai and co-workers (see Stamper and Tsai [33], Tsai et al. [34], using the definition of global condition index (see Gosselin [35]) were the first to define and evaluate the *well conditioned* workspace of a parallel manipulator using the Monte Carlo method. However, the definition of the well conditioned workspace that Tsai and co workers put forth was inadequate in a sense that it did not ensure "well conditioned-ness" for the angular and linear motion of the manipulator. In the current work, we have addressed the issue by defining Jacobians for both the linear and angular motion of the end effector and ensured that the manipulator Jacbians are *sufficiently* well conditioned through out the workspace obtained by our formulation. The Monte carlo technique has been discussed in detail in chapter 2 and its application in obtaining the workspace of a manipulator has been discussed in detail in chapter 3.

1.3 Optimal design of parallel manipulators

Optimal synthesis of parallel manipulators is a continuing theme of research and several researchers have proposed several avenues to address the problem. In a body of literature (see, for example, the work by Boudreau and Gosselin [36] and the book by Davidor [37] and the references contained therein), the authors have recognized the non-convex nature and difficulties in optimizing a parallel manipulator for desired characteristics and have thus not suggested the use of gradient based optimization, involving closed form kinematic equations of parallel manipulators. Genetic algorithm or other evolutionary algorithms have been chosen frequently by researchers (see, for example, the work by Grefenstette [38] and the references contained therein) for optimization problems including but not limited to dimensional synthesis. A different approach to the optimization problem (see, for example, Masory and Wang [39] and Tsai and Soni [40]) is by evaluating and maximizing the boundary curves of the feasible workspace of the manipulator at a particular plane with one of the Cartesian variables as fixed. Pittens and Podhorodeski [41] and Han et al. [42] have used gradient based optimization to obtain the dimensions of a manipulator for highest accuracy by reducing the condition number over the feasible workspace so that the accuracy of the manipulator is good everywhere in the workspace of the manipulator. Gosselin and Guillot [43] have worked on the optimization problem of planar parallel manipulators in Cartesian space. The method used by them obtains the geometric description of the intersection of the available workspace and obtained the workspace of the manipulator and subsequently minimizes the exclusion zone of the intersected workspace, thereby reaching at the optimum configuration. An avenue of research started with the pioneering works of Merlet (see e.g. [19], [27], [44] and the references contained therein) describe the use of interval analysis as a technique to determine the upper and lower bounds of a function and has proposed its use it for the optimal design of parallel mechanisms by maximizing a particular cost function. Methods based on numerical constraint programming exist, which represent and quantify non-singular workspaces of parallel manipulators. A recent work by Caro et al. [28] uses this technique in conjunction with branch and prune algorithms to compute general aspects of parallel manipulators like non-singular self-collision free workspaces. Borras and Dollar [7] have considered two versions of the same parallel manipulator – one as an under-actuated (or hybrid parallel) manipulator and another as a fully actuated version and have generated optimum dimensions of both for the maximal precision workspaces. In the work they have also computed the actual number of configurations (reported to be of the order of 10^7 for the worst case) to be searched through and have suggested random search technique to quickly go through the search space. Among the plethora of techniques discussed above, in the following sections, we point out the most prominent of those and discuss the current state of the art in optimal synthesis of parallel manipulators.

1.3.1 Analytical methods

As it is evident, optimal design of parallel manipulators for the largest workspace, given certain upper bounds on link lengths is not useful for design. Instead, it is common to pose and solve the optimization problem to maximize certain aspects of the motion of the manipulator like dexterous workspace, constant orientation workspace, accuracy etc. (see e.g. Kurtz and Hayward [45]). One prominent theme in optimal design of manipulators was proposed by Gosselin [35] by providing analytical formulations for optimal design of manipulators which maximizes the global condition index (GCI) over the entire workspace ($\mathcal{W}$). The "global condition index" is defined as follows:

$$GCI = \frac{\int_{\mathcal{W}} \frac{1}{\kappa} d\mathcal{W}}{\int_{\mathcal{W}} d\mathcal{W}} \tag{1.1}$$

Where, in equation (1.1), κ is the condition number of the manipulator Jacobian at a given position of the manipulator workspace. This particular formulation has been used extensively in literature, (see e.g. [44], [46] and [47]). In his work, Gosselin [35] presents some interesting results, albeit for simple serial manipulators,– for a 2-R planar robot, the GCI is maximized across the entire workspace if the link lengths are in the ratio $2/\sqrt{2}$ and for a three link open chain spherical manipulator, the GCI is maximized across the entire workspace if the revolute joints connecting the second link to the first link and the third link to the second one are are placed at right angles to each other. However, the same formulation loses it's analytical nature for parallel manipulators mainly because the integral in equation (1.1) cannot be evaluated symbolically.

1.3.2 Search based optimization methods

As discussed in section 1.2.2, there are a number of works where the workspace has been evaluated by using interval analysis based methods. Some of these works have been extended to obtain optimal manipulator dimensions for the parallel manipulators. The most notable of them is the work by Hao and Merlet [26], where the authors have used interval analysis to quantify the manipulator workspace and have subsequently formulated and solved an optimization problem to satisfy the objective of given workspace requirement and accuracy requirements. Lara-Molina et al. [46] have used Gosselin's definition of the global condition index and have subsequently used multi objective genetic algorithms to arrive at an optimal solution. In another avenue of work, Lou and co-workers (see e.g. [48]–[50]) have defined the *effective regular workspace* of a parallel manipulator as a regular object (e.g. a cube or a parallelopiped) in a transformed space and have used Controlled Random Search (CRS) technique (see Törn and Žilinskas [51]) to obtain the globally optimum manipulator. The main limitation of the work is that the globally optimum design parameters are not always practical as the authors have encountered certain *optimal* manipulators where some of the dimensions are an order of magnitude different than the others– thus significantly reducing load carrying capacity.

1.3.3 Gradient based methods

Since the Monte Carlo method can be used to obtain the workspace, one can use an objective function and a gradient based approach for optimization together with the Monte Carlo method. Stamper et al. [33], Tsai and Joshi [52] use Monte Carlo search based methods to optimize manipulators for the largest well-conditioned workspace. Our work closely follows the work by Stamper et al. and Tsai and Joshi with significant improvements on the definition of well conditioned workspace, implementation of the Monte Carlo method and provides many insights on the optimization problem. We also show demonstrate the sensitivity of the manipulator workspace to perturbations in design constraints.

1.4 Contributions of this thesis

The main contributions of the thesis are:

- We show that the Monte Carlo method can be effectively used to represent and quantify the well conditioned workspace of a parallel manipulator. Advantages and limitations of the Monte Carlo method as compared to other well known methods for representing the workspace have been discussed.

- Using the Monte Carlo method as the primary tool we design a 6 degrees of freedom parallel manipulator to emulate the kinematic structure of the human three-fingered grasp. We pose and analytically solve the inverse kinematics problem of the manipulator and define and obtain the well-conditioned workspace of the manipulator. In addition to that, we model the human finger-tips by super-ellipsoids, model the collision of the human fingertips during three fingered manipulation and model the rolling of objects on fingertips during manipulation and provide envelopes of the human hand workspace. Further, we qualify the envelopes to be reasonably conservative by performing experiments with human subjects.

- We also show two generic results – a) the human hand workspace, modeled as a 6 degrees of freedom parallel manipulator is the largest when the object the hand is manipulating has the same cross section area as the palm, and b) the hand workspace is larger (about 1.2 to 1.5 times) when rolling of the fingertips is allowed versus when the fingertips are modeled as spherical joints. We also obtain the workspace of the well known Stanford-JPL hand.
- We demonstrate that Monte Carlo method can be used in conjunction with gradient based optimization methods for optimal synthesis of parallel manipulators by two examples. First we undertake the optimal synthesis of a planar 5R parallel manipulator for a given workspace and then we discuss the optimal synthesis of a Stewart platform manipulator. Finally, we discuss the sensitivity of the manipulator workspace on design constraints for the multi-fingered hands, the 5R planar parallel manipulator and the Stewart platform manipulator.

1.5 Preview

The rest of the thesis is organized as follows: In chapter 2 we review the key mathematical concepts used in the work, following which in chapter 3 we demonstrate how the Monte Carlo method, with techniques from computational geometry can be used to quantify and represent the workspace of a manipulator. In chapter 4 we discuss the problem of optimal synthesis of parallel manipulators for a required workspace. Finally, in chapter 5 we conclude the thesis by summarizing the current work and proposing a few possible directions for future research.

Chapter 2

A review of mathematical concepts

2.1 Introduction

In this chapter we discuss the key mathematical concepts used in the thesis. We start with a brief outline of the Monte Carlo method in section 2.2, following which, we discuss in brief about α shapes and Delaunay triangulation, the key techniques used for representing the workspace of a manipulator in section 2.3. In this section we also discuss formulations for classifying points with respect to a triangulated domain in 3D and calculating the shortest distance of a point from a triangulated domain. In section 2.4 we describe a methodology to approximate human finger tips a super ellipsoids taking data from 3D scans of the human hand. Finally we conclude the chapter by discussing collision detection of two super ellipsoids in space without calculating their distance of nearest approach in section 2.5.

2.2 The Monte Carlo method

The Monte Carlo method can be used to evaluate integrals of arbitrary functions (vector or scalar function of smooth or non-smooth type) over an arbitrary domain [53]. The integral

$$\mathscr{I} = \int_{[0,1]^d} f\left(\mathbf{x}\right)\,\mathrm{d}\mathbf{x}$$

where $f(\cdot)$ is a bounded real valued function, can be obtained as $E(f(\mathbf{U}))$ where $E(\cdot)$ is the expectation of a variable taking a particular probabilistic value, and $\mathbf{U} = [u_1, u_2, ..., u_d]^T$ a $1 \times d$ vector taking random values of $u_i \in [0,1]$ $\forall i = 1, 2, ..., d$. From the strong law of large numbers,[1] the average,

$$S_N = \frac{1}{n}\sum_{i=1}^{n} f\left(u_i\right) \tag{2.1}$$

almost surely [2] converges to $E(f(\mathbf{U}))$ as $n \longrightarrow \infty$. The volume (area) of a manipulator workspace is an integration problem in $\Re^3$ ($\Re^2$) and we use the Monte Carlo method to evaluate the volume (or area) of the workspace of a closed-loop mechanism or a parallel manipulator. It maybe mentioned that there are existing deterministic approaches to determine the volume of the workspace of a closed-loop mechanism or a parallel manipulator (see, for example, Masory et al. [39] and Merlet [19]).

In the following section we show that in the probabilistic approach of using the Monte Carlo method, the error bound can be made smaller than in the deterministic approach with less computational effort and less complexity. This is discussed in brief next - for more details, the reader is referred to Chapter 2 of the book by Fishman [54] and relevant sections of the book by Hammersley et al. [55].

2.2.1 Obtaining volume of a hyper-solid in n-D space by using Monte Carlo method

Let $\mathscr{R}$ denote the region of unknown volume $\lambda(\mathscr{R})$ in the m-D hypercube denoted by $\mathscr{I} = [0,1]^m$. Assume that the region $\mathscr{R}$ is arbitrary and given by a known series of inequalities and implicit functions of several variables in a way that the computation of the exact volume of $\mathscr{R}$, i.e., $\lambda(\mathscr{R})$ is not possible or computationally prohibitive. We further assume that a systematic procedure exists[3] for generating a sequence $\mathscr{H}_{m,N}$ of N points in m-D hypercube

$$\mathscr{H}_{m,N} = \{\mathbf{X}^j = (x_1^{(j)}, x_2^{(j)}, x_3^{(j)}, ...x_m^{(j)})\}. \tag{2.2}$$

The numerical accuracy of obtaining $\widehat{\lambda(\mathscr{R})}$, an estimate of the volume, depends on how $\mathscr{H}_{m,N}$ is populated[4] . The Algorithm 1 given below can be used to obtain $\widehat{\lambda(\mathscr{R})}$ by populating the vector $\mathbf{X}$ in equation (2.2) by generating m random numbers for x_i^s between $[0,1]$. From Algorithm 1,

---

[1] For a sequence of independent,uniformly distributed real random variables $\mathbf{X} = \{x_1, x_2, x_3, ..., x_n\}$ such that $E(f(X_i)) < +\infty$ then $\underset{n\to\infty}{lim} \frac{1}{n}\sum_{i=1}^{n} x_i = E(\mathbf{X})$ with probability 1.

[2] It can be proved that the probability of this convergence is 1.

[3] Chebyshev intervals $x_k = \cos\left(\frac{2k-1}{2n}\right)\pi, k = 1,2,...,n \ \forall x\epsilon[0,1]$ or any other non-repeating, monotonic sequence, see [54], may be used.

[4] Chebyshev intervals, quadrature or an ad-hoc interval generator or probabilistic methods like drawing random numbers can be used.

a point estimate of the quantity $\widehat{\lambda(\mathscr{R})}$ is obtained as $\frac{S}{N}$. Where S is the accumulated value of $\phi(\mathbf{X}^{(j)})$ in each case a randomly generated point is accepted because of its inclusion in the domain of $\mathscr{R}$. The variance $V(\widehat{\lambda(\mathscr{R})})$ can be obtained as

$$\sigma^2(\widehat{\mathrm{W}}) = \frac{S\ N - S^2}{N^2(N-1)} \tag{2.3}$$

```
Purpose : To obtain an estimate of $\lambda(\mathscr{R})$
Input: $\mathscr{R}$ and sample size 'N'
Output: $\widehat{\lambda(\mathscr{R})}$
 1: Initialize j=0, S=0;
 2: Allocate memory for $\mathscr{H}_{m,N}$
 3: while j $\leqslant$ N do
 4:     Initialize i=1;
 5:     while i $\leqslant$ m do
 6:         Populate $\mathbf{X}^j$ using $\mathscr{H}$ such that $x_i^s$ between $[0,1]$;
 7:     end while
 8:     Initialize a counter $\phi(\mathbf{X}^{(j)}) = 0$;
 9:     if $\mathbf{X}^j \ \epsilon\ \mathscr{R}$ then
10:         Assign $\phi(\mathbf{X}^{(j)}) = 1$;
11:         $S = S + \phi(\mathbf{X}^{(j)})$;
12:         $j = j + 1$;
13:     end if
14: end while
15: Compute $\widehat{\lambda(\mathscr{R})} = \frac{S}{N}$
```

Algorithm 1: Algorithm for evaluating volume of a hyper-solid using Monte Carlo Method

In may be mentioned that in line ♯6 of Algorithm 1, if we use an unbiased random number generator for $\mathscr{H}$ then it is a Monte Carlo method. In the Monte Carlo method, the bounds on the error is probabilistically determined and can be reduced by proper choice of number of samples and other parameters.

The worst-case error in estimating the volume, by using deterministic samples, is given by [54] as:

$$\parallel \widehat{\lambda(\mathscr{R})} - \lambda(\mathscr{R}) \parallel \leqslant \frac{s(\mathscr{R})}{n^{1/m}} \tag{2.4}$$

where $s(\mathscr{R})$ is the surface area of the hyper-surface bounding the region $\mathscr{R}$[1]. Therefore to have

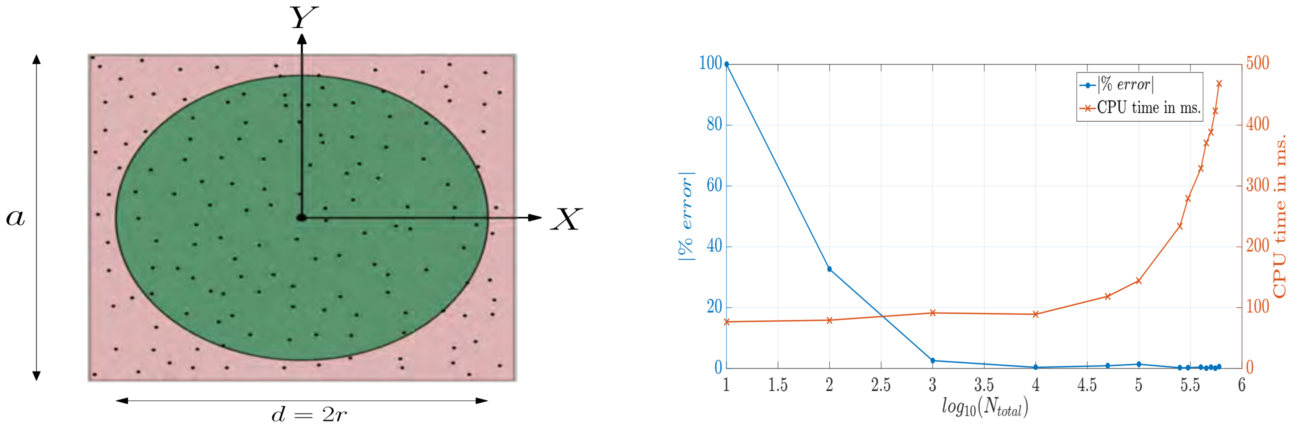
---

[1] The expressions in equation (2.4) are unit less because they have been derived from a counting argument.

an absolute error no larger than ϵ, the number of evaluation points required is,

$$n(\epsilon) = \lceil [\frac{s(\mathscr{R})}{\epsilon}]^m \rceil \tag{2.5}$$

where, $\lceil x \rceil$ is the nearest integer greater or equal to (x). From equation (2.5) it is clear that the required sample size $n(\epsilon)$ is exponentially related to the dimensionality of the problem. Therefore, for deterministic methods, for a change in the order of the allowable absolute error or an increase in dimension of $\mathscr{R}$, there is an exponential and unbounded[1] increase in the number of points at which the step #9 in Algorithm 1 has to be evaluated. In the subsequent section, we show that for lesser computation effort, we can achieve more accuracy by using the Monte Carlo method. It may also be noted that the evaluation of the *while* loop in line # 3 through line # 14 and the inner *if* loop (line # 9 through line # 13) can be executed in parallel while the program is being used to evaluate $\phi(\mathbf{X}^{(j)})$. This possibility of parallelization is one of the biggest advantages of the Monte Carlo method. Figure 2.1b shows a basic example of obtaining the area of a circle using the method. We notice from figure 2.1b that at about 10^5 samples, the error is close to 0.5%.



(a) Obtaining the area of a circle using Monte Carlo method

(b) Computational performance of Monte Carlo method

Figure 2.1: Monte Carlo method- A basic example

---

[1] With finer error tolerances it can be shown that n tends to ∞.

2.2.2 Errors and sample size considerations

From equation (2.3) it can be seen that with increasing number of evaluation points n, the variance of the estimate $\widehat{\lambda(\mathscr{R})}$ decreases. It can be proved that

$$\Pr(\underset{n\to\infty}{lim}\widehat{\lambda(\mathscr{R})} = \lambda(\mathscr{R})) = 1(a.s.). \tag{2.6}$$

which implies that we can get an error free estimate as the number of evaluation points tend to infinity. Furthermore by using the Chebyshev inequality[1], we can prove (for more details see chapter 2 of the book by Fishman [54]),

$$\underset{n\to\infty}{lim}\Pr(\mid \widehat{\lambda(\mathscr{R})} - \lambda(\mathscr{R})) \mid\geqslant \epsilon) = 0, \tag{2.7}$$

which also provides us with a basis to obtain the sample size requirements. However, unlike the deterministic methods we cannot obtain the required sample size from the information about error tolerance alone – a confidence level $(1-\delta)$ is required to account for the randomness of the samples. Using Chebyshev's inequality, we can write,

$$\eta(\epsilon,\delta,\lambda) = \lceil\frac{\lambda\times(1-\lambda)}{\delta\epsilon^2}\rceil, \lambda = \lambda(\mathscr{R}) \tag{2.8}$$

From Algorithm 1, we know that $\lambda(1-\lambda)\leqslant\frac{1}{4}$. Since S given by $\sum_{i=1}^{n}\phi\left(\mathbf{X}^{(j)}\right)$ follows a binomial distribution, $V(\widehat{\lambda}(\mathscr{R})) = \frac{V(S)}{n^2} = \lambda(1-\lambda)/n$, and hence the worst case sample size is

$$\eta_c(\epsilon,\delta) = \lceil\frac{1}{4\delta\epsilon^2}\rceil \tag{2.9}$$

By comparing equations equation (2.9) and equation (2.5) and from Algorithm 1 we can state the following:

- The worst case sample size in case of the Monte Carlo method given by equation (2.9) is independent of the dimension m of the hyperspace $\mathscr{I}^m$ and this is one of the most desirable features of the Monte Carlo method.
- In step ♯6 of Algorithm 1 the time required for calculation of $\mathbf{X}^j$ is $\mathscr{O}(m)$ for random sampling, whereas, for the deterministic method, the same step requires $\mathscr{O}(m^\beta)$, where, $\beta > 1$ and is dependent on the method used for generating sample points. For quadrature

---

[1] Chebyshev's inequality: For a random variable $\mathbf{Z}$ with a probability density function (PDF) f defined on $(-\infty,\infty)$, with $E(\mathbf{Z}) = 0$ and $\sigma^2 = V(\mathbf{Z}) = E(\mathbf{Z}^2)$, and $\beta > 0$, then $\Pr(\frac{\mathbf{Z}}{\sigma}\geq\beta)\leq\frac{1}{\beta^2}$.

methods or Chebyshev intervals due to the presence of non-linear terms like radical signs or trigonometric functions, $\beta > 1$.

- The complexity of Algorithm 1 (while using the Monte Carlo method) is $\mathscr{O}(m)$ for a given (ϵ, δ) error tolerance as obtained from equation (2.9). However, for deterministic sampling, it can be proved (see [54]) that the complexity is exponential $\mathscr{O}\left(\frac{m^{\beta}}{4\delta\epsilon}\right)$ for the same preset tolerance.

From the above three observations, we can conclude that obtaining the workspace volume of a closed-loop mechanism or a parallel manipulator can be obtained more efficiently by using a probabilistic method than with a deterministic method. For more details on sample size and error estimates one can refer to Kleijen et al. [56] and references [54], [55]. In this work, we have use the Monte Carlo method to obtain the volume of the workspace of the closed-loop mechanism and the parallel manipulator. For most of the algorithms discussed later we have used:

- Sample size of $150,000$ or more which gives an error tolerance of less than 1 percent and a confidence bound of approximately $\delta = 0.05$. This is similar to the work by Stamper et al. [33] where 200,000 samples have been used.
- Uniformly distributed random numbers (between $[0, 1]$) were generated by using the pseudo-random number generator `rand` function of `Matlab` [57].

2.2.3 Importance sampling: A variation reduction technique in Monte Carlo simulation

The Monte Carlo method is quite efficient for obtaining the *well conditioned* workspace of parallel manipulators. However, in the previous case, the *obtained workspace* spanned through a large part $\{\sim 10 - 15\%\}$ of the search-space of the Monte Carlo simulation therefore, a large portion of the simulated histories $\{\geq 10\%\}$ were successful. With a relatively large number of samples considered $\{\geq 100,000$ in almost all the cases$\}$, the variation in the estimate obtained for the workspace was within acceptable limits $\{\leq 0.5\%\}$. However, in case we want to segregate a larger *enveloping* workspace (obtained by considering certain parameters) and a smaller *enveloped* workspace, the naive Monte Carlo method[1] will generally fail because the volume of the required workspace is much smaller (less than 1%) than the search space volume. In theory, we can use the naive Monte Carlo method and simulate a large number

---

[1] The Monte Carlo method drawing uniformly distributed random samples throughout the search space

of samples for obtaining a very fine resolution of both of the workspaces while adding a lot of unwanted computational load. Alternatively, we can try a variation reduction technique for Monte Carlo simulation known as *Importance Sampling* method to tackle the same problem. In what follows, we describe the technique in short. The naive Monte Carlo method draws uniform samples (having a PDF $f(\mathbf{X})$) across the entire volume of search space to evaluate $\mathcal{F}(\mathbf{X})$ and according to the definition of the Monte Carlo method, the expected value of $\mathcal{F}(\mathbf{X})$ across the searchspace is given as,

$$\langle \mathcal{F} \rangle = \int\limits_{V_s} \mathcal{F}(\mathbf{X}) f(\mathbf{X}) d\mathbf{X} \simeq \bar{\mathcal{F}} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{F}(X_i) \tag{2.10}$$

However, most of the search space volume (V_s) does not contribute to the value of $\langle \mathcal{F} \rangle$ as the PDF $f(\mathbf{X})$ samples the space infrequently. So the estimate obtained from equation (2.10), by going through a reasonable number of samples, is not a good one because the larger variance of the expected value and poor resolution of the workspace. Therefore, it seems reasonable to sample preferentially from a region where the integrand $|\mathcal{F}(\mathbf{X}) f(\mathbf{X})|$ has a relatively large magnitude. To effect such a biased sampling, the integral in equation (2.10) can be transformed by introducing another arbitrary (but known/chosen) PDF $f^*(\mathbf{x})$ as,

$$\langle \mathcal{F} \rangle = \int\limits_{V_s} \frac{\mathcal{F}(\mathbf{X}) f(\mathbf{X})}{f^*(\mathbf{X})} f^*(\mathbf{X}) d\mathbf{X} \equiv \int\limits_{V_s} \mathcal{F}^*(\mathbf{X}) f(\mathbf{X}) d\mathbf{X} = \langle \mathcal{F}^* \rangle \tag{2.11}$$

where, in equation (2.11), $\mathcal{F}^*(\mathbf{X}) = \mathcal{F}(\mathbf{X}) W(\mathbf{X})$, and the weight or *importance function*[1] is $W(\mathbf{X}) \equiv \dfrac{f(\mathbf{X})}{f^*(\mathbf{X})}$. Therefore, from equations (2.10) and (2.11) we can say,

$$\langle \mathcal{F} \rangle = \langle \mathcal{F}^* \rangle \simeq \bar{\mathcal{F}^*} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{F}(X_i) W(X_i) \tag{2.12}$$

It is clear from equation (2.12) that both $\mathcal{F}(\mathbf{X})$ and $\mathcal{F}^*(\mathbf{X})$ have the same expectations when X_i are sampled from $f^*(\mathbf{X})$. However, the variance of the two expected values $\langle \mathcal{F} \rangle$ and $\langle \mathcal{F}^* \rangle$ are,

$$\sigma^2(\mathcal{F}) = \langle \mathcal{F}^2 \rangle - \langle \mathcal{F} \rangle^2 \tag{2.13}$$

$$\sigma^2(\mathcal{F}^*) = \langle \mathcal{F}^{*2} \rangle - \langle \mathcal{F}^* \rangle^2 = \langle \mathcal{F}^{*2} \rangle - \langle \mathcal{F} \rangle^2 \tag{2.14}$$

---

[1] This is also called likelihood ratio or Radon-Nikodym derivative in literature see e.g. [58]

Comparing the variances of the two expected values of $\langle \mathcal{F} \rangle$ and $\langle \mathcal{F}^* \rangle$ from equations (2.13) and (2.14), we have, $\sigma^2(\mathcal{F}^*) - \sigma^2(\mathcal{F}) = \langle \mathcal{F}^{*2} \rangle - \langle \mathcal{F}^2 \rangle$. Using the definition of W, and equations (2.10) and (2.11), we can write

$$\begin{aligned} \langle \mathcal{F}^{*2} \rangle &= \int\limits_{V_s} \mathcal{F}^{*2}(\mathbf{X}) f^*(\mathbf{X}) d\mathbf{X} = \int\limits_{V_s} \mathcal{F}^2(\mathbf{X}) W^2(\mathbf{X}) f^*(\mathbf{X}) d\mathbf{X} \\ &= \int\limits_{V_s} \mathcal{F}^2(\mathbf{X}) W(\mathbf{X}) f(\mathbf{x}) d\mathbf{X} \leq \int\limits_{V_s} \mathcal{F}^2(\mathbf{X}) f(\mathbf{X}) d\mathbf{X} = \langle \mathcal{F}^2 \rangle \end{aligned} \tag{2.15}$$

It is clear from equation (2.15) that an improvement in the measure of the variance of $\langle \mathcal{F}^* \rangle$ over $\langle \mathcal{F} \rangle$ can be made if we can ensure that the value of $W(\mathbf{X}) < 1$ in the regions of the search space which contributes most to the value of $\langle \mathcal{F} \rangle$ as shown in equation (2.10). Therefore, by choosing a proper distribution $f^*(\mathbf{X})$, preferably from some information about the underlying process contributing to the success of a simulated sample, a significant improvement can be made in calculation of the estimate of the function over a naive Monte Carlo simulation.

We demonstrate the efficiency of importance sampling compared to naive Monte Carlo method through the following example.

Task: To obtain the volume and representation of a spherical zone of radius 0.1 units in a search space of 10 unit^3 with its center located approximately at $\frac{1}{3}^{rd}$ of X direction, $\frac{1}{4}^{th}$ of Y direction and $\frac{1}{5}^{th}$ of Z direction.

Solution: Since the approximate position of the spherical zone is known, we choose a multivariate normal distribution

$$f^*(\mathbf{X}) = \frac{1}{(2\pi)^{1.5}\sqrt{|\Sigma|}} \exp(-0.5[(\mathbf{X} - \mu)^T \Sigma^{-1} (\mathbf{X} - \mu)]) \tag{2.16}$$

where in equation (2.16) we choose, $\mu = [3.33\ 2.50\ 2.0]$ and $\Sigma = bI_3$, where I_3 is a 3×3 identity matrix and b is a positive real number. For this case $b = 1$ is a good choice because 99.7% of the simulated points will be inside $\mu \pm 3\sigma$ and are expected to cover the zone of interest. The choice of the mean vector μ is clearly influenced by the information about the approximate location and size of the spherical zone. The results of a numerical experiment to reflect the same are given in figure 2.2. The normally distributed random numbers for the experiment were drawn using the Box Muller method [59]. It is seen that importance sampling Monte Carlo method is better both in terms of computational performance and accuracy[1].

---

[1] CPU times are for Matlab [57] run in a 64bit Windows 7 PC with a Intel XEON processor (4 cores @ 3.10 GHz) and 16GB RAM

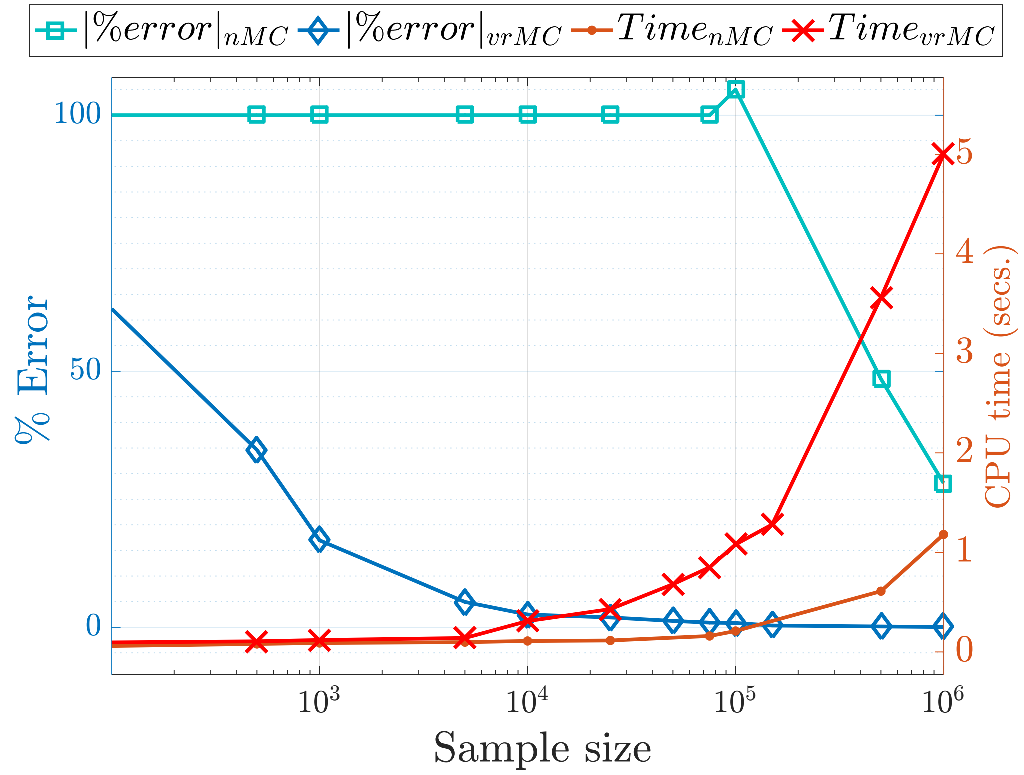


Figure 2.2: Computational performances of two Monte Carlo methods (Symbols in figure: $nMC \rightarrow$ Naive Monte Carlo and $vrMC \rightarrow$ Variation reduction Monte Carlo)

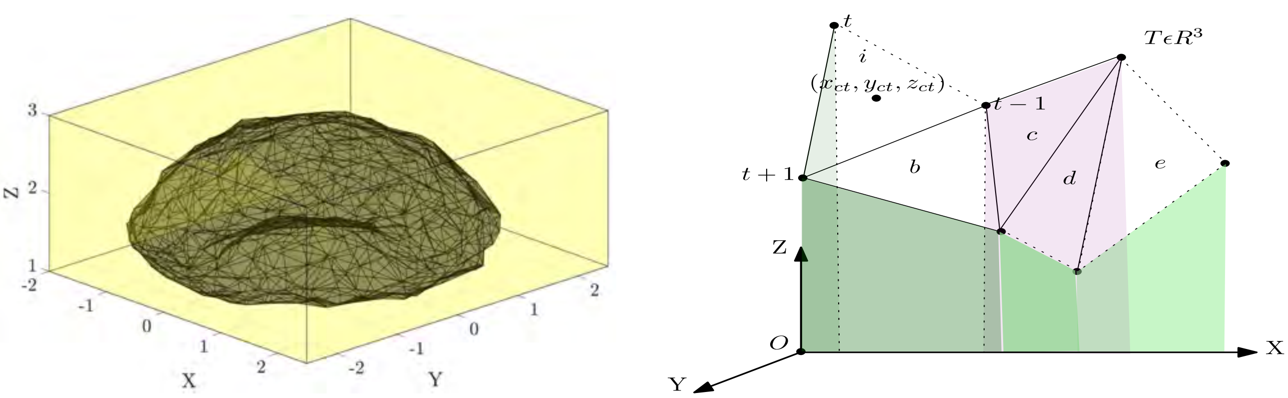
We will use the variation reduction technique later (see section 2.5) to efficiently detect the collision of super-ellipsoids while obtaining the workspace of the human hand .

2.3 Visualizing the workspace

For manipulators with uninterrupted travel between joint limits for all active and passive joints, the workspace is bounded by continuous surfaces or by continuous level set curves (see, Merlet [19], Masory et al. [39] and Tsai et al. [40]) and therefore, the analytical evaluation of the workspace qualifies to be a problem of integration in 3D and hence can be solved by using the Monte Carlo method. In chapter 3 we discuss the definition we have used to obtain the *well conditioned workspace* and as a precursor to that, in the following parts, we discuss a deterministic method of obtaining the volume (area) of a triangulated domain, post processing steps for the obtained cloud of points representing the workspace and techniques and measures we have used to compare the theoretically obtained workspace with it's experimentally obtained equivalent.

2.3.1 α shapes and Delaunay Triangulation

From the cloud of points obtained from the Monte Carlo simulation, we can also obtain *almost exactly* the volume of the workspace expressed as a polyhedron. This involves obtaining the *alpha shape* of the cloud of points and triangulating them by well known Delaunay triangulation algorithms([60]). Figure 2.3a shows the workspace of a Stewart platform manipulator. The gray triangulated domain in the middle is the α shape representation of the workspace of the manipulator, rendered as a 3D object and the yellow box around it is the search space. Following Edelsbrunner [61] we can informally define the α shape of a set of points $\mathbb{S}, \mathbb{S} \in \Re^3$ as a *polytope*[1] which is not necessarily connected or convex. The α shape of $\mathbb{S}$ depends on the value of $\alpha, 0 \leq \alpha \leq \infty$. For $\alpha = \infty$ the α shape of $\mathbb{S}$, given by the script letter ($\mathcal{S}$), is identical to the convex hull of $\mathbb{S}$, but as the value of α decreases, $\mathcal{S}$ may develop holes or cavities. The concept of α shapes, again due to Edelsbrunner [61], may be visualized as follows. We assume that the points of $\mathbb{S}$ are hard particles embedded in a block of Styrofoam representing $\Re^3$. We use a ball-shaped tool of radius α to carve the Styrofoam, leaving behind the particles which constitute $\mathbb{S}$. So it is intuitive that if $\alpha = \infty$ the points in $\mathbb{S}$ is encountered with an open half space and the resulting α shape $\mathcal{S}$ is the convex hull of $\mathbb{S}$ as α, decreases, the boundaries of $\mathcal{S}$, denoted by $\partial\mathcal{S}$ shrinks around the points in $\mathbb{S}$. In the event of $\alpha = 0$, the cloud of points $\mathbb{S}$ is the α shape of itself. Throughout the rest of the thesis, the obtaining of the α shape of $\mathbb{S}$ has been done while ensuring that no major feature of the workspace (like a hole/void etc.) has been ignored.



(a) Relative representation of workspace and search space

(b) Volume enclosed by a triangulated domain by measuring the volume of discrete trapezoids

Figure 2.3: Workspace and analytical method for obtaining the volume of the workspace

---

[1] A geometric shape with flat sides. In $\Re^3$ polytopes are polyhedra.

The step of triangulating the resultant α shape of the cloud of points obtained from the Monte Carlo simulation is required to visualize the workspace in $\Re^3$. For this, we use standard Delaunay triangulation algorithms (see e.g. Lee and Schachter [60]). The Delaunay triangulation algorithm, gives a data structure which lists the vertices and faces of the triangles that constitute the boundary of $\mathcal{S}$, $\partial\mathcal{S}$. Throughout the thesis, the workspaces have been visualized by rendering $\partial\mathcal{S}$ as a triangulated domain.

2.3.2 Deterministic methods for obtaining the area/volume

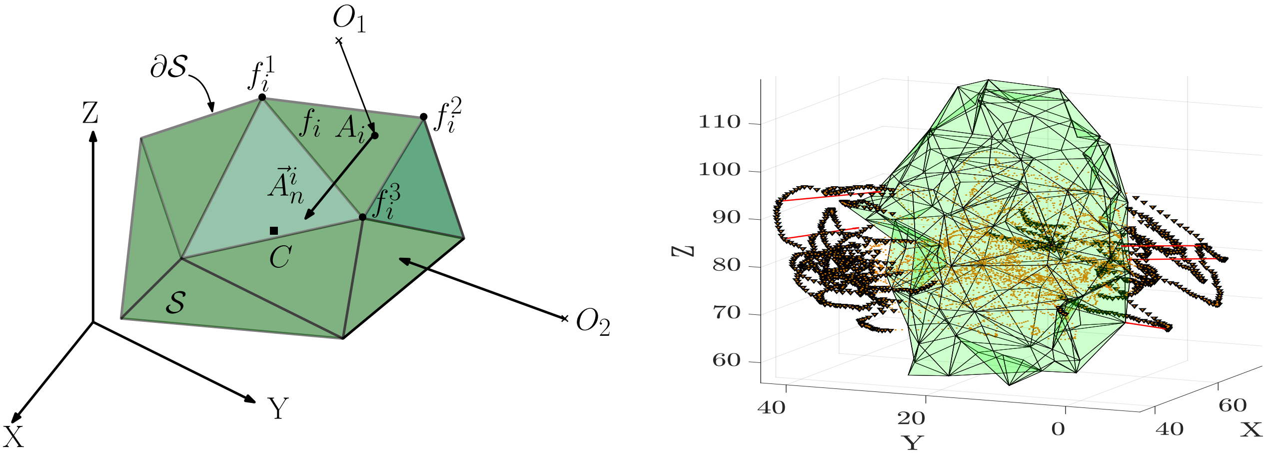
After the 3D polyhedron has been obtained (as shown in figure 2.3a) we obtain the volume using a generalization of the trapezoidal rule in $3D$ (see Allgower et al. [62], [63]). The algorithm considers the domain $D \in \Re^3$ as a set of discrete trapezoids $p(\sigma)$ with one of the non-parallel faces as the facet (σ_i), and another as the projection of the facet on a chosen plane (the $X - Y$ plane in figure 2.3b) and the other faces parallel to an axis perpendicular to the plane (the Z axis in case of figure 2.3b). According to [63], the volume V of the trapezoid is,

$$V = (-1)^2 \sum_{\sigma \in T} \left[\left(\frac{1}{2} \sum_{i=1}^{2} z^{ct^2}_{\;\;i} \right) \cdot \frac{1}{2!} \det \begin{bmatrix} 1 & 1 & 1 \\ x_i^{t-1} & x_i^t & x_i^{t+1} \\ y_i^{t-1} & y_i^t & y_i^{t+1} \end{bmatrix} \right] \tag{2.17}$$

where the outermost summation $\sum_{\sigma \in T}$ indicates that the summation is carried over the total triangulated domain T. According to Allgower et al. [63], this method is more efficient with complexity $\mathscr{O}(n^2)$ as compared to other methods such as finding the sum of the volumes of discrete tetrahedra in which case the complexity is $\mathscr{O}(n^3)$. It may be noted that equation (2.17), works only for evaluating the volume of a simply connected domain, as opposed to Monte Carlo method which applies for continuous as well as discontinuous domains.

2.3.3 Classifying points with respect to a meshed domain

The workspace of a manipulator, is obtained as a cloud of points $\mathbb{S}$ in $\Re^3$ and visualized as $\mathcal{S}$ enclosed by a triangulated surface $\partial\mathcal{S}$. However, the experimentally determined workspace of a manipulator (see section 3.3.7.1 for details on the experimental technique) is also obtained as a cloud of points in and around the region bounded by $\partial\mathcal{S}$. To quantify the effectiveness of our formulation, we need to measure two quantities– a) total number of points of the experimentally obtained workspace which are outside $\mathcal{S}$ and b) the distance of a point outside $\mathcal{S}$ from it's nearest facet in $\partial\mathcal{S}$. In the following parts we address these two problems.



(a) Schematic of algorithm 2

(b) Classifying points with respect to a meshed domain and calculating the distances

Figure 2.4: Comparing theoretically and experimentally obtained workspace. In figure 2.4b, the points inside and outside the workspace are shown in yellow and black respectively. The red lines indicate the distance of the farthest points from their nearest facets.

2.3.3.1 Classifying a point with respect to a triangulated domain

Classification a point with respect to a triangulated domain is a very well known problem and many methods exist, which offer various advantages in terms of computational complexities. However, most of these algorithms work only in the triangulated domain in question is a convex one. To the best of our knowledge, we are not aware of a complete method which classifies a point with respect to a domain of a general shape. Therefore, in this section we will discuss a method of classifying a point with respect to a convex domain obtained as the alpha shape of the points representing the workspace. In case the alpha shape is not convex, we will consider the convex hull of the object. Since the method will be used to test our algorithm with a large number experimentally obtained points, the approximation will wrongly classify the points which are quite near to the surface, thus will not change the general conclusion which we will derive from the experiment. In the algorithm 2, we propose a method, to classify a given set of points O with respect to $\mathcal{S}$ as inside (O_{in}) and outside (O_{out}). Our algorithm and it's Matlab implementation[1] is not the most efficient at higher dimensions but seems to perform reasonably well for 3D, by taking less than 8 seconds to classify about 10,000 points with respect to a meshed domain. Looking closely at figure 2.4a, we notice that due to our convex hull approximation of the green non-convex domain (not plotted in the figure) the points which are very near to the surface have been wrongly classified.

---

[1] We have used a Matlab function inhull.m, a fully vectorized implementation of the algorithm, made available by John D'Errico for free usage.

```
Purpose : To classify a set of points $O$ with respect to $\partial\mathcal{S}$
Input: The set $O$, $O \in \Re^3$ and $O \equiv O_{in} \cup O_{out}$
Output: $O_{in}$, $O_{out}$
Obtain a convex shape $\mathcal{S}$ from the set of points $\mathbb{S}$
Obtain $C$, the center of $\mathcal{S}$ by taking the mean of the vertices of the $N$ facets constituting
  $\partial\mathcal{S}$
for $i = 1, 2, ..., N$ do
    Calculate the normal to the $i^{th}$ facet, $A_n^i = (f_i^1 - f_i^2) \times (f_i^2 - f_i^3)$
    Choose a point $A^i$ on the $i^{th}$ facet and move $A_n^i$ to $A^i$
    Ensure that $A_n^i$ is directed inside, towards $C$
    $H(i) = \langle \overrightarrow{O_i - A_i}, A_n^i \rangle$
end for
if $H(i) \geq 0$ then
    Classify $O_1 \in O_{in}$ as inside, otherwise classify $O_1 \in O_{out}$
end if
```

Algorithm 2: Algorithm for classifying points as inside (O_{in}) or outside (O_{out}) of a triangulated domain

2.3.3.2 Minimum distance of a point from a triangulated domain

The problem of finding the minimum distance of a point from a triangulated domain is well studied problem. The most notable work being the famous Gilbert-Johnson-Keerthi (GJK) distance algorithm [65] and subsequent improvements on it by Ong et al. [66]. For another approach to the same problem, one may refer to the recursive algorithm proposed by Golubitsky et al. [64], which can tackle the problem in k-dimensions involving n *simplexes* with at most $\mathcal{O}(n^4)$ complexity. In our work, since we are interested in calculating the distance of a small number of points form the triangulated domain, following Ju et al. [67] we will express and solve the problem as a quadratic program as shown in equation (2.18) with constraints as given in equation (2.19).

$$\min_{\mathbf{x}} \frac{1}{2}\mathbf{x}\mathbf{H}\mathbf{x}^T \tag{2.18}$$

$$\text{Subject to } \mathbf{A}\mathbf{x} \leq \mathbf{b} \tag{2.19}$$

Where in equation (2.18), $H = \begin{bmatrix} I_3 & \phi_3 \\ \phi_3^T & 0 \end{bmatrix}$, $A = \begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & z_n & 1 \\ -x_r & -y_r & -z_r & \text{-1} \end{bmatrix}$,

$b = [1, 1, ..., 1]^T_{\{n+1 \times 1\}}$, I_3 is the 3×3 identity matrix, ϕ_3 is a 3×1 null vector, $\{x_i, y_i, z_i\}$ are the coordinates of the i^{th} vertex of $\partial\mathcal{S}$ and we want calculate the nearest distance of the point $\{x_r, y_r, z_r\}$ known to be outside of $\mathcal{S}$. The distance of the point from $\partial\mathcal{S}$ is given by $\dfrac{2}{||\mathbf{x}||_2}$.

2.4 Fitting super-ellipsoids to cloud of points in $\Re^3$

Super ellipsoids are an extension of ellipsoids and have been used for solid modeling in diverse fields such as for modeling and representation of medical data ([68]), high fidelity representation of scanned objects ([69]) and as objects for developing obstacle avoidance algorithms ([70]). The main reason behind it's widespread use is the availability of precise closed form expressions for *almost* all geometrical properties like moments of inertia about arbitrary axes, mass and surface properties like tangents and normals (see [71] for a more complete description). In this work, we use an efficient method based on the works by [69] and further improvements on it by [68] to fit super-ellipsoids to a cloud of points in $\Re^3$. The goal is to obtain a more accurate approximation of the geometry of human fingertips over normal ellipsoids used by previous researchers (see e.g. [72] and similar works for more details).

Equations (2.20) to (2.22) are the parametric expressions that map an arbitrary surface with parameters A, B and C having dimensions of length and dimensionless exponents e and n from $\{u, v\} \in \Re^2 \rightarrow \{x, y, z\} \in \Re^3$. The sign function is abbreviated as "sgn" in equations (2.20) to (2.22).

$$x(u, v) = A\,\mathrm{sgn}\left(\cos(v)\right)\left(|\cos(v)|\right)^n \mathrm{sgn}\left(\cos(u)\right)\left(|\cos(u)|\right)^e \tag{2.20}$$

$$y(u, v) = B\,\mathrm{sgn}\left(\cos(v)\right)\left(|\cos(v)|\right)^n \mathrm{sgn}\left(\sin(u)\right)\left(|\sin(u)|\right)^e \tag{2.21}$$

$$z(u, v) = C\,\mathrm{sgn}\left(\cos(v)\right)\left(|\cos(v)|\right)^n \tag{2.22}$$

The expression in equation (2.23) classifies whether a point is in, on or outside the surface expressed by equations (2.20) to (2.22) by assuming negative, zero and a positive values respectively.

$$\mathcal{F}(x, y, z) = \left(\left(\left(\frac{x}{A}\right)^{2\,e^{-1}} + \left(\frac{y}{B}\right)^{2\,e^{-1}}\right)^{\frac{e}{n}} + \left(\frac{z}{C}\right)^{2\,n^{-1}}\right)^{\frac{n}{2}} \tag{2.23}$$

For fitting a superquadric to a cloud of n points contained in the matrix $\mathbf{P}$, $\mathbf{P} \in \Re^{n \times 3}$ we formulate the following optimization problem given in equation (4.1). In equation (4.1), the elements of $\bar{\mathbf{x}}$ are the coordinates for the center of the super ellipsoid, the elements $\vec{\Theta}$ are the Euler angles specifying the orientation of the fitted ellipsoid, $\mathbf{a} = \{A, B, C\}$ and $\mathbf{p} = \{e, n\}$.

$$\underset{\bar{\mathbf{x}},\vec{\Theta},\mathbf{a},\mathbf{p}}{\text{Min}} \sum_{i=1}^{N} \sqrt{ABC}(\mathcal{F}(\mathbf{X}^r, \vec{\Theta}, \mathbf{a}, \mathbf{p}) - 1)^2 \tag{2.24}$$

In equation (4.1), the *in-out* function F from equation (2.23) is used with the cloud of points $\mathbf{P}$ transformed to the original coordinate about which equations (2.20) to (2.22) are defined. An initial guess about the position and orientation of the cloud of points may be given as follows. The value of $\bar{\mathbf{x}}$ in equation (2.25) may be given as the average of the respective coordinates of the points in $\mathbf{P}$, i.e., $\bar{\mathbf{x}} = \left\{ \frac{1}{n}\sum_{i=1}^{n} x_i, \frac{1}{n}\sum_{i=1}^{n} y_i, \frac{1}{n}\sum_{i=1}^{n} z_i \right\}$.

$$\left\{ \begin{matrix} \mathbf{X}^r \\ 1 \end{matrix} \right\} =_O^P [T] \left\{ \begin{matrix} x_i \\ y_i \\ z_i \\ 1 \end{matrix} \right\} = \begin{bmatrix} [R]^* & \bar{\mathbf{x}} \\ 0 & 1 \end{bmatrix} \left\{ \begin{matrix} x_i \\ y_i \\ z_i \\ 1 \end{matrix} \right\} \tag{2.25}$$

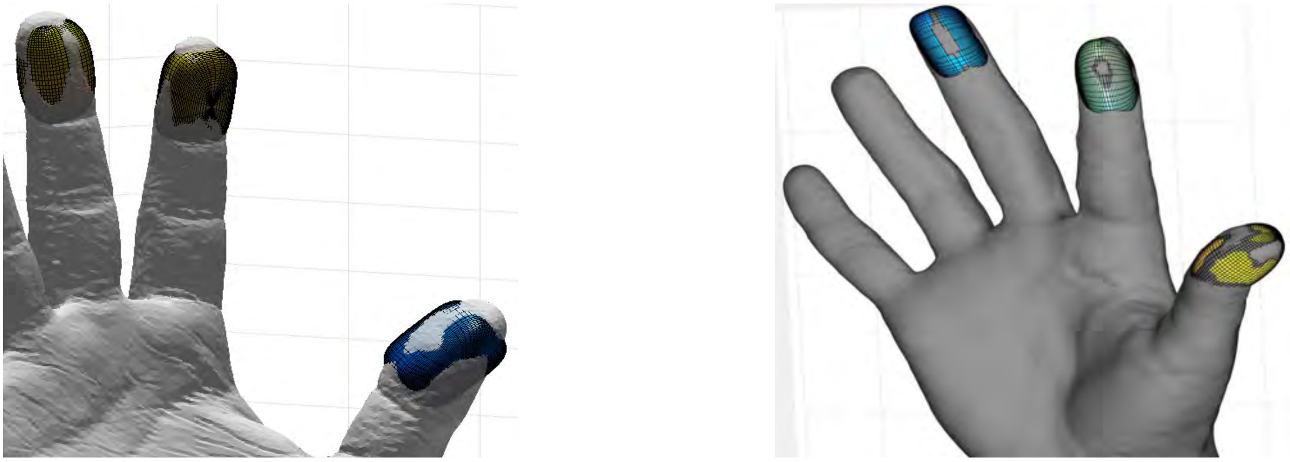
In equation (2.25), $[R]^*$ may be represented as the matrix which diagonalizes the total moment matrix M of the cloud of points $\mathbf{P}$ (see equation (2.26)) to M_0 in the operation $M_0 = R^{*-1}MR^* = R^{*T}MR^*$. In equation (2.26) $\tilde{x}_i = (x_i - \bar{x})$, $\tilde{y}_i = (y_i - \bar{y})$ and $\tilde{z}_i = (z_i - \bar{z})$.

$$M = \frac{1}{n}\sum_{i=1}^{n} \begin{bmatrix} \tilde{y}_i^2 + \tilde{z}_i^2 & -\tilde{x}_i\tilde{y}_i & -\tilde{z}_i\tilde{x}_i \\ -\tilde{x}_i\tilde{y}_i & \tilde{x}_i^2 + \tilde{z}_i^2 & -\tilde{y}_i\tilde{z}_i \\ -\tilde{z}_i\tilde{x}_i & -\tilde{y}_i\tilde{z}_i & \tilde{x}_i^2 + \tilde{y}_i^2 \end{bmatrix} \tag{2.26}$$

The goodness of the fit, according to [73], may be measured by total value of the first order approximation of the distances of the points in $\mathbf{P}$ from the fitted surface $\mathcal{S}$ as given in equation (2.27).

$$d_S = \sum_{i=1}^{n} \left(\frac{|\mathcal{F}(x_i, y_i, z_i, \mathbf{a}, \mathbf{p}) - 1|}{||\nabla\mathcal{F}||_{\{x_i,y_i,z_i\}}} \right)^2 \tag{2.27}$$

For the hand model shown in figure 3.4a, from 296,358 vertexes and 98,786 triangular facets available in the raw data (see figure 3.4a) we pick 40 points from each of the middle finger, index finger and the thumb tips. The unconstrained optimization problem given in equation (2.24) is quadratic in nature and can be solved using the *Levenberg-Marquardt* method (see [74] for more details). The numerical results for the fit are given in tables 2.1 and 2.2. The plotted results



(a) Superellipsoid approximation of a male subject's hand

(b) Superellipsoid approximation of a female subject's hand

Figure 2.5: Superellipsoid approximations of human finger-tips. The approximations for the female subject are better due to the higher resolution of the available 3D scan.

of the same are shown in figure 2.5. It may be mentioned that the underlying computations would be considerably simplified if we choose ellipsoids to approximate the fingertips, but we have avoided that for the following reasons:

- Normal ellipsoids taper off more rapidly and therefore a fitted ellipsoid cannot properly emulate the curvature of the human finger at the tip. The ellipsoid approximation for the fingertips is too pointed at the tips.
- For the thumb, which has a more flatter shape, the ellipsoid approximation correctly capturing this feature is too big and does not taper off evenly at the thumb tip.
- Also, the goodness metric values (see equation (2.27)) are worse in case of fitted ellipsoids.

Table 2.1: Parameters of the super-ellipsoids fitted to the finger-tips in figure 2.5a.

Finger	$\mathcal{S} = \{A, B, C, e, n\}$	$d_{\mathcal{S}}$
Index	{14.98, 9.83, 8.33, 1.14, 0.67}	36.23
Middle	{14.99, 7.94, 9.80, 0.72, 0.5}	52.65
Thumb	{11.78, 9.12, 15.0, 0.58, 0.62}	42.8

2.5 Collision detection of two super-ellipsoids in $\Re^3$

Collision detection and interaction between moving rigid bodies is a generic problem frequently encountered in various fields like robotics (see e.g. [75]), computer aided design(CAD), com-

Table 2.2: Parameters of the super-ellipsoids fitted to the finger-tips in figure 2.5b.

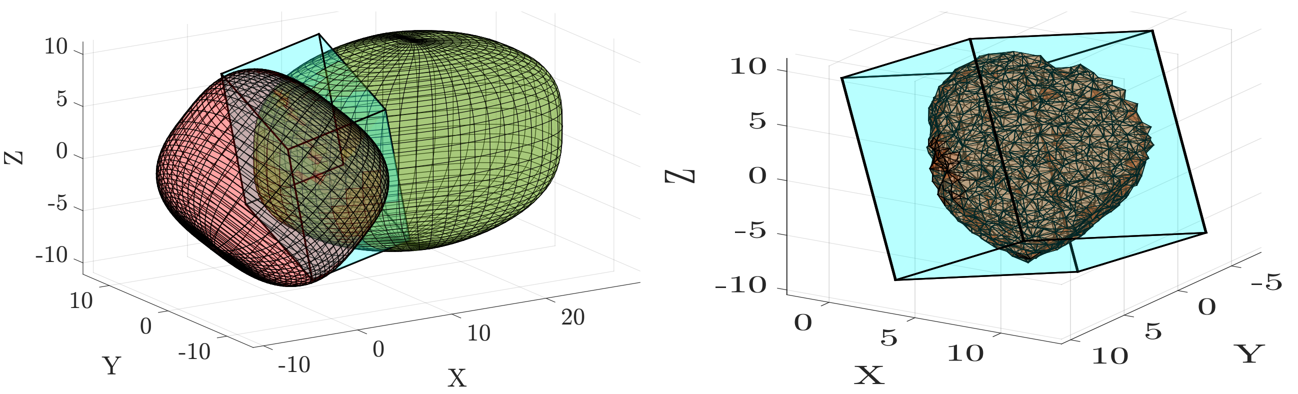
Finger	$\mathcal{S} = \{A, B, C, e, n\}$	d_S
Index	{6.03, 14.8, 7.87, 0.97, 0.80}	24.23
Middle	{6.15, 13.10, 7.97, 0.88, 0.57}	25.65
Thumb	{9.28, 8.10, 15.2, 0.78, 0.93}	16.8

puter vision (see e.g. [76]) and discrete element method (DEM) simulations (see e.g. [77]). In this work, we use a Monte Carlo based *probabilistic* algorithm to obtain the *intersection volume* of two arbitrarily oriented super-ellipsoids in the 3D space. In the previous section we have discussed modeling human fingers by using super ellipsoids, which we have used later in (section 3.3) to obtain the workspace of the actual human hand. In the human hand, the fingers have finite size and to obtain realistic workspace of the multi-fingered hand, it is essential to model the collision and intersection between the human fingers modeled as super-ellipsoids, as this reduces the total workspace. In the following algorithm 3, we describe a Monte Carlo based algorithm to obtain an estimate of the intersection volume between two arbitrarily places ellipsoids in space. The intersection volume $\mathcal{I}(\mathcal{S}_1, \mathcal{S}_2)$ of the two closed solids $\mathcal{S}_1$ and $\mathcal{S}_2$ can be obtained using the algorithm 3 given below.

In algorithm 3, $B(i,:)$ and $B(:,i)$ mean the i^{th} row and column of the matrix B respectively, S_3 is the 3×3 diagonal matrix with the diagonal elements from the vector $\vec{s}$ in order, and $\mathcal{B}^x$ is the range of the box $\mathcal{B}$ in X Cartesian direction. The above algorithm was used on super-ellipsoid models of index finger and the thumb as given in table 2.1. To obtain the intersection volume numerically using the algorithm given above, we need to set a cut-off volume above which the thumb and index finger is considered to be collision. This is assumed to correspond to a penetration depth of 0.9 mm – references [78] and [79] report that for human finger, a normal force of 0.5N is obtained at the finger tips for a penetration depth of approximately 0.9mm. It is assumed that this force does not hinder the assembly of the manipulator at the certain given configuration and therefore the solution of the inverse kinematics problem exists at the said configuration. Considering this, the approximate *cutoff* volume is set as 15 mm^3. Figures 2.6a and 2.6c shows the intersection of two super-ellipsoids in space. The blue box in figures 2.6b and 2.6d contains the intersection volume and is aligned along the principal components of the cloud of points $\mathcal{M}_n$ as obtained in algorithm 3. The intersection volume is detected as 1041.5 mm^3 for the first case (in figure 2.6a), thus the chosen orientation of the super-ellipsoids is impossible. However, for the second case (figure 2.6c) the intersection volume is obtained as 12.4 mm^3, thus indicating the chosen orientation of the super-ellipsoids is possible.

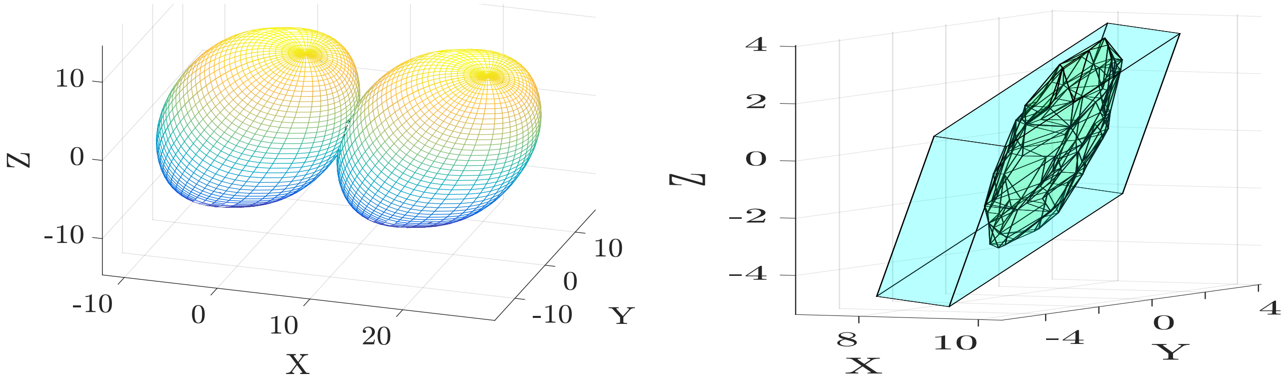
```
Purpose : To obtain an estimate of $\mathcal{I}(\mathcal{S}_1, \mathcal{S}_2)$
Input: $\mathcal{S}_1$, $\mathcal{S}_2$, $[T]^1_O$ and $[T]^2_O$
Output: $\mathcal{I}\widehat{(\mathcal{S}_1, \mathcal{S}_2)}$
  Using inputs, obtain a box $\mathcal{B}_1$ ($\mathcal{B}_1 \in \Re^3$) enveloping $\mathcal{S}_1 \cup \mathcal{S}_2$ in $\Re^3$;
  Populate $\mathcal{H}_1$, ($\mathcal{H}_1 \in \mathcal{B}_1$), an array of $N_n$ uniformly distributed random triplets in $\Re^3$
    (given by $f_U(x_i),\ i = 1, 2, 3$);

  Use naive MC search to check if $\mathcal{H}_1(j, :) \in \mathcal{S}_1 \cap \mathcal{S}_2$ by using equation (2.23);
  Save successful trials, $\mathcal{M}_n(i, :) \longleftarrow \mathcal{H}_1(j, :)$ ;

  Obtain box $\mathcal{B}_2$ from the ranges of columns of $\mathcal{M}_n$ ;
  $\mu \longleftarrow$ center of $\mathcal{B}_2$
  $X_b \longleftarrow {\mathcal{B}_2}^x$, $Y_b \longleftarrow {\mathcal{B}_2}^y$, $Z_b \longleftarrow {\mathcal{B}_2}^z$ ;
  $\vec{s} = 0.25[X_b,\ Y_b,\ Z_b]$;
  $\Sigma \longleftarrow S_3$

  Populate $\mathcal{H}_2$, an array of $N_v$ normally distributed random triplets in $\Re^3$ using $\mu$ and $\Sigma$
    from a multi-variate normal PDF (given by $f_N(x_i),\ i = 1, 2, 3$);

  Use importance sampling MC to check $\mathcal{H}_2(j, :) \in \mathcal{S}_1 \cap \mathcal{S}_2$ also, $\mathcal{F}(\mathcal{H}_2(j, :)) \to 1$;
  Save successful trials, $\mathcal{M}_{vr}(i, :) \longleftarrow \mathcal{H}_2(j, :)$ for representation;
  Obtain an estimate $\mathcal{I}\widehat{(\mathcal{S}_1, \mathcal{S}_2)}$ of the intersection volume as
    $\langle \mathcal{I} \rangle = \frac{1}{N_v} \sum_{i=1}^{N_v} \mathcal{F}(\mathcal{H}_2(i, :)) \frac{f_U(\mathcal{H}_2(i, :))}{f_N(\mathcal{H}_2(i, :))}$ using equation (2.11) ;
```

Algorithm 3: Pseudo-code for evaluating intersection volume of two arbitrarily oriented super-ellipsoids in $\Re^3$



(a) Impossible intersection of 2 super-ellipsoids

(b) Representation of the intersection volume of figure 2.6a



(c) Possible intersection of 2 super-ellipsoids

(d) Representation of the intersection volume of figure 2.6c

Figure 2.6: Representation of 2 cases of interaction between super ellipsoids and their intersection volumes

2.6 Conclusion

In this chapter we have described the key mathematical techniques used in the current work. We started by discussing the Monte Carlo method– a technique central to the research documented in this thesis. Following which, we discussed about the tools and techniques we have used to represent the workspace of a manipulator. Following which in the final two sections, we have described methods to use superquadrics to model human fingertips and a method to detect collision of superquadrics in space without calculating their distance of closest approach.

Chapter 3

Manipulator workspace using Monte Carlo method

3.1 Introduction

In this chapter we discuss the application of the Monte Carlo method to obtain the well conditioned workspace of a manipulator. We start by discussing the definition of the well conditioned workspace of a manipulator, and how we have used the Monte Carlo method to evaluate it and the advantages and limitations of the method in section 3.2. Following which we have provided two demonstrative examples. In the first example in section 3.3 we discuss the use of Monte Carlo method to obtain the well-conditioned workspace of the human hand by approximating the human 3 fingered grasp by a 6 degrees of freedom parallel manipulator. Apart from providing two important results on the human hand workspace, we also demonstrate the accuracy of the formulation by performing experiments with human subjects manipulating rigid objects. In second example in section 3.4 we obtain the workspace of the Stanford-JPL hand (Salisbury hand) using the Monte Carlo method. We end the chapter by summarizing the main results in section 3.5.

3.2 Well conditioned workspace of a manipulator using Monte Carlo method

The workspace of the end-effector of a serial or a parallel 6 degree-of-freedom manipulator involves three quantities representing translation and three quantities representing orientation of the end-effector. It is symbolically expressed (after Selig [80]) as

$$\mathcal{W}_T = \mathcal{W}_p \ltimes \mathcal{W}_o, \quad \mathcal{W}_p \in \Re^3, \; \mathcal{W}_o \in SO(3) \; \& \; \mathcal{W}_T \in SE(3) \tag{3.1}$$

It is difficult to visualize or define a volume of the total workspace and it is common in literature to seek quantities like *constant orientation workspace*, $\mathcal{W}_p$, or *constant position workspace*, $\mathcal{W}_o$, by independently seeking either of $\mathcal{W}_o$ or $\mathcal{W}_p$ ([20], [21]). Here, we will loosely follow the work by Stamper et al. [33] and define well condition workspace as the closed volume in $\Re^3$ which is a subset of W_p in equation (3.1) where each point inside $\mathcal{W}_p$ can be reached by *at least* one known (but randomly generated) orientation of the end-effector, satisfying all joint limit constraints and *sufficiently* well-conditioned.

The well-conditioning of the workspace is related to the condition number of the manipulator Jacobian matrix. For a serial manipulator, the notion of a manipulator Jacobian is very well known and it relates the linear and angular velocity of the end-effector with the joint rates. For a parallel manipulator, with actuated and passive joints, an equivalent manipulator Jacobian can be defined in terms of the actuated joints by incorporating certain terms which ensure that the kinematic constraints associated with the geometry of the manipulator are not violated during the motion of the manipulator. Simply put, the equivalent Jacobian of the manipulator will have the form:

$$V_{\text{platform}} = \mathbf{J}^V_{eqv} \dot{\vec{\theta}} \tag{3.2}$$

$$\omega_{\text{platform}} = \mathbf{J}^\omega_{eqv} \dot{\vec{\theta}} \tag{3.3}$$

Where V_{platform} and ω_{platform} are the linear and angular velocities of the end effector, $\dot{\vec{\theta}}$ are the actuated joint rates and the equivalent Jacobians are denoted by $\mathbf{J}^V_{eqv}$and $\mathbf{J}^\omega_{eqv}$. Following Ghosal [81] and Ghosal and Ravani [82], the loop-closure or constraint equations can be used to relate the passive and the actuated joint angles. When the closed-loop mechanisms or the parallel manipulator is not at a *gain* singularity, the passive joint rates can be solved in terms of the actuated joint rates and relations between the linear and angular velocity of a *chosen* end-effector and the actuated joint rates can be obtained (see also section 4.4.3 and section 4.4.4).

The condition number of a matrix $\mathbf{A}$ is defined as

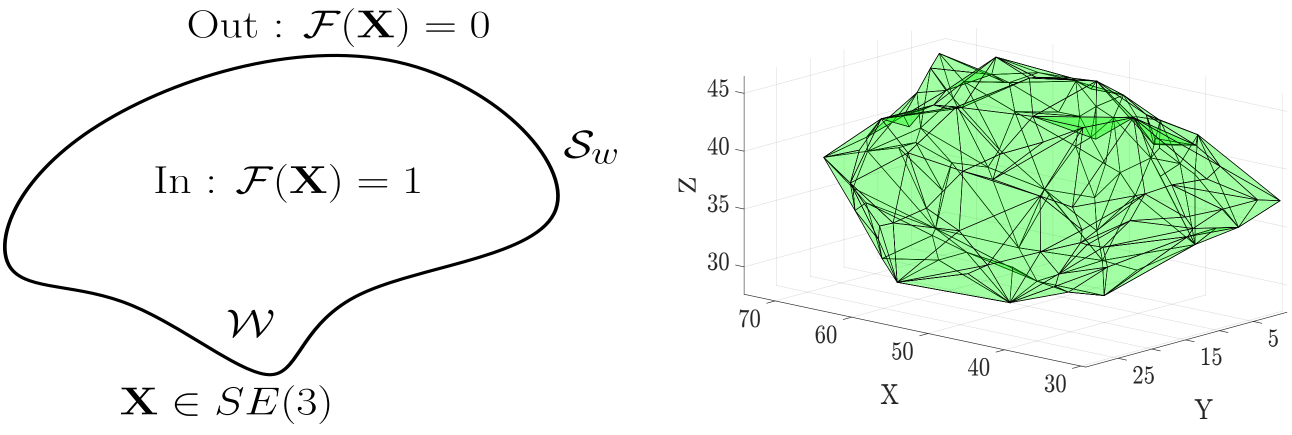
$$\kappa = \parallel \mathbf{A} \parallel \parallel \mathbf{A}^{-1} \parallel \tag{3.4}$$

where $\parallel \cdot \parallel$ is the L^2 norm of a matrix.

We use the definition in equation (3.4) in equations (3.2) and (3.3) to define a condition number for the equivalent Jacobians and denote them by κ_V and κ_ω. To ensure *well-conditionedness* of the manipulator throughout it's entire range of motions, we arbitrarily assign an upper bound on the two condition numbers relating to the translational and rotational motions of the end-effector. Denoting the chosen upper bound by κ^*, we can write

$$\max\{\kappa_V, \kappa_\omega\} \leq \kappa^* \tag{3.5}$$

In the numerical examples, we have used specified values of κ^* which give a conservative estimate of the well-conditioned workspace. A larger κ^* will give a larger well-conditioned workspace (see also figure 4.9 for the increase in workspace with increasing κ^*). For using the Monte Carlo method to evaluate the workspace volume/area of a manipulator, we observe that the Monte Carlo based method proposed in algorithm 4 classifies a given set of n-tuples (line # 5-7) into two sets depending on whether or not they satisfy a given function. In line with this observation, we assume that the well-conditioned workspace $\mathcal{W}$, ($\mathcal{W} \in SE(3)$), of a parallel manipulator is a collection of a finite number (say n) of closed sets in $SE(3)$ bounded by surfaces $\mathcal{S}_w^i$, $\forall i = 1, 2..., n$. We formulate an *in-out* function $\mathcal{F}$ for $\mathcal{S}_w^i$s which takes input of the position and orientation of the end-effector of the manipulator. This function can be represented as



(a) Diagramatic representation of the well conditioned workspace

(b) Actual well conditioned workspace in $\Re^3$

Figure 3.1: Schematic and actual representation of the workspace of a manipulator

$$\mathcal{F}(\mathbf{X}) = \left\{ \begin{array}{ll} 1 \text{ if } & \mathbf{X} \in \mathcal{W} \\ 0 \text{ if } & \mathbf{X} \notin \mathcal{W} \end{array} \right\} \tag{3.6}$$

The inclusion (or exclusion) of a given position and orientation of the manipulator, $\mathbf{X} \in SE(3)$, is determined by the fact that for a given $\mathbf{X}$ the inverse kinematics problem has real solutions, the active and passive joint values in the parallel manipulator are within prescribed limits and the manipulator Jacobian is well conditioned. Using algorithm 4 described below, we can obtain $\widehat{\mathcal{W}}$, an estimate of the well conditioned workspace $\mathcal{W}$ of the chosen parallel manipulator. In algorithm 4 below, V_s is the selected search space in the Cartesian space ($\Re^3$) and V_a is the search space is the space of orientation ($SO(3)$). The total workspace of the manipulator is a subset of $V_s \ltimes V_a$ (see [80]).

```
Purpose : To obtain an estimate of the well conditioned workspace
Input: $\mathcal{F}$, $\mathcal{V}_s \in \Re^3$ and $\mathcal{V}_a \in SO(3)$
Output: $\widehat{\mathcal{W}}$
 1: $S = 0$, $\Phi_k = 0 \; \forall k$, $k = 0$
 2: Choose sample size $\mathcal{N}_{total}$
 3: for $k \in \{1, 2, ...\mathcal{N}_{total}\}$ do
 4:     Obtain a $\mathbf{X}_k \in SE(3)$ form a uniformly distributed random PDF in $\mathcal{V}_s$ and $\mathcal{V}_a$.
 5:     Evaluate $\Phi_k = \mathcal{F}(\mathbf{X}_k)$;
 6:     if $\Phi_k = 1$ then
 7:         $S = S + \Phi_k$
 8:     end if
 9:     $k = k + 1$
10: end for
11: Compute $\widehat{\mathcal{W}} = \frac{S}{\mathcal{N}_{total}} \mathcal{V}_s$
```

Algorithm 4: Algorithm for evaluating the well conditioned workspace of a manipulator.

We present some of the salient features of algorithm 4 which leads to efficient and realistic evaluation of the workspace of a parallel manipulator.

- The `for` loop in lines #3−#10 of algorithm 4 can be executed in parallel. This possibility of parallelization is in fact one of the biggest advantages of the Monte Carlo method and contributes to the exceptional computational performance of the algorithm.

- By using information about the underlying process and choosing a proper distribution $\mathcal{F}^*(\mathbf{X})$, a significant improvement can be made in calculating the estimate of the function over a naive Monte Carlo simulation [53].

- It is well known that typically the inverse kinematics problem of a parallel manipulator is much simpler to solve as compared to the forward kinematics, and we use the inverse kinematics in the formulation. Once the inverse kinematics is solved, we check that the active and passive joint values are within the prescribed limits.

- The well-conditioning at a given $\mathbf{X}$ is based on the condition number of the Jacobian matrices associated with linear and angular velocity of the end-effector. We have used a conservative 1000 as the upper limit of the condition numbers in all our simulations. However, any other upper limit can also be used.

The Monte Carlo simulation as discussed in this section classifies a set of position and orientation of the end-effector of a manipulator according to their occurrence in the well conditioned workspace, satisfaction of the inverse kinematics and the prescribed joint limits. The output is a set of points in $\Re^3$ and $SO(3)$. For better visualization of the workspace, we use standard Delaunay triangulation algorithms [60] and coloring the triangulated domain of the boundary points.

3.2.1 Comparison between Monte Carlo, Chebyshev sampling and analytical methods to find the workspace area/volume

In this section, we use the example of a planar 5R closed-loop mechanism and obtain the workspace of this mechanism by three approaches and compare the results obtained. A schematic description of the 5R mechanism is given in figure 3.2.

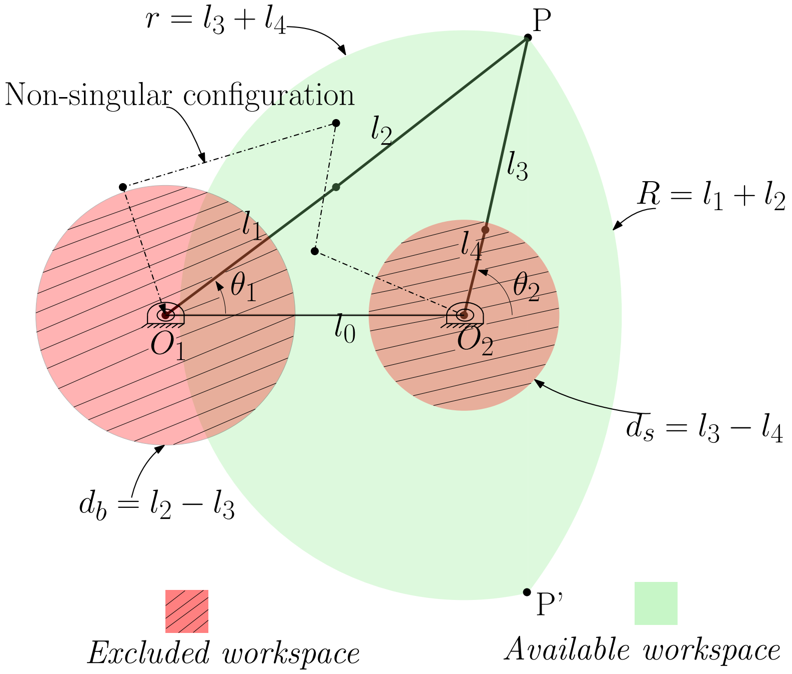


Figure 3.2: General schematic diagram of a 5R manipulator

The 5R manipulator was chosen for the example because it is fairly well documented in literature and the choice of the output point P in figure 3.2 is natural. In figure 3.2, the plain bounded area represents the reachable workspace of P with two hatched exclusion zones. The joints O_1 and O_2 are actuated. The boundary curves of the well-conditioned workspace for the mechanism can be analytically obtained and has been described in detail in literature (see, for example, [83],[84]). In figure 3.2 the larger radius arc bounding the workspace is given by R, the radius of the smaller bounding arc is given by r and the radii of the smaller and larger exclusion zones are given by d_s and d_b. The analytical expressions for R, r, d_s and d_b in figure 3.2 after [84] are

$$\begin{aligned} R &= l_1 + l_2, \quad & r &= l_3 + l_4 \\ d_s &= l_3 - l_4, \quad & d_b &= l_2 - l_1 \end{aligned}$$

We denote the area enclosed by the arcs by $A(\mathbf{L})$, where $\{\mathbf{L}\} = [d, l_1, l_2, l_3, l_4]$ is the vector of design parameters – link lengths in our case. The area A marked as a plain bounded area is given by $A = A_1 - A_{O_2} - A_2$ where the subtracted quantities are the exclusion zones marked as hatched bounded zones. The closed-form expressions of $A(\mathbf{L})$ is given as

$$\mathcal{W}(\mathbf{L}) = r^2 \cos^{-1}\left(\frac{1}{2}\,\frac{-R^2+d^2+r^2}{dr}\right) + R^2 \cos^{-1}\left(\frac{1}{2}\,\frac{R^2+d^2-r^2}{dR}\right) - \tag{3.7}$$
$$\tfrac{1}{2}\sqrt{(-d+R+r)(d+r-R)(d-r+R)(d+R+r)} - \pi\, {d_s}^2 - {d_b}^2 \cos^{-1}\left(\tfrac{1}{2}\,\tfrac{-R^2+d^2+{d_b}^2}{d d_b}\right) -$$
$$R^2 \cos^{-1}\left(\tfrac{1}{2}\,\tfrac{R^2+d^2-{d_b}^2}{dR}\right) + \tfrac{1}{2}\sqrt{(-d+R+d_b)(d+d_b-R)(d-d_b+R)(d+R+d_b)}$$

In the above general expression of the area, the reachable workspace can be partitioned into a few topologically different cases by considering the circular arcs that bound the reachable workspace. In this example, we confine the workspace to a zone between O_1 and O_2 in figure 3.2 which is bounded by 2 *continuous circular* curves. This significantly simplifies the expression of the area $A(\mathbf{L})$ as the first three terms can be removed. For $l_1 = 1$, $l_2 = 1$, $l_3 = 1$, $l_4 = 3$ and $d = 4$, by using equation (3.7), the area is obtained as 5.6123 unit^2. We compare this computed area by the probabilistic Monte Carlo method and deterministic Chebyshev sampling method. In the next section, we also present a discussion on the use of *interval analysis* (see e.g. works by Chablat et al. [25] and Caro et al. [28]) to obtain the area.

To compare this computed value with the area obtained using the other methods, we first populate points in $\Re^2$ with bounds on X, Y as $X \in [-6, 6]$, $Y \in [-6, 6]$. The points in $\Re^2$ are populated by a deterministic sampling and a random number generator. The formula for generating N Chebyshev's intervals is given by

$$x = \frac{1}{2}(b+a) + \frac{1}{2}(b-a)\cos(\frac{\pi(2j-1)}{2k}), x \in [a,b], j = 1,2,...,N \tag{3.8}$$

The computation time[1] and the differences in the obtained areas (in unit^2) from the different approaches are shown in tables 3.1 and 3.2. In the tables, BCE is the best case error in obtaining the area of the workspace (across 10 trials), CPU Time I is the time required to obtain the right hand side of line # 11 algorithm 4, and CPU Time II is the time required to obtain the LHS of equation (2.17). From tables 3.1 and 3.2, we can conclude that Monte Carlo method is faster and more accurate than deterministic sampling for obtaining the workspace area.

# Samples	Area using Alg. 4	CPU Time I	BCE %
5000	5.58	0.12	0.575
10000	5.64	0.15	0.494
100500	5.608	0.36	0.059

Table 3.1: Computational performance of a fully parallel MC method

Intervals	Value of K in eq. 3.8	Area using eq. 2.17	CPU Time II	Error %
317	1	5.438	0.504	3.104
400	1	5.451	0.623	2.87
500	1	5.452	1.55	2.84

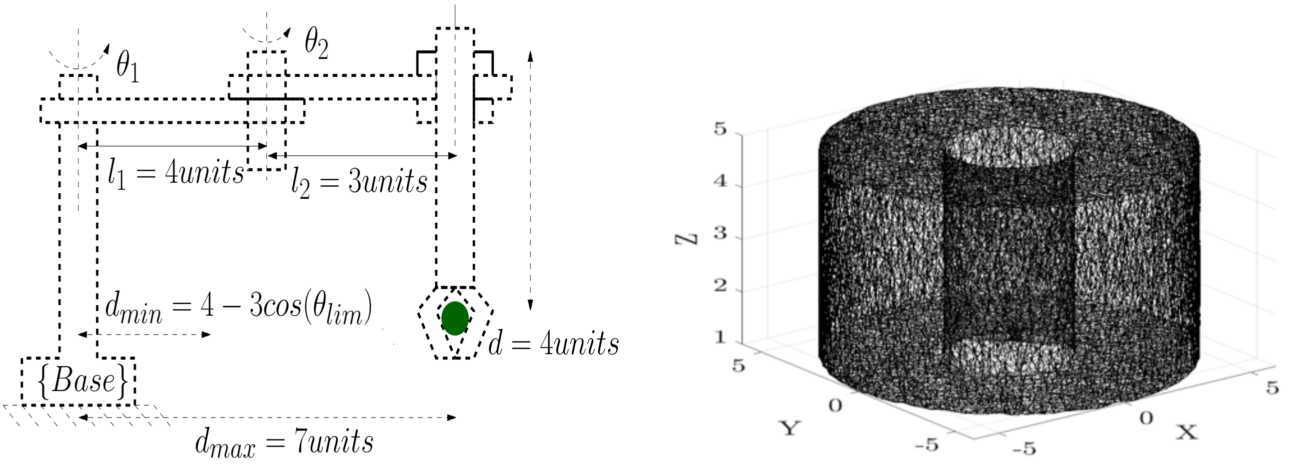
Table 3.2: Computational performance of Chebyshev interval samples

The efficiency of the Monte Carlo method is more evident if we consider a 3D spatial example such as a SCARA robot shown in figure 3.3a. We consider a constraint on the joint θ_2 of the form $[10° \leq \theta_2 \leq 2\pi - 10°]$ to prevent interference of the last link and the base. The reachable workspace of this manipulator is a hollow cylinder with $d_{\max} = 7$ units and $d_{\min} = 4 - 3\cos(\theta_{\lim})$ units and the workspace volume is given by

$$\mathcal{W} = \pi(d_{max}^2 - d_{min}^2)d = \pi(49 - (4 - 3\cos(10°))^2)4 = 602.01 \text{ unit}^3 \tag{3.9}$$

---

[1] The programs were run in Matlab R2015a [57] on a Windows 7 PC with an quad core (3.10GHz) Intel XEON & 16 GB of RAM.



(a) Schematic diagram of a SCARA manipulator

(b) Workspace of the SCARA robot shown in figure 3.3a

Figure 3.3: Schematic representation and workspace of a SCARA manipulator

The inverse kinematics equations of the SCARA manipulator is given as

$$\begin{aligned} d &= -Z \\ \theta_2 &= \cos^{-1}((X^2+Y^2-l_1^2-l_2^2)/(2l_1l_2)) \\ \theta_1 &= \operatorname{atan2}(Y,X)-\operatorname{atan2}(l_2\sin(\theta_2),(l_2+l_2\cos(\theta_2))) \end{aligned} \tag{3.10}$$

Next, we populate points in $\Re^3$ with bounds on X, Y and Z as $X \in [-7.5, 7.5]$, $Y \in [-7.5, 7.5]$ and, $Z \in [-0.5, 5.5]$ Similar to the 5R closed-loop mechanism, the points in $\Re^3$ are populated by a deterministic sampling and a random number generator. The computation time and the differences in the obtained volumes (in unit^3) from the different approaches are shown in tables 3.3 and 3.4. In table 3.3 CPU Time I is the time required to obtain the volume using algorithm 4, and CPU Time II is the time required to obtain the right-hand side of equation (2.17).

# Samples	Vol. using Alg. 4	CPU Time I	Vol. using eq. 2.17	CPU Time II	BCE %
10E5	597.08	0.84	576.88	2.92	0.8165
10E6	598.3	3.25	586.753	28.51	0.6146
3.375× 10E6	599.3	10.42	589.92	118.75	0.4615

Table 3.3: Computational performance of a fully parallel MC method

It can be seen that Chebyshev sampling method searches through $150^3 = 3375000$ points for all the cases and the best case error is 1.76 %, where as the best case Monte Carlo (MC)

Intervals	Value of k in eq. 3.8	Volume using eq. 2.17	CPU Time II	Error %
150	45	585.8236 unit^3	62.73	2.68
150	50	591.3996 unit^3	75.37	1.76
150	55	586.5533 unit^3	92.68	2.56

Table 3.4: Computational performance of Chebyshev interval samples

searching through 3375000 points saves 86.17 % computation time and is 99.54 % accurate. For equal number of samples, the MC sampling is better than the Chebyshev samples in finding the workspace volume and is more than 7 times faster. If we settle for a less stringent error bound, say $\sim$ 0.8%, then the Monte Carlo simulation is even more accurate than the best case Chebyshev samples and is almost 90 times faster. This huge computational advantage is largely attributed to the possibility of very high parallelization of the Monte Carlo method as compared to difficulty in parallelization of deterministic search methods.

3.2.2 Comparison between Monte Carlo and interval analysis based methods

Caro et al.[28] have proposed a numerical constraint programming based method to generate an approximation of the singularity free workspace of a parallel manipulator. The interval analysis (IA) technique together with branch and prune (BPA) algorithm is used to efficiently search through the intervals (and sub-intervals) populated during the solution of the problem. The work uses extensive symbolic computation and the `Intlab` library (see Rump [85]) to pose and solve the problem, respectively. The main differences and advantages of our approach is that our approach only requires the explicit solution of the inverse kinematics problem for the manipulator. The approach by Caro et al. requires obtaining explicit analytical forms of the various constraints for the manipulator and due to the resulting computational complexity, only planar manipulators with 2 or 3 DoFs and simple geometry could be studied. We believe attempting the same for a 6 DoF manipulator will involve a prohibitive amount of algebraic manipulations. Additionally the approach by Caro et al. takes significant amount of computation time as shown in a numerical experiment for obtaining the *singularity free* workspace of a 5R planar parallel manipulator. They report that the computation time[1] for obtaining the workspace by searching through $69,612$ boxes with *precision* of 0.1 is 38 seconds. In section 3.2.1 we obtain a computation time of 0.36 seconds for searching through 100,500 points in a search space and

---

[1] The authors report using a PC with a 3.4 GHz Intel XEON processor and 16GB RAM, which is comparable to the hardware we have used.

the best case error was 0.059%.

Merlet and co-workers [25], [44], [26] and [19] have proposed the interval analysis approach to pose and solve multi-objective optimization problems related to the workspace of parallel manipulators. In a work by Hao and Merlet [26], the solution of the inverse kinematics problem for the manipulator is used and the well-conditioning (or singularities) is checked (obtained) from the inverse of the manipulator Jacobian ($\mathbf{J}^{-T}$). The main differences and advantages of our work as compared with Hao and Merlet[26] are:

- The use of inverse of $\mathbf{J}^T$ only avoids gain type singularities in the task space. For a fully in-parallel Stewart platform with six actuated prismatic joints, this is enough. However a hybrid parallel manipulator can also loose one or more degrees-of-freedom and to overcome this problem, in our formulation, we have used upper bounds on the condition numbers of the manipulator Jacobians (see equation (3.5)).

- A related work (see Chablat et al. [25]) suggests *box validation*, which involves the calculation of the eigenvalues of $\mathbf{J}^{-T}$ for subdividing boxes (analogous to intervals in 3D). For a general 6 DoF parallel manipulator, the eigenvalue problem for $\mathbf{J}^{-T}$ is almost impossible to solve symbolically and for some parallel manipulator, with non-square $\mathbf{J}$, the eigenvalue problem may not exist in exact form. In contrast, we have explicit symbolic expressions for all the equations and quantities used (including equation (3.5)), and hence the computational load is much less.

From sections 3.2.1 and 3.2.2 we can conclude that the Monte Carlo method performs more efficiently than existing methods for obtaining the workspace volume of a manipulator where only the solution of the inverse kinematics problem is exactly known.

In summary, we note that the *well-conditioned* and *reachable* workspace of a parallel manipulator is an integration problem in task space of the parallel manipulator. However, formulating the function $\mathcal{F}$ as shown in equation (3.6) directly in the task space is a very difficult notion for parallel manipulators with redundant joints and higher degrees of freedom because it involves exactly solving the direct kinematics problem. Therefore, to overcome this problem, we formulate the *in-out* classifier $\mathcal{F}$ in the joint space. We do this, by first solving the inverse kinematics problem of the parallel manipulator $\mathcal{IK}(\mathbf{X}) = \{\theta, \phi\}^T$, where $\theta_i,\ \forall i = 1, 2, ..., n$ are the $'n'$ actuated joint variables and $\phi_j,\ \forall j = 1, 2, ..., m$ are the $'m'$ passive joint variables. Next, we check the active and passive joint limits to ensure that they are within the prescribed limits, following which we ensure that, for the given position and orientation of the manipulator, the Jacobians are sufficiently well-conditioned. The Monte Carlo simulation as discussed in this section classifies a set of given position and orientation of the end-effector of a manipulator

according to their occurrence in the well-conditioned workspace and thus we get a set of points in $\Re^3$ and a set of possible orientations of the end effector of the manipulator in $SO(3)$.
We demonstrate the method discussed above by two examples– in the first example we obtain the workspace of the human hand and validate it by performing actual experiments and in the second example, we obtain the workspace of the well known Stanford-JPL hand (Salisbury hand) [5].

3.2.3 Limitations of using Monte Carlo method

In spite of having several advantages over other numerical methods in computing the workspace of a manipulator, the method suffers from two major disadvantages regarding the completeness and exactness of the representation of the workspace.

The representation of the workspace as given by the Monte Carlo method is not complete. The algorithm being probabilistic, can only give the value of the volume/area workspace with a certain accuracy. So, the quantity well conditioned workspace is better represented by Monte Carlo method than quantified by it.

The representation of the workspace as given by the Monte Carlo method is not exact. This means that a point 'P' obtained on the boundary of the workspace ($\partial\mathcal{S}$) for a particular set of joint limits constraints and upper bound on the condition numbers of the manipulator Jacobians, the method cannot give the radius r_B of the ball B centered at P, when all the points inside the ball lie on the well-conditioned workspace. For example if we obtain the workspace of the SCARA robot as given in figure 3.3b, using Monte Carlo methods, we cannot give the guarantee that a point P^*, situated at a distance r_B or less from a point P chosen on the workspace boundary represented by $\partial\mathcal{S}$, is also inside the workspace, neither can we obtain a value of r_B from the Monte Carlo method. This is not a problem for either of analytical or interval analysis based methods for obtaining and representing the workspace.

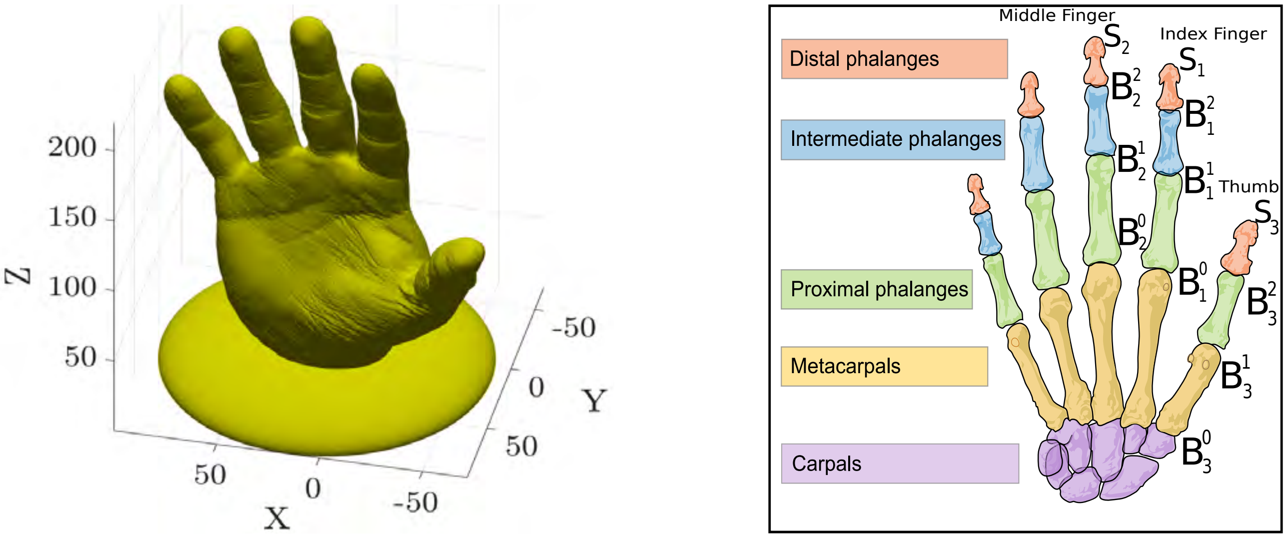
3.3 Example 1: Workspace of multi-fingered hands

The ability of dexterous manipulation has been practiced to perfection by us throughout several thousands of years of evolution ([86]) and therefore it is not surprising that the human hand is the gold standard for design of mechanisms for dexterous manipulation ([87]). With this motivation, the robotics community has been involved in developing multi-fingered hands that can achieve the dexterity, accuracy and load carrying capacity of human hands. Investigation into the topic started in late 1960s with the development of myo-electric devices and subsequently significant advancements have made in the area till date (for a comprehensive review of the state-of-the-art in this topic till early 2000s, see review paper [88] and the references contained therein). Some of the early major advances were a robotic hand with elastic fingers ([89]), the Salisbury hand ([5]), the Utah-MIT hand ([6]) and the Styx hand ([90]). Post 2000s, due to availability of better manufacturing techniques, more complex hands like the DLR hand ([91]) and the Shadow hand ([92]) were designed and developed. Several of these hands have large degrees of freedom (≥ 20), such as the Utah-MIT hand, Shadow hand, DLR hand, and the RBO hand ([93]) have focused on simulating human like grasping and manipulation of objects with complex shapes. While the anatomically correct testbed (ACT) hand ([94]),focuses on accurately mimicking human hand joint kinematics. It maybe noted that none of these hands have capabilities close to that of a human hand and one of the reason could be that the human hand, in addition to sophisticated sensing capabilities, has many more links and controlled joints – it is estimated that there are 27 bones controlled by 38 muscles and almost 40 tendons (see [87] and [95]) and the human hand has 27 degrees of freedom.

From modeling and theoretical development viewpoint, the works on obtaining conditions for form closure ([96], conditions for stable grasps in $\Re^3$ ([97]), contact equations between arbitrary smooth bodies ([98] and [99]), grasp criteria and grasp Jacobians ([100]) are some of significant milestones. In another line of research (see, for example, Salisbury [5] and most recently Boras [7]), researchers have explored dexterous manipulation from the context of a parallel manipulator focusing on dexterity, precision of manipulation of a given object in a given workspace by considering a lower degree of freedom (~ 6) approximation of the human hand. In this work we start with the anatomical model of the palm and 3 fingers (index, middle fingers and thumb) of the human hand ([101]) and approximate it as a 6 degree of freedom hybrid parallel manipulator by ruling out certain impossible and unnecessary motions (see [95]). Next, we describe models of contact between the finger tips and an object during manipulation and we consider two models – a point contact with friction modeled as a three-degree-of-freedom spherical joint and a rolling without slipping contact which is also ‘instantaneously’ modeled as a three-degree-of-freedom joint. From the kinematic models, we solve the inverse kinematics problem for the proposed manipulator and obtain the *well conditioned* workspace of the parallel hybrid manipulator using a Monte Carlo method. The main contributions of this example are: **a**) an approach to find the well-conditioned workspace of a human hand inspired 6 degree-of-freedom hybrid parallel manipulator under two possible models of contact between the fingers and the grasped object, **b**) we show that the optimum workspace is obtained when the size of the palm is approximately equal to the size of the grasped object, and the workspace when rolling without slipping is allowed is $\sim 1.2 - 1.5$ times more than when it is not allowed, and **c**) we perform experiments on dexterous manipulation with a human hand and show that the experimental results are in reasonable agreement with those obtained from numerical simulations and thus validate the algorithm proposed and the methods used in the example.

In section 3.3.1 we propose a human hand inspired parallel manipulator model and describe the two models of contact between fingers and objects during manipulation. In section 3.3.7 we describe numerical simulation results and experiments and discuss them.

3.3.1 Model of a human hand and dexterous manipulation of objects



(a) A 3D scanned model of the human hand

(b) Anatomy of human hand {`https://en.wikipedia.org/wiki/Hand`}

Figure 3.4: Anatomical and schematic representation of the human hand

Figure 3.4a shows a 3D scanned model of a human right hand[1] of a 39 year old Caucasian male. The model of the hand is rendered as an `.stl` file (see figure 3.4a). The anatomical details of a typical human hand are shown in figure 3.4b. For the index and middle fingers, the labels with a suffix 0, i.e., B_1^0 & B_2^0 represent the *metacarpo-phalangeal joints*, B_3^0 is the trapezium joint between the carpals and metacarpal bone of the thumb. For the index and middle fingers, the joints with suffix one, i.e., B_1^1 & B_2^1 are the joints between the proximal and intermediate phalanges, for the thumb, the joint B_3^1 indicates the joint between the metacarpal and the proximal phalanx bone. Finally, B_1^2 & B_2^2 indicate the joints between the intermediate

---

[1] The model was obtained by Mr. Georg Weber-Unger Jr. using an Eva$^{\text{TM}}$ scanner ([102]). The data for figures 2.5b and 3.4a are available in public domain and are being used with permission from their respective owners.

Table 3.5: Sample finger and hand segment lengths (refer figure 3.6 for symbols)

Hand part	**Symbols**	**Values (mm.)**
Index finger	$\{l_{11}, l_{12}, l_{13}\}$	$\{35,\ 23,\ 28\}$
Middle finger	$\{l_{21}, l_{22}, l_{23}\}$	$\{41,\ 22,\ 28\}$
Thumb	$\{l_{31}, l_{32}, l_{33}\}$	$\{45,\ 36,\ 34\}$
Palm	$\{d, h\}$	$\{13,\ 82\}$

and distal phalanges, for the thumb, the joint B_3^2 indicates the joint between proximal and distal phalanx of the thumb. Kinesiological studies (see the work [103] and the references contained therein) have shown that all the joints in the human finger do not equally participate in the prehensile movements of the human hand. For a given grasping task, the motion is generally started from the proximal joints B_1^0 , B_2^0 & B_3^0 and end in the distal joints B_1^2 , B_2^2 & B_3^2, with the proximal joints being active for most of the time [103], the proximal metacarpo-phalangeal joint was active for more than 59% of the motion, the proximal interphalangeal joint was active for 32% of the time and the distal inter-phalangeal joint was active for only 9% of the total time during the execution of a palmer pinching grasping task of a disc from a fully extended position. Based on this reasoning we assume that the proximal joints are actuated and we fix the distal joints of the index and middle fingers B_1^2 , B_2^2 and make B_3^2 passive. This also makes the model amenable to kinematic analysis since we have 9 joints in the three fingers with six actuated joints which can provide six degrees of freedom to the grasped object. The joints in the fingers have limits and we conservatively choose the joint limit ranges to be at most ranging from 0° to 90°. This is somewhat less to the angles specified by [95], [104], and [105]. Table 3.5 gives the value of the finger segments of the hand shown in figure 3.4a measured by using [106]. Table 3.6 discusses about the joint types and the joint ranges and values of the proposed manipulator.

The possible contacts between the finger tips and the grasped object has been extensively studied in literature (see, for example, [72] , [98], [99], and [100]). In this work we study the workspace of the grasped object for two kinds of contact, namely, point contact with friction and point contact with rolling without slipping.

3.3.2 Point contact with friction

Following [107] we can express the wrench basis for point contacts with friction as in equation (3.11) and assuming a Coulomb friction model, the forces transmitted by the contact on the body is given by equation (3.12). In equation (3.11), I_3 is the identity matrix of the order 3 and Φ_3 is the 3×3 null matrix and in equation (3.12), μ is the coefficient of friction and f_X, f_Y

Table 3.6: Joint notations in figure 3.6 and maximal permissible motions. The joints with least motion have been selected to remain passive.

Joint center	**Joint variable**	**Type**	**Value/range**
B_1^0 and B_2^0	θ_1 and θ_2	Active	0° to 90°
B_3^0	θ_3	Active	$-45°$ to 45°
B_1^1 and B_2^1	ϕ_1 and ϕ_2	Active	0° to 90°
B_3^1	ϕ_3	Active	0° to 90°
B_1^0 and B_2^0	ψ_1 and ψ_2	Passive	$-15°$ to 15°
B_3^2	ψ_3	Passive	0° to 60°
B_1^2 and B_2^2	γ_i and γ_m	Fixed	[0° - 30°]
B_3^0	γ_t	Fixed	$\gamma_t = 45°$
S_1, S_2 and S_3	$\{\xi_X^i, \xi_Y^i\}$	Passive	±45°

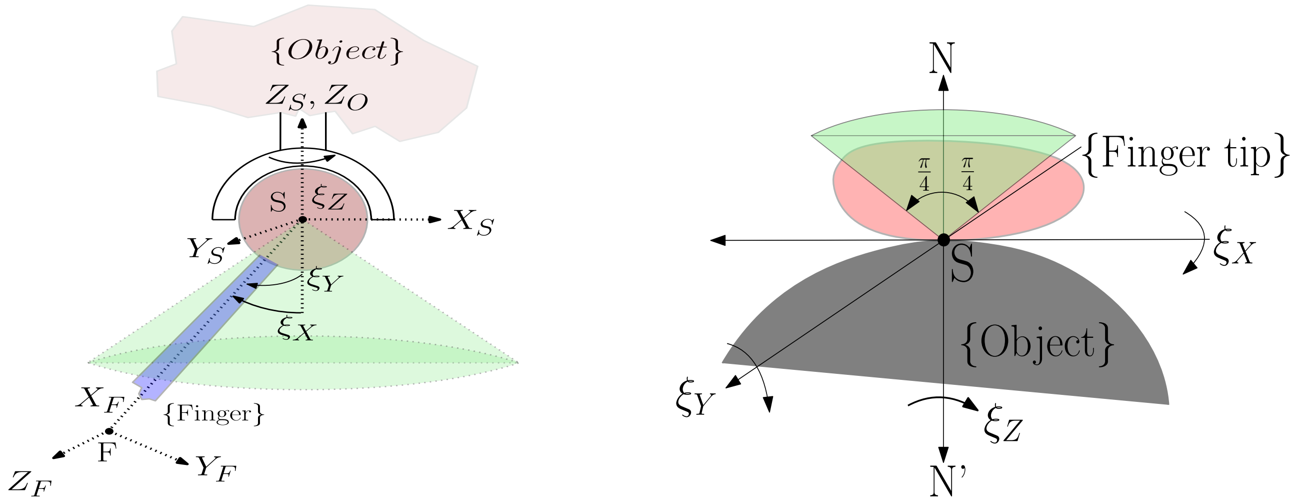
and f_Z are the forces transmitted by the point of contact along the X, Y and Z directions.

$$\mathbb{W} = \begin{bmatrix} I_3 \\ \Phi_3 \end{bmatrix} \tag{3.11}$$

$$F_o = \mathbb{W} \begin{bmatrix} f_X \\ f_Y \\ f_Z \end{bmatrix}, \quad \sqrt{f_X^2 + f_Y^2} \leq \mu f_Z \tag{3.12}$$

Since the *ideal* S joint has the same wrench basis as points of contact with friction and it can transmit forces only in the directions given by equation (3.12), we can replace the contact point by a spherical joint, given that there is negligible motion of the point of contact with respect to the body (see [100] for a similar argument). Figure 3.5 shows the point of contact modeled as spherical joints with the point S in the figures indicating the center of the joint.

Figure 3.5b shows the interaction of the finger and the object, when the finger tip is approximated as a rigid object with lateral dimensions. The point of contact "S" is assumed to be the center of the S-joint. The green cone in figures 3.5a and 3.5b denotes the solid angle subtended by the finger about the axis Z_S. A method to obtain the spherical joint motion has been outlined in appendix A, using which, we can obtain the values ξ_X and ξ_Y as rotations about X_S and Y_S, respectively. With the approximation of the fingertip as a point, the range of the values of ξ_X and ξ_Y can be as high as ±90°, however, for fingertips with physical dimensions, such a high value is not possible since, the finger-tip will have a tendency to roll over the object and the point of contact will change. This is shown in figure 3.5b, where, a smaller permissible range of ξ_X and ξ_Y has to be taken. For our simulation, we will consider a range of ±45°, shown as the green cone in figure 3.5b. A schematic of the three-fingered hand with S



(a) Schematic of the spherical joint (b) Interaction of the finger with the object

Figure 3.5: Spherical joint approximation of the finger-tips with object

joint modeling the contact is shown in figure 3.6.

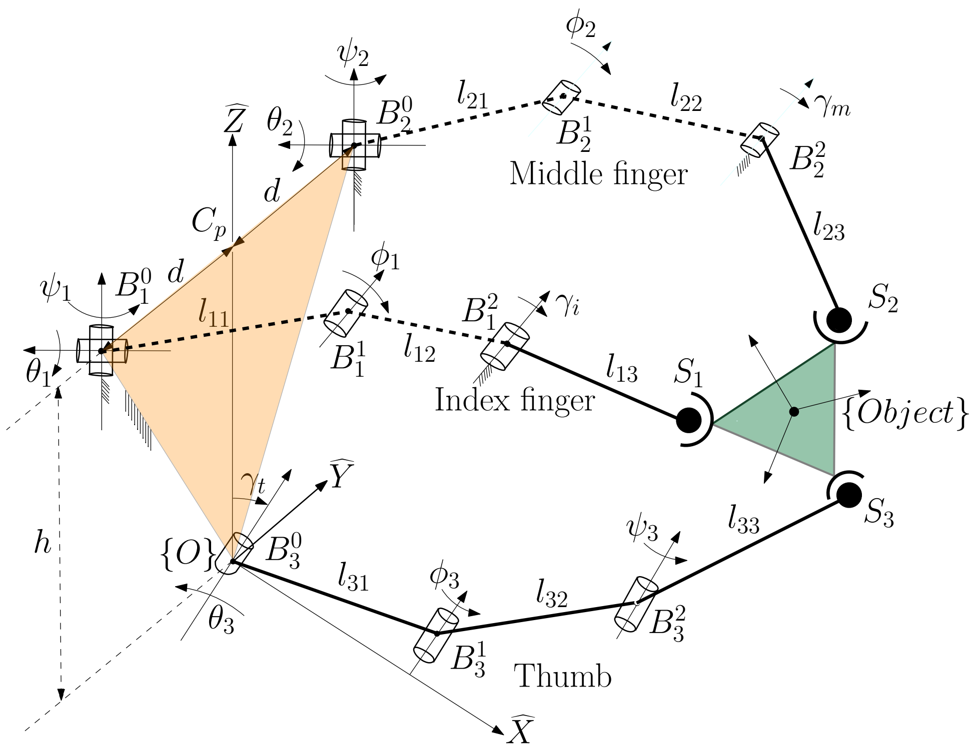


Figure 3.6: Schematic of the parallel manipulator

3.3.3 Inverse kinematics

The hand is approximately modeled as a 6 DoF hybrid parallel manipulator where the palm and the object being manipulated are denoted by platforms and each of the fingers are denoted by a serial 3R chain with two actuated and one passive joint per chain. Using the Grübler-Kutzbach criterion, it can be observed the degree of freedom of the hybrid parallel manipulator shown in Figure 3.6 is six. For a given position vector of the point S_1 (see figure 3.6), the expressions of the X, Y and Z coordinates are given by

$$\begin{aligned} X &= l_{11}\cos\psi_1\cos\theta_1 + l_{12}\cos(\phi_1+\theta_1)\cos\psi_1 \\ &+ l_{12}\cos(\gamma_i+\phi_1+\theta_1)\cos\psi_1 \qquad (3.13) \\ Y &= l_{11}\sin(\psi_1\cos\theta_1 + l_{12}\cos(\theta_1+\phi_1)\sin\psi_1 \\ &+ l_{13}\cos(\gamma_i+\phi_1+\psi_1+\theta_1)\sin\psi_1 - d \qquad (3.14) \\ Z &= -\sin(\phi_1+\gamma_i+\theta_1)l_{13} - \sin(\theta_1+\phi_1)l_{12} \\ &- l_{11}\sin\theta_1 + h \qquad (3.15) \end{aligned}$$

and we can obtain an expression for E_1 denoting the quantity $X^2 + (Y+d)^2 + (Z-h)^2$ as

$$\begin{aligned} E_1 &= (2\,\cos(\gamma_i)\,l_{11}l_{13} + 2\,l_{12}l_{11})\cos(\phi_1) - 2\,l_{13}\sin(\gamma_i)\sin(\phi_1)\,l_{11} \\ &+ 2\,l_{13}\cos(\gamma_i)\,l_{12} + {l_{11}}^2 + {l_{12}}^2 + {l_{13}}^2 \qquad (3.16) \end{aligned}$$

From the expressions for E_1 and Z, noting that γ_i is constant, using the Sylvester's dialytic method, we can obtain the eliminant for θ_1, using standard tangent half angle substitution, as a quartic function of the angular variable. We note that a significant simplification may be obtained by noting that $(2\cos(\gamma_i)l_{12}l_{13} + l_{12}^2 + l_{13}^2)$ is a common factor to either sides of the while obtaining the eliminant from equations 3.15 and 3.16, using Sylvester's dialytic method. The value of ϕ_1 can be obtained symbolically from the dialytic method (see Ghosal [81]) and the value of ψ_1 is obtained symbolically by using terms from the expressions of X and Y as

$$\begin{aligned} \psi_1 = \cos^{-1}[\sin(\pi/4)(X-Y-d)/(\cos(\theta_1+\phi_1)(l_{12}+l_{13}\cos\gamma_i) \\ -\sin(\theta_1+\phi_1)\sin\gamma_i l_{13} + \cos\theta_1 l_{11})] - \pi/4 \qquad (3.17) \end{aligned}$$

The inverse kinematics problem for the middle finger can be solved in a way similar to index finger shown above and the inverse kinematic problem for the thumb is simpler as it can be reduced to a quadratic polynomial. It maybe noted that we have 4 solutions for θ_1 and the choice of the solutions should take into account the joint limits in table 3.6. Of the four solutions of θ_1 obtained from the eliminant, we admit only those which correspond to permissible joint motions (as shown in table 3.6). Each of the permissible values of θ_1 are then used to obtain ϕ_1 and only a pair of $\{\theta_1, \phi_1\}$ are admitted, if *both* correspond to permissible joint motions and finally, the admitted values of θ_1 and ϕ_1 are used to find ψ_1. If and only if $\theta_i, \phi_i, \psi_i,\ \forall\ i = 1, 2, 3$ correspond to permissible joint motions, the solution set is accepted and any violations at any step leads to the rejection of the entire set and also randomly chosen position and orientation of the object. It may also be noted that this check also prevents change in working modes of the manipulator– i.e. flipping of the end effector $(S_1S_2S_3)$ with respect to the palm $(B_1^0B_2^0B_3^0)$ as shown in figure 3.6.

3.3.4 Jacobian matrices and well-conditioning

The position vector of the center of the object in figure 3.6 is given by,

$$ {}^{O}P_{Obj} = \frac{1}{3}\sum_{i=1}^{3} {}^{O}S_i \tag{3.18}$$

and the orientation of the top platform with respect to the base is given by

$$ {}^{O}_{Obj}[R] = \left[\begin{array}{ccc} \dfrac{{}^{O}S_1 -^{O} S_2}{|{}^{O}S_1 -^{O} S_2|} & \hat{Y} & \dfrac{({}^{O}S_1 -^{O} S_1) \times ({}^{O}S_1 -^{O} S_3)}{|({}^{O}S_1 -^{O} S_1) \times ({}^{O}S_1 -^{O} S_3)|} \end{array} \right] \tag{3.19}$$

where $\hat{Y}$ is obtained by the cross product of the third and first column of the matrix in equation (4.22). Differentiating equations (4.21) and (4.22) with respect to time we obtain the expressions for the linear and angular velocities of the manipulator.

The 3 constraint equations ensuring that the distance $||S_i - S_j||$, $\{i, j\} \in [1, 2, 3], i \neq j$, are always constant, may be differentiated to obtain

$$ [\mathbf{K}]\{\dot{\theta}, \dot{\phi}\} + [\mathbf{K}^*]\dot{\psi} = 0 \tag{3.20}$$

where $\{\theta_i, \phi_i,\ i = 1, 2, 3\}$ are assumed to be actuated and $\psi,\ i = 1, 2, 3$ are assumed to be passive. It is easily seen that $[\mathbf{K}^*]$ is a square matrix of dimension 3×3 and equation (3.20) can be solved for $\dot{\psi}$, given $\det(\mathbf{K}^*) \neq 0$[1], and we can obtain

$$\dot{\psi} = -[\mathbf{K}^*]^{-1}[\mathbf{K}]\{\dot{\theta}, \dot{\phi}\} \tag{3.21}$$

By differentiating the expressions for the linear and angular velocities of the end-effector and partitioning the expression for $\{\dot{\theta}, \dot{\phi}\}$ and $\dot{\psi}$, we have,

$${}^0V_{Obj} = [\mathbf{J}_V]\{\dot{\theta}, \dot{\phi}\} + [\mathbf{J}_V^*]\dot{\psi} \tag{3.22}$$

$${}^0\omega_{Obj} = [\mathbf{J}_\omega]\{\dot{\theta}, \dot{\phi}\} + [\mathbf{J}_\omega^*]\dot{\psi} \tag{3.23}$$

Following [82] and using equations (3.20), (4.23) and (4.24) we define the equivalent Jacobian matrices for both linear and angular velocity parts as

$$\mathbf{J}_{eqv}^V = (\mathbf{J}_V - \mathbf{J}_V^*[\mathbf{K}^*]^{-1}[\mathbf{K}]) \tag{3.24}$$
$$\mathbf{J}_{eqv}^\omega = (\mathbf{J}_\omega - \mathbf{J}_\omega^*[\mathbf{K}^*]^{-1}[\mathbf{K}]) \tag{3.25}$$

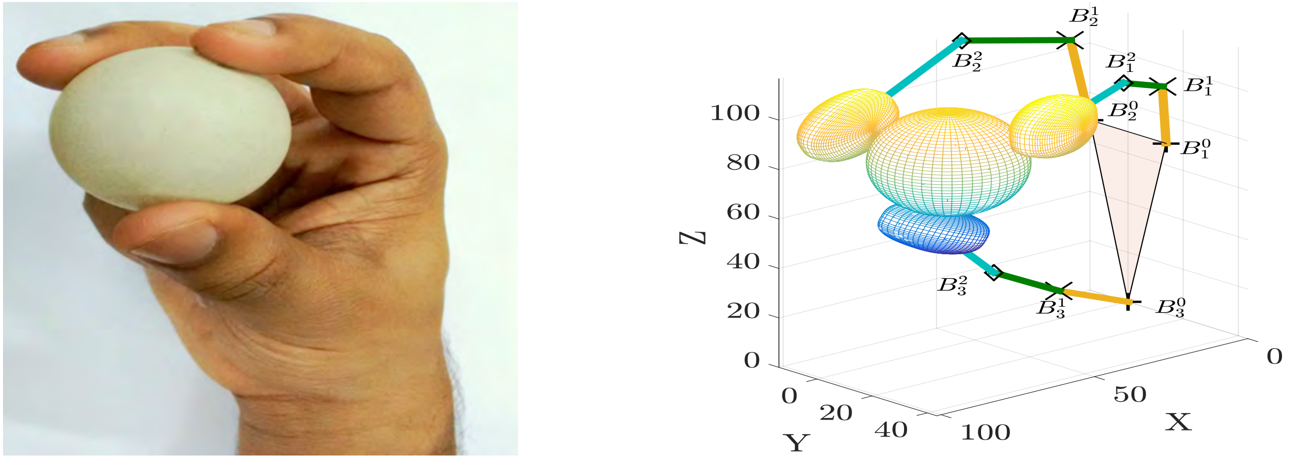
The equivalent, dimensionless condition number of Jacobian for the manipulator, undergoing both linear and angular motions are given as κ_V and κ_ω for $\mathbf{J}_{eqv}^V$ and $\mathbf{J}_{eqv}^\omega$, respectively where we find the 2-Norm condition number of a matrix $\mathbf{A}$ as $\kappa_A = \| \mathbf{A} \|_2 \| \mathbf{A}^{-1} \|_2$. To ensure that a given configuration of the end effector is well conditioned we ensure that both the condition numbers are limited to an upper limit of κ^* as shown in equation (3.5). In our simulations, we ensure that $\kappa^* \leq 1000$. This gives a conservative workspace and a higher value of κ^* can be shown to give a larger well conditioned workspace.

3.3.5 Point contact with rolling without slipping

Majority of dexterous manipulation tasks undertaken by our hands involve rolling and sliding type of contact between the hand and object (see [98] for more details). The rolling type of contacts between the fingertips and the object during dexterous manipulation has been widely studied (see for example the pioneering works by [98], [99] and [100] and the references contained therein) and most of these approaches are based around the definition of a quantity called the *grasp Jacobian* or the *grasp map*. In this work, we assume that the grasped object can roll without slipping and we use the contact equations developed by [98]. In this case too, it can be shown that the contact *instantaneously* allows three relative degrees of freedom. The main

---

[1] In the simulation, it was ensured that $\det(\mathbf{K}^*) \neq 0$ and the condition number of $\mathbf{K}^*$ was $\leq 10^4$ at all points inside the obtained workspace.



(a) Human hand manipulating a ball

(b) Snapshot of the computer simulation of the scenario shown in figure 3.7a

Figure 3.7: Dexterous manipulation in practice and simulation snapshot

difference with the spherical joint model is that the constraints are non-holonomic, i.e., they restricts the space of velocities and not the translation and orientation of the grasped object. In order to model the rolling without slip contact, we need a surface model of the finger and the grasped object. The grasped object is assumed to be a sphere and the well-known parametric equations of a sphere is used. As discussed in section 2.4, we will approximate the finger-tips as ellipsoids and will use the parametic description of the fingertips form equations (2.20) to (2.22) along-with the values from tables 2.1 and 2.2 in the formulating and solving the equations of rolling as discussed in section 3.3.5.2.

3.3.5.1 Dexterous manipulation of objects using rolling contacts

To obtain the workspace when rolling without slipping is permitted at the contact between the fingers and the grasped object, we propose an approach based on the existence of the solution of the inverse kinematics problem of the human hand posed as a hybrid parallel manipulator, condition number of the equivalent Jacobian, experimental information about common three fingered dexterous manipulation tasks (see section 3.3.7.1) and equations of rolling (see section 3.3.5.2). Figure 3.8a describes the finger and the object both modeled as super-ellipsoids in contact at the point C. Axes with suffixes f and o are contiguous to the finger and the object, respectively, O_G is the origin of global coordinate frame in $\Re^3$ and ${}^G\mathbf{P}_C$ is the vector joining the point of contact C with O_G. $\mathcal{S}_f$ and $\mathcal{S}_o$ are the functions that map the 2 dimensional parameter-space $\{u_i, v_i\}$ to *smooth* and *closed* super-ellipsoids in $\Re^3$ using equations (2.20) to (2.22). We define quantity ψ, as the heading angle, which gives the angle by which the finger frame has

to be rotated about the common normal ($\overrightarrow{Z_f Z_0}$ in figure 3.8a) to align X_o and X_f. Due to the contact at C, no motion is possible along the common normal $\overrightarrow{Z_f Z_0}$, therefore, the five other possible motions are solved with respect to the 4 parameters describing the contact point on $\mathcal{S}_f$ and $\mathcal{S}_o$ and the heading angle. A brief formulation of the problem of obtaining the evolution of the contact points and heading angle i.e., $\{u_f, v_f, u_o, v_o, \psi\}^T$ as the object rolls on the fingertips is given in section 3.3.5.2.
To obtain the evolution of $\{u_f, v_f, u_o, v_o, \psi\}^T$, using equations (3.34) to (3.36) given in section 3.3.5.2, we need the initial conditions for $\{u_f, v_f, u_o, v_o, \psi\}^T$ and the velocities of the fingertip with respect to the object in contact, i.e., $\{\omega_x, \omega_y, \omega_z\}$ and $\{v_x, v_y\}$. Using terms for Jacobian from equations (4.27) and (4.28) we can solve for all the joint rates (both actuated and passive) for a finger from the equation

$$\left\{ \begin{array}{c} V_{obj} \\ ----- \\ \Omega_{obj} \end{array} \right\} = \left[\begin{array}{c} \mathbf{J}_v \\ ----- \\ \mathbf{J}_\omega \end{array} \right] \left\{ \begin{array}{c} \dot{\vec{\theta}} \\ \dot{\vec{\phi}} \end{array} \right\} \tag{3.26}$$

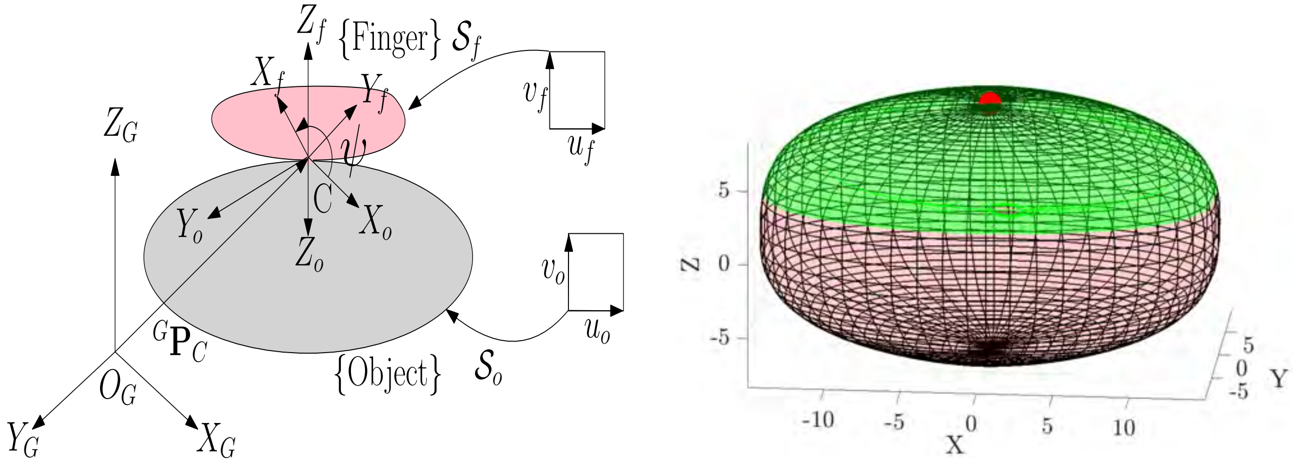
where $\{\dot{\vec{\theta}}, \dot{\vec{\phi}}\}$ denote the active and passive joints, respectively. It maybe noted that the system in equation (3.26) is not square, therefore, we opt for least squares solution [108].

Using the values of $\{V_{obj}\ \Omega_{obj}\}^T$ from section 3.3.7.1 in equation (3.26), we can obtain the angular and linear velocities of each fingertip, and therefore the expressions for $\{\omega_x, \omega_y, \omega_z\}$ and $\{v_x, v_y\}$ for each of the three fingers with respect to the object. The initial value for ψ may be obtained as the angle ξ_Z measured about Z_S in figure 3.5a. The initial values for $\{u_f, v_f, u_o, v_o\}$ are chosen so that the motion starts when the contact is at the center of the finger which is shown as a red disk in figure 3.8b. The green zone indicates the zone in which the contact is allowed between the object and the finger.

3.3.5.2 Equations of rolling after Montana[98]

The common normal $\mathbf{n}$, to the surfaces $\mathcal{S}_f$ and $\mathcal{S}_o$ (in figure 3.8a) is given in equation (3.27) according to [71].

$$\left\{ \begin{array}{c} n_x \\ n_y \\ n_z \end{array} \right\} = \left\{ \begin{array}{c} \dfrac{\operatorname{sgn}(\cos(v))\,(|\cos(v)|)^{2-n}\operatorname{sgn}(\cos(u))\,(|\cos(u)|)^{2-e}}{A} \\ \dfrac{\operatorname{sgn}(\cos(v))\,(|\cos(v)|)^{2-n}\operatorname{sgn}(\sin(u))\,(|\sin(u)|)^{2-e}}{B} \\ \dfrac{\operatorname{sgn}(\sin(v))\,(|\sin(v)|)^{2-n}}{C} \end{array} \right\} \tag{3.27}$$



(a) Contact between the finger and the object (b) Allowable contact zone on the index finger

Figure 3.8: Description of two bodies in contact and permissible contact zone on fingertip

Using equations (2.20) to (2.22) we can obtain a parametric expression for the surface as $\mathbf{f}(u,v) = \left\{ \begin{array}{c} x(u,v) \\ y(u,v) \\ z(u,v) \end{array} \right\}$. At a point P_0 on the surface we define a orthogonal frame $\mathbf{H}$ as

$$\mathbf{H} = \{\mathbf{f}_u/|\mathbf{f}_u|,\ \mathbf{f}_v/|\mathbf{f}_v|,\ \hat{\mathbf{n}}\}^T \tag{3.28}$$

In equation (3.28), $\mathbf{f}_{v^*} = \dfrac{\mathbf{n}}{|\mathbf{n}|} \times \dfrac{\mathbf{f}_u}{|\mathbf{f}_u|}$ [1] and $\mathbf{k}_{(\cdot)} = \dfrac{\partial \mathbf{k}}{\partial(\cdot)}$ and the value of the contact normal $\mathbf{n}$ is obtained from equation (3.27). Based on $\mathbf{H}$ in equation (3.28) we can define the following metric $[\mathbf{M}]$, curvature form $[\mathbf{K}]$ and torsion form $[\mathbf{T}]$ for each of the surfaces $\mathcal{S}_f$ and $\mathcal{S}_o$ as:

$$[\mathbf{M}] = \begin{bmatrix} |\mathbf{f}_u| & 0 \\ 0 & |\mathbf{f}_{v^*}| \end{bmatrix} \tag{3.29}$$

$$[\mathbf{K}] = \begin{bmatrix} \dfrac{\mathbf{f}_u.\mathbf{n}_u}{|\mathbf{f}_u|^2} & \dfrac{\mathbf{f}_u.\mathbf{n}_{v^*}}{|\mathbf{f}_u||\mathbf{f}_{v^*}|} \\ & \\ \dfrac{\mathbf{f}_{v^*}.\mathbf{n}_u}{|\mathbf{f}_u||\mathbf{f}_{v^*}|} & \dfrac{\mathbf{f}_{v^*}.\mathbf{n}_{v^*}}{|\mathbf{f}_{v^*}|^2} \end{bmatrix} \tag{3.30}$$

$$[\mathbf{T}] = \begin{bmatrix} \dfrac{\mathbf{f}_{v^*}.\mathbf{f}_{uu}}{|\mathbf{f}_u|^2|\mathbf{f}_{v^*}|} & \dfrac{\mathbf{f}_{v^*}.\mathbf{f}_{uv^*}}{|\mathbf{f}_{v^*}|^2|\mathbf{f}_u|} \end{bmatrix} \tag{3.31}$$

---

[1] This re-parametrization is required for the general super ellipse as the vectors $\{\mathbf{f}_u/|\mathbf{f}_u|, \mathbf{f}_v/|\mathbf{f}_v|, \hat{\mathbf{n}}\}^T$ do not constitute an orthogonal frame

In equation (3.31), $\mathbf{f}_{uv^*} = \mathbf{f}_{v^*u} = \frac{\partial}{\partial u}(\frac{\mathbf{n}}{|\mathbf{n}|} \times \frac{\mathbf{f}_u}{|\mathbf{f}_u|})$.

Also in equation (3.30), $n_{v^*} = \mathbf{n}_u u_{v^*} + \mathbf{n}_v v_{v^*}$, where, a *least squares estimate* values of u_{v^*} and v_{v^*}

can be obtained by comparing the components of both left and right sides of equation (3.32).

$$\mathbf{f}_{v^*} = \frac{\mathbf{n}}{|\mathbf{n}|} \times \frac{\mathbf{f}_u}{|\mathbf{f}_u|} = \mathbf{f}_u \frac{\partial u}{\partial v^*} + \mathbf{f}_v \frac{\partial v}{\partial v^*} \tag{3.32}$$

Again,

$$[R_\psi] = \begin{pmatrix} \cos\psi & -\sin\psi \\ -\sin\psi & -\cos\psi \end{pmatrix}, \quad [\mathbf{K}_f^*] = [R_\psi][\mathbf{K}_f][R_\psi]^T \tag{3.33}$$

Using equations (3.29) to (3.31) and (3.33) we can write the equations of motion of two bodies in contact undergoing *rolling* after [98] as,

$$(\dot{u}_o, \dot{v}_o) = [\mathbf{M}_o]^{-1}([\mathbf{K}_o] + [\mathbf{K}_f^*])^{-1}[\{-\omega_y, \omega_x\}^T - [\mathbf{K}_f^*]\{v_x, v_y\}^T] \tag{3.34}$$

$$(\dot{u}_f, \dot{v}_f) = [\mathbf{M}_f]^{-1}[R_\psi]([\mathbf{K}_o] + [\mathbf{K}_f^*])^{-1}[\{-\omega_y, \omega_x\}^T + [\mathbf{K}_o]\{v_x, v_y\}^T] \tag{3.35}$$

$$\dot{\psi} = \omega_z + [\mathbf{T}_o][\mathbf{M}_o](\dot{u}_o, \dot{v}_o)^T + [\mathbf{T}_f][\mathbf{M}_f](\dot{u}_f, \dot{v}_f)^T \tag{3.36}$$

Equations (3.34) to (3.36) along with $v_z = 0$ can be solved simultaneously to obtain the evolution of the contact points on the object and the finger.

3.3.6 Algorithm to obtain the well-conditioned workspace for the human hand

Figure 4.1 shows the algorithm we have used to obtain the volume and representation of the workspace of the under two types of contacts. For a given object size, and grip positions (see figure 3.7b) and given hand dimensions (see, for example, table 3.5) and finger tip models (see table 2.1), we define a search space $\mathcal{V}_s$ in $\Re^3$. Then we generate N_n random configurations $\mathbf{X} \in SE(3)$ where, the Cartesian variables are uniformly distributed in $\mathcal{V}_s$ and the Euler angles are uniformly distributed in $[-\pi/2, \pi/2]$. At each of these configurations, the algorithm in figure 4.1 is executed. Each of the blocks of the algorithm have either true or false as output, depending on the success and failure of the check undertaken by the block. Success in the check prompts a movement to the next block, in case of a failure, the configuration is rejected and a new configuration is generated and the check starts afresh. If a configuration passes through all the checks a counter value (N_s or N_r depending upon the type of workspace required) is

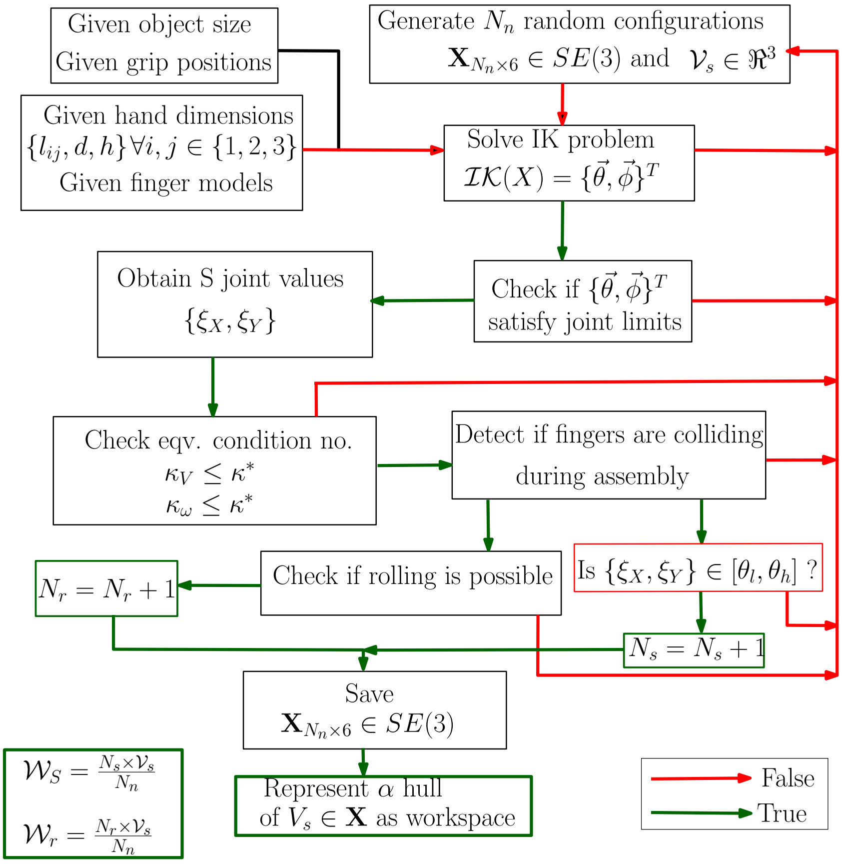


Figure 3.9: Block diagram of the proposed algorithm to obtain workspace. $[\theta_l, \theta_h]$ is the permissible range for the spherical joints.

increased and the configuration is saved for representation. However the arrangement of the different checks suggested in figure 4.1 is not unique and the particular arrangement has been chosen for better computational performance[1].

The blocks, as shown in figure 4.1, are coded to perform different checks independent of each other, with the inputs being hand dimensions and joint angles obtained from supplied data and solution of the IK problem. Therefore, the IK problem has to be solved for all N_n points for the three fingers and each evaluation of the IK problem for the fingers takes a time of the order of 3×10^{-4} seconds which is quite fast because of the symbolic implementation used. Next, we check if the chosen configuration is well conditioned or not by evaluating the condition number of the equivalent Jacobians (see equation (3.5)) and ensuring that they are less than 1000 at all times. This takes about 28.63×10^{-3} seconds due to the symbolic implementation of the total process. Simultaneously, we also obtain the S joint values, which takes 1.58×10^{-4}

---

[1] Computational performances are measured as CPU times are for Matlab R2015a[57] run in a 64bit Windows 7 PC with a Intel XEON processor (4 cores @ 3.10 GHz) and 16GB RAM

seconds. Next we check for fingertip collisions, which takes about 0.73 secs. Finally, we check for the possibility of rolling the given object on the fingertips, which takes about 1.95 secs per iteration. A fully parallel implementation of the algorithm to obtain the workspaces of a parallel manipulator by checking through 150,000 random configurations of the end effector takes about 624 seconds.

3.3.7 Numerical experiments and results

In this section we will discuss a few numerical experiments using the methodology developed so far, to get some insight into the design of the human hand and kinematics of dexterous manipulation using a parallel manipulator framework as described in section 3.3.2. We will start by discussing some physical insights obtained through experiments on dexterous manipulation and use those to design our numerical experiments. Following which we will discuss the numerical experiments and results obtained from the same.

3.3.7.1 Experimental insights into dexterous manipulation

In this section we discuss some experiments on human dexterous manipulation, to gain some insights on the range of motion and the speeds attainable during an in-hand manipulation task attainable by an individual. For recording the motion of the object we have used a [109] electromagnetic position tracker, which can log the position and orientation of an object in real time. The configuration of the object are obtained by rigidly fixing a *sensor* to it and logging the values of the 6 channels per sensor, 3 for absolute position and 3 for absolute orientation of the sensor with respect to the source of the magnetic field. We briefly describe our experimental procedure below.

Experiment: At first the wrist of a healthy adult subject is immobilized by splints and bandages to restrict all the 4 motions of the wrist viz. radial and ulnar deviations, palmar extension and flexion. Next, a tracking sensor is fixed approximately at the base of the trapezium bone (point B_0^3 in figures 3.4b and 3.6) on the dorsal side of the hand, and another tracking sensor is fixed approximately at the center of a spherical object to be used for the manipulation task. Figure 3.10a shows the location of the tracking sensors on the hand and the object during a manipulation task. We have used a standard ping-pong ball ($\sim$ 40mm dia.) and a standard tennis ball ($\sim$ 65mm dia.) for our manipulation tasks. The subject is asked to manipulate the object at *normal* speed while taking notice all three fingers are contacting the object at all times and the point of contacts are within the permissible zone as indicated in figure 3.8b. The manipulation task is usually about 60 to 100 seconds long and configuration data is logged for

(a) Location of sensors in the experiment

(b) Frequently encountered ill conditioned pose

Figure 3.10: Location of sensors on the hand and a known ill conditioned pose

both the sensors at 120 Hz throughout the duration of the experiment.

Data collection and post-processing: The data collected from the sensors are quite noisy with typical values of signal-to-noise ratios of 1.5 to 2 dB. The poor quality of the signals is chiefly attributed to interfering magnetic fields from various sources nearby to the channel. The channel noises for a sensor are found to be normally distributed with zero mean and standard deviations ranging between 0.003 to 0.03. To de-noise the channel signals we have used the method of non-local means (see e.g. [110] and the `Matlab` implementation of the same by [111]). The de-noised signal is then used to obtain configuration of the manipulated object with respect to the frame attached to the hand. Subsequently the gradients of the relative position and orientation of the object are represented as the linear and angular velocities of the manipulated object.

Results: From the analysis of the data from experiments with 5 male subjects we have obtained the following results:

- The manipulation area was bounded by the positive Cartesian octant, with slight deviations of the order of 2-3 mm along the Y direction.
- The orientations possible for the object was at most $\pm 90^\circ$ about an axis, with actual in-task motions ranging between $\pm 10^\circ$ (see figure 3.12b).
- Figure 3.11 shows the velocities attained by the ball in 3 separate experiments. The black lines show the velocities obtained by a subject (M_{30}) manipulating a table tennis ball,

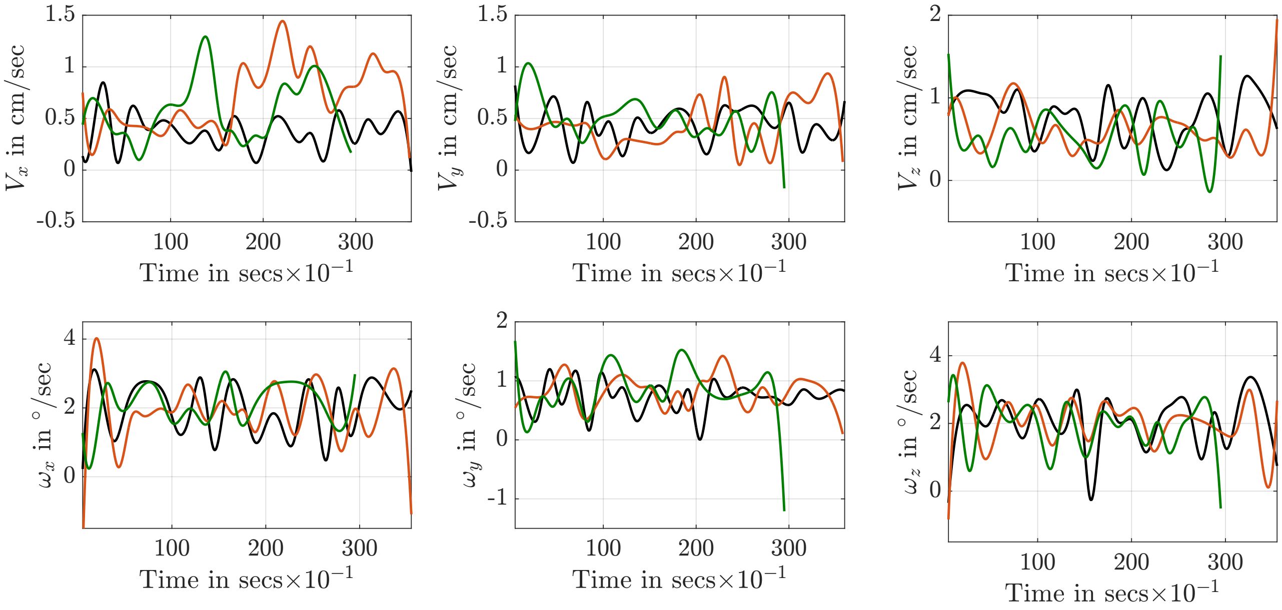


Figure 3.11: Typical velocities encountered during 3 different dexterous manipulation tasks shown in 3 colors

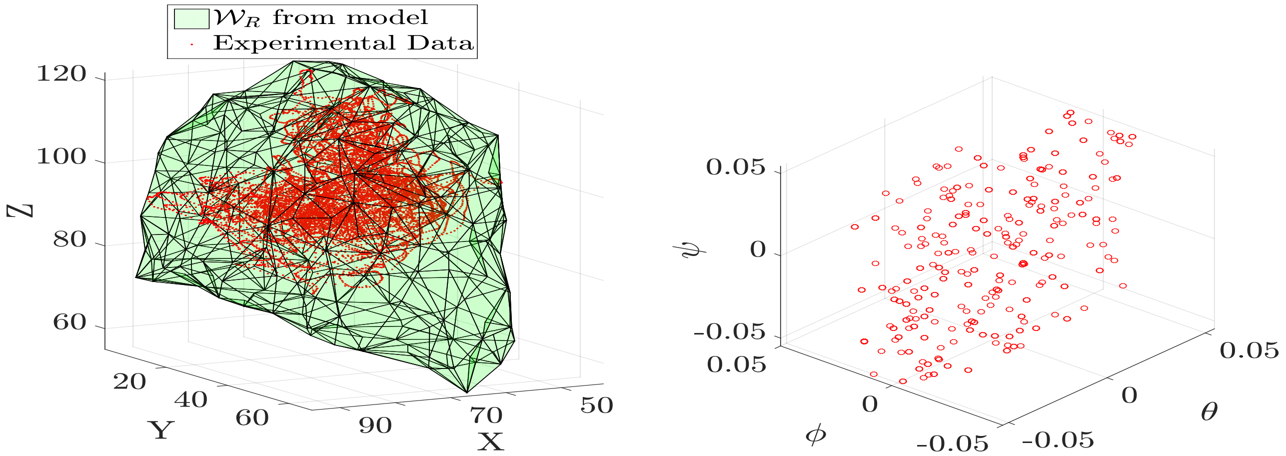
the orange line shows the velocities obtained by a different subject (M_{24}) manipulating the same ball and the green line shown the second subject manipulating a tennis ball. From the experiment we can conclude that there is *not much* variation in the velocities achieved by different subjects across two different manipulation tasks.

- Therefore, from figure 3.11 we obtain $V_x = V_y = V_z = 0.1$m/s and $\omega_x = \omega_y = \omega_z = 0.1$rad/s, which are used in equation (3.26) to obtain the joint rates of the manipulator for achieving the target velocities at the end effector.

Figure 3.12 shows the position and orientation data obtained from experiments. The position workspace (figure 3.12a) obtained from the experiments, when compensated for the sensor dimensions and wrist thickness entirely fits into the theoretically obtained workspace,

3.3.7.2 Obtaining envelopes for the human hand workspace

To further verify the efficacy of our algorithm, we demonstrate the performance of the algorithm for obtaining the workspace envelopes for actual human dexterous manipulation tasks using simple objects– a ping-pong ball (radius =20mm), a tennis ball (radius =34.5mm) and a cylinder (radius=20mm, height=20mm.). Table 3.7 shows the performance of the algorithm in obtaining an envelope of the actual workspace and figures 3.12a and 3.13a to 3.13d demonstrate the actual outputs of the algorithm and experiment plotted together. The cloud of points denote



(a) Comparison of the theoretically and experimentally obtained workspaces ($\mathcal{M}_{24}$ with ping-pong ball)

(b) Orientations achieved in experiments

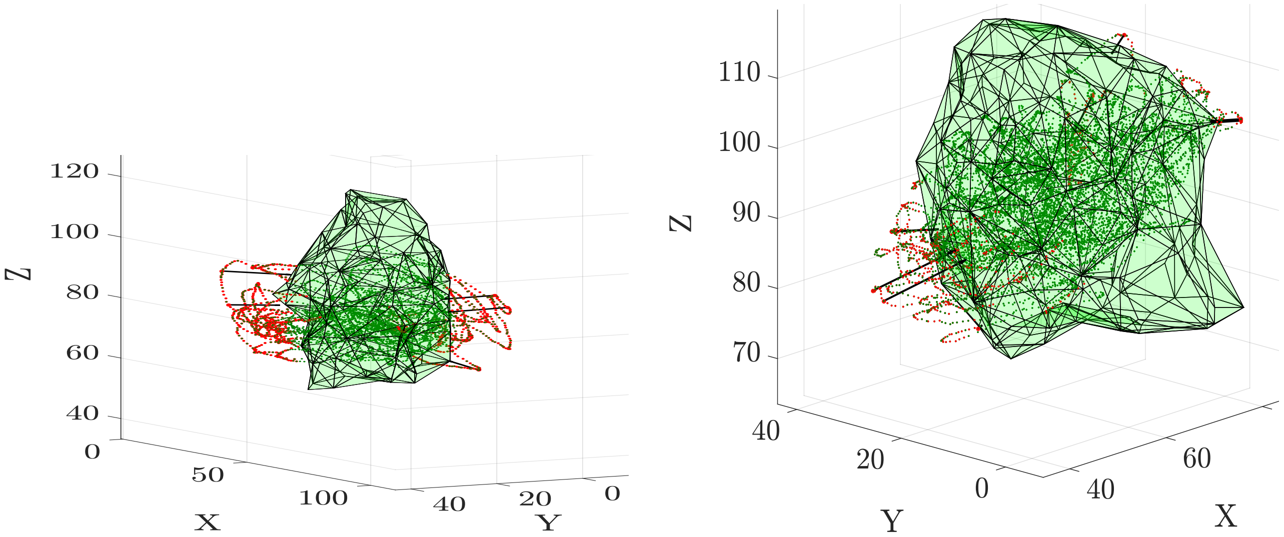
Figure 3.12: Experimental results on human hand workspaces

the experimental data and the portion of the experimental data, inside the envelope obtained by using our algorithm is colored sap green, the theoretically obtained envelope is shown as a transparent light green triangulated domain and the points outside the theoretical envelope are colored red. The black lines denote the distance of some of the farthest experimentally obtained points from the theoretically obtained envelope.

Table 3.7: Comparison of theory and experiment. All dimensions in mm. The normalized errors of the sensors along X,Y and Z direction was of the order of 4 ± 1mm ($\mu \pm \sigma$)

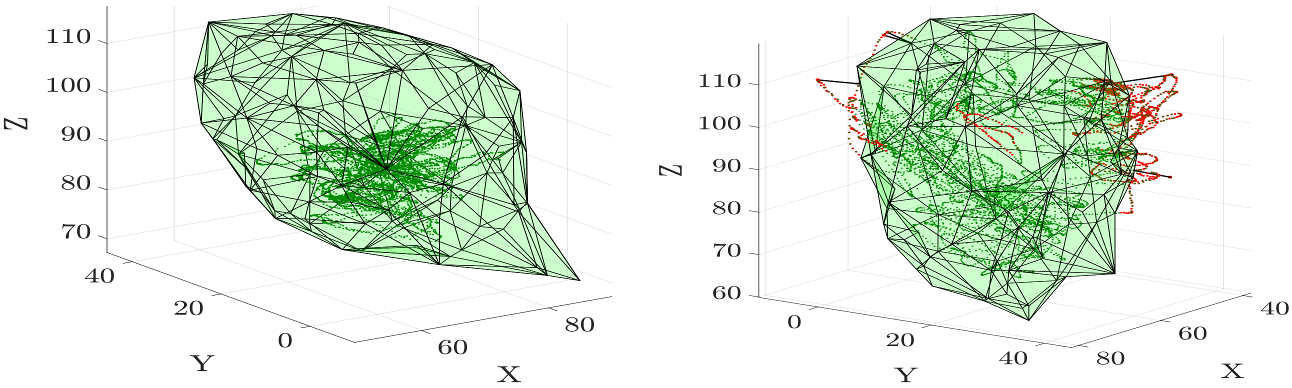
Subject	Object	Center(theory)	Center(expt.)	% points out	d_{max}
$\mathcal{M}_{24}$	Ping-pong	[64.06, 20.06, 93.34]	[64.19,18.5, 90.36]	0% of 10235	–
$\mathcal{M}_{24}$	Tennis	[55.12, 17.75, 87.68]	[56.3, 18.47, 79.32]	23.5% of 5356	15.5
$\mathcal{M}_{24}$	Cylinder	[64.93, 17.24, 93.73]	[63.92, 19.26, 91.72]	12.6% of 9647	17.24
$\mathcal{M}_{32}$	Ping-pong	[70.96, 18.73, 97.38]	[70.7, 16.72, 90.4]	0% of 10134	–
$\mathcal{M}_{32}$	Tennis	[63.41, 20.05, 94.53]	[60.63, 18.84, 94.56]	18.6% of 7184	12.5

We see that barring a few points obtained due to un-modeled motion of the fingers like like possible motion of the joint B_3^1 beyond 90°, possible slipping and temporary detachment of the index finger during the motion and some ill conditioned poses attained frequently during the manipulation task (see e.g. figure 3.10b where $\kappa_V \to \infty$). The inclusion of the experimentally obtained workspace in the theoretically obtained workspace, and the fact that the theoretically obtained workspace puts a reasonably conservative bound on the experimentally workspace (see figure 3.12a) may demonstrate the correctness of our modeling approach. However, it may also be noted that the experimentally obtained may never supersede the theoretically obtained one



(a) $\mathcal{M}_{24}$ with tennis ball

(b) $\mathcal{M}_{24}$ with cylinder



(c) $\mathcal{M}_{32}$ with ping-pong ball

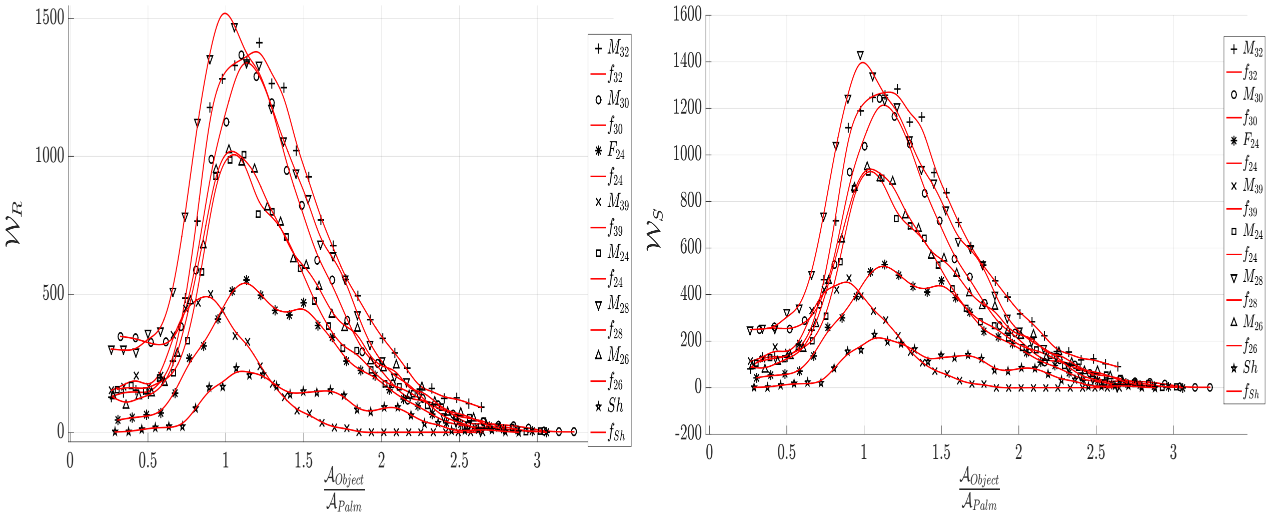
(d) $\mathcal{M}_{32}$ with tennis ball

Figure 3.13: Comparison between experimental workspaces and theoretical estimates of the same for different human subjects manipulating different objects

by large amounts because though we are using practically possible joint limits (see table 3.6), actual motions possible during a dexterous manipulation task are somewhat less than obtained maximal joint limits of each individual finger. We also do not consider the motion of the wrist and the orientations obtained by moving the wrist in the simulations, but the experiment could not completely suppress the motions of the wrist– which may have introduced some error.

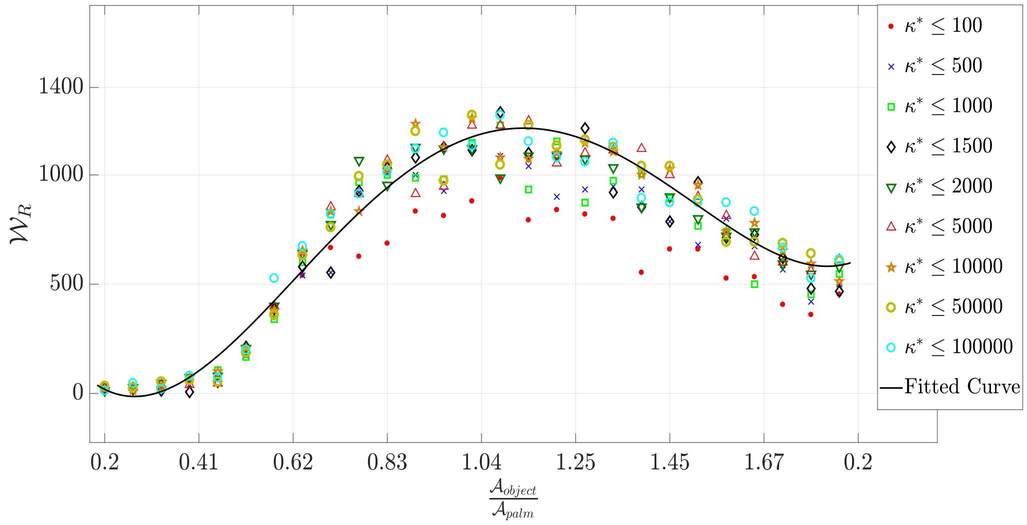
3.3.8 Human hand kinematics and workspaces

In this section we discuss the results we have obtained with our modeling. Through our algorithm as described in figure 4.1 we seek $\mathcal{W}_R$, the available *well conditioned* workspace considering rolling type of contact at the fingertips and $\mathcal{W}_S$, the available *well conditioned* workspace



(a) Rolling Workspace

(b) S-joint workspace

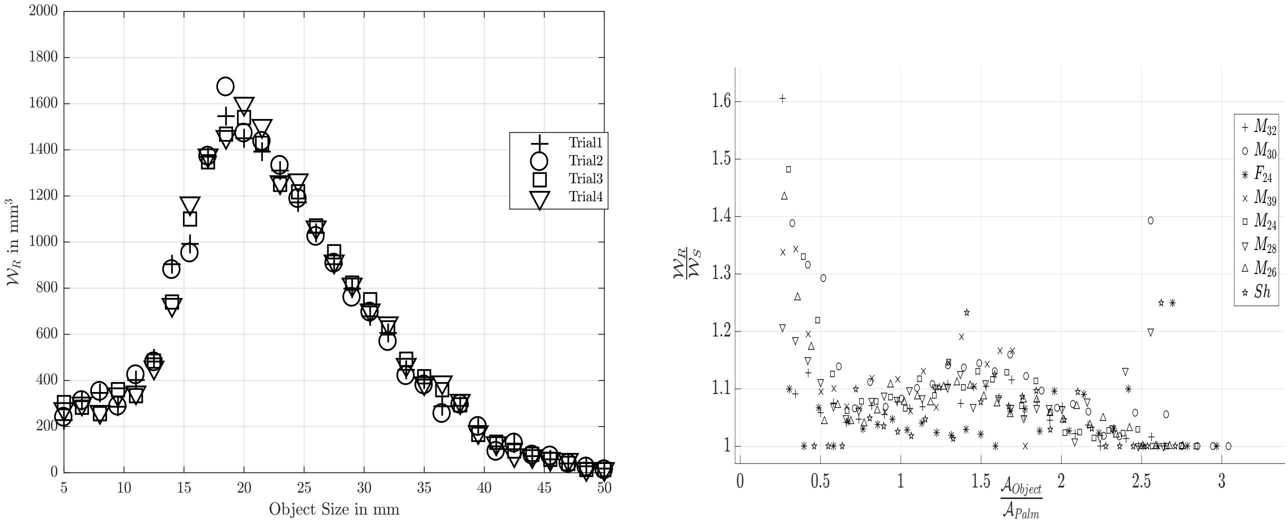


(c) Independence of the result in table 3.8 to choice of condition number in equation (3.5)

Figure 3.14: Comparison of workspaces of hand considering two different models of contact across different hands

considering S joint type of contact between the object and the fingertips. In each case, we start by defining the initial contact points as the vertexes of an equilateral triangle inscribed by the object modeled as a sphere. The search space $\mathcal{V} \in \Re^3$ is a box bounded by the planes $X = (0, 100)$, $Y = (0, 100)$ and $Z = (50, 150)$ (in mm), and the orientation search space in $SO(3)$ is bounded by $\{\theta, \phi, \psi\} \in [-90°, 90°]$. Figure 3.7b shows the snapshot of the simulation for obtaining $\mathcal{W}_R$ for a particular configuration of the object.

In figure 3.15a we show that our algorithm converges across 4 different trials for a particular set of hand dimensions. We have evaluated the algorithm for 60 object sizes ranging from 1mm diameter to 40mm diameter. We also observe from figure 3.15b that the ratio $r_{RS} = \dfrac{\mathcal{W}_R}{\mathcal{W}_s}$ also converges across different subjects and trials.



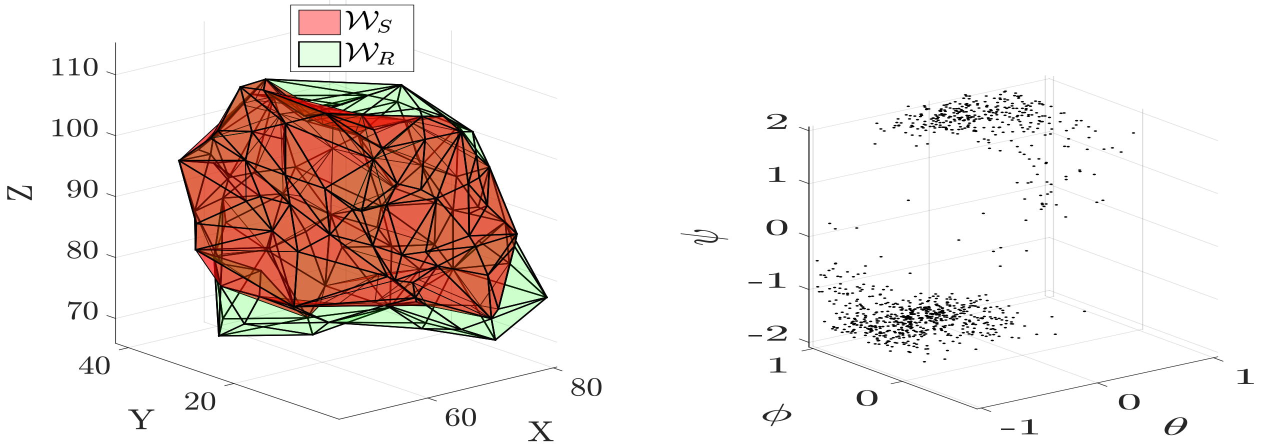
(a) Convergence of the algorithm across 4 different trials

(b) Variation of $\dfrac{\mathcal{W}_R}{\mathcal{W}_S}$ with change in object size

Figure 3.15: Convergence of the algorithm and variation of $\dfrac{\mathcal{W}_R}{\mathcal{W}_S}$ with change in object size

Figure 3.16 shows the possible positional and orientation workspaces of the manipulator. We note from figure 3.16a that $\mathcal{W}_S$ is smaller then $\mathcal{W}_R$ and is inscribed in the same and the maximum orientation workspace shown in figure 3.16b is not very significant for a dexterous manipulation task with a spherical object due to it's symmetry. We have included figure 3.16b to demonstrate that the entire workspace $\mathcal{W}$, $\mathcal{W} \in SE(3)$, can be obtained and represented by our algorithm.

From observations in daily life, the experiments in section 3.3.7.1 and results obtained so far we observe that the values of $\mathcal{W}_R$ and $\mathcal{W}_S$ vary widely with change in object size. Therefore, using the developments so far, we try to obtain a relationship between the *best* cross sectional area of a spherical object ($\mathcal{A}_{object}$) for dexterous manipulation and the human hand dimensions. For this we chose the palm area, area of the $\triangle B_1^0 B_2^0 B_3^0$ in figure 3.6 denoted by $\mathcal{A}_{palm} = d \times h$, as a characteristic dimension of the human hand and obtain the values of $\mathcal{W}_R$ and $\mathcal{W}_R$ with varying palm area-object area ratios $r_{po} = \dfrac{\mathcal{A}_{palm}}{\mathcal{A}_{object}}$. We choose 8 data sets designated as $\{M_{32}, M_{24}, M_{26}, M_{28}, M_{30}, M_{39}, F_{24}, Sh\}$ denoting the hand dimensions of 6 adult



(a) Possible positional workspace

(b) Possible orientation workspace (axes in radians)

Figure 3.16: Workspaces of hand described in tables 2.1, 3.5 and 3.6 manipulating a ball of radius 17.5 mm

male subjects, 1 female subject and the Shadow hand (see [92]) respectively f with suffixes in the figure represent the quartic polynomials fitted to the obtained data sets as given in figures 3.14a and 3.14b. The models for the fingertips for the male subjects were scaled up/down from the model developed in section 2.4 and a separate model was obtained for the female subject's hand by the technique discussed in section 2.4. The finger models and hand dimensions for the Shadow hand were obtained from specifications laid down by the [92]. For each of the subjects values of $\mathcal{W}_R$ and $\mathcal{W}_S$ were obtained for 60 object sizes across 4 separate trials, their convergence was ensured and the combined data is plotted in figure 3.14. A summary of the results is given in table 3.8. From the table we can conclude that the human hand can attain the highest workspace for a spherical object when the object cross section area is *approximately* equal to the palm area. Figure 3.14c shows that the results in table 3.8 is independent of the bounds on the condition number imposed by us in equation (3.5).

Table 3.8: Means and standard deviations of r_{po} for the maximum workspace

Type of contact	$\bar{r}_{po}$	$\sigma(r_{po})$
Rolling type	$\bar{r}_{po} = 1.004$	$\sigma(r_{po}) = 0.0443$
S joint type	$\bar{r}_{po} = 1.016$	$\sigma(r_{po}) = 0.0761$

In summary, the human hand grasping an object has been modeled in two ways, namely point contact with friction and a contact allowing rolling without slipping. In both these situations, we model the three-fingered hand grasping a sphere as a hybrid parallel manipulator with six degrees of freedom and present an algorithm to obtain the well conditioned workspaces

Table 3.9: Sample finger and hand segment lengths (refer figure 3.17 for symbols)

Hand part	**Symbols**	**Values (mm.)**
Index finger	$\{l_{11}, l_{12}, l_{13}\}$	$\{35,\ 23,\ 28\}$
Middle finger	$\{l_{21}, l_{22}, l_{23}\}$	$\{41,\ 22,\ 28\}$
Thumb	$\{l_{31}, l_{32}, l_{33}\}$	$\{45,\ 36,\ 34\}$
Palm	$\{d, h\}$	$\{13,\ 82\}$

of the hybrid parallel manipulators. The main results are presented in section 3.3.7 and we show that the maximum workspace of the grasped object is obtained when the palm size is same as the grasped object size. We also show that the maximum workspace could be as much 1.5 times when rolling without slipping is allowed. In section 3.3.7.1, experimental results on dexterous manipulation are presented to set a perspective for the numerical experiments and to make them more realistic are described. It is shown that the experimental results agree reasonably well with numerical simulation results.

3.4 Example 2: Workspace of the Stanford-JPL hand

In this work, we study the Stanford-JPL hand (also known as the Salisbury hand), a well known six-degree-of-freedom multi-fingered hand, originally proposed by Salisbury[5], as a hybrid parallel manipulator. The work has been very well documented in robotics literature and has been extensively studied by researchers (see e.g. Ghosal [81]) etc. To start with, we define the well conditioned workspace of the manipulator by setting realistic constraints on the actuated and passive joints and by restricting the condition numbers of the equivalent Jacobians (relating the linear and angular velocities of the end effector separately with the joint rates) to be less than 100 at all times. Next, using the definition of the well conditioned workspace, we formulate the problem of obtaining the well conditioned workspace of the parallel manipulator as an integration problem in the task space (in $\Re^3$ for the linear component of the motion and in $SO(3)$ for the angular component of motion of the end effector).

3.4.1 Description of the manipulator and solution of inverse kinematics problem

The kinematic model shown in figure 3.17 represents a three-fingered hand grasping an object. The grasping of the object is assumed to be point contact with friction and thus modeled as spherical joints. The manipulator shown in and schematically in figure 3.17, is a parallel 6-DOF manipulator with a moving platform or the "gripped object" $\{S_1, S_2, S_3\}$ connected to a fixed base $\{B_1^0, B_2^0, B_3^0\}$ connected by three 3-R serial manipulators of link lengths $\{l_{i1}, l_{i2}, l_{i3}\}$

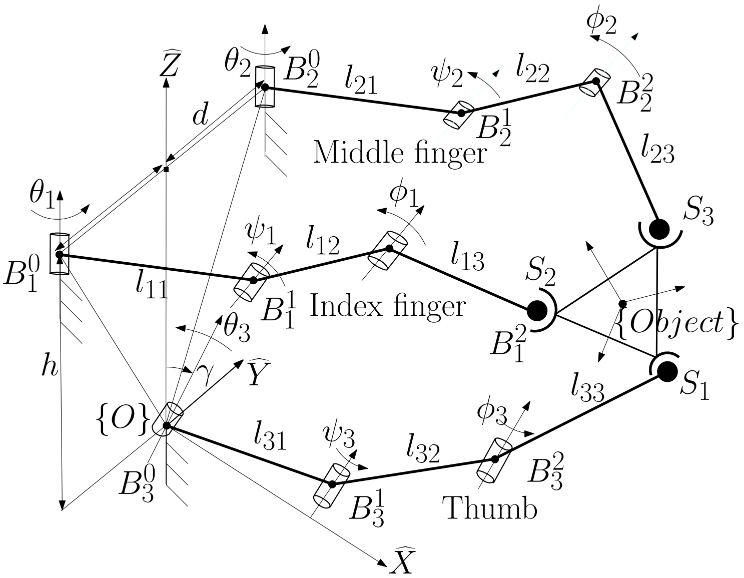


Figure 3.17: Schematic of the Salisbury hand (from Ghosal [81])

Table 3.10: Joint notations in figure 3.17 and maximal permissible motions

Joint center	**Joint variable**	**Type**	**Value/range**
B_1^0 and B_2^0	θ_1 and θ_2	Active	$-45°$ to $45°$
B_3^0	θ_3	Active	$-45°$ to $45°$
B_1^1 and B_2^1	ψ_1 and ψ_2	Active	$0°$ to $90°$
B_3^1	ψ_3	Active	$0°$ to $90°$
B_3^2	ϕ_3	Passive	$0°$ to $30°$
B_1^2 and B_2^2	ϕ_1 and ϕ_2	Passive	$0°$ to $30°$
B_3^0	γ	Fixed	$\gamma = 45°$
$S_1,\ S_2$ and S_3	$\{\xi_X^i, \xi_Y^i\}$	Passive	$\pm 45°$

$\forall i = 1, 2, 3$. The contacts between the gripped object and the distal ends (from the base) of the serial manipulators are modeled as 3 un-actuated "S" joints with three rotational degrees of freedom but no translational degree of freedom. It may be observed that the last "R" joint from the base towards the object shown by $\{B_1^2, B_2^2, B_3^2\}$ in figure 3.17 is un-actuated. Due to this un-actuated joint it not a true parallel manipulator and has been referred to as a hybrid manipulator. Also the thumb axis , shown here to be along the Z direction, need not be like that always, in fact in the analysis an inclination of 45° in the direction out of the plane has been considered. We can find the vectors from the base $\{O\}$ to the points of contact with the object $\{S_1, S_2, S_3\}$.

$$^{O}B_1^0 = \{0, -d, h\}^T;\ ^{O}B_2^0 = \{0, d, h\}^T;\ ^{O}B_3^0 = \{0, 0, 0\}^T \tag{3.37}$$

$$
{}^{O}S_i = {}^{Base}B_i + R[\hat{Y}, \gamma_i] \begin{bmatrix} \cos(\theta_i)\left[l_{i1} + l_{i2}\cos(\psi_i) + l_{i3}\cos(\psi_i + \phi_i)\right] \\ \sin(\theta_i)\left[l_{i1} + l_{i2}\cos(\psi_i) + l_{i3}\cos(\psi_i + \phi_i)\right] \\ l_{i2}\cos(\psi_i) + l_{i3}\cos(\psi_i + \phi_i) \end{bmatrix} \tag{3.38}
$$

$$
\forall i = \{1, 2, 3\};\ \gamma = [0, 0, \pi/4]
$$

Equation (3.38) along with the constraints imposed on the manipulator by the 3 spherical joints will be used to formulate and solve the inverse kinematics problem and the definition of the condition number of the Jacobian for the well conditioned workspace of the manipulator.

The inverse kinematics problem of the manipulator can be very simply stated and solved as follows. For a given position vector of the point S_1, (see figure 3.17) the expressions of the X, Y and Z coordinates of the point S_1 are given as the rows of equation (3.38). From which, by simplifying $X^2 + (Y + d)^2 + (Z - h)^2$ we can obtain the expression with only ϕ_1, given in equation (3.39).

$$
4l_{11}^2(l_{12}^2 + l_{13}^2 + 2l_{12}l_{13}\cos(\phi_1)) = C_1^2 + 4l_{11}C_2^2 \tag{3.39}
$$

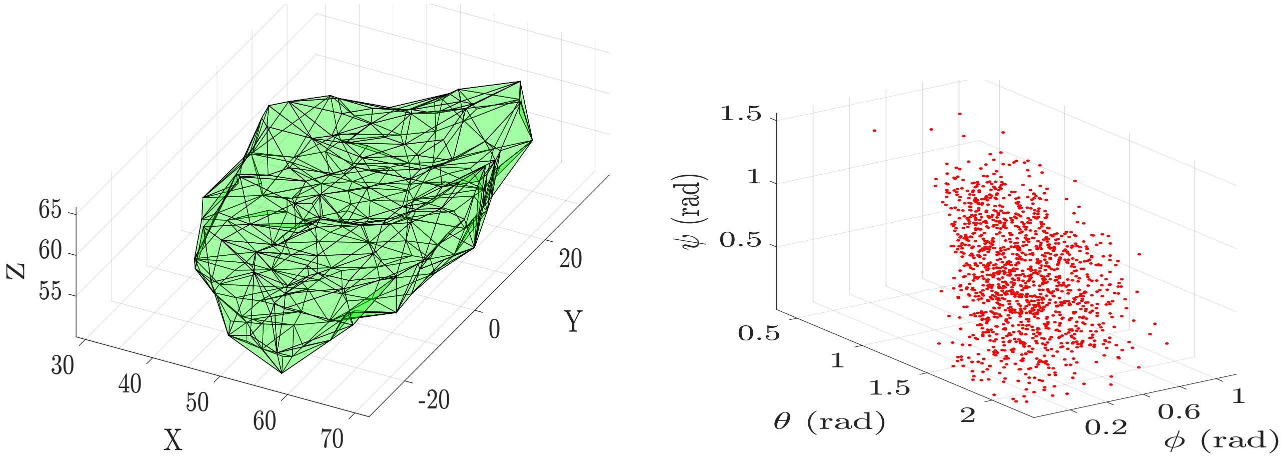
Where, in equation (3.39), $C_1 \equiv C_1(l_{11}, l_{12}, l_{13}, d, h, \phi_1)$ and $C_2 = h - Z$. Substituting $\cos(\phi_1)$ with the its tangent half angle equivalent in equation (3.39) we can obtain a quadratic expression for ϕ_1. ψ_1 can be solved from the eliminant obtained by using Sylvester's dialytic method and θ_1 is obtained as $\theta_1 = \mathrm{atan2(Y + d, X)}$. The inverse kinematics problem for the middle finger and the thumb can be solved in a way similar to index finger shown above.

Since the the manipulator described in section 3.3.1 bears resemblance to the Stanford-JPL hand, the expressions of the manipulator Jacobians can be obtained as shown in section 3.3.4. With the solution of the inverse kinematics and the expressions of the manipulator jacobians at hand, we can proceed to finding the well-conditioned workspace of the manipulator as described in section 3.2.

3.4.2 Workspace of the Stanford-JPL hand

Table 3.11: Means and standard deviations of r_{po} for the maximum workspace

Type of contact	$\bar{r}_{po}$	$\sigma(r_{po})$
S joint type	$\bar{r}_{po} = 1.043$	$\sigma(r_{po}) = 0.05$

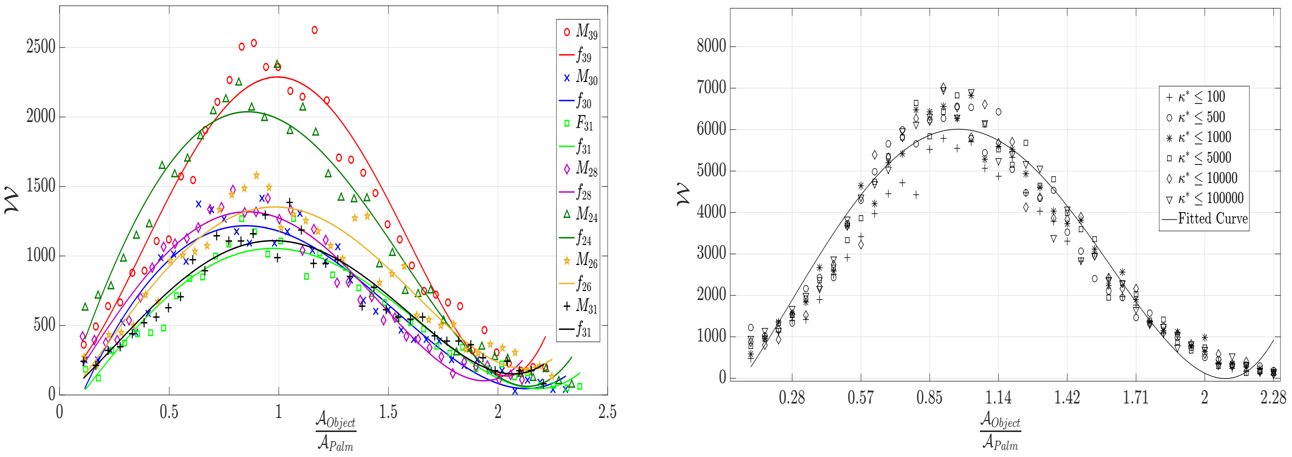


(a) Workspace of the Salisbury hand (b) Angular workspace of the Salisbury hand

Figure 3.18: Workspace of the Salisbury hand (Stanford-JPL hand)

The well-conditioned workspace of the manipulator, denoted by $\mathcal{W}$, is a quantity in $SE(3)$. Using the method described above, we can obtain separate representations of the workspace in $\Re^3$ and $SO(3)$. Figure 3.18a shows the representation of the workspace in $\Re^3$ as a triangulated domain enveloping the cloud of points inside the well-conditioned workspace. Figure 3.18b shows the well-conditioned workspace of the parallel manipulator in $SO(3)$ as a cloud of points. The dimensions of the hand segments were taken from table 3.5 and the object size (circum-radius of $\triangle S_1S_2S_3$ in figure 3.17) was taken as 20mm. The volume of the obtained workspace is $1.83 \times 10^3 \text{mm}^3$. Figure 3.19a shows the variation of the workspace of the hand across varying hand and object sizes. For this, we considered 7 data sets (like the ones shown in table 3.5) from the hand dimensions of 6 male and 1 female subjects. The horizontal axis in figures 3.19a and 3.20a denotes the quantity $r_{po} = \dfrac{\mathcal{A}_{Object}}{\mathcal{A}_{Palm}}$ where $\mathcal{A}_{Object}$ is the area of the circum-circle of $\triangle S_1S_2S_3$ and $\mathcal{A}_{Palm}$ is the area of $\triangle B_1^0B_2^0B_3^0$ in figure 3.17. It may be noted that the hand workspace is the largest when the area of the palm is approximately equal to the object area. The statistical details of the claim are given in table 3.8.

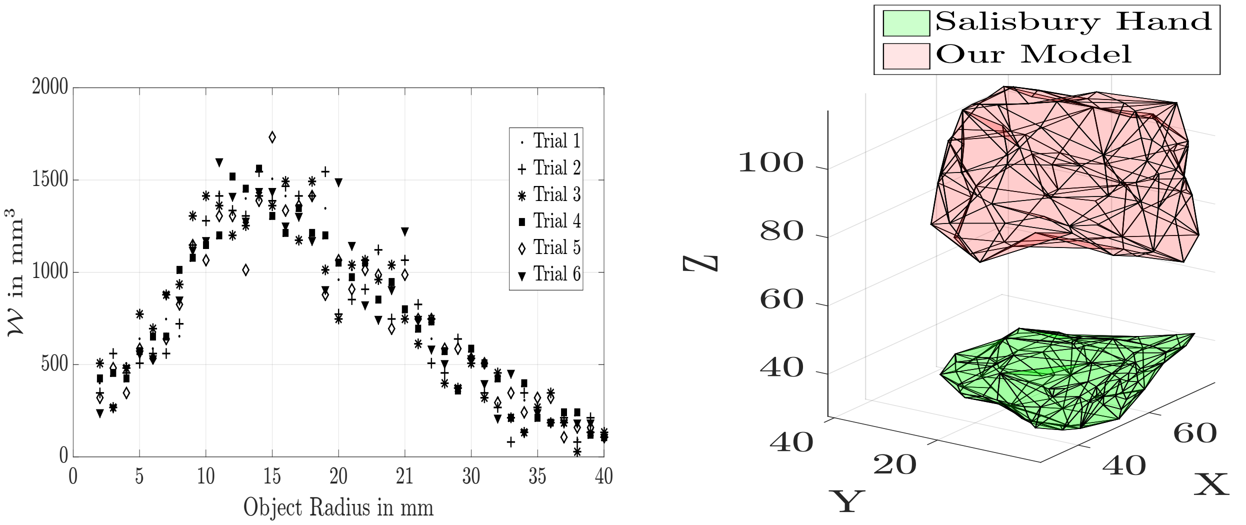
Since the method of obtaining the workspace is an iterative one, we demonstrate the convergence of our algorithm in figure 3.19b. Figure 3.20a demonstrates that the result presented in table 3.11 is independent of the upper bound on the condition number set in equation (3.5), to ensure we obtain a conservative representation of the well-conditioned workspace. In figure 3.20b we show that the workspace of the Salisbury hand using the joint limits prescribed in table 3.10 is smaller than the workspace we have obtained for the human hand.



(a) Variation of the workspace with varying object size

(b) Convergence of algorithm over 6 trials

Figure 3.19: Variation of the workspace with varying object sizes and convergence of the algorithm



(a) Independence of the result in figure 3.19a to upper bounds on κ^* (ball radius of 15.5 mm)

(b) Comparison of the workspace of the manipulator discussed in section 3.3 and the salisbury hand

Figure 3.20: Independence of the workspace to the limit on the condition number and comparison of the workspace of the Salisbury hand and the human hand model

3.5 Conclusion

In this chapter we have demonstrated the use of Monte Carlo method for obtaining the workspace of a manipulator. We started by describing the Monte Carlo method and discussing the advantages and disadvantages of using it to obtain the well conditioned workspace of a parallel manipulator. In the next two sections, we discussed the usage of the method for finding the workspace of 6 degrees of freedom parallel manipulators. The first example dealt with the workspace of the human 3 fingered grasp. The human hand grasping an object has been modeled in two ways, namely point contact with friction and a contact allowing rolling without slipping. In both these situations, we model the three-fingered hand grasping a sphere as a hybrid parallel manipulator with six degrees of freedom and present an algorithm to obtain the well conditioned workspaces of the hybrid parallel manipulators. The main results are presented in section 3.3.7 and we show that the maximum workspace of the grasped object is obtained when the palm size is same as the grasped object size. We have also shown that the maximum workspace could be as much 1.5 times when rolling without slipping is allowed. In section 3.3.7.1, experimental results on manipulating a grasped object are presented to set a perspective for the numerical experiments and to make them more realistic. It is shown that the experimental results agree reasonably well with numerical simulation results. In the next section we obtained the well conditioned workspace of the Stanford-JPL hand by the same method. With these two examples we could show that the Monte Carlo method can represent and quantify the workspace of a manipulator and in the next chapter we demonstrate how the Monte Carlo method can be used for optimal synthesis of parallel manipulator for a desired well conditioned workspace.

Chapter 4

Optimal synthesis of parallel manipulators using Monte Carlo method

4.1 Introduction

In this chapter we discuss the use of Monte Carlo method, in conjunction with gradient based optimization methods for optimal kinematic synthesis of parallel manipulators for a given workspace. We begin by posing the optimization problem, discussing the underlying procedure to carry out the gradient based optimization in section 4.2. Following which we provide two examples of optimal kinematic synthesis of parallel manipulators– in section 4.3.5 we discuss the optimal synthesis of a planar 5R manipulator and in section 4.4.6 we discuss the optimal synthesis of a Stewart platform manipulator. In these examples we demonstrate the sensitivity of the workspace of the manipulator to perturbations in the design constraints and finally in section 4.5 we discuss the sensitivity of the multi-fingered hand workspaces to perturbations in design constraints. We conclude the chapter by summarizing the main results.

4.2 Formulation and solution of the optimization problem

To attempt the problem of optimal synthesis for a manipulator, we need to start by posing an optimization problem with a twice differentiable and continuous objective function. However we note that barring a few cases like the 5-R planar manipulator, the planar 3-R$\underline{\text{R}}$R or the planar 3 R$\underline{\text{P}}$R manipulators, we do not have analytical expressions for the workspace. The workspace for the planar 5R manipulator is described by circular arcs and the expression of the workspace for the most general case, as given in equation (3.7), can be directly taken as the objective function. The situation is more complicated while we are dealing with optimal synthesis of higher degrees of freedom mechanisms. From the developments in chapter 3 we note that the Monte Carlo method provides a *conservative* estimate and representation of the workspace which is known to be not exact. Therefore arriving at a closed form of the objective function is not possible. However, we can get around the problem by assuming, without any loss of generality, a transformed position of the manipulator where the X axis passes through the entire workspace[1]. Now, considering a plane at $X = c$ we can obtain the closed level set curve which forms the boundary of the workspace at that plane. All the points on this curve are on the $Z - Y$ plane, and the square of the distance of any point on the curve from the origin on the plane ($Z, Y = 0$, and $X = c$) is given as $D = z^2 + y^2$, where z and y are the coordinates of the point on the center of end effector (for example the centroid of $\triangle S_1S_2S_3$ in figures 3.6 and 3.17). Maximizing the quantity D for all values of c will result in the maximization of the workspace.

The optimization problems of the 4 manipulators that we have studied in the thesis– the planar 5-R parallel manipulator, the 6 DoF approximation of the human hand, the Salisbury hand and the Stewart platform manipulator are all non-convex in nature. This means that there are multiple local optima in the parameter space of these manipulators and gradient based methods cannot be relied upon to converge to the global optimum[2]. For the planar 5R manipulator, the reason for the non-convex nature has been discussed in detail in section 4.3.5, for the multi fingered hands, the expression of D is identically dependent on the palm dimensions (d and h in figures 3.6 and 3.17) so the Hessian of the objective function is sub-rank and hence the problem becomes non-convex. The optimization problem of the Stewart platform manipulator in the configuration we have considered is known to be non-convex (see Lou et al. [48]) and hence we are not attempting to prove the same.

In the next section, we discuss the "*Karush-Kuhn-Tucker conditions*" or "KKT conditions" and provide a basic introduction to algorithm we have used to solve the optimization problem.

4.2.1 Interior point methods and KKT conditions

In this section we present a brief overview of the KKT conditions and interior point method, which have been used for solving the optimization problem for the parallel manipulators. For

---

[1] An example of the transformation would be the case when the origin is at the centroid of $\triangle B_1^0B_2^0B_3^0$ in figures 3.6 and 3.17 instead of B_3^0

[2] See e.g. the works by Lou et al. [48], [50], where the authors arrive at global optima by using controlled random search based methods.

a more detailed treatment on the subject, one may refer to the chapter 11 of the book by Boyd and Vandenberghe [74]. The general statement for a constrained optimization problem, with regards to optimal synthesis of parallel manipulators may be given as :

$$\begin{aligned} \underset{\mathbf{P}}{Maximize}\ \mathcal{W}(\mathbf{P}) \\ \text{Subject to, } h_i(\mathbf{P}) = 0\ \forall i = 1, 2, ..., m \\ g_j(\mathbf{P}) \leq 0\ \forall j = 1, 2, ..., n \end{aligned} \tag{4.1}$$

Where, $\mathcal{W}(\mathbf{P})$ is the workspace of the manipulator described as a function of the geometrical parameters $\mathbf{P}$ that uniquely describe the manipulator. h_i and g_j are m equality and n inequality constraints specified by the designer. We assume that the problem is feasible, i.e. in the feasible parameter space $\mathcal{P}$, $\exists \mathbf{P}^* \in \mathcal{P}$ which is *an* optimum for the manipulator. Therefore, for the feasible optimization problem, we write the necessary and sufficient conditions for finding and optimum, also known as the KKT conditions as given in equation (4.2).

$$\begin{aligned} h_i(\mathbf{P}^*) = 0 \quad i = 1, 2, ..., m \\ g_j(\mathbf{P}^*) \leq 0 \quad j = 1, 2, ..., n \\ \lambda_j^* \geq 0 \quad j = 1, 2, ..., n \\ \lambda_j^* g_j(\mathbf{P}^*) = 0 \\ \nabla_{\mathbf{P}} \mathcal{W}(\mathbf{P}^*) + \sum_{i=1}^{m} \mu_i^* \nabla_{\mathbf{P}} h_i(\mathbf{P}^*) + \sum_{j=1}^{n} \lambda_j^* \nabla_{\mathbf{P}} g_j(\mathbf{P}^*) = 0 \end{aligned} \tag{4.2}$$

Where, in equation (4.2), λ_j* and μ_i^* are the Lagrange multipliers at the optimum $\mathbf{P}^*$. Interior point methods solve the optimization problem by applying Newton's method to the KKT conditions in equation (4.2). The solver associated with `fmincon` in Matlab[57] uses the *barrier method*– a type of interior point method to solve the problem in equation (4.2). We also note that, in our case, the Monte Carlo methods just gives a scalar value of $\mathcal{W}(\mathbf{P})$ in equation (4.1) therefore, the gradients and the Hessian of the objective function have to be calculated numerically.

4.2.2 Design sensitivity analysis

To understand the sensitivity of the objective function to slight changes in the constraints, we introduce small perturbation $\{\mathbf{b}_i\}$ in the neighborhood of 0 for the equality constraint and $\{\mathbf{e}_j\}$ for the inequality constraints in equation (4.1), noting that both $\{\mathbf{b}_i\} \longrightarrow 0$ and $\{\mathbf{e}_j\} \longrightarrow 0$

i.e. are sufficiently small.

Clearly, the new optimum point of problem in equation (4.1), **P'** has changed from the original solution $\mathbf{P}^*$ and now depends on the infinitesimally small perturbations about zero. Therefore, we may write, $\mathbf{P'} = \mathbf{P'}(\mathbf{b},\mathbf{e})$ and the new cost function value also depends on the value of the perturbations, however there is little merit in finding the same. Assuming that $\mathcal{W}(\mathbf{P}), h(\mathbf{P})$ and $g(\mathbf{P})$ have two continuous derivatives, and the optimal solution $\mathbf{P}^*$, the Lagrange multipliers μ^* for the equality constraints and λ^* for the inequality constraints satisfy all the necessary and sufficient KKT conditions (equation (4.2)), we can state the sensitivity theorem after Arora [112] as,

If for each $g(\mathbf{P'})$ it is true that $\lambda_i^s > 0$, then the solution of the system $\mathbf{P}' = \mathbf{P'}(\mathbf{b},\mathbf{e})$ of the modified system is a continuously differentiable function in the neighborhood of $\{\mathbf{b}\} = 0$ and $\{\mathbf{e}\} = 0$. Also,

$$\frac{\partial \mathcal{W}}{\partial b_i} = \frac{\partial \mathcal{W}\left(\mathbf{P}'(0,0)\right)}{\partial b_i} = -\lambda_i \forall i = 1,2,...m \tag{4.3}$$

$$\frac{\partial \mathcal{W}}{\partial e_j} = \frac{\partial \mathcal{W}\left(\mathbf{P}'(0,0)\right)}{\partial e_j} = -\mu_i \forall j = 1,2,...n \tag{4.4}$$

Therefore, from the above equations (4.3) and (4.4) and considering that $\{\mathbf{b},\mathbf{e}\}$ are sufficiently small, we can further write an expression for the perturbed value (first order change) of the cost function as,

$$\mathcal{W}(b_i,e_j) - \mathcal{W}^*(0,0) = \partial\mathcal{W} = -\{\lambda_i b_i + \mu_j e_j\}, \forall i = 1,2,...m, j = 1,2,...n \tag{4.5}$$

Where $\mathcal{W}^*$ is the value of the original cost function before perturbation. The above outcomes of the theorem are stated without a proof. For a more rigorous treatment one can refer to the textbook by Arora [112]. For evaluating equation (4.5) we need to extract the Lagrange multipliers at the optimal point. We can easily get them while using `fmincon` in Matlab [57]. Design sensitivity analysis is only possible because we are using gradient based methods for optimization. This feature lets us analyze the sensitivity of the manipulator workspace to all the manipulator dimensions, put forth as constraints, which is an added advantage. Interval analysis based methods like the ones by Hao and Merlet [26] propose the use of a threshold quantity ϵ to accommodate manufacturing errors that might creep in and result in a non-optimal manipulator, however, it does not tell anything about the effects of *each* of the individual design parameters to the workspace of the manipulator.

In the next two sections we will use the methodology developed above to perform dimensional

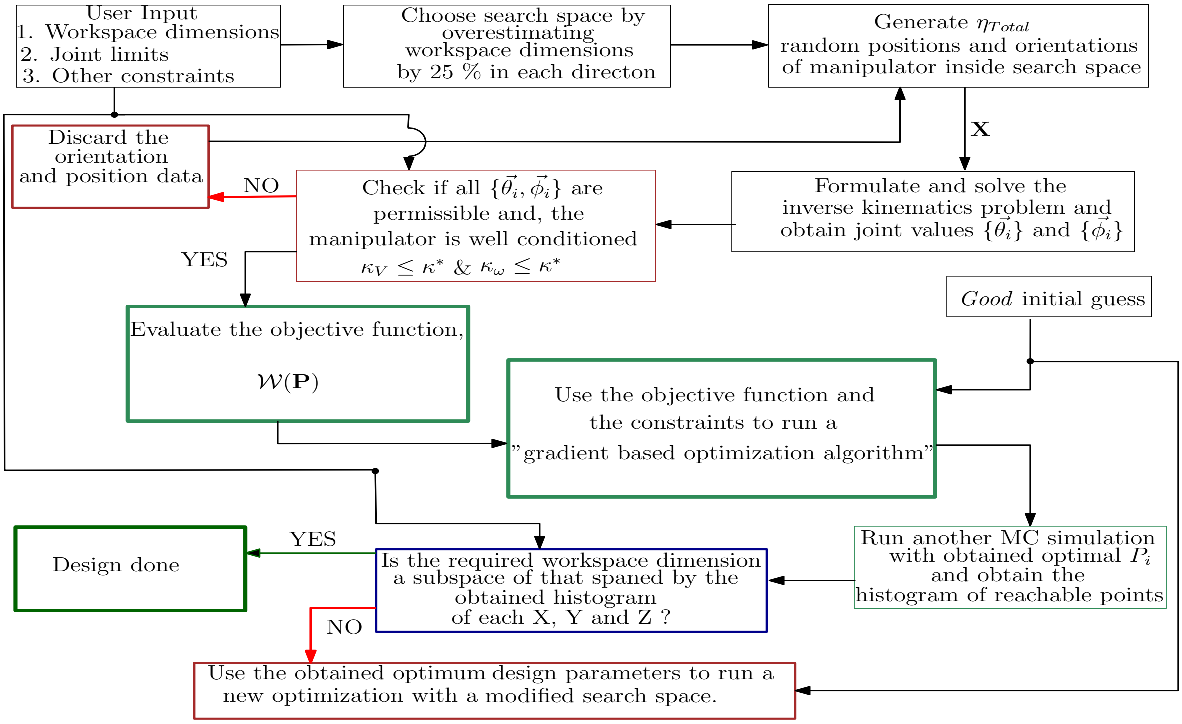


Figure 4.1: Block diagram of the design process

synthesis for two parallel manipulators. In the first example, we choose the earlier described planar 5 bar closed-loop mechanism with two degrees-of-freedom and obtain the dimensions for the largest well-conditioned workspace. In the second example, we perform dimensional synthesis for the well-known semi-regular Stewart platform manipulator. The optimization procedure is outlined in figure 4.1.

4.3 Example 1: Optimal synthesis of a 5 bar planar closed-loop mechanism

The workspace of a 5R two-degree-freedom closed-loop mechanism can be thought of as intersection of the workspace of two planar 2R manipulators. The workspace of a planar 2R manipulator, in general, is a a hollow circular disk with an inner and outer radius. Depending on the inner and outer radius for each of the two disks, the shape of the workspace of the 5R mechanism can be of four generic types with bounding circular arcs whose equations can be easily obtained from the equations given in section 3.2.1 (see also the works by Macho et al. [84], Cerventes-Sanchez et al. [83], and Liu et al. [20]). To make the optimization problem realistic, we impose generic constraints on the rotations at the joints and the link lengths. The generic constraints are as given in table 4.1. The constraint on the last column has been used so that all the link lengths are of the same order and some special optimum cases with one link length equal to zero can be avoided. This special case of a symmetrical manipulator with $d = 0$, was considered by Liu et al. [20], where a considerably large annular workspace was obtained. In this example, we perform dimensional synthesis for a general planar 5R manipulator for four generic workspace shapes with the above mentioned constraints. We use equation (3.7) along with geometric constraints to form the objective function and use a gradient based optimization method to analyze the four different workspace configurations of the 5R planar manipulator. It was observed that the optimization problem is non-convex, and a probable reason for its non-convexity is given in section 4.3.5.

Total length of links	Constraints on joint rotations	Constraints on link lengths
10 units	$\theta_j = [0, 2\pi]\ \forall j = 1, ..., 5$	$l_i \geq 1\ \forall j = 1, ..., 5$ unit

Table 4.1: Generic constraints for the manipulator

From Algorithm 1, the equivalent definition of workspace is $\mathcal{W} = \frac{\eta_{Posible} A}{\eta_{Total}}$. For a *unit* rectangle *completely* enveloping the possible workspace denoted by the plain zone and excluded workspace as the hatched zone.

4.3.1 Case I: Workspace bounded by 2 continuous circular arcs

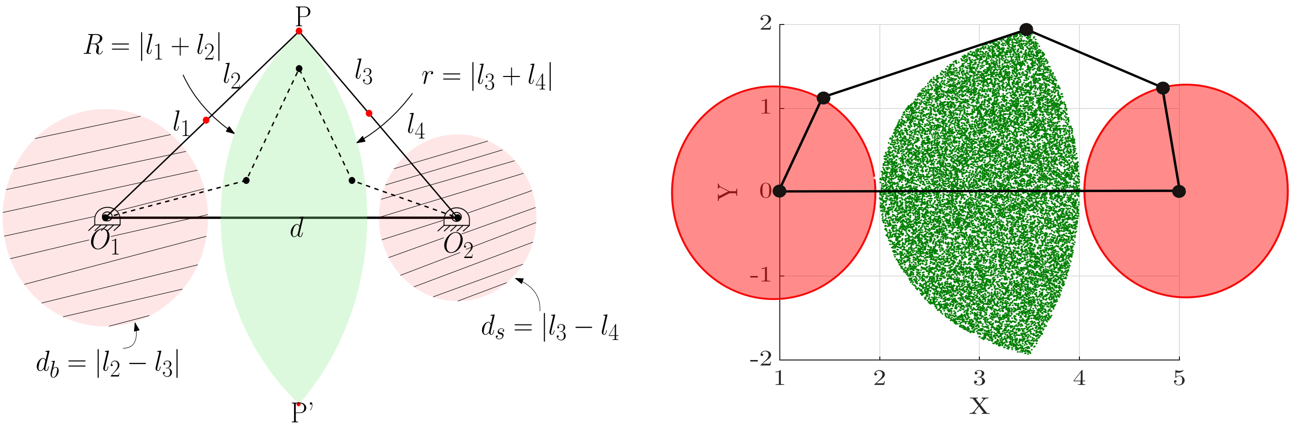
In this case, the workspace is bounded by two continuous circular arcs placed between the two centers O_1 & O_2, as shown in the figure 4.2a. For this case, the constraints are,

$$\begin{aligned} l_1 + l_2 - d &< 0 \\ l_3 + l_4 - d &< 0 \\ d_s &< d - (l_1 + l_2) \\ d_b &< d - (l_3 + l_4) \end{aligned} \tag{4.6}$$

The result of the optimization problem with the constraints listed in table 4.1 and in equation (4.6) yielded the results given in table 4.2. The maximum workspace is shown in figure 4.2b.

In this case we observe the following:

- Both the solutions given in table 4.2 are the same, i.e., one is the reflection of the other about the perpendicular line form P to O_1O_2 in figure 4.2a.



(a) Generic workspace shape in Case I

(b) Maximum workspace in Case I

Figure 4.2: Workspace bounded by two continuous circular arcs

Initial guess for {L}	Optimum value for {L}	Workspace area in unit^2
[4,1,3,3,1]	[4,3,1,1,1]	5.6123
[4,3,3,3,3]	[4,1,1,1,3]	5.4612

Table 4.2: Optimal solutions for Case I

- Both the cases yield about the same workspace area but for the second case the workspace is the reflection of the plain bounded zone in (figure 4.2b) about the line $x = 4$.

This case has been analytically solved in section 2.3.

4.3.2 Case II: Workspace bounded by 2 circular arcs outside and 2 circular arcs inside

The shape of the workspace is as shown in figure 4.3a. For this case, the general constraints are

$$\begin{aligned} l_1 + l_2 - d &> 0 \\ l_3 + l_4 - d &> 0 \\ d_s &< d - (l_1 + l_2) \\ d_b &< d - (l_3 + l_4) \end{aligned} \tag{4.7}$$

Another constraint on d_b and d_s can be added to the problem to have two disjoint circles centered at O_1 and O_2.

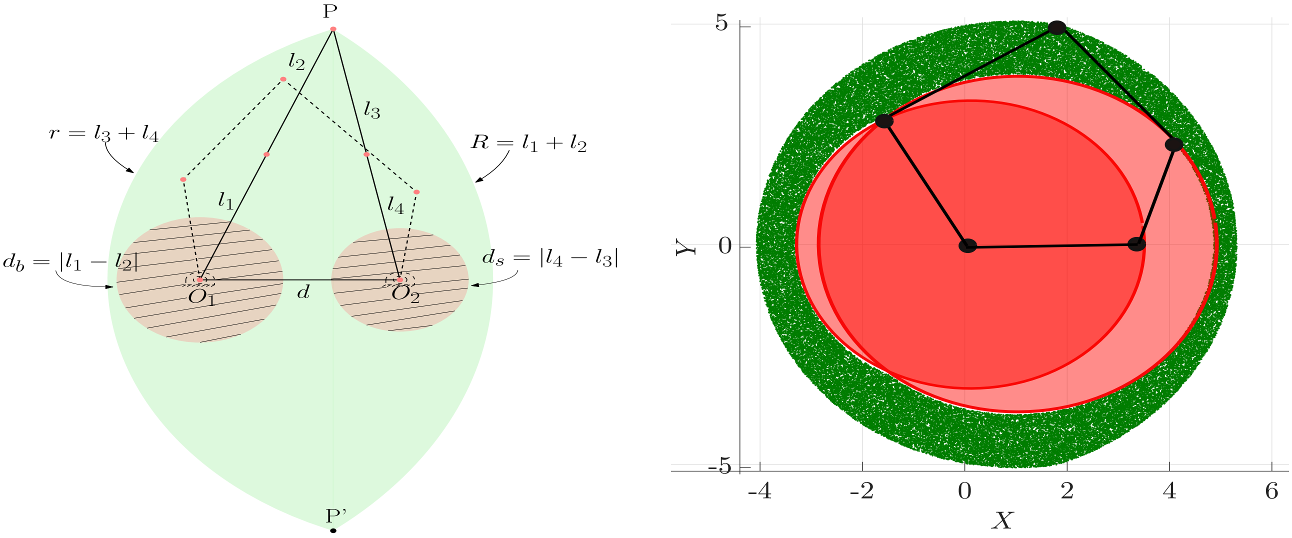
$$l_1 - l_2 + l_4 - l_3 - d < 0 \tag{4.8}$$

The result of the optimization problem attempted with general constraints from table 4.1, the constraints from equation (4.7) and the additional constraint form equation (4.8) yielded the results given in table 4.3.

Initial guesses for {L}	Optimum values for {L}	Workspace area in unit^2	Constraints
[3, 3, 2, 2, 3]	[1.00, 1.00, 4.31, 2.68, 1.00]	25.04	Eq. 4.7
[3, 3, 2, 2, 3]	[1.25, 3.74, 1.05, 1.26, 2.68]	24.95	Eqs. 4.7 & 4.8

Table 4.3: Optimal solutions for Case II

The maximum closed and bounded workspace obtained by using the proposed optimization algorithm is shown in figure 4.3b. In this case, we can see that as we increase the number of constraints, the usable workspace decreases.



(a) Generic workspace shape in Case II

(b) Maximum workspace in Case II

Figure 4.3: Workspace bounded by 2 circular arcs outside and 2 circular arcs inside

4.3.3 Case III: Workspace bounded by 4 circular arcs

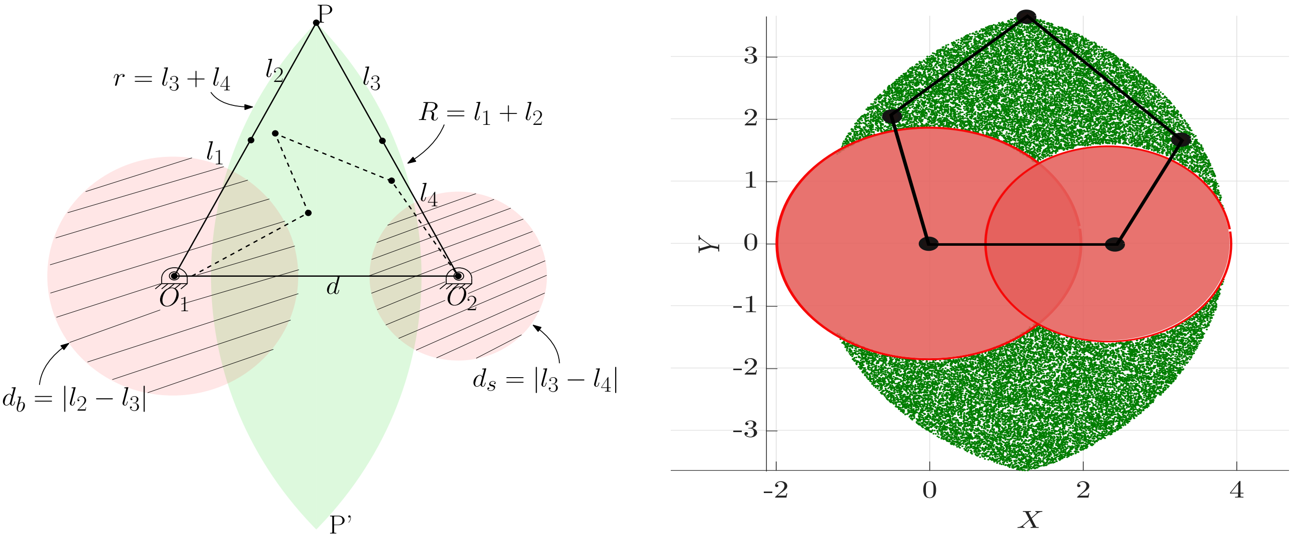
In this case the shape of the workspace is as shown in figure 4.4b. For this case, the general constraints are

$$\begin{aligned} l_1 + l_2 &< d_s + d \\ l_3 + l_4 &< d_b + d \\ l_1 + l_2 &> d - d_s \\ l_3 + l_4 &> d - d_b \end{aligned} \tag{4.9}$$

Initial Guesses for $\{L\}$	Optimal values for $\{L\}$	Workspace area in unit^2
[2.85, 2.63, 0.93, 0.93, 2.63]	[2.33, 2.87, 1.00, 1.12, 2.67]	12.74
[2, 2.5, 1.5, 1.5, 2.5]	[1.75, 2.14, 1.37, 1.37, 3.34]	22.94

Table 4.4: Optimal solutions for Case III

The maximum workspace obtained after optimization is shown in figure 4.4b and obtained link lengths are given in table 4.4. It maybe mentioned that the initial guesses used are from the work by Cerventes-Sanchez et al. [83], where this case has been analyzed.



(a) Generic workspace shape in Case III

(b) Maximum workspace in Case III

Figure 4.4: Workspace bounded by 4 circular arcs

4.3.4 Case IV: Workspace bounded by 3 circular arcs with a circular exclusion zone

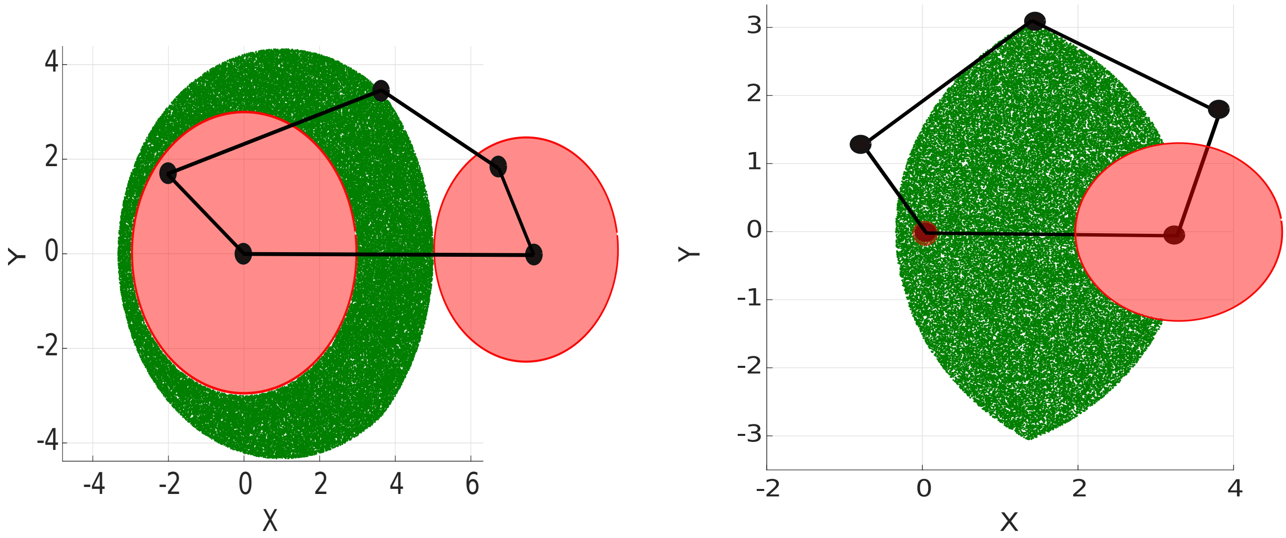
In this section, the schematic diagram of the workspace is shown in figure 3.2. The constraints for the case are as follows:

$$\begin{aligned} l_1 + l_2 &> d + d_s \\ l_3 + l_4 &< d + d_b \\ d_s &< l_1 + l_2 - d \\ d_b &> l_3 + l_4 - d \end{aligned} \tag{4.10}$$

The results of the optimization problem, after using the constraints form equation (4.10) are given in table 4.5. The two optimum workspaces for the manipulator are given in figure 4.5.

Initial Guesses for $\{L\}$	Optimal values for $\{L\}$	Workspace area in unit^2
[2.85, 2.63, 0.93, 0.93, 2.63]	[1.00, 4.00, 1.00, 1.82, 2.17]	28.63
[2, 2.5, 1.5, 1.5, 2.5]	[3.26, 1.73, 1.63, 1.63, 1.73]	13.56

Table 4.5: Optimal solutions for Case IV



(a) Generic workspace shape for Case IV

(b) Maximum workspace for result # 2 in table 4.5

Figure 4.5: Workspace bounded by 3 circular arcs with a complete circular exclusion zone.

As it is clear from the above 4 examples, the objective function, unlike general optimization problems, is related to the constraints we choose for a given case. This effect is well known for the simple case of a 5R planar manipulator. In fact, the optimization problem can be analytically shown to be a non-convex one. We attempt to demonstrate the same in the next section.

4.3.5 Proof of non-convexity of the optimization problem for 5R mechanism

The general statement for the constrained optimization problem of 5 R manipulator may be given as :

$$\underset{\{\mathbf{L}\}}{\text{Min}} \mathcal{W}(\mathbf{L}) \tag{4.11}$$

$$\text{Subject to, } E_1 : d + l_1 + l_2 + l_3 + l_4 - L_{Total} = 0$$
$$Ie_1 : l_3 + l_4 - d + d_b < 0$$

where L_{Total} is the total length of all the links combined. The inequality constraint Ie_1 was chosen so that the point P is always on the right side or near O_2 in figure 3.2 and the formula for workspace in (equation (3.7)) is always valid. The problem is known to be feasible because it has been successfully attempted numerically (see for example Huang [113]). In this section we show that it is a non-convex problem.

The Lagrangian formulation of the problem with the constraints is given as,

$$\mathcal{L} = \mathcal{W}(\mathbf{L}) + \lambda E_1 + \mu I e_1 \tag{4.12}$$

Following Boyd and Vandenberghe [74], we use the second-order condition for testing non convexity. The condition states that the function $\mathcal{L}$ in equation (4.12) is strictly convex if the Hessian $[\mathcal{H}(\mathcal{L})]_{i,j} = \dfrac{\partial^2 \mathcal{L}}{\partial L_i \partial L_j}$ of the function $\mathcal{L}$ is *positive definite*. For our case, the Hessian matrix $\mathcal{H}(\mathcal{L})$ turns out to be rank deficient and hence is not positive definite but at best positive semi-definite. This is further validated by the negative determinant value for the second principal sub matrix, or the first 2×2 sub matrix on top left in our case. For a simpler choice of objective function as stated before, the matrix is still rank deficient. We present the findings in table 4.6. The data for the design variable $\{\mathbf{L}\}$ has been used from [83] for the first two cases and from [113] for the last case. The cause of rank deficiency is equivalent dependence of the objective function on two design variables, i.e., at a point $\mathbf{P} = \{d^P, l_1^P, l_2^P, l_3^P, l_4^P\}$ in parametric space the gradient of the objective function in equation (4.12) does not have unique components in each of the parameter space directions. We have

$\{\mathbf{L}\} = [d, l_1, l_2, l_3, l_4]$	Cause of rank deficiency	Form of A from eq. 3.7
$\{12, 13, 4.5, 4.5, 13\}$	$\mathcal{H}(4,i)$=$\mathcal{H}(5,i)$	Full
$\{12, 13, 4.5, 4.5, 13\}$	$\mathcal{H}(2,i)$=$\mathcal{H}(3,i)$ & $\mathcal{H}(4,i)$=$\mathcal{H}(5,i)$	1^{st} 3 terms
$\{1.16, 1, 1, 1, 1\}$	$\mathcal{H}(4,i)$=$\mathcal{H}(5,i)$	Full

Table 4.6: Reason for rank deficiency of $\mathcal{H}$

$$\nabla \mathcal{L} = \frac{\partial \mathcal{L}}{\partial d}\Big|_{\mathbf{P}} \widehat{e_d} + \frac{\partial \mathcal{L}}{\partial l_1}\Big|_{\mathbf{P}} \widehat{e_{l_1}} + \frac{\partial \mathcal{L}}{\partial l_2}\Big|_{\mathbf{P}} \widehat{e_{l_2}} + \frac{\partial \mathcal{L}}{\partial l_3}\Big|_{\mathbf{P}} (\widehat{e_{l_3}} + \widehat{e_{l_4}}) \tag{4.13}$$

From equation (4.13) it is clear that the 4^{th} & the 5^{th} columns of the Hessian will be the same and hence the Hessian will be of lower rank.

It was also seen that the objective function in equation (4.11) given by equation (3.7) is dependent of the constraints in equation (4.11), which is not common for general optimization problems. However it is easy to see the dependence of the objective function with the constraints in this problem because of the planar nature and simple geometry of the manipulator. The constraint-objective function relationship is much difficult to visualize for a parallel manipulator with multi degree of freedom, principally because the geometry is complex and the direct kinematics problem is hard to solve.

4.3.6 Conclusions and sensitivity analysis

For this particular problem, we found that the largest workspace was obtained for *Case IV* with constraints in equation (4.10). However, there may be other special case with other particular geometric constraints which may increase the area further. Given the non-convex nature of the problem, we cannot exactly say which constraints we should choose. However if we perform a sensitivity analysis we can get some insight on the problem. The Lagrange multipliers for the constraints were extracted as the optimization algorithm converged to the result given in the first column of table 4.3. It is known (see chapter 4 of Arora [112]) that any positive perturbations to the constraints associated with the negative Lagrange multipliers will increase the value of the objective function from the value obtained at an optimum. Therefore, we can conclude the following:

1. The Lagrange multiplier for the equality constraint $l_1 + l_2 + l_3 + l_4 + d - L_{Total} = 0$ was obtained to be -3.877×10^3. The non-zero value of the multiplier indicates that this constraint was active and the large value with a negative sign indicates that the workspace area is very sensitive to this constraint and with increasing the value of L_{Total} the workspace would increase by a factor of 3.877×10^3 with other factors remaining same.

2. The only negative Lagrange multiplier is the one associated with the equality constraint. All other Lagrange multipliers are positive, and hence perturbations of those constraints might decrease the workspace area.

3. All the constraints in equation (4.7) remained inactive and hence the optimum value was not influenced by these constraints. Therefore the dependence of the problem on the geometry of the 5R mechanism is less and these constraints would serve as a good choice for a further complicated optimization problem with joint limit constraints.

4. The Lagrange multipliers for the constraints given in the last column of table 4.1 were large positive numbers. Therefore, the constraints preventing any difference in magnitude between the values of the link lengths were active. The largest positive Lagrange multiplier was associated with $d > 1$, $l_1 > 1$ and $l_4 > 1$.

4.3.7 Dimensional synthesis of a 5R mechanism

In all the cases studied above, the coordinates of O_1 in figures 3.2, 4.2a, 4.3a and 4.4a, has been chosen to be the origin and O_2 was fixed to lie on the X axis. As a result the workspaces obtained in figures 4.2b, 4.3b, 4.4b, 4.5a and 4.5b are all symmetric about the horizontal axis. It may be noted that to cross the symmetry axis, the 5R mechanism will have to go through a singular configuration. To avoid singularity, the desired workspace is chosen on one side of the X axis and for a required workspace, we can obtain the link lengths using the algorithm developed in this work. For the cases discussed, the search-space was assumed to be bounded by $X \in [-6, 6]$, $Y \in [-6, 6]$ and having an area of 144 unit^2.

Example 1: Required workspace: $X_d \in [2, 4]$, $Y_d \in [2, 4]$.
From the figures showing obtained workspaces, it is clearly seen that figure 4.5a includes most of the area of the area of the design space. The optimum link lengths for this example are found to be $[1.00, 4.00, 1.00, 1.82, 2.17]$ length units.

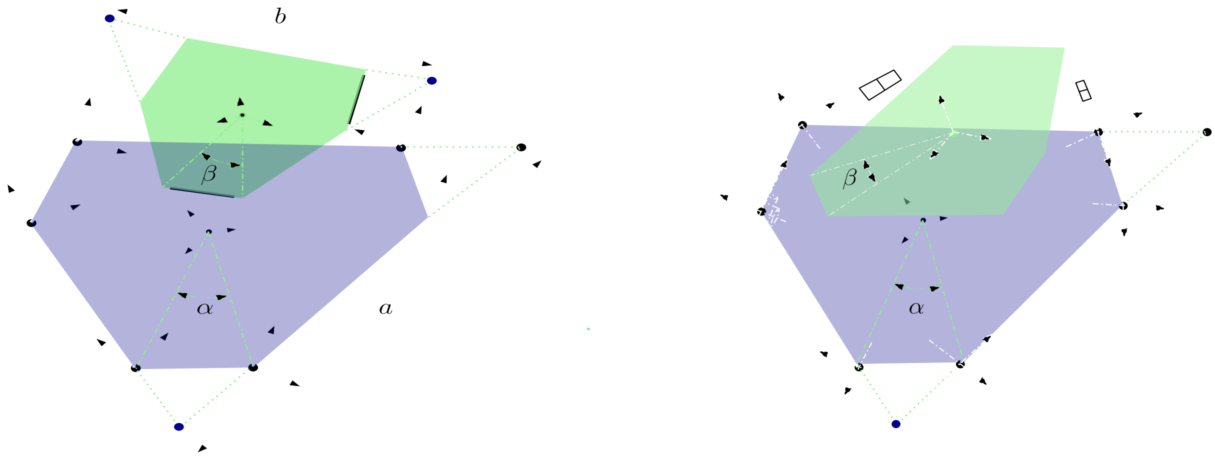
Example 2: Required workspace: $X_d \in [-4, 4]$, $Y_d \in [2, 4]$
For this we undertook the optimization procedure once more with a relaxed boundary of $X_s \in [-4.5, 4.5]$, $Y_s \in [1.5, 4.5]$. The following results given in table 4.7 were obtained with the binding constraints in table 4.1. It may be noticed that wanted workspace is symmetric about the Y axis therefore, the choice of the coordinates for O_1 at the origin is obvious. In case, required workspace is not symmetric the coordinate of O_1 can be set to the middle of range of X.

Case	Design parameters	Area
I	[1, 1, 4.31, 2.68, 1]	8.55
II	[2.24, 3.75, 1.04, 1.25, 2.70]	9.08
III	[1, 3, 1, 1, 4]	7.85
IV	[1, 4, 1, 1.82, 2.17]	10.62

Table 4.7: Results of the design problem

4.4 Example 2: Optimal synthesis of a semi-regular Stewart platform manipulator

The Stewart platform manipulator is a six degrees-of-freedom platform type parallel manipulator extensively studied by several researchers (see the review paper by Dasgupta and Mruthyun-



(a) SRSPM in the *normal* configuration (b) SRSPM in the *crossed* configuration

Figure 4.6: Two configurations of the SRSPM

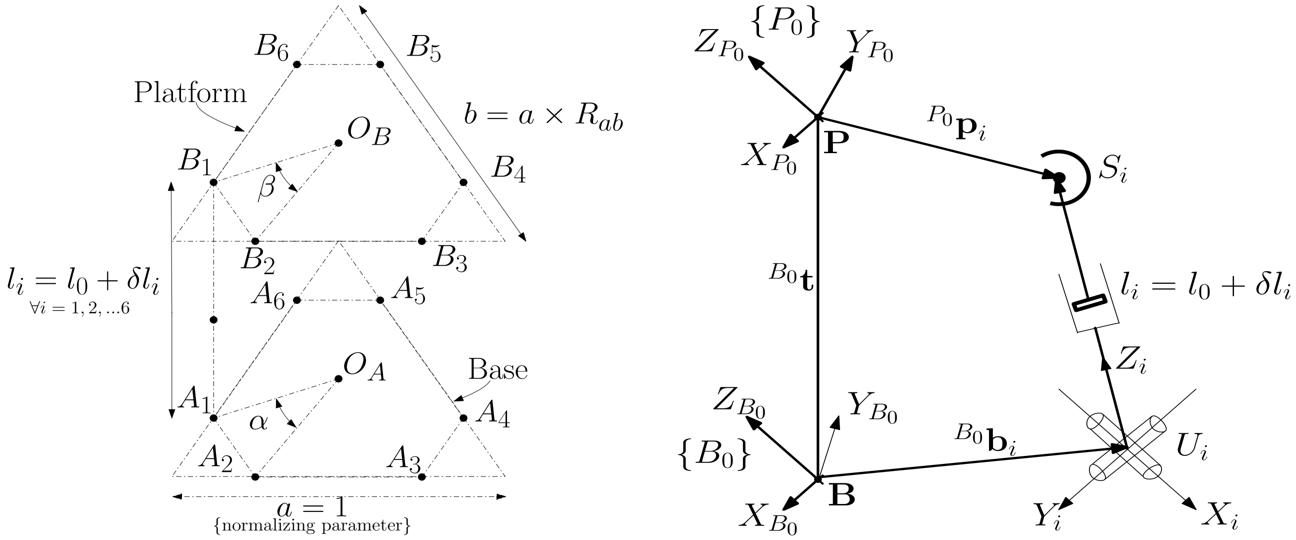
jaya [114] and the references contained therein). In this section, we attempt the optimization of a Stewart platform in a special configuration known as the Semi Regular Stewart Platform Manipulator or *SRSPM*, first proposed, analyzed and constructed by Fichter [115]. The design of the SRSPM for a desired workspace was attempted by Chatterjee and Ghosal [116], where the authors have used a predefined search path to search through the parameter space and attain an optimal result. Lou et al. [50] attempted the optimization of an SRSPM by using fewer parameters and a controlled random search (CRS) method to obtain the global optimum solution. We use the method presented in this work to address the following questions:

- What are the optimum design parameters for a SRSPM for the largest workspace in a given design space with joint constraints?
- How can one design SRSPM for a desired workspace?
- Which of the two configurations of SRSPM shown in figure 4.6 gives a larger workspace with joint limits? Or, in other words, which of the two configurations is better in terms of kinematics and workspace?

4.4.1 Description of the manipulator

As shown in figure 4.7a, in a SRSPM the top and bottom platforms are described by two equilateral triangles with the sides truncated before the vertices and forming an angle of α for the base triangle and β for the top triangle. The ratio of the side lengths of the manipulator platform and base triangles are given as $R_{ab} = \dfrac{b}{a}$, where a is the side of the platform triangle and b is the side of the base triangle. All 6 actuators for the manipulators are considered to be identical and have unextended length of l_0 and maximum possible extension of δl units. The SRSPM can be uniquely described the geometric parameters α, β, a, b, l_0 and δl. In the optimization, we have used R_{ab} since normalization, by considering $a = 1$, does not change the geometry of the SRSPM and reduces the number of variables by 1.

In figure 4.7b, the base frame is denoted by $\{B_0\}$ and the top platform frame is denoted by $\{P_0\}$. From a given distance ${}^{B_0}\mathbf{t}$ of the platform center P from the base center B, the vector from $\{B_0\}$ to any point P_i on the platform can be given as,



(a) Geometric description of the SRSPM

(b) Schematic diagram of a leg of the SRSPM

Figure 4.7: Geometric description and schematic diagram of the SRSPM

$$ {}^{B_0}\mathbf{p}_i = {}^{B_0}_{P_0}[R]\ {}^{P_0}\mathbf{p}_i + {}^{B_0}\mathbf{t} \tag{4.14} $$

From equation (4.14), the location of the i^{th} spherical joint from the base can be given as:

$$ {}^{B_0}\mathbf{S}_i = {}^{B_0}\mathbf{p}_i - {}^{B_0}\mathbf{b}_i $$

4.4.2 Inverse kinematics of the SRSPM

The solution to the inverse kinematics problem of the manipulator is well known and can solved in the following steps.

- Obtain input ${}^{B_0}\mathbf{t}$ and ${}^{B_0}_{P_0}[\mathbf{R}]$ from the random configuration generating subroutine.
- From the known geometry of the manipulator and figures 4.7a and 4.7b, obtain expressions for ${}^{P_0}\mathbf{p}_i$, ${}^{B_0}\mathbf{b}_i$ and $[R(\hat{\mathbf{Z}}, \alpha_i)]$, or the orientation of the i^{th} universal joint U_i with respect to $\{B_0\}$.
- The inverse kinematics equation are obtained as

$$[R(\hat{\mathbf{Z}}, \alpha_i)]^T \ [(x, y, z)^T -^{B_0} \mathbf{b}_i] = l_i \begin{pmatrix} \sin(\phi_i)\cos(\psi_i) \\ -\sin(\psi_i) \\ \cos(\phi_i)\cos(\psi_i) \end{pmatrix} \tag{4.15}$$

Using equation (4.15), we can obtain the actuated joint values, i.e., l_i. The U joint variables, ϕ_i and ψ_i, can also be solved from twelve constraints, 6 for the 6 S joints and 6 which ensures the planarity of the top platform points (see section 4.4.3). At this point, it may be noted that the expressions for the quantities ${}^{P_0}\mathbf{p}_i$, ${}^{B_0}\mathbf{b}_i$ and $[R(\hat{\mathbf{Z}}, \alpha_i)]$ will be different for the *normal* and *crossed* configurations of the manipulators and hence the constraints will also be different.

4.4.3 Constraints for an SRSPM and definition of equivalent Jacobian

The SRSPM is a six degree-of-freedom parallel manipulator. The six actuated joints are the prismatic joints in each leg and the six universal (U) and six spherical (S) joints are passive. As shown in Ghosal [81], we can derive 12 constraint equations which can be used to solve the angles ϕ_i and ψ_i $(i = 1, ..., 6)$ in the U joints. The first six constraint equations are derived from the fact that the distance between two consecutive S joint is fixed. We get

$$|{}^{B_0}\mathbf{p}_i -^{B_0} \mathbf{p}_{i+1}|^2 = |\mathbf{S}_i - \mathbf{S}_{i+1}|^2 \tag{4.16}$$

where S_i and S_{i+1} are the position vector of the two consecutive spherical (S) joints from the chosen origin. The second set of constraints ensure that a) the distance between two non-consecutive points on the platform, for example S_1 & S_3, are also fixed, and b) the diagonal lines connecting two pair of non-consecutive vertices, for example (S_1 & S_3) and (S_1 & S_4) are on the same plane, namely the plane of the top platform. These six constraints are given by

$$\left.\begin{aligned} |{}^{B_0}\mathbf{p}_1 -^{B_0} \mathbf{p}_3|^2 = |\mathbf{S}_1 - \mathbf{S}_3|^2 \\ |{}^{B_0}\mathbf{p}_1 -^{B_0} \mathbf{p}_4|^2 = |\mathbf{S}_1 - \mathbf{S}_4|^2 \\ |{}^{B_0}\mathbf{p}_1 -^{B_0} \mathbf{p}_5|^2 = |\mathbf{S}_1 - \mathbf{S}_5|^2 \end{aligned}\right\} \tag{4.17}$$

$$\left.\begin{array}{l} ({}^{B_0}\mathbf{p}_1 - {}^{B_0}\mathbf{p}_3) \times ({}^{B_0}\mathbf{p}_1 - {}^{B_0}\mathbf{p}_4) \cdot ({}^{B_0}\mathbf{p}_1 - {}^{B_0}\mathbf{p}_2) = 0 \\ ({}^{B_0}\mathbf{p}_1 - {}^{B_0}\mathbf{p}_4) \times ({}^{B_0}\mathbf{p}_1 - {}^{B_0}\mathbf{p}_5) \cdot ({}^{B_0}\mathbf{p}_1 - {}^{B_0}\mathbf{p}_3) = 0 \\ ({}^{B_0}\mathbf{p}_1 - {}^{B_0}\mathbf{p}_5) \times ({}^{B_0}\mathbf{p}_1 - {}^{B_0}\mathbf{p}_6) \cdot ({}^{B_0}\mathbf{p}_1 - {}^{B_0}\mathbf{p}_4) = 0 \end{array}\right\} \tag{4.18}$$

In may be noted that all the above 12 equations are only functions of the translation at the six actuated prismatic (P) joints variables denoted by $\mathbf{l}$ and the twelve (2 × 6) rotations at the passive U joints denoted by γ.

The twelve constraints equations can be differentiated and partitioned according to terms associated with actuated and passive variables, $\mathbf{l}$ and γ, to obtain

$$[\mathbf{K}(\mathbf{l}, \gamma)]\dot{\mathbf{l}} + [\mathbf{K}^*(\mathbf{l}, \gamma)]\dot{\gamma} = 0 \tag{4.19}$$

It is easily seen that $[\mathbf{K}^*]$ is a square matrix of dimension 12 × 12. Equation (4.19) can be solved for $\dot{\gamma}$, given $\det(\mathbf{K}^*) \neq 0$[1], and we can obtain

$$\dot{\gamma} = -[\mathbf{K}^*]^{-1}[\mathbf{K}]\dot{\mathbf{l}} \tag{4.20}$$

The position vector of the center of the platform in figure 4.7a is given by,

$${}^{O_A}P_{O_B} = \frac{1}{6}\sum_{i=1}^{3} {}^{O_A}B_i \tag{4.21}$$

and the orientation of the top platform with respect to the base can be written as

$${}^{B_0}_{P_0}[R] = \left[\begin{array}{ccc} \dfrac{{}^{O_A}B_1 - {}^{O_A}B_3}{|{}^{O_A}B_1 - {}^{O_A}B_3|} & \hat{Y} & \dfrac{({}^{O_A}B_1 - {}^{O_A}B_5) \times ({}^{O_A}B_1 - {}^{O_A}B_3)}{|({}^{O_A}B_1 - {}^{O_A}B_5) \times ({}^{O_A}B_1 - {}^{O_A}B_3)|} \end{array}\right] \tag{4.22}$$

where $\hat{Y}$ is obtained by the cross product of the third and first column of the matrix in equation (4.22).

By differentiating the expressions for the position and orientation of the end-effector obtained from equations (4.21) and (4.22) and partitioning the expression for actuated and passive joints, we have,

$${}^{B_0}V_{P_0} = [\mathbf{J}_V]\dot{\mathbf{l}} + [\mathbf{J}_V^*]\dot{\gamma} \tag{4.23}$$

---

[1] In the simulation, it was ensured that $\det(\mathbf{K}^*) \neq 0$ and the condition number of $\mathbf{K}^*$ was $\leq 10^4$ at all points inside the obtained workspace.

$$ {}^{B_0}\omega_{P_0} = [\mathbf{J}_\omega]\dot{\mathbf{l}} + [\mathbf{J}^*_\omega]\dot{\gamma} \tag{4.24} $$

Using equation (4.20) in equations (4.23) and (4.24) we obtain

$$ {}^{B_0}\mathbf{V}_{P_0} = ([\mathbf{J}_V] - [\mathbf{J}^*_V][\mathbf{K}^*]^{-1}[\mathbf{K}])\dot{\mathbf{l}} \tag{4.25} $$

$$ {}^{B_0}\omega_{P_0} = ([\mathbf{J}_\omega] - [\mathbf{J}^*_\omega][\mathbf{K}^*]^{-1}[\mathbf{K}])\dot{\mathbf{l}} \tag{4.26} $$

The matrices multiplying $\dot{\mathbf{l}}$ are the *equivalent* Jacobian matrices for the linear and angular velocity parts. These are given as

$$ \mathbf{J}^V_{eqv} = (\mathbf{J}_V - \mathbf{J}^*_V[\mathbf{K}^*]^{-1}[\mathbf{K}]) \tag{4.27} $$

$$ \mathbf{J}^\omega_{eqv} = (\mathbf{J}_\omega - \mathbf{J}^*_\omega[\mathbf{K}^*]^{-1}[\mathbf{K}]) \tag{4.28} $$

The equivalent Jacobian matrices are used to obtain the κ_V and κ_ω, the 2-norm condition numbers of $\mathbf{J}^V_{eqv}$ and $\mathbf{J}^\omega_{eqv}$.

4.4.4 Formulation and results of the optimization problem

The non-convex nature of the optimization problem of a SRSPM for the maximum well-conditioned workspace is known from literature (for example see the work by Chatterjee and Ghosal [116] and Lou et al. [50]) and hence, we attempt to find a local optimum by starting from a reasonable guess. The optimization problem for this case can be formulated as,

$$ \text{Minimize } \mathcal{W}(\alpha, \beta, R_{ab}, l_0, \delta l) \tag{4.29} $$

$$ \begin{gathered} \text{Subject to,} \quad a = 1 \\ l_i + \delta l_i = 1.8 \ , \ \forall i = 1, 2, ...6 \\ \tfrac{\pi}{12} < \alpha < \tfrac{\pi}{4} \quad \& \quad \tfrac{\pi}{12} < \beta < \tfrac{\pi}{4} \\ 0.5 < R_{ab} < 1.0 \quad \& \quad l_i > 0 \ \ \& \ \delta l_i > 0 \end{gathered} $$

Out of the three angles (θ_i, ζ_i, η_i) in a S joint, the range of η_i, measured about the vector $\overrightarrow{U_iS_i}$ in figure 4.7b is typically 0 to 2π. We restrict the ranges of the other two angles for a more practical design and the ranges are given in table 4.8. It may be noted that S joint values (θ_i and ξ_i) for the crossed configuration of the SRSPM are generally higher than that of the normal configuration because of the skewed arrangement of the legs. This was also noted by Fichter et al. [115] where the S joints were replaced by *gimbals* which provide significantly higher range of motion than conventional S joints.

Configuration	ϕ_i	ψ_i	θ_i	ζ_i
Normal	$\left[-\frac{\pi}{3}, \frac{\pi}{3}\right]$	$\left[-\frac{\pi}{3}, \frac{\pi}{3}\right]$	$\left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$	$\left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$
Crossed	$\left[-\frac{\pi}{3}, \frac{\pi}{3}\right]$	$\left[-\frac{\pi}{3}, \frac{\pi}{3}\right]$	$\left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$	$\left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$

Table 4.8: Un-actuated joint limits for SRSPM

In this example the starting guesses are the vectors given in equations (4.30) and (4.31) and the search space was chosen as $X \in [-2.5, 2.5]$, $Y \in [-2.5, 2.5]$ and $Z \in [1, 3]$ to capture the entire workspace of the SRSPM with normalized dimensions.

$$(\alpha, \beta, R_{ab}, l_0, \delta l) = \{1.2, 0.8, 0.3, 1, 0.8\} \text{ for normal configuration} \tag{4.30}$$

$$(\alpha, \beta, R_{ab}, l_0, \delta l) = \{0.8, 0.8, 0.36, 1.4, 0.4\} \text{ for crossed configuration} \tag{4.31}$$

The results of the optimization problem (equation (4.29)) is shown in table 4.9 below.

Configuration	Optimum design parameters	Workspace volume (unit^3)
Normal	{0.60, 0.5, 0.75, 1.44, 0.35}	2.030
Crossed	{0.76, 0.74, 0.726, 1.48 0.32 }	1.843

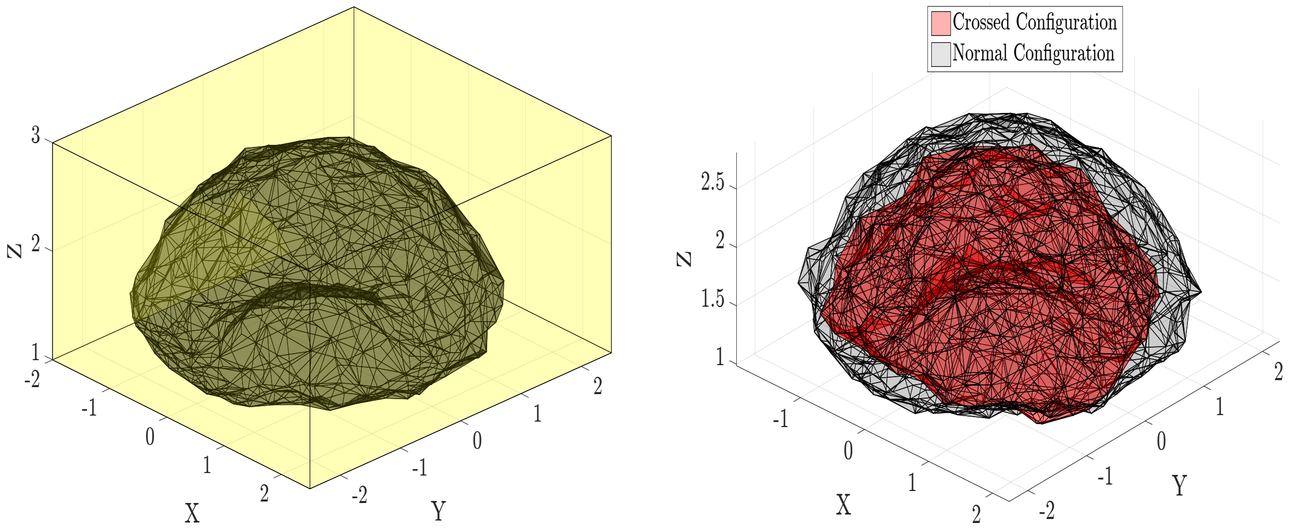
Table 4.9: Optimum design parameters for the SRSPM

Figure 4.8 shows the available workspaces for the SRSPM. In figure 4.8b a comparison of the available workspaces of the SRSPM in normal configuration (indicated in gray) and crossed configuration (in yellow) is given. As shown in table 4.9, the workspace of the manipulator is significantly less in crossed configuration than in the normal configuration. Additionally, from figure 4.8c one can observe that the workspace of the SRSPM is not continuous and there are ‘holes’ inside the well-conditioned workspace.

In equation (3.5) we have used an upper bound on the condition numbers, κ^* to be 100. Though the value 100 was arbitrarily chosen, similar results can be obtained for any chosen upper bound κ^*. Figure 4.9 shows the obtained workspace as the upper bound is increased. It can be seen that the chosen upper bound of 100 gives a conservative estimate and if the upper bound is increased we get a larger workspace.

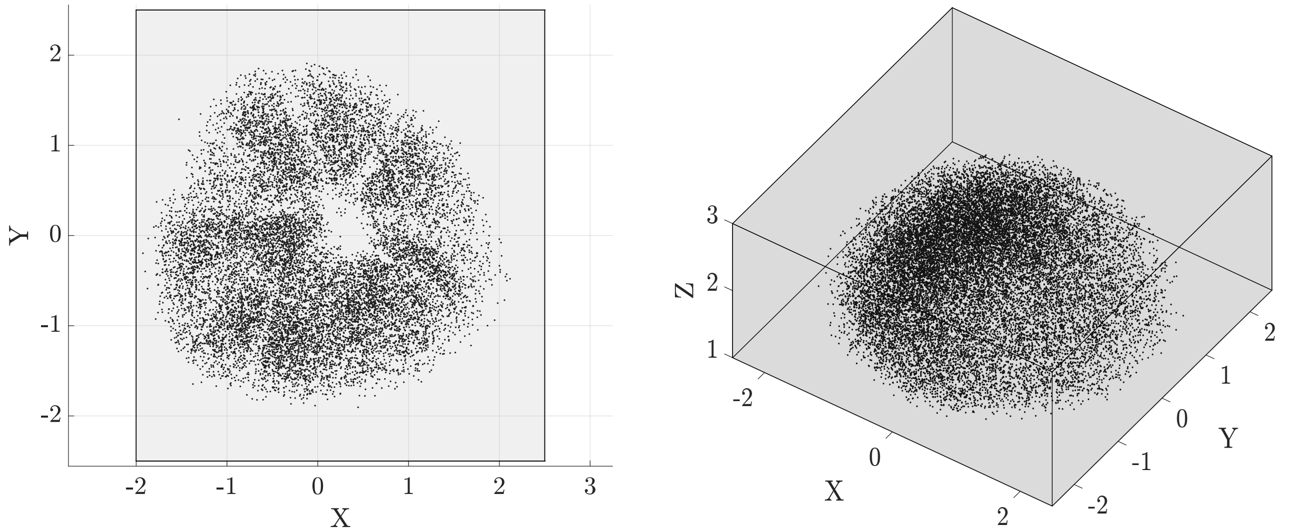
4.4.5 Sensitivity analysis and observations

The SRSPM has many more parameters when compared to the 5R example and the effect of the geometric constraints on the workspace volume is more difficult to obtain. The values of



(a) Available workspace for the normal configuration – outer box indicates search space

(b) Comparison between the available workspaces for normal and crossed configurations



(c) Workspace of the SRSPM in crossed configuration is not continuous (Top view)

(d) Workspace of the SRSPM in normal configuration is continuous

Figure 4.8: Workspaces of the optimum SRSPMs obtained for a given search space

the Lagrange multipliers for the problem have been extracted and plotted in figure 4.10 and we can make the following general observations.

- The values of Lagrange multipliers for the normal case are lesser than the crossed case by 4 orders of magnitude. This indicates that the optimization problem for the normal case was not much affected by the chosen constraints. However, the common theme is that the nature of the Lagrange multipliers for each case are the same, i.e., the equality

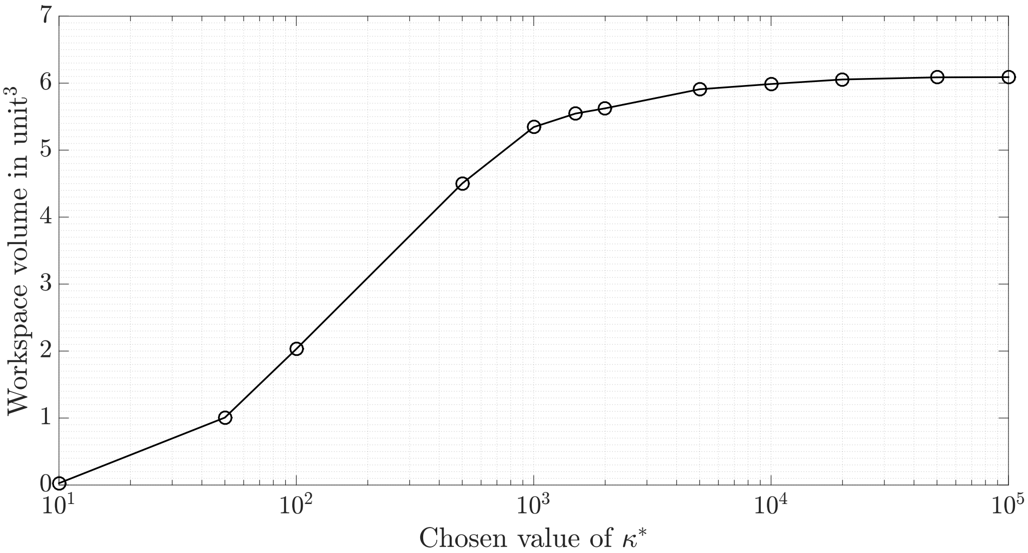


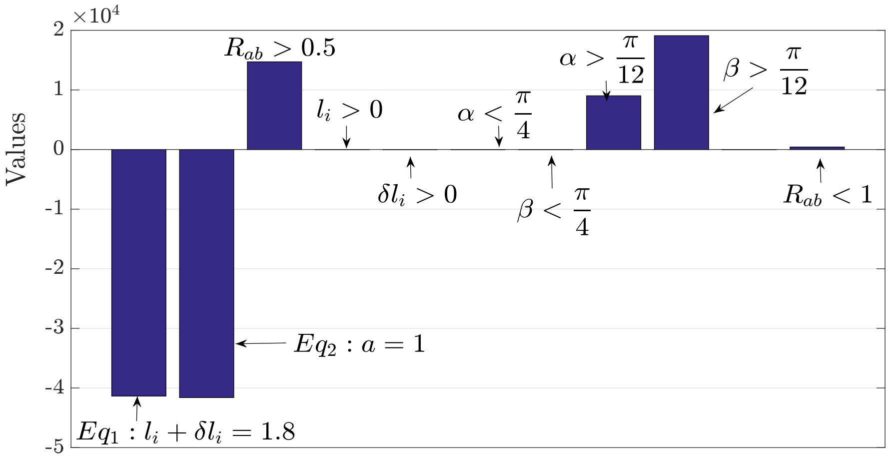
Figure 4.9: Dependence of well-conditioned workspace on chosen κ^* for the SRSPM in normal configuration

constraints have negative Lagrange multipliers and the inequality constraints have positive valued multipliers.

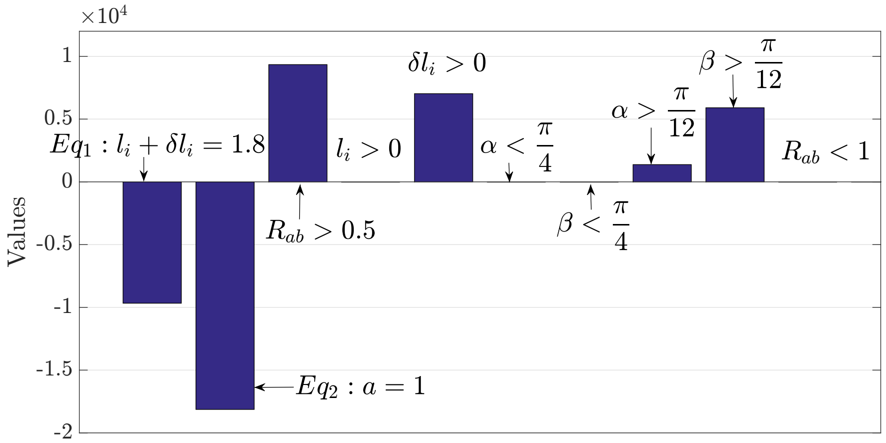
- In each case the equality constraints, i.e., $a = 1$ and $l_i + \delta l_i = 1.8\ \forall i = 1, 2, ..., 6$ bear a negative valued Lagrange multiplier. This indicates the obvious result that the larger manipulator with a larger actuator range will have a larger workspace.

- The Lagrange multiplier associated with the constraint $R_{ab} > 0.5$ remained positive for both the cases indicating that the constraint reduced the workspace volume value at the optimum. This constraint was used to ensure that the top platform is not smaller than half of the base.

- The constraints $l_i > 0$ remained inactive for both of the problems. This is consistent with the intuition that linear actuator should have a positive length. The constraint $\delta l_i > 0$ remained inactive for the normal configuration but had a high positive value for the crossed configuration. We recall that the equality constraint on the extension of the linear actuator is at most 80 % of the original length. Since at most of the feasible configurations of the crossed manipulators, the linear actuator is slanted, the points at the lower level (about the Z direction) of the search space were omitted since $\delta l_i > 0$ and used the particular un-actuated length of the linear actuators. The high positive value of the $\delta l_i > 0$ constraint for the crossed configuration of the manipulator is due to this reason.



(a) Lagrange multipliers for the normal configuration at *an* optimum



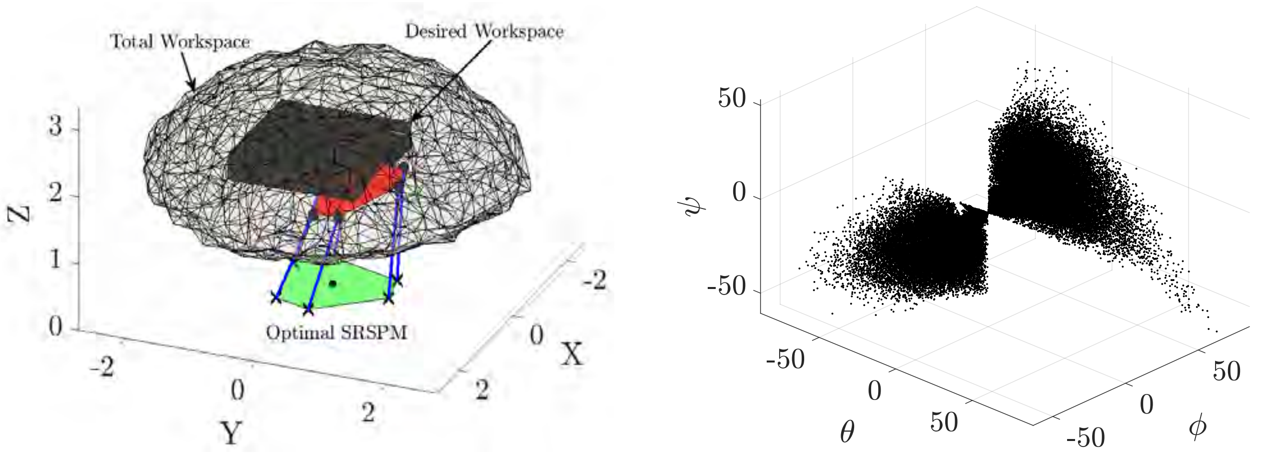
(b) Lagrange multipliers for the crossed configuration at *an* optimum

Figure 4.10: Lagrange multipliers for the optimization problem at *an* optimum

- The constraints on the upper limits of α and β were inactive but the constraints $\alpha > \frac{\pi}{12}$ and $\beta > \frac{\pi}{12}$ both had positive valued Lagrange multiplier. This finding suggests that a triangular top platform (also known as the 6-3 configurations) gives a larger workspace for both manipulator configurations. As documented in [115], Fichter has used a triangular top platform to analyze the SRSPM.

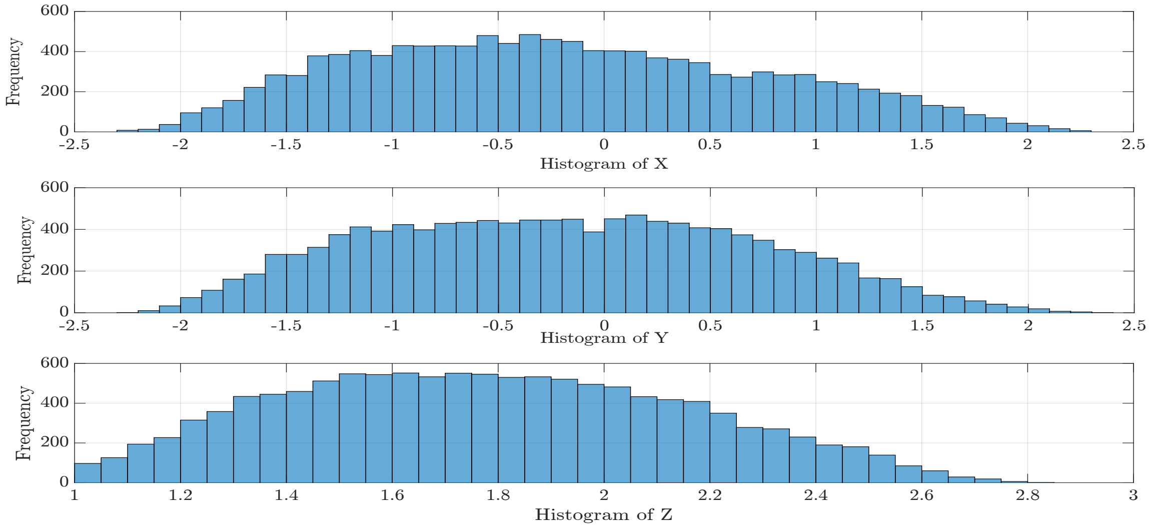
4.4.6 Dimensional synthesis of a SRSPM for a desired workspace

In this section, we attempt the problem of designing an optimal Stewart platform manipulator for a desired workspace by using the optimal dimensions we obtained (see table 4.9) in section 4.4.4. For a arbitrarily desired well-conditioned workspace of 1.5 units along X, 2 units along Y and 0.4 units along Z and a volume of 1.2 unit^3, the search space is chosen as



(a) Total workspace of the SRSPM

(b) Orientation workspace of the SRSPM (degrees)



(c) Histogram of the total workspace of the SRSPM

Figure 4.11: Workspaces and histogram of the SRSPM in *normal* configuration

$X \in [-2.5, 2.5]$, $Y \in [-2.5, 2.5]$ and $Z \in [1, 3]$ to capture the entire workspace of the SRSPM. The parameters giving the maximum workspace for both configurations are given in table 4.9. The histogram obtained for the optimal SRPSM, using a Monte Carlo simulation, is shown in figure 4.11c. The histograms show that the ranges in the normal configuration can be chosen as $X \in [-0.5, 1]$, $Y \in [-1, 1]$ and $Z \in [1.8, 2.2]$. Using the ranges, we run a separate Monte Carlo simulation and obtain the volume of the workspace as 0.87 unit^3 which is somewhat less than the desired workspace volume.

To obtain the desired larger volume, we observe from figure 4.11a that the workspace of the synthesized Stewart platform manipulator is *almost* equally distributed around the Z axis (also observed by Masory and Wang [39]). To increase the volume, we can modify the equality constraints involving a and $l_0 + \delta l$. From the sensitivity analysis in section 4.4.5, we recollect that the equality constraints $a = 1$ and $l_0 + \delta l_i = 1.8$, are associated with a negative Lagrange multiplier. Therefore, by introducing a as a design variable with a suitable upper and lower bound (to ensure that the optimization problem is bounded), a larger workspace may be obtained. To obtain the value of a which gives the desired volume of 1.2 unit^3, we relax the constraints a and $l_0 + \delta l_i$ to $a = 1.5$ and $l_0 + \delta l_i = 2.5$ and perform the optimization again with the values obtained above as initial guess. It was found that the desired workspace volume 1.2 unit^3 can be achieved from a new histogram where the ranges chosen are $X \in [-0.5, 1]$, $Y \in [-1, 1]$ and $Z \in [2.0, 2.4]$. This is shown in figure 4.11a. The parameters of the Stewart platform manipulator, $(\alpha, \beta, R_{ab}, l_0, \delta l, a)$, to obtain the desired workspace are $(0.713, 0.708, 0.802, 1.72, 0.78, 1.5)$, respectively.

Figure 4.11b shows the *orientation* workspace of the optimal manipulator at the center of the *well-conditioned* workspace shown in figure 4.11a. As shown in figure 4.7a, θ, ϕ and ψ are the Euler angles about X, Y and Z axes indicating the orientation of the top platform P with respect to the base B.

4.5 Design parameter sensitivity of Multi-fingered hands

Hand workspaces vary as the hand is manipulating objects of varied sizes. This is evident from our daily experiences and the same effect is also prevalent for multi-fingered robotics hands. This effect is demonstrated in figures 3.14 and 3.19a. We observe that the available workspaces for hands of different sizes vary widely and for manipulating the same object a *larger* hand has a larger workspace. To analyze the dependence of the human hand workspace volume on individual hand dimensions like finger lengths and palm area etc. we pose the following problem:

Problem To obtain the dimensions of the 95th percentile male human hand with the largest well conditioned workspace area while manipulating a spherical object of a given size.

Solution To solve the problem, we parametrize the hand (as shown in figures 3.6 and 3.17) with the dimensions given below

$$\mathbf{P} = \{d, h, l_{11}, l_{12}, l_{13}, r_m, r_t\} \tag{4.32}$$

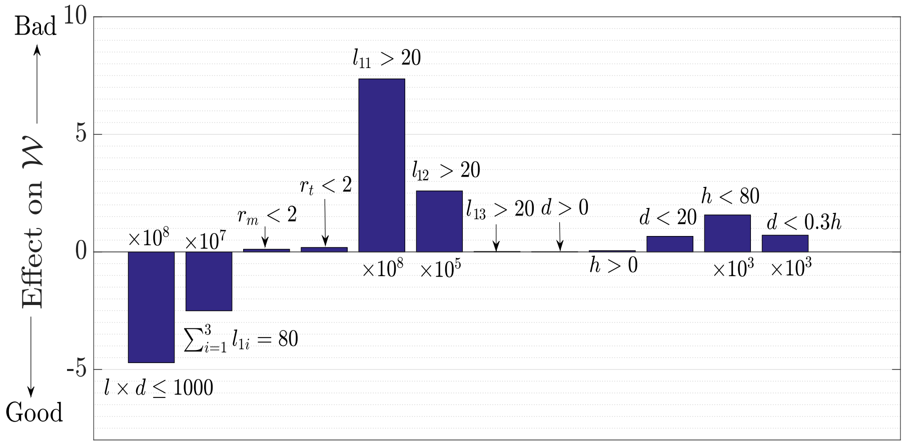
where, in equation (4.32), r_t and r_m are the ratios of the total length of the thumb and the middle fingers divided by the total length of the index finger. Next, we formulate the following optimization problem

$$\begin{aligned} &\underset{\mathbf{P}}{\text{Maximize}}\ \mathcal{W}_S(\mathbf{P}) \\ \textit{Subject to}\quad & h \times d \leq 1000,\ \sum_{i=1}^{3} l_{1i} = 80 \\ & r_m < 2,\ r_t < 2,\ l_{1i} \geq 20,\ \forall i = 1, 2, 3 \\ & d > 0,\ h > 0\ ,\ d \leq 20,\ h \leq 80\ \&\ d \leq 0.3h \end{aligned} \tag{4.33}$$

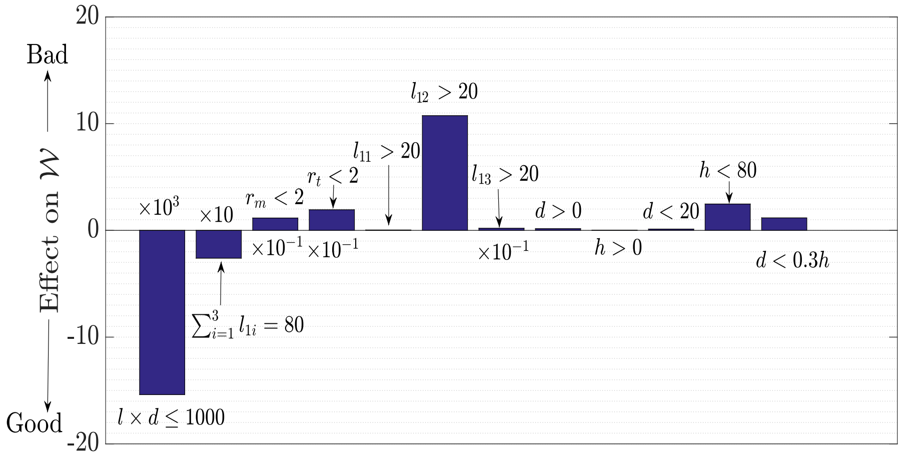
The constraints as well as their numerical values have been obtained for the 95^{th} percentile male according to the studies by [117] and [118]. The Lagrange multipliers for the equality and inequality constraints are also obtained at the optimum. A scaled plot containing the constraint Lagrange multipliers is given in figure 4.12. At *an* optimum point $\vec{\mathscr{P}}$, the Lagrange multipliers, $\vec{\mu}$ and $\vec{\lambda}$, give the effect of the perturbation on the constraints. Any positive perturbations to the constraints associated with the negative Lagrange multipliers will increase the value of the objective function from the value obtained at an optimum [112]. Therefore, from figures 4.12a and 4.12b, we can conclude the following:

- The constraints limiting the hand size i.e., palm area and index finger length given by $l \times d < 1000$ and $\sum_{i=1}^{3} l_{1i} = 80$, have negative Lagrange multipliers associated with them, which signifies the obvious result that a *larger hand* has a larger workspace.
- From the value of the Lagrange multipliers for the constraints we observe that the workspace is more sensitive to a change in palm area than a change in finger length.
- The workspace is not very sensitive to the upper limits on r_m and r_t. Also, at an optimum we obtain $r_m = 1.1$ and $r_t = 1.35$ which are quite close to the values suggested by [117] and [118].
- The workspace is quite sensitive to lower bounds on the fingers segments, however, the workspace is not sensitive to the lower limits on d and h.
- Values of the Lagrange multipliers associated with constraints on the upper limits on d and h suggest that the workspace is equally sensitive to these constraints.

As it is evident from the schematic of the multi-fingered hands from figures 3.6 and 3.17, the variation of the hand workspaces with object size from figures 3.14 and 3.19a and the sensitivity of the hand workspace to the hand dimensions form figure 4.12, we can conclude



(a) Lagrange multipliers for the human hand model at *an* optimum



(b) Lagrange multipliers for the Salisbury hand at *an* optimum

Figure 4.12: Lagrange multipliers for the multi-fingered hands at *an* optimum

that, geometrically the 6 DoF approximation of the human hand (as discussed in section 3.3) and the Salisbury hand (as discussed in section 3.4) are quite similar.

4.6 Conclusion

In this chapter, we have shown that we can use the Monte Carlo method in conjunction with a traditional gradient based optimization method to formulate the optimization problem for a parallel manipulator. We have presented two demonstrative examples of optimization of parallel manipulators. The first example in section 4.3 compares analytical and numerical approach for the optimization of the 5R planar mechanism and provided some mathematical insights into the dependence of the objective function on constraints. In section 4.4 we have attempted a general optimal design problem of the well known semi-regular Stewart platform manipulator

(SRSPM). In both examples, we have attempted a design problem with realistic constraints and demonstrated that the approach presented in this work can be used by a designer to efficiently design an optimum parallel manipulator. Finally in section 4.5 we have demonstrated the sensitivity of the well conditioned workspace of the multi fingered hands to perturbations in design constraints.

Chapter 5

Conclusions and Future work

5.1 Summary

In this thesis, we have attempted to put forth a method for optimal kinematic design of parallel manipulators. We began by presenting the general motivation for the problem in chapter 1. We noted that for designing parallel manipulators, we need to start by representing and quantifying the workspace of the manipulator. To this end, we presented a review of the current state of the art in obtaining the workspace of the manipulators and subsequently proposed a Monte Carlo based method for obtaining the workspace. Next, we discussed the problem of optimum design of parallel manipulators for the largest workspace. Regarding this, we reviewed the current state of the art to outline the different choices of objective functions and techniques for undertaking the said problem. Subsequently, we chose the quantity called "well-conditioned workspace" of the manipulator as the objective function and proposed the use of gradient based optimization techniques to arrive at the optimal dimensions of the manipulator.

In chapter 2 we reviewed the key mathematical and modeling concepts used in the work. We started by discussing the Monte Carlo method. We presented a basic introduction to the technique, a possible improvement on the naive Monte Carlo method and discussed how we have used it in our work. Following which, we discuss how we have represented the workspace of the manipulator using techniques from computational geometry. We finished the chapter by discussing a method of fitting super-ellipsoids to a cloud of points and a technique to obtain the volume of intersection between two arbitrarily placed super-ellipsoids without calculating their distance of closest approach.

In the following chapter 3, we have discussed the application of Monte Carlo method to obtain the workspace of a manipulator. We started by discussing how we have used the Monte Carlo method for obtaining the workspace of the manipulator, as well as advantages and disadvantages of using the Monte Carlo method. Next, we demonstrated the technique with two examples of obtaining workspaces of multi-fingered hands. In the first example, we discussed a 6 degrees of freedom hybrid parallel manipulator to emulate the kinematics of the human three fingered manipulation and obtained the workspace of the hand. We demonstrated the effectiveness of our formulation by comparing theoretically and experimentally obtained workspaces of the hand. In the second example, we obtained the workspace of the Stanford-JPL hand.

Finally, in chapter 4 we presented a method for optimal synthesis of parallel manipulators for a required workspace. We started by discussing the gradient based optimization method and presented two examples to demonstrate the technique. In the first example, we discussed the optimal synthesis of a 2 degrees of freedom 5R planar parallel manipulator for a given workspace considering four different workspace topologies arising from four different assembly conditions. In the second example, we discuss the optimal synthesis of a Stewart platform for a required workspace. Finally, as a consequence of using gradient based optimization techniques, we could also demonstrate the sensitivity of the well conditioned workspace of the manipulator to perturbations to the design constraints of the manipulator.

5.2 Scope of future work

The thesis focuses on kinematic design and optimal synthesis of parallel manipulators, which, concerning design of parallel manipulators, can be regarded as the first step. However, based on the current developments documented in the thesis, the following areas may be explored in future:

- **Optimal synthesis of parallel manipulators**
  The work presented in the thesis may be regarded as the first step in design of parallel manipulators. Results from the current work can be used as possible design parameters to start with and techniques described for visualizing the workspace may be considered.

- **Control and path generation of multi-fingered manipulation**
  We have discussed the kinematics of the three fingered manipulation in section 3.3, by formulating and solving the inverse kinematics problem, assuming realistic joint ranges, finger segment dimensions and nature of contacts between the fingers and the objects. These can be extended to solve path generation and control of multi-fingered manipulation of objects.

Appendix

A Obtaining the joint values of a spherical joint connecting two bodies

In this appendix, we present a method to obtain the joint values of a spherical joint connecting two bodies. For this we present a general case as shown in figure 1.1a, where an object is connected to a body by a spherical joint S, three revolute joints ($\theta_i,\ i = 1, 2, 4$), a prismatic joint δ_3 and 4 links of link lengths $l_i, i = 1, 2, 3, 4$. We need to obtain the joint values to describe the spherical joint S in a given configuration of the manipulator in figure 1.1a.

As the configuration of the manipulator is known, we know all the joint values $\{\theta_1, \theta_2, \delta_3, \theta_4\}$ and the position and orientation of the object with respect to the body (i.e. ${}^{Object}_{Body}[T]$). Since the joint values are known, we can also obtain the position and orientation of the link connected to the object by the spherical joint (i.e. ${}^{l_4}_{Body}[T]$). From this information, we can obtain the orientation of the link l_4 and the object with respect to the body frame (i.e. ${}^{l_4}_{Body}[R]$ and ${}^{Object}_{Body}[R]$) individually by extracting the top left 3×3 matrix from each of ${}^{l_4}_{Body}[T]$ and ${}^{Object}_{Body}[T]$ respectively.

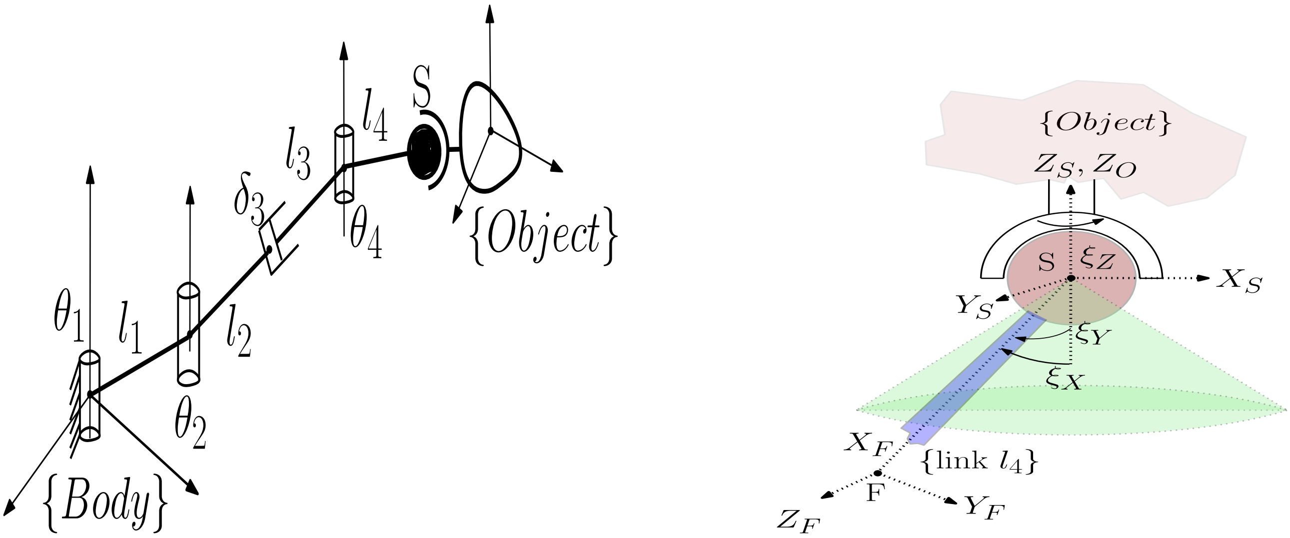
Subsequently to obtain the rotation of the S joint we observe, following Ghosal [81], that a spherical joint can be expressed as a combination of 3 mutually perpendicular revolute joints. The relative rotation between the link l_4 and the object is given by the rotation matrix R^* as given in equation (1.1)

$$R^* = {}^{l_4}_{Body}[R]^T \, {}^{Object}_{Body}[R] \tag{1.1}$$

For the rotation matrix R^* with elements $r_{ij},\ i, j = 1, 2, 3$, the three rotations (Z-Y-X rotation)[1] is found out by using the following algorithm:

---

[1] For details on the choice of this rotation matrix see Shyam [119]



(a) Generic 4 link manipulator

(b) Schematic of the spherical joint

Figure 1.1: Manipulator with a spherical joint

If $r_{31} \neq \pm 1$, then

$$\begin{aligned}
\xi_Y &= \mathrm{atan2}[-\mathrm{r}_{31}, \pm\sqrt{\mathrm{r}_{32}^2 + \mathrm{r}_{33}^2}\ \ ] \\
\xi_Z &= \mathrm{atan2}[\mathrm{r}_{21}/\cos(\theta_{\xi_\mathrm{Y}}), \mathrm{r}_{11}/\cos(\theta_{\xi_\mathrm{Y}})] \\
\xi_Z &= \mathrm{atan2}[\mathrm{r}_{32}/\cos(\theta_{\xi_\mathrm{Y}}), \mathrm{r}_{33}/\cos(\theta_{\xi_\mathrm{Y}})]
\end{aligned}$$

If r_{31} = 1, then

$$\xi_Y = -\frac{\pi}{2}, \quad \xi_Y = 0, \quad \xi_X = \mathrm{atan2}[-\mathrm{r}_{12}, -\mathrm{r}_{13}]$$

If r_{31} = -1, then

$$\xi_Y = \frac{\pi}{2}, \quad \xi_Z = 0, \quad \xi_X = \mathrm{atan2}[\mathrm{r}_{12}, \mathrm{r}_{13}]$$

where atan2 is the four quadrant inverse tangent function.

Bibliography

[1] C. Innocenti and V. Parenti-Castelli, "Echelon form solution of direct kinematics for the general fully-parallel spherical wrist", *Mechanism and Machine Theory*, vol. 28, no. 4, pp. 553–561, 1993 (cit. on p. 1).

[2] R. A. Shyam and A. Ghosal, "Three-degree-of-freedom parallel manipulator to track the sun for concentrated solar power systems", *Chinese Journal of Mechanical Engineering*, vol. 28, no. 4, pp. 793–800, 2015 (cit. on p. 1).

[3] ——, "Path planning of a 3-upu wrist manipulator for sun tracking in central receiver tower systems", *Mechanism and Machine Theory*, vol. 119, pp. 130–141, 2018 (cit. on p. 1).

[4] R. A. Shyam, M. Acharya, and A. Ghosal, "A heliostat based on a three degree-of-freedom parallel manipulator", *Solar Energy*, vol. 157, pp. 672–686, 2017 (cit. on p. 1).

[5] J. K. Salisbury and J. J. Craig, "Articulated hands: Force control and kinematic issues", *The International Journal of Robotics Research*, vol. 1, no. 1, pp. 4–17, 1982 (cit. on pp. 1, 37, 38, 59).

[6] S. Jacobsen, E. Iversen, D. Knutti, R. Johnson, and K. Biggers, "Design of the Utah/MIT dextrous hand", in *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, IEEE, vol. 3, 1986, pp. 1520–1532 (cit. on pp. 1, 38).

[7] J. Borràs and A. M. Dollar, "Dimensional synthesis of three-fingered robot hands for maximal precision manipulation workspace", *The International Journal of Robotics Research*, vol. 34, no. 14, pp. 1731–1746, 2015 (cit. on pp. 1, 4, 38).

[8] Phantom premium product range, *Haptic products by geomagic inc. 3d systems, morrisville, north carolina, usa.* `https://www.3dsystems.com/haptics-devices/3d-systems-phantom-premium`, Accessed: 2017-11-03 (cit. on p. 1).

[9] F. Wen and C. Liang, "Displacement analysis of the 6-6 stewart platform mechanisms", *Mechanism and Machine Theory*, vol. 29, no. 4, pp. 547–557, 1994 (cit. on p. 1).

[10] M. Raghavan, “The stewart platform of general geometry has 40 configurations”, *Journal of Mechanical Design*, vol. 115, no. 2, pp. 277–282, 1993 (cit. on p. 1).

[11] S. Bandyopadhyay and A. Ghosal, “Analysis of configuration space singularities of closed-loop mechanisms and parallel manipulators”, *Mechanism and Machine Theory*, vol. 39, no. 5, pp. 519–544, 2004 (cit. on p. 1).

[12] ——, “Geometric characterization and parametric representation of the singularity manifold of a 6–6 Stewart platform manipulator”, *Mechanism and Machine Theory*, vol. 41, no. 11, pp. 1377–1400, 2006 (cit. on p. 1).

[13] ——, “An algebraic formulation of kinematic isotropy and design of isotropic 6-6 stewart platform manipulators”, *Mechanism and Machine Theory*, vol. 43, no. 5, pp. 591–616, 2008 (cit. on p. 1).

[14] B. Dasgupta and T. Mruthyunjaya, “Closed-form dynamic equations of the general Stewart platform through the newton–euler approach”, *Mechanism and machine theory*, vol. 33, no. 7, pp. 993–1012, 1998 (cit. on p. 1).

[15] O. Hatip and M. Ozgoren, “Utilization of a Stewart platform mechanism as a stabilizator”, in *9th world congress on the theory of machines and mechanisms. Milan*, vol. 1, 1995, pp. 1393–1396 (cit. on p. 1).

[16] S. Narasimhan, “Dexterous robotic hands: Kinematics and control”, DTIC Document, Tech. Rep., 1988 (cit. on p. 1).

[17] H. Wang, C. Xue, and W. Gruver, “Neural network control of a parallel robot”, in *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on*, IEEE, vol. 3, 1995, pp. 2934–2938 (cit. on p. 1).

[18] K. M. Lynch and F. C. Park, *Modern Robotics: Mechanics, Planning, and Control.* Cambridge University Press, 2017 (cit. on p. 2).

[19] J.-P. Merlet, *Parallel robots.* Springer Science & Business Media, 2012, vol. 74 (cit. on pp. 2–4, 9, 16, 36).

[20] X.-J. Liu, J. Wang, and H.-J. Zheng, “Optimum design of the 5r symmetrical parallel manipulator with a surrounded and good-condition workspace”, *Robotics and Autonomous Systems*, vol. 54, no. 3, pp. 221–233, 2006 (cit. on pp. 2, 28, 69, 70).

[21] J.-P. Merlet, “Determination of 6d workspaces of gough-type parallel manipulator and comparison between different geometries”, *The International Journal of Robotics Research*, vol. 18, no. 9, pp. 902–916, 1999 (cit. on pp. 2, 28).

[22] J.-P. Merlet, C. M. Gosselin, and N. Mouly, “Workspaces of planar parallel manipulators”, in *ROMANSY 11*, Springer, 1997, pp. 37–44 (cit. on p. 2).

[23] R. A. Srivatsan and S. Bandyopadhyay, “Determination of the safe working zone of a parallel manipulator”, in *Computational Kinematics*, Springer, 2014, pp. 201–208 (cit. on p. 3).

[24] D. Patel, R. Kalla, H. Tetik, G. Kiper, and S. Bandyopadhyay, “Computing the safe working zone of a 3-rrs parallel manipulator”, in *New Trends in Mechanism and Machine Science*, Springer, 2017, pp. 113–120 (cit. on p. 3).

[25] D. Chablat, P. Wenger, F. Majou, and J.-P. Merlet, “An interval analysis based study for the design and the comparison of three-degrees-of-freedom parallel kinematic machines”, *The International Journal of Robotics Research*, vol. 23, no. 6, pp. 615–624, 2004 (cit. on pp. 3, 32, 36).

[26] F. Hao and J.-P. Merlet, “Multi-criteria optimal design of parallel manipulators based on interval analysis”, *Mechanism and machine theory*, vol. 40, no. 2, pp. 157–171, 2005 (cit. on pp. 3, 5, 36, 68).

[27] J.-P. Merlet, “A formal-numerical approach for robust in-workspace singularity detection”, *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 393–402, 2007 (cit. on pp. 3, 4).

[28] S. Caro, D. Chablat, A. Goldsztejn, D. Ishii, and C. Jermann, “A branch and prune algorithm for the computation of generalized aspects of parallel robots”, *Artificial Intelligence*, vol. 211, pp. 34–50, 2014 (cit. on pp. 3, 4, 32, 35).

[29] O. Bohigas, M. Manubens, and L. Ros, “A complete method for workspace boundary determination on general structure manipulators”, *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 993–1006, 2012 (cit. on p. 3).

[30] J. Rastegar and B. Fardanesh, “Manipulation workspace analysis using the monte carlo method”, *Mechanism and Machine Theory*, vol. 25, no. 2, pp. 233–239, 1990 (cit. on p. 3).

[31] J. Rastegar and D. Perel, “Generation of manipulator workspace boundary geometry using the monte carlo method and interactive computer graphics”, *Journal of mechanical design*, vol. 112, no. 3, pp. 452–454, 1990 (cit. on p. 3).

[32] D. Alciatore and C. Ng, “Determining manipulator workspace boundaries using the monte carlo method and least squares segmentation”, *ASME Robotics: Kinematics, Dynamics and Controls*, vol. 72, pp. 141–146, 1994 (cit. on p. 3).

[33] R. E. Stamper, L.-W. Tsai, and G. C. Walsh, "Optimization of a three DOF translational platform for well-conditioned workspace", in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, IEEE, vol. 4, 1997, pp. 3250–3255 (cit. on pp. 3, 6, 13, 28).

[34] L.-W. Tsai, G. C. Walsh, and R. E. Stamper, "Kinematics of a novel three DOF translational platform", in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, IEEE, vol. 4, 1996, pp. 3446–3451 (cit. on p. 3).

[35] C. Gosselin, "Kinematic analysis, optimization and programming of parallel robotic manipulators", PhD thesis, Department of Mechanical Engineering, McGill University, Canada, 1988 (cit. on pp. 3, 5).

[36] R. Boudreau and C. Gosselin, "The synthesis of planar parallel manipulators with a genetic algorithm", *Journal of mechanical design*, vol. 121, no. 4, pp. 533–537, 1999 (cit. on p. 4).

[37] Y. Davidor, *Genetic Algorithms and Robotics: A heuristic strategy for optimization.* World Scientific, 1991, vol. 1 (cit. on p. 4).

[38] J. Grefenstette, "Evolutionary algorithms in robotics", in *International Symposium on Robotics and Manufacturing (ISRAM)*, Citeseer, 1994 (cit. on p. 4).

[39] O. Masory and J. Wang, "Workspace evaluation of stewart platforms", *Advanced robotics*, vol. 9, no. 4, pp. 443–461, 1994 (cit. on pp. 4, 9, 16, 88).

[40] Y. Tsai and A. Soni, "An algorithm for the workspace of a general nR robot", *Journal of Mechanisms, Transmissions, and Automation in Design*, vol. 105, no. 1, pp. 52–57, 1983 (cit. on pp. 4, 16).

[41] K. H. Pittens and R. P. Podhorodeski, "A family of stewart platforms with optimal dexterity", *Journal of Robotic systems*, vol. 10, no. 4, pp. 463–479, 1993 (cit. on p. 4).

[42] C.-S. Han, D. Tesar, and A. Traver, "Optimum design of a 6 dof fully-parallel micromanipulator for enhanced robot accuracy", in *American Soc. of Mechanical Engineers (ASME)*, 1989 (cit. on p. 4).

[43] C. M. Gosselin and M. Guillot, "The synthesis of manipulators with prescribed workspace", *ASME J. Mech. Des*, vol. 113, no. 4, pp. 451–455, 1991 (cit. on p. 4).

[44] J.-P. Merlet, "Optimal design of robots", in *Proceedings of Robotics: Science and Systems*, Cambridge, USA, Jun. 2005 (cit. on pp. 4, 5, 36).

[45] R. Kurtz and V. Hayward, “Multiple-goal kinematic optimization of a parallel spherical mechanism with actuator redundancy”, *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 644–651, 1992 (cit. on p. 5).

[46] F. Lara-Molina, J. Rosario, and D. Dumur, “Multi-objective design of parallel manipulator using global indices”, *Open Mechanical Engineering Journal*, vol. 4, no. 1, pp. 37–47, 2010 (cit. on p. 5).

[47] J.-P. Merlet, “Jacobian, manipulability, condition number, and accuracy of parallel robots”, *Journal of Mechanical Design*, vol. 128, no. 1, pp. 199–206, 2006 (cit. on p. 5).

[48] Y. Lou, G. Liu, and Z. Li, “A general approach for optimal design of parallel manipulators”, *IEEE Transactions on Automation science and engineering*, 2005 (cit. on pp. 6, 66).

[49] Y. Lou, G. Liu, N. Chen, and Z. Li, “Optimal design of parallel manipulators for maximum effective regular workspace”, in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2005, pp. 795–800 (cit. on p. 6).

[50] Y. Lou, G. Liu, and Z. Li, “Randomized optimal design of parallel manipulators”, *IEEE transactions on automation science and engineering*, vol. 5, no. 2, pp. 223–233, 2008 (cit. on pp. 6, 66, 78, 82).

[51] A. Törn and A. Žilinskas, *Global Optimization*. Springer-Verlag New York, Inc., 1989 (cit. on p. 6).

[52] L.-W. Tsai and S. Joshi, “Kinematics and optimization of a spatial 3-upu parallel manipulator”, *Journal of Mechanical Design*, vol. 122, no. 4, pp. 439–446, 2000 (cit. on p. 6).

[53] W. L. Dunn and J. K. Shultis, *Exploring Monte Carlo Methods*. Elsevier, 2011 (cit. on pp. 8, 30).

[54] G. Fishman, *Monte Carlo: Concepts, Algorithms, and Applications*. Springer Science & Business Media, 2013 (cit. on pp. 9, 10, 12, 13).

[55] H. J. M. and D. C. Handscomb, “Monte carlo methods. pp 134-141.”, *Springer Netherlands,*, 1964 (cit. on pp. 9, 13).

[56] J. P. Kleijnen, A. A. Ridder, and R. Y. Rubinstein, *Variance reduction techniques in Monte Carlo methods*. Springer, 2013 (cit. on p. 13).

[57] MATLAB, *version 8.5.0.197613 (R2015a)*. Natick, Massachusetts: The MathWorks Inc., 2015 (cit. on pp. 13, 15, 33, 50, 67, 68).

[58] P. Glasserman, *Monte Carlo Methods in Financial Engineering.* Springer Science & Business Media, 2013, vol. 53 (cit. on p. 14).

[59] G. E. Box, M. E. Muller, *et al.*, “A note on the generation of random normal deviates”, *The Annals of Mathematical Statistics*, vol. 29, no. 2, pp. 610–611, 1958 (cit. on p. 15).

[60] D. Lee and B. J. Schachter, “Two algorithms for constructing a delaunay triangulation”, *International Journal of Computer & Information Sciences*, vol. 9, no. 3, pp. 219–242, 1980 (cit. on pp. 17, 18, 31).

[61] H. Edelsbrunner and E. P. Mücke, “Three-dimensional alpha shapes”, *ACM Transactions on Graphics (TOG)*, vol. 13, no. 1, pp. 43–72, 1994 (cit. on p. 17).

[62] E. L. Allgower and P. H. Schmidt, “An algorithm for piecewise-linear approximation of an implicitly defined manifold”, *SIAM journal on numerical analysis*, vol. 22, no. 2, pp. 322–346, 1985 (cit. on p. 18).

[63] E. L. Allgower and P. H. Schmidt, “Computing volumes of polyhedra”, *Mathematics of Computation*, vol. 46, no. 173, pp. 171–174, 1986 (cit. on p. 18).

[64] O. Golubitsky, V. Mazalov, and S. M. Watt, “An algorithm to compute the distance from a point to a simplex”, *ACM Commun. Comput. Algebra*, vol. 46, pp. 57–57, 2012 (cit. on p. 20).

[65] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, “A fast procedure for computing the distance between complex objects in three-dimensional space”, *IEEE Journal on Robotics and Automation*, vol. 4, no. 2, pp. 193–203, 1988 (cit. on p. 20).

[66] C. J. Ong and E. G. Gilbert, “The gilbert-johnson-keerthi distance algorithm: A fast version for incremental motions”, in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, IEEE, vol. 2, 1997, pp. 1183–1189 (cit. on p. 20).

[67] T. Ju, T. Simpson, J. O. Deasy, and D. A. Low, “Geometric interpretation of the γ dose distribution comparison technique: Interpolation-free calculation”, *Medical physics*, vol. 35, no. 3, pp. 879–887, 2008 (cit. on p. 20).

[68] E. Bardinet, N. Ayache, and L. Cohen, “Fitting of iso-surfaces using superquadrics and free-form deformations”, in *Biomedical Image Analysis, 1994., Proceedings of the IEEE Workshop on*, IEEE, 1994, pp. 184–193 (cit. on p. 21).

[69] F. Solina and R. Bajcsy, “Recovery of parametric models from range images: The case for superquadrics with global deformations”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 2, pp. 131–147, 1990 (cit. on p. 21).

[70] M. S. Menon, V. Ravi, and A. Ghosal, “Trajectory planning and obstacle avoidance for hyper-redundant serial robots”, *Journal of Mechanisms and Robotics*, vol. 9, no. 4, p. 041 010, 2017 (cit. on p. 21).

[71] A. H. Barr, “Rigid physically based superquadrics”, in *Graphics Gems III*. Academic Press Professional, Inc., 1992, ch. III.8, pp. 137–159 (cit. on pp. 21, 47).

[72] A. B. Cole, J. E. Hauser, and S. S. Sastry, “Kinematics and control of multifingered hands with rolling contact”, *IEEE Transactions on Automatic Control*, vol. 34, no. 4, pp. 398–404, 1989 (cit. on pp. 21, 40).

[73] G. Taubin, “An improved algorithm for algebraic curve and surface fitting”, in *Computer Vision, 1993. Proceedings., Fourth International Conference on*, IEEE, 1993, pp. 658–665 (cit. on p. 22).

[74] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge university press, 2004 (cit. on pp. 22, 67, 75).

[75] S. Agarwal, R. A. Srivatsan, and S. Bandyopadhyay, “Analytical determination of the proximity of two right-circular cylinders in space”, *Journal of Mechanisms and Robotics*, vol. 8, no. 4, p. 041 010, 2016 (cit. on p. 23).

[76] M. Moore and J. Wilhelms, “Collision detection and response for computer animation”, in *ACM Siggraph Computer Graphics*, ACM, vol. 22, 1988, pp. 289–298 (cit. on p. 24).

[77] C. Wellmann, C. Lillie, and P. Wriggers, “A contact detection algorithm for superellipsoids based on the common-normal concept”, *Engineering Computations*, vol. 25, no. 5, pp. 432–442, 2008 (cit. on p. 24).

[78] F. Barbagli, A. Frisoli, K. Salisbury, and M. Bergamasco, “Simulating human fingers: A soft finger proxy model and algorithm”, in *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004. HAPTICS’04. Proceedings. 12th International Symposium on*, IEEE, 2004, pp. 9–17 (cit. on p. 24).

[79] R. D. Howe, I. Kao, and M. R. Cutkosky, “The sliding of robot fingers under combined torsion and shear loading”, in *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, IEEE, 1988, pp. 103–105 (cit. on p. 24).

[80] J. M. Selig, *Geometric fundamentals of robotics*. Springer Science & Business Media, 2004 (cit. on pp. 28, 30).

[81] A. Ghosal, *Robotics: Fundamental Concepts and Analysis*. Oxford University Press, 2006 (cit. on pp. 28, 43, 59, 60, 80, 94).

[82] A. Ghosal and B. Ravani, “A differential-geometric analysis of singularities of point trajectories of serial and parallel manipulators”, *Transactions of the American Society of Mechanical Engineers Journal of Mechanical Design*, vol. 123, no. 1, pp. 80–89, 2001 (cit. on pp. 28, 45).

[83] J. J. Cervantes-Sanchez, J. C. Hernandez-Rodriguez, and J. G. Rendon-Sanchez, “On the workspace, assembly configurations and singularity curves of the RRRRR-type planar manipulator”, *Mechanism and Machine Theory*, vol. 35, no. 8, pp. 1117–1139, 2000 (cit. on pp. 32, 69, 73, 75).

[84] E. Macho, O. Altuzarra, C. Pinto, and A. Hernandez, “Workspaces associated to assembly modes of the 5r planar parallel manipulator”, *Robotica*, vol. 26, no. 03, pp. 395–403, 2008 (cit. on pp. 32, 69).

[85] S. M. Rump, “Intlab—interval laboratory”, in *Developments in reliable computing*, Springer, 1999, pp. 77–104 (cit. on p. 35).

[86] H. Ritter and R. Haschke, “Hands, dexterity, and the brain”, in *Humanoid Robotics and Neuroscience: Science, Engineering and Society*, CRC Press, 2014, pp. 49–77 (cit. on p. 37).

[87] R. Balasubramanian and V. J. Santos, *The Human Hand as an Inspiration for Robot Hand Development*. Springer, 2014, vol. 95 (cit. on pp. 37, 38).

[88] A. M. Okamura, N. Smaby, and M. R. Cutkosky, “An overview of dexterous manipulation”, in *Robotics and Automation, 2000. Proceedings. ICRA’00. IEEE International Conference on*, IEEE, vol. 1, 2000, pp. 255–262 (cit. on p. 38).

[89] H. Hanafusa and H. Asada, “A robot hand with elastic fingers and its application to assembly process”, in *Proc. of IFAC First Symposium on Information Control Problems in Manufacturing Technology*, 1977, pp. 127–138 (cit. on p. 38).

[90] R. M. Murray and S. S. Sastry, “Control experiments in planar manipulation and grasping”, in *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, IEEE, 1989, pp. 624–629 (cit. on p. 38).

[91] J. Butterfaß, M. Grebenstein, H. Liu, and G. Hirzinger, “DLR-Hand II: Next generation of a dextrous robot hand”, in *Proceedings of ICRA. IEEE International Conference on Robotics and Automation, 2001.*, IEEE, vol. 1, 2001, pp. 109–114 (cit. on p. 38).

[92] Shadow Robot Company, *Shadow Hand*, `https://www.shadowrobot.com/products/dexterous-hand/`, Accessed: 2017-03-15, 2013 (cit. on pp. 38, 58).

[93] R. Deimel and O. Brock, “A novel type of compliant and underactuated robotic hand for dexterous grasping”, *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 161–185, 2016 (cit. on p. 38).

[94] A. D. Deshpande and Y. Matsuoka, “Development of an anatomically correct testbed (ACT) hand”, in *The Human Hand as an Inspiration for Robot Hand Development*, Springer, 2014, pp. 453–475 (cit. on p. 38).

[95] J. Lin, Y. Wu, and T. S. Huang, “Modeling the constraints of human hand motion”, in *Human Motion, 2000. Proceedings. Workshop on*, IEEE, 2000, pp. 121–126 (cit. on pp. 38, 40).

[96] B. Dizioğlu and K. Lakshiminarayana, “Mechanics of form closure”, *Acta Mechanica*, vol. 52, no. 1, pp. 107–118, 1984 (cit. on p. 38).

[97] V.-D. Nguyen, “Constructing stable grasps in 3D”, in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, IEEE, vol. 4, 1987, pp. 234–239 (cit. on p. 38).

[98] D. J. Montana, “The kinematics of contact and grasp”, *The International Journal of Robotics Research*, vol. 7, no. 3, pp. 17–32, 1988 (cit. on pp. 38, 40, 45, 47, 49).

[99] C. S. Cai and B. Roth, “On the planar motion of rigid bodies with point contact”, *Mechanism and Machine Theory*, vol. 21, no. 6, pp. 453–466, 1986 (cit. on pp. 38, 40, 45).

[100] J. Kerr and B. Roth, “Analysis of multifingered hands”, *The International Journal of Robotics Research*, vol. 4, no. 4, pp. 3–17, 1986 (cit. on pp. 38, 40, 41, 45).

[101] S. Cobos, M. Ferre, M. S. Uran, J. Ortego, and C. Pena, “Efficient human hand kinematics for manipulation tasks”, in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, IEEE, 2008, pp. 2246–2251 (cit. on p. 38).

[102] Artec3D, *Arteceva scanner*, `https://www.artec3d.com/3d-scanner/artec-eva`, Accessed: 2017-03-15 (cit. on p. 39).

[103] M. Nakamura, C. Miyawaki, N. Matsushita, R. Yagi, and Y. Handa, “Analysis of voluntary finger movements during hand tasks by a motion analyzer”, *Journal of Electromyography and Kinesiology*, vol. 8, no. 5, pp. 295–303, 1998 (cit. on p. 40).

[104] R. Degeorges and C. Oberlin, “Measurement of three-joint-finger motions: Reality or fancy? a three-dimensional anatomical approach”, *Surgical and Radiologic Anatomy*, vol. 25, no. 2, pp. 105–112, 2003 (cit. on p. 40).

[105] R. Degeorges, S. Laporte, E. Pessis, D. Mitton, J.-N. Goubier, and F. Lavaste, "Rotations of three-joint fingers: A radiological study", *Surgical and Radiologic Anatomy*, vol. 26, no. 5, pp. 392–398, 2004 (cit. on p. 40).

[106] Autodesk-netfabb, *version 7.4.0 532(2013).* Mill Valley, California, USA: Autodesk, Inc., 2013 (cit. on p. 40).

[107] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation.* CRC press, 1994 (cit. on p. 40).

[108] G. Strang, *Introduction to Linear Algebra.* Wellesley-Cambridge Press Wellesley, MA, 1993, vol. 3 (cit. on p. 47).

[109] Polhemus-Liberty, *Polhemus-liberty wired electromagnetic position and orientation tracker*, `http://polhemus.com/motion-tracking/all-trackers/liberty`, Accessed: 2017-04-22 (cit. on p. 51).

[110] J. Darbon, A. Cunha, T. F. Chan, S. Osher, and G. J. Jensen, "Fast nonlocal filtering applied to electron cryomicroscopy", in *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, IEEE, 2008, pp. 1331–1334 (cit. on p. 52).

[111] B. H. Tracey and E. L. Miller, "Nonlocal means denoising of ECG signals", *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 9, pp. 2383–2386, 2012 (cit. on p. 52).

[112] J. Arora, *Introduction to Optimum Design, Chapter 4, pp 154-157,* Academic Press, 2004 (cit. on pp. 68, 76, 89).

[113] M. Z. Huang, "Performance analysis and design optimization of 5r planar parallel robots", in *ASME 2012 International Mechanical Engineering Congress and Exposition*, American Society of Mechanical Engineers, 2012, pp. 351–356 (cit. on p. 75).

[114] B. Dasgupta and T. Mruthyunjaya, "The Stewart platform manipulator: A review", *Mechanism and machine theory*, vol. 35, no. 1, pp. 15–40, 2000 (cit. on p. 78).

[115] E. F. Fichter, "A stewart platform-based manipulator: General theory and practical construction", *The International Journal of Robotics Research*, vol. 5, no. 2, pp. 157–182, 1986 (cit. on pp. 78, 82, 86).

[116] D. Chatterjee and A. Ghosal, "Design of a semi-regular Stewart platform manipulator for a desired workspace", *NACOMM 2009 NIT Durgapur*, 2009 (cit. on pp. 78, 82).

[117] R. M. White, "Comparative anthropometry of the hand", DTIC Document, Tech. Rep., 1980 (cit. on p. 89).

[118] J. T. Manning, “Sex differences and age changes in digit ratios: Implications for the use of digit ratios in medicine and biology”, in *Handbook of Anthropometry*, Springer, 2012, pp. 841–851 (cit. on p. 89).

[119] R. B. A. Shyam, “Design and development of a three-degree-of-freedom parallel manipulator to track the sun for concentrated solar power towers”, PhD thesis, Indian Institute of Science Bangalore, 2017 (cit. on p. 94).