D. Oetomo[1]
Department of Mechanical Engineering,
University of Melbourne,
Melbourne, VIC 3010, Australia;
INRIA Sophia Antipolis,
BP93,
06902 Sophia Antipolis Cedex, France
e-mail: doetomo@unimelb.edu.au

D. Daney
INRIA Sophia Antipolis,
BP93,
06902 Sophia Antipolis Cedex, France
e-mail: david.daney@sophia.inria.fr

B. Shirinzadeh
Department of Mechanical Engineering,
Monash University,
Clayton, VIC 3800, Australia
e-mail: bijan.shirinzadeh@eng.monash.edu.au

J.-P. Merlet
INRIA Sophia Antipolis,
BP93,
06902 Sophia Antipolis Cedex, France
e-mail: jean-pierre.merlet@sophia.inria.fr

An Interval-Based Method for Workspace Analysis of Planar Flexure-Jointed Mechanism

This paper addresses the problem of certifying the performance of a precision flexure-based mechanism design with respect to the given constraints. Due to the stringent requirements associated with flexure-based precision mechanisms, it is necessary to be able to evaluate and certify the performance at the design stage, taking into account the possible sources of errors such as fabrication tolerances and the modeling inaccuracies in flexure joints. An interval-based method is proposed to certify whether various constraints are satisfied for all points within a required workspace. Unlike the finite-element methods that are commonly used today to evaluate a design, where material properties are used for evaluation on a point-to-point sampling basis, the proposed technique offers a wide range of versatility in the design criteria to be evaluated and the results are true for all continuous values within the certified range of the workspace. This paper takes a pedagogical approach in presenting the interval-based methodologies and the implementation on a planar 3revolute-revolute-revolute (RRR) parallel flexure-based manipulator.
[DOI: 10.1115/1.3042151]

1 Introduction

Flexure-jointed mechanisms [1] have been widely utilized in precision positioning and manipulation devices [2], such as for pattern alignment in semiconductor fabrication, micro-assembly, microsurgery, and various scanning microscopy techniques [3–5]. It has also been used for the design of microscaled mechanisms such as microgrippers and microtransducers, especially as the scale involved does not allow the use of conventional roller-bearing type of joints.

Due to the high precision nature of the applications, there is a stringent demand on the performance of the manipulator. Currently, finite-element methods are often employed to simulate the performance of the mechanism design; however, this is limited in the types of criteria and does not allow a guaranteed solution. It also tends to be computationally intensive. Topological information of the mechanism is often not considered and evaluation of the performance is on a point sampling basis. It is therefore difficult to guarantee that the manipulator satisfies the required constraints for all poses of its required workspace. Furthermore, there is also the issue of modeling inaccuracies in flexure mechanisms. Generally, when constructing the kinematic model of a flexure-joint mechanism, an ideal model is used for the flexure joint, for example, a revolute flexure joint is modeled as an ideal revolute joint. The deformation of the joint during deflection, however, produces residual translational motion. An attempt to take into account this residual motion was given in Ref. [6], where a revolute flexure joint was represented as a pair of revolute and prismatic joints. However, an accurate model for such parasitic motion is complex to obtain.

In this paper, an interval-based method is proposed to evaluate the various constraints and to certify whether or not they are achieved within the desired workspace of the flexure-based manipulator. This would allow a designer to verify that various constraints, such as the reachable range of motion and the required motion resolutions, are achieved with the specific design. Furthermore, various uncertainties, including the inaccuracies of joint modeling and fabrication tolerances, can be accommodated as bounded variations in the kinematic parameters. When a given workspace is certified as having satisfied all the constraints, the certification is valid for all the continuous values within the bounds, even in the face of the above-mentioned uncertainties.

The work on obtaining the workspace of a manipulator has been presented in the past through the geometrical approach [7,8] and the screw theory [9]. These methods define the boundaries of the manipulators geometrically and provide algebraic expressions to the boundary curves. The method presented in this paper obtains the same results as the analysis presented in Ref. [7]—where the workspace boundary of a specific manipulator was described through the geometric approach. The interval analysis method proposed in this paper has the advantages of being more general, easily adaptable to other kinematic topologies, capable of handling any type of constraints expressed as mathematical equalities/inequalities, and uncertainties in parameters and rounding error. Geometric approaches, in comparison, require complex mathematical derivation prior to numerical computation to obtain the workspace boundaries specific to the mechanism, however, it is computationally more efficient. It also calculates only the workspace boundary given the topology, but it is not able to take into account joint limits or any other types of constraints, such as motion resolution and singularities.

This paper presents a complete workspace evaluation technique for a flexure mechanism. A brief introduction of flexure mechanisms and the overall constraint satisfaction method through interval analysis are presented in Secs. 2 and 3, respectively. The implementation of the algorithm, through constraints commonly required for precision flexure mechanism, namely,

[1]Corresponding author.

Contributed by the Design Theory and Methodology Committee of ASME for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received February 11, 2008; final manuscript received September 23, 2008; published online December 16, 2008. Review conducted by Hong S. Yan. Paper presented at the IEEE Conference of Robotics and Automation, 2008.

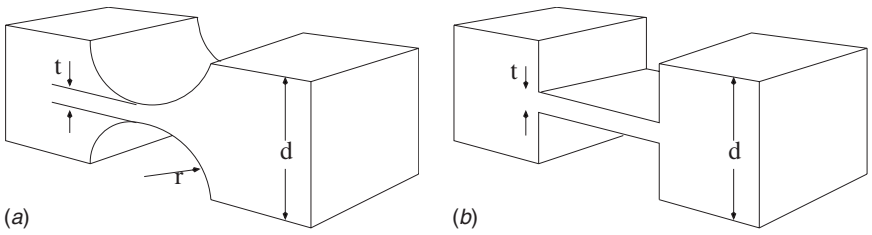

Fig. 1 Types of common flexure joints: (*a*) notch type flexure joint and (*b*) leaf type flexure joint

- the reachable workspace, given the allowable joint space displacement within the linear deformation of all the flexure joints in the mechanism and taking into account the uncertainties in the link lengths bounded within the fabrication tolerance
- singularity-free workspace
- the required motion resolution at the end-effector, given the joint space motion resolution

is presented in Sec. 4, illustrated through the example of a 3RRR planar flexure mechanism. This paper discusses the results of the implementation through the direct implementation of the described algorithm and through more advanced interval techniques available to achieve a better (sharper) result. Concepts of the advanced techniques are elaborated, with proper references given to cover the details of these techniques. A note on the numerical efficiency of the algorithm is given to summarize the discussion.

2 Flexure Mechanisms

A flexure mechanism [1] is formed by significantly reducing the cross-sectional area of a member at a particular point so that deflection through elastic deformation can be induced about that point while treating the rest of the member as an ideal rigid body. As such, flexure joints do not suffer from any nonlinearities commonly associated with conventional joints such as friction, stiction, and backlash. They do, however, provide a much smaller range of displacement compared with conventional joints. Hence, they are suitable for precision manipulation.

The range of deflection that a flexure joint can undergo depends on the shear modulus of the material and the design of the joint. This range of deflection provides a natural bound to the joint displacement variables, which then acts as a constraint in determining the achievable workspace of the end-effector through the interval analysis method. Figure 1 shows two of the most common types of flexure joints, depending on the shape of the cut to produce the elastically deformable section.

Due to the high precision nature of the applications, flexure-based mechanisms have a strict fabrication tolerance and therefore a high cost of fabrication. This is true for the larger flexure mechanisms for micro-/nanoprecision manipulation as well as for microscaled mechanisms. In this paper, we focus on the notch type joints, which (in an ideal case) produce a 1DOF revolute joint motion, without loss of generality in the algorithm presented for performance evaluation and the guarantee of constraint satisfaction. This type of flexure joints, as demonstrated in Ref. [6], exhibits a residual translational motion in addition to the ideal revolute displacement. This motion is complex to model accurately and it affects the accuracy of the forward and inverse kinematics of the mechanism. Other types of potential errors in the fabrication and assembly of a flexure jointed mechanism are outlined in Ref. [10].

3 Interval-Based Kinematics

The goal of the algorithm is to solve the kinematics of a given mechanism to obtain the range of end-effector workspace such that the required constraints are satisfied. In this paper, $\mathbf{x}$ is defined as a vector containing the task space variables $(x, y, \theta)^T$, $\mathbf{h}$ is a vector containing the design parameters of the mechanism (such as link lengths), and $C(\mathbf{x},\mathbf{h}) \leq 0$ is the mathematical inequality representing the constraints to be satisfied. The problem can therefore be formulated such that

$$\{\forall \mathbf{x} \in [\underline{\mathbf{x}}, \overline{\mathbf{x}}],\ \forall \mathbf{h} \in [\underline{\mathbf{h}}, \overline{\mathbf{h}}];\ \ C(\mathbf{x},\mathbf{h}) \leq 0\} \tag{1}$$

where $\underline{\mathbf{x}}, \overline{\mathbf{x}}$ and $\underline{\mathbf{h}}, \overline{\mathbf{h}}$ are the lower and upper bounds of the range of values in $\mathbf{x}$ and $\mathbf{h}$, respectively.

As an overview, the interval analysis method involves the following main components:

- interval extension or evaluation of functions
- testing against the constraints and obtaining the inner, outer, or boundary boxes
- filtering to enforce the consistency of various variables in the constraints
- the branch-and-bound loop within which the other components are carried out [11]

The goal of the strategy is to certify whether a particular range of workspace $[\underline{\mathbf{x}}, \overline{\mathbf{x}}]$ and design parameters $[\underline{\mathbf{h}}, \overline{\mathbf{h}}]$ yield either inner or outer boxes to constraint $C(\mathbf{x},\mathbf{h})$. If the range of the solution is too wide, it is often not possible to obtain a decision; in which case, filtering and branch-and-bound processes are utilized. The filtering process sharpens the result of constraint evaluation, while the branch-and-bound process splits the variables into smaller ranges and evaluates the constraints as a function of each subset of variables individually. The details of the components are presented in Secs. 3.1–3.5.

3.1 Interval Extension. The interval extension of variable x is defined as X, bounded within its lower and upper bounds $[\underline{x}, \overline{x}]$, where $\underline{x} \leq x \leq \overline{x}$. The width of the range is defined as $\overline{x} - \underline{x}$. The interval extension of a function is the evaluation of a function with interval variables. The two main types of function interval extension are *natural extension* [12] and *Taylor form extension* [13,14]. Natural extension is where real variables in a function are substituted by the equivalent interval variables. Hence, in this case

$$\forall x \in X, \quad f(x) \in F(X) \tag{2}$$

is the natural extension of $f(x)$. Taylor form extension utilizes the partial derivative of the function $f(x)$. Interval methods can be used conveniently to bound the remainder of the truncated Taylor series. In this paper, natural extension was utilized.

During the interval evaluation of a function, as numerical values are substituted into the function, the relationship between various variables is lost. Overestimation occurs when the same variables appear more than once within the function, and they are regarded as independent variables. The evaluation of a function where all variables involved appear only once is *sharp* (within rounding errors), meaning it is bounded within the smallest possible "box." For example, let $X=[1,2]$ and $Y=[3,6]$. Evaluating $F(X,Y)=X+Y=[4,8]$ would therefore be sharp. However, evaluating $G(X,Y)=X-X=[1,2]-[1,2]$ results in $[-1,1]$ and is not equal to zero, although we know it should. This is because the two

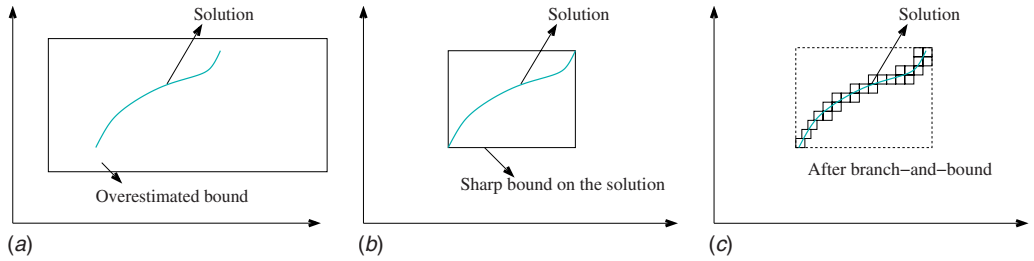

Fig. 2 Illustration of the effect of the branch-and-bound on an equality constraint: (*a*) The original (overestimated) bound of the solution, obtained by interval evaluation. (*b*) The sharp result with filtering. (*c*) Branch-and-bound process repeatedly bisects the solution box to a predefined threshold box dimension ϵ to provide a better bound to the solution.

variables, X, are taken as independent and not as the same variable. The effect of overestimation can be seen further in later subsections.

3.2 Types of Solutions. After the evaluation, it is necessary to test whether a required constraint is satisfied in the system. In our problem, it is desired to verify whether the required performance constraint $C(\mathbf{X},\mathbf{H})$ of the mechanism is true for the set of given *workspace pose (variables)* of interest ($\mathbf{x} \in \mathbf{X}$) and *mechanism parameters (link lengths)* ($\mathbf{h} \in \mathbf{H}$), as presented in Eq. (1). The following are the two types of constraints to be evaluated.

- *Inequality constraint*. When this is the case, the evaluated function $C(\mathbf{x},\mathbf{h})$ is said to satisfy the required constraints when

$$\forall \mathbf{x} \in \mathbf{X}, \quad \forall \mathbf{h} \in \mathbf{H}; \quad \underline{C}_R \leq C(\mathbf{X},\mathbf{H}) \leq \bar{C}_R \tag{3}$$

  where $\underline{C}_R$ and $\bar{C}_R$ are the lower and upper bounds of the requirements for the constraints. This is also termed the *inner solution* or the *inner box* of the constraint. The *outer box* is obtained when $\forall \mathbf{x} \in \mathbf{X}$, $\forall \mathbf{h} \in \mathbf{H}$; $(C(\mathbf{X},\mathbf{H}) \leq \underline{C}_R)$ or $(\bar{C}_R \leq C(\mathbf{X},\mathbf{H}))$. A *boundary solution* or a *boundary box* is found when it cannot be decided whether $(\mathbf{x},\mathbf{h})$ is an inner or outer box of the constraint.
- *Equality constraint*. When it is desired to obtain the solution of

$$\forall \mathbf{x} \in \mathbf{X}, \quad \forall \mathbf{h} \in \mathbf{H}; \quad C(\mathbf{X},\mathbf{H}) = 0 \tag{4}$$

then it is a comparatively easier task to obtain the outer box of the constraint. The outer box is obtained by solving for $\forall \mathbf{x} \in \mathbf{X}$, $\forall \mathbf{h} \in \mathbf{H}$; $0 \notin C(\mathbf{X},\mathbf{H})$. When $0 \in C(\mathbf{X},\mathbf{H})$, then it is only possible to deduce that a solution exists within $(\mathbf{X},\mathbf{H})$; however, it is not enough to define inner boxes.

3.3 Filtering. The filtering process enforces the consistency in the variables involved in the evaluation of an interval function/constraint. These techniques originated from the constraint programming field of study. It involves removing the segments in the interval variables involved that do not hold within the constraints. In this paper, the filtering process is used to reduce the effect of overestimation on the interval extensions of functions.

Overestimation of an interval function makes it difficult to decide whether or not a set of interval variables satisfies the given constraints. Consistency filtering is therefore required to sharpen the resulting boxes. The process utilizes the additional information contained within the mathematical equations or the physical constraints. In the case of mathematical equations, different ways of expressing the same equations yield different bounds to the interval evaluation due to varying degrees of overestimation. This is utilized in the filtering process by ensuring the consistency of the solutions throughout the various ways of expressing the same equation. In the case of physical constraints, additional information obtained from the physical or mechanical properties of the system is utilized to obtain a consistent solution. In essence, the physical properties provide the additional constraints that help produce a better evaluation of the interval solutions. In this paper, for example, a parallel mechanism is constructed out of several articulation chains that connect the base platform to a common moving platform. Forward kinematics of each chain to the common end-effector, for example, provides additional constraints that can be used to reduce the effect of overestimation.

In solving for the consistency of an equation, the concept is to ensure that the interval extension of a function produces a solution that is consistent with the given constraint. For example, let $f(x,y)=x^2-xy+2y=0$ be the specified constraint and that the initial estimates of the interval extension of variables x and y be $X \in [3,9]$ and $Y \in [1,4]$, respectively. If the interval extension of function $f(x,y)$ is evaluated, we obtain $F(X,Y)=[9,81]-[3,36]+[2,8]=[-25,86]$. However, to ensure the consistency in the constraint, it is possible to rewrite the equality such that

$$\begin{gathered} X^2 = Y(X-2) = [1,28] \\ X = [-5.3,5.3] \end{gathered} \tag{5}$$

taking the intersection of Eq. (5) and the initial estimate of X to obtain $X \in [-5.3,5.3)] \cap [3,9] = [3,5.3]$. Similarly, this can be performed on variable Y with the improved estimate of X, where

$$\begin{gathered} Y = X^2/(X-2) = [9,28]/[1,3.3] \\ Y = [2.73,28] \end{gathered} \tag{6}$$

Taking the intersection of Eq. (6) with the initial estimate of Y yields a new estimate of $Y \in [2.73,4]$. Therefore, the first iteration of the filtering procedure is shown to have sharpened the bounds of variables X and Y from $X \in [3,9]$ and $Y \in [1,4]$ to $X \in [3,5.3]$ and $Y \in [2.73,4]$. This can be iterated until such time that the improvement in the sharpness of the bounds is no longer worth the computational effort.

The procedure described above is termed 2B consistency [15,16]. Other filtering techniques are available such as 3B and interval Newton [12,13,17–19].

3.4 Branch-and-Bound. It is often difficult to conclude whether a given box constitutes an inner or outer box when it is evaluated as a function of interval variables with a large width. While the filtering process contracts the box and attempts to obtain a sharp solution, it can only return the sharpest box that would contain the solution. Within the box, the solution often occupies only a portion of the bounded space. The branch-and-bound strategy [11] is therefore utilized to automate the solution search of the algorithm such that a better definition of the solution may be found (Fig. 2).

The branch-and-bound algorithm searches for the solution by evaluating a box and deciding whether it yields an inner, outer, or

boundary box. If it yields a boundary box, then the box is split (bisected) and each of the bisected boxes is iteratively evaluated following the same process. The bisection process is iteratively performed until an inner or outer box is found, or until a threshold dimension of the variable boxes ϵ is reached. Boxes that remain as boundary solutions at this point will form the boundary solutions of the system.

In this paper, to evaluate the performance of a mechanism within the specified workspace $\mathbf{X}$ with specific mechanism parameters (link lengths) $\mathbf{H}$, bisection is performed on the M dimensional interval box $\mathbf{X}$ where M is the number of task space pose (workspace) variables. In the case of our planar parallel mechanism, $M=3$ and is made up of 2 translation and 1 orientation DOFs. The bisection process is performed across all the pose variables in a different order, depending on the bisection algorithms. Several possible bisection algorithms are available [20]. The simpler approaches are the *round robin* (bisection of variables by turn) and *largest first* (bisection of the variables with the largest width first). Both the strategies above suffer from the simplistic approach. For example, when dealing with a robot pose consisting of position and orientation values, orientation is expressed in radians (bounded within $\pm\pi$), while position is expressed in unit lengths. In this example, in the case that the positional workspace is much larger than $\pm\pi$ in numerical value, the *largest first* algorithm tends to bisect heavily on the position variables, neglecting the orientation variables. A possible solution is to normalize all variables with respect to the initial range of the interval variable. A more intelligent bisection algorithm was proposed [21] by utilizing the derivative of the system. This algorithm defines the *smear* function of the system and aims to identify the dominant variable that affects the system the most at that iteration. The most dominant variable is then selected for bisection. This approach is effective as it attacks the problem where bisection would yield the most effect. However, it has a drawback that it may continuously bisect specific dominant variables until ϵ is reached. This is amended by defining a minimum allowable threshold in the ratio of the width of a variable to its original width, beyond which the bisection turn is handed over to the next dominant variable.

3.5 Summary of Algorithm. The proposed algorithm is summarized in Table 1. Constraint function $C(\mathbf{X},\mathbf{H})$ can contain a single design constraint to be evaluated, expressed as mathematical equalities or inequalities, or a list of constraints. In the case where there are multiple constraints to be evaluated, an inner box is obtained only if $\mathbf{X}_i$ satisfies all the constraints (forms an inner box to all constraints) and an outer box is obtained if $\mathbf{X}_i$ fails to satisfy any of the constraints. The resulting solutions are contained in the following lists (see Table 1): List $\mathcal{L}_{\text{in}}$ contains the inner solutions that satisfy all the constraints, while $\mathcal{L}_{\text{out}}$ contains outer boxes, i.e., when a box fails to satisfy any one of the constraints in $\mathcal{C}$. The boundary boxes are given in list $\mathcal{L}_B$.

The variable $\mathbf{X}$ contains the end-effector workspace description, i.e., position (x,y) and orientation (θ) for the planar case considered in this paper (3DOF planar mechanism). However, a 6DOF mechanism will have the bisection process performed across the six-dimensional box describing the workspace (as can be deduced from the algorithm in Table 1). This would increase the computational cost of the algorithm. It should be noted, however, that the complexity in formulating and using the algorithm does not change a great deal. This is one of the advantages of the proposed algorithm, i.e., its versatility, often as a trade-off to computational cost, when compared with a more problem specific algorithm. As long as constraints are expressed mathematically, most of the work will be carried out by the numerical computation. In contrast to other methodologies, such as the geometric approach, as mentioned in Sec. 1, where the mathematical derivation is very much mechanism specific and a 6DOF mechanism will be much harder to solve than a 3DOF mechanism.

Table 1 Summary of algorithm for workspace constraint analysis with interval analysis

```
1   Initialize empty lists L_in, L_out, and L_B.
2   Initialize list L containing initial task space intervals
    (boxes) to be analyzed.
3   While (L not empty)
    (a) Extract manipulator pose X_i from list L.
    (b) Evaluate constraints C(X,H)
    (c) Test C(X,H) against the required performance
    [C_R, C̄_R].
    (d) Filtering process is carried out if necessary.
    (e) Return whether X constitutes an inner, outer, or
    boundary box.
    (f) Case result is inner, outer, or boundary box:
        (a) Case 1: The solution lies within the ALL
            constraints C(X,H)
            Remove X_i from L and add to list L_in
        (b) Case 2: The solution lies outside ANY of the
            constraints in C(X,H)
            Remove X_i from L and add to list L_out
        (c) Case 3: If X_i is a boundary solution
            If (dimension of box X_i) > ε)
                Bisect X_i into X_i1 and X_i2
                Remove X_i from L and add X_i1 and X_i2 into
                the list L.
            Else If (threshold dimension ε has been reached)
                Remove X_i from L and add to list L_B
            End If
    (g) End Case
4   End While
```

4 Interval Analysis on 3RRR Planar Parallel Flexure Mechanism

In this section, the workspace verification problem of a 3RRR planar flexure mechanism with respect to the various constraints relevant to the functionality of a precision manipulator is solved using the interval-based techniques presented in Sec. 3. The problem is to evaluate the performance of the manipulator, with respect to design constraints $C(\mathbf{X},\mathbf{H})$, defined by the design parameters $\mathbf{H}$ at the required workspace pose $\mathbf{X}$ and to certify whether $\mathbf{X}$ is a solution to the performance constraint.

The performance criteria of the planar manipulator that are presented to help illustrate the algorithm in this paper are (1) the amount of workspace reachable by the allowable deflection of the flexure joints, (2) singularity-free workspace, and (3) the workspace that yields the required motion resolution given the resolution of the joint space. Within the algorithm, uncertainties in the fabrication tolerance and the unmodeled kinematics of the flexure joints are taken into account in obtaining the solution. The 3RRR planar parallel mechanism, with the definitions of the variables and frame assignments, is given in Fig. 3.

In this paper, it is assumed that the positions of the revolute joints on the base $(\mathcal{O}_1,\mathcal{O}_2,\mathcal{O}_3)$ and the moving platforms (B_1,B_2,B_3) form equilateral triangles. These assumptions do not affect the generality of the analysis and were made so that some of the equations could be arranged in a simpler manner for clearer presentation.

The planar workspace of the manipulator is defined as $\mathbf{X}={}^{\mathcal{O}}\mathbf{p}_e=(x_e,y_e,\theta)^T$, which comprises the position and the orientation of the end-effector, respectively (see Fig. 3). For simplicity, the task space variable is always expressed with respect to the base frame $\mathcal{O}$; therefore, reference to the frame is omitted in the presentation in this paper, i.e., the task space variable will be written simply as $\mathbf{p}_e=[x_e,y_e,\theta_e]$. Joint space variables are the $(\alpha_i,\beta_i,\gamma_i)$, where $i=1,2,3$ represents each of the three serial chains connecting the base and moving platforms. It is assumed that only joints α_1, α_2, and α_3 are actuated ($\underline{R}RR$ chains) and

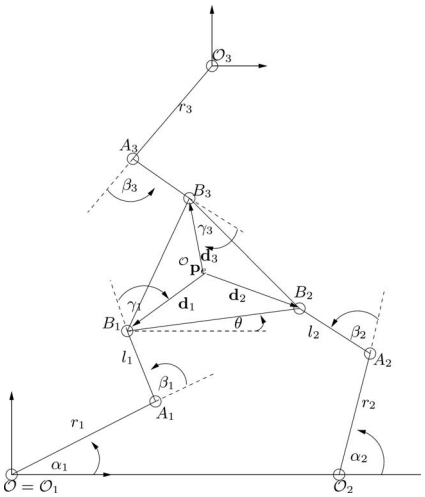

Fig. 3 A 3RRR planar parallel mechanism

displacement sensor feedbacks are only available on these joints. This is used as the case study to illustrate the algorithm, although it is possible in precision mechanisms to install displacement sensors to measure motion directly at the end-effector. Several planar parallel micropositioning mechanisms in literature are designed in this topology [6,22].

The position vectors of points B_1, B_2, and B_3 with respect to base points $\mathcal{O}_1$, $\mathcal{O}_2$, and $\mathcal{O}_3$ are defined as $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$, respectively (where $\mathbf{p}_i=(x_i,y_i)^T$). These points move with the end-effector; hence, their positions are given by the position of the end-effector (x_e,y_e), plus the displacement caused by the orientation of the end-effector θ due to the offset distance $\mathbf{d}=[\mathbf{d}_1,\mathbf{d}_2,\mathbf{d}_3]^T$. With respect to the global origin $\mathcal{O}$, these vectors are defined as ${}^{\mathcal{O}}\mathbf{p}_1$, ${}^{\mathcal{O}}\mathbf{p}_2$, and ${}^{\mathcal{O}}\mathbf{p}_3$.

The position vectors $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ are obtained through

$$\mathbf{p}_i = {}^{\mathcal{O}}\mathbf{p}_i - \mathcal{O}_i = \mathbf{p}_e + \mathbf{R}_Z(\theta + \angle\,\mathbf{d}_i)\cdot\begin{bmatrix}0\\0\\d\end{bmatrix} - \mathcal{O}_i \tag{7}$$

where d is the length of vectors $\mathbf{d}_1$, $\mathbf{d}_2$, and $\mathbf{d}_3$, which are assumed to be of equal length, $(\angle\mathbf{d}_1,\angle\mathbf{d}_2,\angle\mathbf{d}_3)=(7(\pi/6), -(\pi/6),(\pi/2))$ are the headings of vectors $\mathbf{d}_1$, $\mathbf{d}_2$, and $\mathbf{d}_3$ when the flexure joints are at rest.

The manipulator design is used as an example in Secs. 4.1–4.4 to better illustrate the algorithm proposed. The various link lengths of the selected 3RRR planar parallel mechanism are summarized in Table 2. The pose of the end-effector when the 3RRR mechanism is symmetrical is defined as $(\mathbf{p}_{em})=(x_{em},y_{em},\theta_{em})^T$, which is also the pose when the flexure joints are at rest, i.e., when they undergo zero deflections. The algorithm was implemented in C++ with the ALIAS library developed on the BIAS/PROFIL platform within the COPRIN project [23].

Table 2 Parameters of the case study 3RRR planar parallel flexure-based mechanism

Parameters	Values
Origin of $\mathcal{O}_1$	$(0,0)^T$ mm
Origin of $\mathcal{O}_2$	$(167.27,0)^T$ mm
Origin of $\mathcal{O}_3$	$(83.64,144.86)^T$ mm
$\lvert\mathbf{d}_i\rvert$	10 mm
r_i	66 mm
l_i	46 mm
$(x_{em},y_{em},\theta_{em})^T$	$(83.64,48.29,-10.3\ \text{deg})^T$

4.1 Reachable Workspace by Joint Limits. It is imperative that the joint displacement in a flexure joint takes place only within the linear deformation region of the member. This depends primarily on the shear modulus of the material and the design of the joint. After taking into account the safety margin, the selected values of maximum allowable joint deflection used in the interval analysis of the flexure mechanism workspace should constitute a bound for which the joints behave linearly within the elastic region.

To obtain the joint displacement of the mechanism for a given end-effector pose, inverse kinematics is carried out on the desired end-effector workspace ($\mathbf{X}$), with the given interval link length parameters ($\mathbf{H}$). The inverse kinematics of planar parallel mechanisms is often discussed in literature [24,25]. Generally, the inverse kinematic solution can be obtained by first calculating the angle β_i, which has two possible solutions within $[0,2\pi]$. Choosing one of the two possible solutions for each β_i, the joint displacement of angle α_i can be obtained. Obtaining the inverse cosine or inverse sine of an interval variable does not uniquely define the solution angle as these trigonometric functions are periodic. To overcome this problem, the constraints on the allowable joint deflection are expressed as the limits on the sine and cosine of the joint limit angles. For a flexure jointed mechanism, however, the unique solution to β_i can be predetermined (whether it is the elbow up or elbow down solution) due to the limited motion of the mechanism. The closed-form solution of the inverse kinematics is therefore given as follows:

$$\cos(\beta_i) = \frac{x_i^2 + y_i^2 - (r_i^2 + l_i^2)}{2l_i r_i}$$

$$\cos(\alpha_i) = \frac{x_i(r_i + l_i\cos(\beta_i)) + y_i l_i \sin(\beta_i)}{x_i^2 + y_i^2} \tag{8}$$

$$\sin(\alpha_i) = \frac{-x_i l_i \sin(\beta_i) + y_i(r_i + l_i\cos(\beta_i))}{x_i^2 + y_i^2}$$

Although angles γ_i are not usually considered theoretically in the inverse kinematics of a 3RRR planar parallel mechanism, it is important in practical cases to take the limits of these joints into account—such as to avoid collisions among the links of the mechanism. In the case of flexure jointed mechanisms, it is also necessary to impose a joint limit constraint on these joints, as they are also flexure joints. From Fig. 3, it can be observed that

$$\alpha_i + \beta_i + \gamma_i = \angle\,(-\mathbf{d}_i) \tag{9}$$

4.1.1 Constraint Definition. Interval extension of end-effector workspace variables $\mathbf{p}_e=(x_e,y_e,\theta_e)^T$ were utilized to describe the desired range of the workspace $\mathbf{X}$. Interval variables for the link lengths ($\mathbf{H}$), however, were used to express the fabrication tolerances and other unmodeled sources of errors. With these variables defined, the constraint for the workspace as defined by the allowable joint deflections can be defined by

$$\begin{aligned}
C_1(\mathbf{X},\mathbf{H}) &= \cos(\beta_i) \in [\underline{\cos(\beta)},\overline{\cos(\beta)}]\\
C_2(\mathbf{X},\mathbf{H}) &= \cos(\alpha_i) \in [\underline{\cos(\alpha)},\overline{\cos(\alpha)}]\\
C_3(\mathbf{X},\mathbf{H}) &= \sin(\alpha_i) \in [\underline{\sin(\alpha)},\overline{\sin(\alpha)}]\\
C_4(\mathbf{X},\mathbf{H}) &= \cos(\gamma_i) \in [\underline{\cos(\gamma)},\overline{\cos(\gamma)}]\\
C_5(\mathbf{X},\mathbf{H}) &= \sin(\gamma_i) \in [\underline{\sin(\gamma)},\overline{\sin(\gamma)}]
\end{aligned} \tag{10}$$

A box of solution $\mathbf{X}_i$ is an inner solution when all of the constraints (Eq. (10)) are satisfied. The workspace described by the inner box is reachable by the end-effector of the mechanism,

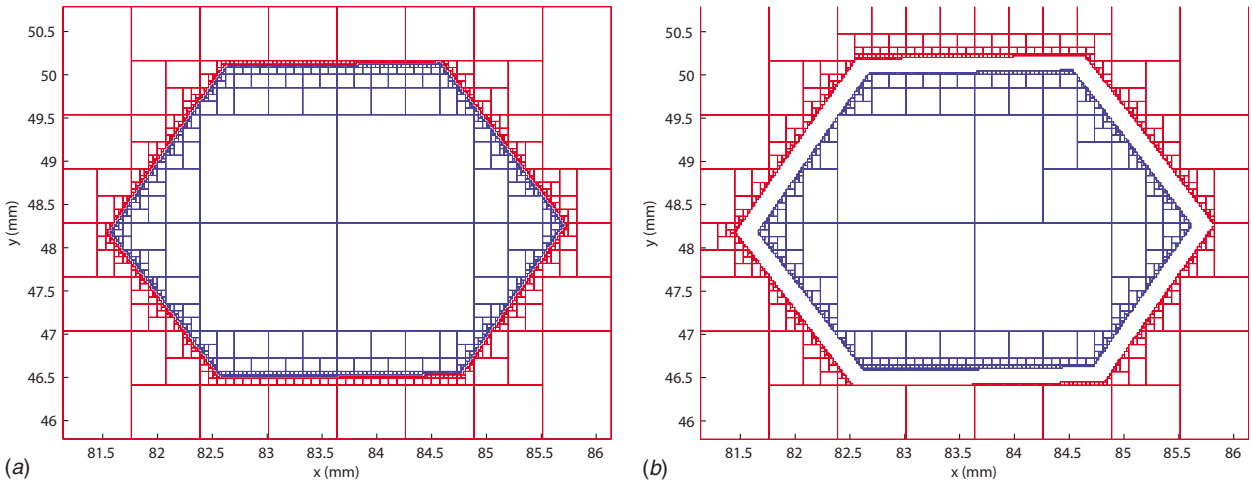

Fig. 4 Workspace of the 3RRR planar parallel mechanism. The workspace, constrained by joint limits, after consistency filtering: (*a*) without considering fabrication tolerances and (*b*) assuming ±50 μm tolerance on link length *r* and *l*. These two dimensional plots are generated at constant $\theta=\theta_m=-10.3$ deg. Allowable flexure-joint deflection is ±3 deg.

given the allowable flexure-joint deflection. A box $\mathbf{X}_i$ is an outer box if any of the interval inverse kinematic solutions falls within the complement of Eq. (10).

4.1.2 Results and Discussion: Workspace by Flexure Joint Limit. In this example, the allowable joint deflection was set at ± 3 deg. The workspace to be evaluated is selected as a motion range of ± 2.5 mm in the x and y directions ($x_r=y_r=2.5$ mm) and ± 17.5 mrad ($\theta_r=1$ deg) for orientation about $(x_{em}, y_{em}, \theta_{em})^T$, respectively.

Due to the overestimation, direct interval evaluation of the constraints yielded no solutions in the inner boxes. Consistency filtering is required to sharpen the evaluation of the inverse kinematic solution. The multiple chains i, which connect to the common moving platform, was utilized as an additional constraint to reduce the amount of uncertainty involved in the interval calculation of the inverse kinematic problem. The following set of physical constraints, essentially forming the direct kinematics of vector $\mathbf{p}_i$, were utilized as additional constraints in enforcing the consistency of the following inverse kinematic solutions:

$$C_1(\mathbf{X},\mathbf{H}) = R_i \cos(\alpha_i) + L_i \cos(\alpha_i + \beta_i) - X_i = 0$$

$$C_2(\mathbf{X},\mathbf{H}) = R_i \sin(\alpha_i) + L_i \sin(\alpha_i + \beta_i) - Y_i = 0 \qquad (11)$$

$$C_3(\mathbf{X},\mathbf{H}) = (\cos(\alpha_i))^2 + (\sin(\alpha_i))^2 - 1 = 0$$

A 2B/3B consistency filtering was implemented and the improvement in the inner solution is shown in Fig. 4(*a*). This two-dimensional figure displays the inner and outer boxes of the manipulator workspace when the flexure-joint limits are imposed after consistency filtering. It is taken at constant $\theta=\theta_m=-10.3$ deg. The effectiveness of the filtering technique can be clearly seen in the ability of the algorithm in admitting inner solutions.

The result in Fig. 4(*a*) was produced by assuming zero uncertainties in the fabrication tolerances and no uncertainties in the flexure-joint modeling. In this case, interval variables R_i and L_i were defined as degenerate intervals, such that they have the same values of the lower and upper bounds (zero interval width). As explained in Sec. 2 and presented in Ref. [6], the unmodeled kinematics of a notch type revolute flexure joint manifests itself in an additional amount of translational motion. This can be modeled as additional uncertainties in the link lengths L_i and R_i. This unmodeled degree of freedom is complex to model and comparatively small in magnitude. In our technique, the bounds of the error estimate are used to account for the additional translation in the absence of a complex and accurate model. Additional uncertainties due to fabrication tolerances are also added to the interval variables L_i and R_i. To include such uncertainties into the evaluation process, these bounds of uncertainties are added to the link lengths L_i and R_i.

In our example, the additional uncertainties due to unmodeled kinematics and the flexure fabrication tolerance are defined as being bounded within ± 50 μm for each link length r_i and l_i. The resulting workspace within the limits of allowable flexure-joint deflection is shown in Fig. 4(*b*) for comparison. As expected, there is a larger area of boundary solutions compared with when fabrication tolerances and modeling errors were not considered. However, the inner solutions exist such that the represented workspace range is certified to be within the required constraints, with the fabrication tolerances and the kinematics model uncertainties are taken into account. It is therefore demonstrated that various uncertainties, including fabrication limitations, can be included in the calculation during the design process to guarantee that the performance of the resulting mechanism is within the specified requirements.

Figure 5 demonstrates the workspace of the mechanism bounded by the allowable joint deflection for a range of orientation Θ, represented by the vertical axis of the plot. The range of interval Θ is ± 17.5 mrad. The workspace is represented in solid and wire frames plots for clarity.

4.2 Singularity. It is also desired to evaluate the workspace of the mechanism to certify that the operational region is free of singularity. The constraint, in this case, is the function defining the loci of singularity. The singularity-free region is the end-effector workspace that can be guaranteed to contain no solution to the constraint. For simple mechanisms, it is possible to obtain the symbolic expression of the determinant of the Jacobian matrices. However, obtaining a symbolic expression of singularity for more complex mechanisms with higher degrees of freedom may not be practical. An efficient method was proposed in Ref. [26] to evaluate the regularity of the interval Jacobian matrix numerically. This method is used in this paper to obtain the nonsingular workspace of the mechanism.

The differential kinematic relationships can be obtained from Ref. [24] as

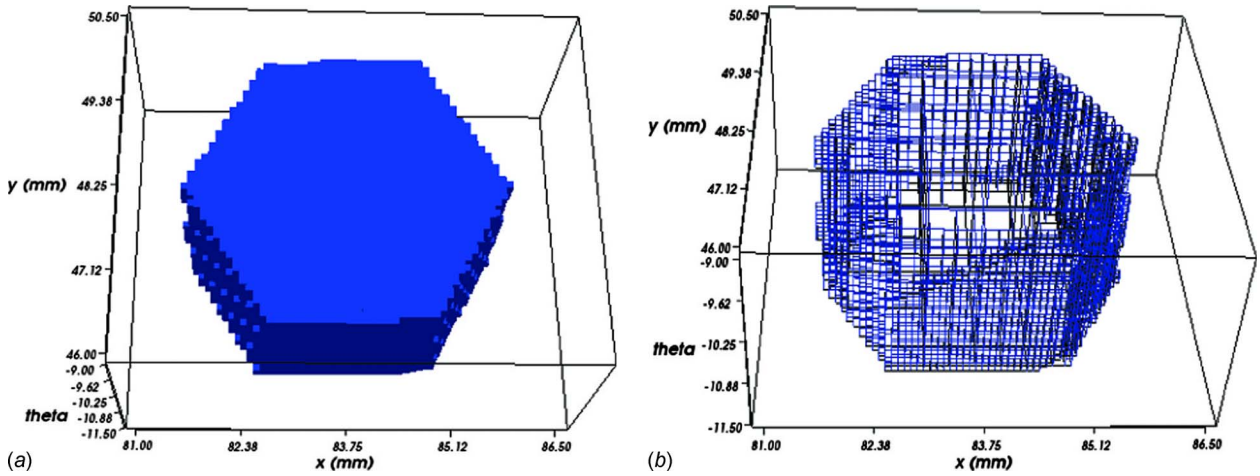

Fig. 5 The inner solution admitted into the workspace of the 3RRR planar parallel flexure mechanism. This result takes into account the uncertainties in the kinematic modeling and fabrication tolerances, constrained by the bounds of the allowable flexure-joint deflections. The orientation range is $\theta^{\mathcal{I}}=\theta_m\pm17.5$ mrad. The workspace is presented in solid (*a*) and wire frames (*b*).

$$[\mathbf{f}_i^T,\mathbf{f}_i^T\mathbf{d}_i^\perp]\begin{bmatrix}\dot{\mathbf{p}}_e\\ \omega\end{bmatrix}=r_i\mathbf{f}_i^T\begin{bmatrix}-\sin(\alpha_i)\\ \cos(\alpha_i)\end{bmatrix}\dot{\boldsymbol{\alpha}}_i \tag{12}$$

where $\mathbf{f}_i^T$ is the unit vector in the direction of the reciprocal screws passing through the revolute joints at points A and B

$$\mathbf{f}_i^T=\frac{1}{l_i}\begin{bmatrix}x_i-r_i\cos(\alpha_i)\\ y_i-r_i\sin(\alpha_i)\end{bmatrix} \tag{13}$$

$\mathbf{d}_i^\perp$ is the vector perpendicular to $\mathbf{d}_i$, or $\mathbf{d}_i^\perp=(-d_{iy},d_{ix})^T$, and $\dot{\boldsymbol{\alpha}}$ is the vector containing the rate of the actuated joints $(\dot{\alpha}_1,\dot{\alpha}_2,\dot{\alpha}_3)$, as defined in Fig. 3. The overall differential kinematics of the mechanism can be described by

$$\mathbf{J}_1\cdot\dot{\mathbf{x}}_e=\mathbf{J}_2\cdot\dot{\boldsymbol{\alpha}} \tag{14}$$

where $\dot{\mathbf{x}}_e=(\dot{\mathbf{p}}_e^T,\omega)^T=(\dot{x}_e,\dot{y}_e,\dot{\theta})^T$, and $\mathbf{J}_1$ and $\mathbf{J}_2$ are the 3×3 Jacobian matrices, with each row representing the relationship (Eq. (12)) for individual leg i. Note that $\mathbf{J}_2$ is a diagonal matrix.

4.2.1 Constraint Definition. The constraint for singular workspace of the mechanism is defined as

$$\begin{aligned}C_1(\mathbf{X},\mathbf{H})&=\det(\mathbf{J})_1=0\\ C_2(\mathbf{X},\mathbf{H})&=\det(\mathbf{J})_2=0\end{aligned} \tag{15}$$

It is possible to guarantee that a specific box in the workspace does not contain singularity by solving for the workspace such that

$$\{\forall\mathbf{x}\in\mathbf{X},\forall\mathbf{h}\in\mathbf{H},(0\notin C_1(\mathbf{X},\mathbf{H}))\wedge(0\notin C_2(\mathbf{X},\mathbf{H}))\} \tag{16}$$

4.2.2 Results and Discussion: Singularity-Free Workspace. Singularities associated with a rank deficient $\mathbf{J}_1$ and $\mathbf{J}_2$ are the internal and the boundary singularities of the mechanism, respectively. To demonstrate the interesting features in this evaluation, the singularity analysis is performed considering the entire workspace the 3RRR mechanism, independent of the joint displacement limits. The resulting loci of singular-free configurations, obtained by the direct evaluation of constraint (16), are shown in Fig. 6 with an orientation range of [30 deg,50 deg]. The result shows a large region of the workspace that is occupied by boundary solutions.

For the clarity of further analysis, the singularity-free workspace for constant orientation is presented as two-dimensional plots in Fig. 7(*a*), taken at $\theta=40$ deg. It can be seen from Figs. 6 and 7(*a*) that in this example, direct evaluation does not produce a very sharp result even after the consistency filtering process, leaving a large region of boundary solutions. As highlighted earlier, it is difficult to decide on the solutions of the equality constraint. To improve the sharpness of the solution, an advanced numerical regularity test that has been implemented on the ALIAS library [23] was then utilized to enhance the performance of the algorithm. This regularity test utilizes the following components.

Singularity identification algorithm component 1. The Rohn consistency test is used to determine the interval matrices that are regular. The Rohn consistency test states that for an interval matrix $\mathcal{I}$, if a well defined set of scalar matrices derived from $\mathcal{I}$ have determinants of the same sign, then there is no singular matrix in $\mathcal{I}$. This is a powerful test to certify the outer box of the equality, i.e., to certify that a box of workspace does not contain any singularity, i.e., $0\notin C(\mathbf{X},\mathbf{H})$.

Singularity identification algorithm component 2. A matrix regularity test through the sign of the matrix determinant (as proposed in Ref. [26]) is used to certify that a matrix contains singularity. This is carried out by sampling points within a given box and comparing the signs of the determinants for these points. If there exists any point in the interval box that displays a different sign of determinant from other points, then singularity exists within the interval box. This provides a strong tool to certify the existence of a singularity within an interval matrix.

Combined, the two techniques provide an effective tool to evaluate the singularity of mechanisms.

Another point to note is if this test concludes that a solution to the equality exists in the interval box $(\mathbf{X},\mathbf{H})$, it does not mean the entire box is singular (as singularity is a point). Therefore, the interval box should not be immediately assigned as singular but as a boundary solution to be bisected further to localize the singularity loci. It is therefore possible to narrow down the boundary boxes to the size of ϵ (the threshold of the smallest dimension of workspace region where the bisection process is terminated). This method identifies singularity numerically down to the threshold dimension ϵ. The large improvement in the sharpness of the solution provided by this approach over the direct evaluation method is shown in Fig. 7(*b*), where the loci of singular configurations are marked with a solid line for a clearer view.

It should be noted that evaluating the singularity-free constraint over the end-effector workspace within the allowable joint displacement limits (Sec. 4.1) yields no singularity.

Referring to Fig. 7, the center portion of the workspace is where the determinant of the Jacobian matrix is negative, while the three portions along the edges are of positive determinant. It is

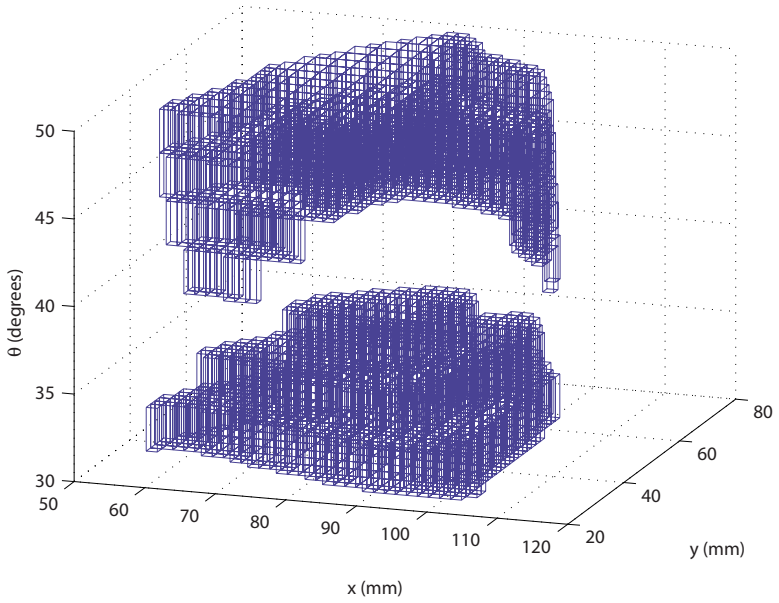

Fig. 6 Singularity free workspace as obtained by evaluation of constraint (16)

important that the operational workspace and the motion path planning of the end-effector do not cross between positive and negative determinant regions. This demonstrates the ease with which interval analysis techniques can be adapted to verify a singularity-free path planning problem.

It might also be of interest to note that an optional technique exists to trace the loci of singularity once a singular point is found in the workspace. Once a singularity point is found, the continuation method [27,28] can be employed to trace and identify the loci of singular configurations. This may provide a faster solving algorithm.

4.3 Task Space Motion Resolution. Another important characteristic in the flexure mechanism is the task space motion resolution. As these mechanisms are generally employed for precision manipulation, the resolution of the smallest step possible in the motion of the end-effector is often an important performance criterion. Generally, it is possible to directly establish the bounds of joint space motion resolutions from the specifications of the sensors and actuators used. Incremental step size in the task space can therefore be calculated by the differential kinematic relationship with the incremental step size in joint space displacement

$$\mathbf{J}_1 \cdot \Delta \mathbf{x}_e = \mathbf{J}_2 \cdot \Delta \mathbf{q} \tag{17}$$

where $\Delta \mathbf{q}$ is the incremental step in joint space. Since a 3RRR mechanism is considered in our case, then only the three base joints are actuated. It can be assumed that only these active joints are equipped with displacement sensors. Hence, $\Delta \mathbf{q} = (\delta\alpha_1, \delta\alpha_2, \delta\alpha_3)^T$. It is then required to solve for $\Delta \mathbf{x}_e$ in linear equation (17). This is done in this paper using the Gaussian method [18]. Several interval arithmetic packages have Gaussian solving functions ready, hence users do not need to code this function from scratch.

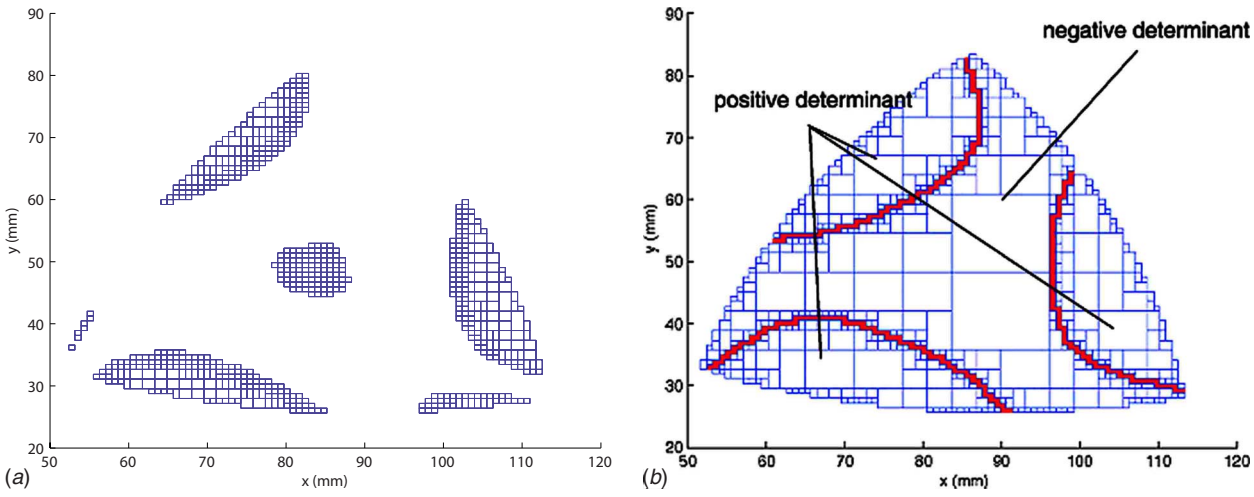

Fig. 7 Singularity free workspace of the 3RRR planar parallel mechanism, taken at θ=40 deg. The result (*a*) was obtained by direct evaluation of constraint (16) and is the 2D representation of the result in Fig. 6 at θ=40 deg. The result (*b*) shows the large improvement provided by the matrix regularity test algorithm, as provided by the ALIAS library. The loci of singular workspace are marked in red (solid color) for clarity.

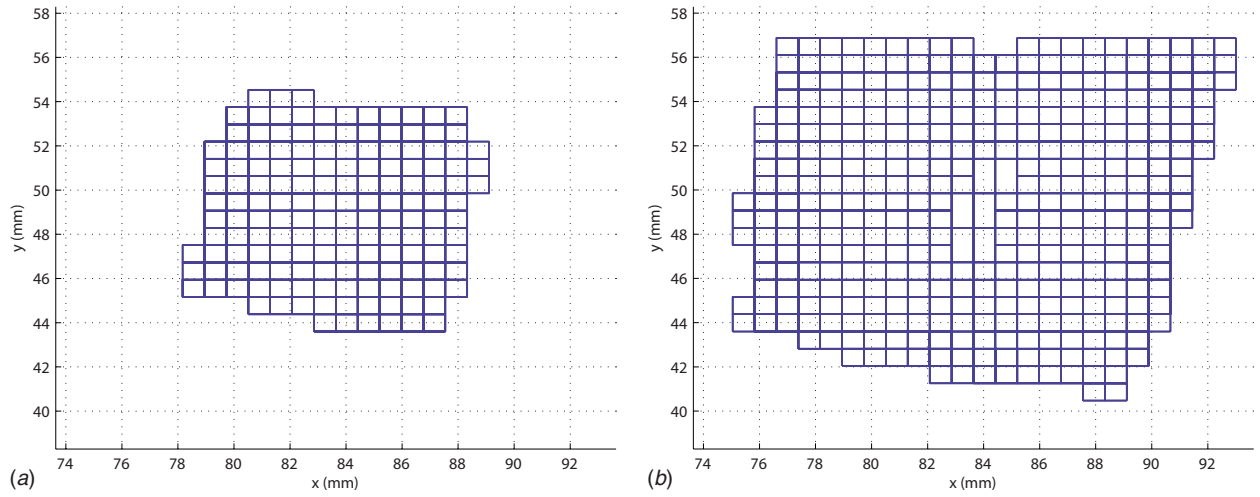

Fig. 8 Two dimensional workspace at constant $\theta=\theta_m=-10.3$ deg that satisfies the required motion requirement, given the joint space motion resolution. Solving algorithms were (*a*) preconditioned Hansen–Bliek and (*b*) symbolic preconditioning with Gaussian elimination.

4.3.1 Constraint Definition. The constraint to be satisfied is therefore defined as

$$|\Delta \mathbf{X}_e| \leq \Delta \mathbf{X}_{\max} \tag{18}$$

where $\Delta \mathbf{X}_e$ is the interval extension of vector $\mathbf{x}_e$ as defined in Eq. (17) and $\Delta \mathbf{X}_{\max}$ is the largest end-effector motion resolution required for the task. Inner boxes are obtained when the specified workspace satisfies Eq. (18) and outer boxes when $|\Delta \mathbf{X}_e| > \Delta \mathbf{X}_{\max}$.

For this example, it is given that the joint space displacement resolution is bounded within 0.06 mrad. The desired resolution ($\Delta \mathbf{X}_{\max}$) for the task space translational motion and orientation are set at 0.5 μm and 0.5 mrad, respectively.

4.3.2 Results and Discussion: Task Space Resolution. When a Gaussian elimination technique was utilized, the algorithm yielded no inner solution to the constraints. A preconditioning process [13] was performed to improve the sharpness of the solution. Preconditioning was applied by premultiplying both sides of the equation with matrix $\mathbf{M}$, where

$$\mathbf{M} = (\text{mid}(\mathbf{J}_1))^{-1} \tag{19}$$

where $\text{mid}(\mathbf{J}_1)$ is the matrix containing the midvalues of the elements of the interval matrix $\mathbf{J}_1$.

When Eqs. (17) and (18) are evaluated for the workspace of our 3RRR planar parallel manipulator, the workspace that satisfies the required motion resolution is given in Fig. 8(*a*). The solution was obtained through a variation of the Gaussian technique, named the *Hansen–Bliek* solving algorithm [29–31], which is numerically preconditioned.

Further improvement can be obtained through *symbolic preconditioning*, as proposed in Ref. [26]. It has been shown to be effective in complex systems to improve the sharpness of the solutions. This approach is possible when symbolic expressions of the linear system of equations are given. The idea is to minimize symbolically the number of multiple occurrences of the variables in an interval function. This method is performed by evaluating matrix $\mathbf{M}$ but keeping $\mathbf{J}_1$ symbolic. Premultiplying the system with $\mathbf{M}$ and keeping $\mathbf{J}_1$ symbolic allows the elements of the resulting matrix to be rearranged symbolically to minimize the multiple occurrences of various interval variables, hence reducing the effect of dependency. Consistency filtering is also included in the algorithm to further sharpen the results. The improvement in the algorithm's ability to admit an inner solution is demonstrated in the amount of workspace that can be certified as the inner solution of the constraint dictated by the desired end-effector motion resolutions. This is shown in Fig. 8(*b*). The results in Fig. 8 were obtained with the exact same conditions, taken at constant $\theta=\theta_m$, with the only differences being the algorithms used for solving the linear equations: (a) the preconditioned Hansen–Bliek algorithm and (b) the symbolically preconditioned Gaussian elimination method.

4.4 Overall Available Workspace. The certified available workspace of the planar parallel mechanism can be evaluated by imposing all of the constraints that have been presented and discussed above. A desired interval of end-effector workspace can be tested against the set of constraints. For the desired workspace to satisfy all the performance criteria required of the manipulator, it is necessary that the procedure results in inner boxes to all the given constraints for the entire desired workspace. It is therefore desirable to be able to obtain a sharp solution to decide whether or not an interval in the workspace satisfies the design requirements. If a bounded solution cannot be obtained, then it cannot be guaranteed that all the design constraints are satisfied. It is then necessary to alter the design or to relax some of the requirements.

In the use of interval analysis, it is important to note that various constraints require different levels of computational resources. The computational load increases exponentially with every additional bisection in the algorithm. In implementing our algorithm, computationally cheap constraints were calculated first within each iteration, and whenever possible, were used to eliminate boxes that do not satisfy the constraint before calculating the more computationally expensive constraints. These can then be housed in a nested heuristic structure where the computationally most expensive constraints are calculated only when every other (computationally cheaper) constraint has failed to produce an inner or outer box. If an inner or outer box is not obtained even after filtering, then bisection is performed.

In the case of flexure mechanism, the joint limit is generally the largest contributor in reducing the available workspace, as flexure-joint workspace constitutes only a very small portion of the overall mechanism workspace. In our example, it can be seen that the workspace allowed by the joint deflection limit is situated well within the singularity-free region with a negative determinant. An example of the usable workspace that satisfies all the criteria (joint limit, singularity-free, and required motion resolution) is therefore given in Fig. 9. The figure shows the zoomed-in view of the workspace, located well within the singularity-free region at the center of the manipulator workspace, with nominal orientation at $\theta_m=-10.3$ deg. The two most strin-

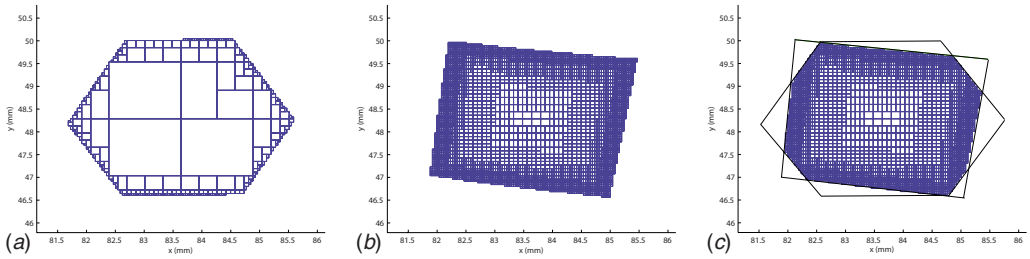

Fig. 9 Overall workspace due to multiple constraints. (*a*) The workspace allowable by limits of joint deflection. (*b*) The workspace that satisfies the required motion resolution. (*c*) The intersection of all given constraints.

gent constraints in this example were the joint limit and the end-effector motion resolution. The workspace allowed by the joint limit is reproduced in Fig. 9(*a*). The workspace that produces an end-effector motion resolution of 0.5 μm (translation) and 0.5 mrad (orientation), given a joint motion resolution ($\delta\alpha_i$) of 0.06 mrad, is given in Fig. 9(*b*), and the resulting intersection, which is the workspace that satisfies all the given criteria, is given in Fig. 9(*c*). The plot of usable workspace for an orientation range of $[\theta_m-\theta_r, \theta_m+\theta_r]$, for $\theta_m=-10.3$ deg and $\theta_r=1$ deg(=17.5 mrad), is shown in Fig. 10.

In verifying the design of a flexure mechanism, the workspace shown in Fig. 10 is guaranteed to satisfy all the given criteria. It can be deduced that an end-effector motion range of ± 1 mm in translation and ± 1 deg in orientation is possible within the given criteria. To test a desired workspace **X**, with a particular manipulator design and link length **H**, the interval variables of the workspace are substituted directly into the algorithm and are certified whether or not they form inner boxes of the constraints. This is the case when the entire desired workspace consists of only inner boxes for all the given constraints.

Implemented on Dual Core Intel 2.4 GHz processors (4 Mbyte cache) with 1 Gbyte RAM, the algorithm took 15 s to verify that the flexure mechanism workspace with a range of ± 1 mm in translation and ± 1 deg in orientation from its zero deflection point (x_m, y_m, θ_m) is entirely contained within the inner solution of all the constraints (represented by 8728 inner boxes). This demonstrates the efficiency of the algorithm. This level of efficiency also allows multiple iterations of the evaluation algorithm to be run as a constraint satisfaction and optimization algorithm for a mechanism design to determine a suitable range of values of the design parameter **H** that satisfies all the given constraints. This provides a strategy to automatically generate sets of design parameters of the flexure-based precision mechanism, where all of the requirements are guaranteed to be satisfied within the desired workspace. An optimization technique can then be further performed on these possible designs to select the best one based on a cost function. This is part of the future work in the development of the interval-based design strategy.

5 Conclusion

A technique to address workspace verification problem of a precision flexure-based mechanism is presented in this paper. The technique certifies whether or not the required workspace satisfies certain a set of performance criteria, taking into account the modeling and fabrication uncertainties. Performance features relevant to a flexure-based mechanism are presented and the efficient interval-based methods in evaluating and resolving the features

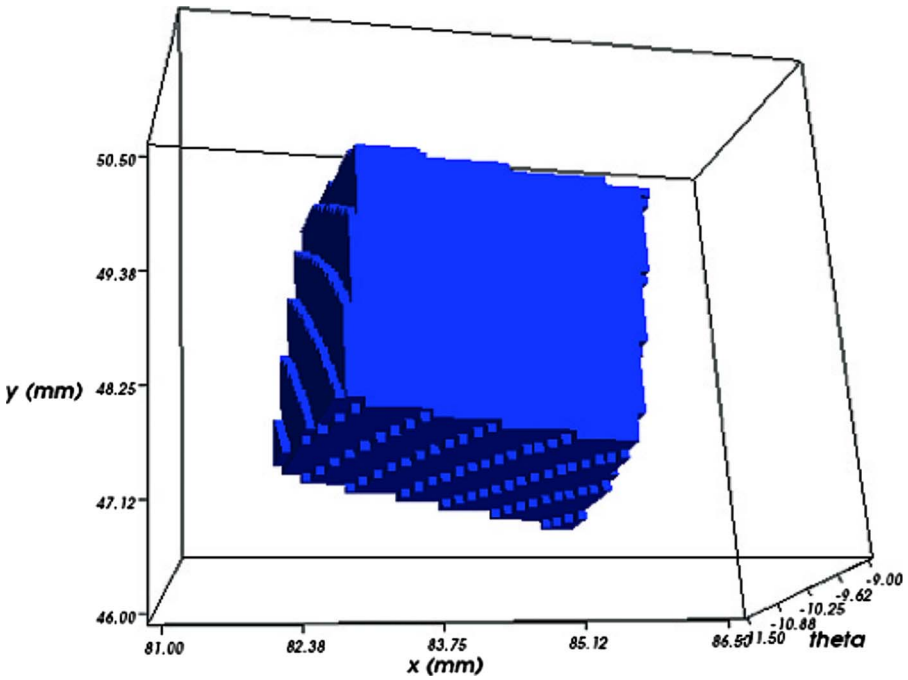

Fig. 10 Usable workspace of the planar flexure-based manipulator for orientation range of –10.3 deg±1 deg

were proposed, implemented, and discussed. Future work is aimed at extending the algorithm to a produce an efficient synthesis algorithm that would enable the determination of design parameters of a mechanism that satisfy a set of given constraints. This method would require not only the verification of constraint satisfaction for a set of nominal design parameters but also for a continuous range of possible design parameters.

Acknowledgment

This study is conducted under Project ARES (Assembly of Reconfigurable Endoluminal System), a NEST-Adventure European Union funded project (EU Contract No. 015653).

References

[1] Paros, J. M., and Weisbord, L., 1965, "How to Design Flexure Hinge," Mach. Des., **37**(27), pp. 151–156.
[2] Scire, F. E., and Teague, E. C., 1978, "Piezodriven 50-μm Range Stage With Subnanometer Resolution," Rev. Sci. Instrum., **49**(12), pp. 1735–1740.
[3] Choi, D., and Riviere, C., 2005, "Flexure-Based Manipulator for Active Handheld Microsurgical Instrument," *Proceedings of the IEEE Conference on Engineering in Medicine and Biology Society*, pp. 2325–2328.
[4] Gao, P., Swei, S.-M., and Yuan, Z., 1999, "A New Piezodriven Precision Micropositioning Stage Utilizing Flexure Hinges," Nanotechnology, **10**, pp. 394–398.
[5] Kim, D., Kang, D., Shim, J., Song, I., and Gweon, D., 2005, "Optimal Design of a Flexure Hinge-Based XYZ Atomic Force Microscopy Scanner for Minimizing Abbe Errors," Rev. Sci. Instrum., **76**(7), pp. 073706.1–073706.7.
[6] Yi, B.-J., Chung, G. B., Na, H. Y., Kim, W. K., and Suh, I. H., 2003, "Design and Experiment of a 3-DOF Parallel Micromechanism Utilizing Flexure Hinges," IEEE Trans. Rob. Autom., **19**(4), pp. 604–612.
[7] Merlet, J.-P., Gosselin, C., and Mouly, N., 1998, "Workspaces of Planar Parallel Manipulators," Mech. Mach. Theory, **33**(1–2), pp. 7–20.
[8] Pennock, G., and Kassner, D., 1993, "The Workspace of a General Geometry Planar Three Degree of Freedom Platform Manipulator," ASME J. Mech. Des., **115**(2), pp. 269–276.
[9] Kumar, V., 1992, "Characterization of Workspaces of Parallel Manipulators," ASME J. Mech. Des., **114**(3), pp. 368–375.
[10] Niaritsiry, T.-F., Fazenda, N., and Clavel, R., 2004, "Study of the Sources of Inaccuracy of a 3 DOF Flexure Hinge-Based Parallel Manipulator," Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 4, pp. 4091–4096.
[11] Land, A., and Doig, A., 1960, "An Automatic Method of Solving Discrete Programming Problems," Econometrica, **28**(3), pp. 497–520.
[12] Moore, R., 1966, *Interval Analysis*, Prentice-Hall, Englewood Cliffs, NJ.
[13] Hansen, E., and Walster, G., 2004, *Global Optimization Using Interval Analysis*, 2nd ed., Dekker, New York.
[14] Berz, M., and Hoffstätter, G., 1998, "Computation and Application of Taylor Polynomials With Interval Remainder Bounds," Reliable Computing, **4**, pp. 83–97.
[15] Benhamou, F., Goualard, F., and Granvilliers, L., 1999, "Revising Hull and Box Consistency," *Proceedings of the. International Conference on Logic Programming*, Las Cruces, NM, pp. 230–244.
[16] Collavizza, M., Delobe, F., and Rueher, M., 1999, "Comparing Partial Consistencies," Reliable Computing, **5**, pp. 213–228.
[17] Lhomme, O., 1993, "Consistency Techniques for Numeric CSPs," *Proceedings of the IJCAI 93*, Chambery, France, Aug., pp. 232–238.
[18] Neumaier, A., 1990, *Interval Methods for Systems of Equations*, Cambridge University, Cambridge, London.
[19] Alefeld, G., 1984, "On the Convergence of Some Interval-Arithmetic Modifications of Newton's Method," SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal., **21**(2), pp. 363–372.
[20] Kearfott, R. B., 1988, "Corrigenda: Some Tests of Generalized Bisection," ACM Trans. Math. Softw., **14**(4), p. 399.
[21] Kearfott, R. B., and Novoa, M., III, 1990, "Algorithm 681: INTBIS, a Portable Interval Newton/Bisection Package," ACM Trans. Math. Softw., **16**(2), pp. 152–157.
[22] Lu, T.-F., Handley, D. C., Yong, Y. K., and Eales, C., 2004, "A Three-DOF Compliant Micromotion Stage With Flexure Hinges," Ind. Robot, **31**(4), pp. 355–361.
[23] Merlet, J.-P., 2000, "ALIAS: An Interval Analysis Based Library for Solving and Analyzing System of Equations," *Proceedings of the SEA*, Toulouse, France, June 14–16.
[24] Bonev, I., 2002, "Geometric Analysis of Parallel Mechanisms," Ph.D. thesis, Université Laval, Québec, QC, Canada.
[25] Merlet, J.-P., 2000, *Parallel Robots*, Kluwer, Dordrecht.
[26] Merlet, J.-P., and Donelan, P., 2006, "On the Regularity of the Inverse Jacobian of Parallel Robot," *Proceedings of the International Symposium of Advances in Robot Kinematics (ARK)*, Ljubljana, Slovenia, June, pp. 41–48.
[27] Merlet, J.-P., 2004, "Solving the Forward Kinematics of a Gough-Type Parallel Manipulator With Interval Analysis," Int. J. Robot. Res., **23**, pp. 221–235.
[28] Raghavan, M., 1991, "The Stewart Platform of General Geometry Has 40 Configurations," *Proceedings of the ASME Design and Automation Conference*, Vol. 32, pp. 397–402.
[29] Hansen, E., 1992, "Bounding the Solution of Interval Linear Equations," SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal., **29**, pp. 1493–1503.
[30] Rohn, J., 1993, "Cheap and Tight Bounds: The Recent Result by E. Hansen Can be Made More Efficient," Interval Computations, **4**, pp. 13–21.
[31] Neumaier, A., 1999, "A Simple Derivation of the Hansen–Bliek–Rohn–Ning–Kearfott Enclosure for Linear Interval Equations," Reliable Computing, **5**(2), pp. 131–136.