Workspace Computation of Planar Continuum Parallel Robots

Workspace Computation of Planar Continuum Parallel Robots

Federico Zaccaria[1,3], Edoardo Idá[1], Sébastien Briot[2] and Marco Carricato[1]

***Abstract*—Continuum parallel robots (*CPRs*) comprise several flexible beams connected in parallel to an end-effector. They combine the inherent compliance of continuum robots with the high payload capacity of parallel robots. Workspace characterization is a crucial point in the performance evaluation of *CPRs*. In this paper, we propose a methodology for the workspace evaluation of planar continuum parallel robots (*PCPRs*), with focus on the constant-orientation workspace. An explorative algorithm, based on the iterative solution of the inverse geometrico-static problem is proposed for the workspace computation of a generic *PCPR*. Thanks to an energy-based modelling strategy, and derivative approximation by finite differences, we are able to apply the Kantorovich theorem to certify the existence, uniqueness, and convergence of the solution of the inverse geometrico-static problem at each step of the procedure. Three case studies are shown to demonstrate the effectiveness of the proposed approach.**

***Index Terms*—Continuum Parallel Robots, Workspace Evaluation, Kantorovich Theorem,**

I. Introduction

CONTINUUM robots (*CRs*) [1] are manipulators that achieve movement through controlled displacement and deformation of slender elastic links. *CRs* are well suited for maneuvers in complex curvilinear pathways and when intrinsic flexibility is important. In particular, continuum parallel robots (*CPRs*) employ multiple slender links connected in parallel to an end-effector (*EE*) and, compared with their serial counterparts, they may exhibit higher precision, stability, and payload capacity [2],[3]. As recently highlighted [4], *CPRs* are good candidates for high precision manipulation tasks, or applications where intrinsic compliance is required and serial *CRs* suffer from payload limitations. While the basic architecture of a *CPR* is similar to a Gough-Stewart platform, different designs may lead to an increase in performance [5]. In order to assess which design better suits a specific operational task, performance quantification is required.

Workspace evaluation, i.e. the identification of all poses where the robot may lie in a stable equilibrium, is a crucial performance assessment tool. Though several geometrical, discretization and numerical methods are available in the literature for rigid-link manipulators [6], workspace computation algorithms for *CRs* are at a preliminary stage. Joint-space sampling approaches are used in [7],[8], for the computation of the *CR* reachable workspace. However, these approaches are computationally expensive to achieve accuracy. To overcome this disadvantage, an approximated approach is proposed in [9]. However, because this method can detect only points on the external surface of the workspace, it seems difficult to extend it to the identification of inner boundaries. An optimization-based approach is proposed in [10] for the workspace computation under a general modelling strategy that can encompass various *CRs*. However, even in this case, inner boundaries are not easy to detect.

As far as *CPRs* are concerned, the workspace of a 3-leg *CPR* is calculated in [11] by intersecting the workspace of each leg, with the latter being independently obtained by sampling the actuation space and computing the resulting *EE* poses. This approach, also used in [12] for a planar *CPRs* (*PCPRs*), allows the results to be obtained in a straightforward way, but it may lack accuracy since no interaction force is modelled between flexible legs and the rigid *EE*. A different strategy is employed in [13] for a 6-*DoF CPR*: the actuation space is sampled uniformly and, by iteratively solving the forward geometrico-static problem, the reachable workspace is obtained. Strain limits are considered by excluding configurations that overcome a certain threshold. This approach can be employed for a generic *CPR*, but singularity and stability analyses are not performed. Equilibrium stability is instead analyzed in [14], from an optimal-control point of view. The above limitations are also encountered in [4], where an inverse geometrico-static discretization approach is successfully used for the evaluation of the constant orientation workspace of *PCPRs*, but the modelling strategy does not include singularity and stability analysis. Similarly, an inverse kinematics discretization strategy is used in [15], and singular configurations are additionally detected by employing a simplified mathematical model; however, the latter cannot account for the action of external loads on the robot. The workspace of a two-leg *CPR* is computed in [16], by including the possibility of analyzing different working modes (i.e multiple solutions of the inverse problem [17]). However, the modelling strategy, based on the solution of elliptic integrals and limited to planar robots, does not seem to be easily applicable to a generic *CPR*.

A fundamental part of each workspace evaluation algorithm is the computation of the robot pose. In *CPRs*, this task is not trivial, since the geometry of the manipulator is not sufficient

Manuscript received: August, 31, 2021; Revised: November, 1, 2021, Accepted December, 27, 2021. (Corresponding Author: Federico Zaccaria)

This paper was recommended for publication by Editor A. Tamim upon evaluation of the Associate Editor L. Pallottino and Reviewers' comments.

[1]Federico Zaccaria, Edoardo Idá and Marco Carricato are with the Department of Industrial Engineering, University of Bologna, Italy (e-mail: {federico.zaccaria3,edoardo.ida2,marco.carricato}@unibo.it)

[2]Sébastien Briot is with the Laboratoire des Sciences du Numérique de Nantes (LS2N), Nantes, France (e-mail: Sebastien.Briot@ls2n.fr)

[3] Federico Zaccaria is with École Centrale de Nantes, LS2N, Nantes, France

Digital Object Identifier (DOI): 0000–0000/00$00.00.

to describe the pose of the whole robot, and its configuration is defined by the elastic deformation of the links. This problem is called geometrico-static and, in *CPRs*, forward and inverse problems do not generally admit an analytical solution. Consequently, the geometrico-static model is usually simplified by introducing numerical approximations [18] with the approximate solution being computed by numerical schemes (e.g. Newton-based methods). However, it may happen that no or multiple solutions exist for the same problem, and in this case solution certification becomes a crucial point. Interval analysis (IA) applies a branch-and-prune approach to solve a set of equations: it is mainly employed (i) when it is needed to certify that all solutions are found in a bounded region [19], and (ii) when dealing with bounded model parameter uncertainties [20],[21]. As a drawback, computational time of IA methods exponentially depends on the number of variables [22]. On the other side, the Kantorovich theorem [23] is a useful tool when the solution certification is important. This theorem establishes sufficient conditions on the initial guess of the Newton iteration to certify the existence, uniqueness, and convergence of the solution on a defined region [24]. While the application of the Kantorovich theorem is computationally efficient, certification of multiple solutions, and dealing with uncertainties are not possible, in contrast to IA approaches.

Our goal is to develop a methodology for the workspace evaluation of *PCPRs*, a class of *CPRs* which recently obtained higher scientific interest [4],[15]. We want to include several aspects usually of interest for workspace evaluation that state-of-the-art approaches cannot include simultaneously, namely equilibrium stability, singularity identification, strain limits, and external load influence. For this purpose, we decided to adopt the modelling strategy proposed in [25], since it allows us to apply external loads to any robot component, to enforce generic geometric constraints, to assess equilibrium stability, and to identify singular configurations (by the procedure introduced in [26]).

Our workspace-evaluation algorithm is based on the iterative solution of the inverse geometrico-static problem (*IGSP*) [26]. In order to certify the existence of a unique solution of the *IGSP* at each step of the workspace computation algorithm, and thus certify that all the computed configurations are characterized by the same working mode, we propose to apply the Kantorovich theorem. With this theorem, it is possible to ensure that a computed solution lies in the neighborhood of a previously computed configuration, and to certify the convergence of the numerical method used to find the new *IGSP* solution. By employing an energy-based modelling strategy combined with finite differences [25], we can obtain an analytical expression of the constants required for the solution certification through the Kantorovich theorem that may be not trivial to obtain with other approaches (i.e. the continuous formulation in [2], which may require the solution of an optimization problem). As it will also be shown, this methodology for the workspace computation is time-efficient[1].

[1]Since the model proposed in [25] uses discretized continuum-robot equations, a large number of independent variables is needed for the accurate solution of the *IGSP*, and high computational time is thus expected for IA-based approaches [22]

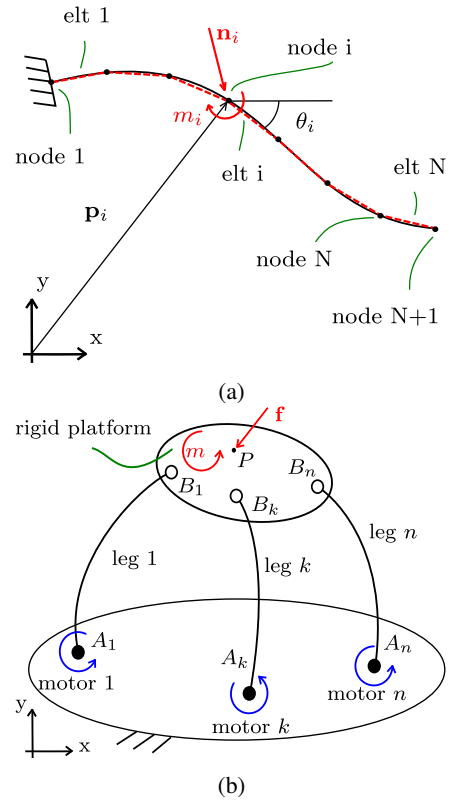


Fig. 1: (a) Discretization of a flexible link and (b) representation of a generic *PCPR*.

It allows obtaining certified results in a reduced computational time, which is beneficial since fast workspace computation algorithms may be used for workspace-guided design tools.

We additionally propose the use of a preconditioning matrix for the Newton iteration, to alleviate the numerical ill-conditioning of the *IGSP* problem and, consequently, to facilitate the *IGSP* solution certification.

Finally, we propose an adaptive-grid workspace-exploration algorithm. At each step of the workspace algorithm, the grid is refined in case the solution of the IGSP is not certifiable, so that a refined grid is used only where necessary (in contrast to the fixed grid flooding algorithm of [26]): the overall computation time is therefore reduced. The algorithm is then employed to identify stable and unstable workspace regions, and to detect singularities that determine workspace boundaries as in [26].

The paper is structured as follows. Section II introduces the *PCPR* modelling strategy, analyzes configuration stability and singularities, and discusses the Kantorovich theorem application. Section III describes the novel workspace exploration algorithm in detail. Section IV proposes three case studies to show the effectiveness of our approach. Section V draws conclusions and outlines future work directions.

II. Modelling

A. Modelling Approach

Consider a slender beam of length L as represented in Fig. 1a. Shear and extensibility are neglected. Each rod is

discretized into N elements of equal length $L_e = L/N$ and the orientation of the *i*-th element ($i = 1, ..., N$), with respect to (w.r.t.) a fixed frame, is θ_i. The deformation energy of the beam, assumed as initially straight, can be written as:

$$V_b(\theta_1, ..., \theta_N) = V_e + V_l \tag{1}$$

where V_e and V_l represent beam's deformation energy and the potential energy of concentrated and distributed loads acting on the beam, computed by approximating derivatives with finite differences (see [25],[26] for their expressions).

We assume that the *PCPR* comprises n beams, as in Fig. 1b, with the generic beam being identified by the index $k = 1, ..., n$. We can identify n controlled variables[2], collected in the vector $\mathbf{q}_A = [q_{a1}, ..., q_{an}]$. Moreover, we define $\boldsymbol{\theta}_k = [\theta_{2k}, ..., \theta_{Nk}]$ and $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_n]$. The position and orientation of the *EE* are denoted as $\mathbf{p}_P$ and ϕ, respectively, and grouped in the array $\mathbf{q}_P = [\mathbf{p}_P, \phi]$. The array $\mathbf{x} = [\boldsymbol{\theta}, \mathbf{q}_P]$ contains all non-actuated variables. Finally, the total potential energy of the *PCPR* can be written as:

$$V_t(\mathbf{q}_A, \mathbf{x}) = \sum_{k=1}^{n} V_{b,k}(q_{ak}, \boldsymbol{\theta}_k) - \mathbf{f}^T(\mathbf{p}_P - \mathbf{p}_P^*) - m(\phi - \phi^*) \tag{2}$$

where $\mathbf{f}, m$ are, respectively, a *EE* constant force and a constant moment (which ensures that the overall system is conservative), and $(.)^*$ denotes the undeformed state.

Due to the parallel architecture, the configuration variables are related by geometric constraints, such as closure equations. If beams are connected to the platform by revolute joints, closure equations can be written as:

$$\mathbf{p}_{Bk}(q_{Ak}, \boldsymbol{\theta}_k) - \mathbf{p}_{Bk}(\mathbf{p}_P, \phi) = \mathbf{0} \tag{3}$$

where $\mathbf{p}_{Bk}(q_{Ak}, \boldsymbol{\theta}_k)$ is function of the leg variables [25] and $\mathbf{p}_{Bk}(\mathbf{p}_P, \phi)$ is function of platform variables. Other beam connections to the *EE* can be modelled (such as fixed constraints [4]): $\boldsymbol{\Phi}$ is the array stacking m_ϕ general geometric constraints.

A static robot configuration is feasible, for an assigned value of $\mathbf{q}_A$, if and only if the manipulator is in a stable equilibrium, that is, the total potential energy is at a minimum. However, variables are related by constraints $\boldsymbol{\Phi}$, and local extrema of V_t are characterized by Lagrange conditions [27]. Assuming that $\nabla_{\mathbf{x}}\boldsymbol{\Phi}$ has full rank, $\mathbf{x}$ is a local extrema of V_t if Lagrange multipliers $\boldsymbol{\lambda}$ exist such as:

$$\nabla_{\mathbf{x}}\mathcal{L} = \mathbf{0} \ , \ \nabla_{\boldsymbol{\lambda}}\mathcal{L} = \mathbf{0} \tag{4}$$

where the Lagrangian function $\mathcal{L}$ is:

$$\mathcal{L}(\mathbf{q}_A, \mathbf{x}, \boldsymbol{\lambda}) = V_t(\mathbf{q}_a, \mathbf{x}) + \boldsymbol{\Phi}(\mathbf{q}_a, \mathbf{x})^T\boldsymbol{\lambda} \tag{5}$$

In order to assess if a configuration belongs to the static workspace, we are interested in determining if a value of $\mathbf{q}_A$ exists so that an assigned *EE* pose $\mathbf{q}_P^d$ is an equilibrium, that is, we seek the solution of the *IGSP*, defined as:

$$\mathbf{F}(\mathbf{y}) = \begin{cases} \mathbf{0} = \nabla_{\mathbf{x}}\mathcal{L} = \nabla_{\mathbf{x}}V_t + \nabla_{\mathbf{x}}\boldsymbol{\Phi}^T\boldsymbol{\lambda} \\ \mathbf{0} = \nabla_{\boldsymbol{\lambda}}\mathcal{L} = \boldsymbol{\Phi} \\ \mathbf{0} = \mathbf{q}_P - \mathbf{q}_P^d \end{cases} \tag{6}$$

[2]In the case of a revolute joint, $q_{ak} = \theta_{1k}$ whereas, in the case of a prismatic joint, the orientation θ_{1k} of the first element is a constant value and q_{ak} defines the position of the first node of the beam.

Equations (6) form a square system of $N_{eq} = n(N+2) + 3$ nonlinear equations in $\mathbf{y} = (\mathbf{q}_A, \boldsymbol{\theta}, \mathbf{q}_P, \boldsymbol{\lambda})$, which can be solved by using root-finding techniques, such as the Newton method.

B. Equilibrium Stability and Singularity Conditions

Once a solution of Eq. (6) is identified, its stability, and its singularities can be evaluated by the method presented in [26]. The Jacobian matrix $\mathbf{J}$ of Eq. (6), can be computed as:

$$\mathbf{J}(\mathbf{y}) = \frac{\partial \mathbf{F}(\mathbf{y})}{\partial \mathbf{y}} = \begin{bmatrix} \mathbf{A}_{\mathcal{L}} & \mathbf{U}_{\mathcal{L}} & \mathbf{P}_{\mathcal{L}} & \boldsymbol{\Lambda}_{\mathcal{L}} \\ \mathbf{A}_{\Phi} & \mathbf{U}_{\Phi} & \mathbf{P}_{\Phi} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix} \tag{7}$$

where:

- $\mathbf{A}_{\mathcal{L}} = \nabla_{\mathbf{q}_A}(\nabla_{\mathbf{x}}\mathcal{L})$, $\mathbf{U}_{\mathcal{L}} = \nabla_{\boldsymbol{\theta}}(\nabla_{\mathbf{x}}\mathcal{L})$
- $\mathbf{P}_{\mathcal{L}} = \nabla_{\mathbf{q}_P}(\nabla_{\mathbf{x}}\mathcal{L})$, $\boldsymbol{\Lambda}_{\mathcal{L}} = \nabla_{\boldsymbol{\lambda}}(\nabla_{\mathbf{x}}\mathcal{L})$
- $\mathbf{A}_{\Phi} = \nabla_{\mathbf{q}_A}\boldsymbol{\Phi}$, $\mathbf{U}_{\Phi} = \nabla_{\boldsymbol{\theta}}\boldsymbol{\Phi}$, $\mathbf{P}_{\Phi} = \nabla_{\mathbf{q}_P}\boldsymbol{\Phi}$

and $\mathbf{I}$ is the identity matrix. The analytical expression of all terms in $\mathbf{J}$ is obtainable, but not reported here for brevity. Equilibrium stability can be assessed by verifying that the reduced hessian matrix $\mathbf{H}^r$ of the total potential energy is positive definite [27]. $\mathbf{H}^r$ can be obtained as:

$$\mathbf{H}^r = \mathbf{Z}^T\frac{\partial^2 V_t}{\partial \mathbf{x}^2}\mathbf{Z} = \mathbf{Z}^T\begin{bmatrix}\mathbf{U}_{\mathcal{L}} & \mathbf{P}_{\mathcal{L}}\end{bmatrix}\mathbf{Z} \tag{8}$$

where $\mathbf{Z}$ denotes the left nullspace of $\boldsymbol{\Lambda}_{\mathcal{L}}$. Moreover, configuration singularities can be assessed if the following matrices degenerate [26]:

$$\mathbf{T}_1 = \begin{bmatrix}\mathbf{Z}^T\mathbf{A}_{\mathcal{L}} & \mathbf{Z}^T\mathbf{U}_{\mathcal{L}} \\ \mathbf{A}_{\Phi} & \mathbf{U}_{\Phi}\end{bmatrix}, \mathbf{T}_2 = \begin{bmatrix}\mathbf{Z}^T\mathbf{P}_{\mathcal{L}} & \mathbf{Z}^T\mathbf{U}_{\mathcal{L}} \\ \mathbf{P}_{\Phi} & \mathbf{U}_{\Phi}\end{bmatrix} \tag{9}$$

Type 1 (or serial) singularities are related to the degeneracy of $\mathbf{T}_1$ and define the limits of the robot workspace. Type 2 (or parallel) singularities, related to $\mathbf{T}_2$, delimit stable-to-unstable regions.

C. Kantorovich Theorem for Newton Iteration and Preconditioning

If a Newton method is employed for the solution of the *IGSP* in Eq. (6), the Kantorovich theorem establishes sufficient conditions for the existence and uniqueness of the solution [23]. Let $\chi, \delta, \lambda \in \mathbb{R}^+$, $\mathbf{y}_0$ be the initial guess given to the numerical solver, and $\mathcal{B}(\mathbf{y}_0, 2\delta) = \{\mathbf{y} : \|\mathbf{y} - \mathbf{y}_0\| \leq 2\delta\}$ a ball of radius 2δ centered in $\mathbf{y}_0$. For convenience, let the infinite norm being used. Let $\mathbf{J}(\mathbf{y})$ be the Jacobian matrix of $\mathbf{F}(\mathbf{y})$ w.r.t. $\mathbf{y}$, and χ, δ, λ are chosen such as:

$$\chi \geq \|\mathbf{J}^{-1}(\mathbf{y}_0)\| \tag{10}$$

$$\delta \geq \|\mathbf{J}^{-1}(\mathbf{y}_0)\mathbf{F}(\mathbf{y}_0)\| \tag{11}$$

$$\lambda \geq \max_{\mathbf{y}\in\mathcal{B}}\left(\max_{h\in[1,N_{eq}]}\left(\sum_{i,j}\left|\frac{\partial^2 F_h(\mathbf{y})}{\partial y_i \partial y_j}\right|\right)\right) \tag{12}$$

Then, if χ, δ, λ exist such that $2\chi\delta\lambda \leq 1$, the Kantorovich theorem states that a solution $\mathbf{y}^* \in \mathcal{B}(\mathbf{y}_0, t^*)$ exists, where $t^* = \frac{1-\sqrt{1-2\chi\delta\lambda}}{\lambda\chi} \in [\delta, 2\delta]$. Moreover, the solution is unique inside $\mathcal{B}(\mathbf{y}_0, 2\delta)$, and the Newton iteration, starting from $\mathbf{y}_0$,

converges to $\mathbf{y}^*$. Kantorovich constants have a numerical meaning: χ is related to the absolute conditioning of the problem, δ represents the closeness to the linearized solution, and λ is influenced by the non-linearity of the problem.

However, if a discretization approach is used for modelling *PCPRs*, the Jacobian matrix $\mathbf{J}$ is usually large and ill-conditioned, resulting in high values of χ. Therefore, the certification of the existence, uniqueness, and convergence of the *IGSP* solution holds only for $\mathbf{y}_0$ sufficiently close to $\mathbf{y}^*$. To overcome this difficulty, we can preconditionate the *IGSP* to reduce the value of χ. We propose to solve an equivalent and modified inverse geometrico-static problem (*MIGSP*), where each *IGSP* equation is multiplied by a constant that represents the physical dimensions of the equation terms:

$$\mathbf{F}_M(\mathbf{y}) = \begin{cases} \mathbf{0} = \frac{1}{EI}(\nabla_{\boldsymbol{\theta}} V_t + \nabla_{\boldsymbol{\theta}} \boldsymbol{\Phi}^T \boldsymbol{\lambda}) \\ \mathbf{0} = \frac{L}{EI}(\nabla_{\mathbf{p}_P} V_t + \nabla_{\mathbf{p}_P} \boldsymbol{\Phi}^T \boldsymbol{\lambda}) \\ \mathbf{0} = \frac{1}{EI}(\nabla_{\phi} V_t + \nabla_{\phi} \boldsymbol{\Phi}^T \boldsymbol{\lambda}) \\ \mathbf{0} = \frac{1}{L}\boldsymbol{\Phi} \\ \mathbf{0} = \frac{1}{L}(\mathbf{p}_P - \mathbf{p}_P^d) \\ \mathbf{0} = \boldsymbol{\phi} - \boldsymbol{\phi}^d \end{cases} \tag{13}$$

or equivalently:

$$\mathbf{F}_M(\mathbf{y}) = \mathbf{M}\mathbf{F}(\mathbf{y}) \tag{14}$$

where $\mathbf{M}$ is a diagonal matrix collecting the coefficients that multiply each equation of Eq. (13). The Jacobian matrix of the *MIGSP* w.r.t. $\mathbf{y}$ is:

$$\mathbf{J}_M(\mathbf{y}) = \frac{\partial \mathbf{F}_M(\mathbf{y})}{\partial \mathbf{y}} = \mathbf{M}\frac{\partial \mathbf{F}(\mathbf{y})}{\partial \mathbf{y}} = \mathbf{M}\mathbf{J}(\mathbf{y}) \tag{15}$$

The Kantorovich constants of the *MIGSP* can be obtained from Eqs. (10),(11),(12) by employing $\mathbf{F}_M, \mathbf{J}_M$ instead of $\mathbf{F}, \mathbf{J}$. While the computation of χ, δ is straightforward, the computation of λ requires to analytically compute all the second derivative of *MIGSP* equations, which is long and involved. According to Eq. (12), we have to identify where the sum in absolute value of the second derivatives assumes the maximum value inside the ball for each equation. However, this sum is not constant w.r.t. $\mathbf{y}$ and we may identify the maximum by solving a constrained optimization problem, at the cost of high computational time. We preferred to approximate λ with absolute value inequalities [27]: in this way we consider the worst-case scenario and we obtain an expression of λ that depends on $\mathbf{y}_0, \delta$. The details of this computation are not provided here due to space limitations.

III. Workspace Evaluation

This Section describes in detail the innovative workspace-computation strategy, which we call Adaptive Flooding Algorithm (Alg. 1 lines 1-19). In particular, we modified the Flooding Algorithm of [26] by introducing a grid-adaptation procedure: during each iteration, the grid size is adjusted when the certification is not feasible with the initial grid size. This is done in order to certify as much workspace as possible, since the Kantorovich constant δ depends on the distance of the initial guess from the solution. We focus on the constant-orientation workspace of *PCPRs*, i.e. the set of all possible locations of the robot *EE* that can be reached with a given orientation, though our approach can be easily extended to other types of workspace. A discretization approach is employed and the *MIGSP* is solved iteratively over a grid of *EE* locations.

Algorithm 1: Adaptive flooding algorithm.

```
1  Initialize grid, toDo, toDoEnd, Config, Results;
2  while toDo≠ ∅ or toDoEnd≠ ∅ do
3      if toDo= ∅ then
4          toDo ← toDoEnd; toDoEnd ← ∅ ;
5      end
6      p_P = toDo(1), toDo ← toDo\p_P;
7      [flag_k,y_0,n] = FindGuess(p_P, s_c);
8      if s_c > s_m then
9          if flag_k ≤ 1 then
10             Compute(y_0,n);
11         else
12             p_new = Generate points(p_P,s_c);
13             Replace p_P with p_new;
14             toDo ← toDo ∪ p_new
15         end
16     else
17         Compute(y_0,n);
18     end
19 end
20 Function FindGuess(p_P, s_c):
21     Set neighbors radius r_N = √2 s_c;
22     do
23         n = Find neighbors EE positions to p_P ∈ grid;
24         n_WK = n ∈WK;
25         if (n_WK = ∅) then
26             r_N ← 2r_N;
27         end
28     while (n_WK = ∅);
29     Compute Kantorovich flag for each n_WK;
30     n_best = n_WK with best Kantorovich flag;
31     y_0 = Config(n_best);
32     return [flag_k, y_0, n];
33 Function Compute(y_0,n):
34     y = Solve IGSP starting from y_0;
35     [T_1, T_2] = Singularity(y);
36     if (Solver Converged & T_1 < TOL & mechconstr(y)) then
37         Values = CalculateOutputs(y);
38         Save Values in Results, Save y in Config;
39         n_1 = select n ∉(WK,toDo,toDoEnd);
40         if T_2 < TOL then
41             toDo ← toDo ∪ n_1;
42         else
43             toDoEnd ← toDoEnd ∪ n_1;
44         end
45     end
46     return;
```

The first step of the algorithm requires the initialization of some entities: a 2-dimensional uniform grid is generated accordingly to an initial stepsize (s_i). The grid discretizes a user-defined box (assumed to fully include the workspace) and in the middle of each square of the grid, a point representing the *EE* position is placed (Fig. 2a). An initial point (named $\mathbf{p}_P$) where the explorative algorithm starts is selected within the grid. For the first solution of the *MIGSP*, an initial guess obtained through constant-curvature modelling approach is employed. As shown in [15], the inverse problem under constant-curvature approach admits a finite number of solutions (but usually low solution accuracy), and we can decide which solution to use as initial guess, in order to let the algorithm converge to the desired working mode [17]. The resulting output configuration is stacked in the *Config* list to be employed as a future guess. Then, neighboring *EE* positions (Fig. 2a) are computed and saved in the *toDo* list, which contains points to be processed. The *toDoEnd* list,

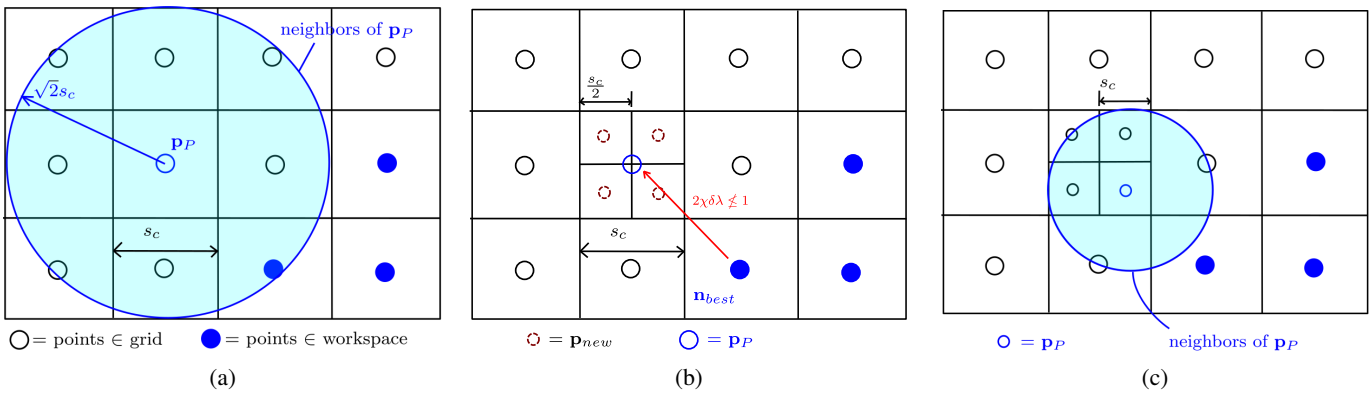


Fig. 2: Workspace algorithm: (a) representation of the grid, points along the grid and condition for neighborhood, (b) grid refinement strategy; (c) situation when neighbor radius should be enlarged.

that contains points that are neighbors to Type 2 singular configurations, is initialized as empty. Then, the iterative algorithm starts, a new $\mathbf{p}_P$ is obtained from the first element of the $toDo$ list and employed as the *EE* location to be reached. The initial guess $\mathbf{y}_0$ for the *MIGSP*, the Kantorovich flag associated to $\mathbf{y}_0$ and neighbors of $\mathbf{p}_P$ are obtained through the *FindGuess* procedure. The latter, which plays an important role in the certification of the *MIGSP* solution, is detailed in subsection III-A.

If the current step s_c (i.e the edge size of the square centered in $\mathbf{p}_P$) is greater to the minimum stepsize allowed for the computation (s_m), the value of the Kantorovich constants is checked to certify the solution: if $2\chi\delta\lambda \leq 1$, the *MIGSP* is solved and new points to be processed are created in the *Compute* procedure (detailed in subsection III-B). If the manipulator is not cuspidal [28], and the solution is certified, the working mode of the initial guess is preserved also on the configuration resulting from the *MIGSP* solution. In the case of the solution is not certifiable, the grid is refined: the square centered in $\mathbf{p}_P$ is divided into new four equal squares (Fig. 2b), and new four points $\mathbf{p}_{new}$ are placed in the middle of each new square. Then, the original *EE* location $\mathbf{p}_P$ is replaced with $\mathbf{p}_{new}$ and added to the *toDo* list. In the case the minimum stepsize is reached ($s_c \leq s_m$), the *Compute* procedure is executed even if the *MIGSP* solution is not certified, in order to fully compute the workspace. Finally, the algorithm restarts until some elements are present in the lists *toDo* and *toDoEnd*. If *toDo* is empty, then it is refilled with *toDoEnd* (how *toDo* and *toDoEnd* are managed is explained in subsection III-B).

A. Choice of the Initial Guess

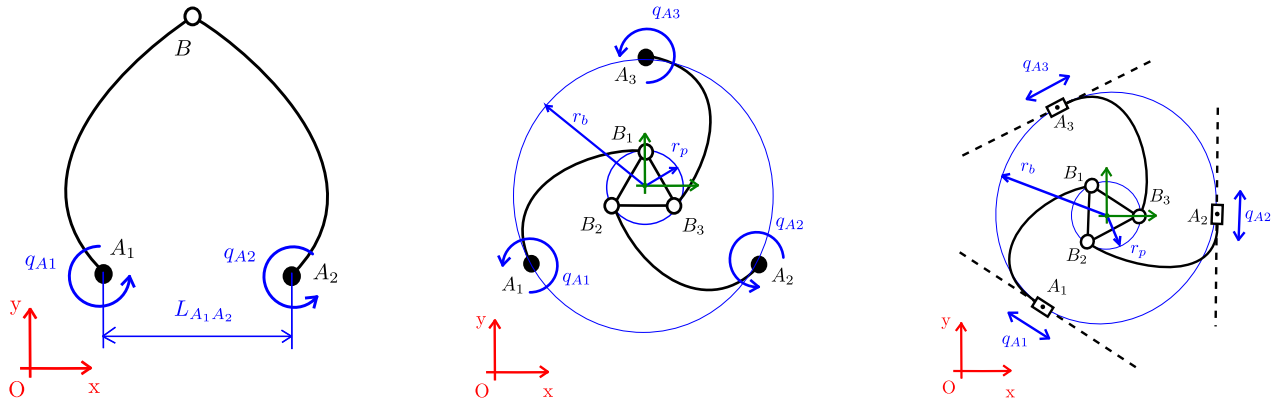
The choice of the initial guess at each iteration plays an important role in the *MIGSP* solution certification. The routine of the initial guess selection is described in lines 20-32 of Alg. 1. Given a desired *EE* location $\mathbf{p}_P$, we want to identify an initial-guess configuration $\mathbf{y}_0$ to be used for the *MIGSP* solution. Initially, the distance r_N for which two *EE* locations are considered to be neighbors is set as $\sqrt{2}s_c$ (Fig. 2a). This way, neighbors of $\mathbf{p}_P$ are identified in the grid and stacked in the array $\mathbf{n}$. Neighbors in the workspace are extracted from $\mathbf{n}$ and collected in the array $\mathbf{n}_{WK}$. However, caused by the grid refinement process, it can happen that no neighbors are in the workspace (Fig. 2c): in this case, the radius r_N for which points are considered to be neighbors is multiplied by two, and the selection of $\mathbf{n}$ is repeated until workspace points are found.

In order to increase the possibility to certify the *MIGSP* solution, the Kantorovich constants for the configurations associated to $\mathbf{n}_{WK}$ *EE* locations are computed. Then, the *EE* location that ensures the lowest value of $2\chi\delta\lambda$ (named $\mathbf{n}_{best}$) is identified, and the robot configuration $\mathbf{y}_0$ associated to $\mathbf{n}_{best}$ is extracted from *Config*. If the solution certification is not required, $\mathbf{n}_{best}$ can be chosen as the one that ensures the best inverse conditioning of $\mathbf{J}$ to speed up the computation.

B. Computation Process

This subsection describes the routine for the computation of the *MIGSP* solution and for the creation of new points to be processed (lines 33-46 of Alg. 1). Starting from a given initial guess $\mathbf{y}_0$ and a set of neighboring *EE* locations $\mathbf{n}$, the *MIGSP* is solved by a Newton scheme. Then, $\mathbf{T}_1, \mathbf{T}_2$ are computed according to Eqs. (9). In particular, we compute the inverse condition number of $\mathbf{T}_1, \mathbf{T}_2$ in order to identify their degeneracy. Then, we check if the Newton solver converges and the resulting configuration is not Type-1 singular. Moreover, mechanical constraints are verified in *mechconstr*: these include strain limits on the legs, as well as joint limits. If the check succeeded, outputs associated with the resulting configuration $\mathbf{y}$ (e.g. internal energy of the robot, equilibrium stability, number of inflection points) are computed in *Values*. These results, as well as singularity flags and Kantorovich constants, are saved in *Results* and $\mathbf{y}$ is stored in *Config* as a future initial guess. Subsequently, neighbors not in the workspace and not in *toDo*,*toDoEnd*, are stored in $\mathbf{n}_1$. If the actual configuration $\mathbf{y}$ is not T_2-singular, $\mathbf{n}_1$ is added to the *toDo* list, else in *toDoEnd*.

In this way, Type 1 singularities, associated with boundaries of the workspace, are not crossed but only approached. Type 2 singularities, which delimit stable from unstable regions, are crossed in a second stage of the algorithm (only when *toDo* is empty) in order to discover possible stable regions separated by unstable transitions.



(a) $\underline{R}FRF\underline{R}$ (b) $3-\underline{R}FR$ (c) $3-\underline{P}FR$

Fig. 3: Three *PCPRs* object of our case studies: the $\underline{R}FRF\underline{R}$ (a), the $3-\underline{R}FR$ (b), and the $3-\underline{P}FR$ (c). Relevant design dimensions are displayed.

IV. Case Studies

In this Section, three case studies, focusing on different *PCPRs* (Fig. 3), are reported. Workspace evaluation is discussed with a focus on the identification of different regions (e.g. stable, unstable, regions where stress limits are exceeded) and certification of the *IGSP* solution. Our algorithm is compared to the flooding algorithm of [26], to demonstrate the benefit of the grid-adaptation routine in terms of computational time. For all case studies, beams are made of harmonic steel with Young modulus $E = 210$ GPa, maximum stress $\sigma_{max} = 1800$ MPa, density $\rho = 7800$ kg/m^3, length $L = 1$m , circular cross-section of radius $r = 1$mm. Simulations are performed in the Matlab enviroment.

A. $\underline{R}FRF\underline{R}$ robot

This subsection investigates the workspace of a $\underline{R}FRF\underline{R}$ robot. The aim of this case study is twofold: on the one hand, it shows the capability of our algorithm to detect singularities and unstable regions, as well as to include external loads and strain limits in the model; on the other hand, it investigates the influence of the stepsize and the preconditioner on the *IGSP* solution certification.

This manipulator, borrowed from [16],[25], has two actuated revolute joints in A_1, A_2 ($\underline{R}$) and two flexible links (F) connected by a passive revolute joint centered in B (Fig. 3a). The distance between the actuators is $L_{A_1A_2} = 0.4$m. An external force of 1.5N is applied on the *EE*, and legs are subjected to gravity. Simulations are performed with $N = 50$, ensuring sufficient *MIGSP* solution accuracy.

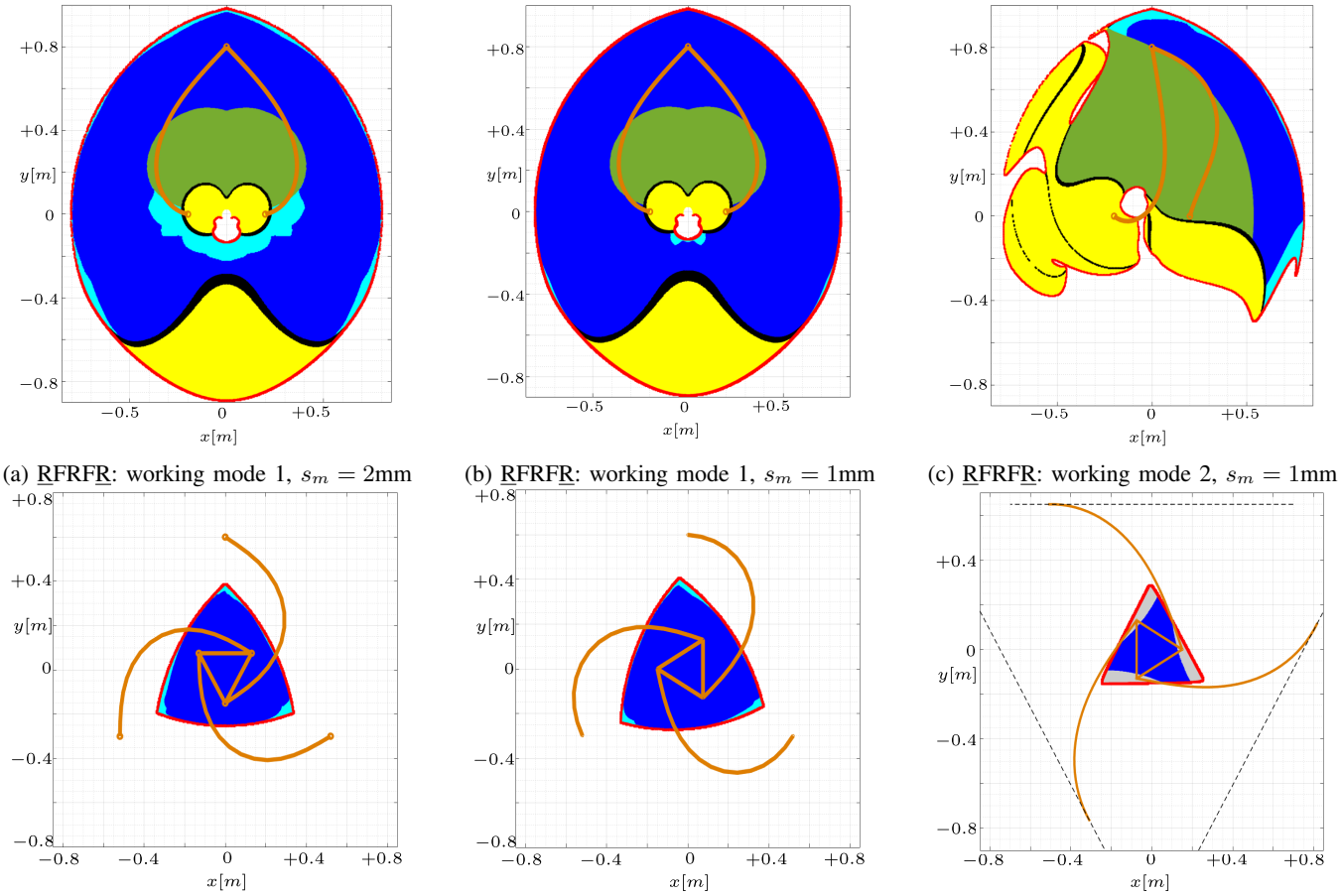
The workspace of the $\underline{R}FRF\underline{R}$, computed by the algorithm presented in Section III, is shown in Figs. 4a, 4b, 4c. Configurations are marked as singular when the inverse condition number of matrices reported in Eq. (9) is lower than a certain threshold TOL. We experienced that $TOL = 10^{-6}$ correctly identify singularities. Stress limits are considered by evaluating whether a first-order approximation of stress [29] on each element does not overcome σ_{max}. By considering stress limits, the workspace is considerably reduced, in particular in the working mode showed in Fig. 4c (0.96 m^2 without considering stress limits to 0.36 m^2 with stress limit included).

We also tested the influence of the minimum stepsize s_m and the influence of the preconditioner on the *MIGSP* certification of the workspace, with a focus on the stable and feasible workspace (i.e where stress limits or joint bounds are not exceeded). To quantify how many configurations are computed in a certified way, we introduce the certified percentage of the workspace $C_\%$ as:

$$C_\% = 100 \cdot A_C / A_W \tag{16}$$

where A_C is the certified area and A_W is the workspace area. These areas are obtained by summing the area of each square of the grid that lies in the workspace (for A_W) and by summing the area of each square that is computed in a certified way and belongs to the workspace for A_C. Results are reported in Table I. With $s_i = 4$mm, by passing from $s_m = 2$mm (Fig. 4a) to $s_m = 1$mm (Fig. 4b), $C_\%$ grows from 80.4% to 92.1%. As expected, the computational time[3] increases from 32 min to 84 min. To achieve $C_\% = 92.1\%$ the flooding algorithm of [26] required 192 min with 1 mm stepsize, which is considerably higher. If the preconditioner is not used, $C_\%$ drops to 35.1% (with $s_m = 1$mm) and at the same time, the computational time reaches 383 min. The increase of the computational time is due to the higher number of processed points ($18.9 \cdot 10^5$ compared to $7.1 \cdot 10^5$ in the preconditioned case) for the adaptation of the grid, and not for a considerable increase in the *MIGSP* solution time. This is confirmed by the data relative to the case of $s_m = s_i = 4$mm (where no grid refinement is possible), with the same grid being employed with and without the preconditioner, and the resulting computational times are comparable. Also in the case of a different working mode (Fig. 4c), we certified a significant amount of the workspace (70.2%) in a reduced time (44 min) with $s_m = 1$mm.

[3]Results are obtained by a CPU Intel Core i7-8700K,3.7GHz,32Gb RAM



(a) RFRFR: working mode 1, $s_m = 2$mm (b) RFRFR: working mode 1, $s_m = 1$mm (c) RFRFR: working mode 2, $s_m = 1$mm

(d) 3-RFR: $\phi = -\frac{\pi}{3}$, $s_m = 1$mm (e) 3-RFR: $\phi = -\frac{\pi}{6}$, $s_m = 1$mm (f) 3-PFR:$\phi = +\frac{\pi}{6}$, $s_m = 1$mm

Fig. 4: Workspaces of *PCPRs*. Type 1 and Type 2 singularities are drawn in red and black, respectively. Certified workspace is depicted in blue, non certified workspace in light blue and non certified unstable regions in yellow. Non certified stable regions where stress limits or joint limits are exceeded are represented in green and grey, respectively.

B. $3-\underline{R}FR$ *robot*

This subsection investigates the workspace of the $3-\underline{R}FR$ robot. This case study shows the possibility of identifying the workspace and certifying the *IGSP* solution with different *EE* orientations. This manipulator borrowed from [15] has three actuated revolute joints A_1, A_2, A_3 ($\underline{R}$) and three flexible links (F) connected by passive revolute joints B_1, B_2, B_3 at a rigid *EE*. Actuators are equally placed along a circumference of radius $r_b = 0.6$m, whereas passive revolute joints are placed on a circumference of radius $r_p = 0.15$m (Fig. 3b). Simulations are performed with $N = 30$ and no external loads are included.

The workspace of the $3-\underline{R}FR$ robot is illustrated in Fig. 4d and 4e by fixing $\phi = -\frac{\pi}{3}$, $\phi = -\frac{\pi}{6}$, respectively. In both cases, no unstable regions and Type-2 singularities are detected. Stress limits are included, but no point exceeds σ_{max}. As before, $s_m = 1$mm is chosen to guarantee a sufficient value of $C_\%$ (81% and 84.6%) in a reasonable computational time (11 and 10 min). Again, the flooding algorithm of [26] required higher computational time to obtain the same $C_\%$ (20 and 18.5 min with 1 mm stepsize).

C. $3-\underline{P}FR$ *robot*

This subsection studies the workspace of the $3-\underline{P}FR$ robot. This case study shows the possibility of analyzing *PCPRs* with different actuators and including joint limits. This manipulator is similar to the one proposed in [4], except for the connection of the flexible links with the platform (passive revolute joints in our case, in contrast with fixed connections in [4]). The $3-\underline{P}FR$ robot has three actuated prismatic joints A_1, A_2, A_3 ($\underline{P}$) and three flexible links (F) connected by passive revolute joints B_1, B_2, B_3 at a rigid *EE*. Actuators are equally spaced along a circumference of radius $r_b = 0.65$m, and passive revolute joints are placed on the platform of radius $r_p = 0.15$m (Fig. 3c). Simulations are performed with $N = 30$ and no external loads are included..

The workspace of the $3-\underline{P}FR$ robot is illustrated in Fig. 4f, where the *EE* orientation is $\phi = \frac{\pi}{6}$ and 1.4m-long rails are symmetrically placed around a circle of radius r_b. As for the $3-\underline{R}FR$ robot, maximum strain limits are included, but not exceeded. With $s_m = 1$ mm, we reached $C_\% = 94.5\%$ in 5.2 min of computational time whereas the algorithm of [26] required 17 min with 1 mm stepsize.

$\underline{R}FRF\underline{R}$						
s_m	With preconditioning			Without preconditioning		
	Certified [%]	Total Time $[min]$	N° of points	Certified [%]	Total Time $[min]$	N° of points
4	33.8	11	1.3 $\cdot 10^5$	0	13	1.3 $\cdot 10^5$
2	80.4	32	4.0 $\cdot 10^5$	5.5	80	5.4 $\cdot 10^5$
1	92.1	84	7.1 $\cdot 10^5$	35.1	383	18.9 $\cdot 10^5$

TABLE I: Influence of the preconditioner and the stepsize on the certification of the workspace. Simulations, relative to the workspace displayed in Fig. 4a, 4b, are performed with $s_i = 4$mm, and $N = 50$.

V. Conclusions

This paper presented an adaptive flooding algorithm for the workspace computation of *PCPRs*. The algorithm may identify unstable regions, singularity loci, incorporate external loads, set maximum stress limits and joint bounds. Thanks to an energy-based modelling strategy approximated through finite differences for derivatives, the IGSP solution was certified in terms of existence, uniqueness, and convergence of the solution, by verifying Kantorovich conditions during the Newton-based problem-solving procedure. With this approach, we certified the IGSP solution over a large percentage of the workspace in a reduced computational time in comparison with previous algorithms. However, with large workspaces and/or small stepsizes, the flooding approach may require the computation of a large number of points, which may be not computationally efficient. Future work will be directed on the integration of the adaptive flooding algorithm on a workspace-guided design tool for *PCPRs*. Moreover, the authors will investigate the workspace evaluation of *CPRs* and the certification of the solution of the IGSP problem in the spatial case.

References

[1] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum Robots for Medical Applications: A Survey," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1261–1280, 2015.

[2] C. B. Black, J. Till, and D. C. Rucker, "Parallel continuum robots: Modeling, analysis, and actuation-based force sensing," *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 29–47, 2017.

[3] G. Chen, Y. Kang, Z. Liang, Z. Zhang, and H. Wang, "Kinetostatics modeling and analysis of parallel continuum manipulators," *Mechanism and Machine Theory*, vol. 163, p. 104380, 2021.

[4] B. Mauzé, R. Dahmouche, G. J. Laurent, A. N. André, P. Rougeot, P. Sandoz, and C. Clévy, "Nanometer precision with a planar parallel continuum robot," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 3806–3813, 2020.

[5] J. A. Childs and C. Rucker, "Leveraging geometry to enable high-strength continuum robots," *Frontiers in Robotics and AI*, vol. 8, 2021.

[6] J.-P. Merlet, *Parallel robots*. Springer Science & Business Media, 2005, vol. 128.

[7] J. Burgner-Kahrs, H. B. Gilbert, J. Granna, P. J. Swaney, and R. J. Webster, "Workspace characterization for concentric tube continuum robots," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, USA*, 2014, pp. 1269–1275.

[8] D. Trivedi, D. Lesutis, and C. D. Rahn, "Dexterity and workspace analysis of two soft robotic manipulators," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Quebec, Canada*, 2010, pp. 1389–1398.

[9] K. Cao, R. Kang, D. T. Branson III, S. Geng, Z. Song, and J. S. Dai, "Workspace analysis of tendon-driven continuum robots based on mechanical interference identification," *ASME Journal of Mechanical Design*, vol. 139, no. 6, pp. 1–11, 2017.

[10] W. Amehri, G. Zheng, A. Kruszewski, and F. Renda, "Discrete cosserat method for soft manipulators workspace estimation: an optimization-based approach," *Journal of Mechanisms and Robotics*, pp. 1–12, 2021.

[11] I. Singh, M. Singh, P. M. Pathak, and R. Merzouki, "Optimal work space of parallel continuum manipulator consisting of compact bionic handling arms," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2017, pp. 258–263.

[12] K. Nuelle, T. Sterneck, S. Lilge, D. Xiong, J. Burgner-Kahrs, and T. Ortmaier, "Modeling, calibration, and evaluation of a tendon-actuated planar parallel continuum robot," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5811–5818, 2020.

[13] A. L. Orekhov, C. B. Black, J. Till, S. Chung, and D. C. Rucker, "Analysis and validation of a teleoperated surgical parallel continuum manipulator," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 828–835, 2016.

[14] J. Till and D. C. Rucker, "Elastic stability of cosserat rods and parallel continuum robots," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 718–733, 2017.

[15] S. Lilge, K. Nülle, G. Böttcher, S. Spindeldreier, and J. Burgner-Kahrs, "Tendon Actuated Continuous Structures in Planar Parallel Robots: A Kinematic Analysis," *Journal of Mechanisms and Robotics*, vol. 13, p. 011025, 2020.

[16] O. Altuzarra, D. Caballero, F. J. Campa, and C. Pinto, "Position analysis in planar parallel continuum mechanisms," *Mechanism and Machine Theory*, vol. 132, pp. 13–29, Feb. 2019.

[17] I. A. Bonev, D. Chablat, and P. Wenger, "Working and assembly modes of the agile eye," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, Orlando, Florida, USA*, 2006, pp. 2317–2322.

[18] J. Simo and L. Vu-Quoc, "A three-dimensional finite-strain rod model. Part II: Computational aspects," *Computer methods in applied mechanics and engineering*, vol. 58, no. 1, pp. 79–116, 1986.

[19] O. Altuzarra and J. P. Merlet, "Certified kinematics solution of 2-dof planar parallel continuum mechanisms," in *Advances in Mechanism and Machine Science, IFToMM WC 2019. Mechanisms and Machine Science*, T. Uhl, Ed., vol. 73, 2019, pp. 197–208.

[20] N. Tan, X. Gu, and H. Ren, "Pose characterization and analysis of soft continuum robots with modeling uncertainties based on interval arithmetic," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 2, pp. 570–584, 2018.

[21] S. Iqbal, S. Mohammed, and Y. Amirat, "A guaranteed approach for kinematic analysis of continuum robot based catheter," in *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO), Guilin, Chins*. IEEE, 2009, pp. 1573–1578.

[22] J.-P. Merlet, "Interval analysis for certified numerical solution of problems in robotics," *International Journal of Applied Mathematics and Computer Science*, vol. 19, no. 3, pp. 399–412, 2009.

[23] L. Kantorovich, "On Newton's method for functional equations," in *Dokl. Akad. Nauk SSSR*, vol. 59, no. 7, 1948, pp. 1237–1240.

[24] A. Goldsztejn, S. Caro, and G. Chabert, "A three-step methodology for dimensional tolerance synthesis of parallel manipulators," *Mechanism and Machine Theory*, vol. 105, pp. 213–234, 2016.

[25] F. Zaccaria, S. Briot, M. T. Chikhaoui, E. Idá, and M. Carricato, "An Analytical Formulation for the Geometrico-static Problem of Continuum Planar Parallel Robots," in *Symposium on Robot Design, Dynamics and Control, ROMANSY2020, G.Ventuare et al. (Eds.)*, no. 512–520, 2020.

[26] S. Briot and A. Goldsztejn, "Singularity conditions for continuum parallel robots," *IEEE Transactions on Robotics*, pp. 1–19, 2021.

[27] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed. Springer, 2006.

[28] P. Wenger, "Cuspidal and noncuspidal robot manipulators," *Robotica*, vol. 25, no. 6, p. 677–689, 2007.

[29] S. Antman, *Nonlinear Problems of Elasticity*. Springer Verlag New York, 1995, vol. 107.