

Optimum design of multi-degree-of-freedom closed-loop mechanisms and parallel manipulators for a prescribed workspace using Monte Carlo method

Arkadeep Narayan Chaudhury ^{*}, Ashitava Ghosal [†]

Abstract

In this work, we use the Monte Carlo method in conjunction with gradient based optimization algorithms to optimally design multi-degree-of-freedom parallel manipulators and closed-loop mechanisms. The design procedure takes into account practical constraints such as joint limits and guarantees well-conditioning of the desired workspace. As a first step, an appropriate bounding box representing the wanted workspace is obtained by using the Monte Carlo method and then the geometrical dimensions of the manipulator are obtained through a gradient based optimization method by accounting for the joint and other constraints. The computational advantages of the Monte Carlo technique over other search based methods in evaluating the objective function for the optimization problem is illustrated. The constraint Lagrange multipliers are obtained and sensitivity of the workspace dimensions to the constraints on joint limits and conditioning have been demonstrated. The approach is illustrated with the design of a two-degree-of-freedom planar 5-bar closed-loop mechanism and a spatial, six-degree-of-freedom Stewart platform manipulator.

Keywords: Monte Carlo method, workspace, condition number, joint limits, gradient based optimization, constraint sensitivity analysis.

Nomenclature and list of symbols

${}^B_A[R]$ Rotation matrix of frame $\{B\}$ with respect to frame $\{A\}$

\mathcal{R} Set of real numbers

$SO(3)$ Special orthogonal group of order 3

u_i i^{th} Element of a vector \mathbf{U}

$\Pr(X)$ Probability of a random variable X

$E(X)$ Expectation of a variable X taking a probabilistic value

S_n Average of n quantities (x_1, x_2, \dots, x_n)

$V(X)$ Variance of a variable X taking a probabilistic value

\hat{X} Estimate of the quantity X

\mathbf{J} Jacobian matrix

κ Condition number of a matrix

$tr(\mathbf{A})$ Trace of a matrix \mathbf{A}

\mathcal{O} The 'Big O' notation denoting computational complexity

\mathcal{W} Workspace of a manipulator

S_i, U_i Centers of spherical and universal joints respectively

^{*}Graduate Student, Department of Mechanical Engineering, Indian Institute of Science, Bangalore, India. Email: arkadeep@mecheng.iisc.ernet.in

[†]Corresponding Author, Professor, Department of Mechanical Engineering, Indian Institute of Science, Bangalore, India. Email: asitava@mecheng.iisc.ernet.in

1 Introduction

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 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]. Hybrid parallel manipulators have been proposed as a model of multi-fingered hands (see, for example, the Stanford-JPL hand by Salisbury and Craig [3] and the Utah-MIT hand by Jacobsen et al. [4] and the 3 fingered hand by Borras and Dollar [5]). 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 [6] 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 [7] and Raghavan and Roth [8] and the references contained therein), perform singularity analysis (see, for example, Bandyopadhyay and Ghosal [9, 10, 11] and the references contained therein), derive and numerically solve the dynamic equations of motion (see, for example, the comprehensive review by Dasgupta and Mruthyunjaya [12] and the references contained therein) and for control (see, for example, the works by Hatip and Ozgoren [13], Narasimhan [14], Wang et al. [15], Wen et al. [7] 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 work, we focus on the problem of optimal dimensional synthesis of a parallel manipulator or a closed-loop mechanism for a specified workspace, subject to joint rotation limits and obtaining a well-conditioned workspace. In the review paper on Stewart platform manipulators [16], the authors have also recognized the above as an open area of research. 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. In the following we present a brief summary of the current state-of-the-art in this topic.

In a body of literature (see, for example, the work by Boudreau and Gosselin [17] and the book by Davidor [18] 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 [19] 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 [20] and Tsai and Soni [21]) 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 [22] and Han et al. [23] 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 [24] 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. [25]-[28] and [29] 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 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. [30] 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 [5] 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. Lou et al. ([31, 32, 33]) have used the controlled random search (CRS) method to optimize robots for regular workspaces with good dexterity. The works by Stamper et al. [34], Tsai and Joshi [35] use Monte Carlo search based methods to optimize manipulators for the largest well-conditioned workspace. The objective function is generally a representative of the quantity of the actual workspace. The works by Lou et al. and Tsai et al. ([31], [32] – [35]) are closest to the current work and in this paper, we extend some of the ideas presented in these works.

The configuration space for a 6 degree-of-freedom (DOF) parallel manipulator is large and the joint space (both actuated and un-actuated) is even larger – in the Stewart platform manipulator, the number of actuated joints is 6 and total number of joint variables, active and passive, is 18. For posing the problem in the joint space, an efficient closed-form solution of the direct kinematics problem for the manipulator should be at hand and this is often difficult – the direct kinematics of the most general Stewart platform requires the solution of a 40^{th} degree polynomial [8]. Therefore posing and solving the optimization problem with constraints (both geometric and joint limit) in either of the spaces is difficult. Additionally, most of the above works do not consider the measure of the well-conditioning in both of the configuration and orientation workspaces. In this paper, we show the Monte Carlo method, as described in the work by Tsai and Soni [21], can be used very successfully in design of parallel manipulators and closed-loop mechanisms. We show that a) the solution to the direct kinematics problem is not required and the simpler inverse kinematics solution is enough, b) joint and other constraints can be easily incorporated, c) the design gives the largest well-conditioned workspace, and d) in compared to other existing approaches, it is computationally efficient. We also show that a gradient based optimization method can also be used in conjunction

to the Monte Carlo based search and this provides insights on the dependence of the workspace of the manipulator on the constraints.

This paper is organized as follows: In section 2, we briefly describe the Monte Carlo search based approach and how it can be applied to obtain a cloud of points which satisfies inverse kinematics and specified joint limits. We discuss error, choice of sample size, representation of the workspace and compare the workspace obtained using direct kinematics and the Monte Carlo method for an example. The representation of the workspace volume is then used in section 3 and a gradient based optimization problem is formulated for obtaining the dimensions of closed-loop mechanisms and parallel manipulators. The gradient based optimization allows us to perform a sensitivity analysis which gives more insight into the optimization. In section 4 we present two examples, namely a two-degree-of-freedom planar 5R closed-loop mechanism and the well-known six-degree-of-freedom semi-regular Stewart platform manipulator. In section 5. the paper is summarized and scope for future work is presented.

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 [36]. The integral

$$\mathcal{I} = \int_{[0,1]^d} f(\mathbf{x}) \, d\mathbf{x}$$

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 $\mathbf{U} = [u_1, u_2, \dots, u_d]^T$ a $1 \times d$ vector taking random values of $u_i \in [0, 1] \forall i = 1, 2, \dots, d$. From the strong law of large numbers¹ the average,

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

*almost surely*² converges to $E(f(\mathbf{U}))$ as $n \rightarrow \infty$. The volume (area) of a manipulator workspace is an integration problem in \mathfrak{R}^3 (\mathfrak{R}^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 and Wang [20] and Merlet [29]).

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 [37] and relevant sections of the book by Hammersley and Handscomb [38].

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

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

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

Let \mathcal{R} denote the region of unknown volume $\lambda(\mathcal{R})$ in the m -D hypercube denoted by $\mathcal{I} = [0, 1]^m$. Assume that the region \mathcal{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 \mathcal{R} , i.e., $\lambda(\mathcal{R})$ is not possible or computationally prohibitive. We further assume that a systematic procedure exists³ for generating a sequence $\mathcal{H}_{m,n}$ of n points in m -D hypercube

$$\mathcal{H}_{m,n} = \{\mathbf{X}^j = (x_1^{(j)}, x_2^{(j)}, x_3^{(j)}, \dots, x_m^{(j)})\}. \quad (2)$$

The numerical accuracy of obtaining $\widehat{\lambda(\mathcal{R})}$, an estimate the volume, depends on how $\mathcal{H}_{m,n}$ is populated⁴. The Algorithm 1 given below can be used to obtain $\widehat{\lambda(\mathcal{R})}$ by populating the vector \mathbf{X} in equation (2) by generating m random numbers for x_i^s between $[0, 1]$. From Algorithm 1, a point estimate of the quantity $\widehat{\lambda(\mathcal{R})}$ 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 \mathcal{R} . The variance $V(\widehat{\lambda(\mathcal{R})})$ can be obtained as

$$V(\widehat{\lambda(\mathcal{R})}) = \frac{S}{n} \left(1 - \frac{S}{n}\right) \left(\frac{1}{n-1}\right) \quad (3)$$

Purpose : To obtain an estimate of $\lambda(\mathcal{R})$

Input: \mathcal{R} and sample size 'n'

Output: $\widehat{\lambda(\mathcal{R})}$

- 1: Initialize $j=0, S=0$;
- 2: Allocate memory for $\mathcal{H}_{m,n}$
- 3: **while** $j \leq n$ **do**
- 4: Initialize $i=1$;
- 5: **while** $i \leq m$ **do**
- 6: Populate \mathbf{X}^j using \mathcal{H} such that x_i^s between $[0, 1]$;
- 7: **end while**
- 8: Initialize a counter $\phi(\mathbf{X}^{(j)}) = 0$;
- 9: **if** $\mathbf{X}^j \in \mathcal{R}$ **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 $\widehat{\lambda(\mathcal{R})} = \frac{S}{n}$

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

In maybe mentioned that in line #6 of Algorithm 1, if we use an unbiased random number generator for \mathcal{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

³Chebyshev intervals $x_k = \cos\left(\frac{2k-1}{2n}\pi\right), k = 1, 2, \dots, n \forall x \in [0, 1]$ or any other non-repeating, monotonic sequence, see [37], may be used.

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

and other parameters.

The worst-case error in estimating the volume, by using deterministic samples, is given by [37] as:

$$\| \widehat{\lambda(\mathcal{R})} - \lambda(\mathcal{R}) \| \leq \frac{s(\mathcal{R})}{n^{1/m}} \quad (4)$$

where $s(\mathcal{R})$ is the surface area of the hyper-surface bounding the region \mathcal{R} .⁵ Therefore to have an absolute error no larger than ϵ , the number of evaluation points required is,

$$n(\epsilon) = \lceil \lceil \frac{s(\mathcal{R})}{\epsilon} \rceil^m \rceil \quad (5)$$

where, $\lceil x \rceil$ is the nearest integer greater or equal to (x) . From equation (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 \mathcal{R} , there is an exponential and unbounded⁶ 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.

2.2 Errors and sample size considerations

From equation (3) it can be seen that with increasing number of evaluation points n , the variance of the estimate $\widehat{\lambda(\mathcal{R})}$ decreases. It can be proved that

$$\Pr(\lim_{n \rightarrow \infty} \widehat{\lambda(\mathcal{R})} = \lambda(\mathcal{R})) = 1(a.s.). \quad (6)$$

which implies that we can get an error free estimate as the number of evaluation points tend to infinity. Furthermore by using the Chebyshev inequality⁷, we can prove (for more details see [37], chapter 2),

$$\lim_{n \rightarrow \infty} \Pr(|\widehat{\lambda(\mathcal{R})} - \lambda(\mathcal{R})| \geq \epsilon) = 0, \quad (7)$$

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,

$$\eta(\epsilon, \delta, \lambda) = \lceil \frac{\lambda \times (1 - \lambda)}{\delta \epsilon^2} \rceil, \lambda = \lambda(\mathcal{R}) \quad (8)$$

⁵The expressions in equation (4) are unit less because they have been derived from a counting argument.

⁶ With finer error tolerances it can be shown that n tends to ∞ .

⁷Chebyshev'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 $\Pr(\frac{\mathbf{Z}}{\sigma} \geq \beta) \leq \frac{1}{\beta^2}$.

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

$$\eta_c(\epsilon, \delta) = \lceil \frac{1}{4\delta\epsilon^2} \rceil \quad (9)$$

By comparing equations equation (9) and equation (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 (9) is independent of the dimension m of the hyperspace \mathcal{S}^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 $\mathcal{O}(m)$ for random sampling, whereas, for the deterministic method, the same step requires $\mathcal{O}(m^\beta)$, where, $\beta > 1$ and is dependent on the method used for generating sample points. For quadrature 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 $\mathcal{O}(m)$ for a certain preset (ϵ, δ) error tolerance as obtained from equation (9). For deterministic sampling, it can be proved (see [37]) that the complexity is exponential $\mathcal{O}\left(\frac{m^\beta}{4\delta\epsilon}\right)$ 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. [39] and references [37, 38]. 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. In all the simulations undertaken 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 Stamper et al. [34] 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` [40].

2.3 Representation of the workspace

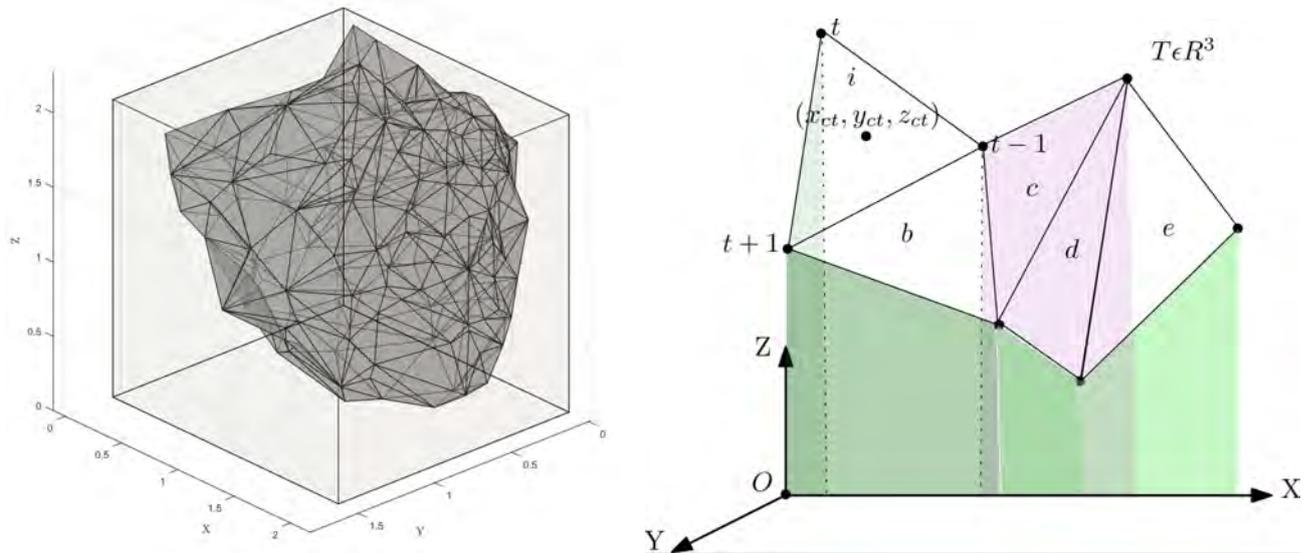
For manipulators with uninterrupted travel between joint limits for all active and passive joints, the workspace is bounded by continuous surfaces or by continuous level set curves (see, Merlet [29], Masory and Wang [20] and Tsai and Soni [21]) and therefore, the analytical evaluation of the workspace qualifies to be a problem of integration in 3D. In the Monte Carlo based approach, we first select a search space shown schematically by the light gray cube in figure 1a and let its volume be V_{Search} . Next, the *well-conditioned* workspace, defined in section 3 with volume $V_{Workspace}$, is obtained and this is shown schematically by the region bounded by dark triangles in

figure 1a. Since both of the search space and well-conditioned workspace are continuous regions in \mathbb{R}^3 , the probability that a randomly selected point in the search space, will also lie inside the *well-conditioned* workspace is,

$$P(\text{Point inside workspace}) = \frac{V_{\text{Workspace}}}{V_{\text{Search}}} = \frac{\eta_{\text{Total}}}{\eta_{\text{Random}}} \quad (10)$$

where, η_{Total} is the number of searched points inside $V_{\text{Workspace}}$ and η_{Random} is the total number of points populated in the search space. The expression in equation (10) is obtained from the simulation, and since V_{Search} is exactly known from the user input, we can find the volume of the *well-conditioned* workspace from equation (10).

Alternately, 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 convex hull of the cloud of points and triangulating them by well known Delaunay triangulation algorithms [41]– we have used `Matlab` [40] functions. After the 3D polyhedron has been obtained (as shown in figure 1a) we obtain the volume using a generalization of the trapezoidal rule in 3D (see Allgower and Schmidt [42, 43]). The algorithm considers the domain $D \in \mathbb{R}^3$ as a



(a) Relative representation of workspace and search space

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

Figure 1: Representation and calculation of workspace volume

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 1b) and the other faces parallel to an axis perpendicular to the plane (the Z axis in case of figure 1b). According to [43], the volume V of the trapezoid is,

$$V = (-1)^2 \sum_{\sigma \in T} \left[\left(\frac{1}{2} \sum_{i=1}^2 z_i^{ct2} \right) \cdot \frac{1}{2!} \det \begin{bmatrix} 1 & 1 & 1 \\ x_i^{t-1} & x_i^t & x_i^{t+1} \\ y_i^{t-1} & y_i^t & y_i^{t+1} \end{bmatrix} \right] \quad (11)$$

where the outermost summation $\sum_{\sigma \in T}$ indicates that the summation is carried over the total triangulated domain T . According to Allgower and Schmidt [43], this method is more efficient with

complexity $\mathcal{O}(n^2)$ as compared to other methods such as finding the sum of the volumes of discrete tetrahedra in which case the complexity is $\mathcal{O}(n^3)$. It may be noted that equation (11), 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.

2.4 Comparison between Monte Carlo, Chebyshev sampling and analytical methods to find the 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 2.

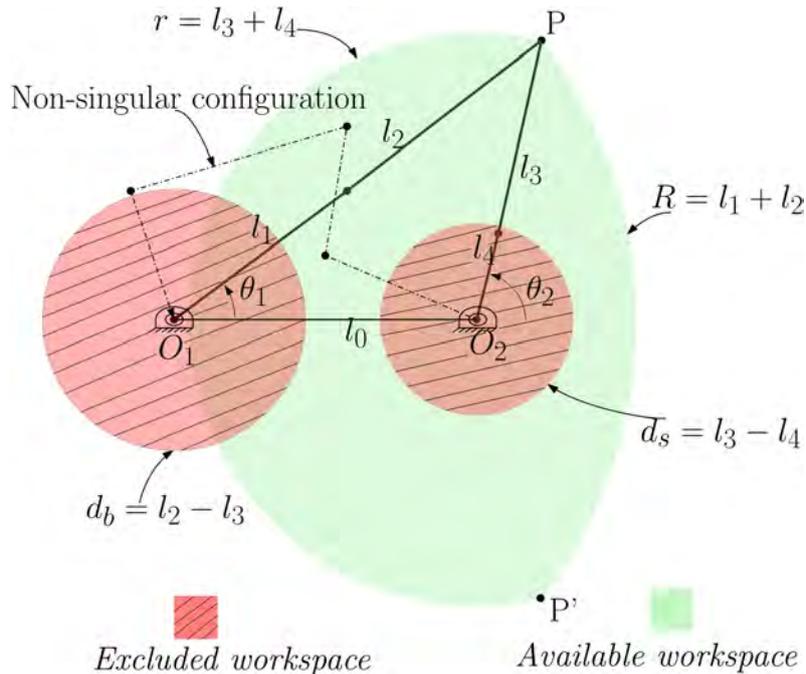


Figure 2: General schematic diagram of a 5R manipulator

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 2 is natural. In figure 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, [44],[45]). In figure 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 2 after [45] are

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

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

$$\begin{aligned} \mathcal{W}(\mathbf{L}) = & r^2 \cos^{-1} \left(\frac{1}{2} \frac{-R^2 + d^2 + r^2}{dr} \right) + R^2 \cos^{-1} \left(\frac{1}{2} \frac{R^2 + d^2 - r^2}{dR} \right) - \\ & \frac{1}{2} \sqrt{(-d + R + r)(d + r - R)(d - r + R)(d + R + r)} - \pi d_s^2 - d_b^2 \cos^{-1} \left(\frac{1}{2} \frac{-R^2 + d^2 + d_b^2}{dd_b} \right) - \\ & R^2 \cos^{-1} \left(\frac{1}{2} \frac{R^2 + d^2 - d_b^2}{dR} \right) + \frac{1}{2} \sqrt{(-d + R + d_b)(d + d_b - R)(d - d_b + R)(d + R + d_b)} \end{aligned} \quad (12)$$

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 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 (12), the area is obtained as 5.6123 unit². 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. [27] and Caro et al. [30]) to obtain the area.

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

$$x = \frac{1}{2}(b + a) + \frac{1}{2}(b - a) \frac{\cos(2j - 1)}{2k}, x \in [a, b], j = 1, 2, \dots, N \quad (13)$$

The computation time⁸ and the differences in the obtained areas (in unit²) from the different approaches are shown in tables 1 and 2. In the tables, *BCE* is the best case error in obtaining the

# Samples	Area using eq. 10	CPU Time I	BCE %
5000	5.58	0.12	0.575
10000	5.64	0.15	0.494
150000	5.608	0.36	0.059

Table 1: Computational performance of a fully parallel MC method

area of the workspace (across 10 trials), CPU Time I is the time required to obtain the LHS of equation (10), and CPU Time II is the time required to obtain the LHS of equation (11). From tables 1 and 2, we can conclude that Monte Carlo method is faster and more accurate than deterministic sampling for obtaining the workspace area.

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 3a. We consider a constraint on the joint θ_2 of the form

⁸The programs were run in Matlab R2015a[40] on a Windows 7 PC with an quad core (3.10GHz) Intel XEON & 16 GB of RAM.

Intervals	Value of K in eq. 13	Area using eq. 11	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 2: Computational performance of Chebyshev interval samples

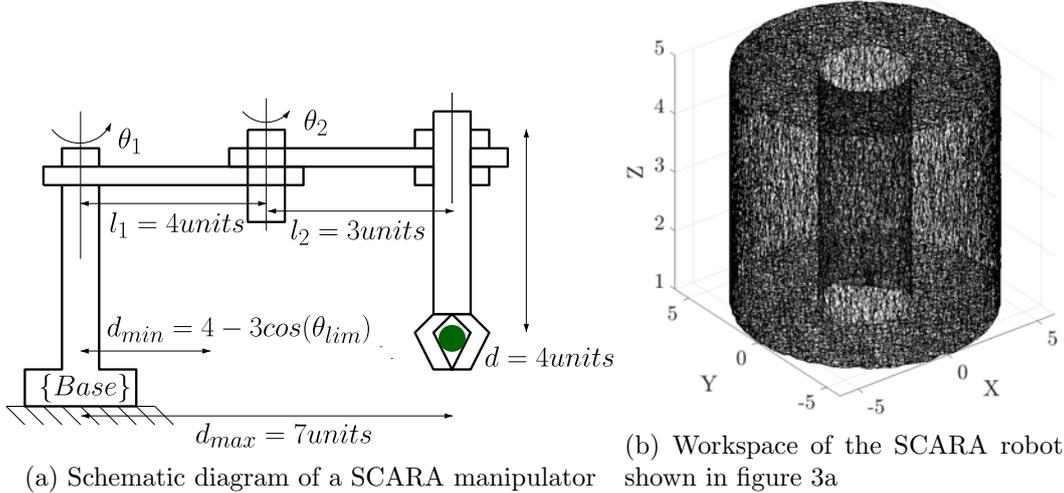


Figure 3: Schematic representation and workspace of a SCARA manipulator

$[10^\circ \leq \theta_2 \leq 2\pi - 10^\circ]$ 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

$$\mathcal{W} = \pi(d_{\max}^2 - d_{\min}^2)d = \pi(49 - (4 - 3 \cos(10^\circ))^2)4 = 602.01 \text{ unit}^3 \quad (14)$$

The inverse kinematics equations of the SCARA manipulator is given as

$$\begin{aligned} d &= -Z \\ \theta_2 &= \cos^{-1}((X^2 + Y^2 - l_1^2 - l_2^2)/(2l_1l_2)) \\ \theta_1 &= \text{atan2}(Y, X) - \text{atan2}(l_2 \sin(\theta_2), (l_2 + l_2 \cos(\theta_2))) \end{aligned} \quad (15)$$

Next, we populate points in \mathfrak{R}^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 \mathfrak{R}^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 and 4. In

# Samples	Volume using eq. 10	CPU Time I	Volume using eq. 11	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
$3.375 \times 10E6$	599.3	10.42	589.92	118.75	0.4615

Table 3: Computational performance of a fully parallel MC method

table 3 CPU Time I is the time required to obtain the left-hand side of equation (10), and CPU Time II is the time required to obtain the left-hand side of equation (11).

Intervals	Value of k in eq. 13	Volume using eq. 11	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 4: Computational performance of Chebyshev interval samples

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) 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.

2.5 Comparison between Monte Carlo and interval analysis based methods

Caro et al.[30] 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 [46]) 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 (see section 4 for details). 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 time⁹ for obtaining the workspace by searching through 69,612 boxes with *precision* of 0.1 is 38 seconds. In section 2.4 we obtain a computation time of 0.36 seconds for searching through 150,000 points in a search space and the best case error was 0.059%.

Merlet and co-workers[25]-[29] 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

⁹The 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.

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 (18)).
- 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. In our approach, we use constraint sensitivity analyses at the optimum point (see sections 4.1.5 and 4.2.4) to identify the effects of *each* of the individual design parameters to the workspace of the manipulator.
- A related work (see Chablat et al. [27]) 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 (18)), and hence the computational load is much less.

In summary, from sections 2.4 and 2.5 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. We use the Monte Carlo method for the design of closed-loop mechanisms and parallel manipulators in this work. In the next section, we define the well-conditioned workspace of the manipulator and formulate the optimization problem.

3 Formulation of the optimization problem

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 as

$$\mathcal{W}_T = \mathcal{W}_p \times \mathcal{W}_o, \quad \mathcal{W}_p \in \mathfrak{R}^3, \quad \mathcal{W}_o \in SO(3) \ \& \ \mathcal{W}_T \in SE(3) \quad (16)$$

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 ([49, 50]). In this work, we will loosely follow the work by Stamper et al. [34] and define well condition workspace as the closed volume in \mathfrak{R}^3 which is a subset of \mathcal{W}_p in equation (16) 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.

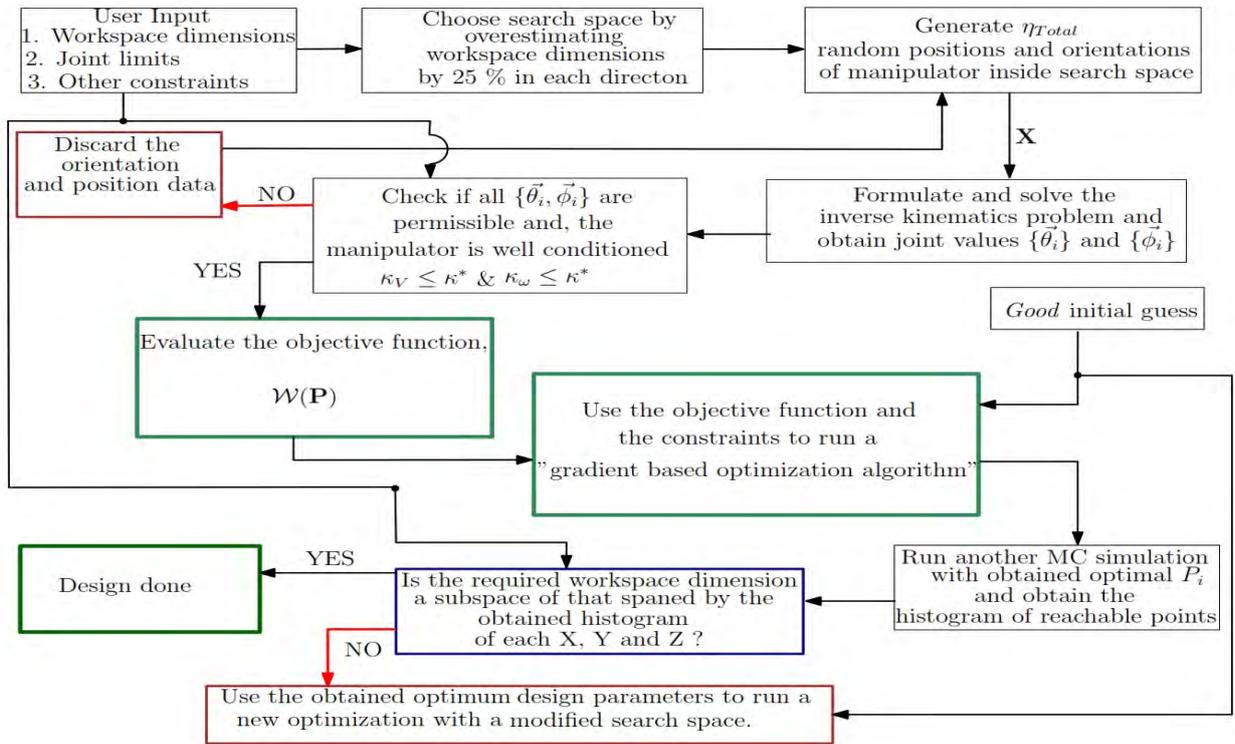


Figure 4: Flowchart of the design process

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. As shown in Ghosal [47] and Ghosal and Ravani [48], 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 Appendix A and section 4.2.3).

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

$$\kappa = \| \mathbf{A} \| \| \mathbf{A}^{-1} \| \quad (17)$$

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

Denoting the equivalent Jacobians by \mathbf{J}_{equ}^V and \mathbf{J}_{equ}^ω (see appendix A), we use these in the above equation to define a condition number for the equivalent Jacobians and denote them by κ_V and κ_ω . To ensure *well-conditioned-ness* of the manipulator throughout its 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

$$\max\{\kappa_V, \kappa_\omega\} \leq \kappa^* \quad (18)$$

In the numerical examples, we have used κ^* as 100 as this gives a conservative estimate of the well-conditioned workspace. A larger κ^* will give a larger well-conditioned workspace (see also figure 12 for the increase in workspace with increasing κ^*).

We observe that the Monte Carlo algorithm proposed in algorithm 1 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 , $\forall i = 1, 2, \dots, n$. To classify a given position and orientation of the end-effector of the manipulator (denoted by $\mathbf{X} \in SE(3)$) we formulate an *in-out* function $\mathcal{F}(\mathbf{X})$ which takes input of the position and orientation of the end-effector of the manipulator and classifies the given input according to its existence inside or outside the well-conditioned workspace of the parallel manipulator. The function can be symbolically written as

$$\mathcal{F}(\mathbf{X}) = \begin{cases} 1 & \text{if } \mathbf{X} \in \mathcal{W} \\ 0 & \text{if } \mathbf{X} \notin \mathcal{W} \end{cases} \quad (19)$$

A ratio of the cumulative value of $\mathcal{F}(\mathbf{X})$ obtained by using equation (19) (given as η_{in} in equation (20)) to the total number of points (given as η_{Total} in equation (20)) simulated in the search space ($V_{Search\ space}$ in equation (20)) gives an estimate to the probability of a randomly generated position and orientation of the end-effector of the manipulator to be in the well-conditioned workspace of the manipulator. Since the values of the vector \mathbf{X} are sampled from an uniform distribution in $V_{Search\ space}$, the total volume of the well-conditioned workspace can be calculated by equation (20) below

$$\mathcal{W}_T = \frac{\eta_{in}}{\eta_{Total}} \times V_{Search\ space} \quad (20)$$

Using equation (20) we formulate the optimization problem as

$$\text{Maximize } \mathcal{W}(\mathbf{P}) \quad (21)$$

$$\begin{aligned} \text{Subject to, } & \vec{\theta}_{min} \leq \vec{\theta} \leq \vec{\theta}_{max} \\ & \vec{\phi}_{min} \leq \vec{\phi} \leq \vec{\phi}_{max} \end{aligned}$$

Other geometric equality and inequality constraints

where $\vec{\theta}$ is the vector containing the actuated joint variables and $\vec{\phi}$ is the vector containing the un-actuated joint variables, and \vec{P} is the set of parameters uniquely representing the geometry of the manipulator. The optimization procedure is outlined in figure 4.

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 (19) 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 θ_i , $\forall i = 1, 2, \dots, n$ are the ' n ' actuated joint variables and ϕ_j , $\forall j = 1, 2, \dots, m$ 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-conditioned¹⁰. The Monte Carlo simulation as discussed in this section and the algorithm in figure 4 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 \mathfrak{R}^3 and $SO(3)$.

4 Illustrative examples

In this section we will use the methodology developed above to perform dimensional 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. In the first example, the direct kinematics is easily solved and hence the optimization using the direct kinematics equations can be compared with the Monte Carlo based method. In the second example the direct kinematics problem is fairly difficult to solve and we use the Monte Carlo method which does not use the solution of the direct kinematics problem.

4.1 Example 1: 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 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 2.4 (see also the works by Macho et al. [45], Cerventes-Sanchez et al. [44], and Liu et al. [49]). 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 5. The

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

Table 5: Generic constraints for the manipulator

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. [49], 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 (12) along with geometric constraints to form the objective function and use a gradient based optimization method to analyze the four different workspace configurations

¹⁰We have used a definition of the condition number, which encompasses both linear and angular motion of the manipulator at the said position and orientation, as given in appendix A. The well-conditioned-ness is ensured by restricting the condition numbers to be less than a chosen κ^* at all times.

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 appendix B.

From Algorithm 1, the equivalent definition of workspace is $\mathcal{W} = \frac{\eta_{Possible}A}{\eta_{Total}}$. For a *unit* rectangle *completely* enveloping the possible workspace denoted by the plain zone and excluded workspace as the hatched zone. In figure 2, since the numbers generated by the *rand* command is uniformly distributed on the envelope area the probability that a point generated lies *inside* the feasible zone is tends to A .

4.1.1 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 5a. For this case, the constraints are,

$$\begin{aligned} l_1 + l_2 - d &< 0 \\ l_3 + l_4 - d &< 0 \\ d_s &< d - (l_1 + l_2) \\ d_b &< d - (l_3 + l_4) \end{aligned} \quad (22)$$

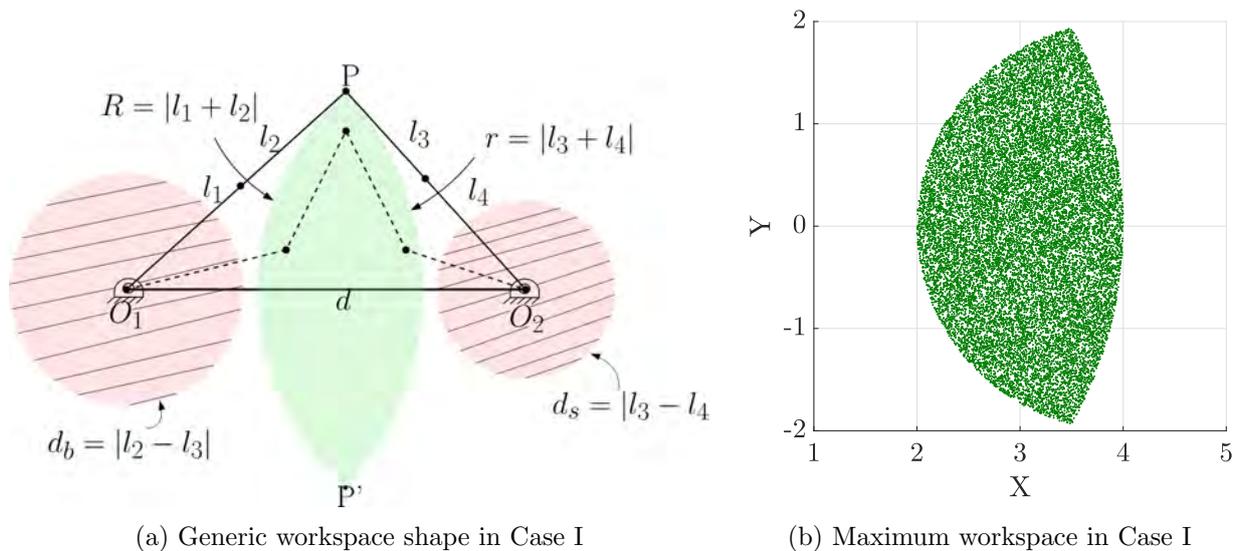


Figure 5: Workspace bounded by two continuous circular arcs

The result of the optimization problem with the constraints listed in table 5 and in equation (22) yielded the results given in table 6. The maximum workspace is shown in figure 5b.

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 6: Optimal solutions for Case I

In this case we observe the following:

- Both the solutions given in table 6 are the same, i.e., one is the reflection of the other about the perpendicular line from P to O_1O_2 in figure 5a.

- 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 5b) about the line $x = 4$.

This case has been analytically solved in section 2.3.

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

The shape of the workspace is as shown in figure 6a. For this case, the general constraints are

$$\begin{aligned}
 l_1 + l_2 - d &> 0 \\
 l_3 + l_4 - d &> 0 \\
 d_s &< d - (l_1 + l_2) \\
 d_b &< d - (l_3 + l_4)
 \end{aligned} \tag{23}$$

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 .

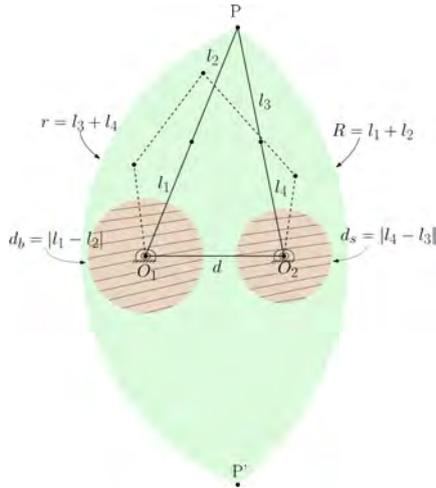
$$l_1 - l_2 + l_4 - l_3 - d < 0 \tag{24}$$

The result of the optimization problem attempted with general constraints from table 5, the constraints from equation (23) and the additional constraint from equation (24) yielded the results given in table 7.

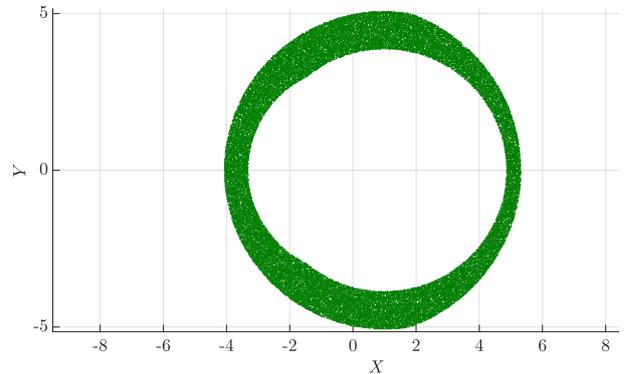
Initial guesses for {L}	Optimum values for {L}	Workspace area in unit ²	Constraints
[3, 3, 2, 2, 3]	[1.00, 1.00, 4.31, 2.68, 1.00]	25.04	Eq. 23
[3, 3, 2, 2, 3]	[1.25, 3.74, 1.05, 1.26, 2.68]	24.95	Eqs. 23 & 24

Table 7: Optimal solutions for Case II

The maximum closed and bounded workspace obtained by using the proposed optimization algorithm is shown in figure 6b. 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 6: Workspace bounded by 2 circular arcs outside and 2 circular arcs inside

4.1.3 Case III: Workspace bounded by 4 circular arcs

In this case the shape of the workspace is as shown in figure 7b. For this case, the general constraints are

$$\begin{aligned}
 l_1 + l_2 &< d_s + d \\
 l_3 + l_4 &< d_b + d \\
 l_1 + l_2 &> d - d_s \\
 l_3 + l_4 &> d - d_b
 \end{aligned} \tag{25}$$

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 8: Optimal solutions for Case III

The maximum workspace obtained after optimization is shown in figure 7b and obtained link lengths are given in table 8. It maybe mentioned that the initial guesses used are from the work by Cerventes-Sanchez et al. [44], where this case has been analyzed.

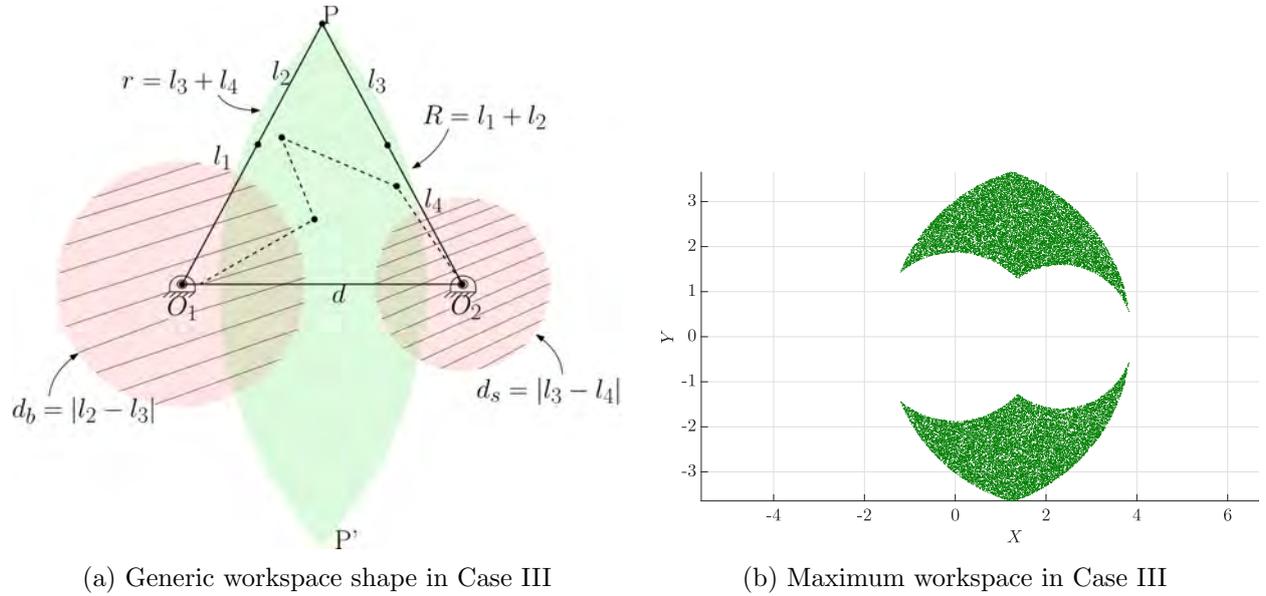


Figure 7: Workspace bounded by 4 circular arcs

4.1.4 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 2. The constraints for the case are as follows:

$$\begin{aligned}
 l_1 + l_2 &> d + d_s \\
 l_3 + l_4 &< d + d_b \\
 d_s &< l_1 + l_2 - d \\
 d_b &> l_3 + l_4 - d
 \end{aligned} \tag{26}$$

The results of the optimization problem, after using the constraints from equation (26) are given in table 9. The two optimum workspaces for the manipulator are given in figure 8.

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 9: Optimal solutions for Case IV

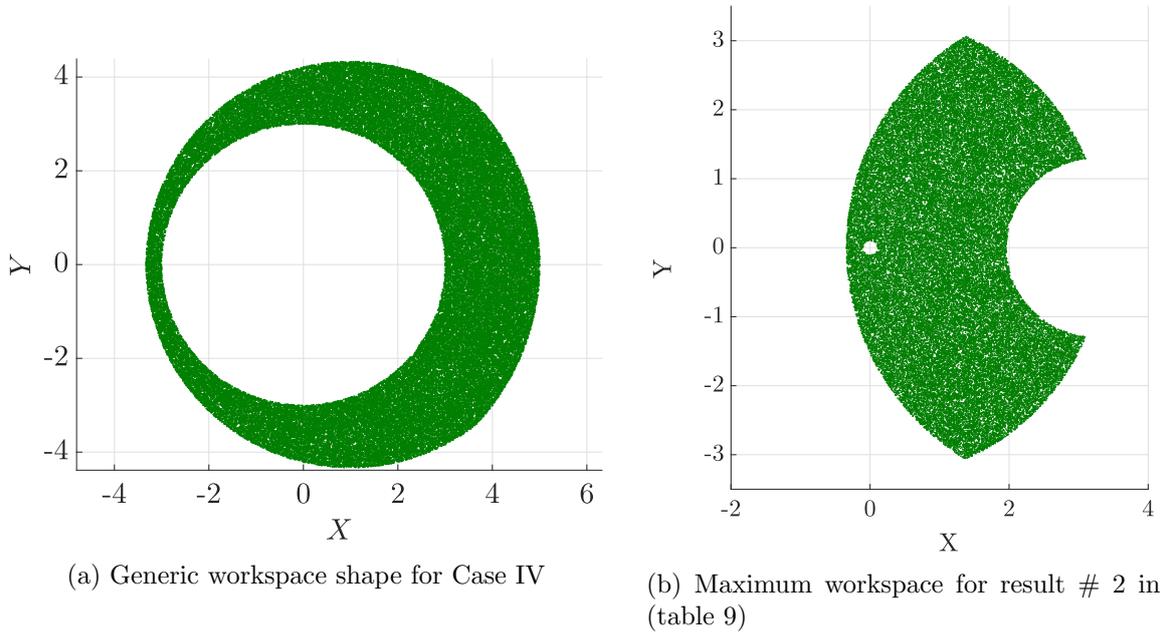


Figure 8: Workspace bounded by 3 circular arcs with a complete circular exclusion zone.

4.1.5 Conclusions and sensitivity analysis

For this particular problem, we found that the largest workspace was obtained for *Case IV* with constraints in equation (26). 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 7. It is known (see chapter 4 of Arora [53]) 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 parameters 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 (23) 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 5 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 $l_4 > 1$.

4.1.6 Dimensional synthesis of a 5R mechanism for a required workspace

In all the cases studied above, the coordinates of O_1 in figures 2, 5a, 6a and 7a, 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 5b, 6b, 7b, 8a and 8b 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 unit².

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 8a 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: $X_d \in [-4, 4]$, $Y_d \in [2, 4]$

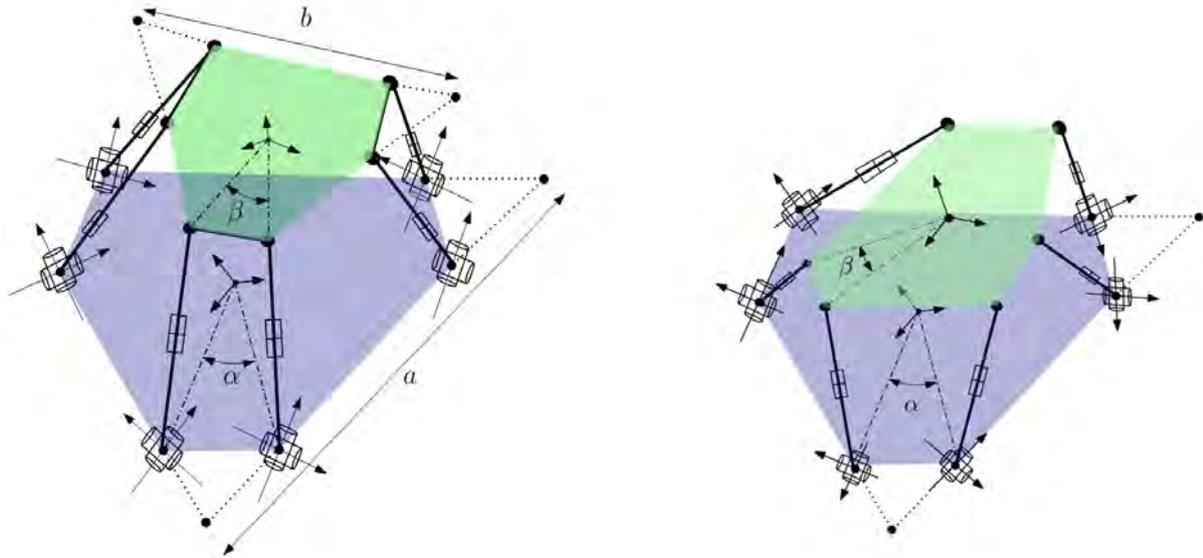
For this we undertook the optimization procedure once more with a relaxed boundary of $X_s \in [-4.5, 4.5]$, $Y_s \in [1.5, 4.5]$. The following results given in table 10 were obtained with the binding constraints in table 5. It may be noticed that wanted workspace is symmetric about the Y axis therefore, 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 10: Results of the design problem

4.2 Example 2: 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 Mruthyunjaya [16]



(a) SRSPM in the *normal* configuration

(b) SRSPM in the *crossed* configuration

Figure 9: Two configurations of the SRSPM

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 [51]. The design of the SRSPM for a desired workspace was attempted by Chatterjee and Ghosal [52], where the authors have used a predefined search path to search through the parameter space and attain an optimal result. Lou et al. [33] 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 9 gives a larger workspace with joint limits? Or, in other words, which of the two configurations is better in terms of kinematics and workspace?

4.2.1 Description of the manipulator

As shown in figure 10a, 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 $R_{ab} = \frac{b}{a}$, 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.

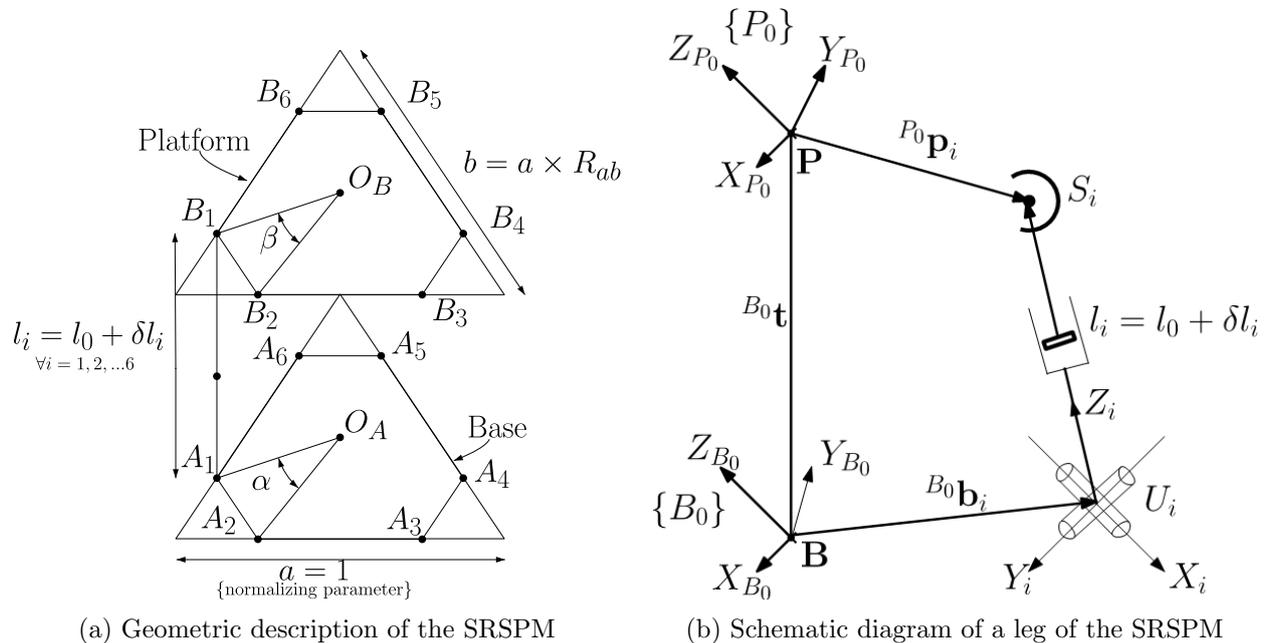


Figure 10: Geometric description and schematic diagram of the SRSPM

In figure 10b, 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,

$${}^{B_0}\mathbf{p}_i = {}^{B_0}_{P_0} [R] {}^{P_0}\mathbf{p}_i + {}^{B_0}\mathbf{t} \quad (27)$$

From equation (27), the location of the i^{th} spherical joint from the base can be given as:

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

4.2.2 Inverse kinematics of the SRSPM

The solution to the inverse kinematics problem of the manipulator is well known and can be 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 10a and 10b, 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 equations are obtained as

$$[R(\hat{\mathbf{Z}}, \alpha_i)]^T [(x, y, z)^T - {}^{B_0}\mathbf{b}_i] = l_i \begin{pmatrix} \sin(\phi_i) \cos(\psi_i) \\ -\sin(\psi_i) \\ \cos(\phi_i) \cos(\psi_i) \end{pmatrix} \quad (28)$$

Using equation (28), 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 also Appendix A). 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.

4.2.3 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 [52] and Lou et al. [33]) 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,

$$\begin{aligned} & \text{Minimize } \mathcal{W}(\alpha, \beta, R_{ab}, l_0, \delta l) & (29) \\ & \text{Subject to, } a = 1 \\ & l_i + \delta l_i = 1.8, \quad \forall i = 1, 2, \dots, 6 \\ & \frac{\pi}{12} < \alpha < \frac{\pi}{4} \quad \& \quad \frac{\pi}{12} < \beta < \frac{\pi}{4} \\ & 0.5 < R_{ab} < 1.0 \quad \& \quad l_i > 0 \quad \& \quad \delta l_i > 0 \end{aligned}$$

Out of the three angles ($\theta_i, \zeta_i, \eta_i$) in a S joint, the range of η_i , measured about the vector $\overrightarrow{U_i S_i}$ in figure 10b 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 11. It may be noted that S joint values (θ_i and ζ_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. [51] 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 11: Un-actuated joint limits for SRSPM

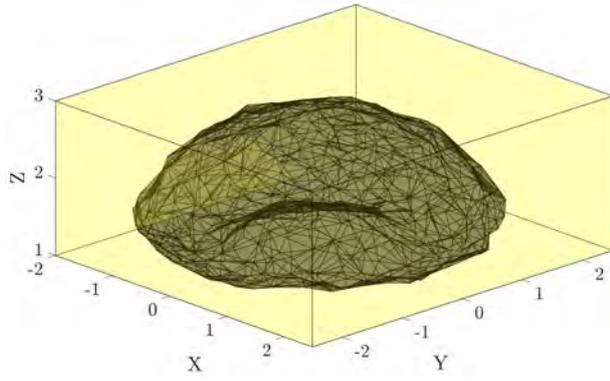
In this example the starting guesses are the vectors given in equations (30) and (31) and the search space was chosen as $X \in [-2.5, 2.5]$, $Y \in [-2.5, 2.5]$ and $Z \in [1, 3]$ to capture the entire workspace of the SRSPM with normalized dimensions.

$$(\alpha, \beta, R_{ab}, l_0, \delta l) = \{1.2, 0.8, 0.3, 1, 0.8\} \text{ for normal configuration} \quad (30)$$

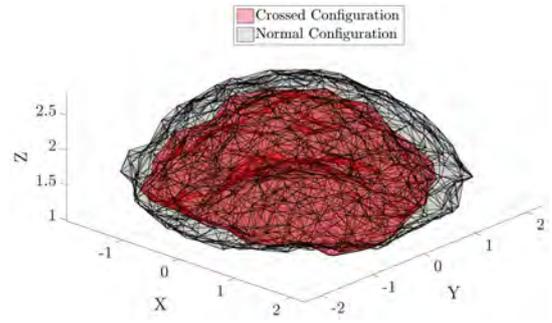
$$(\alpha, \beta, R_{ab}, l_0, \delta l) = \{0.8, 0.8, 0.36, 1.4, 0.4\} \text{ for crossed configuration} \quad (31)$$

The results of the optimization problem (equation (29)) is shown in table 12 below.

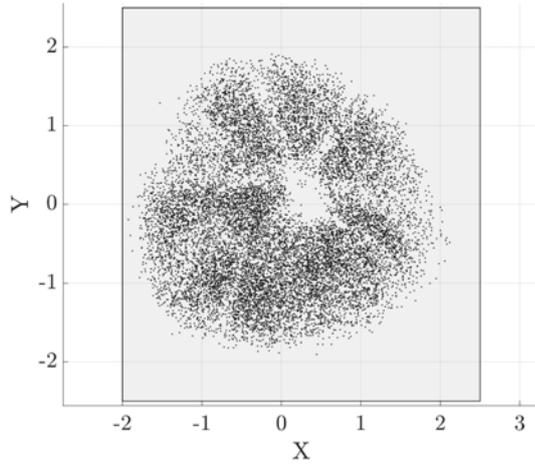
Figure 11 shows the available workspaces for the SRSPM. In figure 11b a comparison of the available workspaces of the SRSPM in normal configuration (indicated in gray) and crossed configuration (in yellow) is given. As shown in table 12, the workspace of the manipulator is significantly



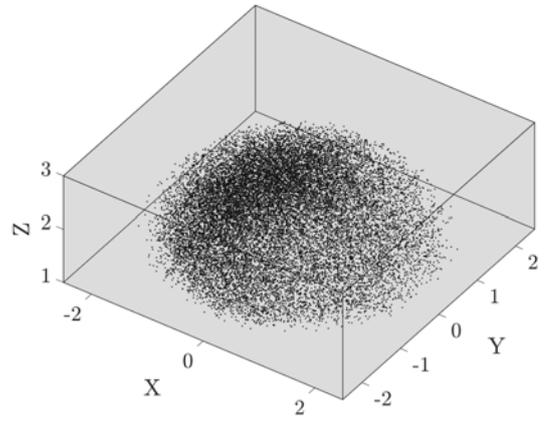
(a) Available workspace for the normal configuration – outer box indicates search space



(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 11: Workspaces of the optimum SRSPMs obtained for a given search space

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 12: Optimum design parameters for the SRSPM

less in crossed configuration than in the normal configuration. Additionally, from figure 11c one can observe that the workspace of the SRSPM is not continuous and there are ‘holes’ inside the well-conditioned workspace.

In equation (18) 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 12 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.

4.2.4 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 the

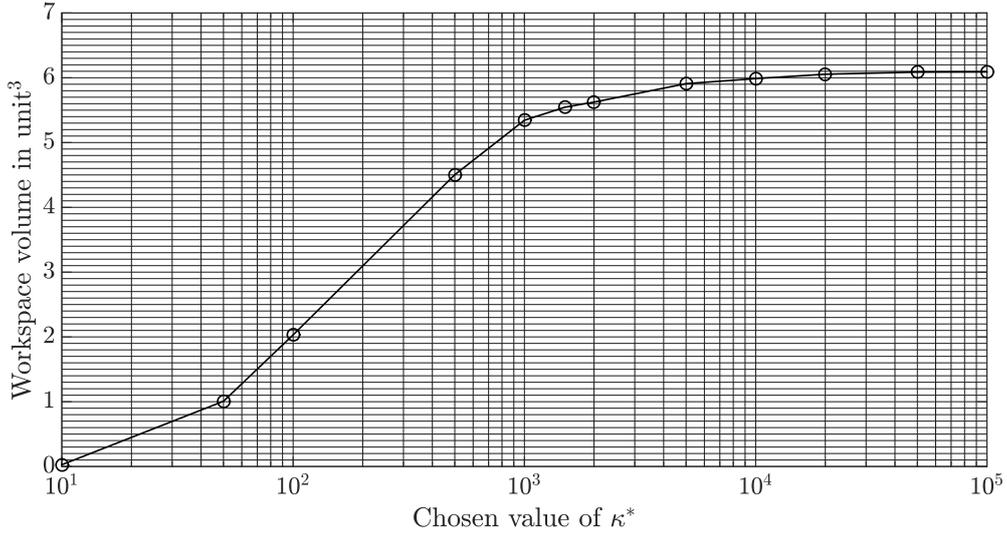
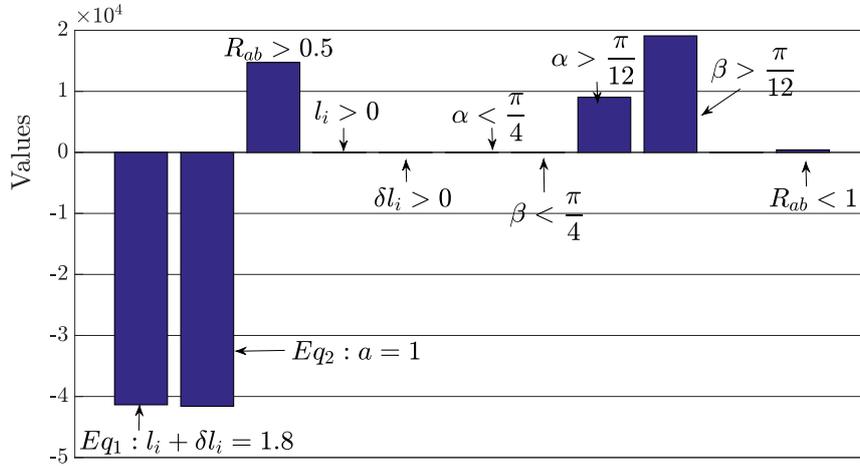
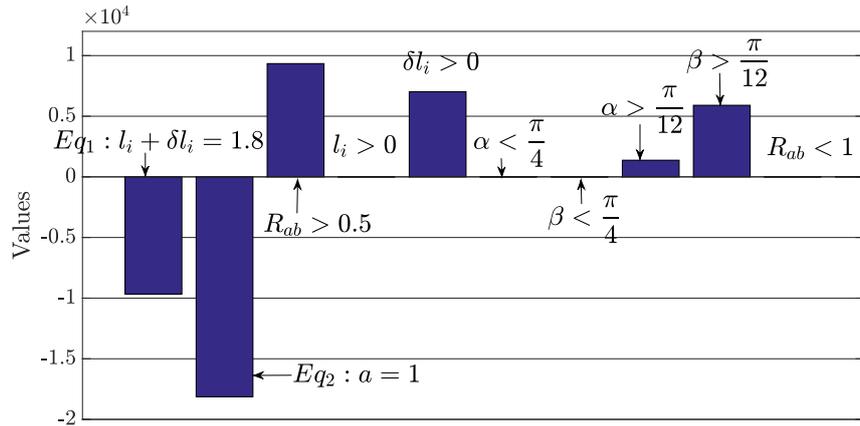


Figure 12: Dependence of well-conditioned workspace on chosen κ^* for the SRSPM in normal configuration



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



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

Figure 13: Lagrange multipliers for the optimization problem at *an* optimum

Lagrange multipliers for the problem have been extracted and plotted in figure 13 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 constraints have negative Lagrange multipliers and the inequality constraints have positive valued multipliers.

- In each case the equality constraints, i.e., $a = 1$ and $l_i + \delta l_i = 1.8 \forall i = 1, 2, \dots, 6$ 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.
- 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 [51], Fichter has used a triangular top platform to analyze the SRSPM.

4.2.5 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 12) in section 4.2.3. 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 unit³, the search space is 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. The parameters giving the maximum workspace for both configurations are given in table 12. The histogram obtained for the optimal SRPSM, using a Monte Carlo simulation, is shown in figure 14c. 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 unit³ which is somewhat less than the desired workspace volume.

To obtain the desired larger volume, we observe from figure 14a that the workspace of the synthesized Stewart platform manipulator is *almost* equally distributed around the Z axis (also

observed by Masory and Wang [20]). To increase the volume, we can modify the equality constraints involving a $l_0 + \delta l$. From the sensitivity analysis in section 4.2.4, 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 unit³, 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 unit³ 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 14a. 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.

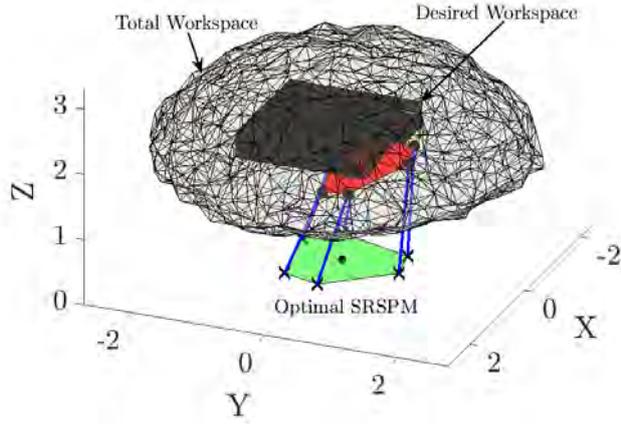
Figure 14b shows the *orientation* workspace of the optimal manipulator at the center of the *well-conditioned* workspace shown in figure 14a. As shown in figure 10a, θ, ϕ and ψ are the Euler angles about X, Y and Z axes indicating the orientation of the top platform P with respect to the base B.

5 Conclusion

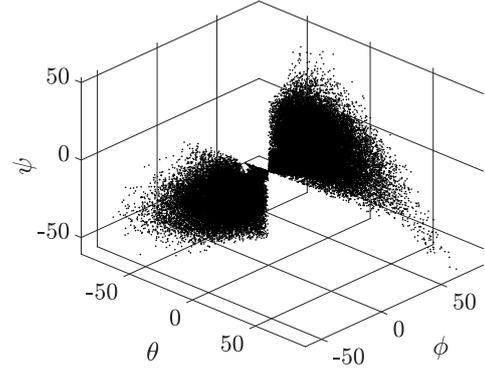
In this work we show that the Monte Carlo method is an efficient tool for finding the volume of the workspace of a manipulator and it is computationally more efficient than other deterministic search methods used in literature. In this work we have used a definition of the well-conditioned workspace for the translational and angular motion of the end-effector of the manipulator. Through section 2 we have shown that Monte Carlo method is an efficient method to solve for the well-conditioned workspace area and volumes of manipulators. In section 3, 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. In section 4 we have presented demonstrative examples of optimization of parallel manipulators. The first example in section 4.1 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 4.2 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.

A Appendix A: Constraints for an SRSPM and definition of equivalent Jacobian

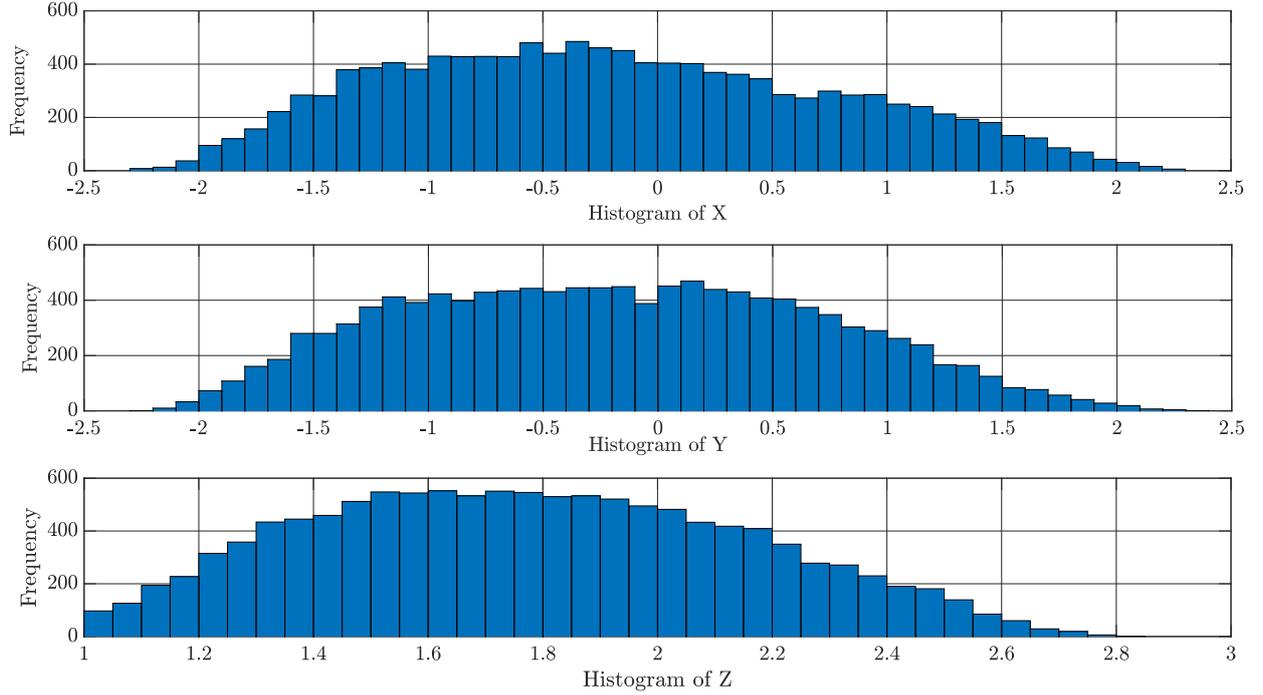
The SRSPM is a six degree-of-freedom parallel manipulator. The six actuated joints are the prismatic joints in each leg and the six universal (U) and six spherical (S) joints are passive. As shown in Ghosal [47], we can derive 12 constraint equations which can be used to solve the angles ϕ_i and ψ_i ($i = 1, \dots, 6$) in the U joints. The first six constraint equations are derived from the fact that



(a) Total workspace of the SRSPM



(b) Orientation workspace of the SRSPM (degrees)



(c) Histogram of the total workspace of the SRSPM

Figure 14: Workspaces and histogram of the SRSPM in *normal* configuration

the distance between two consecutive S joint is fixed. We get

$$|{}^{B_0}\mathbf{p}_i - {}^{B_0}\mathbf{p}_{i+1}|^2 = |\mathbf{S}_i - \mathbf{S}_{i+1}|^2 \quad (32)$$

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

$$\left. \begin{aligned} |{}^{B_0}\mathbf{p}_1 - {}^{B_0}\mathbf{p}_3|^2 &= |\mathbf{S}_1 - \mathbf{S}_3|^2 \\ |{}^{B_0}\mathbf{p}_1 - {}^{B_0}\mathbf{p}_4|^2 &= |\mathbf{S}_1 - \mathbf{S}_4|^2 \\ |{}^{B_0}\mathbf{p}_1 - {}^{B_0}\mathbf{p}_5|^2 &= |\mathbf{S}_1 - \mathbf{S}_5|^2 \end{aligned} \right\} \quad (33)$$

$$\left. \begin{aligned} ({}^{B_0}\mathbf{p}_1 - {}^{B_0}\mathbf{p}_3) \times ({}^{B_0}\mathbf{p}_1 - {}^{B_0}\mathbf{p}_4) \cdot ({}^{B_0}\mathbf{p}_1 - {}^{B_0}\mathbf{p}_2) &= 0 \\ ({}^{B_0}\mathbf{p}_1 - {}^{B_0}\mathbf{p}_4) \times ({}^{B_0}\mathbf{p}_1 - {}^{B_0}\mathbf{p}_5) \cdot ({}^{B_0}\mathbf{p}_1 - {}^{B_0}\mathbf{p}_3) &= 0 \\ ({}^{B_0}\mathbf{p}_1 - {}^{B_0}\mathbf{p}_5) \times ({}^{B_0}\mathbf{p}_1 - {}^{B_0}\mathbf{p}_6) \cdot ({}^{B_0}\mathbf{p}_1 - {}^{B_0}\mathbf{p}_4) &= 0 \end{aligned} \right\} \quad (34)$$

It 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

$$[\mathbf{K}(\mathbf{l}, \gamma)]\dot{\mathbf{l}} + [\mathbf{K}^*(\mathbf{l}, \gamma)]\dot{\gamma} = 0 \quad (35)$$

It is easily seen that $[\mathbf{K}^*]$ is a square matrix of dimension 12×12 . Equation (35) can be solved for $\dot{\gamma}$, given $\det(\mathbf{K}^*) \neq 0$ ¹¹, and we can obtain

$$\dot{\gamma} = -[\mathbf{K}^*]^{-1}[\mathbf{K}]\dot{\mathbf{l}} \quad (36)$$

The position vector of the center of the platform in figure 10a is given by,

$${}^{O_A}P_{O_B} = \frac{1}{6} \sum_{i=1}^3 {}^{O_A}B_i \quad (37)$$

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

$${}^{B_0}_{P_0}[R] = \left[\begin{array}{c|c} \frac{{}^{O_A}B_1 - {}^{O_A}B_3}{{}^{O_A}B_1 - {}^{O_A}B_3} & \hat{Y} \\ \hline \frac{({}^{O_A}B_1 - {}^{O_A}B_5) \times ({}^{O_A}B_1 - {}^{O_A}B_3)}{|({}^{O_A}B_1 - {}^{O_A}B_5) \times ({}^{O_A}B_1 - {}^{O_A}B_3)|} & \end{array} \right] \quad (38)$$

where \hat{Y} is obtained by the cross product of the third and first column of the matrix in equation (38).

By differentiating the expressions for the position and orientation of the end-effector obtained from equations (37) and (38) and partitioning the expression for actuated and passive joints, we have,

$${}^{B_0}V_{P_0} = [\mathbf{J}_V]\dot{\mathbf{l}} + [\mathbf{J}_V^*]\dot{\gamma} \quad (39)$$

$${}^{B_0}\omega_{P_0} = [\mathbf{J}_\omega]\dot{\mathbf{l}} + [\mathbf{J}_\omega^*]\dot{\gamma} \quad (40)$$

Using equation (36) in equations (39) and (40) we obtain

$${}^{B_0}\mathbf{V}_{P_0} = ([\mathbf{J}_V] - [\mathbf{J}_V^*][\mathbf{K}^*]^{-1}[\mathbf{K}])\dot{\mathbf{l}} \quad (41)$$

$${}^{B_0}\omega_{P_0} = ([\mathbf{J}_\omega] - [\mathbf{J}_\omega^*][\mathbf{K}^*]^{-1}[\mathbf{K}])\dot{\mathbf{l}} \quad (42)$$

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

$$\mathbf{J}_{eqv}^V = (\mathbf{J}_V - \mathbf{J}_V^*[\mathbf{K}^*]^{-1}[\mathbf{K}]) \quad (43)$$

$$\mathbf{J}_{eqv}^\omega = (\mathbf{J}_\omega - \mathbf{J}_\omega^*[\mathbf{K}^*]^{-1}[\mathbf{K}]) \quad (44)$$

The equivalent Jacobian matrices are used to obtain the κ_V and κ_ω in section 3.

¹¹In 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 Appendix B: 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 :

$$\text{Min}_{\{\mathbf{L}\}} A(\mathbf{L}) \quad (45)$$

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

$$Ie_1 : l_3 + l_4 - d + d_b < 0$$

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 2 and the formula for workspace in (equation (12)) is always valid. The problem is known to be feasible because it has been successfully attempted numerically (see for example Huang [54]). In this appendix we show that it is a non-convex problem.

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

$$\mathcal{L} = A(\mathbf{L}) + \lambda E_1 + \mu Ie_1 \quad (46)$$

Following Boyd and Vandenberghe [55], we use the second-order condition for testing non convexity. The condition states that the function \mathcal{L} in equation (46) is strictly convex if the Hessian $[\mathcal{H}(\mathcal{L})]_{i,j} = \frac{\partial^2 \mathcal{L}}{\partial L_i \partial L_j}$ 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 13. The data for the design variable $\{\mathbf{L}\}$ has been used from [44] for the first two cases and from [54] for the last case. The cause of rank deficiency is equivalent dependence of the objective function

$\{\mathbf{L}\} = [d, l_1, l_2, l_3, l_4]$	Cause of rank deficiency	Form of A from eq. 12
{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 13: Reason for rank deficiency of \mathcal{H}

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 (46) does not have unique components in each of the parameter space directions. We have

$$\nabla \mathcal{L} = \frac{\partial \mathcal{L}}{\partial d} \Big|_{\mathbf{P}} \widehat{e}_d + \frac{\partial \mathcal{L}}{\partial l_1} \Big|_{\mathbf{P}} \widehat{e}_{l_1} + \frac{\partial \mathcal{L}}{\partial l_2} \Big|_{\mathbf{P}} \widehat{e}_{l_2} + \frac{\partial \mathcal{L}}{\partial l_3} \Big|_{\mathbf{P}} (\widehat{e}_{l_3} + \widehat{e}_{l_4}) \quad (47)$$

From equation (47) it is clear that the 4th & the 5th 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 (45) given by equation (12) is dependent of the constraints in equation (45), 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.

Acknowledgment

The first author would like to thank Dr. J-P. Merlet and Prof. Sandipan Bandyopadhyay for discussions which led to significant improvements of the paper. We also thank the anonymous reviewers for their comments which have improved the paper.

References

- [1] Innocenti C and Parenti-Castelli V. Echelon form solution of direct kinematics for the general fully-parallel spherical wrist. *Mechanism and Machine Theory*, 28(4):553–561, 1993.
- [2] Ashith Shyam R B and Ghosal A. Three-degree-of-freedom parallel manipulator to track the sun for concentrated solar power systems. *Chinese Journal of Mechanical Engineering*, 28(4):793–800, 2015.
- [3] Salisbury J K and Craig J J. Articulated hands force control and kinematic issues. *The International Journal of Robotics Research*, 1(1):4–17, 1982.
- [4] Jacobsen S, Iversen E, Knutti D, Johnson R and Biggers K. Design of the Utah/MIT dexterous hand. In *Proceedings of IEEE ICRA 1986*, volume 3, pages 1520–1532. IEEE, 1986.
- [5] Borràs J and Dollar A M. Dimensional synthesis of three-fingered robot hands for maximal precision manipulation workspace. *The International Journal of Robotics Research*, 34(14):1731–1746, 2015.
- [6] Phantom premium product range. Haptic products by Geomagic Inc. 3D Systems. Morrisville, North Carolina, USA.
- [7] Wen F and Liang C. Displacement analysis of the 6-6 Stewart platform mechanisms. *Mechanism and Machine Theory*, 29(4):547–557, 1994.
- [8] Raghavan M. The Stewart platform of general geometry has 40 configurations. *Trans. ASME, Journal of Mechanical Design*, 115(2):277–282, 1993.
- [9] Bandyopadhyay S and Ghosal A. Analysis of configuration space singularities of closed-loop mechanisms and parallel manipulators. *Mechanism and Machine Theory*, 39(5):519–544, 2004.

- [10] Bandyopadhyay S and Ghosal A. Geometric characterization and parametric representation of the singularity manifold of a 6–6 Stewart platform manipulator. *Mechanism and Machine Theory*, 41(11):1377–1400, 2006.
- [11] Bandyopadhyay S and Ghosal A. An algebraic formulation of kinematic isotropy and design of isotropic 6-6 Stewart platform manipulators. *Mechanism and Machine Theory*, 43(5):591–616, 2008.
- [12] Dasgupta B and Mruthyunjaya T S. Closed-form dynamic equations of the general Stewart platform through the Newton–Euler approach. *Mechanism and Machine theory*, 33(7):993–1012, 1998.
- [13] Hatip O E and Ozgoren M K. Utilization of a Stewart platform mechanism as a stabilizer. In *9th World Congress on the Theory of Machines and Mechanisms. Milan*, volume 1, pages 1393–1396, 1995.
- [14] Narasimhan S. *Dexterous robotic hands: Kinematics and control*. Doctoral Dissertation, Massachusetts Institute of Technology, USA, 1988.
- [15] Wang H, Xue C and Gruver W A. Neural network control of a parallel robot. In *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on*, volume 3, pages 2934–2938. IEEE, 1995.
- [16] Dasgupta B and Mruthyunjaya T S. The Stewart platform manipulator: a review. *Mechanism and Machine Theory*, 35(1):15–40, 2000.
- [17] Boudreau R and Gosselin C M. The synthesis of planar parallel manipulators with a genetic algorithm. *Trans. ASME, Journal of Mechanical Design*, 121(4):533–537, 1999.
- [18] Davidor Y. *Genetic Algorithms and Robotics: A Heuristic Strategy for Optimization*, Volume 1. World Scientific, 1991.
- [19] Grefenstette J. Evolutionary algorithms in robotics. In *International Symposium on Robotics and Manufacturing (ISRAM)*. Citeseer, 1994.
- [20] Masory O and Wang J. Workspace evaluation of Stewart platforms. *Advanced Robotics*, 9(4):443–461, 1994.
- [21] Tsai Y C and Soni A H. An algorithm for the workspace of a general n-R robot. *Journal of Mechanisms, Transmissions, and Automation in Design*, 105(1):52–57, 1983.
- [22] Pittens K H and Podhorodeski R P. A family of Stewart platforms with optimal dexterity. *Journal of Robotic systems*, 10(4):463–479, 1993.
- [23] Han C S, Tesar D and Traver A E. Optimum design of a 6 DOF fully-parallel micro-manipulator for enhanced robot accuracy. In *Published by American Society of Mechanical Engineers (ASME)*, 1989.

- [24] Gosselin C M and Guillot M. The synthesis of manipulators with prescribed workspace. *Journal of Mechanical Design*, 113(4):451–455, 1991.
- [25] Merlet J-P. Optimal design of robots. In *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.
- [26] Hao F and Merlet J-P. Multi-criteria optimal design of parallel manipulators based on interval analysis. *Mechanism and Machine theory* 40(2) (2005): 157-171.
- [27] Chablat D, Wenger P, Majou F and Merlet J-P. 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* 23(6) (2004): 615-624.
- [28] Merlet J-P. A formal-numerical approach for robust in-workspace singularity detection. *IEEE Transactions on Robotics*, 23(3) (2007): 393-402.
- [29] Merlet J-P. *Parallel Robots*, volume 74. Springer Science & Business Media, 2012.
- [30] Caro S, Chablat D, Goldsztejn A, Ishii D and Jermann C. A branch and prune algorithm for the computation of generalized aspects of parallel robots. *Artificial Intelligence*, 211 (2014): 34-50.
- [31] Lou Y, Liu G, Xu J and Li Z. A general approach for optimal kinematic design of parallel manipulators. In *Proceedings of IEEE Conference on Robotics and Automation*, Vol. 4, pp. 3659-3664. IEEE, 2004.
- [32] Lou Y, Liu G, Chen N and Li Z. Optimal design of parallel manipulators for maximum effective regular workspace. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 795–800. IEEE, 2005.
- [33] Lou Y, Liu G and Li Z. Randomized optimal design of parallel manipulators. *IEEE Transactions on Automation Science and Engineering*, 5(2):223–233, 2008.
- [34] Stamper R E, Tsai L W and Walsh G C. Optimization of a three DOF translational platform for well-conditioned workspace. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 4, pages 3250–3255. IEEE, 1997.
- [35] Tsai L W and Joshi S. Kinematics and optimization of a spatial 3-UPU parallel manipulator. *Journal of Mechanical Design*, 122(4):439–446, 2000.
- [36] Dunn W L and Shultis J K. *Exploring Monte Carlo Methods*. Elsevier, 2011.
- [37] Fishman G. *Monte Carlo: Concepts, Algorithms, and Applications*. Springer Science & Business Media, 2013.
- [38] Hammersley J M and Handscomb D C. Percolation processes. In *Monte Carlo Methods*, pages 134–141. Springer, 1964.

- [39] Kleijnen J P C, Ridder Ad A N and Rubinstein R Y. *Variance reduction techniques in Monte Carlo methods*. Springer, 2013.
- [40] Matlab R2015a, 64bit version Mathworks Inc., Natick Massachussets, 2015.
- [41] Edelsbrunner, H, Tan, T S and Waupotitsch, R. An $\mathcal{O}(n^2 \log n)$ Time Algorithm for the Minmax Angle Triangulation. *SIAM Journal on Scientific and Statistical Computing*, 13(4), pp.994-1008. 1992.
- [42] Allgower E L and Schmidt P H. An algorithm for piecewise-linear approximation of an implicitly defined manifold. *SIAM Journal on Numerical Analysis*, 22(2):322–346, 1985.
- [43] Allgower E L and Schmidt P H. Computing volumes of polyhedra. *Mathematics of Computation*, 46(173):171–174, 1986.
- [44] Cervantes-Sanchez J J, Hernandez-Rodriguez J C and Rendon-Sanchez J Gabriel. On the workspace, assembly configurations and singularity curves of the RRRRR-type planar manipulator. *Mechanism and Machine Theory*, 35(8):1117–1139, 2000.
- [45] Macho E, Altuzarra O, Pinto C and Hernandez A. Workspaces associated to assembly modes of the 5R planar parallel manipulator. *Robotica*, 26(03):395–403, 2008.
- [46] Rump S M. *INTLABinterval laboratory*. Developments in reliable computing. Springer Netherlands, 1999. 77-104.
- [47] Ghosal A. *Robotics: Fundamental Concepts and Analysis*. Oxford University Press, 2006.
- [48] Ghosal A and Ravani B. A differential-geometric analysis of singularities of point trajectories of serial and parallel manipulators. *Trans. ASME Journal of Mechanical Design*, 123(1), pp.80-89. 2001
- [49] Liu X-J, Wang J and Zheng H-J. Optimum design of the 5R symmetrical parallel manipulator with a surrounded and good-condition workspace. *Robotics and Autonomous Systems*, 54(3):221–233, 2006.
- [50] Merlet J-P. Determination of 6D workspaces of Gough type parallel manipulator and comparison between different geometries. *The International Journal of Robotics Research*, 18(9):902–916, 1999.
- [51] Fichter E F. A Stewart platform-based manipulator: general theory and practical construction. *The International Journal of Robotics Research*, 5(2):157–182, 1986.
- [52] Chatterjee D and Ghosal A. Design of a semi-regular Stewart platform manipulator for a desired workspace. *Proc. of NaCoMM 2009*, NIT Durgapur, 2009.
- [53] Arora J S. *Introduction to Optimum Design, Chapter 4, pp 154-157.*. Academic Press, 2004.

- [54] Huang M-Z. Performance analysis and design optimization of 5R planar parallel robots. In *ASME 2012 International Mechanical Engineering Congress and Exposition*, pages 351–356. American Society of Mechanical Engineers, 2012.
- [55] Boyd S and Vandenberghe L. *Convex Optimization*. Cambridge University Press, 2004.