Robotica: (2020) volume 38, pp. 861–885. © Cambridge University Press 2019
doi:10.1017/S0263574719001139

Computation of the safe working zones of Planar and Spatial Parallel Manipulators

Murali K. Karnam†, Aravind Baskar‡, Rangaprasad A. Srivatsan¶ and Sandipan Bandyopadhyay§*

† *Department of Biomedical Engineering, University of Basel, 4001 Basel, Switzerland*
‡ *Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, IN 46556, USA*
¶ *Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213, USA*
§ *Department of Engineering Design, Indian Institute of Technology Madras, Chennai 600 036, India*

(Accepted June 24, 2019. First published online: July 25, 2019)

SUMMARY
This paper presents the computation of the safe working zone (SWZ) of a parallel manipulator having three degrees of freedom. The SWZ is defined as a continuous subset of the workspace, wherein the manipulator does not suffer any singularity, and is also free from the issues of link interference and physical limits on its joints. The proposed theory is illustrated via application to two parallel manipulators: a planar 3-RRR manipulator and a spatial manipulator, namely, MaPaMan-I. It is also shown how the analyses can be applied to any parallel manipulator having three degrees of freedom, planar or spatial.

KEYWORDS: Safe working zone; Parallel manipulator; Singularities; Link interference; Joint limits; Singularity-free workspace.

Nomenclature

SWZ	Safe working zone
DoF	Degree(s)-of-freedom
MaPaMan	Madras parallel manipulator
SRSPM	Semi-regular Stewart parallel manipulator
IFT	Implicit function theorem
$\boldsymbol{o}$	A given point of interest inside the workspace of a parallel manipulator
$\mathcal{W}(\boldsymbol{o})$	SWZ containing the point $\boldsymbol{o}$
S_1	Conditional expression for loss-type singularity
$\mathcal{W}_1(\boldsymbol{o})$	Subset of the workspace, containing the point $\boldsymbol{o}$, which is free of loss-type singularity
S_2	Conditional expression for gain-type singularity
$\mathcal{W}_2(\boldsymbol{o})$	Subset of the workspace, containing the point $\boldsymbol{o}$, which is free of gain-type singularity
S_3	Conditional expression for link interference
$\mathcal{W}_3(\boldsymbol{o})$	Subset of the workspace, containing the point $\boldsymbol{o}$, which is free of link interference

Murali K. Karnam contributed to the paper while working as a Project Officer at the Indian Institute of Technology Madras, India. Aravind Baskar contributed to the paper as an MS scholar at the Indian Institute of Technology Madras, India. Rangaprasad A. Srivatsan contributed to the paper while working as a Project Officer at the Indian Institute of Technology Madras, India.

* Corresponding author. E-mail: sandipan@iitm.ac.in

S_4	Conditional expression for joint limit violations
$\mathcal{W}_4(\boldsymbol{o})$	Subset of the workspace, containing the point $\boldsymbol{o}$, which is free of joint-limit violations
$\boldsymbol{\theta}$	Active/actuated joint variables of a parallel manipulator
$\boldsymbol{\phi}$	Passive/un-actuated joint variables of a parallel manipulator
$\boldsymbol{x}$	Variables expressing the pose of the end-effector of a parallel manipulator
$\mathcal{W}_c(\boldsymbol{o})$	Convex subset of the SWZ, $\mathcal{W}(\boldsymbol{o})$
$\mathcal{W}_{c_i}(\boldsymbol{o})$	Largest convex subset of $\mathcal{W}_i(\boldsymbol{o})$, $i = 1, \ldots, 4$
$\boldsymbol{\chi}_{ij}$	Class of three-DoF manipulators with i rotational and j translational DoF
$\boldsymbol{g}$	Loop-closure equations, expressed in the variables, $\boldsymbol{\theta}$, $\boldsymbol{\phi}$ and $\boldsymbol{x}$
$\boldsymbol{h}$	Loop-closure equations, expressed in the variables, $\boldsymbol{\theta}$ and $\boldsymbol{x}$
$\boldsymbol{\eta}$	Loop-closure equations, expressed in the variables, $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$

1. Introduction

This paper proposes a strategy to identify a continuous subset of the workspace of a parallel manipulator in such a manner that when the pose of the manipulator belongs to this space, it is free of not just singularities, but also of link interference, as well as the physical limits on the motion of its joints. Such a region has been termed as the "safe working zone (SWZ)" in ref. [1]. In this paper, the general mathematical formulation of the SWZ is presented, followed by its application to two different parallel manipulators.

A parallel manipulator can be thought of as a number of serial ones, coupled at the *end-effector*. Thus, the workspace for the end-effector lies in the *intersection* of the individual workspaces that the end-effector would have had, considering only one serial chain at a time. Naturally, determination of the workspace of a parallel manipulator is harder than the same task in the case of serial manipulators. In addition, the said workspace typically contains certain singularities, known as the *gain-type* singularities[1] (see refs. [4, 5] and [6], pp. 159–166), which divide the workspace into smaller parts, thus limiting its useful expanse. There are four distinct factors which delimit the useful portion of the workspace of a parallel manipulator, and make the task of finding a "safe" path challenging, namely:

1. loss-type singularity[2],
2. gain-type singularity,
3. interference between the links, and
4. physical limits on the joint motions.

These factors find a mention in ref. [7], p. 213, albeit using a different terminology.

Paths avoiding singularities have been identified by researchers by careful planning. For example, a path connecting two given points in the workspace is numerically identified in ref. [8] by avoiding singularities. The same is found in ref. [9] by modelling the singularity surfaces as obstacles. In ref. [10], singularity-free paths are found using a variational approach and an artificial potential field. Another strategy is to bypass the singularities by raising the number of actuators beyond the dimension of the workspace of the manipulator (i.e., introducing an *actuator redundancy*). This scheme allows controlled navigation of a path containing a singularity (in the original workspace)(ref. [11]). The same result can be achieved in a slightly different manner, in which the required redundancy is artificially created by confining the workspace of the manipulator to a lower dimension, rather than using an additional actuator.[12] While many of the above strategies have been demonstrated successfully through experiments or simulations, they have a common inherent issue – the solutions obtained are specific to the given desired path, and hence need to be computed afresh every time the desired path changes.

[1]These type of singularities have also been termed as the *singularities of the second kind* (ref. [2]), or *direct* (ref. [3]) or *parallel singularities* (ref. [3]).

[2]These are the "serial type" singularities in the individual limbs of the parallel manipulator (see, e.g., ref. [6], pp. 158–159).

https://doi.org/10.1017/S0263574719001139 Published online by Cambridge University Press

Another approach to this problem is to identify *a priori* subsets of the workspace which are free of one or more of the issues listed above, ensuring that any path contained within these regions is safe with respect to these issues. For instance, "singularity-free zones" have been defined previously and have also been computed in the case of a few manipulators. In ref. [13], the maximal singularity-free zone centred about a desired point for the 3-RPR manipulator is identified, using a constrained optimisation approach. This is extended to the Stewart–Gough platform in ref. [14]. In both of these cases, closed-form expressions for the gain-type singularity manifold of the manipulators have been used. Unfortunately, such expressions are not yet available for many parallel manipulators. In such cases, numerical techniques are resorted to. The maximal singularity-free zone of the 3-RPR manipulator has also been obtained as a cylinder using the particle swarm optimisation technique in ref. [15]. Here, the centre for such a cylinder is varied to maximise the size of the singularity-free zone of the manipulator. Maximal singularity-free boxes are found for the 3-RPS manipulator in the joint space in ref. [16]. The design space is optimised to obtain a singularity-free zone for parallel manipulators using constraints derived from geometry in ref. [17]. The reachable workspace of the Stewart–Gough platform is found by taking into account the singularities and link interference in ref. [18]. The workspace analysis of the 3-RUU manipulator in its translational operation mode is carried out in ref. [19], wherein only its singularities are considered. Similarly, the maximal singularity-free workspace is identified for the 3-PRR planar manipulator in ref. [20] using a geometric approach, operating on sections of the workspace corresponding to the fixed orientations of the end-effector. Recently, a method to compute the singularity-free spheres in the constant orientation workspace of the SRSPM has been presented in ref. [21]. In another recent work, the constant orientation workspace of a cable-driven parallel manipulator is identified, which includes conditions for wrench-feasibility.[22] Monte Carlo simulation-based approach has been adopted to identify the workspace of a parallel robot in an optimal manner, which ensures the well-conditioning of certain Jacobian matrices inside the said workspace while conforming to some constraints specifying the ranges of motions at various joints.[23]

Most of these methods take care of only one or two of the four possible issues in identifying the safe workspace. In contrast, in this paper, the SWZ is computed by exhaustively addressing *all* the issues. The mathematical functions which represent these issues are formulated, and a numerical scheme is used to evaluate them, which computes the SWZ, subject to certain prescribed resolutions. The numerical scheme is simple to implement and computationally efficient when compared to the existing methods reported in literature, making it feasible to identify the SWZ at resolutions of practical significance swiftly (e.g., within a real time[3] of 1.125 s for the 3-RRR manipulator[4]). Such an efficiency of computation enables the development of design algorithms for parallel manipulators, in which the dimensions of the desired SWZ are used as constraints. An example of such a design process that makes use of the concept and computational method of the SWZ presented in this paper can be found in ref. [24]. Moreover, finding the SWZ is a one-time process for a manipulator of given dimensions, and the formulation is valid for *any* parallel manipulator. For example, an application to the spatial 3-RRS manipulator may be found in ref. [25]. These observations summarise the contributions of the paper.

The rest of the paper is organised as follows. The concept of SWZ is formally established in Section 2. The procedure for evaluating it is elaborated in Section 3. In the later two sections, this formulation is applied to two parallel manipulators – a planar three-DoF parallel manipulator, namely, the 3-RRR manipulator, and, a spatial three-DoF parallel manipulator, namely, MaPaMan-I. The conclusions of the paper are presented in Section 6.

2. Definition and Structure of the SWZ

Terms such as "singularity-free",[14] "practical",[26] "desired",[27] "specific",[27] "complete",[28] and "operational"[29] have been used in literature to define a region inside the workspace of a parallel manipulator,

[3]All the computations in this work have been performed on a computer with an Intel® Core™ i7-3770 CPU @ 3.40GHz processor and 32GB of RAM, using a `C++` code parallelised with `OpenMP`, employing eight threads.

[4]Computed with a linear resolution of 0.0025 normalised length and an angular resolution of 0.5°. All the linear dimensions are normalised with respect to the respective base dimension of the manipulator, to accommodate for the variations in the existing methods in literature.

https://doi.org/10.1017/S0263574719001139 Published online by Cambridge University Press

which is free of either singularities, or physical constraints, or both. It is interesting to note, however, that in spite of the significant variety that exists in the related terminology, none of the existing terms relate to *all* of the four issues described in Section 1. This observation motivates the necessity of defining a new term, namely, SWZ (denoted by $\mathcal{W}(\boldsymbol{o})$), which fills the gap adequately.

Definition 1: The SWZ, $\mathcal{W}(\boldsymbol{o})$, is defined as a subset of the workspace of the manipulator satisfying the following requirements:

1. $\mathcal{W}(\boldsymbol{o})$ is a *connected* set, containing a given point of interest (denoted by $\boldsymbol{o}$).
2. $\mathcal{W}(\boldsymbol{o})$ is contained entirely *inside* the workspace, that is, $\mathcal{W}(\boldsymbol{o})$ is free of loss-type singularities.
3. $\mathcal{W}(\boldsymbol{o})$ does not intersect the gain-type singularity manifold, that is, $\mathcal{W}(\boldsymbol{o})$ is free of gain-type singularities as well.
4. There is no interference between the links at any point of $\mathcal{W}(\boldsymbol{o})$. This requires the consideration of the physical shape and dimensions of the links.
5. At no point of $\mathcal{W}(\boldsymbol{o})$ does any joint violate a physical limit on its range of motion. Once again, the actual physical dimensions are to be considered.

It may be noted further that the SWZ, as defined above, depends on the choice of a point of interest, namely, $\boldsymbol{o}$. Indeed, one can argue that a parallel manipulator could have *multiple* SWZ, for various choices of $\boldsymbol{o}$, and thus the inclusion of $\boldsymbol{o}$ in the definition appears to render it ambiguous. In practice, however, there is hardly any difficulty in choosing the point $\boldsymbol{o}$ *uniquely*, which remedies this issue as a consequence. Most practical parallel manipulators have identical architecture of their limbs, which, in turn, defines an *axis of symmetry* in each case. The point $\boldsymbol{o}$ can be located conveniently on this axis, depending on the functional requirements of the manipulator. Even if the practical applications do not motivate a particular choice of $\boldsymbol{o}$ on this axis of symmetry, it is possible to find $\boldsymbol{o}$ in a manner that optimises the extent of the SWZ (which is not in the scope of this paper). The examples of the manipulators appearing later in the paper support the above observations.

In the definition of the SWZ, the requirements 2–5 have a common characteristic: each defines a subset of the workspace, which is *enclosed* by the *zero-level set* of a corresponding boundary function (denoted henceforth by S_i). The following define the terminology associated with these boundary functions.

- The zero-level set of the condition for loss-type singularity, defined by $S_1 = 0$, bounds the workspace, as depicted schematically in Fig. 1(a). The region of interest is the subset of this workspace which contains $\boldsymbol{o}$ (denoted by $\mathcal{W}_1(\boldsymbol{o})$).
- The gain-type singularity manifold is obtained by evaluating the condition for gain-type singularity, which is given by the solution set of $S_2 = 0$. The region $\mathcal{W}_2(\boldsymbol{o})$, which contains $\boldsymbol{o}$, is free of gain-type singularities and is bounded by the set of points defining the gain-type singularity manifold, as seen in Fig. 1(b).
- The region that includes $\boldsymbol{o}$ is free of link interference as well, is denoted by $\mathcal{W}_3(\boldsymbol{o})$, and is bounded by the set satisfying $S_3 = 0$, as seen in Fig. 1(c).
- The solution set of $S_4 = 0$ delimits $\mathcal{W}_4(\boldsymbol{o})$, that is, the space that contains $\boldsymbol{o}$, and is free of joint-limit violations (see Fig. 1(d)).

As seen in Fig. 1(e), the SWZ of the manipulator can be computed as:

$$\mathcal{W}(\boldsymbol{o}) = \bigcap_{i=1}^{4} \mathcal{W}_i(\boldsymbol{o}). \tag{1}$$

Note that not all the manipulators have all of the four requirements. For example, as explained in Section 4, the 3-RRR manipulator has no physical limits on the joint motions.

3. Finding the SWZ for a Parallel Manipulator

The computation of the SWZ involves multiple steps – the various boundary functions need to be formulated, and subsequently the zero-level sets of those boundary functions are to be evaluated.

https://doi.org/10.1017/S0263574719001139 Published online by Cambridge University Press

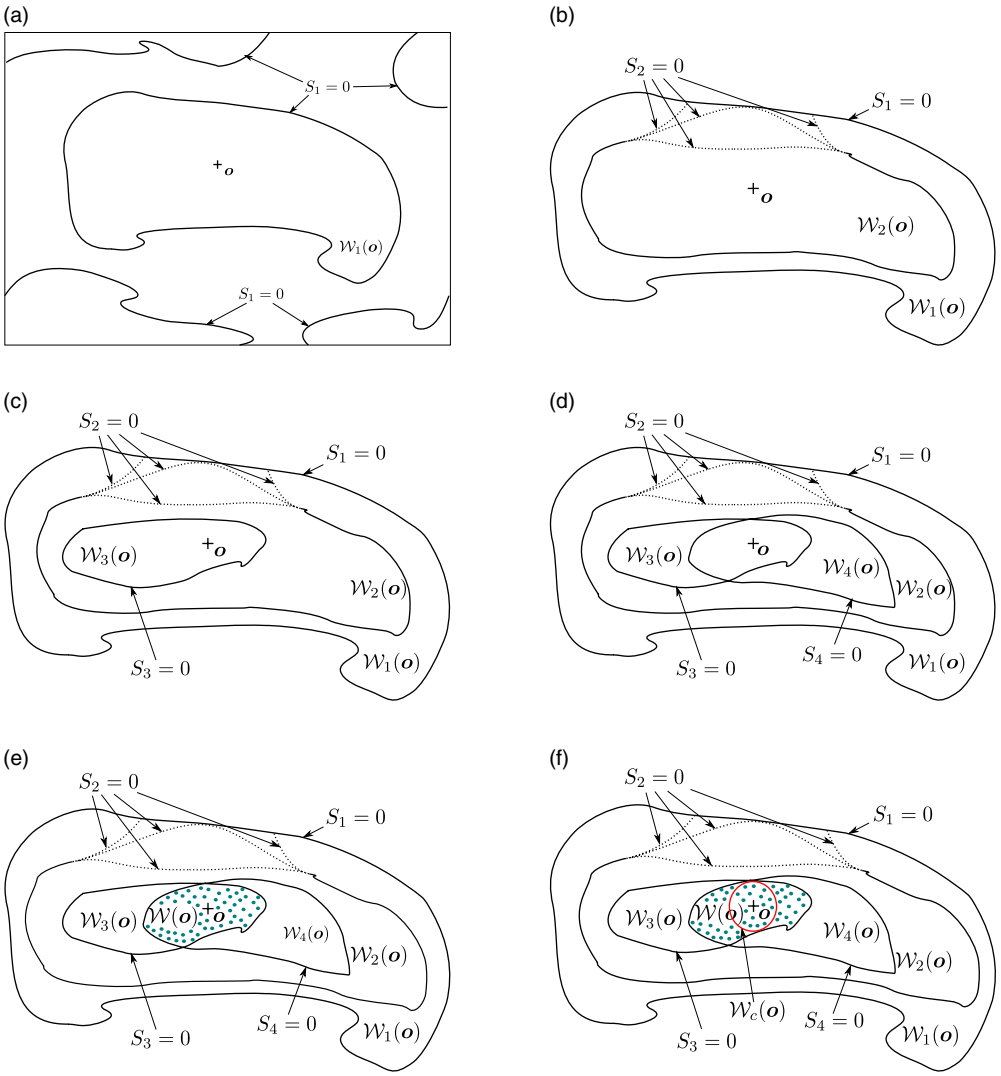

Fig. 1. Definition of $\mathcal{W}(\boldsymbol{o})$ and $\mathcal{W}_c(\boldsymbol{o})$. The dotted lines indicate some of the various possible relations between the bounding curves.

In this section, a scheme for formulating the various boundary functions, along with the analytical and the numerical computational strategies to find the SWZ, is presented.

3.1. Formulation of the SWZ

The SWZ for any parallel manipulator is bounded by the zero-level sets of four boundary functions (S_i, $i = 1, \ldots, 4$). The conditions associated with the link interferences ($S_3 = 0$), and joint limits ($S_4 = 0$), are architecture dependent, which have to take into consideration the actual geometry of the links of a manipulator, and hence are discussed on a case-by-case basis. This section elaborates the strategy used for finding the boundary functions $S_1 = 0$ and $S_2 = 0$ in the generic case and the derivation of the singularity manifolds in terms of the variables describing the pose (i.e., position and/or orientation) of the end-effector.

3.1.1. Condition for the loss-type singularity ($S_1 = 0$). The loop-closure equations for a parallel manipulator can always be written in the form:

$$\boldsymbol{g}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{x}) = \boldsymbol{0}; \quad \boldsymbol{g} \in \mathbb{R}^{m+n}; \quad \boldsymbol{\theta}, \boldsymbol{x} \in \mathbb{R}^n; \quad \text{and,} \quad \boldsymbol{\phi} \in \mathbb{R}^m, \tag{2}$$

https://doi.org/10.1017/S0263574719001139 Published online by Cambridge University Press

where $\boldsymbol{\theta}$ represents the *active/actuated* joint variables, $\boldsymbol{\phi}$ represents the *passive/un-actuated* joint variables[5], and $\boldsymbol{x}$ represents the *end-effector pose* variables. From Eq. (2), the passive variables, $\boldsymbol{\phi}$, can be eliminated to obtain an equivalent set of constraints as:

$$\boldsymbol{h}(\boldsymbol{\theta}, \boldsymbol{x}) = \mathbf{0}; \quad \boldsymbol{h}, \boldsymbol{\theta}, \boldsymbol{x} \in \mathbb{R}^n. \tag{3}$$

In general, for an n-DoF manipulator, it is possible to obtain a set of n such scalar equations, which define its *inverse kinematics*. At a generic or *regular* point, it is possible to find all the distinct solutions of Eq. (3) in the form $\boldsymbol{\theta} = \boldsymbol{\theta}(\boldsymbol{x})$. These solutions are known to coincide, in a pairwise manner, at the boundary of the workspace of the manipulator, where the loss-type singularity is said to occur (see, e.g., ref. [6], pp. 158–159). Using the implicit function theorem (IFT), the condition for loss-type singularity can be obtained as $S_1 = 0$, where:

$$S_1 = \det\left(\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\theta}}\right). \tag{4}$$

Following the statement of the IFT, if $S_1 = 0$, the inverse kinematics problem has non-unique solutions – that is, the branches of inverse kinematics meet, signifying the loss-type singularity.

3.1.2. Condition for gain-type singularity ($S_2 = 0$). The end-effector pose variables, $\boldsymbol{x}$, can be eliminated from Eq. (2), instead of $\boldsymbol{\phi}$, to obtain another equivalent set of constraints relating the active, and the passive joint variables as:

$$\boldsymbol{\eta}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbf{0}; \quad \boldsymbol{\eta}, \boldsymbol{\phi} \in \mathbb{R}^m; \text{ and, } \boldsymbol{\theta} \in \mathbb{R}^n. \tag{5}$$

Equation (5) can be solved *uniquely* to obtain $\boldsymbol{\phi} = \boldsymbol{\phi}(\boldsymbol{\theta})$ at all points which are free of gain-type singularities. At the boundaries of the *configuration space*[6], these solutions merge. Invoking the IFT again, the condition for the gain-type singularity can be obtained as $S_2 = 0$,[6] where:

$$S_2 = \det\left(\frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\phi}}\right). \tag{6}$$

It has to be noted that the gain-type singularity manifold obtained by using the condition in Eq. (6) would include the gain-type singularities associated with all the *working modes*[7]. Nevertheless, in the numerical scheme utilised for finding the said manifold, it is ensured that only the desired subset of the manifold, corresponding to a particular working mode (chosen *a priori*), is obtained by using the inverse kinematic solutions corresponding to only the desired branch in the computation.

3.1.3. Computing the singularity manifolds for the loss-type and the gain-type singularities. The SWZ of a manipulator is defined as a subset of its workspace. To compute it analytically, it is vital to obtain the singularity manifolds corresponding to both $S_1 = 0$ and $S_2 = 0$ in the end-effector pose variables alone. The conditions obtained in Eq. (4) and Eq. (6) are not expressed solely in terms of the end-effector pose variables, since they involve the joint-space variables as well. The equation set defined by Eq. (2), along with $S_1 = 0$, contains $(m + n + 1)$ equations in the joint-space variables, since $\boldsymbol{\phi} \in \mathbb{R}^m$, $\boldsymbol{\theta} \in \mathbb{R}^n$. From these, the $(m + n)$ joint variables can be systematically eliminated to obtain the loss-type singularity manifold in the end-effector pose variables alone. Similarly, the equation set defined by Eq. (2), along with $S_2 = 0$, can be used to find the gain-type singularity manifold. The elimination procedure for obtaining the manifold is explained in a case-by-case manner for each manipulator (see Section 4.3, and Section 5.3 for the details).

[5]Strictly speaking, the rotary joint motions can only be approximated *locally* as elements of $\mathbb{R}^p$ for p such joints about a regular configuration/pose.

[6]The configuration space is defined as the space of variables defining the configuration of the manipulator. It can have a mix of active, passive, and output variables. In the present context, it is the space of joint variables, as defined in ref. [5].

[7]Working modes are the different configurations of the manipulator associated with the solutions of the inverse kinematics problem.[28]

https://doi.org/10.1017/S0263574719001139 Published online by Cambridge University Press

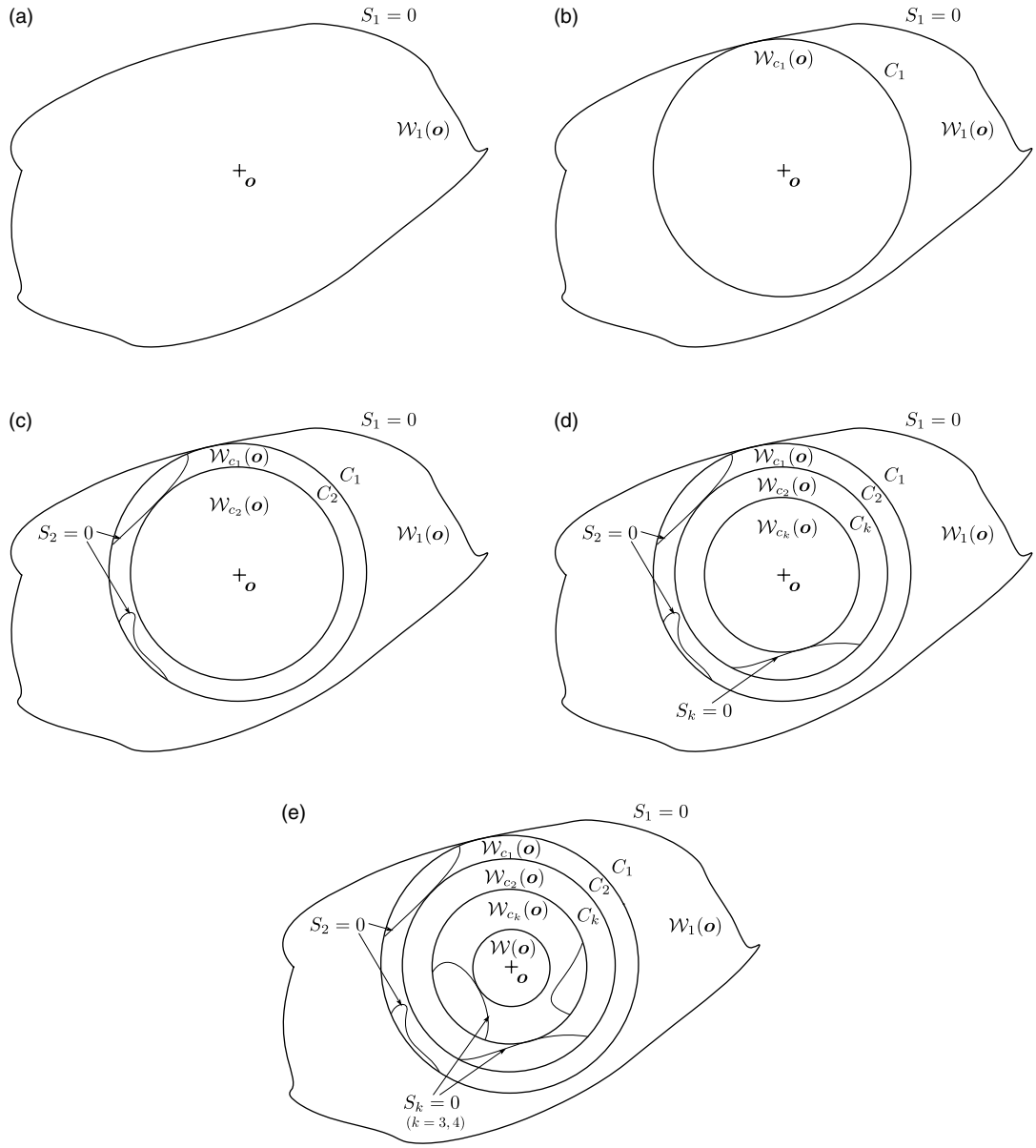

Fig. 2. Strategy for computing $\mathcal{W}_c(\boldsymbol{o})$.

3.2. Computation of the SWZ

The SWZ of a manipulator is not necessarily convex, as it depends on the zero-level sets of various boundary functions. It is generally preferred to identify a *convex subset*, $\mathcal{W}_c(\boldsymbol{o}) \subseteq \mathcal{W}(\boldsymbol{o})$, which makes *point-to-point* path-planning with piece-wise continuous linear segments trivial inside $\mathcal{W}_c(\boldsymbol{o})$. This is very important for industrial applications, where linear segments are the most common elements used for robot path-planning. From a physical standpoint, the different boundary functions follow a hierarchy as shown in Fig. 2, imposing the following structure on $\mathcal{W}_i(\boldsymbol{o})$:

$$\mathcal{W}_c(\boldsymbol{o}) \subseteq \mathcal{W}(\boldsymbol{o}) = \left(\mathcal{W}_3(\boldsymbol{o}) \bigcap \mathcal{W}_4(\boldsymbol{o})\right) \subseteq \mathcal{W}_2(\boldsymbol{o}) \subseteq \mathcal{W}_1(\boldsymbol{o}). \tag{7}$$

The above observations motivate a scheme for the computation of the final result, namely, $\mathcal{W}_c(\boldsymbol{o})$, following the steps given below:

1. Compute $\mathcal{W}_1(\boldsymbol{o})$. Find the *largest* convex subset $\mathcal{W}_{c_1}(\boldsymbol{o})$ in it, containing $\boldsymbol{o}$. The subset can be in the form of convex polyhedra, super-ellipsoids, ellipsoids, so on. Without any loss of generality,

https://doi.org/10.1017/S0263574719001139 Published online by Cambridge University Press

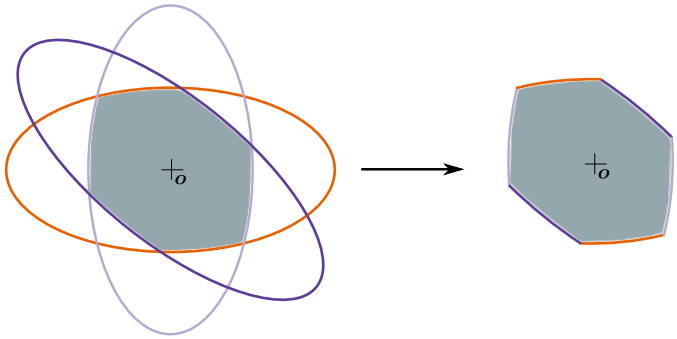

Fig. 3. A convex region at the intersection of three concentric ellipses of the same size.

and for the ease of computation, the *circle* has been used as the chosen shape for $\mathcal{W}_{c_1}(\boldsymbol{o})$ in the 2-D sections of the workspace in this work. As shown in Fig. 2, the circle C_1 bounds the region $\mathcal{W}_{c_1}(\boldsymbol{o})$.

2. In a similar manner, find C_2, which bounds the region $\mathcal{W}_{c_2}(\boldsymbol{o})$. Obviously, $\mathcal{W}_{c_2}(\boldsymbol{o}) \subseteq \mathcal{W}_{c_1}(\boldsymbol{o})$.
3. Compute the corresponding entities, namely, $\mathcal{W}_{c_3}(\boldsymbol{o})$, C_3, $\mathcal{W}_{c_4}(\boldsymbol{o})$, and C_4 accordingly. Finally, find:

$$\mathcal{W}_c(\boldsymbol{o}) \subseteq \left(\mathcal{W}_{c_3}(\boldsymbol{o}) \bigcap \mathcal{W}_{c_4}(\boldsymbol{o})\right) \subseteq \mathcal{W}_{c_2}(\boldsymbol{o}) \subseteq \mathcal{W}_{c_1}(\boldsymbol{o}) \subseteq \mathcal{W}_1(\boldsymbol{o}). \tag{8}$$

The choice of circle as the convex bounding curve in each 2-D-slice is not unique and can be modified to fit the specific application at hand. In this work, the circular geometry has been adopted for the 2-D sections, motivated by the practical considerations listed below.

- The circle has the highest order of symmetry among *all* the simple planar closed curves. Manipulators are rarely built for a specific task warranting an asymmetry in their SWZ. In the absence of an express need for such a bias in certain directions, the highest order symmetry of the SWZ is the optimal choice due to the uniform motion capability that it ensures in *all* possible directions.
- A circle encloses a *convex region*, namely, a disc. The intersection of any number of concentric discs is still a convex region, that is, the smallest of the discs, which can be identified *trivially*. While such convex regions of intersection can be computed easily in the case of other convex regions as well, the computational cost is higher in *all* of them. This point has been illustrated in Fig. 3, where the SWZ is the shaded region, computed as the intersection of three concentric ellipses. This area is bounded by *six* individual elliptical segments. Given this form of the SWZ, to determine if a given point is *inside* it, or *outside*, one needs to perform more computations than to simply compare two real numbers, namely, the distance of the said point from the centre of the SWZ, $\boldsymbol{o}$, and the radius of the SWZ had circular sections been used instead. In real-life design problems, as in ref. [24], the number of such sections used is of the order of tens, and hence the situation is lot more than demanding that what Fig. 3 represents.
- If, for instance, one were to choose elliptical sections instead of circular, the difference in the CPU time required for one section would be insignificant. However, one needs to keep in mind that the actual utility of such computations is to enable the design algorithms and any additional computation makes a significant impact in the total time needed in the design computations.
  For example, in ref. [24], the number of candidate designs for which the SWZ had to be computed was 120,000, and for each of these designs, the specified resolution required the computation of 20 2-D circular sections. At this level of resolution, the overall design required 4 h of computation on a Tesla K40 GPU, with 2880 CUDA cores. Unless there is a practical justification to add to the computational complexity at the level of formulation, one might wish to invest additional computational time/resources, if affordable, towards the improvement of the optimisation process in terms of scan-resolutions and number of candidate solutions considered.

https://doi.org/10.1017/S0263574719001139 Published online by Cambridge University Press

The implication of the hierarchy of the sets depicted in Eq. (8) is very significant in the actual implementation of the scheme. When a numerical scheme is used to find the SWZ (as explained in Section 3.3), a progressively diminishing domain for the search algorithm helps in reducing the computational effort (for a given desired level of resolution of the results obtained). Instead of computing the sets $\mathcal{W}_i(\boldsymbol{o})$, and subsequently their intersections to obtain $\mathcal{W}(\boldsymbol{o})$, and finally $\mathcal{W}_c(\boldsymbol{o})$, the *convex sets* $\mathcal{W}_{c_i}(\boldsymbol{o}) \subseteq \mathcal{W}_i(\boldsymbol{o})$ are computed from the beginning itself, and the search space is reduced in a progressive manner. For example, as seen in Fig. 2, the zero-level set of S_2 is computed only inside $\mathcal{W}_{c_1}(\boldsymbol{o})$, instead of inside $\mathcal{W}_1(\boldsymbol{o})$, to obtain $\mathcal{W}_{c_2}(\boldsymbol{o})$, and so on.

3.3. Numerical scheme for the computation of the zero-level sets of $S_i = 0$

Although a strategy for obtaining the singularity manifolds is described in Section 3.1.3, it is not always possible to compute them analytically. Even when such a computation is successful, the end-result is typically too large an expression to be amenable to further analysis. Moreover, the boundary functions corresponding to link interference (S_3) and joint limits (S_4) present in the SWZ formulation most often are not available analytically. Thus, a numerical scheme is adopted in this work to find the zero-level sets of all the boundary functions.

A number of numerical schemes have already been reported for the analysis and computation of the workspace boundaries of a manipulator. For example, interval analysis is used to obtain the $\mathcal{W}_1(\boldsymbol{o})$ in ref. [27]. A numerical technique to find $\mathcal{W}_1(\boldsymbol{o})$ and $\mathcal{W}_2(\boldsymbol{o})$ is presented in ref. [30]. Numerical schemes that assume the links to be line segments have been employed to find $\mathcal{W}_3(\boldsymbol{o})$ and $\mathcal{W}_4(\boldsymbol{o})$ in ref. [31]. Analytical means to compute the shortest distances between convex shapes have been presented in refs. [32, 33]. Alternatively, numerical schemes can also be employed to find link interference, as shown in ref. [34]. A numerical framework to find out each of $\mathcal{W}_1(\boldsymbol{o})$–$\mathcal{W}_4(\boldsymbol{o})$, individually, for a general hexagonal Stewart platform, has been presented in ref. [26]. In the numerical scheme presented in ref. [22], first a small sphere is defined with its centre at the point of interest, $\boldsymbol{o}$. Its radius is then increased gradually, until at least one of the constraints is violated. The surface of the sphere is tessellated in terms of triangles to facilitate the computation. In the present work, it is attempted to systematically define all the individual conditions determining the $\mathcal{W}(\boldsymbol{o})$ of a parallel manipulator and to obtain the region of interest in a hierarchical manner.

For the numerical evaluation of the zero-level sets of the boundary functions required in the computation of SWZ, a simple 2-D grid-search algorithm[8] is used, following refs. [9, 18]. A routine is developed in C++, which returns the zero-level set of a function in two variables. The search space is overlaid with a rectangular grid of *reasonably* high initial resolution, and the zero-crossings of the function are detected in terms of a change in the sign of the function over the grid boundaries. This routine is called recursively, only at the identified blocks of the grid, to improve the accuracy without compromising the computational efficiency. The scheme is illustrated via an application to the sample function[9], which is shown in Fig. 4(a). As this function defines a three-dimensional surface, the Z-direction is discretised to obtain a set of 2-D sections parallel to the XY-plane. At every such 2-D section, the zero-level set of the said function is evaluated. Such a section at $z = 0.5$ is shown in Fig. 4(b). To extend this approach to manipulators with more than 2-DoF, a suitable 2-D subspace (e.g., a circular region) of the workspace is selected, and the remaining directions are discretised to evaluate the zero-level sets. These regions are subsequently stacked up along the Z-axis to obtain the required zero-level set of each boundary function in the 3-D space. This strategy is applicable to *any* 3-DoF manipulator, since all 3-DoF manipulators can be classified into one of the four classes as defined in ref. [35]:

1. Class $\boldsymbol{\chi}_{30}$ (all the DoF are rotational): any one of the DoF can be chosen for the discretisation, with the 2-D search algorithm being used for the remaining two-DoF.
2. Class $\boldsymbol{\chi}_{21}$(one of the DoF is translational): the translational DoF is chosen for discretisation, with the 2-D search algorithm being used for the remaining two rotational DoF.

[8] An alternative to the proposed method of discretisation and using a 2-D search algorithm is to use a 3-D grid search algorithm, as proposed, for example, in ref. [30]. This constitutes one of the potential extensions of the work reported in this paper.

[9] The function $f(x, y, z) \triangleq z - \mathrm{sinc}(\sqrt{x^2 + y^2})$ is used as an example, where sinc is the *cardinal sine* function, defined as $\mathrm{sinc}\, x = \frac{\sin x}{x}$.

https://doi.org/10.1017/S0263574719001139 Published online by Cambridge University Press

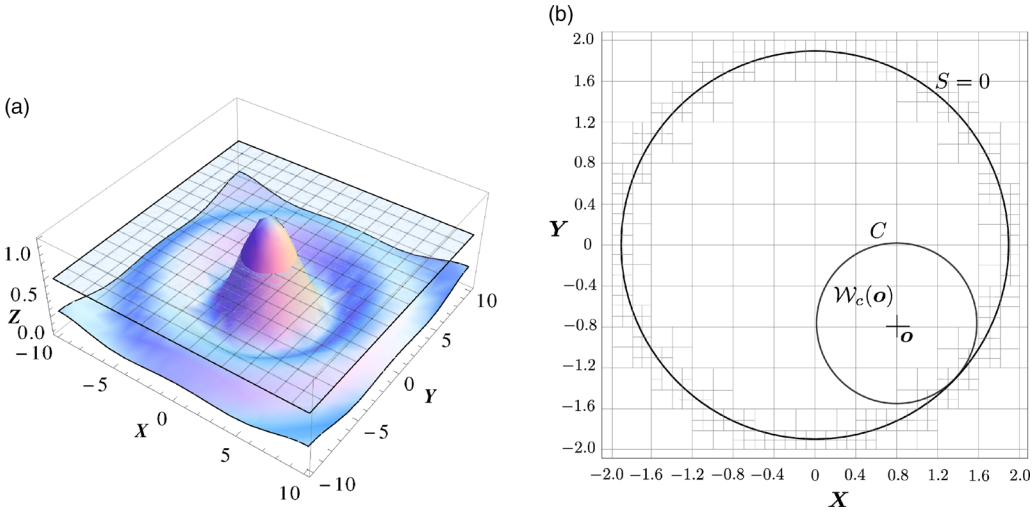

Fig. 4. Numerical scheme for computing the zero-level sets of the boundary functions, illustrated for a function, $f(x, y, z) = z - \mathrm{sinc}(\sqrt{x^2 + y^2})$.

3. Class $\boldsymbol{\chi}_{12}$ (one of the DoF is rotational): the rotational DoF is chosen for discretisation with the 2-D search algorithm being used for the remaining two translational DoF.
4. Class $\boldsymbol{\chi}_{03}$ (all the DoF are translational): any one of the DoF can be chosen for the discretisation, with the 2-D search algorithm being used for the remaining 2-DoF.

A manipulator with 4-, 5-, or 6-DoF may also be analysed in a similar manner. For instance, in the case of the Stewart platform manipulator, it is possible to identify a sphere in the constant orientation workspace that is free of singularities, as shown in ref. [21]. If the orientation workspace is scanned using the numerical scheme described in this paper, and the smallest sphere is found, then that would be the singularity-free sphere in the 6-DoF workspace of the manipulator. Similarly, ref. [36] presents a method for computing a sphere in the constant orientation workspace that is free of leg interference. It is possible to find the spheres corresponding to the other issues, such as the motion limits of the spherical joints as well as the workspace boundaries in a similar manner, and then the SWZ can be computed for the 6-DoF manipulator. This research is currently under progress, and is not in the scope of the current paper.

4. Computation of the SWZ for the 3-RRR Planar Parallel Manipulator

The formulation discussed in Section 3 is illustrated via its application to a 3-RRR planar manipulator in this section. For the physical dimensions and design of the prototype under consideration, there are no limits on the joint angles, but interferences between the links are possible. Thus, computation of $\mathcal{W}_4(\boldsymbol{o})$ (or equivalently, $\mathcal{W}_{c_4}(\boldsymbol{o})$) is not required, and consequently, $\mathcal{W}_c(\boldsymbol{o}) = \mathcal{W}_{c_3}(\boldsymbol{o})$.

4.1. Geometry of the 3-RRR manipulator

The 3-RRR manipulator is a planar parallel manipulator with 3-DoF. Its workspace can be parametrised in terms of the sway, surge, and yaw, which are represented by the variables x, y, and α, respectively. The coordinates of the vertices of the end-effector, $\boldsymbol{p}_i$, $i = 1, 2, 3$, are obtained using the end-effector pose variables, $\boldsymbol{x} = (x, y, \alpha)^\top$. The active joint angles are given by $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)^\top$, and the passive joint angles by $\boldsymbol{\phi} = (\phi_1, \phi_2, \phi_3)^\top$ (see Fig. 5(b)).

4.2. Condition for the loss-type singularity ($S_1 = 0$)

An approach similar to that followed in Section 3.1.1 is used to find the condition for loss-type singularity in the 3-RRR manipulator. The inverse kinematics equations in Eq. (3) take the shape $\boldsymbol{h}(\boldsymbol{\theta}, \boldsymbol{x}) \triangleq (h_1, h_2, h_3)^\top = \boldsymbol{0}$, where

$$h_i(\theta_i, \boldsymbol{x}) = \|\boldsymbol{a}_i - \boldsymbol{p}_i\|^2 - r^2, \;\; i = 1, 2, 3. \tag{9}$$

https://doi.org/10.1017/S0263574719001139 Published online by Cambridge University Press

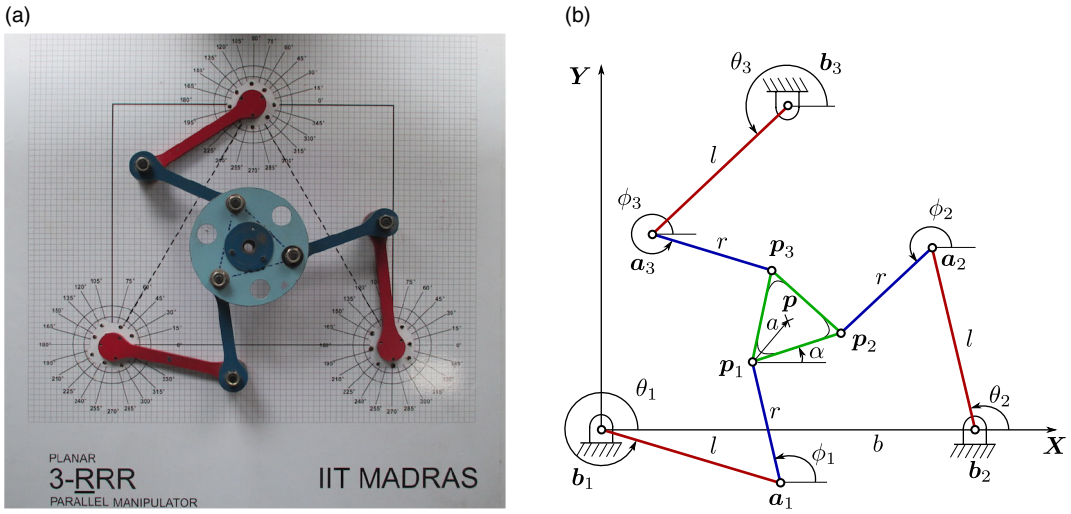

Fig. 5. Prototype and kinematic details of the 3-RRR manipulator.

Subsequently, the condition for the loss-type singularity, following the steps described in Section 3.1.1, is derived as $S_1 = \det\left(\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\theta}}\right) = 0$. For brevity, the expressions for $h_i(\theta_i, \boldsymbol{x})$ and S_1 are not included here.

4.3. Condition for the gain-type singularity ($S_2 = 0$)

Following the formulation outlined in Section 3.1.2, the loop-closure constraints of the 3-RRR manipulator are cast in the form $\boldsymbol{\eta}(\boldsymbol{\theta}, \boldsymbol{\phi}) = (\eta_1, \eta_2, \eta_3)^\top = \boldsymbol{0}$, where

$$\begin{aligned} \eta_1 &= \| \boldsymbol{p}_2 - \boldsymbol{p}_1 \| - 3a^2, \\ \eta_2 &= \| \boldsymbol{p}_3 - \boldsymbol{p}_2 \| - 3a^2, \\ \eta_3 &= \| \boldsymbol{p}_1 - \boldsymbol{p}_3 \| - 3a^2. \end{aligned} \tag{10}$$

From these, the condition for the gain-type singularity can be obtained (by invoking the IFT, as elaborated in Section 3.1.2) as $S_2 \triangleq \det\left(\frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\phi}}\right) = 0$.

Obtaining the singularity manifold as a closed-form expression in (x, y, α) alone has been attempted for the 3-RRR manipulator, which, if available, could aid in the analytical determination of the SWZ (see Appendix B). It is found to result in a huge final expression (≈ 40 GB in *size*[10]), which is not amenable to further mathematical analysis. Hence, numerical schemes are resorted to find SWZ for the 3-RRR manipulator, as explained below.

First, inverse kinematics is performed at each of the nodes of the grid and the configuration of the limbs is obtained. Since each limb can have two branches of solution to the inverse kinematics problem, there is a total of $2 \times 2 \times 2$ possible working modes for this manipulator (in the general case). From these, a symmetric mode is chosen, as the symmetric modes are more likely to maximise the SWZ. All the subsequent numerical analyses are therefore specific to this working mode. The analyses can be repeated for any other mode, without any additional effort or difference in approach.

4.4. Conditions for interferences among the passive links ($S_3 = 0$)

The prototype of the 3-RRR manipulator under study is designed such that all the links attached to the motors lie in the same plane, while the links attached to the end-effector all lie in another plane, above the former (see Fig. 6(a)). The dimensions of the active links are chosen such that the length of the base exceeds the sum of the length of any two active links, thus ensuring that there is no interference between the active links (see Fig. 6(b)). Link interference is, nevertheless, possible among the passive links. The links are assumed to be enclosed in bounding boxes in the form of

[10]The term size refers to the amount of internal memory consumed to describe the expression in the Computer Algebra System (CAS) `Mathematica`.[37]

https://doi.org/10.1017/S0263574719001139 Published online by Cambridge University Press

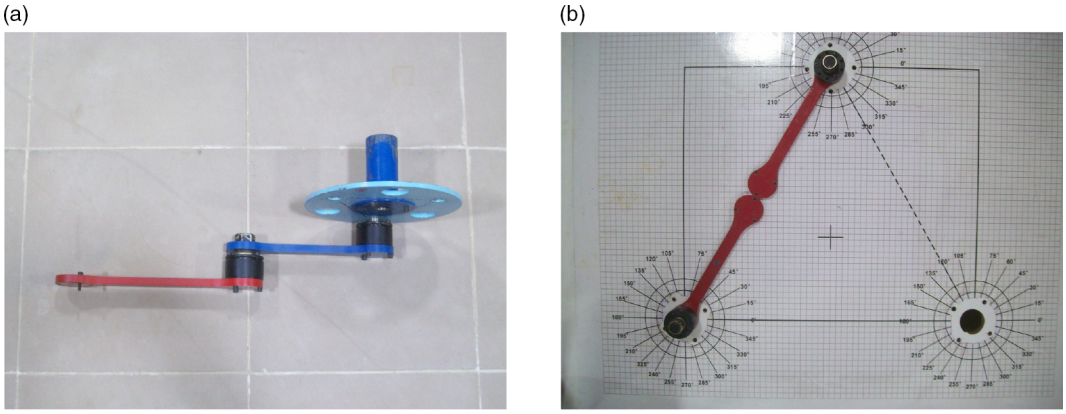

Fig. 6. Design features of links in the 3-RRR manipulator, aimed at avoiding any interference between them.

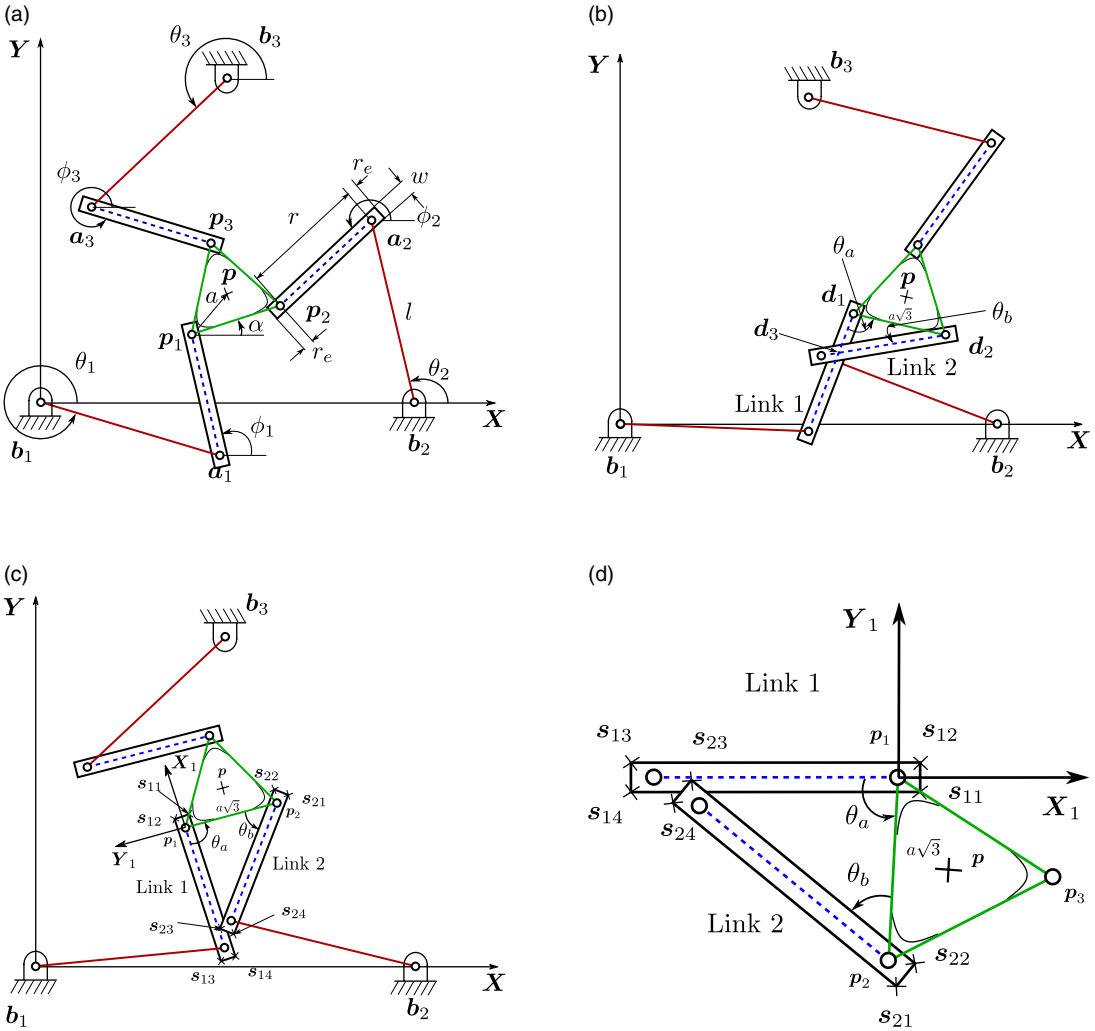

Fig. 7. Encapsulation of the passive links of the 3-RRR manipulator in rectangular bounding boxes, and identification of the interference between the bounding boxes.

rectangles of length $r + 2r_e$, and breadth w, as shown in Fig. 7(a). The link interference is checked in a pairwise fashion. Without any loss of generality, the interference is studied in detail between Links 1 and 2, and the results are generalised to the other two pairs. At any configuration, angles θ_a and θ_b can be obtained from the inverse kinematics. The links, encapsulated within rectangles in

https://doi.org/10.1017/S0263574719001139 Published online by Cambridge University Press

planes parallel to the $\boldsymbol{XY}$ plane, can interfere in two ways: one link completely overlaps the other (see Fig. 7(b)) or one of the links partially overlaps the other (see Fig. 7(c)). Although the first kind of interference cannot happen in reality as partial overlap precedes it, this case must be considered for the mathematical completeness of the formulation. The first kind of interference can be detected using the following steps:

- If the links overlap completely, the centre line of the links would intersect each other.
- As shown in Fig. 7(b), applying the sine law to the $\triangle \boldsymbol{d}_1\boldsymbol{d}_2\boldsymbol{d}_3$:

$$\frac{a\sqrt{3}}{\sin(\pi - (\theta_a + \theta_b))} = \frac{\|\overline{\boldsymbol{d}_1\boldsymbol{d}_3}\|}{\sin\theta_b} = \frac{\|\overline{\boldsymbol{d}_2\boldsymbol{d}_3}\|}{\sin\theta_a}. \tag{11}$$

- Upon finding the values of $\|\overline{\boldsymbol{d}_1\boldsymbol{d}_3}\|$ and $\|\overline{\boldsymbol{d}_2\boldsymbol{d}_3}\|$, the condition for the centre lines to intersect can be obtained as $\|\overline{\boldsymbol{d}_1\boldsymbol{d}_3}\| < r$, or $\|\overline{\boldsymbol{d}_2\boldsymbol{d}_3}\| < r$.
- Thus, if the centre lines intersect, then $S_3 = 0$.

If the first kind of interference is not detected, then it is required to check if there is any partial overlap of the rectangles (see Fig. 7(c)). This can be done following the steps given below.

- A reference frame is chosen with its origin at $\boldsymbol{p}_1$ and its $\boldsymbol{X}_1$ axis aligned to the Link 1, as shown in Fig. 7(c). The coordinates of the vertices of Link 2 are computed in the new reference frame:

$$^1\boldsymbol{s}_{21} = \boldsymbol{R}_Z(\pi + \theta_a)\left(\boldsymbol{c}_2 + \boldsymbol{R}_Z(\theta_b)(r_e, -w/2)^\top\right), \tag{12}$$
$$^1\boldsymbol{s}_{22} = \boldsymbol{R}_Z(\pi + \theta_a)\left(\boldsymbol{c}_2 + \boldsymbol{R}_Z(\theta_b)(r_e, w/2)^\top\right), \tag{13}$$
$$^1\boldsymbol{s}_{23} = \boldsymbol{R}_Z(\pi + \theta_a)\left(\boldsymbol{c}_2 + \boldsymbol{R}_Z(\theta_b)(-r - r_e, w/2)^\top\right), \tag{14}$$
$$^1\boldsymbol{s}_{24} = \boldsymbol{R}_Z(\pi + \theta_a)\left(\boldsymbol{c}_2 + \boldsymbol{R}_Z(\theta_b)(-r - r_e, -w/2)^\top\right), \quad \text{where} \tag{15}$$
$$\boldsymbol{c}_2 = \boldsymbol{d}_2 - \boldsymbol{d}_1 = \left(a\sqrt{3}, 0\right)^\top. \tag{16}$$

- Partial overlap happens if any vertex of the bounding box of Link 2, say, $^1\boldsymbol{s}_{21}$, lies inside the bounding box of Link 1, which is captured by the conditions:

$$-r_e \leq {}^1\boldsymbol{s}_{21} \cdot \boldsymbol{e}_X \leq r + r_e, \quad \text{and} \tag{17}$$
$$-\frac{w}{2} \leq {}^1\boldsymbol{s}_{21} \cdot \boldsymbol{e}_Y \leq w/2, \tag{18}$$

  where $\boldsymbol{e}_X$ and $\boldsymbol{e}_Y$ represent the unit vectors in the $\boldsymbol{X}_1$ and $\boldsymbol{Y}_1$ directions, respectively.
- If any vertex of the bounding box of Link 2 lies on/inside the bounding box of Link 1, then the links interfere and the condition for link interference $S_3 = 0$ is obtained. Similarly, it is checked if any vertex of the bounding box of Link 1 lies on/inside the bounding box of Link 2. The steps are repeated for every pair of passive links in the manipulator.

4.5. Numerical results

This section describes the results of the application of the above formulations to the 3-RRR manipulator prototype, whose dimensions are given in ref. [12], with the manipulator base side length, $b = 500$ mm, length of the active links, $l = 220$ mm, length of the passive links, $r = 170$ mm, and side of the end-effector triangle, $a = 125$ mm. The workspace of the 3-RRR manipulator is parametrised by (x, y, α). However, the boundary functions S_i are not available solely in terms of these variables. Therefore, instead of direct computation of the zero-level sets, these are found by applying a numerical scheme described in Section 3.3. The workspace is discretised into planes of constant values of α, at a resolution of 0.5°, and 2-D search is performed in each such plane for the zero-level sets of the functions, $S_i = 0,\ i = 1, 2, 3$. Figure 8 shows these zero-level sets in such a plane obtained for the fixed orientation, $\alpha = -10°$.

As noted earlier, in this case, $\mathcal{W}_c(\boldsymbol{o}) = \mathcal{W}_{c_3}(\boldsymbol{o})$, and the *envelope* of the family of C_3 curves yields the SWZ. Since the workspace for this manipulator has two translational and one rotational DoF, a

https://doi.org/10.1017/S0263574719001139 Published online by Cambridge University Press

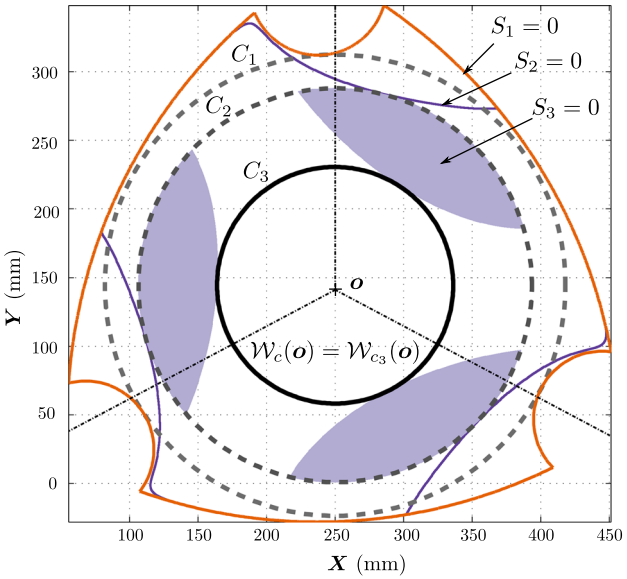

Fig. 8. Zero-level sets of S_i, and C_i in the constant orientation section of the workspace at $\alpha = -10°$. Also shown are the three lines about which the zero-level sets are symmetric.

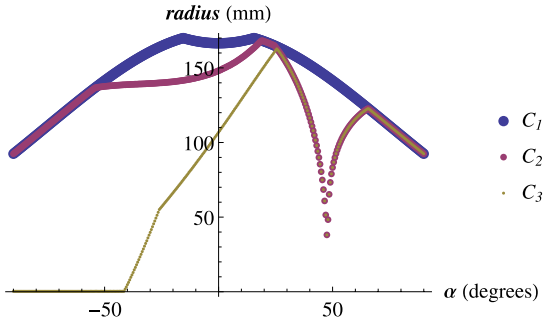

Fig. 9. Radii of C_1, C_2, and C_3 at different values of α for the 3-RRR manipulator.

cylinder with its axis aligned with the yaw axis is an appropriate convex shape of the $\mathcal{W}_c(\boldsymbol{o})$ to be fit inside the $\mathcal{W}(\boldsymbol{o})$.

The shape of the desired cylinder depends on the functional requirements. A longer cylinder provides for larger yaw motions, while a wider one allows larger surge/sway motion. This is more generic than trying to fit in a cylinder of maximal volume, for example.

The variations of the radii of C_1, C_2, and C_3 with respect to α are shown in Fig. 9. The same, in the complete workspace, are shown in Fig. 10(a). It may be noted that the hierarchy described in Section 3 is followed in these plots. As the envelope of the C_3 family has been obtained for the entire range of yaw, $\mathcal{W}_c(\boldsymbol{o})$ can be obtained by fitting a cylinder to the stack for any desired subset of the complete range of yaw. For example, the radius of the cylinder for a desired yaw range of 65.5° is found to be 60 mm (see Fig. 10(b)). Similarly, for a yaw range of 113.5°, the cylinder radius is found to be 38 mm (see Fig. 10(c)). Thus, based on the intended application, a convex shape of interest can be fit into $\mathcal{W}(\boldsymbol{o})$ to obtain the $\mathcal{W}_c(\boldsymbol{o})$ desired.

A comparison of the time taken to compute the SWZ for this manipulator clearly shows the advantage of imposing the hierarchy of evaluation. It can be observed that an improvement of 77% in the time required has been achieved for the same by imposing the proposed hierarchy as compared to without it (see Table I). The improvement in computational time can also be verified by counting the number of grid points of the search space visited during evaluation, which reduces from 69,099,131 to 29,799,618 points when hierarchy is utilised, showing a 57% reduction.

The computational effort can be reduced further, as the prototype of the 3-RRR manipulator considered has a three-way symmetry in its design. Thus, any 120° slice of the workspace about $\boldsymbol{o}$ can

https://doi.org/10.1017/S0263574719001139 Published online by Cambridge University Press

Table I. Comparison of the computation time for finding the SWZ for the 3-RRR manipulator with and without utilisation of hierarchy and symmetry, with a search resolution of 0.125 mm in XY motion and 0.5° in yaw motion of the end-effector.

		Real time for 1000 runs (s)	CPU time for 1000 runs (s)	Average real time for 1 run (s)
Not considering symmetry	Without hierarchy	14,979	104,733	14.97
	With hierarchy	**3352**	**18, 036**	**3.35**
Considering symmetry	Without hierarchy	4752	31,543	4.75
	With hierarchy	**1617**	**8513**	**1.62**

The rows highlighted in bold indicate the fastest computation time achieved for finding the SWZ with and without considering symmetry.

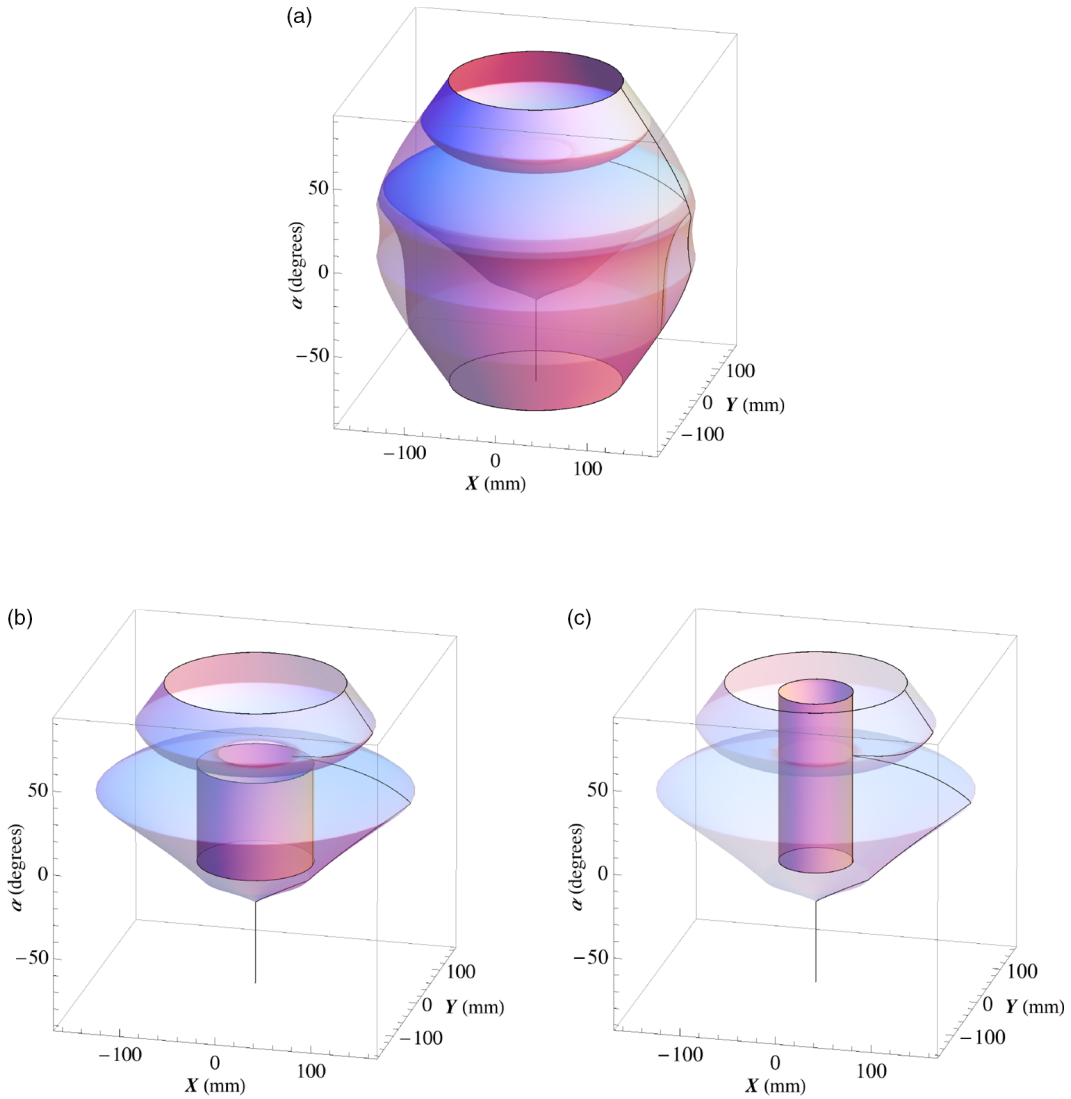

Fig. 10. Envelope of families of C_1, C_2, C_3, and $\mathcal{W}(\boldsymbol{o})$ for the 3-RRR manipulator, with a desired $\mathcal{W}_c(\boldsymbol{o})$ fit to the $\mathcal{W}(\boldsymbol{o})$ – a larger translational (surge and sway) workspace is chosen for (b) as compared to (c), which has a larger yaw workspace.

be chosen as the search space, leading to identical results. A notable reduction in the computational time, namely, 52%, is achieved when this symmetry is exploited, as shown in Table I. Thus, at a resolution of 0.125° in yaw and 0.25 mm in the sway and surge search space, the given design of the 3-RRR manipulator is analysed in merely 1.62 s real time for its SWZ.

https://doi.org/10.1017/S0263574719001139 Published online by Cambridge University Press

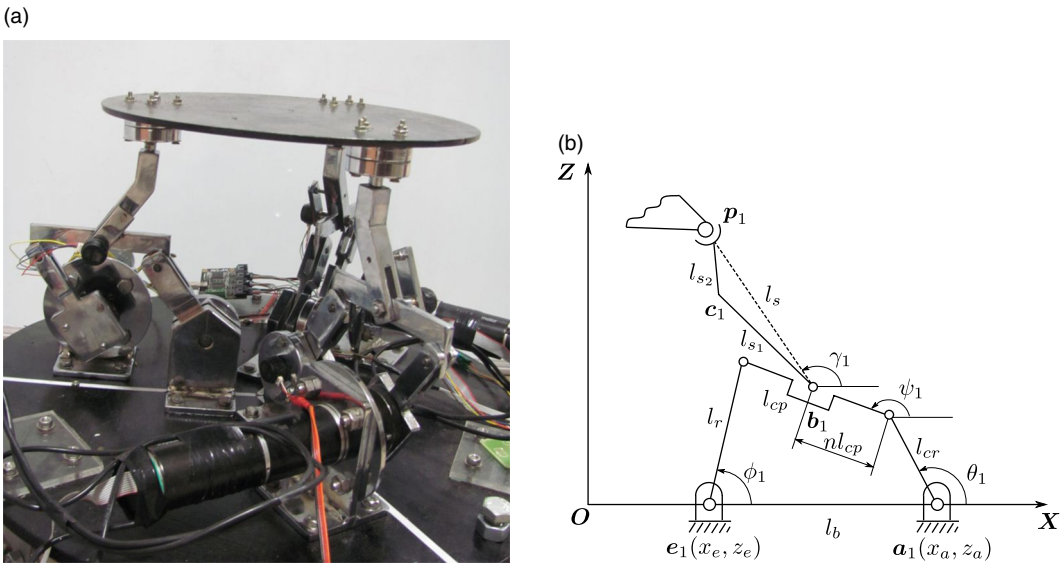

Fig. 11. Prototype and kinematic details of a leg of the MaPaMan-I.

5. Computation of the SWZ for the MaPaMan-I

The generic theoretical framework described in Section 3 is illustrated in this section by an application to a spatial parallel manipulator, namely, MaPaMan-I.[38,39] For the physical dimensions of the manipulator described in ref. [39], the joints have limits on their motions, and there is a link interference. Thus, computation of both $\mathcal{W}_{c_3}(\boldsymbol{o})$ and $\mathcal{W}_{c_4}(\boldsymbol{o})$ is required.

5.1. Geometry of the MaPaMan-I

MaPaMan-I is a 3-DoF manipulator having a motion similar to the 3-RPS (both belong to Class $\boldsymbol{\chi}_{21}$[35]), and its pose can be parametrised in terms of roll (α), pitch (β), and heave (z_c),[39] that is, its end-effector pose variables are given by $\boldsymbol{x} = (\alpha, \beta, z_c)^\top$. The active joint angles are given by $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)^\top$. There are three sets of passive joint variables, given by $\boldsymbol{\phi} = (\phi_1, \phi_2, \phi_3)^\top$ (i.e., rocker orientations), $\boldsymbol{\psi} = (\psi_1, \psi_2, \psi_3)^\top$ (i.e., coupler orientations), and $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \gamma_3)^\top$ (i.e., strut orientations) in each leg of the manipulator (see Fig. 11(b)).

5.2. Condition for the loss-type singularity ($S_1 = 0$)

Each leg of the MaPaMan-I incorporates a four-bar mechanism at its base, which in itself contains a loop and thereby has its own gain-type singularity conditions. For the four-bar mechanism (see Fig. 11(b)), a single constraint equation can be obtained by relating the active variable, θ_i to ψ_i, after eliminating ϕ_i:[39]

$$\begin{aligned} k_i(\theta_i, \psi_i) \triangleq\ & l_b^2 + l_{cr}^2 + l_{cp}^2 - l_r^2 + 2l_b l_{cr}\cos\theta_i + 2l_b l_{cp}\cos\psi_i \\ & +\ 2l_{cr}l_{cp}\cos\theta_i\cos\psi_i + 2l_{cr}l_{cp}\sin\theta_i\sin\psi_i = 0. \end{aligned} \tag{19}$$

The kinematic constraints for each leg can be formulated to relate the end-effector pose variables to the active variables. The length of the strut, l_s, is fixed, hence the loop-closure constraints can be cast as $\boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{x}) = (f_1, f_2, f_3)^\top = \boldsymbol{0}$, where

$$f_i(\theta_i, \psi_i, \boldsymbol{x}) = \|\boldsymbol{b}_i - \boldsymbol{p}_i\|^2 - l_s^2, \quad i = 1, 2, 3. \tag{20}$$

From each pair of $f_i = 0$ and $k_i = 0$, the passive variable ψ_i is eliminated to obtain the constraints of the form given in Eq. (3), that is, $h_i(\theta_i, \boldsymbol{x}) = 0$, $i = 1, 2, 3$. Following the procedure described in Section 3.1.1, the condition for loss-type singularity is obtained using the IFT as $S_1 \triangleq \det\left(\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\theta}}\right) = 0$.

https://doi.org/10.1017/S0263574719001139 Published online by Cambridge University Press

5.3. Condition for the gain-type singularity ($S_2 = 0$)
Differentiating Eq. (19), $\dot{\psi}_i$ can be related to $\dot{\theta}_i$ as $\dot{\psi}_i = \frac{\partial \psi_i}{\partial \theta_i}\dot{\theta}_i$. For instance, when $i = 1$,

$$\frac{\partial \psi_1}{\partial \theta_1} = \frac{l_{cr}((x_a - x_e)\sin\theta_1 - l_{cp}\sin(\theta_1 + \psi_1))}{l_{cp}((x_e - x_a)\sin\psi_1 + l_{cr}\sin(\theta_1 + \psi_1))}. \tag{21}$$

Thus, the Jacobian matrix mapping $\dot{\boldsymbol{\psi}}$ to $\dot{\boldsymbol{\theta}}$ can be written as:

$$\boldsymbol{J}_{\boldsymbol{\psi\theta}} = \text{diag}\left(\frac{\partial \psi_i}{\partial \theta_i}\right),\ i = 1, 2, 3. \tag{22}$$

By choosing link lengths of the four-bar mechanism such that they satisfy Grashof condition, the four-bar mechanism is freed of singularities, and thus, $\boldsymbol{J}_{\boldsymbol{\psi\theta}}$ is always non-singular. Following,[39] the loop-closure constraints for the manipulator are cast in the form $\boldsymbol{\eta}(\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\gamma}) = \mathbf{0}$, which upon time differentiation yields:

$$\mathbf{0} = \frac{d\boldsymbol{\eta}}{dt} = \boldsymbol{J}_{\boldsymbol{\eta\theta}}\dot{\boldsymbol{\theta}} + \boldsymbol{J}_{\boldsymbol{\eta\psi}}\dot{\boldsymbol{\psi}} + \boldsymbol{J}_{\boldsymbol{\eta\gamma}}\dot{\boldsymbol{\gamma}},\ \text{where} \tag{23}$$
$$\boldsymbol{J}_{\boldsymbol{\eta\theta}} = \frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\theta}},\ \boldsymbol{J}_{\boldsymbol{\eta\psi}} = \frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\psi}},\ \text{and}\ \boldsymbol{J}_{\boldsymbol{\eta\gamma}} = \frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\gamma}}.$$

As $\boldsymbol{J}_{\boldsymbol{\psi\theta}}$ is well defined, using $\dot{\boldsymbol{\psi}} = \boldsymbol{J}_{\boldsymbol{\psi\theta}}\dot{\boldsymbol{\theta}}$, Eq. (23) can be reduced to:

$$(\boldsymbol{J}_{\boldsymbol{\eta\theta}} + \boldsymbol{J}_{\boldsymbol{\eta\psi}}\boldsymbol{J}_{\boldsymbol{\psi\theta}})\dot{\boldsymbol{\theta}} + \boldsymbol{J}_{\boldsymbol{\eta\gamma}}\dot{\boldsymbol{\gamma}} = \mathbf{0}. \tag{24}$$

The gain-type singularity occurs when $\dot{\boldsymbol{\gamma}}$ cannot be found uniquely/finitely for a given $\dot{\boldsymbol{\theta}}$. The condition for the corresponding gain-type singularity is found as $S_2 \triangleq \det(\boldsymbol{J}_{\boldsymbol{\eta\gamma}}) = 0$. Examples illustrating the resulting singular configuration of the MaPaMan-I have been discussed in ref. [35].

To determine the SWZ analytically, it is attempted to eliminate the joint variables systematically from $S_2 = 0$, and obtain the expression for the gain-type singularity manifold in the end-effector pose variables in this work. As elaborated in Appendix A, it results in huge and unwieldy expressions ($\approx$ 89GB in size), suggesting the use of numerical schemes for finding the SWZ.

As in the case of the 3-RRR manipulator, the configuration of the links of the manipulator is obtained by performing inverse kinematics, and choosing one branch (i.e., the working mode) for the remaining analysis.

5.4. Conditions for link interference ($S_3 = 0$)
In MaPaMan-I, link interference can be observed between the crank–rocker, and the coupler–strut link pairs. The conditions for these can be modelled as described below:

- The crank is situated such that it always stays above the base of the manipulator. From considerations of the physical design, a restriction is imposed upon the maximum and minimum angles of rotation of the crank, denoted by $\theta_{\min}$ and $\theta_{\max}$ respectively, so that the crank does not interfere with the rocker (see Fig. 12(a)). Thus, $\theta_{\min} < \theta_i < \theta_{\max}$, $i = 1, 2, 3$. The acceptable sets free from interference between the crank–rocker link pair are characterised by $s_{1i} > 0$, and $s_{2i} > 0$, where

$$s_{1i} = \theta_i - \theta_{\min}, \tag{25}$$
$$s_{2i} = \theta_{\max} - \theta_i. \tag{26}$$

- The strut can also interfere with the coupler, which can be seen in Fig. 12(b). The limiting angles made by the strut relative to the coupler, denoted by $\gamma_{\min}$ and $\gamma_{\max}$, define the limiting cases of interference. Here, $\gamma_{\min} < (\pi + \gamma_i - \psi_i) < \gamma_{\max}$, $i = 1, 2, 3$. Thus the acceptable sets are characterised by $s_{3i} > 0$, and $s_{4i} > 0$, where,

$$s_{3i} = (\pi + \gamma_i - \psi_i) - \gamma_{\min}, \tag{27}$$
$$s_{4i} = \gamma_{\max} - (\pi + \gamma_i - \psi_i). \tag{28}$$

https://doi.org/10.1017/S0263574719001139 Published online by Cambridge University Press

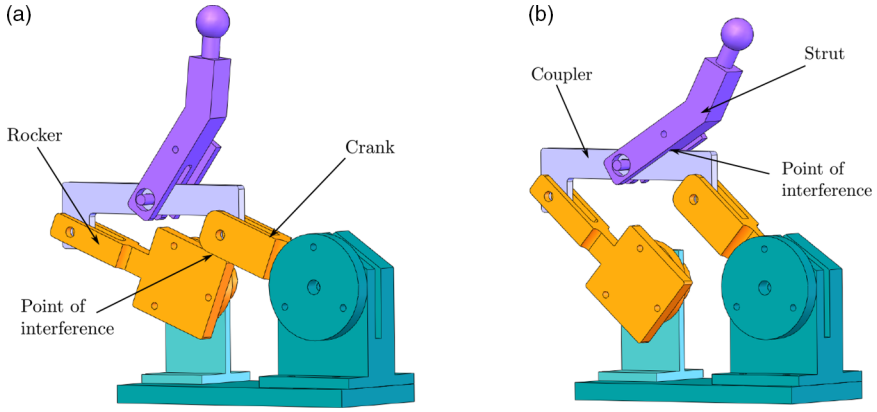

Fig. 12. Interference between the different links of the MaPaMan-I.

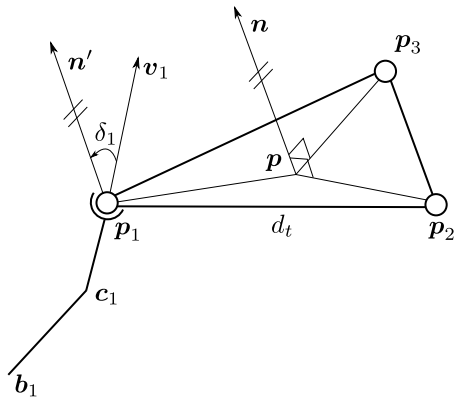

Fig. 13. Joint angle at the spherical joint for MaPaMan-I, shown for the first leg. Here δ_1 represents the angle subtended at the spherical joint between the end-effector and the strut, and $\delta_1 \in (0, \delta_{\max})$. It is evaluated by calculating the angle between the vector along the strut ($\boldsymbol{v}_1$) and the vector $\boldsymbol{n}'$, which passes through the point $\boldsymbol{p}_1$, and is normal to the moving platform.

The boundary function S_3 characterising the acceptable sets, free from the interference between the links, is found by checking each individual function:

$$S_3 : s_{ij} > 0, \quad \text{where } i = 1, 2, 3, \text{ and } j = 1, 2, 3. \tag{29}$$

5.5. Condition for the violation of the joint limits ($S_4 = 0$)

The issue of the joints reaching their physical limits within the desired range of motion is possible at the spherical joints attached to the end-effector. The spherical joints have restricted motions due to the physical dimensions of their constituent mechanical components. These can be modelled as limits imposed on the angle δ_i, such as $0 < \delta_i < \delta_{\max}$ (see Fig. 13). The angle δ_1 is computed by first finding a vector, $\boldsymbol{v}_1$, along the direction of the strut and then measuring the angle between it and the normal, $\boldsymbol{n}$, to the end-effector:

$$\begin{aligned} \boldsymbol{v}_1 &= (\boldsymbol{c}_1 - \boldsymbol{p}_1)/l_s, \\ \boldsymbol{n} &= (\boldsymbol{p}_1 - \boldsymbol{p}_2) \times (\boldsymbol{p}_3 - \boldsymbol{p}_1)/d_t^2, \\ \delta_1 &= \arccos(\boldsymbol{n} \cdot \boldsymbol{v}_1), \end{aligned} \tag{30}$$

where d_t is the length of each side of the top platform. Similarly, δ_2 and δ_3 are computed. Therefore the functions defining the boundary of the desirable set are given by $s_{5i} > 0$, where:

$$s_{5i} = \delta_{\max} - \delta_i. \tag{31}$$

https://doi.org/10.1017/S0263574719001139 Published online by Cambridge University Press

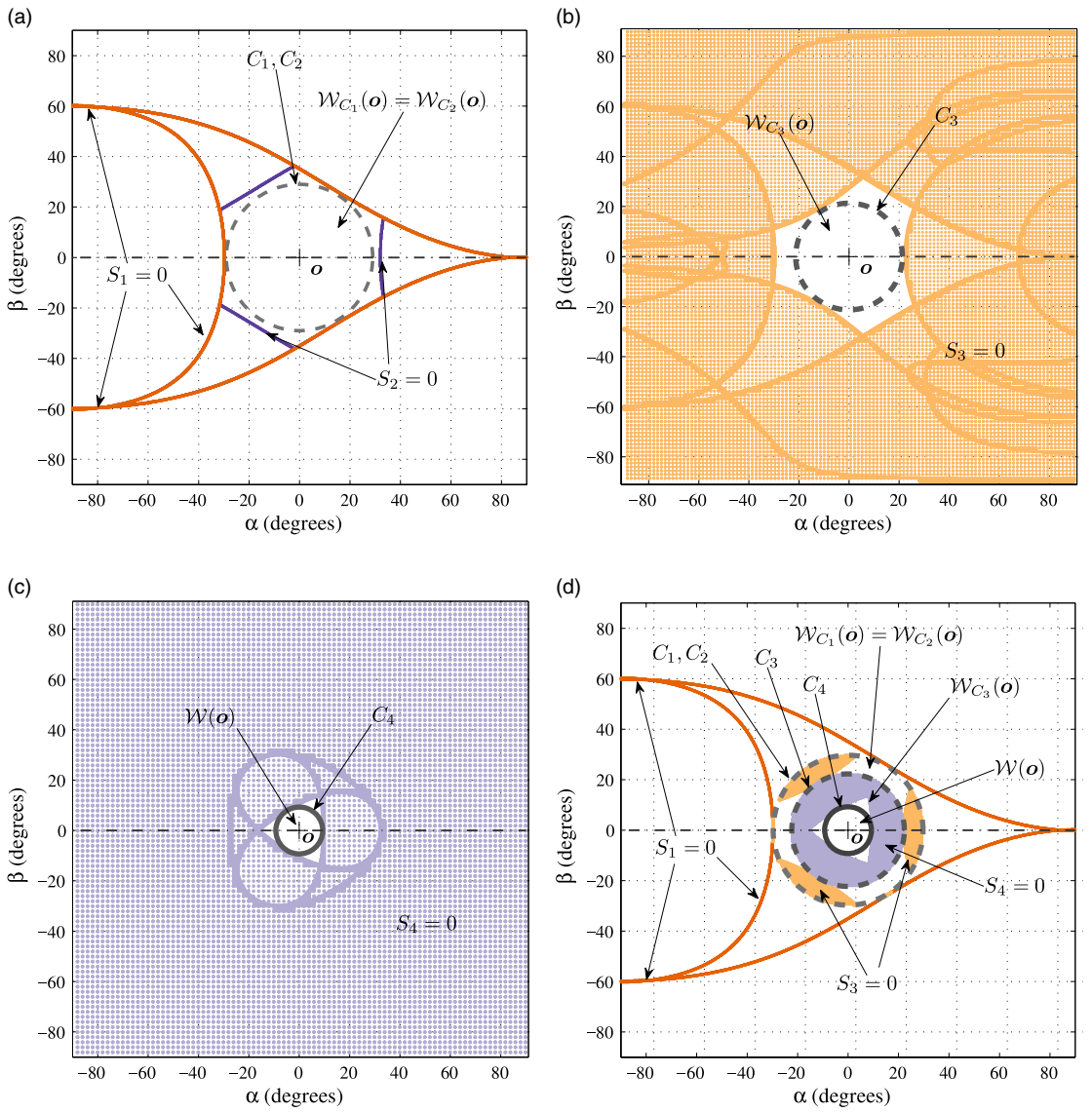

Fig. 14. Zero-level sets of S_i and C_i in the α-β plane of the workspace at $z_c = 125$ mm evaluated with and without hierarchy. Also shown is the symmetry line ($\beta = 0$) for the workspace.

Finally, the boundary function S_4 defining the sets free from the joint limits is obtained checking each individual condition:

$$S_4 : s_{5i} > 0, \quad \text{where} \quad i = 1, 2, 3. \tag{32}$$

5.6. Numerical results

This section describes the results of the application of the above formulations to the MaPaMan-I prototype, whose dimensions are given in Table 1 of ref. [39]. The joint limits used are: $\theta_{\min} = 0°$, $\theta_{\max} = 90°$, $\gamma_{\min} = 30°$, $\gamma_{\max} = 150°$, and $\delta_{\max} = 30°$. The workspace of the MaPaMan-I is parametrised by (α, β, z_c). The boundary functions S_i are not available solely in terms of these variables. Therefore, instead of direct computation of the zero-level sets, the inverse kinematic solutions are used, and the zero-level sets are found by using the numerical scheme described in Section 3.3. The heave motion is taken as the third dimension and is discretised at a resolution of 1 mm to find the zero-level sets of S_1, S_2, S_3, and S_4 in the resulting roll-pitch (α-β) planes. Figure 14 shows all the zero-level sets at the $z_c = 125$ mm plane. Note that the points satisfying $S_2 = 0$ fall outside $\mathcal{W}_{c_1}(\boldsymbol{o})$ as seen in Fig. 14(a) in this particular case, and therefore $C_2 = C_1$. The circle C_3 is found next. The parts of S_4 appearing inside $\mathcal{W}_{c_3}(\boldsymbol{o})$ alone are considered for the computation of $\mathcal{W}_{c_4}(\boldsymbol{o})$, which are

https://doi.org/10.1017/S0263574719001139 Published online by Cambridge University Press

Table II. Comparison of the computation time for finding the SWZ of the MaPaMan-I, with and without utilisation of hierarchy and symmetry, with a search resolution of 0.125° in roll and pitch and 1.0 mm in heave.

		Real time for 1000 runs (s)	**CPU time for 1000 runs (s)**	**Average real time for one run (s)**
Not considering symmetry	Without hierarchy	8443	55,467	8.44
	With hierarchy	**912**	**4563**	**0.91**
Considering symmetry	Without hierarchy	4203	27,608	4.20
	With hierarchy	**745**	**3606**	**0.75**

The rows highlighted in bold indicate the fastest computation time achieved for finding the SWZ.

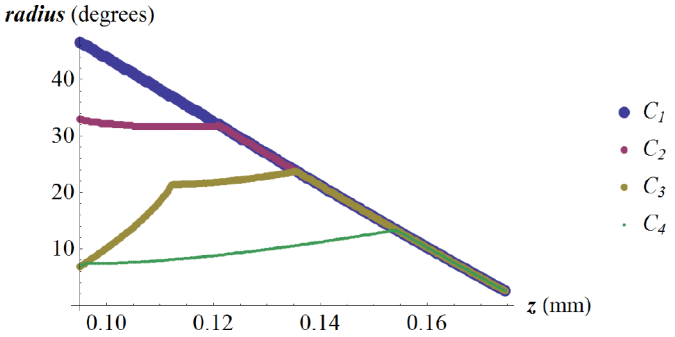

Fig. 15. Radii of C_1, C_2, C_3, and C_4 at different values of heave (z) for the MaPaMan-I.

shown in Fig. 14(d). As noted earlier, in this case, $\mathcal{W}_c(\boldsymbol{o}) = \mathcal{W}_{c_4}(\boldsymbol{o})$. Naturally, the envelope of C_4 obtained for all the planes, when put together, yields a subset of $\mathcal{W}(\boldsymbol{o})$ that is convex in each plane. This does not necessarily imply that the envelope delimits a convex region as a whole. However, one can choose a convex volume inside this envelope to obtain the desired $\mathcal{W}_c(\boldsymbol{o})$ in the (α-β-z_c) space. Due to the nature of the DoF of the manipulator (of Class $\boldsymbol{\chi}_{21}$,[35] i.e., two rotational and one translational DoF), the cylinder is chosen as the convex shape to be fit inside $\mathcal{W}(\boldsymbol{o})$, with the heave motion occurring along the axis of the cylinder.

The variation of the radii of C_1, C_2, C_3, and C_4 with heave motion is shown in Fig. 15, and their respective envelopes in the complete workspace are visualised in Fig. 16(a). As the envelope of C_4 has been obtained for the entire range of heave motion, $\mathcal{W}_c(\boldsymbol{o})$ can be obtained by fitting a cylinder to the envelope for any desired subset of the complete range of heave motion. For example, a large desired heave workspace of 50 mm is fit in the SWZ and has a maximum tilt workspace of 8.17° as shown in Fig. 16(b). Similarly, a cylinder with a desired radius of 10° is fit in the SWZ, as shown in Fig. 16, resulting in a heave range of 29 mm.

A comparison of the time taken to compute the SWZ for this manipulator clearly shows the advantage of incorporating the hierarchy in the computation process (see Table II). It can be observed that a reduction of 89% in the time taken for computation has been achieved by imposing the proposed hierarchy, as compared to without it, thus illustrating its advantage. The reduction in time can be visualised at a particular heave position, by comparing the set of points evaluated when not considering the hierarchy (see Fig. 14(a–c)), with the case where hierarchy is employed (see Fig. 14(d)).

It has to be noted that the time taken for the 3-RRR manipulator is high as compared with the MaPaMan-I. This is expected since the link interference function of the 3-RRR manipulator is computationally more expensive as it yields a surface patch as the solution at each plane (see $S_3 = 0$ in Fig. 8), whereas all the other boundary functions are 1-D curves. Hence, the improvement in the time taken for the computation is also pronounced when hierarchy is imposed, justifying the ordering of the boundary functions.

Additionally, the symmetry of the manipulator architecture is utilised in reducing the search space for finding the zero-level sets. In the case of the 3-RRR manipulator, the three-way symmetry was inherent in the architecture of the manipulator itself. In this case, too, the manipulator has a three-way symmetry. However, when the workspace of the manipulator is parametrised in terms of roll, pitch, and heave, every constant heave section has a two-way symmetry in the resulting roll-pitch plane.

https://doi.org/10.1017/S0263574719001139 Published online by Cambridge University Press

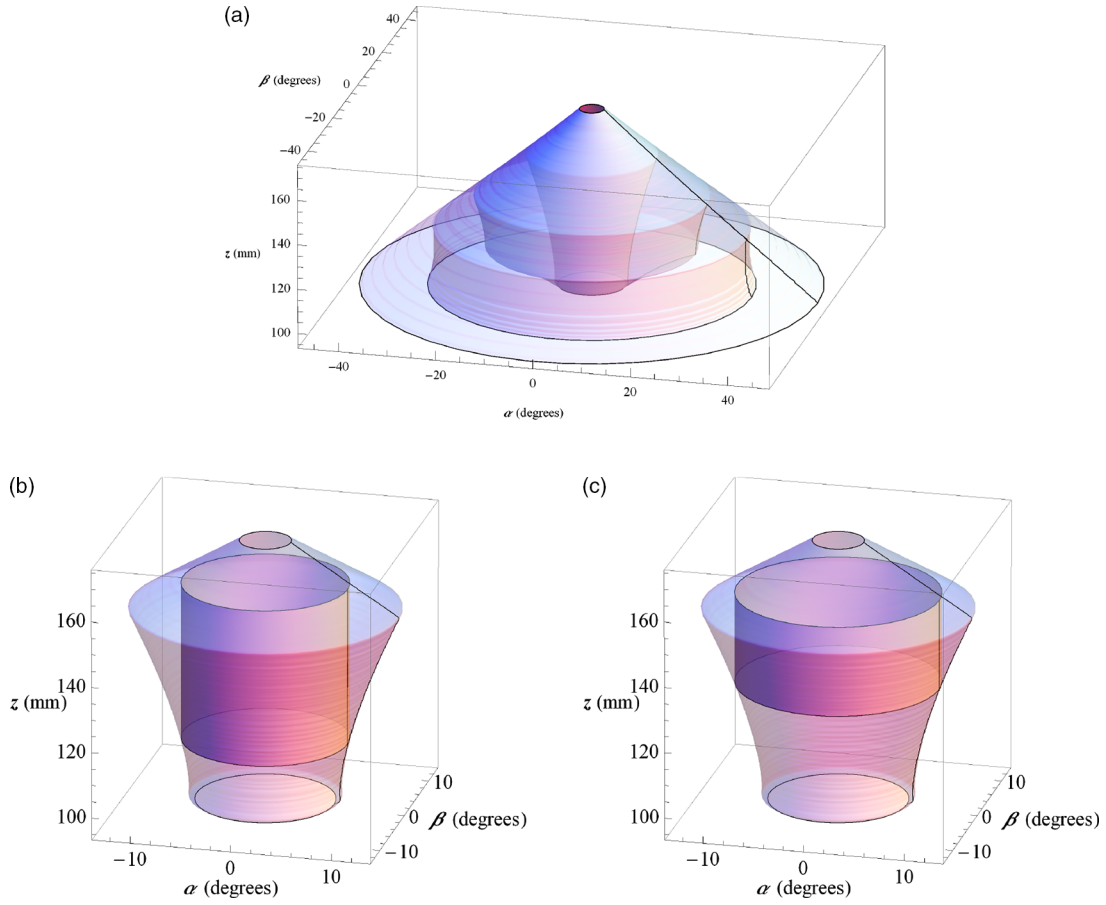

Fig. 16. Envelope of C_1, C_2, C_4, and $\mathcal{W}(\boldsymbol{o})$ for the MaPaMan-I, with a desired $\mathcal{W}_c(\boldsymbol{o})$ fit to $\mathcal{W}(\boldsymbol{o})$, with (b) showing a large desired heave workspace as compared with (c), which has a larger desired tilt (pitch and roll) workspace.

Exploitation of this symmetry affords a significant reduction in the required computations, without any loss of information/resolution. The time taken to compute the SWZ utilising the symmetry is also mentioned in Table II, which shows a further 18% reduction. Thus, one design of the MaPaMan-I is completely analysed and its SWZ is computed in 0.75 s of real time. The search resolution for finding the SWZ is fixed at 1.0 mm in the heave dimension, and 0.125° in both the roll and pitch dimensions. The resolution can be increased to a desired accuracy level in the end-effector pose variables, which when set to 0.1 mm and 0.025°, increases the real time for computation only to 75 s.

5.6.1. Comparison of the numerical scheme with some of the existing methods. The proposed numerical scheme is compared with a few recently developed methods from literature for a better understanding of the computational efficiency. The results are summarised in Table III. It is to be noted that the objectives of these methods do not match those of the present work identically, that is, either they do not consider all the four issues described, or they additionally identify maximal workspaces.[20] Nevertheless, these present the best options for a comparison of the numerical efficiency (to the best of the knowledge of the authors). The comparisons here are made directly with the results available in the literature, and the authors have not reproduced them, thus the computational power used is also not identical. The time taken for computing the maximal singularity-free workspace (MFSC) in ref. [20] is reported to be 1.5 s for the 3-PRR, at a fixed orientation. The resolution for this work can be represented by the smallest limit on the workspace area, which forms a pie sector of radius 0.01 (length normalised by the base dimension) and a subtended angle of 1°. In comparison, using the proposed formulation, the SWZ of the 3-RRR manipulator at a fixed orientation is evaluated in 0.03 s, with a resolution of 0.0025 normalised length[11]. Similarly, computation of

[11]The computational ability is adjusted for a more realistic comparison, by using only a single processor thread to execute the proposed scheme just for this case.

https://doi.org/10.1017/S0263574719001139 Published online by Cambridge University Press

Table III. Computational costs of the proposed algorithm as well as those existing in literature.

Article/method and issues considered[+]	Manipulator and search dimension	Real time[#] (s) × threads	Linear resolution[*]	Angular resolution
Kalazoori et al.,[20] {1,2,5}	3-PRR, 2 (fixed orientation)	1.500 × 1	0.100	1.000°
	Planar 2-DoF, 2	≈10 × 160	0.010	–
Bohigas et al.,[30] {1}	Stewart–Gough, 3 (fixed position)	79 × 160	0.020	–
	Stewart–Gough, 3 (fixed orientation)	2554 × 160	0.250	–
Huang et al.,[40] {2,3,4}	Stewart–Gough, 3 (fixed position)	14,400 × 1	Not reported	
Proposed, {1,2,3,4}	3-RRR, 3-DoF	1.620 × 8	0.0025	0.500°
	MaPaMan-I, 3-DoF	0.745 × 8	0.010	0.125°

[+]The issues considered are enumerated as: (1) loss-type singularity, (2) gain-type singularities, (3) joint limits, (4) link interference, and (5) largest convex subset of the workspace.
[*]The linear resolution is normalised by the base dimension.
[#]It is not clearly mentioned in literature whether the time presented is CPU time or real time. The results of the proposed scheme are reported in terms of the real time.

the singularities alone in the configuration space using a numerical algorithm takes time in the order of 10 s, employing a Xeon processor grid running 160 threads, for a planar 2-DoF manipulator, at a resolution of 0.01 normalised length units.[30] In comparison, the SWZ of the 3-RRR manipulator is evaluated in 1.62 s real time using a single processor running eight threads, with the division of orientation subspace in steps of 0.01 radians. Furthermore, identification of the zero-level sets is subjected to a numerical tolerance (10^{-5}) on the function values in ref. [30], whereas the proposed method identifies zero in an *absolute* sense due to nature of the algorithm used. Similarly, the method presented in ref. [40] finds the orientation workspace of a Stewart–Gough platform, at a fixed position, free from singularities and link interference in 14,400 s, is subjected to a numerical tolerance on the function value. In comparison, the computation of the SWZ of MaPaMan-I, which also results in a 3-DoF search, is completed in a real time of 0.75 s, using the proposed method. In another recent work,[22] a numerical scheme is proposed for finding the constant orientation workspace for a cable-driven parallel manipulator in less than 0.03 s. But once again, the number of constraint function checks makes this hard to do a fair comparison with the proposed method.

6. Conclusions

This paper has described a set of computational strategies for the determination of the SWZ of planar and spatial parallel manipulators having three degrees-of-freedom. For a parallel manipulator, the knowledge of its SWZ is very important, as it allows the manipulator to move freely inside this region. Further, the SWZ needs to be computed for a given manipulator only once. The algorithm depends primarily on the computation of the zero-level sets of a set of functions. It is generic enough to handle numerically evaluated functions of both smooth and non-smooth nature. The examples show that very high resolutions can be achieved in practice at the cost of a few CPU-seconds only, even in the case of a 3-DoF spatial parallel manipulator. The computational economy of the proposed method allows it to be the backbone of design procedures for parallel manipulators for prescribed SWZ dimensions/locations. Future extensions include applications to spatial parallel manipulators with six-DoF, such as the Stewart platform, and the 6-RSS manipulators.

https://doi.org/10.1017/S0263574719001139 Published online by Cambridge University Press

References

1. R. A. Srivatsan and S. Bandyopadhyay, "Determination of the Safe Working Zone of a Parallel Manipulator," *Proceedings of the International Conference on Computational Kinematics*, Barcelona, Spain (2014) pp. 201–208.
2. C. Gosselin and J. Angeles, "Singularity analysis of closed-loop kinematic chains," *IEEE Trans. Rob. Autom.* **6**, 281–290 (1990).
3. L. Tsai, *Robot Analysis: The Mechanics of Serial and Parallel Manipulators* (A Wiley-Interscience Publication, Wiley, 1999).
4. A. Ghosal and B. Ravani, "A differential-geometric analysis of singularities of point trajectories of serial and parallel manipulators," *Trans. ASME, J. Mech. Design* **123**(1), 80–89 (2001).
5. S. Bandyopadhyay and A. Ghosal, "Analysis of configuration space singularities of closed-loop mechanisms and parallel manipulators," *Mech. Mach. Theory* **39**(5), 519–544 (2004).
6. A. Ghosal, *Robotics: Fundamental Concepts and Analysis* (Oxford University Press, New Delhi, 2014).
7. J.-P. Merlet, *Parallel Robots* (Springer Science & Business Media, Springer Netherlands, Berlin/Heidelberg, 2006).
8. O. Bohigas, M. E. Henderson, L. Ros, M. Manubens and J. M. Porta, "Planning singularity-free paths on closed-chain manipulators," *IEEE Trans. Robot.* **29**(4), 888–898 (2013).
9. A. K. Dash, I.-M. Chen, S. H. Yeo and G. Yang, "Workspace generation and planning singularity-free path for parallel manipulators," *Mech. Mach. Theory* **40**(7), 776–805 (2005).
10. S. Sen, B. Dasgupta and A. K. Mallik, "Variational approach for singularity-free path-planning of parallel manipulators," *Mech. Mach. Theory* **38**(11), 1165–1183 (2003).
11. S. Leguay-Durand and C. Reboulet, "Optimal design of a redundant spherical parallel manipulator," *Robotica* **15**(4), 399–405 (1997).
12. A. Agarwal, C. Nasa and S. Bandyopadhyay, "Dynamic singularity avoidance for parallel manipulators using a task-priority based control scheme," *Mech. Mach. Theory* **96**, 107–126 (2016).
13. H. Li, C. M. Gosselin and M. J. Richard, "Determination of maximal singularity-free zones in the workspace of planar three-degree-of-freedom parallel mechanisms," *Mech. Mach. Theory* **41**(10), 1157–1167 (2006).
14. Q. Jiang and C. M. Gosselin, "The maximal singularity-free workspace of the Gough-Stewart platform for a given orientation," *J. Mech. Design* **130**(11), 1281–1293 (2008).
15. G. Abbasnejad, H. M. Daniali and S. M. Kazemi, "A new approach to determine the maximal singularity-free zone of 3-RPR planar parallel manipulator," *Robotica* **30**(6), 1005–1012 (2012).
16. M. Zein, P. Wenger and D. Chablat, "Singularity Surfaces and Maximal Singularity-Free Boxes in the Joint Space of Planar 3-RPR Parallel Manipulators," *Proceeding of 12th IFToMM World Congress*, Besançon, France (2007) pp. 1–6.
17. Y. Yang and J. O'Brien, "A Geometric Approach for the Design of Singularity-free Parallel Robots," *Proceedings of the IEEE International Conference on Robotics and Automation*, Kobe, Japan (2009) pp. 1801–1806.
18. F. Pernkopf and M. L. Husty, "Workspace analysis of Stewart-Gough-type parallel manipulators," *Proc. Inst. Mech. Eng. C* **220**(7), 1019–1032 (2006).
19. T. Stigger, M. Pfurner and M. Husty, "Workspace and Singularity Analysis of a 3-RUU Parallel Manipulator," *Proceedings of the European Conference on Mechanism Science*, Aachen, Germany (2018) pp. 325–332.
20. M.-H. F. Kaloorazi, M. T. Masouleh and S. Caro, "Determination of the maximal singularity-free workspace of 3-DOF parallel mechanisms with a constructive geometric approach," *Mech. Mach. Theory* **84**, 25–36 (2015).
21. A. Nag, V. Reddy, S. Agarwal and S. Bandyopadhyay, "Identifying singularity-free spheres in the position workspace of semi-regular Stewart platform manipulators," *Proceedings of the International Conference on Advances in Robot Kinematics*, Bologna, Italy (2018) pp. 421–430.
22. A. Pott, "Efficient Computation of the Workspace Boundary, its Properties and Derivatives for Cable-Driven Parallel Robots," *Proceedings of the International Conference on Computational Kinematics*, Poitiers, France (2018) pp. 190–197.
23. A. N. Chaudhury and A. Ghosal, "Optimum design of multi-degree-of-freedom closed-loop mechanisms and parallel manipulators for a prescribed workspace using Monte Carlo method," *Mech. Mach. Theory* **118**, 115–138 (2017).
24. J. Kilaru, M. K. Karnam, S. Agarwal and S. Bandyopadhyay, "Optimal Design of Parallel Manipulators Based on Their Dynamic Performance," *Proceedings of the 14th IFToMM World Congress*, Taipei, Taiwan (2015) pp. 406–412.
25. D. Patel, R. Kalla, H. Tetik, G. Kiper and S. Bandyopadhyay, "Computing the Safe Working Zone of a 3-RRS Parallel Manipulator," *Proceedings of the 6th European Conference on Mechanism Science (EUCOMES)*, Nantes, France (2016) pp. 113–120.
26. Y. Cao, Q. Zhang, H. Zhou and H. Ding, "Orientation-workspace analysis of the Stewart-Gough manipulator," *Discrete Cont. Dyn. B* **13**, 646–652 (2006).
27. J.-P. Merlet, "Designing a parallel manipulator for a specific workspace," *Int. J. Rob. Res.* **16**(4), 545–556 (1997).

28. E. Macho, O. Altuzarra, C. Pinto and A. Hernandez, "Workspaces associated to assembly modes of the 5R planar parallel manipulator," *Robotica* **26**(3), 395–403 (2008).
29. E. Macho, O. Altuzarra and A. Hernandez, "Maximal Operational Workspace of Parallel Manipulators," **In:** *Robot Manipulators, New Achievements* (A. Lazinica and H. Kawai, eds.) (In-tech, Croatia, 2010) pp. 577–600.
30. O. Bohigas, D. Zlatanov, L. Ros, M. Manubens and J. Porta, "A general method for the numerical computation of manipulator singularity sets," *IEEE Trans. Rob.* **30**(2), 340–351 (2014).
31. J. Hudgens and T. Arai, "Planning Link-interference-free Trajectories for a Parallel link Manipulator," *Proceeding of the International Conference on Industrial Electronics, Control, and Instrumentation*, New Orleans, LA (1993) pp. 1506–1511.
32. N. Chakraborty, J. Peng, S. Akella and J. E. Mitchell, "Proximity queries between convex objects: An interior point approach for implicit surfaces," *IEEE Trans. Rob.* **24**(1), 211–220 (2008).
33. S. Agarwal, R. A. Srivatsan and S. Bandyopadhyay, "Analytical determination of the proximity of two right-circular cylinders in space," *J. Mech. Rob.* **8**(4), 041010 (2016).
34. E. G. Gilbert, D. W. Johnson and S. S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space," *IEEE J. Rob. Autom.* **4**(2), 193–203 (1988).
35. R. A. Srivatsan, S. Bandyopadhyay and A. Ghosal, "Analysis of the degrees-of-freedom of spatial parallel manipulators in regular and singular configurations," *Mech. Mach. Theory* **69**(11), 127–141 (2013).
36. A. Nag and S. Bandyopadhyay, "Analytical Determination of a Sphere Inside Which the Stewart Platform Translates Without Suffering Any Leg Interference," *Proceedings of the International Conference on Advances in Robot Kinematics*, Bologna, Italy (2018) pp. 74–82.
37. Wolfram Research, Inc., *Mathematica*, Version 9.0 (Champaign, Illinois, 2012).
38. S. Bandyopadhyay, R. A. Srivatsan and T. Mehta, "A reconfigurable parallel manipulator," *Indian Patent Application*, 5187/CHE/2012, 2012.
39. R. A. Srivatsan and S. Bandyopadhyay, "On the position kinematic analysis of MaPaMan: A reconfigurable three-degrees-of-freedom spatial parallel manipulator," *Mech. Mach. Theory* **62**, 150–165 (2013).
40. C. Huang and K. Tsai, "A general method to determine compatible orientation workspaces for different types of 6-DOF parallel manipulators," *Mech. Mach. Theory* **85**, 129–146 (2015).

Appendix A Singularity Manifold of the MaPaMan-I

In order to obtain the gain-type singularity manifold, the procedure described in Section 3.1.3 is followed. Equations (19) and (20) can be solved pairwise to eliminate the passive variables, ψ_i, to obtain three equations of the form:

$$h_i(\theta_i, \boldsymbol{x}) = 0, \;\; i = 1, 2, 3. \tag{A1}$$

As explained in Section 5.3, the condition for gain-type singularity is given as:

$$S_2 = h_4(\boldsymbol{\theta}, \boldsymbol{x}) = \det(\boldsymbol{J_{hx}}) = 0, \;\; \text{where } \boldsymbol{J_{hx}} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}. \tag{A2}$$

Thus, upon eliminating θ_i, $i = 1, 2, 3$, from the four equations $\boldsymbol{h}(\boldsymbol{\theta}, \boldsymbol{x}) = 0$, the gain-type singularity manifold in the end-effector pose variables can be obtained. The variable $\boldsymbol{\theta}$ appearing in these as trigonometric functions, that is, $\cos\theta_i$ and $\sin\theta_i$, is converted to algebraic functions by the standard half-tangent substitutions:

$$\cos\theta_i = \frac{1 - t_i^2}{1 + t_i^2}, \quad \sin\theta_i = \frac{2t_i}{1 + t_i^2}, \quad \text{where } t_i = \tan\left(\frac{\theta_i}{2}\right), \tag{A3}$$

to obtain $g_i(t_i, \boldsymbol{x}) = 0$, from $h_i(\theta_i, \boldsymbol{x}) = 0$, $i = 1, 2, 3$, and $g_4(t_1, t_2, t_3, \boldsymbol{x}) = 0$ from $h_4(\boldsymbol{\theta}, \boldsymbol{x}) = 0$, respectively. The following scheme is used to eliminate t_i in a step-by-step manner:

$$\left.\begin{matrix} g_1(t_1, \boldsymbol{x}) = 0 \\ g_4(t_1, t_2, t_3, \boldsymbol{x}) = 0 \end{matrix}\right) \xrightarrow{\times t_1} \left.\begin{matrix} g_5(t_2, t_3, \boldsymbol{x}) = 0 \\ g_2(t_2, \boldsymbol{x}) = 0 \end{matrix}\right) \xrightarrow{\times t_2} \left.\begin{matrix} g_6(t_3, \boldsymbol{x}) = 0 \\ g_3(t_3, \boldsymbol{x}) = 0 \end{matrix}\right) \xrightarrow{\times t_3} g_7(\boldsymbol{x}) = 0. \tag{A4}$$

The functions g_1 and g_4 are quadratic in t_1. The variable t_1 is eliminated from $g_1 = 0$ and $g_4 = 0$ to obtain a new equation, $g_5(t_2, t_3, \boldsymbol{x}) = 0$, which is quartic in both t_2 and t_3. As the size of the expression

https://doi.org/10.1017/S0263574719001139 Published online by Cambridge University Press

is large (≈ 4.3 MB), the coefficients of t_2 and t_3 in g_5 are replaced by symbols A_i, $i = 1, \ldots, 9$, (which shall be referred to as *dummy coefficients*[12]) such that:

$$g_5 = A_1 t_2^2 t_3^2 + A_2 t_2^2 t_3 + A_3 t_2^2 + A_4 t_2 t_3^2 + A_5 t_2 t_3 + A_6 t_2 + A_7 t_3^2 + A_8 t_3 + A_9. \tag{A5}$$

The function g_5 is quartic in t_2, while g_2 is quadratic in t_2. Dividing g_5 by g_2 by treating both as polynomials in t_2 returns a remainder that is no more than linear in t_2. Setting the remainder to zero and solving linearly to obtain t_2 and substituting it back in $g_2 = 0$ yields $g_6(t_3, \boldsymbol{x}) = 0$, where g_6 is an 8-degree polynomial in t_3. The coefficients of t_3 in g_6 are once again replaced by dummy coefficients due to their large size. The function g_3 is quadratic in t_3 and hence t_3 is eliminated from g_6 and g_3 by following the same process as in the previous stage of elimination to obtain $g_7(\boldsymbol{x})$, that is free of t_i, $i = 1, 2, 3$. Upon substituting back the actual expressions of the dummy coefficients into the final expression, one obtains an extremely large expression (≈ 89 GB in size), whose coefficients are obtained in the closed form in terms of length parameters in the end-effector pose variables.

Appendix B Singularity Manifold of the 3-RRR Manipulator

The inverse kinematics equations are cast in the form $h_i(\theta_i, \boldsymbol{x}) = 0$, $i = 1, 2, 3$ as given in Eq. (9). The condition for gain-type singularity can be obtained as given in Section 4.3 as:

$$h_4(\boldsymbol{\theta}, \boldsymbol{x}) \triangleq \det(\boldsymbol{J_{hx}}) = 0. \tag{B1}$$

As in the case of the MaPaMan-I, these are four equations, with three active variables, $\boldsymbol{\theta}$, which can be reduced to obtain the gain-type singularity manifold in end-effector pose variables. Upon converting the functions of $\boldsymbol{\theta}$ to their algebraic forms using the standard half-tangent substitutions, the corresponding functions $g_1(t_1, \boldsymbol{x})$, $g_2(t_2, \boldsymbol{x})$, $g_3(t_3, \boldsymbol{x})$, and $g_4(t_1, t_2, t_3, \boldsymbol{x})$ are obtained, where, t_i is given by Eq. (A3). The same elimination scheme as used for finding the gain-type singularity manifold of the MaPaMan-I is used to eliminate t_i, $i = 1, 2, 3$, in a step-by-step manner. Using dummy coefficients, the sizes of the equations are reduced and the variables are eliminated to finally obtain an expression (≈ 40 GB in size) whose coefficients are obtained in the closed-form in terms of the length parameters and the end-effector pose variables.

[12]This reduces the size of the expression while retaining the algebraic structure of the equations, albeit at the cost of the potential simplifications that may be possible only with the knowledge of the actual coefficients.

https://doi.org/10.1017/S0263574719001139 Published online by Cambridge University Press