



HAL
open science

Certified Kinematic Tools for the Design and Control of Parallel Robots

Alexandre Lê, Fabrice Rouillier, Guillaume Rance, Damien Chablat

► **To cite this version:**

Alexandre Lê, Fabrice Rouillier, Guillaume Rance, Damien Chablat. Certified Kinematic Tools for the Design and Control of Parallel Robots. Mechanism and Machine Theory, 2025, 205, pp.105865. 10.1016/j.mechmachtheory.2024.105865 . hal-04821252

HAL Id: hal-04821252

<https://hal.science/hal-04821252v1>

Submitted on 5 Dec 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

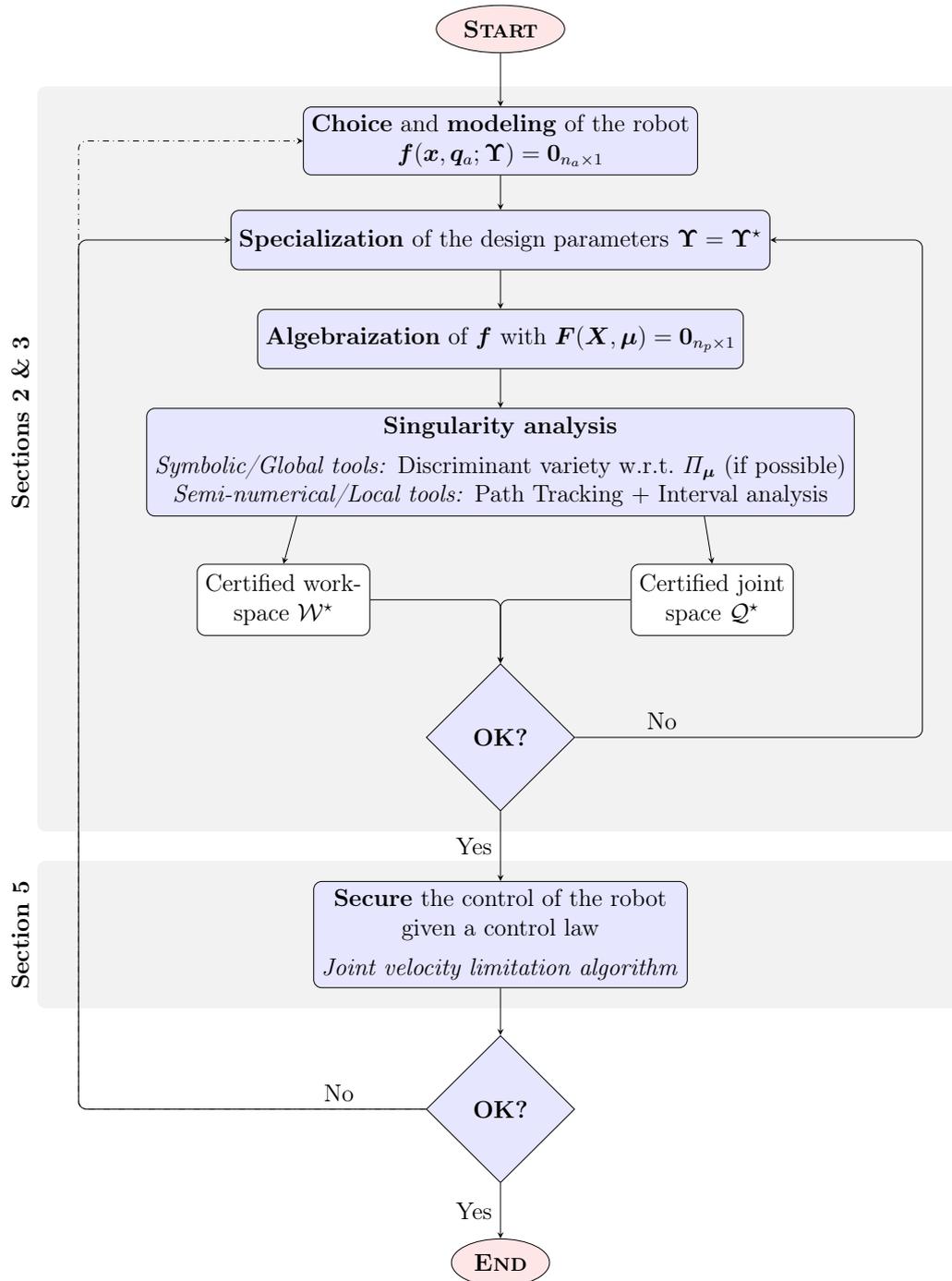


Distributed under a Creative Commons Attribution 4.0 International License

Graphical Abstract

Certified Kinematic Tools for the Design and Control of Parallel Robots

Alexandre Lê, Fabrice Rouillier, Guillaume Rance, Damien Chablat



Highlights

Certified Kinematic Tools for the Design and Control of Parallel Robots

Alexandre Lê, Fabrice Rouillier, Guillaume Rance, Damien Chablat

- Certifying the kinematics of parallel robots while accounting design uncertainties
- Using an appropriate combination of global and local strategies for the certification
- Improving the kinematic certification of robots using multiple precision computations
- Implementing a joint velocity algorithm to keep the robot in its certified workspace

Certified Kinematic Tools for the Design and Control of Parallel Robots

Alexandre Lê^{a,b,c}, Fabrice Rouillier^{b,c}, Guillaume Rance^a, Damien Chablat^d

^a*Safran Electronics & Defense, 100 avenue de Paris, Massy CEDEX, 91344, Île-de-France, France*

^b*Sorbonne Université, Université de Paris Cité, Institut de Mathématiques de Jussieu Paris Rive Gauche, 4 place Jussieu, Paris CEDEX, 75252, Île-de-France, France*

^c*Inria Paris, 48 rue Barrault, Paris, 75013, Île-de-France, France*

^d*Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, F-44000, 2 rue de la Houssinière, Nantes CEDEX 03, 44322, Pays de la Loire, France*

Abstract

This paper presents a methodology for the design and control of Parallel Kinematic Robots (PKRs). First, one focuses on the problematics of design. In particular, given a parallel mechanism defined by its design parameters and its kinematic modeling as well as its prescribed workspace, the idea is to certify the absence of any numerical instabilities (computational and physical singularities) that may jeopardize the integrity of the robot. This is achieved through two complementary approaches: a global method using symbolic computation and a local one based on continuation techniques and interval calculus, accounting for uncertainties in the design. The methodology is then applied to real PKR examples. Secondly, the paper proposes a control strategy that limits the active joint velocities to ensure the robot remains within its certified workspace. It will be applied to a special class of parallel robots: Spherical Parallel Manipulators (SPM) with coaxial input shafts (CoSPM).

Keywords:

Parallel robots, Design, Robot Control, Singularities, Computational Kinematics, Certification, Polynomial systems, Nonlinear systems, Symbolic computation, Interval arithmetic, Semi-numerical algorithms, Continuation methods, Joint limitation

1. Introduction & motivation

1.1. Context

Parallel robots [1, 2, 3] are closed-loop kinematic manipulators usually with two platforms (a base and an upper platform as the end-effector) that are linked by several kinematic chains. Each kinematic chain has joints that are either actuated (active joints generating the motions) or not (passive joints). The consecutive joints are linked by a body. Such an architecture bestows numerous advantages: these mechanisms are best known for presenting interesting performances in terms of dynamics, stiffness and accuracy. Moreover, parallel robots are mainly actuated at their base (the other joints being usually passive). As a result, many parts of such robots are subject to traction or compression constraints [4] making it

possible to use less powerful actuators and reduce the mass of the movable links, unlike their serial counterparts. Although parallel robots are less spread than the serial ones, they are gaining interest over the last decades in multiple domains: from the food industry for pick-and-place applications [5] (*e.g.* the Adept Quattro 4-DOF Delta robot) to the medical world (*e.g.* the SurgiScope 4-DOF Delta robot), through the simulated environments (*e.g.* the Rexroth 6-DOF hexapod dedicated to flight motion simulations). Moreover, other potential applications arise such as the inertial stabilization [6] of cameras [7] or sighting devices in the naval environment for the defense industry [8]. The last example notably explores the use of 3-DOF Spherical Parallel Manipulators [9, 10, 11] (SPM) instead of the classical 2-DOF (serial) gimbal architectures [12, 13].

1.2. Challenges arising by Parallel Robotics

Despite all the before-mentioned advantages of parallel robots, the latter are also well-known for being complex systems. Various historical challenges [14] arose regarding these architectures: determining the singularity loci of closed-loop robots, solving their so-called Forward Kinematics while taking into account the effects of heterogeneous uncertainties (*e.g.* design parameters, measurement errors) on the control [15] of such devices given their inherent nonlinearities. One can also mention studies on the performance evaluation of PKRs from a kinematic [16, 17] or dynamic [18, 19] perspective. These studies aim to provide a physical understanding of PKR behavior, specifically regarding velocity transmission from actuators to the end-effector and the motion capabilities of the end-effector but their use may be questioned [20, 21], for several reasons that will be explained later in this paper.

1.3. Organization of the paper

This paper is devoted to the kinematic analysis of PKRs and is organized as follows:

- *Basic notions and notations.* Section 2 recalls the basic notions and the appropriate notations required for the kinematic analysis of PKRs while reviewing the usual methods and tools used by roboticists to investigate the latter, along with their limitations.
- *Tools for the Design.* Section 3 introduces some certified kinematic tools and the mathematical background behind them. Detailed examples of real-case parallel robots (3-DOF 3-RPS tripods, 3-DOF 3-RRR SPMs) will be given in Section 4.
- *Tools for the Control.* Section 5 explains a strategy to secure the control of a special case of parallel robots, Spherical Parallel Manipulators with coaxial input shafts (CoSPM). One ensures that the robot of interest never leaves the safe regions delimited by the tools previously introduced in Section 3, given the physical characteristics of its actuators. Results can however be generalized to other parallel manipulators.

2. Kinematic analysis of parallel robots

2.1. Kinematic modeling

In the sequel, n_a denotes the number of actuators of a parallel kinematic robot (PKR) and n_{dof} its number of DOF. A parallel robot can be algebraically described by its *geometric*

model, namely a nonlinear system \mathbf{f} of n_f independent equations depending on the operational coordinates $\mathbf{x} \in \mathbb{R}^{n_{\text{dof}}}$ of the PKR (also called pose), the active joint values $\mathbf{q}_a \in \mathbb{R}^{n_a}$ as well as its design parameters $\Upsilon \in \mathbb{R}^{n_c}$ in an implicit manner, *i.e.*

$$\mathbf{f}(\mathbf{x}, \mathbf{q}_a; \Upsilon) := \begin{bmatrix} f_1(x_1, \dots, x_{n_{\text{dof}}}, q_{a,1}, \dots, q_{a,n_a}; \Upsilon_1, \dots, \Upsilon_{n_c}) \\ f_2(x_1, \dots, x_{n_{\text{dof}}}, q_{a,1}, \dots, q_{a,n_a}; \Upsilon_1, \dots, \Upsilon_{n_c}) \\ \vdots \\ f_{n_f}(x_1, \dots, x_{n_{\text{dof}}}, q_{a,1}, \dots, q_{a,n_a}; \Upsilon_1, \dots, \Upsilon_{n_c}) \end{bmatrix} = \mathbf{0}_{n_f \times 1}. \quad (1)$$

Such a system describes the input/output relationships between the base and the end-effector of the robot. It is composed of:

- $n_f = n_a = n_{\text{dof}}$ independent equations in the case of *nonredundant* robots;
- $n_f = n_{\text{dof}}$ independent equations for *kinematic redundant* [22] robots;
- $n_f = n_a$ equations with $n_{\text{dof}} < n_a$ independent ones for robots with *actuation redundancy* [23].

In the sequel, one will focus the nonredundant case. The system (1) can be obtained through the kinematic closure of the PKR of interest. A configuration \mathfrak{C}_0 of a robot is given by a pair $(\mathbf{x}_0, \mathbf{q}_{a,0})$. In fact, all the possible configurations of a robot given its general geometric model can be described using the *variety* \mathcal{U} defined by

$$\mathcal{U} := \{\mathfrak{C} = (\mathbf{x}, \mathbf{q}_a) \in \mathbb{R}^{n_{\text{dof}} + n_a} : \mathbf{f}(\mathbf{x}, \mathbf{q}_a; \Upsilon) = \mathbf{0}_{n_a \times 1}\}. \quad (2)$$

Let $\Pi_{\mathbf{q}_a}$ and $\Pi_{\mathbf{x}}$ be respectively the projection operator onto the \mathbf{q}_a - and \mathbf{x} -spaces. Given a known joint vector $\mathbf{q}_{a,0} \in \Pi_{\mathbf{q}_a}(\mathcal{U})$, there possibly exists several solutions to the so-called *Forward Kinematic Problem* (FKP) defined by the zero-dimensional system $\mathbf{f}(\mathbf{q}_{a,0}, \mathbf{x}) = \mathbf{0}$. Conversely, given a known operational vector $\mathbf{x}_0 \in \Pi_{\mathbf{x}}(\mathcal{U})$, there possibly exists several solutions to the so-called *Inverse Kinematic Problem* (IKP) defined by the zero-dimensional system $\mathbf{f}(\mathbf{q}_a, \mathbf{x}_0) = \mathbf{0}$. One then sets an initial configuration $\mathfrak{C}_{\text{init}} = (\mathbf{x}_{\text{init}}, \mathbf{q}_{a,\text{init}}) \in \mathcal{U}^* \subset \mathcal{U}$ and suppose that \mathcal{U}^* is a simply connected set that does not contain any singularity of \mathcal{U} nor any critical point of $\Pi_{\mathbf{q}_a}$ and $\Pi_{\mathbf{x}}$. Then, for any $\mathbf{q}_{a,0} \in \Pi_{\mathbf{q}_a}(\mathcal{U}^*)$ (*resp.* $\mathbf{x}_0 \in \Pi_{\mathbf{x}}(\mathcal{U}^*)$) one defines \mathbf{x}_0 (*resp.* $\mathbf{q}_{a,0}$) such that $(\mathbf{x}_0, \mathbf{q}_{a,0}) \in \mathcal{U}^*$ as being “the” solution of the FKP, *i.e.* $\mathbf{x}_0 = \text{fkp}(\mathbf{q}_{a,0})$ (*resp.* IKP, *i.e.* $\mathbf{q}_{a,0} = \text{ikp}(\mathbf{x}_0)$) accordingly to the implicit function theorem. It is worth stressing that the notion of *workspace* must be viewed as the projection of \mathcal{U} onto the \mathbf{x} -space, *i.e.* $\Pi_{\mathbf{x}}(\mathcal{U}) = \mathcal{W}$. Moreover, the notion of *joint space* must also be understood as the projection of \mathcal{U} onto the \mathbf{q}_a -space, *i.e.* $\Pi_{\mathbf{q}_a}(\mathcal{U}) = \mathcal{Q}$. Under such conditions, Figure 1 summarizes the relationships between the Forward, Inverse Problem from the geometric model viewpoint for any robot.

Remark 1. *In the robotics terminology, solutions to the FKP are called assembly modes and those of the IKP are called working modes [4]. Moreover, parallel robots may have several types of motion called operation modes [24, 25].*

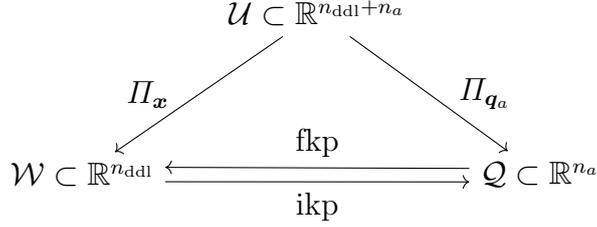


Figure 1: Links between the Forward, Inverse Problem of the geometric model

Differentiating the geometric model (1) w.r.t. time provides the *first order kinematic model* defined by

$$\dot{\mathbf{f}}(\mathbf{x}, \mathbf{q}_a; \Upsilon) = \mathbf{J}_x(\mathbf{x}, \mathbf{q}_a; \Upsilon) \dot{\mathbf{x}} + \mathbf{J}_{q_a}(\mathbf{x}, \mathbf{q}_a; \Upsilon) \dot{\mathbf{q}}_a = \mathbf{0}_{n_a \times 1}, \quad (3)$$

where, for the sake of simplicity in the notations, one defines and denotes the *parallel jacobian* as $\mathbf{J}_x := \frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ and the *serial jacobian* as $\mathbf{J}_{q_a} := \frac{\partial \mathbf{f}}{\partial \mathbf{q}_a}$. Suppose that both matrices are invertible. Then, as $\dot{\mathbf{x}}$ denotes the time variation of the operational coordinates and $\dot{\mathbf{q}}_a$ the vector of the active joint rates, one can respectively define the first order forward and inverse kinematic problems with the relationships

$$\dot{\mathbf{x}} = -\mathbf{J}_x^{-1}(\mathbf{x}, \mathbf{q}_a; \Upsilon) \mathbf{J}_{q_a}(\mathbf{x}, \mathbf{q}_a; \Upsilon) \dot{\mathbf{q}}_a = \mathbf{J}(\mathbf{x}, \mathbf{q}_a; \Upsilon) \dot{\mathbf{q}}_a, \quad (4a)$$

$$\dot{\mathbf{q}}_a = -\mathbf{J}_{q_a}^{-1}(\mathbf{x}, \mathbf{q}_a; \Upsilon) \mathbf{J}_x(\mathbf{x}, \mathbf{q}_a; \Upsilon) \dot{\mathbf{x}} = \mathbf{J}^{-1}(\mathbf{x}, \mathbf{q}_a; \Upsilon) \dot{\mathbf{x}}. \quad (4b)$$

From the control point of view, it is sometimes better to express the first order kinematic model (4a) in terms of the *twist* coordinates of the upper platform \mathbf{t}_* (giving its instantaneous velocity) instead of the operational angular rates $\dot{\mathbf{x}}$. Such a formalism is called *Instantaneous First Order Kinematic Model* [2]. As soon as at least two rotational DOFs are involved, one inevitably has $\mathbf{t}_* \neq \dot{\mathbf{x}}$. The two values are however linked by $\mathbf{t}_* = \Xi(\mathbf{x}) \dot{\mathbf{x}}$, where $\Xi(\mathbf{x})$ denotes the *parametrization matrix* whose expression depends on the frame being used and the choice of the operational vector \mathbf{x} for the description of the motion of the upper platform.

Remark 2. *From now on, unless it is necessary, the implicit dependence on the design parameters Υ will be omitted for the sake of readability.*

Studying the nonlinear geometric model in its original form (1) by solving the IKP or FKP is delicate and tedious in many cases. A convenient formalism to overcome these difficulties is the consideration of a *polynomial system* \mathbf{F} associated with \mathbf{f} : such a mathematical object can be obtained for most of the robots through appropriate changes of variables, *e.g.*:

1. half-angle tangent formulas $x_i \mapsto X_i$ such that $X_i := \tan\left(\frac{x_i}{2}\right)$, so that $\cos(x_i) = \frac{1-X_i^2}{1+X_i^2}$ and $\sin(x_i) = \frac{2X_i}{1+X_i^2}$,
2. sine/cosine changes of variables $x_i \mapsto (c_{x_i}, s_{x_i})$ such that $c_{x_i} := \cos(x_i)$ and $s_{x_i} := \sin(x_i)$ with $c_i^2 + s_i^2 - 1 = 0$.

While the first transformation has the advantage of keeping the same number of parameters, unknowns and equations unlike the second one that doubles the latter, it is not defined for $x_i = \pi \bmod 2\pi$ as $x_i \mapsto \tan(x_i/2)$ diverges. The second transformation is however well-defined for all real values x_i . The user should therefore pay attention to the prescribed workspace $\mathcal{W}^* \subseteq \mathcal{W}$ and associated desired joint space $\mathcal{Q}^* \subseteq \mathcal{Q}$ of the robot of interest that affects the choice of the polynomial transformation: a rough overview of \mathcal{W}^* and \mathcal{Q}^* is critical to do so. There also exists other algebraic formulations to describe the rotational and/or translational motion of rigid bodies with *unit* [26] or *dual* [27] *quaternions* for instance. In the sequel, one denotes the vector of the n unknowns of \mathbf{F} as $\mathbf{X} := (X_1, \dots, X_n)$ and the d parameters of \mathbf{F} as $\boldsymbol{\mu} := (\mu_1, \dots, \mu_d)$ so that \mathbf{F} reads

$$\mathbf{F}(\mathbf{X}, \boldsymbol{\mu}) := \begin{bmatrix} F_1(X_1, \dots, X_n, \mu_1, \dots, \mu_d) \\ F_2(X_1, \dots, X_n, \mu_1, \dots, \mu_d) \\ \vdots \\ F_{n_p}(X_1, \dots, X_n, \mu_1, \dots, \mu_d) \end{bmatrix} = \mathbf{0}_{n_p \times 1}, \quad (5)$$

where n_p is the number of the polynomial equalities of \mathbf{F} ($n_p = n_a$ for the half-angle tangent transformation and $n_p = 2n_a$ for the sine/cosine transformation). In the case of the FKP (*resp.* IKP), \mathbf{X} are related to the operational values \mathbf{x} (*resp.* active joint values \mathbf{q}_a). A fundamental assumption is used in order to *solve* the FKP or IKP: the polynomial system \mathbf{F} has a finite number of complex solutions described by the *algebraic variety* \mathcal{V} , *i.e.* for almost all d -uples $\boldsymbol{\mu}^* := (\mu_1^*, \dots, \mu_d^*) \in \mathbb{C}^d$, the system $\mathbf{F}(\mathbf{X}, \boldsymbol{\mu} = \boldsymbol{\mu}^*) = \mathbf{0}$ has finitely many complex solutions. In other words, such a system is *generically zero dimensional*. Solving the FKP or the IKP in its polynomial form can be done using *Algebraic Geometry* techniques. The reader may refer to [28] and [29] for further information, especially on the notions of *algebraic variety* and *ideal* whose details are not recalled in this paper. The main idea is to associate $\mathbf{F}(\mathbf{X}, \boldsymbol{\mu}) = \mathbf{0}$ with $\mathfrak{J} = \langle F_1, \dots, F_{n_p} \rangle$ being the *ideal* of $\mathbb{Q}[\boldsymbol{\mu}, \mathbf{X}]$ generated by the polynomials F_1, \dots, F_{n_p} such that $\overline{\Pi_{\boldsymbol{\mu}}(\mathcal{V}(\mathfrak{J}))} = \mathbb{C}^d$, where $\Pi_{\boldsymbol{\mu}}$ denotes the projection onto the polynomial parameter space and $\overline{\mathcal{V}}$ the algebraic closure of any subset $\mathcal{V} \in \mathbb{C}^d$. As a result, the complex solutions of $\mathbf{F}(\mathbf{X}, \boldsymbol{\mu}) = \mathbf{0}$ define the *algebraic variety* $\mathcal{V}(\mathfrak{J})$.

2.2. Singularity analysis

The singularities of a robot are configurations where the latter does not behave properly: its number of Degree Of Freedom (DOF) differs from the nominal value n_{dof} . There are several types of parallel robot singularities, namely input/output (I/O) singularities [30], “serial” singularities (LAJTS and LPJTS) [4, 31], and constraints singularities [24] for lower-mobility¹ parallel robots. This paper focuses on the I/O singularities. From the geometric model viewpoint, I/O singularities appear when:

- the IKP has at least two *identical* solutions (working modes): these configurations are called *Type 1 singularities*;

¹A robot is said to have a *full mobility* if it has 3 DOFs in rotation and 3 DOFs in translation (6 in total). Parallel robots with a *lower mobility* verify $n_{\text{dof}} < 6$.

- the FKP has at least two *identical* solutions (assembly modes): these configurations are called *Type 2 singularities*.

Configurations being simultaneously Type 1 and Type 2 singular are called *Type 3 singularities*. From the first order kinematic viewpoint, Type 1 singularity occurs when the matrix $\mathbf{J}_{\mathbf{q}_a}$ defined in (3) is rank-deficient. The inverse problem (4b) is singular: the PKR loses at least one DOF and the efforts required to move the end-effector in the neighborhood of such areas are substantial for a small displacement. Conversely, Type 2 singularity happens when the matrix $\mathbf{J}_{\mathbf{x}}$ defined in (3) is rank-deficient. The forward problem (4a) is singular: the PKR gains at least one uncontrollable DOF, a small variation of the joint values generates a huge displacement of the end-effector in the neighborhood of such areas. In some cases, the robot may have infinitely many possible poses \mathbf{x} for the same active joint variables $\mathbf{q}_{a,0}$: these Type 2 configurations are called *self-motions* [32, 33]. These phenomena are amplified by the uncertainties of the PKR that are inherent (design parameters, assembling tolerances of the mechanical parts [2]) or external (measurements) and may jeopardize the integrity of the mechanism in the worst cases. Figure 2 illustrates the notions of I/O singularities, showing the Five-Bar mechanism (a RRRRR planar parallel robot) as well as its I/O singularity loci projected onto its operational space: an (almost) Type 1 configuration is depicted in Fig. 2a and an (almost) Type 2 one is presented in Fig. 2b.

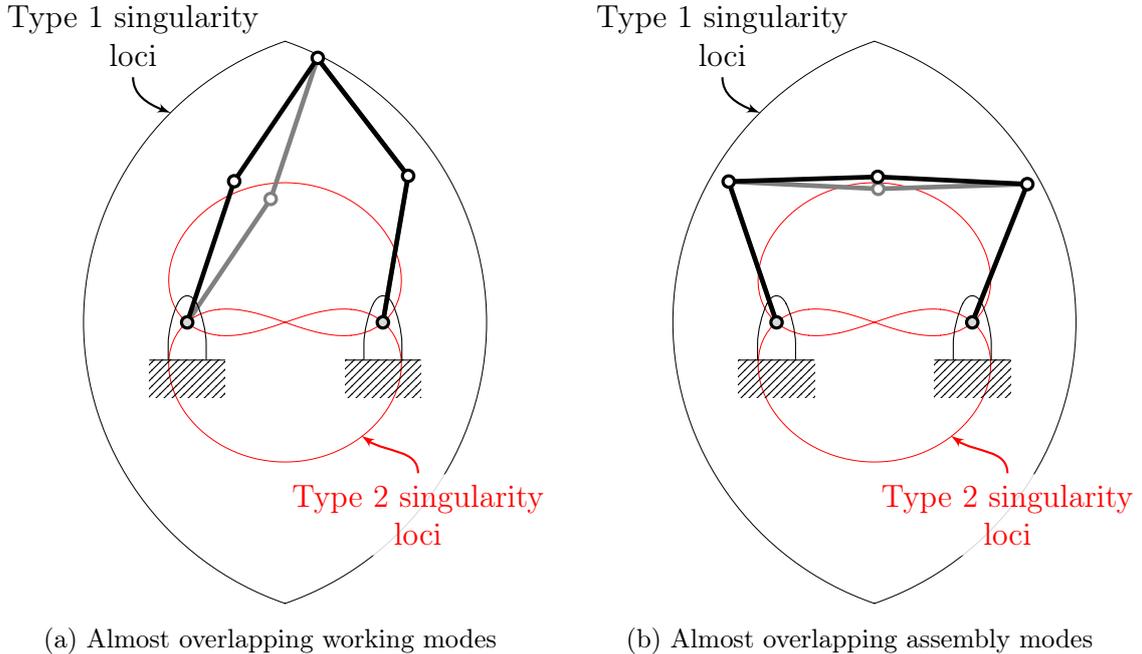


Figure 2: Illustration of I/O singularities on a RRRRR Five-Bar Mechanism

Theoretically speaking, one can investigate I/O singularities by computing the determinant of the appropriate jacobian matrices: $\det(\mathbf{J}_{\mathbf{q}_a}(\mathbf{x}, \mathbf{q}_a)) = 0$ for the inverse problem and $\det(\mathbf{J}_{\mathbf{x}}(\mathbf{x}, \mathbf{q}_a)) = 0$ for the forward one (given that $(\mathbf{x}, \mathbf{q}_a) \in \mathcal{U}$, with the previous hypothesis made in Subsection 2.1). Such a work is however challenging as these expressions are

tedious or hard to compute and exploit in most cases. In order to investigate the singularity loci of robots, numerical-based methods bypassing the before-mentioned difficulties arose with the so-called *conditioning index* of a jacobian matrix. In its local formulation, *i.e.* at a given configuration $\mathfrak{C}_0 = (\mathbf{x}_0, \mathbf{q}_{a,0})$, the (local) conditioning index [2] of \mathbf{J} denoted as $\text{cond}_p^{-1}(\mathbf{J}(\mathfrak{C}_0))$ is defined as

$$\text{cond}_p^{-1}(\mathbf{J}(\mathfrak{C}_0)) = \frac{1}{\|\mathbf{J}(\mathfrak{C}_0)\|_p \|\mathbf{J}^{-1}(\mathfrak{C}_0)\|_p}, \quad (6)$$

where $\|\cdot\|_p$ is the matrix norm taken into consideration (for the sake of simpler computations [34], roboticists mostly use a weighted² Frobenius norm). Given that $0 \leq \text{cond}_p^{-1}(\mathbf{J}(\mathfrak{C}_0)) \leq 1$, \mathfrak{C}_0 describes a singular configuration if $\text{cond}_p^{-1}(\mathbf{J}(\mathfrak{C}_0)) \sim 0$ and is an *isotropic* one if $\text{cond}_p^{-1}(\mathbf{J}(\mathfrak{C}_0)) \sim 1$. A global formulation of this index was introduced in [35, 16]: the idea is to evaluate the local index over the whole prescribed workspace \mathcal{W}^* . This method attempts to take into account all the possible configurations \mathfrak{C} of interest by computing the mean value of the local index over \mathcal{W}^* . The resulting value is then called *global conditioning index*. Despite the practicality of the method, several drawbacks arise:

- Such indices are dependent on the choice of the matrix norm [20] as shown in (6): results differs from one norm to another and may even not be coherent with themselves (see [20] for examples). In addition, real significance of these indices (dexterity, manipulability or accuracy of the robot) is not always well understood, especially for robots with both translational and rotational DOFs.
- Computations are almost always made based on a discretization of the domain of interest: the obtained results are extremely sensitive to the step size of discretization. In the worst cases, it may even miss low-conditioned configurations and skew the computed global index value.
- Even if discretization issues are put aside, these criteria are subject to the *reliability* [36] of their numerical computations that may be jeopardized by numerical instabilities or round-off errors. A famous example of this assertion is the Rump's function [37] evaluated at a certain value: computations made using several languages or tools with floating point numbers all returned wrong and very different results: they are not *certified*.

The purpose of this paper is to provide certified tools related to the kinematic analysis of parallel robots so that one can ensure (or not) that the robot under study and given its application will not meet any physical instabilities (singularities) nor numerical ones. More precisely, the novelty and main contribution of the kinematic certification methodology

²by the number of actuators n_a through a matrix $\mathbf{W} = \frac{1}{n_a} \mathbf{I}_{n_a}$ so that the resulting weighted Frobenius norm of \mathbf{J} writes $\|\mathbf{J}\|_{\mathbf{F}, \mathbf{W}} = \sqrt{\text{Tr}(\mathbf{J}^T \mathbf{W} \mathbf{J})}$

both rely on an *appropriate combination* of symbolic and numerical tools as well as their *implementations*. Symbolic tools include the notion of discriminant variety that will be recalled in this article and numerical tools involve continuation methods [38] coupled with interval analysis and multiple precision computations.

3. Tools for the design of parallel robots

The tools presented in this section are based on two different but complementary approaches:

- (i) a *global* strategy (detailed in Subsection 3.1) in which one studies $\mathbf{F}(\mathbf{X}, \boldsymbol{\mu})$ as a *parametric system* that is generically zero-dimensional;
- (ii) a *local* one (detailed in Subsection 3.2) involving continuation methods and interval calculus to solve *zero-dimensional systems* $\mathbf{F}(\mathbf{X}, \boldsymbol{\mu} = \boldsymbol{\mu}^*)$ in a certified manner.

3.1. Discriminant variety

The previous remarks on Type 1 and Type 2 singularities being the superposition of working modes (solutions of the IKP) and assembly modes (solutions to the FKP) raise the following statement: a nonsingular zone in the $\boldsymbol{\mu}$ -parameter space is free of any singularity or numerical instability if the system \mathbf{F} has a constant number of real distinct solutions \mathbf{X}^* satisfying $\mathbf{F}(\mathbf{X}^*, \boldsymbol{\mu}) = \mathbf{0}$, where the parameter vector $\boldsymbol{\mu}$ is left free. The idea is then to provide a partition of the parameter space into regions (subsets) in which the parametric system $\mathbf{F}(\mathbf{X}, \boldsymbol{\mu})$ has a constant number of real solutions. This partition relies on the description of an algebraic variety (of \mathbb{C}^d) called the *discriminant variety* w.r.t. the projection onto the parameter space.

3.1.1. Definition

The notion of discriminant variety was initially established in [39] to describe all the problematic areas of a generically zero-dimensional system projected onto its parameter space. This paper however presents another definition (from [40]) adapted for the study of robotic systems.

Definition 1 (Discriminant Variety [40]). *The (minimal) discriminant variety of $\mathcal{V}(\mathfrak{J})$ w.r.t. $\Pi_{\boldsymbol{\mu}}$ denoted as $\mathcal{W}_D(\mathbf{F})$ is the (smallest) algebraic variety of \mathbb{C}^d such that given any simply connected subset \mathcal{C} of $\mathbb{R}^d \setminus \mathcal{W}_D(\mathbf{F})$, the number of real solutions of \mathbf{F} is constant over $\boldsymbol{\mu}$. In our case,*

$$\mathcal{W}_D(\mathbf{F}) := \mathcal{W}_{\text{sd}}(\mathbf{F}) \cup \mathcal{W}_c(\mathbf{F}) \cup \mathcal{W}_{\infty}(\mathbf{F})$$

where:

- $\mathcal{W}_{\text{sd}}(\mathbf{F})$ is the closure of the projection by $\Pi_{\boldsymbol{\mu}}$ of the components of $\mathcal{V}(\mathfrak{J})$ of dimension less than d ;
- $\mathcal{W}_c(\mathbf{F})$ is the union of the closure of the critical values of $\Pi_{\boldsymbol{\mu}}$ in restriction to $\mathcal{V}(\mathfrak{J})$ and of the projection of singular values of $\mathcal{V}(\mathfrak{J})$;

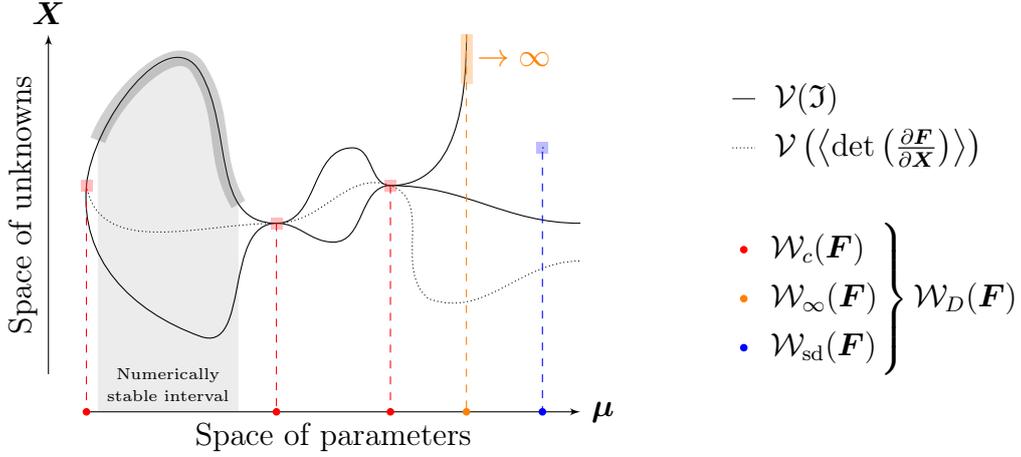


Figure 3: Certification by avoiding the discriminant variety \mathcal{W}_D w.r.t. the projection onto the parameter space

- $\mathcal{W}_\infty(\mathbf{F})$ is the set of $\boldsymbol{\mu}$ such that $\Pi_\mu^{-1}(\mathcal{C}) \cap \mathcal{V}(\mathcal{J})$ is not compact for any compact neighborhood \mathcal{C} of $\boldsymbol{\mu}$ in $\Pi_\mu(\mathcal{V}(\mathcal{J}))$.

Remark 3. Note that following the previous notations and Definition 1, algebraic systems with as many polynomials as unknowns always verify

$$\mathcal{W}_{sd}(\mathbf{F}(X_1, \dots, X_n; \boldsymbol{\mu}) = \mathbf{0}_{1 \times n}) = \emptyset. \quad (7)$$

Figure 3 illustrates the notion of the so-called discriminant variety that divides the parameter space into subsets where the model \mathbf{F} is exempt of any source of instability. The algebraic viewpoint distinguishes the notion of *singular* points of \mathbf{F} that are intrinsic to the system and the *critical* points of \mathbf{F} depending on the projection being used (and thus, the choice of the polynomial parameters $\boldsymbol{\mu}$ depending on the IKP or the FKP). The Type 1 (*resp.* Type 2) singularities are then the intersection of the *singular* points of \mathbf{F} and its *critical* points w.r.t. the projection onto the polynomial operational (*resp.* active joint) space and Type 3 singularities can be viewed as the singular points of \mathbf{F} .

3.1.2. Computation of the discriminant variety

In the general case, the discriminant variety of a parametric system $\mathbf{F}(\mathbf{X}, \boldsymbol{\mu}) = \mathbf{0}$ is computed using Gröbner basis techniques [39, 41]. The main idea is to compute a Gröbner basis of \mathcal{J} for a monomial ordering eliminating all the unknowns X_1, \dots, X_n . An efficient monomial ordering to do so is the product order $>_{\boldsymbol{\mu}, \mathbf{X}}$. The function `DiscriminantVariety` implemented in MAPLE computes the discriminant variety w.r.t. the projection onto the parameter space of the system of interest as long as its polynomials are square-free, generically solvable ($n_p \geq n$) and generically zero-dimensional. Such a function is also called in the SiRoPa MAPLE library [40] dedicated to the study of serial and parallel manipulators using symbolic computations.

There are however some simpler cases in which Gröbner basis are not mandatory, namely for the Inverse Kinematics of most PKRs. Indeed, as they mostly only have one actuated joint per leg (joint values are usually *decoupled*), the algebraic IKP of a PKR is usually a system of univariate polynomials (w.r.t. the joint variables). In such a case, studying the discriminant variety of the IKP denoted as $\mathcal{W}_{\text{ikp}}(\mathbf{F})$ w.r.t. the projection onto the (operational) parameter space means computing the *resultant* of each polynomial $F_i, \frac{\partial F_i}{\partial X_i}$ w.r.t. X_i as we only need to eliminate one variable X_i per polynomial F_i . By denoting $\phi_i = \phi$ the degree of F_i w.r.t. X_i , one has

$$\begin{aligned} \mathcal{W}_{\text{ikp}}(\mathbf{F}) &= \bigcup_{i=1}^{n_p} \text{Res} \left(F_i, \frac{\partial F_i}{\partial X_i}, X_i \right), \\ &= \bigcup_{i=1}^{n_p} \left\{ (-1)^{\frac{\phi(\phi-1)}{2}} \text{lc}(F_i, X_i) \text{discrim}(F_i, X_i) \right\}, \end{aligned} \quad (8)$$

where $\text{discrim}(F_i, X_i)$ is the *discriminant* of F_i w.r.t. X_i and $\text{lc}(F_i, X_i)$ the *leading coefficient* of $F_i(X_i)$. In this particular case, the hypersurface described by $\text{lc}(F_i, X_i) = 0$ represents the projection of the infinity points onto the parameter space *i.e.* $\mathcal{W}_{\infty}(\mathbf{F})$ whereas $\text{discrim}(F_i, X_i) = 0$ is the projection of the singular and critical points of \mathbf{F} onto the parameter space, *i.e.* $\mathcal{W}_c(\mathbf{F})$. The latter set then encodes the Type 1 singularity loci of the PKR.

3.1.3. Limitations

On the computation. Computing the discriminant variety may be challenging in some cases especially for polynomial systems with high degrees. In this case, the object simply becomes too substantial to exploit or compute as it relies on Gröbner basis even though it is *theoretically* possible to do so. Moroz [42] gave an upper bound on the total degree of the minimal discriminant variety of a parametric system: its total degree D is bounded by

$$D \leq \prod_{i=1}^{n_p} d_i \left[1 + \sum_{i=1}^{n_p} (d_i - 1) \right], \quad (9)$$

where d_i is the total degree of F_i , $i \in \llbracket 1, n_p \rrbracket$. Moreover, its complexity is at least exponential making it not suitable for polynomial systems with a high total degree.

On the results. Although the discriminant variety provides an exact expression of all the regions to avoid in the parameter space, it is only computed for parametric systems of robots whose design parameters is fixed. Consequently, it cannot take into account the uncertainties of the mechanism (namely on its design parameters Υ).

In order to overcome these limitations, the idea is to also rely on a semi-numerical strategy involving the certification of continuation methods taking into account the deformations of the system due to the before-mentioned uncertainties. Such a work is treated through the *Path tracking* problem [2] which is detailed in the sequel.

3.2. Certified path tracking

Given a zero-dimensional polynomial system $\mathbf{F}(\mathbf{X}, \boldsymbol{\mu}_0) = \mathbf{0}$ where $\boldsymbol{\mu}_0$ is specified, the purpose is to guaranty:

- the existence and *uniqueness* of the solution \mathbf{X}^* such that $\mathbf{F}(\mathbf{X}^*, \boldsymbol{\mu}_0) = \mathbf{0}$ in the zone of interest (absence of physical singularity);
- the *convergence* of numerical continuation methods towards it, with the desired precision (numerical stability).

There are several continuation methods such as the Newton scheme [43] or the homotopy continuation method [44, 38, 45] as well as several unicity operators to certify them as the Kantorovich [46] or the Smale [47] one for instance. In this article, the certification of the continuation strategy for the path tracking will be done under the *Kantorovich unicity operator* framework using a Newton scheme. The methodology can however be adapted to any appropriate choice of continuations methods and unicity operators.

3.2.1. Kantorovich unicity operator

There exists several formulations of the Newton-Kantorovich theorem: from strong [46, 48, 49] to weak formulations [50, 51, 52]. The latter mainly relies on the Lipschitz behavior of the all the functions $f_i \in \mathbf{f}$ by computing the eponymous constant. The strong formulation is however more suitable for the implementation of the path tracking and will be used throughout this paper. Before introducing the theorem, let us recall the following definitions. The ℓ^∞ -norm on \mathbb{R}^n is defined for a vector $\mathbf{x} \in \mathbb{R}^n$ as $\|\mathbf{x}\|_\infty := \max_{i \in \llbracket 1, n \rrbracket} |x_i|$ while the induced matricial counterpart on $\mathbb{R}^{n \times n}$ is defined for a matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ with coefficients M_{ij} as $\|\mathbf{M}\|_\infty := \max_{i \in \llbracket 1, n \rrbracket} \sum_{j=1}^n |M_{ij}|$.

Theorem 1 (Newton-Kantorovich [46, 49]). *Let $\mathbf{f} : \mathcal{D} \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ a vector function of class \mathcal{C}^2 . Let $\mathbf{x}_0 \in \mathcal{D}$, $\bar{\mathcal{U}}_\infty(\mathbf{x}_0, \mathfrak{H})$ its \mathfrak{H} -closed neighborhood defined by $\bar{\mathcal{U}}_\infty(\mathbf{x}_0, \mathfrak{H}) := \{\mathbf{x} \in \mathcal{D} : \|\mathbf{x} - \mathbf{x}_0\|_\infty \leq \mathfrak{H}\}$ and $\mathbf{J}_0 := \partial \mathbf{f} / \partial \mathbf{x}|_{\mathbf{x}=\mathbf{x}_0}$ be the invertible jacobian matrix w.r.t. the variables \mathbf{x} evaluated at \mathbf{x}_0 . If there exists three real constants A_0 , B_0 and C such that:*

- (i) $\|\mathbf{J}_0^{-1}\|_\infty \leq A_0$,
- (ii) $\|\mathbf{J}_0^{-1} \mathbf{f}(\mathbf{x}_0)\|_\infty \leq B_0 \leq \mathfrak{H}/2$,
- (iii) $\forall i \in \llbracket 1, n \rrbracket, \forall j \in \llbracket 1, n \rrbracket$ and $\mathbf{x} \in \bar{\mathcal{U}}(\mathbf{x}_0, \mathfrak{H})$, $\sum_{k=1}^n \left| \frac{\partial^2 f_i(\mathbf{x})}{\partial x_j \partial x_k} \right| \leq C$,
- (iv) $\nu_0 := 2nA_0B_0C \leq 1$,

then there is a unique solution of $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ in $\bar{\mathcal{U}}(\mathbf{x}_0, 2B_0)$ and the (real) Newton iterative scheme $\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{J}^{-1}(\mathbf{x}_k) \mathbf{f}(\mathbf{x}_k)$ with the initial estimate \mathbf{x}_0 quadratically converges towards this unique solution.

This formulation requires \mathbf{f} to be at least of class \mathcal{C}^2 which the case as we only manipulate nonlinear equations in cosine or sine and polynomials. Furthermore, it uses a *sufficient but nonnecessary condition* on the regularity of a configuration and its neighborhood: having $\nu_0 > 1$ does not systematically mean that the robot is going towards a singularity. The next subsection details the implementation of the Kantorovich unicity operator for the Path Tracking problem as well as some strategies to favor a successful Kantorovich test.

3.2.2. Implementation of a certified continuation strategy

First, it is worth pointing out that although the Newton-Kantorovich theorem applies to any \mathcal{C}^2 vector function, one only considers polynomial systems $\mathbf{F}(\mathbf{X}, \boldsymbol{\mu} = \boldsymbol{\mu}_0)$, where \mathbf{F} is the vector function and \mathbf{X} the unknowns of interest. The constants A_0 , B_0 and C of Theorem 1 and the Newton's iterations are computed using *multiple precision interval arithmetic* [53]. Interval Arithmetic (IA) is a framework in which one manipulates (closed) *intervals* [54, 55] instead of floating-point numbers.

Definition 2 (Interval). An interval $[x] := [\underline{x}, \bar{x}]$ is the set of all real numbers x such that $\underline{x} \leq x \leq \bar{x}$.

Such a mathematical object is also defined by its *width* $\text{wid}([x]) := \bar{x} - \underline{x}$, its *midpoint* $\text{mid}([x]) := (\bar{x} + \underline{x})/2$ and its *radius* $\text{rad}([x]) := (\bar{x} - \underline{x})/2 = \text{wid}([x])/2$. The set of all intervals is denoted as \mathbb{IR} . The notion of intervals can also be extended to other objects. A *box* $[\mathbf{x}] = ([x_1], \dots, [x_n])$ is a vector of intervals and an *interval matrix* $[\mathbf{M}]$ is a matrix whose coefficients $[M_{ij}]$ are intervals. The set of all boxes of size n (*resp.* $(n \times m)$ interval matrices) is denoted as \mathbb{IR}^n (*resp.* $\mathbb{IR}^{n \times m}$). From the computational point of view, an important characteristic of an interval is its *precision*.

Definition 3 (Precision of an interval). Let $[x] \in \mathbb{IR}$ with bounds \underline{x} and \bar{x} of the form $m \cdot 2^e$. The interval $[x]$ is said to have a binary precision p if $|m| < 2^p$. This interval will be denoted in the sequel as $[x]_p$.

Example 1. The irrational number π (that cannot be represented exactly as a floating-point number) is included in an interval of precision of 3 digits, *i.e.* $p = 10$ bits as $[\pi]_{10} = [3.141, 3.142]$.

Intervals are a precious tool in the goal of certification: IA is a reliable formalism in the sense that any computed intervals contain the correct result of interest. Most importantly, using IA to study the Kinematics of a PKR provides undeniable benefits: uncertainties (*e.g.* on its design) and measurement errors (used for its control) can be taken into consideration. In addition, rounding errors are taken into account as computer implementations perform outward rounding [56]. Although these (numerous) advantages, it also suffers from drawbacks [57], namely *wrapping effect* (overestimation) and *data dependency* (sensitivity to the formulation of the expressions).

Some implementations of interval analysis have been made with C++ packages such as BIAS/PROFIL (Basic Interval Arithmetic Subroutines/Programmer's Runtime Optimized

Fast Interval Library) [58]. Based on BIAS/PROFIL, another library named ALIAS [59, 60] (which stands for Algorithms Library of Interval Analysis for equation Systems) was developed by Jean-Pierre Merlet using C++ and MAPLE. This library was devoted to solve systems of equations and inequalities, optimization, linear algebra and was for instance used in [61] to solve the FKP of a Gough-type parallel robot. ALIAS however uses a fixed-precision floating-point interval arithmetic which means that the computations only rely on the machine precision and can lead to accuracy issues (large intervals whose results are unusable in practice). A way to get rid of this problem is the consideration of *multiple precision* arithmetic [56].

It must be understood that using a multiple precision arithmetic means that the computing precision is not limited to the (single or double precision) machine precision. In particular, the number of digits of the mantissa can be any fixed or variable value which can be helpful for high accuracy computations [53]: loss of accuracy can be handled whether it is predictable or unpredictable (by bounding it). Among all the libraries implementing multiple precision arithmetic, one could for instance name Arb (Arbitrary-precision Ball arithmetic) [62] or MPFR (which means Multiple Precision Floating-point Reliable library) [63] dedicated to multiple precision floating-point computations with correct rounding. Moreover, Revol and Rouillier [56] introduced MPFI (standing for Multiple Precision Floating-point Interval library). The latter is based on MPFR and implements interval arithmetic on top of it. Both libraries whose source codes are available online are written in C. Moreover MPFI brings another feature: it has the advantage of minimizing the number of bits during the computational process (which is for instance suitable for real-time computations with low precisions) and will be chosen in the sequel for the implementation of the Kantorovich unicity operator. In the case of MPFI, intervals are implemented using their lower and upper bounds (also called *endpoints*) that are MPFR floating-point numbers. Arithmetic operations are then performed on these multiple precision endpoints so that the correct result is always contained in the output interval while controlling its precision.

Back to the Kantorovich unicity operator the Path Tracking works as follows. One sets a starting point $\boldsymbol{\mu}_0$ in the parameter space with the corresponding known solution \mathbf{X}_0 such that $\mathbf{F}(\mathbf{X}_0, \boldsymbol{\mu}_0) = \mathbf{0}$ and moves towards another point $\boldsymbol{\mu}_1$. One afterwards considers a family $\boldsymbol{\Sigma}_1$ of polynomial systems in the parameter space of the form $\mathbf{F}(\mathbf{X}, \boldsymbol{\mu} = \boldsymbol{\mu}_1)$ defined by

$$\boldsymbol{\Sigma}_1(\mathbf{s}) := \begin{bmatrix} F_1(\mathbf{X}, \boldsymbol{\mu} = \boldsymbol{\mu}_1; \mathbf{s}) \\ \vdots \\ F_{n_p}(\mathbf{X}, \boldsymbol{\mu} = \boldsymbol{\mu}_1; \mathbf{s}) \end{bmatrix} \quad (10)$$

such that $\forall j \in \llbracket 1, n_p \rrbracket$, each polynomial F_j writes

$$F_j(\mathbf{X}, \boldsymbol{\mu} = \boldsymbol{\mu}_1; \mathbf{s}) = \sum_{\alpha \in \mathbb{N}^n} [c_{\alpha,j}(\boldsymbol{\mu}_1)]_{\mathbf{s}} X^\alpha, \quad (11)$$

where $\mathbf{s} \in \mathbb{N}^*$ denotes the *system precision* of $\boldsymbol{\Sigma}_1$, that is, the precision of the interval coefficients of the monomials $X^\alpha = \prod_{i=1}^n X_i^{\alpha_i}$ ($\alpha_i \in \mathbb{N}, \forall i \in \llbracket 1, n \rrbracket$) for each polynomial F_j ,

$j \in \llbracket 1, n_p \rrbracket$. In fact, the system precision \mathbf{s} encodes all the desired uncertainties, namely those on the design parameters \mathbf{Y} of the mechanism of interest. The choice of its value is left to the robot designer, keeping in mind that it must be in accordance with the requirements of his/her specifications in terms of precision. The Kantorovich unicity operator is then applied to Σ_1 : $[A_0]$, $[B_0]$ are computed based on $\mathbf{F}(\mathbf{X}_0, \boldsymbol{\mu}_1; \mathbf{s})$ and its interval Jacobian $[\mathbf{J}_0]$ whereas $[C]$ is obtained through a Hessian criterion on $\mathbf{F}(\mathbf{X}, \boldsymbol{\mu}_1; \mathbf{s})$ evaluated over all the \mathfrak{H} -closed neighborhood of \mathbf{X}_0 , *i.e.* $\forall \mathbf{X} \in \overline{U}_\infty(\mathbf{X}_0, \mathfrak{H})$, given that $\mathfrak{H} \geq 2\overline{B}_0$. If the interval condition number $[\nu_0] = [\underline{\nu}_0, \overline{\nu}_0]$ verifies $\overline{\nu}_0 < 1$, the Kantorovich test is successful and guarantees the existence, unicity of a real solution \mathbf{X}_1 satisfying $\mathbf{F}(\mathbf{X}_1, \boldsymbol{\mu}_1) = \mathbf{0}$ and the quadratic convergence of continuation numerical methods [44] (Newton iterative scheme) towards it. In fact, the implemented Newton scheme also uses multiple precision interval arithmetic so that \mathbf{X}_1 is undoubtedly contained in an interval $[\mathbf{X}_1]$ that is either reached with the desired output precision \mathbf{f} , *i.e.* $[\mathbf{X}_1]_{\mathbf{f}}$ or not *i.e.* $[\mathbf{X}_1]_{\mathbf{f}'}$, with $\mathbf{f}' > \mathbf{f}$. All the computations are made under a certain *working precision* \mathbf{w} (highest among all the precisions considered). Note that using multiple precision allows us to distinguish between the numerical errors arising from the computations and the purely physical uncertainties of the robot. In this regard, multiple precision facilitates obtaining a successful Kantorovich test, unlike the usual double fixed precision which is more likely to accumulate the numerical errors so that the interval $[\nu_0]$ is widened with $\overline{\nu}_0 > 1$.

At the end of a successful Kantorovich test for Σ_1 , the parameter $\boldsymbol{\mu}_1$ (with the known solution $\mathbf{X}_1 \in [\mathbf{X}_1]$) becomes the new starting point for another displacement in the parameter space towards $\boldsymbol{\mu}_2$, a new Kantorovich test is then performed and so on, until the whole regular workspace \mathcal{W}^* is entirely covered with the unicity balls, in one way or another, so that the Kinematic is certified for the zone of interest. **Paving the workspace of interest certifies the appropriate behavior of weaker numerical methods such as a Newton scheme with fixed precision.**

Figure 4 illustrates the notion of path tracking as described above, applied to the certification of the Kinematics of PKRs. From the general point of view, given a parameter point $\boldsymbol{\mu}_k$ and at each Kantorovich test, the family $\Sigma_k(\mathbf{s})$ of systems taken into consideration is defined by

$$\Sigma_k(\mathbf{s}) := \begin{bmatrix} F_1(\mathbf{X}, \boldsymbol{\mu} = \boldsymbol{\mu}_k; \mathbf{s}) \\ \vdots \\ F_{n_p}(\mathbf{X}, \boldsymbol{\mu} = \boldsymbol{\mu}_k; \mathbf{s}) \end{bmatrix} \quad (12)$$

such that $\forall j \in \llbracket 1, n_p \rrbracket$, each polynomial F_j writes

$$F_j(\mathbf{X}, \boldsymbol{\mu} = \boldsymbol{\mu}_k; \mathbf{s}) = \sum_{\alpha \in \mathbb{N}^n} [c_{\alpha, j}(\boldsymbol{\mu}_k)]_{\mathbf{s}} X^\alpha. \quad (13)$$

It should be emphasized that $\Sigma_k(\mathbf{s})$ takes into account an infinite number of polynomial systems inside it, defining a neighborhood around the nominal system $\mathbf{F}(\mathbf{X}, \boldsymbol{\mu}_k)$ in the parameter space. Practically speaking, this amounts to consider an infinite number of robots that can be viewed as slightly deformed versions of the nominal one.

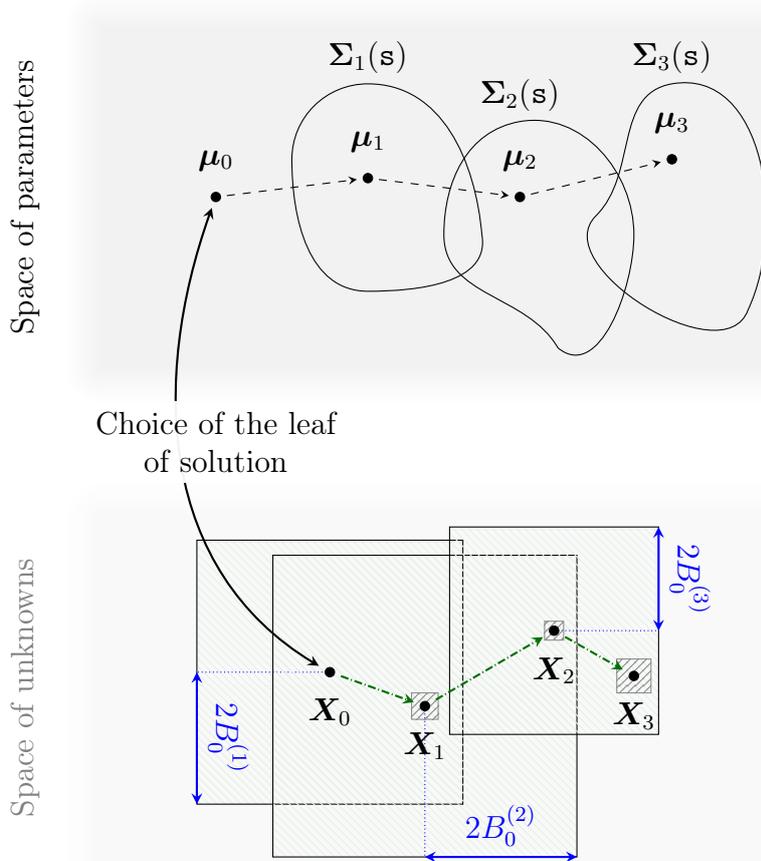


Figure 4: Path Tracking applied to the Forward or Inverse Kinematic Problem

In case of a failed Kantorovich test, *i.e.* if $\bar{\nu}_0 > 1$, the nonnecessary condition of the Newton-Kantorovich theorem allows us to reattempt another one. In order to favor a successful test, one could consider the following strategies:

- *decrease the displacement* in the parameter space which amounts to examine another system that is nearer to the starting point (see Fig. 5a);
- *increase the system precision* and thus reduce the size of the uncertainties taken into account (see Fig. 5b) which amounts to study a smaller family Σ of polynomials.

One could also run a few Newton's iteration before relaunching a new Kantorovich test. It is worth stressing that by defining a starting point μ_0 in the parameter space along with the corresponding known solution X_0 such that $F(X_0, \mu_0) = \mathbf{0}$, one sets the leaf of solution of interest. Therefore, certifying the workspace of interest using the Kantorovich unicity operator only means that this specific leaf of solution does not meet another one (and hence is singularity-free within the regions of interest). However, there is no guaranty that other leaves of solution are free of any numerical instability. Such information is however

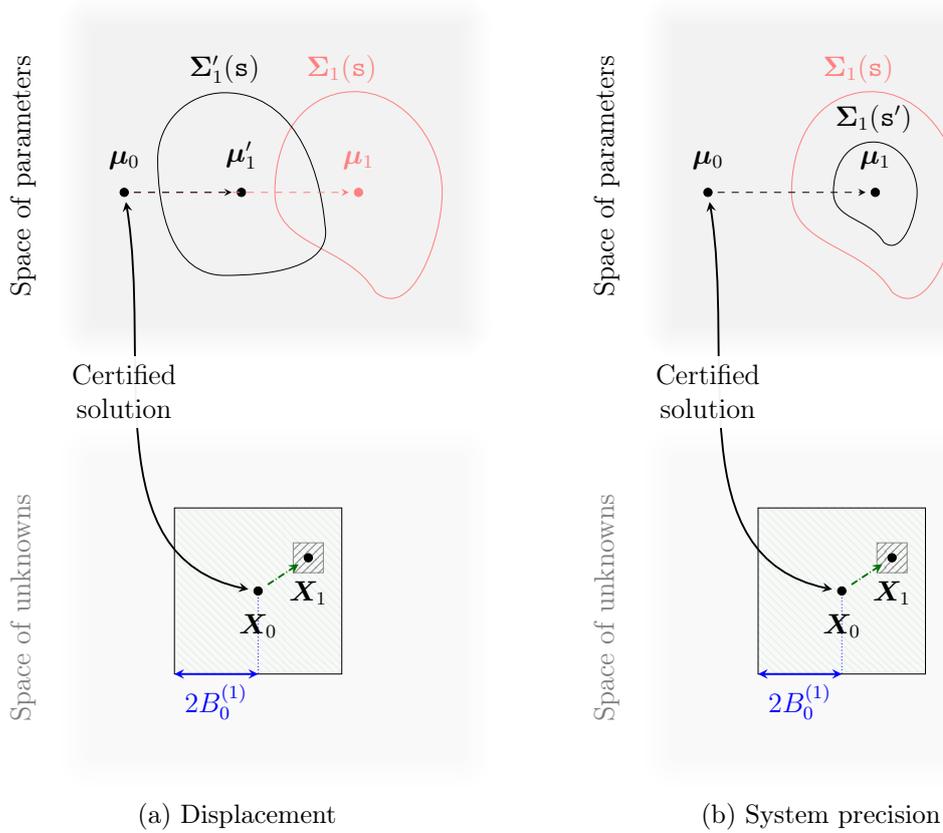


Figure 5: Strategies to refine Path Tracking with the Kantorovich test

not crucial since users are mostly interested in a unique leaf of solution which defines the utilization of the robot. In this respect, this semi-numerical strategy only provides local information unlike the discriminant variety (if computable and exploitable) that projects onto the parameter space all the instable regions coming from all the existing leaves of solution.

4. Study case of some real-case parallel robots

This section presents some real-case examples of parallel in which one applies the tools for the certification of their kinematics. Examples that will be considered are exclusively from rigid and nonredundant PKRs as they are part from the authors' previous works. Note however that this methodology and its implementation is also suitable for redundant [15] and flexible (*e.g.* continuum [64] or cable-driven [65, 45]) robots. In the case of robots with redundant actuation, uncertainties may render their geometric model inconsistent by altering the structure of its polynomial counterpart, so that it is no longer zero-dimensional. Indeed, as pointed out in [66], geometric uncertainties of redundantly actuated PKM may cause (possibly high) actuator loads that have no effect on the motion.

4.1. 3-RPS Tripod

This example focuses on a 3-DOF 3-RPS Tripod depicted in Figure 6.

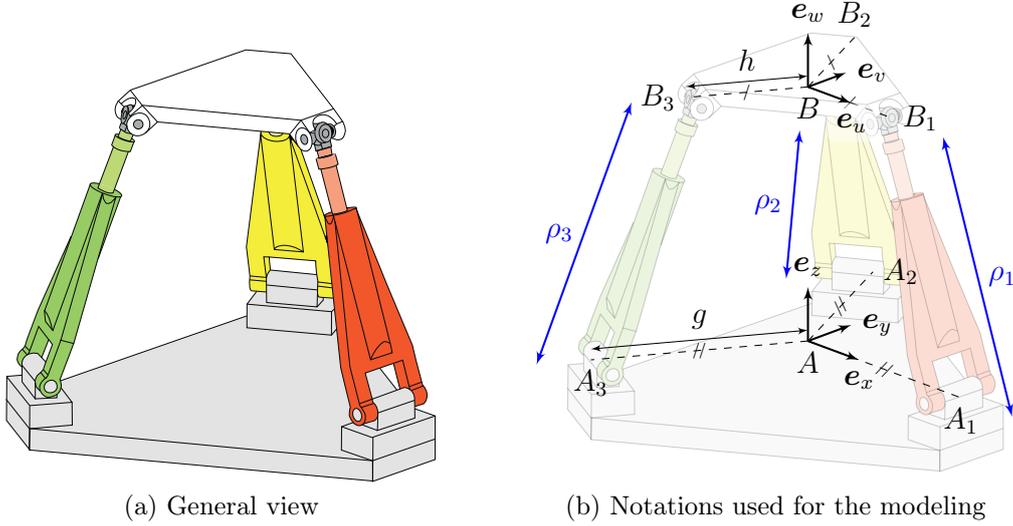


Figure 6: Design of the 3-RPS Tripod

4.2. Kinematic description and modeling

It is a nonredundant parallel manipulator with two platforms that can be seen as two equilateral triangles connected with three prismatic legs. Each leg is linked with the base through a revolute joint and with the upper platform through a spherical joint.

Let A_1, A_2 and A_3 (*resp.* B_1, B_2 and B_3) be the vertices of the lower (*resp.* upper) equilateral triangle having center A (*resp.* B). One can describe such a manipulator by:

- its design parameters $\Upsilon := (g, h)$, where g (*resp.* h) denotes the circumradius of $A_1A_2A_3$ (*resp.* $B_1B_2B_3$);
- its active joint variables $\boldsymbol{\rho} := (\rho_1, \rho_2, \rho_3)$ being the lengths of the prismatic legs;
- its operational coordinates $\tilde{\boldsymbol{x}}$ describing the orientation and translation of the upper platform. The translational part is described by the coordinates of B , *i.e.* (x, y, z) whereas the orientational one is parametrized using unit quaternions so that $\tilde{\boldsymbol{x}} := (x, y, z, q_w, q_x, q_y, q_z)$.

In the sequel, one also defines the frames $\mathcal{F}_b := (A, \mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$ attached to the base and $\mathcal{F}_* := (B, \mathbf{e}_u, \mathbf{e}_v, \mathbf{e}_w)$ attached to the upper platform. The kinematics of 3-RPS tripods have already and deeply been investigated in [25, 67, 68] using symbolic computations. The latter references pointed out the existence of two operation modes: $q_x = 0$ or $q_z = 0$ which will be chosen in the sequel so that the operational coordinates of interest writes $\boldsymbol{x} = (z, q_w, q_x, q_y)$.

Taking into account the distance equations leads to the following geometric model \mathbf{f} defined with $q_z = 0$ by

$$\mathbf{f}(\mathbf{x}, \boldsymbol{\rho}; \boldsymbol{\Upsilon}) = \begin{bmatrix} \|\mathbf{b}\mathbf{r}_{AA_1} - \mathbf{b}\mathbf{r}_{AB_1}\|_2^2 - \rho_1^2 \\ \|\mathbf{b}\mathbf{r}_{AA_2} - \mathbf{b}\mathbf{r}_{AB_2}\|_2^2 - \rho_2^2 \\ \|\mathbf{b}\mathbf{r}_{AA_3} - \mathbf{b}\mathbf{r}_{AB_3}\|_2^2 - \rho_3^2 \\ q_w^2 + q_x^2 + q_y^2 - 1 \end{bmatrix} = \mathbf{0}_{4 \times 1}, \quad (14)$$

where \mathbf{r}_{AB} represents the coordinates of the vector \overrightarrow{AB} , the superscript \mathbf{b} means “expressed in the frame \mathcal{F}_b ” and $\|\cdot\|_2$ denotes the euclidian norm. The reader may refer to Appendix A for the details on the modeling of 3-RPS tripods. Notice that the system \mathbf{f} is already polynomial due to the quaternion formulation of the end-effector orientation: $\mathbf{F} = \mathbf{f}$.

4.2.1. Inverse Kinematics

Let $\tilde{\mathbf{f}} := (f_1, f_2, f_3)$ be the equations of interest for the Inverse Kinematics. It must first be pointed out that the Inverse Kinematics of 3-RPS tripods are trivial and only admit one working mode (if feasible) since each of its equation is decoupled and quadratic w.r.t. ρ_i with $\rho_i > 0$, $\forall i \in \llbracket 1, 3 \rrbracket$. It is also straightforward that the Type 1 singularities only occur when $\rho_1 = \rho_2 = \rho_3 = 0$ as

$$\mathbf{J}_\rho := \frac{\partial \tilde{\mathbf{f}}}{\partial \boldsymbol{\rho}} = \mathbf{diag}(-2\rho_1, -2\rho_2, -2\rho_3). \quad (15)$$

The last case is however in practice impossible because of the before-mentioned positiveness of the lengths ρ_i . The boundaries of reachable workspace then entirely depends on the actuation limits, *i.e.* the minimal and maximal lengths of the legs: $\rho_{i,\min} \leq \rho_i \leq \rho_{i,\max}$, $\forall i \in \llbracket 1, 3 \rrbracket$. Such results hold for any design parameters $\boldsymbol{\Upsilon} = (g, h)$.

4.2.2. Forward Kinematics

At (g, h) fixed, the geometric model (see Appendix A) has four polynomials whose total degree does not exceed 4: it is in this respect reasonable enough to attempt a computation of a discriminant variety of the Forward Kinematics w.r.t. Π_ρ . Symbolic and semi-numerical tools can therefore both be used on this mechanism to certify its kinematics.

Symbolic tools. Starting with the symbolic tools, a discriminant variety of the Forward Kinematics of 3-RPS tripod can be obtained by computing Gröbner bases w.r.t. the product order $>_{\boldsymbol{\rho}, \mathbf{x}}$, given that (ρ_1, ρ_2, ρ_3) is the parameter space of the Forward Kinematics. By doing so, one obtains for $g = h = 1$ the components $\mathcal{W}_{\text{sd}} = \emptyset$ (as $n_p = n = 4$), $\mathcal{W}_\infty = \emptyset$ and

$$\begin{aligned} \mathcal{W}_c = & (\rho_1^4 \rho_2^4 - \rho_1^4 \rho_2^2 \rho_3^2 + \rho_1^4 \rho_3^4 - \rho_1^2 \rho_2^4 \rho_3^2 - \rho_1^2 \rho_2^2 \rho_3^4 + \rho_2^4 \rho_3^4 - 9\rho_1^2 \rho_2^2 \rho_3^2) \\ & (\rho_1^8 - 2\rho_1^6 \rho_2^2 - 2\rho_1^6 \rho_3^2 + 3\rho_1^4 \rho_2^4 + 3\rho_1^4 \rho_3^4 - 2\rho_1^2 \rho_2^6 - 2\rho_1^2 \rho_3^6 + \rho_2^8 - 2\rho_2^6 \rho_3^2 + 3\rho_2^4 \rho_3^4 \\ & - 2\rho_2^2 \rho_3^6 + \rho_3^8 - 24\rho_1^6 + 36\rho_1^4 \rho_2^2 + 36\rho_1^4 \rho_3^2 + 36\rho_1^2 \rho_2^4 - 144\rho_1^2 \rho_2^2 \rho_3^2 + 36\rho_1^2 \rho_3^4 \\ & - 162\rho_1^2 \rho_2^2 + 36\rho_2^4 \rho_3^2 + 36\rho_2^2 \rho_3^4 - 24\rho_3^6 + 162\rho_1^4 - 24\rho_2^6 - 162\rho_1^2 \rho_3^2 + 162\rho_2^4 \\ & - 162\rho_2^2 \rho_3^2 + 162\rho_3^4 - 2187), \end{aligned} \quad (16)$$

so that $\mathcal{W}_D = \mathcal{W}_c$. Figure 7 depicts the plot of the discriminant variety of the Forward Kinematics w.r.t. the projection onto the joint space for $\rho_i \in [0, 1.5]$, $\forall i \in \llbracket 1, 3 \rrbracket$. Such a hypersurface exactly describes the Type 2 singularities of the 3-RPS tripod with an inner and upper platform of radius 1 for all the existing leaves of solution of the FK.

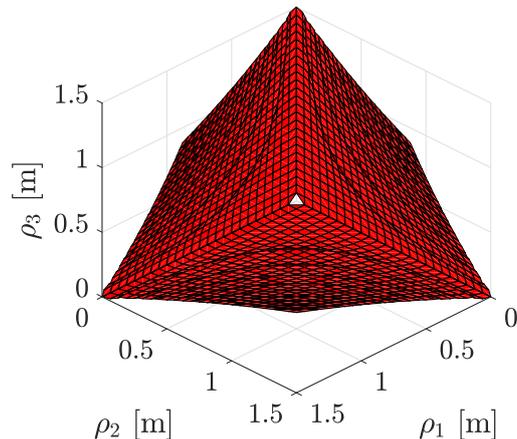


Figure 7: Discriminant variety of the Inverse Kinematics of the 3-DOF 3-RPS tripod of interest (with $g = h = 1$)

Semi-numerical tools. Using the semi-numerical tools involving the certified path tracking of the FKP requires the specification of a leaf of solution. One chooses an initial configuration with the upper platform being parallel to the base ($q_{w,0} = 1$, $q_{x,0} = 0$, $q_{y,0} = 0$) at a height of $z_0 = 1$ m ($\rho_1 = \rho_2 = \rho_3 = z_0$ accordingly to the design parameters). One defines the following scenario for the path³ tracking problem:

- a heave motion satisfying $z(t) = z_0 + T_z \sin(2\pi f_z t)$, such that $T_z = 3.5 \times 10^{-2}$ m and $f_z = 0.5$ Hz.
- a motion in bank θ_b defined by $\theta_b(t) = \Theta_b \sin(2\pi f_b t)$, with $\Theta_b = 10^\circ \simeq 0.1745$ rad and $f_b = 1$ Hz.

This trajectory is discretized with a sampling period of $T_e = 10$ ms. As it is expressed in the operational space, its translation into the active joint space can be generated by computing for each sample k the Inverse Kinematics using (14) with $\mathbf{x}[k]$ being specialized. The Kantorovich test is then applied to the joint reference $\boldsymbol{\rho}[k]$ using the principles stated earlier:

- If $\bar{\nu}_0 \leq 1$ (success), then $\mathbf{x}[k]$ is computed using a Newton scheme with the working precision of interest and becomes the starting point for the next test on the sample $k + 1$. Moreover, the displacement into the (joint) parameter space δ_ρ is realized by moving from $\boldsymbol{\rho}[k]$ to $\boldsymbol{\rho}[k + 1]$.

³As we introduce the notion of time, the path tracking in question is rather a *trajectory* one.

- If $\bar{\nu}_0 > 1$ (failure), then we decrease δ_ρ by moving to $\rho'[k]$ instead of $\rho[k+1]$ or increase the system precision before relaunching the test until either $\bar{\nu}_0 \leq 1$ or all the limits are reached.

Table 1 details the results on cases where the Kantorovich test is not successful on the first try. The number k denotes the sample in question and the **flag** is defined by

$$\mathbf{flag} := K + 2c, \quad (17)$$

where $K \in \llbracket 0, 1 \rrbracket$ (1 if the test succeeded or 0 if it fails) and:

- $c = 0$ if the precision of the Newton scheme is reached;
- $c = 1$ if the interval jacobian matrix w.r.t. the variables is not invertible;
- $c = 2$ if the precision of the Newton scheme is not reached and can no longer be refined (input precision or working precision insufficient);
- $c = 3$ if the maximum number of Newton iteration has been reached (here set as 10).

k	$\rho[k]$	Flag	$[\nu_0]$	Kantorovich radius	$\rho[k-1]$	$\rho'[k]$
512	[0.9869, 1.091, 0.8828]	4	[1.082, 1.120]	—	[0.9874, 1.086, 0.8904]	[0.9872, 1.088, 0.8866]
- -	- -	0	[1.163, 1.212]	—	- -	[0.9872, 1.088, 0.8879]
- -	- -	0	[1.163, 1.212]	—	- -	[0.9873, 1.087, 0.8885]
- -	- -	0	[1.0167, 1.0535]	—	- -	[0.98730, 1.0870, 0.88888]
- -	- -	1	[0.93625, 0.96950]	[0.0044220, 0.0044220]	- -	—
563	[0.9680, 0.8574, 1.078]	0	[0.9911, 1.026]	—	[0.9680, 0.8634, 1.072]	[0.9680, 0.8604, 1.075]
- -	- -	4	[1.102, 1.148]	—	- -	[0.9680, 0.8614, 1.074]
- -	- -	1	[0.7956, 0.8273]	[0.004854, 0.004854]	- -	—
590	[0.9889, 0.9017, 1.077]	0	[1.069, 1.110]	—	[0.9879, 0.8934, 1.083]	[0.9884, 0.8976, 1.080]
- -	- -	5	[0.8340, 0.8616]	[0.005893, 0.005893]	- -	—
604	[1.004, 1.043, 0.9653]	4	[1.005, 1.044]	—	[1.003, 1.033, 0.9729]	[1.004, 1.038, 0.9691]
- -	- -	1	[0.7920, 0.8336]	[0.006332, 0.006332]	- -	—
813	[1.014, 1.123, 0.9043]	4	[1.173, 1.219]	—	[1.013, 1.116, 0.9097]	[1.014, 1.120, 0.9070]
- -	- -	0	[1.182, 1.240]	—	- -	[1.013, 1.118, 0.9079]
- -	- -	1	[0.8642, 0.9008]	[0.005133, 0.005133]	- -	—

Table 1: Results on the trajectory tracking for problematic cases

Such computations are made using a double precision (working precision $w = 52$ bits) and the system precision taking into account is $s = 14$ bits $\simeq 4$ digits for a simulation spanning $T_{\text{simu}} = 10$ s. In our case, the whole trajectory is certified with a successful Kantorovich test on the first try except for some samples (see Table 1) for which a smaller displacement in the parameter space is sufficient to get $\bar{\nu}_0 < 1$: the **flag** number is systematically odd (1 or 5 for $k = 590 \iff t = 5.9$ s). For the latter case, another refinement in the displacement is sufficient to get **flag** = 1. Finally, the trajectory of this simulation and the certified domain covering it are shown in Figure 8. The latter domain (in green) acts like a safe zone for our trajectory (in black).

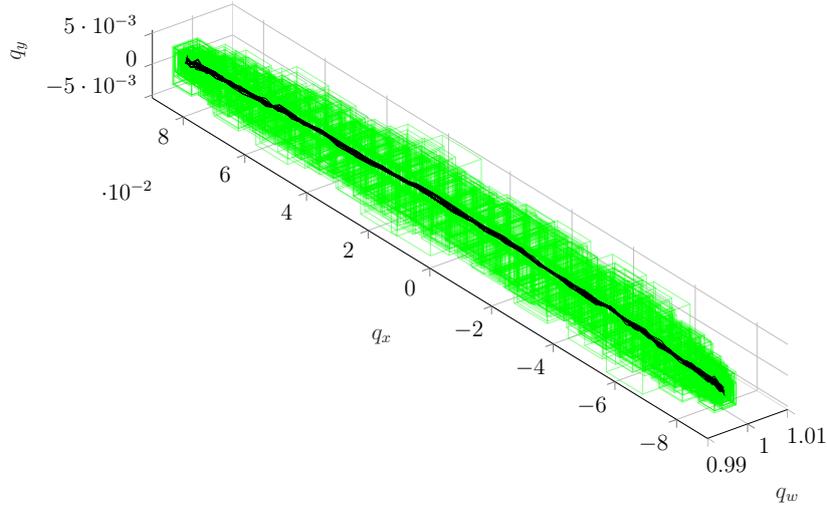


Figure 8: Trajectory (in black) and its safe area (in green) represented in the (q_w, q_x, q_y) -space (z -data is not shown here)

4.3. 3-RRR Spherical Parallel Manipulators

In this example, we will apply the previous tools to study the Kinematics of Spherical Parallel Manipulators (SPMs) [9, 69, 10, 3]. We first recall the kinematic modeling of the SPMs in the general case before focusing on two specific examples: Symmetrical [17] and Asymmetrical [8] SPMs with coaxial [10] input shafts (CoSPM & AsyCoSPM).

4.3.1. Kinematic description and modeling

Spherical Parallel Manipulators (SPMs) are parallel robots whose upper platforms only have rotational DOFs.

This subsection focuses on 3-RRR 3-DOF SPMs whose general kinematic diagram is depicted in Figure 9. According to the latter, any SPM can be described as a manipulator that has two platforms connected with three legs. Each platform is attached to a (direct) base: $\mathcal{B}_b := \{\mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b\}$ for the inner platform and $\mathcal{B}_* := \{\mathbf{x}_*, \mathbf{y}_*, \mathbf{z}_*\}$ for the upper one. The design parameters Υ of an SPM are angles related to the geometry of:

- the legs (see Fig. 9a in magenta): each leg has two bodies and an actuated joint at its base. Given the i^{th} leg, the nearest link to the base is called *proximal link* of angle $\alpha_{1,i}$ whereas the farthest one is called *distal link* of angle $\alpha_{2,i}$;
- the platforms (see Fig. 9a, 9b and 9c in green): that of the base (*resp.* upper platform) is described by the angle β_1 (*resp.* β_2). Each platform has three pivot linkages whose dispositions are identical and defined by the angle η_k , $k \in \llbracket 1, 3 \rrbracket$.

Moreover, both active and passive joints are revolute. On the one hand, the actuated joint variables (angles) will be denoted as θ_i for the i^{th} leg so that the joint vector is defined by $\boldsymbol{\theta} := (\theta_1, \theta_2, \theta_3)$. On the other hand, one chooses the ZYX Euler Tait-Bryan angles

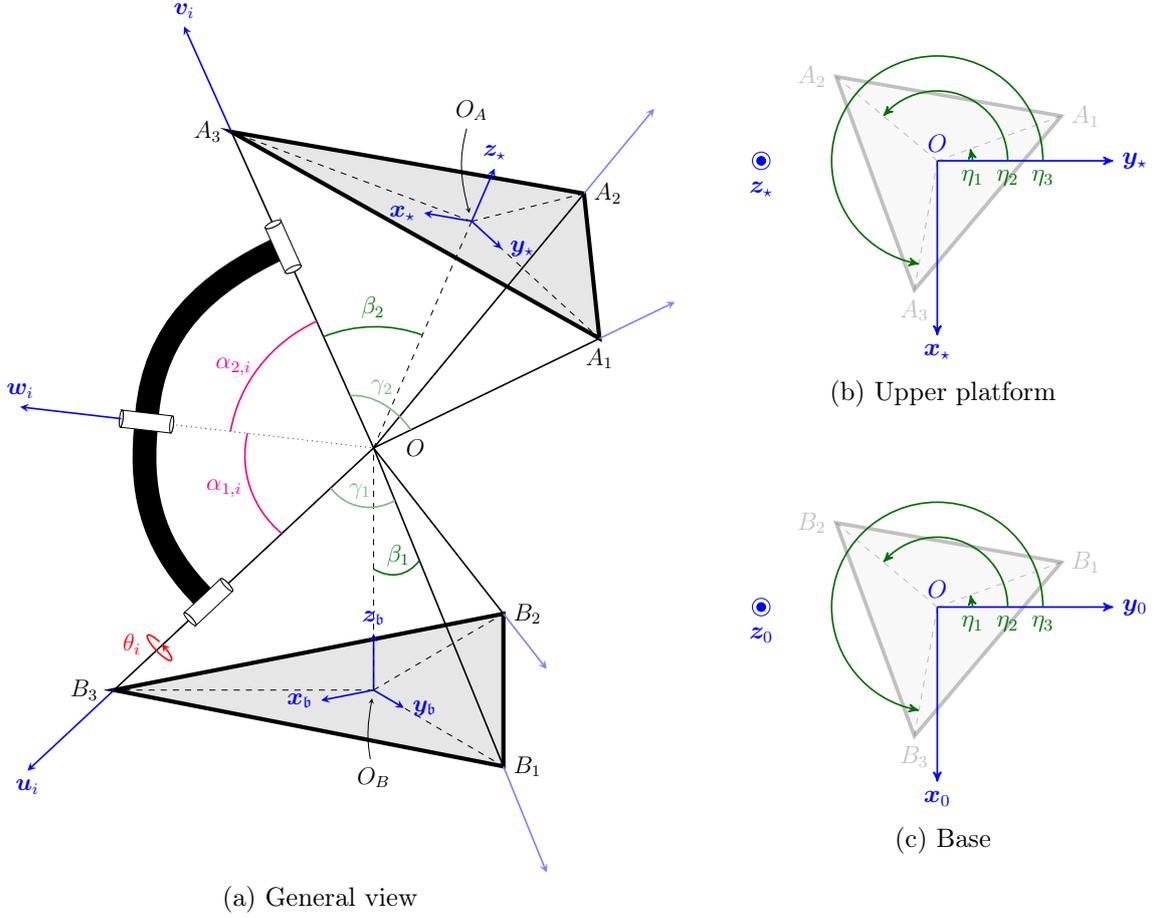


Figure 9: Kinematic diagram and design parameters of a general 3-DOF 3-RRR Spherical Parallel Manipulator

formulation⁴ to describe the three DOFs of the upper platform, so that the operational vector is defined by $\boldsymbol{\chi} := (\chi_1, \chi_2, \chi_3)$. In the sequel, χ_1 will be called the *bank* angle, χ_2 the *elevation* angle and χ_3 the *bearing* angle. Fig. 9a also highlights the fact that SPMs only make pure rotations around O called *center of rotation* of the SPM. Using this property, their motions can be described with only vectors that are normal w.r.t. each revolute joint and oriented outward (see Fig. 9a). There are three types of vectors: the ones describing the base denoted as \boldsymbol{u}_i , the ones describing the moving platform denoted as \boldsymbol{v}_i and the ones describing the intermediate joints denoted as \boldsymbol{w}_i , with $i \in \llbracket 1, 3 \rrbracket$. An important consideration is that all these vectors are *theoretically* concurrent in O . The reader may refer to [9, 3] and Appendix B for the details on the computation of the before-mentioned vectors. One can show that the *geometric model* of a general SPM is established using the following kinematic

⁴Appendix B provides the details on the formalism used to describe the orientation coordinates.

closure:

$$\mathbf{f}(\boldsymbol{\theta}, \boldsymbol{\chi}; \boldsymbol{\Upsilon}) := \begin{bmatrix} \mathbf{w}_1^\top(\theta_1; \boldsymbol{\Upsilon}) \mathbf{v}_1(\boldsymbol{\chi}; \boldsymbol{\Upsilon}) - \cos(\alpha_{2,1}) \\ \mathbf{w}_2^\top(\theta_2; \boldsymbol{\Upsilon}) \mathbf{v}_2(\boldsymbol{\chi}; \boldsymbol{\Upsilon}) - \cos(\alpha_{2,2}) \\ \mathbf{w}_3^\top(\theta_3; \boldsymbol{\Upsilon}) \mathbf{v}_3(\boldsymbol{\chi}; \boldsymbol{\Upsilon}) - \cos(\alpha_{2,3}) \end{bmatrix} = \mathbf{0}_{3 \times 1}. \quad (18)$$

The latter nonlinear system (18) can be turned into a polynomial one using the before-mentioned half-angle tangent identities. By setting $\forall i \in \llbracket 1, 3 \rrbracket$, $\Theta_i := \tan\left(\frac{\theta_i}{2}\right)$ and $X_i := \tan\left(\frac{\chi_i}{2}\right)$ with $\boldsymbol{\Theta} := (\Theta_1, \Theta_2, \Theta_3)$ and $\mathbf{X} := (X_1, X_2, X_3)$ respectively being the polynomial joint and operational vectors of the SPM, one has a polynomial system of the form

$$\mathbf{F}(\boldsymbol{\Theta}, \mathbf{X}; \boldsymbol{\Upsilon}) := \begin{bmatrix} a_1(\mathbf{X}; \boldsymbol{\Upsilon}) \Theta_1^2 + b_1(\mathbf{X}; \boldsymbol{\Upsilon}) \Theta_1 + c_1(\mathbf{X}; \boldsymbol{\Upsilon}) \\ a_2(\mathbf{X}; \boldsymbol{\Upsilon}) \Theta_2^2 + b_2(\mathbf{X}; \boldsymbol{\Upsilon}) \Theta_2 + c_2(\mathbf{X}; \boldsymbol{\Upsilon}) \\ a_3(\mathbf{X}; \boldsymbol{\Upsilon}) \Theta_3^2 + b_3(\mathbf{X}; \boldsymbol{\Upsilon}) \Theta_3 + c_3(\mathbf{X}; \boldsymbol{\Upsilon}) \end{bmatrix} = \mathbf{0}_{3 \times 1}, \quad (19)$$

where a_i , b_i and c_i are coefficients of the polynomial $F_i(\Theta_i)$ that exclusively depend on the polynomial operational variables X_i (along with the design parameters $\boldsymbol{\Upsilon}$). Their tedious expressions are not shown in this paper. It is however then worth mentioning that this polynomial formalism is particularly convenient to solve the IKP. Indeed, as each polynomial F_i is univariate and quadratic w.r.t. Θ_i , one computes at a given orientation $\boldsymbol{\chi}^*$ the corresponding \mathbf{X}^* and deduces $\theta_i^* = 2 \arctan(\Theta_i^*)$ where

$$\begin{aligned} \Theta_i^* &= \frac{-b_i(\mathbf{X}^*) \pm \sqrt{b_i^2(\mathbf{X}^*) - 4a_i(\mathbf{X}^*)c_i(\mathbf{X}^*)}}{2a_i(\mathbf{X}^*)}, \\ &= \frac{-b_i(\mathbf{X}^*) \pm \sqrt{\text{discrim}(F_i, \Theta_i)}}{2\text{lc}(F_i, \Theta_i)}. \end{aligned} \quad (20)$$

Equations (19) and (20) clearly show that SPMs have at most $2^3 = 8$ working modes (real and distinct for nonsingular configurations). Taking into account the previous remarks on the singularity analysis of PKRs, Type 1 singularities occur when at least two working modes are identical for the same operational vector: this requires to investigate $\text{discrim}(F_i, \Theta_i)$, *i.e.* the discriminant of each F_i w.r.t. Θ_i and therefore the critical points \mathcal{W}_c . Moreover, one should pay attention to the denominator of (20): orientation values making $a_i(\mathbf{X}) = \text{lc}(F_i, \Theta_i)$ vanish belong to \mathcal{W}_∞ .

4.3.2. Symmetrical Coaxial Spherical Parallel Manipulator (CoSPM)

An SPM is said to be *coaxial* (CoSPM) if its actuators located at its base share the same axis of rotation. The geometrical condition allowing that is $\beta_1 = 0$. This example focuses on a CoSPM whose design is depicted in Figure 10: it is called *symmetrical* in the sense that the pivot linkages of the upper platform are regularly spaced (see Fig. 10e), and the links (proximal and distal) are identical for each leg (see Fig. 10b–10d). The design parameters considered in this study are shown in Table 2 and taken from [17] maximizing the global conditioning index of the SPM over its reachable workspace.

Given the Euler Tait-Bryan ZYX parametrization of the orientation $\boldsymbol{\chi}$, an important kinematic property of any CoSPM is that its I/O singularity loci are independent w.r.t. the χ_3 -coordinate [8, 70]. As a result, one can exploit the latter fact to simplify the I/O sin-

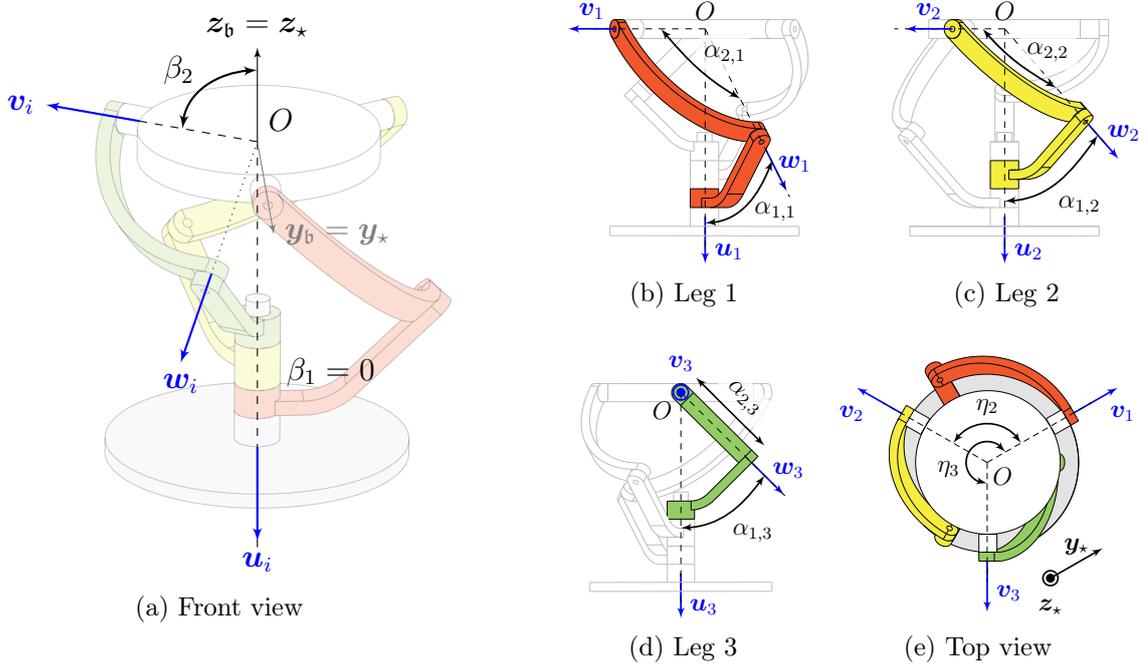


Figure 10: Details on the Symmetrical Coaxial Spherical Parallel Manipulator

gularity analysis by considering the projection of the workspace \mathcal{W} into the (χ_1, χ_2) -plane (setting an arbitrary value for $\chi_3 \in \mathbb{R}$, *e.g.* 0) or the projection of the joint space \mathcal{Q} into the (θ_1, θ_2) -plane by setting θ_3 at an arbitrary real value. Note that all the joint values belonging

Design parameter	Notation	Value (rad)
Proximal link $\alpha_{1,i}$ (i^{th} leg)	$\alpha_{1,1}$	$\pi/4$
	$\alpha_{1,2}$	$\pi/4$
	$\alpha_{1,3}$	$\pi/4$
Distal link $\alpha_{2,i}$ (i^{th} leg)	$\alpha_{2,1}$	$\pi/2$
	$\alpha_{2,2}$	$\pi/2$
	$\alpha_{2,3}$	$\pi/2$
Pivot linkage disposition η_i (i^{th} leg)	η_1	0
	η_2	$2\pi/3$
	η_3	$4\pi/3$
Inner platform's geometry	β_1	0
Upper platform's geometry	β_2	$\pi/2$

Table 2: Design parameters of the CoSPM of interest

to the second projection should generate all the possible orientation coordinates χ_1 and χ_2 with a varying χ_3 -coordinate, which does not matter since the latter is singularity free.

Inverse Kinematics. With \mathbf{X} being the vector of parameters, a discriminant variety \mathcal{W}_D of (19) can be found by taking the union of the resultant of each polynomial F_i and its derivative $\frac{\partial F_i}{\partial \Theta_i}$ w.r.t. Θ_i , $\forall i \in \llbracket 1, 3 \rrbracket$. Writing

$$\mathcal{W}_D(\mathbf{F}) = \bigcup_{i=1}^{n_a=3} \mathcal{W}_D(F_i) = \bigcup_{i=1}^{n_a=3} \left(\bigcup_{j=1}^3 \mathcal{W}_{i,j} \right)$$

with $\mathcal{W}_c(F_i) = \mathcal{W}_{i,1} \cup \mathcal{W}_{i,2}$ and $\mathcal{W}_\infty(F_i) = \mathcal{W}_{i,3}$ leads to

$$\mathcal{W}_{1,1} = -4(X_3^2 + 1)^2, \quad (21a)$$

$$\mathcal{W}_{1,2} = X_1^4 X_2^4 + 2X_1^4 X_2^2 - 6X_1^2 X_2^4 + X_1^4 + 20X_1^2 X_2^2 + X_2^4 - 6X_1^2 + 2X_2^2 + 1, \quad (21b)$$

$$\begin{aligned} \mathcal{W}_{1,3} = & X_1^2 X_2^2 X_3^2 - 2X_1 X_2^2 X_3^2 - X_1^2 X_2^2 + X_1^2 X_3^2 - X_2^2 X_3^2 - 2X_1 X_2^2 + 8X_1 X_2 X_3 \\ & + 2X_1 X_3^2 - X_1^2 + X_2^2 - X_3^2 + 2X_1 + 1, \end{aligned} \quad (21c)$$

$$\mathcal{W}_{2,1} = 16(X_3^2 + 1)^2, \quad (21d)$$

$$\begin{aligned} \mathcal{W}_{2,2} = & -X_1^4 X_2^4 + 4X_1^3 \sqrt{3} X_2^3 + 4X_1^4 X_2^2 - 4X_1^3 X_2 \sqrt{3} + 4X_1 \sqrt{3} X_2^3 - X_1^4 + 4X_1^2 X_2^2 \\ & - X_2^4 - 4\sqrt{3} X_1 X_2 + 4X_2^2 - 1, \end{aligned} \quad (21e)$$

$$\begin{aligned} \mathcal{W}_{2,3} = & 2\sqrt{3} X_1^2 X_2 X_3^2 + 2X_1^2 X_2^2 X_3^2 - 2\sqrt{3} X_1 X_2 X_3^2 + 2X_1 X_2^2 X_3^2 + 2\sqrt{3} X_1^2 X_2 \\ & + 2\sqrt{3} X_1^2 X_3 - 2\sqrt{3} X_2^2 X_3 + 2\sqrt{3} X_2 X_3^2 - 2X_1^2 X_2^2 - X_1^2 X_3^2 + X_2^2 X_3^2 \\ & + 2\sqrt{3} X_1 X_2 + 2X_1 X_2^2 + 4X_1 X_2 X_3 - 2X_1 X_3^2 + 2X_2 \sqrt{3} + X_1^2 - X_2^2 \\ & - 2X_3^2 - 2X_1 + 2, \end{aligned} \quad (21f)$$

$$\mathcal{W}_{3,1} = 16(X_3^2 + 1)^2, \quad (21g)$$

$$\begin{aligned} \mathcal{W}_{3,2} = & X_1^4 X_2^4 + 4X_1^3 \sqrt{3} X_2^3 - 4X_1^4 X_2^2 - 4X_1^3 X_2 \sqrt{3} + 4X_1 \sqrt{3} X_2^3 + X_1^4 - 4X_1^2 X_2^2 \\ & + X_2^4 - 4\sqrt{3} X_1 X_2 - 4X_2^2 + 1, \end{aligned} \quad (21h)$$

$$\begin{aligned} \mathcal{W}_{3,3} = & 2\sqrt{3} X_1^2 X_2 X_3^2 - 2X_1^2 X_2^2 X_3^2 - 2\sqrt{3} X_1 X_2 X_3^2 - 2X_1 X_2^2 X_3^2 + 2\sqrt{3} X_1^2 X_2 \\ & + 2\sqrt{3} X_1^2 X_3 - 2\sqrt{3} X_2^2 X_3 + 2\sqrt{3} X_2 X_3^2 + 2X_1^2 X_2^2 + X_1^2 X_3^2 - X_2^2 X_3^2 \\ & + 2\sqrt{3} X_1 X_2 - 2X_1 X_2^2 - 4X_1 X_2 X_3 + 2X_1 X_3^2 + 2X_2 \sqrt{3} - X_1^2 + X_2^2 \\ & + 2X_3^2 + 2X_1 - 2. \end{aligned} \quad (21i)$$

Figure 11 depicts this algebraic variety: the critical points and the infinity points in the \mathbf{X} -space are respectively shown in Fig. 11a and Fig. 11b. A projection of these components in the (χ_1, χ_2) -plane at $X_3 = 0$ is represented in Fig. 11c.

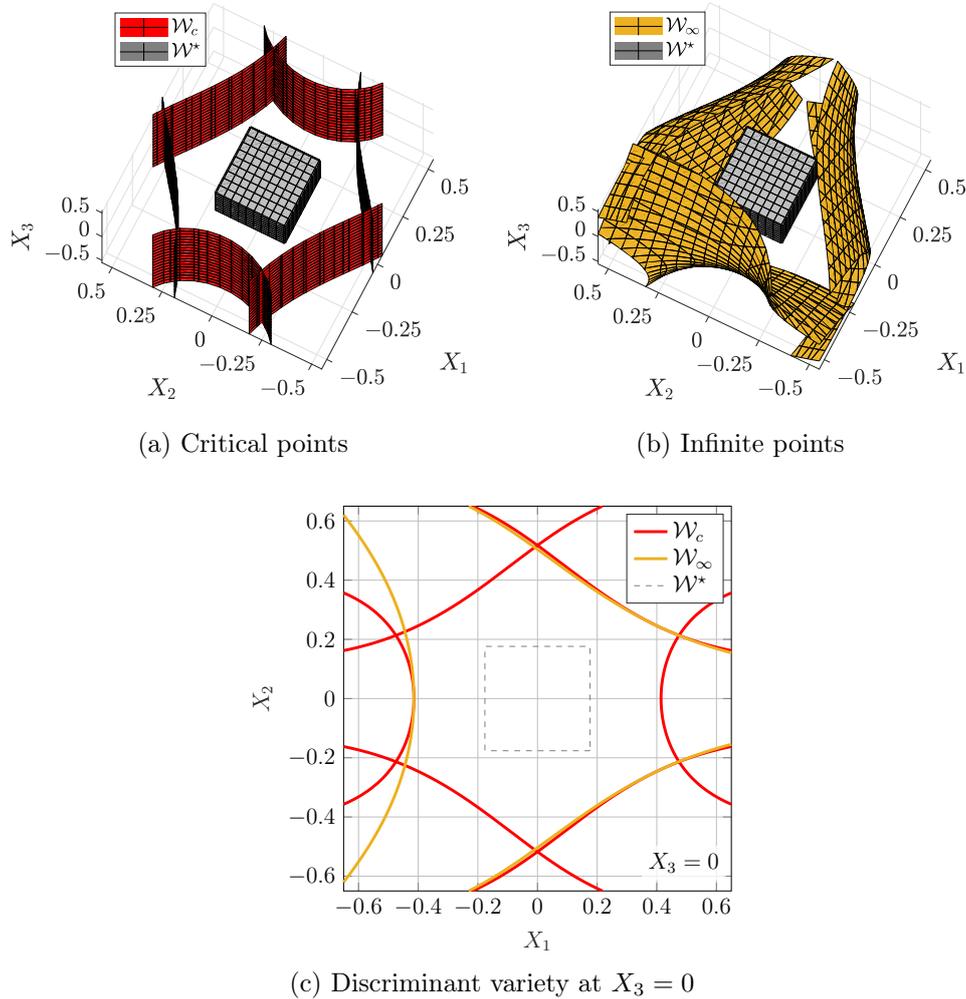


Figure 11: Discriminant variety of the polynomial system \mathbf{F} modeling the CoSPM w.r.t. $\Pi_{\mathbf{X}}$

As expected, the critical points encoding the Type 1 singularity loci of our SPM does not depend on the X_3 - (and thus χ_3 -) coordinate. The infinite points of the polynomial IKP do however depend on this coordinate: these loci are modeling artifacts and only concern the polynomial system in its tangent half-angle formulation. In all cases, one can guaranty a singularity-free prescribed workspace \mathcal{W}^* enabling a motion in bank and elevation such that $\chi_1, \chi_2 \in [-20^\circ, +20^\circ]$, with unlimited bearing $\chi_3 \in \mathbb{R}$.

Forward Kinematics. The certification of the FKP is done considering the path tracking problem in orientation $\boldsymbol{\chi}$ and a twofold strategy [3]. After setting the leaf of solution $\Theta = (1, 1, 1) \leftrightarrow \mathbf{X} = (0, 0, 0)$ corresponding to the home configuration of the CoSPM defined by $\theta = (\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}) \leftrightarrow \boldsymbol{\chi} = (0, 0, 0)$:

1. One generates a displacement $\delta_{\mathbf{X}}$ in the space of unknowns and solve the IKP over the

whole reachable workspace bounded by the Type 1 singularities (found by the critical points of the polynomial IKP).

2. This indirectly generates a displacement in the joint (parameter) space where a Kantorovich test is applied.

The whole reachable workspace can be studied by scanning its projection into the (X_1, X_2) -plane (at $X_3 = 0$ for instance) using a radial path tracking starting from the home configuration. One moves from a radius to another if one of the following conditions are met: either the Kantorovich test failed with the smallest displacement $\delta_{X,\min}$ allowed and the highest system precision \mathbf{s}_{\max} tolerated, either a Type 1 singular locus is reached.

In the sequel one sets the displacement in the \mathbf{X} -space at 2×10^{-2} not below 10^{-4} and a system precision at $\mathbf{s} = 14$ bits $\simeq 4$ digits because one wants to control the orientation of the upper platform with a precision of approximately 100 microradians taking into account an uncertainty of the design parameters of magnitude 10^{-5} rad.

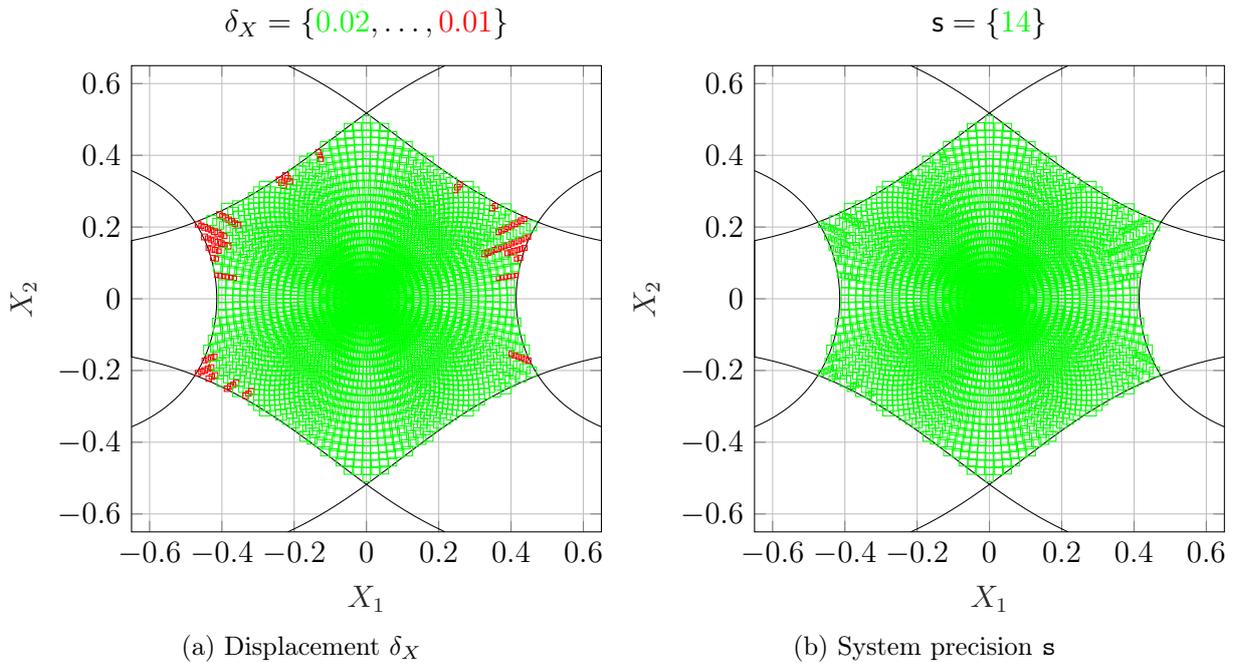


Figure 12: Workspace in the (X_1, X_2) -plane at $X_3 = 0$ and the certified area obtained by the Kantorovich unicity operator (green: initial best tuning, red: worst case tuning)

Figure 12 shows the Type 1 singular loci previously computed using the symbolic approach and the unicity domains of the FKP in the (X_1, X_2) -plane obtained using the certified numeric strategy with the Kantorovich tests. This strategy certifies the absence of any instability of the FKP (numerical or physical). Type 2 singularities never occur for this mechanism given our leaf of solution. Such a result holds given a system precision of $\mathbf{s} = 14$ bits and a displacement in the polynomial orientation space going from the nominal

value 2×10^{-2} ($\sim 10^{-2}$ rad, in green) to 1×10^{-2} ($\sim 5 \times 10^{-3}$ rad). Unicity domains that are not obtained with the nominal displacement especially occurs when the CoSPM is almost reaching the limits of its workspace.

4.3.3. Asymmetric Coaxial Spherical Parallel Manipulator (AsyCoSPM)

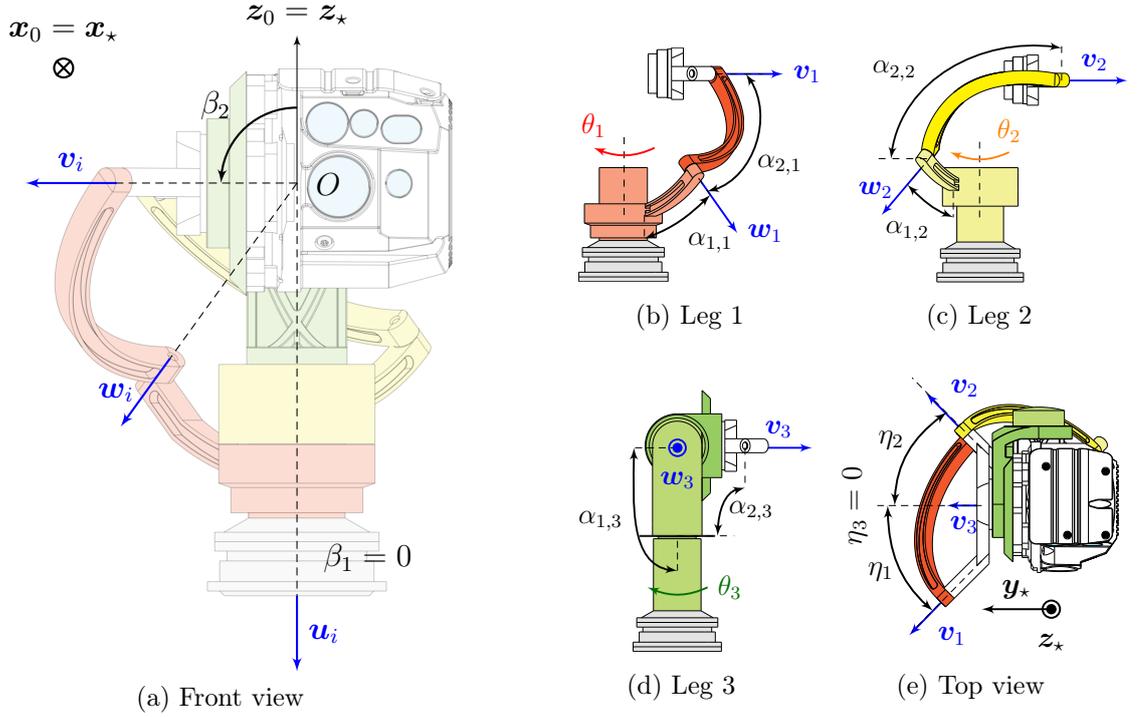


Figure 13: Details on the Asymmetric Coaxial Spherical Parallel Manipulator of [8]

A variation of the previous CoSPM called Asymmetric Coaxial SPM (AsyCoSPM) is depicted in Figure 13. It is called *asymmetric* in the sense that the geometry of the proximal and distal links vary from one leg to another (see Fig. 13b–13d or Table 3). In addition, the revolute joints of the upper platform are not regularly spaced as illustrated in Fig. 13e and Table 3. Such a design was introduced in [8] for the purpose of Line-Of-Sight (LOS) stabilization in a naval environment. The aim is to stabilize a set of camera despite the motion of the carrier that are subject to waves. The asymmetric architecture was then established for several reasons: one wanted to coincide the center of rotation O of the robot with the center of mass of the camera and widen the field of view by setting the legs behind the latter.

Inverse Kinematics. Using the same method as for the symmetrical case, one computes the discriminant variety of the IKP, i.e. w.r.t. the projection onto the (orientation) parameter space. Writing

$$\mathcal{W}_D(\mathbf{F}) = \bigcup_{i=1}^{n_a=3} \mathcal{W}_D(F_i) = \bigcup_{i=1}^{n_a=3} \left(\bigcup_{j=1}^4 \mathcal{W}_{i,j} \right)$$

Design parameter	Notation	Value (rad)
Proximal link $\alpha_{1,i}$ (i^{th} leg)	$\alpha_{1,1}$	$\pi/4$
	$\alpha_{1,2}$	$\pi/4$
	$\alpha_{1,3}$	$\pi/2$
Distal link $\alpha_{2,i}$ (i^{th} leg)	$\alpha_{2,1}$	$\pi/2$
	$\alpha_{2,2}$	$\pi/2$
	$\alpha_{2,3}$	$\pi/2$
Pivot linkage disposition η_i (i^{th} leg)	η_1	$\pi/4$
	η_2	$-\pi/4$
	η_3	0
Inner platform's geometry	β_1	0
Upper platform's geometry	β_2	$\pi/2$

Table 3: Design parameters of the AsyCoSPM of interest

with $\mathcal{W}_c(F_i) = \mathcal{W}_{i,1} \cup \mathcal{W}_{i,2} \cup \mathcal{W}_{i,3}$ and $\mathcal{W}_\infty(F_i) = \mathcal{W}_{i,4}$ leads to

$$\mathcal{W}_{1,1} = -4(X_3^2 + 1)^2(X_2 - 1)(X_2 + 1), \quad (22a)$$

$$\mathcal{W}_{1,2} = X_1^2X_2 + X_1^2 - 2X_1X_2 + 2X_1 + X_2 + 1, \quad (22b)$$

$$\mathcal{W}_{1,3} = X_1^2X_2 - X_1^2 + 2X_1X_2 + 2X_1 + X_2 - 1, \quad (22c)$$

$$\begin{aligned} \mathcal{W}_{1,4} = & \sqrt{2}X_1^2X_2X_3^2 - \sqrt{2}X_1X_2^2X_3^2 + X_1^2X_2^2X_3^2 + \sqrt{2}X_1^2X_2 - \sqrt{2}X_1X_2^2 + \sqrt{2}X_1X_3^2 \\ & + \sqrt{2}X_2X_3^2 - X_1^2X_2^2 + 2X_1X_2X_3^2 - 2X_1^2X_3 + 4X_3X_2X_1 + 2X_2^2X_3 + X_1\sqrt{2} \\ & + X_2\sqrt{2} - 2X_1X_2 - X_3^2 + 1, \end{aligned} \quad (22d)$$

$$\mathcal{W}_{2,1} = 4(X_3^2 + 1)^2(X_2 - 1)(X_2 + 1), \quad (22e)$$

$$\mathcal{W}_{2,2} = X_1^2X_2 - X_1^2 - 2X_1X_2 - 2X_1 + X_2 - 1, \quad (22f)$$

$$\mathcal{W}_{2,3} = X_1^2X_2 + X_1^2 + 2X_1X_2 - 2X_1 + X_2 + 1, \quad (22g)$$

$$\begin{aligned} \mathcal{W}_{2,4} = & \sqrt{2}X_1^2X_2X_3^2 + \sqrt{2}X_1X_2^2X_3^2 - X_1^2X_2^2X_3^2 + \sqrt{2}X_1^2X_2 + \sqrt{2}X_1X_2^2 - \sqrt{2}X_1X_3^2 \\ & + \sqrt{2}X_2X_3^2 + X_1^2X_2^2 + 2X_1X_2X_3^2 - 2X_1^2X_3 - 4X_3X_2X_1 + 2X_2^2X_3 - X_1\sqrt{2} \\ & + X_2\sqrt{2} - 2X_1X_2 + X_3^2 - 1, \end{aligned} \quad (22h)$$

$$\mathcal{W}_{3,1} = -4(X_3^2 + 1)^2, \quad (22i)$$

$$\mathcal{W}_{3,2} = X_1^2X_2^2 - 2X_1X_2^2 + X_1^2 + X_2^2 + 2X_1 + 1, \quad (22j)$$

$$\mathcal{W}_{3,3} = X_1^2 X_2^2 + 2X_1 X_2^2 + X_1^2 + X_2^2 - 2X_1 + 1, \quad (22k)$$

$$\mathcal{W}_{3,4} = X_1^2 X_2^2 X_3^2 - X_1^2 X_2^2 + X_1^2 X_3^2 - X_2^2 X_3^2 + 8X_3 X_2 X_1 - X_1^2 + X_2^2 - X_3^2 + 1. \quad (22l)$$

Figure 14 shows the plot of this discriminant variety in the (X_1, X_2, X_3) -space (Fig. 14a) and the (X_1, X_2) -plane at $X_3 = 0$ (Fig. 14b).

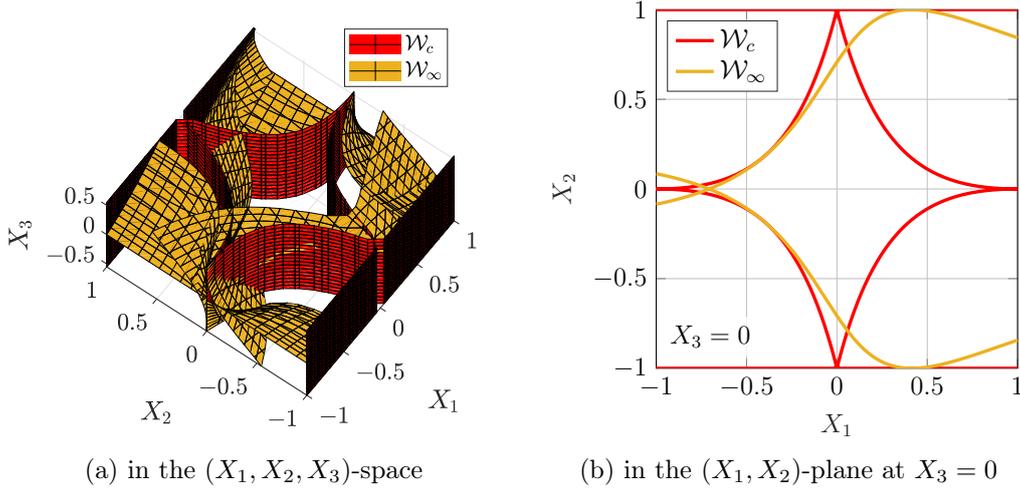


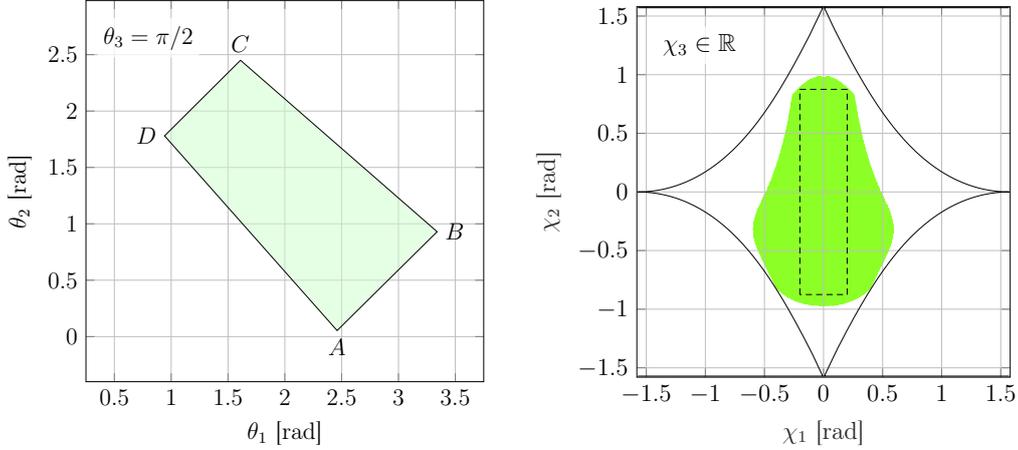
Figure 14: Discriminant variety of the polynomial IKP of the AsyCoSPM

As for the CoSPM previously studied, the Type 1 singularity loci encoded by \mathcal{W}^* does not depend on the X_3 -coordinate. However, such a design allows our robot to reach a bigger workspace than the previous version: for instance, at $X_1 = 0$ (zero bank), the robot can theoretically reach an elevation angle χ_2 such that $|X_2| < 1 \iff |\chi_2| < \pi/2 = 90^\circ$. In the sequel, one focuses on the prescribed workspace \mathcal{W}^* defined by:

- an unlimited motion in bearing, *i.e.* $\chi_3 \in \mathbb{R}$;
- a motion in elevation χ_2 such that $|\chi_2| \leq 50^\circ$;
- a motion in bank χ_1 such that $|\chi_1| \leq 10^\circ$.

Certifying the Forward Kinematics for the set \mathcal{W}^* ensures the absence of any numerical or physical singularities, namely the Type 2 ones.

Forward Kinematics. As for the symmetric case, one uses the Path Tracking problem in orientation to certify the Forward Kinematics of the AsyCoSPM. The previous twofold strategy could be applied for the current robot. However, one wants this time to directly delimit a subset \mathcal{Q}^* in the joint space \mathcal{Q} that is singularity-free and whose the image through the FKP given the leaf of solution $\boldsymbol{\theta} = (\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}) \leftrightarrow \boldsymbol{\chi} = (0, 0, 0)$ contains \mathcal{W}^* . In the sequel, one establishes such a set by defining a quadrilateral $ABCD$ in the (θ_1, θ_2) -plane at $\theta_3 = \pi/2$ where $A(2.21, 0.05, \pi/2)$, $B(3.09, 0.93, \pi/2)$, $C(1.36, 2.45, \pi/2)$ and $D(0.69, 1.78, \pi/2)$



(a) Joint space of interest \mathcal{Q}^* in the (θ_1, θ_2) -plane at $\theta_3 = \pi/2$, defined by the quadrilateral $ABCD$ (b) \mathcal{W}^* (dashed), Type 1 singularity loci (solid) and $\text{FKP}(\mathcal{Q}^*)$ (green) projected into the (χ_1, χ_2) -plane

Figure 15: Joint- and workspace of the AsyCoSPM of interest

according to Fig. 15a. Choosing such a plane simplifies the Forward Kinematics studies of CoSPMs:

- First, this physically amounts to immobilize the 3rd leg and check if the robot still reaches all the desired ranges for χ_1 and χ_2 while being indifferent to the value of χ_3 [70].
- Secondly, a plane can easily be scanned using the previous radial path tracking instead of doing the same in three dimensions with redundant results.
- Last but not least, a simple geometry of \mathcal{Q}^* is helpful for the software limitation of the (active) joint velocities discussed in Section 5.

Remark 4 (On the choice of \mathcal{Q}^*). *Note that the choice of a quadrilateral for \mathcal{Q}^* is not mandatory: one could for instance choose any convex (or even nonconvex) polyhedron. This would however complexify the joint velocity limitation later developed in Section.*

Starting from the home configuration $\boldsymbol{\theta} = (\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}) \leftrightarrow \boldsymbol{\chi} = (0, 0, 0)$ i.e. setting the leaf of solution containing $\boldsymbol{\Theta} = (1, 1, 1) \leftrightarrow \boldsymbol{X} = (0, 0, 0)$, with:

- a system precision of $\mathbf{s} = 14$ bits $\simeq 4$ digits (enough to take into account uncertainties related to the design parameter (10^{-4} rad) while having a precision in orientation of the same magnitude);
- a displacement in the joint space of $\delta_\theta \in [10^{-4} \text{ rad}, 10^{-2} \text{ rad}]$, starting from the maximum value,

one applies the radial path tracking over \mathcal{Q}^* with the polynomial system \mathbf{F} . Results are shown in green in Fig. 15b. Note that although the path tracking is done with the polynomial system, results are here converted into physical values (corresponding to the variables of the geometric model \mathbf{f}). One clearly sees that the image of \mathcal{Q}^* through the FKP given the leaf of solution of interest projected in the (χ_1, χ_2) -plane contains \mathcal{W}^* projected in the same plane. This entirely certifies the AsyCoSPM given its application.

4.4. Choice of the design of SPMs

One can also exploit the certified tools used in the last two specific designs of SPMs to study the latter in a more general case. For instance, one could study the influence of each design parameter on the Type 1 and Type 2 singularity loci of SPMs: such a work can be achieved by considering a sensitivity analysis of the singularities w.r.t. the design parameters of interest. Starting from a set of robots that share the same design except for the parameter of interest:

- One computes the Type 1 singularities for each design using the discriminant variety of the IKP, *i.e.* w.r.t. the projection onto the orientation (parameter) space. Such a computation is easily obtained in the case of 3-RRR SPMs, as shown earlier. This certifies the IKP for any leaf of solution, given the theoretical design parameters and delimits a first set in the workspace that is Type 1 singularity-free.
- The Kantorovich test will then be applied over the whole subset previously delimited to ensure (or not) the absence of Type 2 singularities (and numerical instabilities),

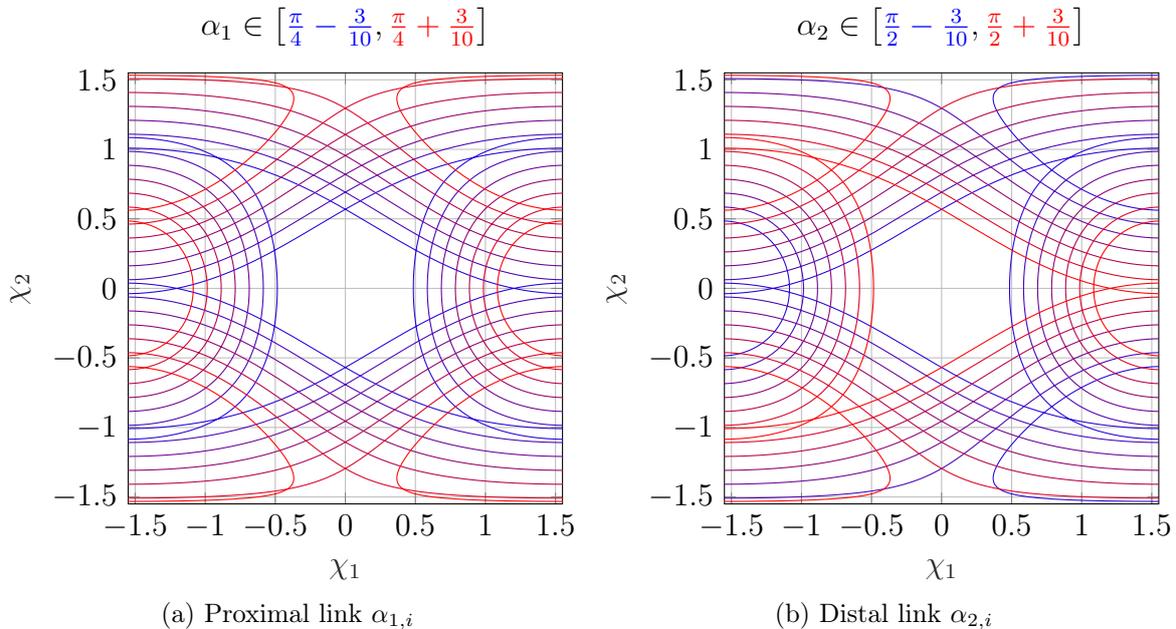


Figure 16: Sensitivity analysis on the projection of Type 1 singularity loci onto the (χ_1, χ_2) -plane (proximal and distal links of the SPM)

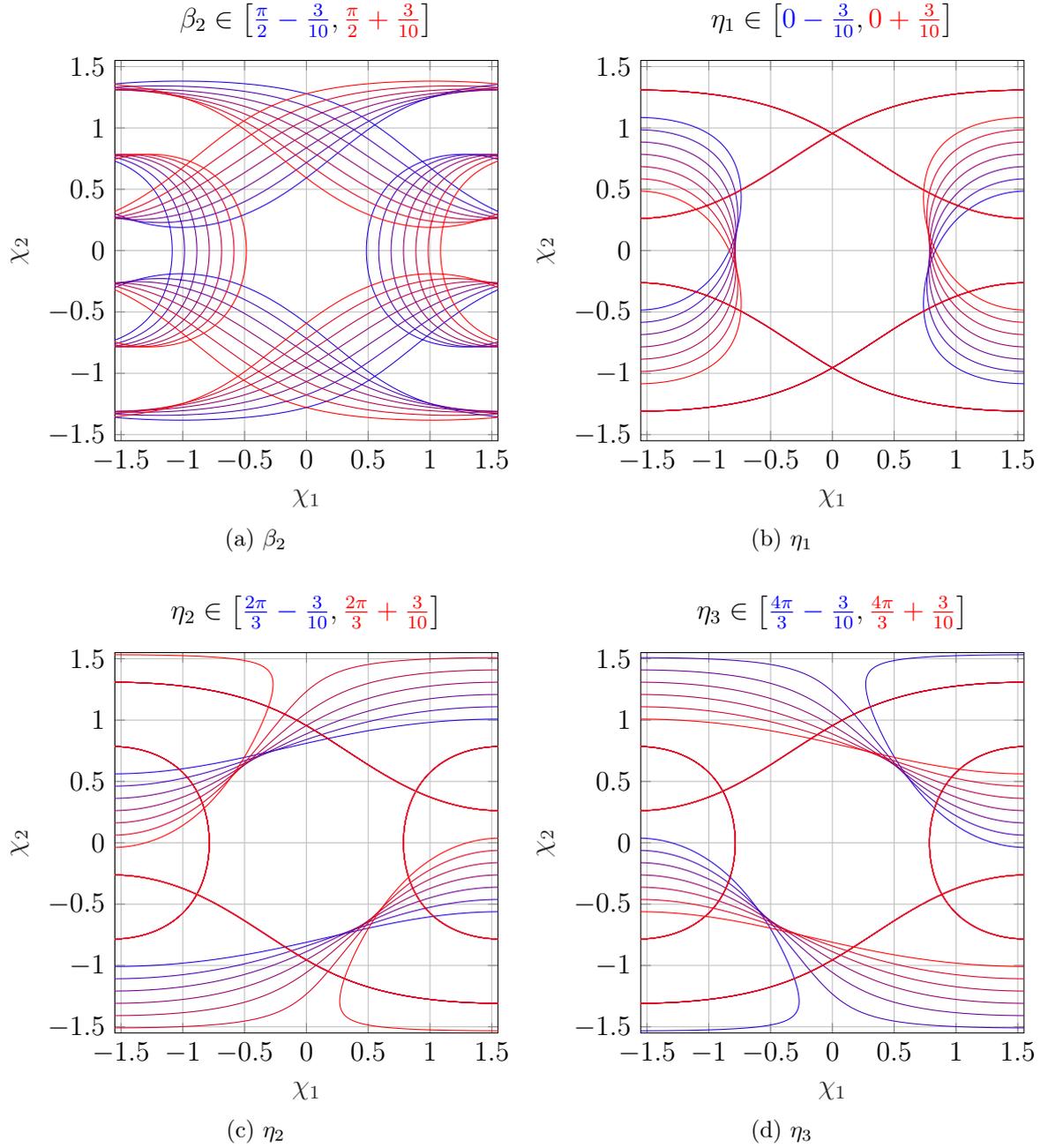


Figure 17: Sensitivity analysis on the projection of Type 1 singularity loci onto the (χ_1, χ_2) -plane for β_2 and η_i , $i \in [1, 3]$

given a leaf of solution and taking into account the desired range of uncertainties on the design parameters.

Remark 5. *If the discriminant variety cannot be obtained (e.g. due to a polynomial system of high total degree), a path tracking in the joint values can be considered. Results are then*

only given for the specific leaf of solution on the contrary of the discriminant variety.

In the sequel, one chooses the CoSPM of Subsection 4.3.2 and performs a sensitivity analysis on each design parameter considering the nominal case with a maximal variation of 0.3 rad. Figure 16 shows the results for the proximal $\alpha_{1,i}$ (Fig. 16a) and distal $\alpha_{2,i}$ (Fig. 16b) links. Figure 16 does the same for the geometry of the upper platform defined by β_2 (Fig. 17a) and the disposition of the i^{th} revolute joint η_i , $i \in \llbracket 1, 3 \rrbracket$ (Fig. 17b–17d).

The angle of the proximal links $\alpha_{1,i}$ plays a significant role in the size of the reachable workspace of the SPM, as observed in [17, 71, 72] with the dexterity criteria based on the global conditioning index of the robot. Choosing a small value of $\alpha_{1,i}$ tends to reduce the reachable workspace (while keeping the symmetric shape of the Type 1 singularity loci if the proximal links are identical from one leg to another). This is however not the case with the distal link $\alpha_{2,i}$ that does not keep the initial symmetric shape, as for the sensitivity analysis on β_2 and the η_i 's shown in Figure 17. On the contrary, a bigger value of $\alpha_{1,i}$ tends to enlarge the reachable workspace so that the latter may be unbounded. This is indeed the case for the design Υ_5 in Fig. 16a (red curve) whose parameters are the same as for the CoSPM of Sub-Section 4.3.2 except that $\alpha_{1,i} = \frac{\pi}{4} + \frac{3}{10}$, $i \in \llbracket 1, 3 \rrbracket$. Such a design seems at first glance attractive since no Type 1 singularity crosses the X_2 -axis (*i.e.* no serial singularity occurs for pure elevation). However, investigating the Forward Kinematics through a path tracking in orientation does not certify that the robot can reach an elevation angle higher than 90° ($X_2 > 1$), as shown in Figure 18. In fact, the Kantorovich test systematically fails

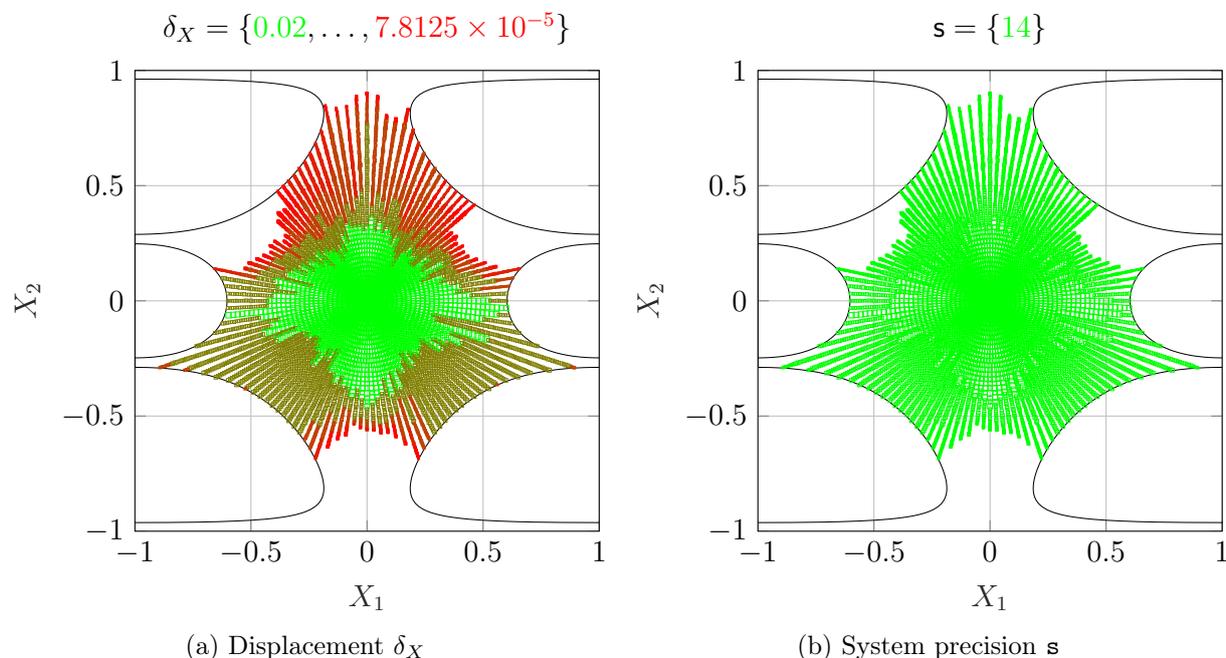


Figure 18: Workspace in the (X_1, X_2) -plane at $X_3 = 0$ of the SPM with design Υ_5 and the certified area obtained by the Kantorovich unicity operator (green: initial best tuning, red: worst case tuning)

in areas where one cannot guarantee that the interval Jacobian matrix w.r.t. the variables is invertible (*i.e.* $0 \ni [\det(\frac{\partial \mathbf{F}}{\partial \mathbf{X}})]$), despite a displacement of $\delta_X = 7.8125 \times 10^{-5}$ and a system precision of $\mathbf{s} = 14$ bits (4 digits). This strongly suggests the existence of Type 2 singularities. Such loci are naturally forbidden areas for the robot in question.

5. Tools for the Control of parallel robots

There are several ways to control the end-effector of parallel robots. One can for instance use a position or speed control. In any case, one cannot always guarantee that the robot will stay in its singularity-free workspace. In order to overcome such an issue, some strategies based on the limitation of the active joint values have been investigated. For instance, one can mention from the literature strategies based on the evaluation of kinematic indices, as for the Orthoglide [73] with an offline evaluation of its conditioning index and velocity transmission factor or as for the CoSPM with an online evaluation of the conditioning index with a link collision detection [11, Algorithm 4]. While the first case benefits from a simple forward kinematics (quadratic polynomials with appropriate variable changes), the second one is rather a saturation of the signals at the boundaries of joint space and does not take into account the feasibility of the motion. For this purpose, an algorithm limiting the joint velocities of the actuators can be implemented in order to ensure that:

- (i) the robot never leaves its certified workspace previously established using the previous kinematic tools;
- (ii) given the physical properties of the actuators.

From the kinematic point of view, the condition (ii) can be translated in terms of maximum joint velocity and acceleration. Such a limitation strategy has already been explored in [70] and applied to the previous AsyCoSPM. The idea here is to highlight the main results and emphasize the global methodology proposed for the design and control of parallel robots.

5.1. General case

We first recall the framework in which the *joint velocity limitation algorithm* is applied. Let $\dot{\mathbf{q}}_c$ and $\dot{\mathbf{q}}_{lim}$ respectively be the active joint command vector and its limited counterpart. The first vector is the output of the controller and an input of the algorithm whereas the last vector is its output that is delivered to the physical system, as illustrated in Figure 19.

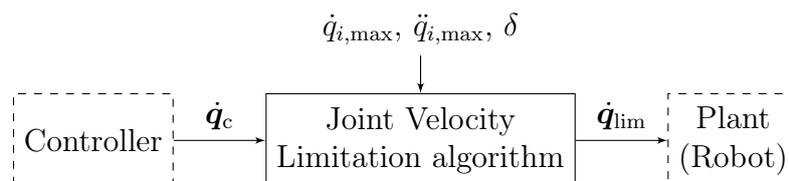


Figure 19: Inputs and outputs of the joint velocity limitation algorithm

Before diving into the algorithm, one denotes the discretization of any time signal $x(t)$ as $x[k] := x(t = kT_e)$, where T_e is the sampling period of the control and k the current sample. Moreover, one defines the saturation signal $\text{sat}(x, \epsilon)$ as

$$\text{sat}(x, \epsilon) := \begin{cases} -\epsilon & (\text{if } x \leq -\epsilon), \\ \epsilon & (\text{if } x \geq \epsilon), \\ x & (\text{otherwise}). \end{cases} \quad (23)$$

The limitation algorithm takes into account the maximal active joint velocities $\dot{\mathbf{q}}_c$ and accelerations $\ddot{\mathbf{q}}_{\text{lim}}$ as well as the distance δ towards the nearest singularity. It is mainly divided into two steps. First, one performs a preliminary saturation of $\dot{\mathbf{q}}_c$ ensuring that each resulting signal $\dot{q}_{\text{sat},i}$ verifies

$$\forall i \in \llbracket 1, n_a \rrbracket, \forall k \in \mathbb{N}, \quad \begin{cases} \dot{q}_{\text{sat},i}[k] = \text{sat}(\dot{q}_{c,i}[k], \dot{q}_{i,\text{max}}) \\ \ddot{q}_{\text{sat},i}[k] = \text{sat}(\ddot{q}_{c,i}[k], \ddot{q}_{i,\text{max}}) \end{cases} \iff \begin{cases} |\dot{q}_{\text{sat},i}[k]| \leq \dot{q}_{i,\text{max}}, \\ |\ddot{q}_{\text{sat},i}[k]| \leq \ddot{q}_{i,\text{max}}. \end{cases} \quad (24)$$

In the case of a speed control, $\ddot{q}_{\text{sat},i}[k]$ can be obtained using the following Euler approximation:

$$\ddot{q}_{\text{sat},i}[k] \simeq \frac{\dot{q}_{c,i}[k] - \dot{q}_{\text{lim},i}[k-1]}{T_e}. \quad (25)$$

Secondly, one computes δ as the distance towards the nearest boundaries of the previously certified workspace and limit the joint velocities w.r.t. it. Let $\dot{\gamma}$ the joint velocity coordinate taken into account for the limitation. The maximal joint velocity and acceleration values can then be expressed as $\dot{\gamma}_{\text{max}}$ and $\ddot{\gamma}_{\text{max}}$. As a result, one applies the joint velocity profile of Figure 20 that can be piecewise-defined w.r.t. δ as:

$$\dot{\gamma}_{\text{lim}} = \begin{cases} \min(\dot{\gamma}, \text{sgn}(\delta)\dot{\gamma}_{\text{max}}) & (\text{if } |\delta| > \delta_\varphi), \\ \min\left(\dot{\gamma}, \text{sgn}(\delta)\sqrt{2\ddot{\gamma}_{\text{max}}(|\delta| - \delta_f)}\right) & (\text{if } \delta_{\text{int}} < |\delta| \leq \delta_\varphi), \\ \min(\dot{\gamma}, K\delta) & (\text{if } |\delta| \leq \delta_{\text{int}}). \end{cases} \quad (26)$$

given that

$$\delta_{\text{int}} := \frac{\ddot{\gamma}_{\text{max}}}{K^2}, \quad (27a)$$

$$\delta_f := \frac{\dot{\gamma}_{\text{max}}}{2K^2} = \frac{\delta_{\text{int}}}{2}, \quad (27b)$$

$$\delta_\varphi := \delta_f + \frac{(\dot{\gamma}_{\text{max}})^2}{2\ddot{\gamma}_{\text{max}}}, \quad (27c)$$

are the several thresholds of the profile depending on $\dot{\gamma}_{\text{max}}$, $\ddot{\gamma}_{\text{max}}$ and K being the tuning parameters. More precisely:

- $[0, \delta_{\text{int}}]$ is the closest zone to the boundaries of the prescribed joint space in which the

limited joint speed is proportional to δ , such that

$$-K\delta \leq \dot{\gamma}_{\text{lim}} \leq K\delta, \quad (28)$$

where K is a positive gain. As δ is small, this allows the joint velocity to reach 0 without abusively using the actuators. The parameter δ_{int} defines the *linear zone* threshold.

- $[\delta_{\text{int}}, \delta_\varphi]$ is the *maximal deceleration* zone defined by the parameters δ_f and δ_φ .

In this zone, the joint acceleration is defined by $\ddot{\gamma} = -\ddot{\gamma}_{\text{max}}$ and the joint velocity is bounded by

$$|\dot{\gamma}_{\text{lim}}| \leq \sqrt{2\ddot{\gamma}_{\text{max}} (|\delta| - \delta_f)}. \quad (29)$$

Such a condition allows the robot to stop with a zero speed.

- $\delta > \delta_\varphi$ is the *nominal* zone in which the joint position is considered to be sufficiently far from any singularity. The only limitations are then based on the performance of the actuators so that

$$|\dot{\gamma}_{\text{lim}}| \leq \dot{\gamma}_{\text{max}}. \quad (30)$$

Finally, having $\dot{\gamma}_{\text{lim}}$ gives $\dot{\mathbf{q}}_{\text{lim}}$ the output of the joint velocity limitation algorithm depending on which coordinate(s) we applied the limitation. It must be kept in mind that the choice of the thresholds δ_{int} and δ_φ is crucial since they define a buffer zone in which the limitation algorithm may downgrade the performances of the controller in order to secure the robot.

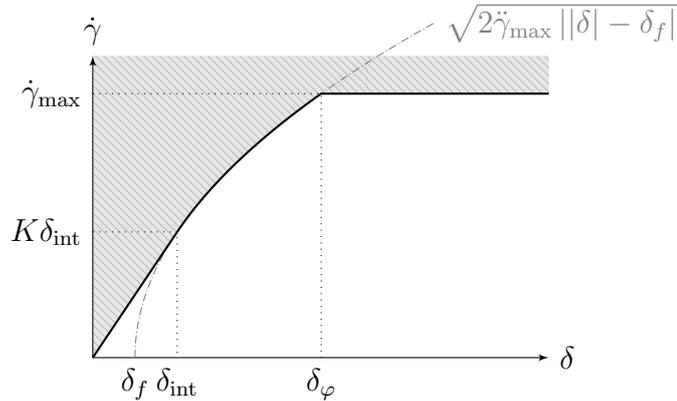


Figure 20: Joint velocity profile w.r.t. the distance to the nearest stop δ

5.2. Case of the AsyCoSPM

5.2.1. Implementation

In the sequel, one implements the joint velocity limitation algorithm previously stated in the general case to the AsyCoSPM. As it has been shown that the third orientation

coordinate χ_3 is singularity-free, moving along the corresponding axis (in the sequel called *invariance axis*) preserves the distance towards the nearest singularity. Given [8, Prop. 1], the latter can be translated into the joint space as $\theta_1 = \theta_2 = \theta_3$ as illustrated in Figure 21.

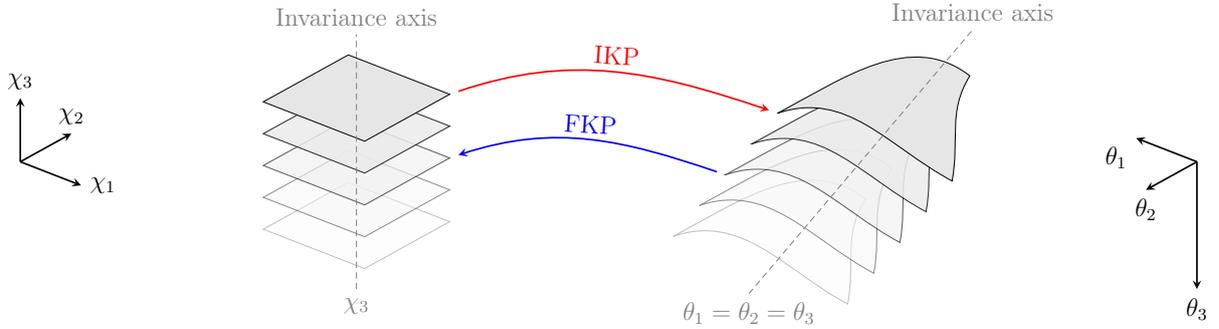


Figure 21: Operational and joint spaces of CoSPMs

Applying the previous algorithm while exploiting the entire joint and workspace of interest (*i.e.* without limiting the bearing motion in χ_3) requires considering auxiliary joint coordinates $\mathbf{q} := (q_1, q_2, q_3)$ that brings the invariance to one of the components. As shown in [70, Prop. 1], choosing the third one q_3 leads to the rotational mapping $\boldsymbol{\theta} \mapsto \mathbf{q} = \mathbf{Q}\boldsymbol{\theta}$ defined by

$$\mathbf{Q} := \mathbf{R}_y\left(-\arctan\left(\frac{\sqrt{2}}{2}\right)\right) \mathbf{R}_x\left(\frac{\pi}{4}\right), \quad (31)$$

where $\mathbf{R}_x(\cdot)$, $\mathbf{R}_y(\cdot)$ and $\mathbf{R}_z(\cdot)$ denote the usual elementary rotation matrices shown in (B.2a), (B.2b) and (B.2c). By doing so, the entire joint space of interest \mathcal{Q}^* can be represented in the (q_1, q_2) plane, with $q_3 \in \mathbb{R}$ (see Figure 22).

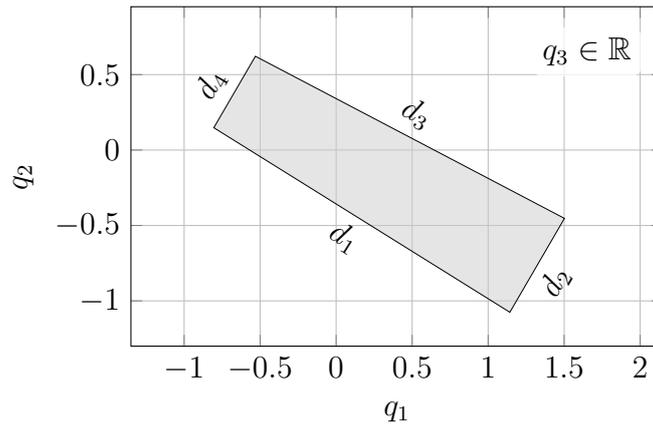


Figure 22: Regular joint space \mathcal{Q}^* represented in the (q_1, q_2) -plane with q_3 invariant

In our case, the stops d_i are defined by the following equations of the form

$$\forall i \in \llbracket 1, 4 \rrbracket, \quad d_i: a_i q_1 + b_i q_2 + c_i = 0, \quad (32)$$

where

$$\begin{aligned} a_1 &= 0.624, & b_1 &= 1, & c_1 &= 0.353, \\ a_2 &= -1.72, & b_2 &= 1, & c_2 &= 3.05, \\ a_3 &= -0.528, & b_3 &= -1, & c_3 &= 0.346, \\ a_4 &= 1.72, & b_4 &= -1, & c_4 &= 1.54. \end{aligned}$$

Remark 6. The signs of the coefficients a_i , b_i and c_i are here chosen such that

$$\forall i \in \llbracket 1, 4 \rrbracket, d_i \geq 0 \iff (q_1, q_2) \in \mathcal{Q}^*.$$

Lê et al. [70] chose to limit the *normal* velocity w.r.t. the nearest stop that defines the boundaries of \mathcal{Q}^* . This requires to make the transformations

$$\forall j \in \llbracket 1, n_s \rrbracket, \begin{bmatrix} \dot{q}_j^\perp \\ \dot{q}_j^\parallel \\ \dot{q}_3 \end{bmatrix} = \mathbf{R}_z(\zeta_j) \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}, \quad (33)$$

where \dot{q}_j^\perp denotes the normal velocity w.r.t. the j^{th} stop, \dot{q}_j^\parallel the tangential component and $n_s = 4$ the number of stops taken into account given that \mathcal{Q}^* is a quadrilateral in the (q_1, q_2) -plane (see Remark 4).

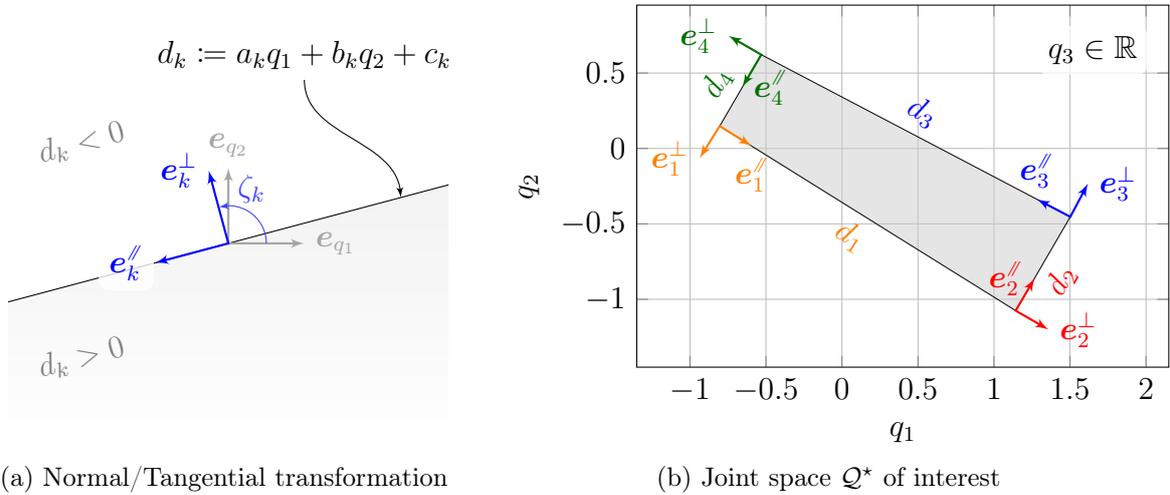


Figure 23: Normal and tangential transformation applied to the joint space

Figure 23 shows the normal/tangential transformation on the joint space \mathcal{Q}^* . The angles ζ_i are in our case given by $\zeta_1 = -2.13$ rad, $\zeta_2 = -0.53$ rad, $\zeta_3 = 1.085$ rad and $\zeta_4 = 2.61$ rad. As each stop d_j is defined by $d_j : a_j q_1 + b_j q_2 + c_j = 0$, the *relative* distance δ_j between a configuration $\mathbf{q}^* := (q_1^*, q_2^*, q_3^*)$ and a stop d_j is then given by

$$\delta_j(q_1^*, q_2^*) = \frac{a_j q_1^* + b_j q_2^* + c_j}{\sqrt{a_j^2 + b_j^2}}. \quad (34)$$

Let k be the index of the nearest stop d_k . The shortest distance to the joint space boundary is then given by $\delta = \delta_k$. As the normal component of the joint velocity is the only one to be considered in the joint velocity limitation profile (26) *i.e.* $\dot{\gamma} := \dot{q}_k^\perp$, the tangential one can for instance be proportionally limited as

$$\dot{q}_{k,\text{lim}}^\parallel = \begin{cases} \frac{\dot{q}_k^\parallel \dot{q}_{k,\text{lim}}^\perp}{\dot{q}_k^\perp} & (\text{if } \dot{q}_k^\perp \neq 0), \\ \dot{q}_k^\parallel & (\text{otherwise}). \end{cases} \quad (35)$$

This disables any motion w.r.t. bank χ_1 or elevation χ_2 once the robot reaches the boundaries of its certified workspace. Note that the component \dot{q}_3 describing the motion in bearing is the only one that is not limited w.r.t. the certified workspace. Finally, the limited joint velocities can be expressed using the original coordinates so that

$$\dot{\theta}_{\text{lim}} = \begin{bmatrix} \dot{\theta}_{\text{lim},1} \\ \dot{\theta}_{\text{lim},2} \\ \dot{\theta}_{\text{lim},3} \end{bmatrix} = \mathbf{R}_z^\top(\zeta_k) \mathbf{Q}^\top \begin{bmatrix} \dot{q}_{k,\text{lim}}^\perp \\ \dot{q}_{k,\text{lim}}^\parallel \\ \dot{q}_3 \end{bmatrix}. \quad (36)$$

5.2.2. Example of simulation

As in [70], we implement the joint velocity limitation algorithm given the principles of the diagram of Figure 24 using MATLAB 2022a & Simulink, along with a CATIA 3D model of the AsyCoSPM and interface it with Simulink using the Simscape framework.

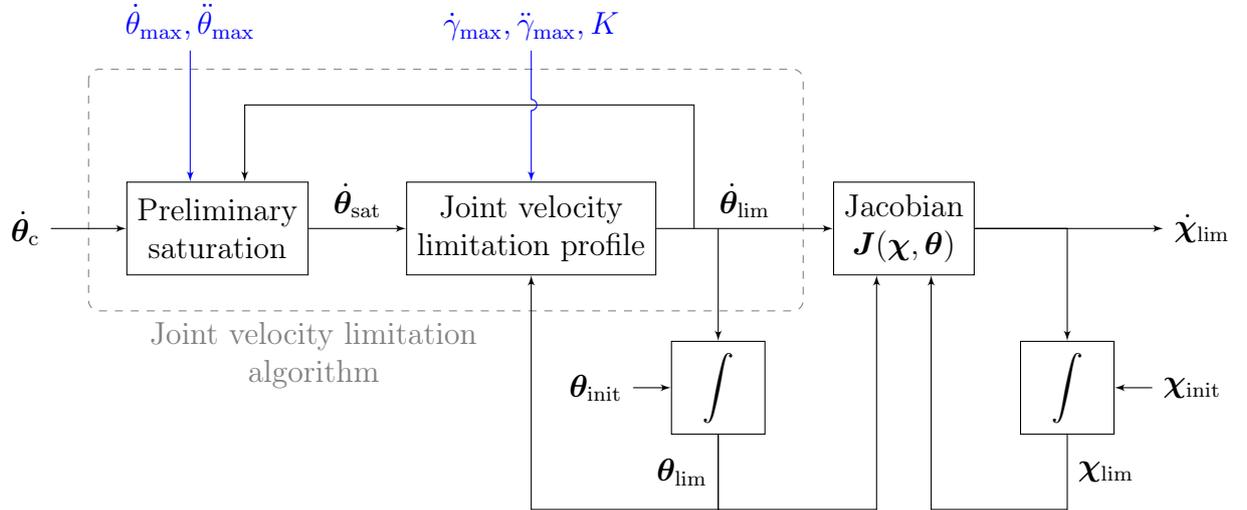


Figure 24: Diagram of the joint velocity algorithm implemented in the control of the AsyCoSPM

For this specific simulation, the AsyCoSPM is modelled by its kinematic Jacobian matrix \mathbf{J} whose expression is given in [8]. This matrix takes as input the limited joint velocity $\dot{\theta}_{\text{lim}}$ returned by the algorithm and also depends on the current (limited) orientation $\chi = \chi_{\text{lim}}$ and active joint values $\theta = \theta_{\text{lim}}$. The latter values are obtained by integrating their time

derivative counterparts with the initial conditions being the home position described earlier.

The control is discretized using a sampling period of $T_e = 1$ ms. The joint velocity command vector $\dot{\theta}_c := (\dot{\theta}_{c,1}, \dot{\theta}_{c,2}, \dot{\theta}_{c,3})$ is defined such that

$$\dot{\theta}_{c,1} = \begin{cases} 0 & (t \leq 0) \\ -0.5 & (0 < t \leq 2.5) \\ 0.5 & (t > 2.5) \end{cases}, \quad \dot{\theta}_{c,2} = -\dot{\theta}_{c,1}, \quad \dot{\theta}_{c,3} = \begin{cases} 0 & (t \leq 3) \\ 0.4 & (t > 3) \end{cases}$$

which corresponds to a threefold scenario in which the end-effector:

1. makes a pure motion in elevation and goes towards the well-known Type 1 singularity locus at $\chi_2 = 90^\circ$ (for $t \in [0, 2.5]$ seconds);
2. tries to go back to the initial home configuration (for $t \in [2.5, 3]$ seconds);
3. while drifting from $t > 3$ s and thus leaving the certified workspace.

Note that the joint velocity command signals $\theta_{c,i}(t)$ are intentionally discontinuous which violates the maximum velocity and acceleration allowed for the actuators. Recall that initial home configuration is set such that the end-effector orientation is described by $\chi_i = 0$ with the actuators at $\theta_i = \frac{\pi}{2}$, $\forall i \in \llbracket 1, 3 \rrbracket$. We also choose two tuning sets parametrizing the algorithm that are detailed in Table 4.

Tuning set	K	$\ddot{\gamma}_{\max}$ (rad/s ²)	$\dot{\gamma}_{\max}$ (rad/s)	δ_f (rad)	δ_{int} (rad)	δ_φ (rad)
A	50	$4\sqrt{2}$	$1\sqrt{2}$	10^{-3}	2×10^{-3}	0.17
B	10	$1\sqrt{2}$	$1\sqrt{2}$	0.014	7.1×10^{-3}	0.714

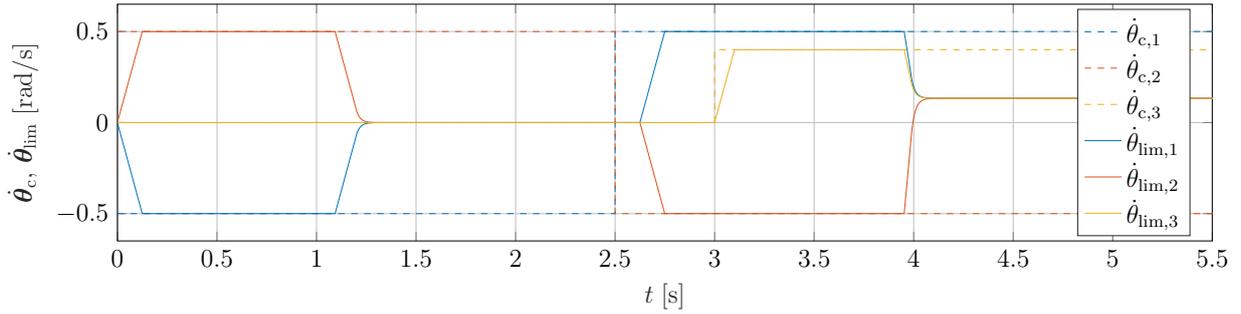
Table 4: Set of the parameters related to the joint velocity limitation algorithm tested in the simulation

For purpose of simplification (see [70, p. 48]), the parameters $\dot{\gamma}_{\max}$ and $\ddot{\gamma}_{\max}$ are assumed to be linked by the relationship

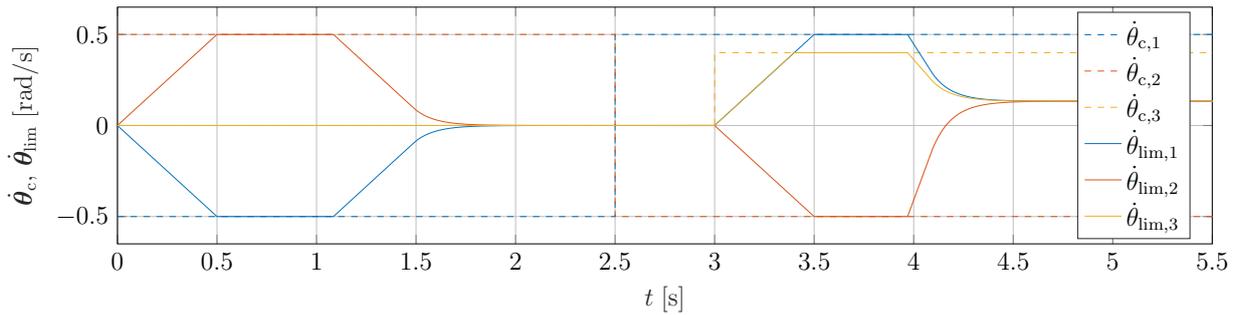
$$\frac{\dot{\gamma}_{\max}}{\dot{\theta}_{\max}} = \frac{\ddot{\gamma}_{\max}}{\ddot{\theta}_{\max}} = \sqrt{2}. \quad (37)$$

Figure 25 shows the joint velocities signals (reference $\dot{\theta}_c$ and limited $\dot{\theta}_{\text{lim}}$) for the tuning set A (Fig. 25a) and B (Fig. 25b). Figure 27 does the same for the auxiliary joint velocities \dot{q}_k^\perp and \dot{q}_k^{\parallel} , where k denotes the index of the nearest stop (results for tuning set A are shown in Fig. 27a and those for tuning set B in Fig. 27b). For both tuning sets, the constraints on the maximum joint velocity and acceleration is taken into account. Indeed, despite the discontinuity of the reference signals, the limited versions are continuous starting with a slope (in absolute value) of 4 rad/s^2 for tuning set A and 1 rad/s^2 for tuning set B. Such values correspond to the maximum joint acceleration $\ddot{\theta}_{\max}$ allowed given (37) and Table 4. Moreover, whenever the robot tries to leave its certified workspace, the joint velocity

limitation algorithm ensures that it decelerates near the boundaries. It is in this respect interesting to compare the results with tuning sets A et B as the first one generates a much thinner safe zone than the second. Given that the threshold δ_φ for the velocity limitation is higher for the tuning set B, the algorithm tends to judge that the robot is less often in the nominal zone as seen in Fig. 27b where it is actually never in the nominal zone (labelled 3) unlike with the tuning set A. As a result, the performances of the robot may be much more

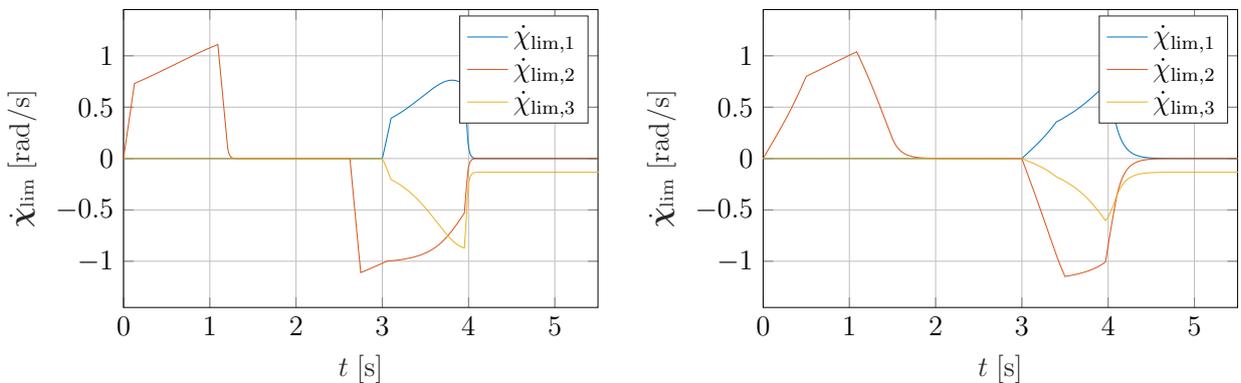


(a) for tuning set A

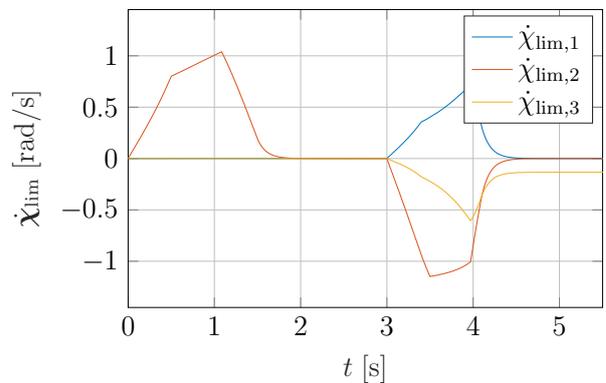


(b) for tuning set B

Figure 25: Joint velocities (reference and limited values)

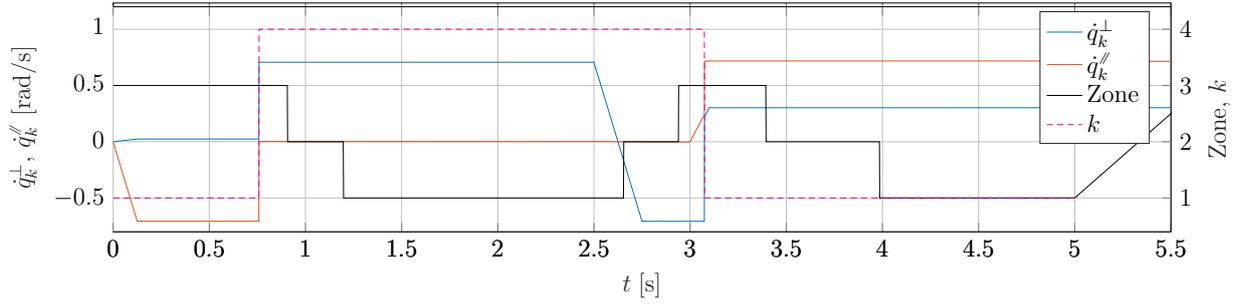


(a) for tuning set A

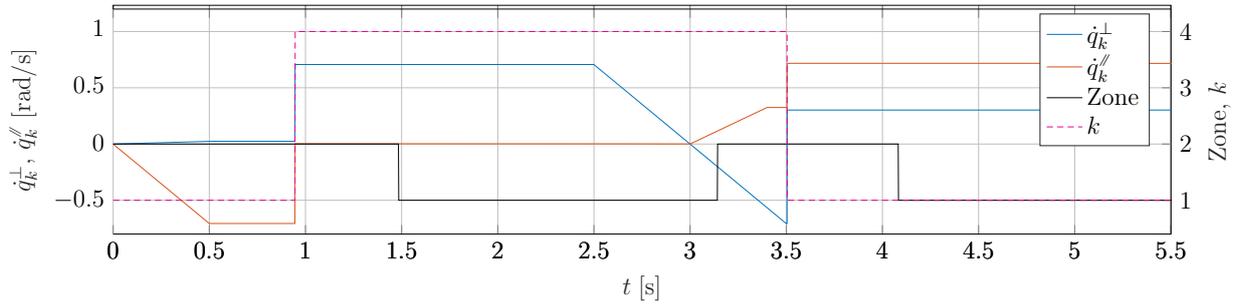


(b) for tuning set B

Figure 26: Angular rates $\dot{\chi}_{lim} := \mathbf{J}\dot{\theta}_{lim}$



(a) for tuning set A

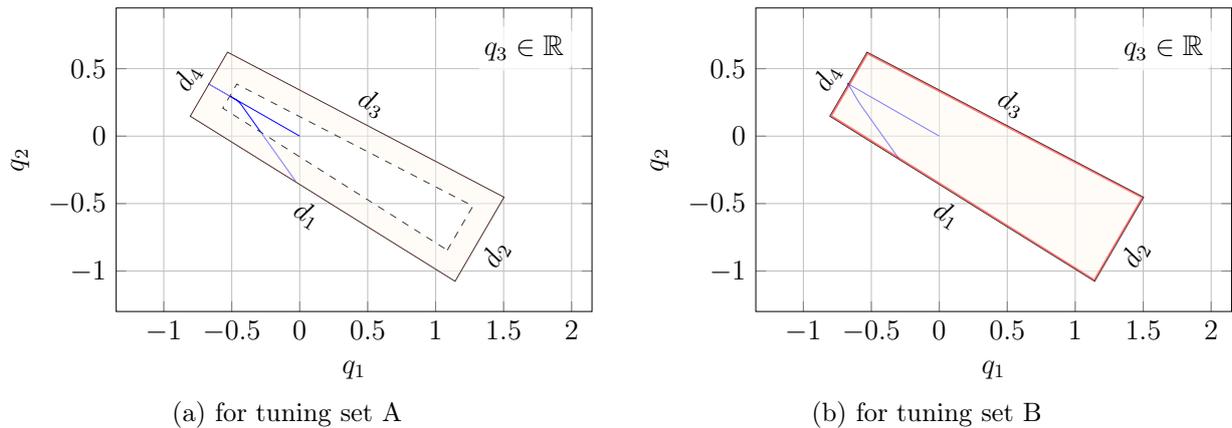


(b) for tuning set B

Figure 27: Normal and tangential joint velocities w.r.t. k^{th} stop and zone flag (3: nominal, 2: maximal deceleration, 1: linear)

limited as δ_φ is high. In the same time, having a low maximal acceleration and velocity also downgrades the performances of the robot as the latter takes much more time to decelerate and stop (~ 125 ms for simulation A vs ~ 625 ms for simulation B).

Furthermore, implementing the joint limitation algorithm in the AsyCoSPM led to con-



(a) for tuning set A

(b) for tuning set B

Figure 28: Trajectory of the end-effector in the (q_1, q_2) -plane (orange area: maximal deceleration zone, red area: linear zone)

sider an additional kinematic property (a pure bearing motion will never lead to a singular configuration and thus $\dot{\chi}_3$ should not vanish if initially nonzero). This is the case in both simulations as the only nonvanishing component is $\dot{\chi}_3$ at the end of the scenario, as seen in Figure 26 with $\dot{\chi}_3 \simeq -0.13$ rad/s. Finally, Figure 28 shows the trajectory of the end-effector projected into the (q_1, q_2) -plane for the simulation with the tuning sets A (Fig. 28a) and B (Fig. 28b). Despite the fact that in both cases the robot does not leave the previously delimited certified workspace, the trajectories are different. For simulation B, this is due to a lower capacity in terms of velocity and acceleration and thus to follow the command that is subject to the preliminary saturation defined by $\dot{\theta}_{\text{sat},i}[k] = \text{sat}(\dot{\theta}_i[k], \dot{\theta}_{i,\text{max}})$ and $\ddot{\theta}_{\text{sat},i}[k] = \text{sat}(\ddot{\theta}_i[k], \ddot{\theta}_{i,\text{max}})$. This can be visually confirmed using the digital model of the AsyCoSPM whose positions are displayed at the end of the first ($t = 2.23$ s) and last ($t = 5$ s) sequences of the simulation in the Figure 29.

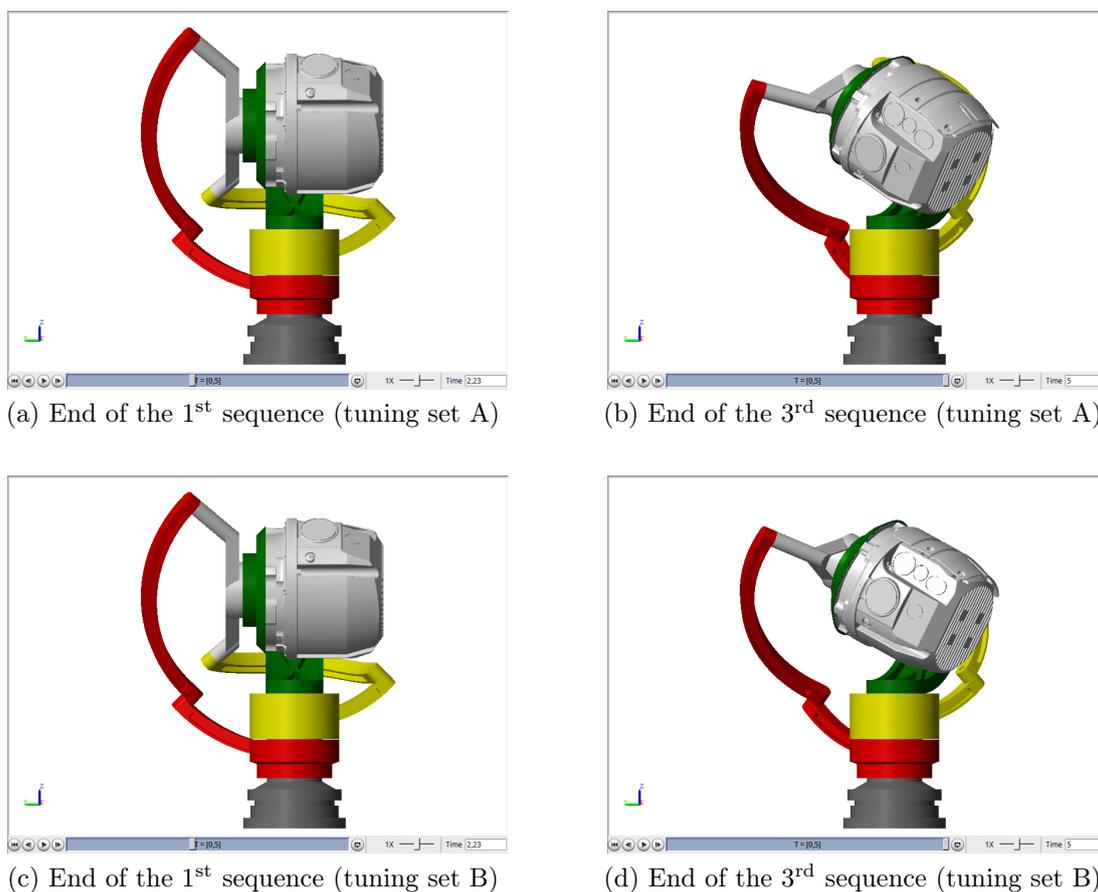


Figure 29: Orientation of the AsyCoSPM at the end of the first and last sequences of the simulation

Note that for both tuning sets A and B, the AsyCoSPM escaped a Type 1 singularity in pure elevation as the red links are not entirely unfolded.

6. Conclusion and outlooks

6.1. Conclusion

This paper proposed several tools related to the design and control of parallel robots. On the one hand, one used certified computational kinematic tools that ensures an appropriate behavior of the mechanism of interest (absence of physical or numerical instabilities) while coping with uncertainties. Two complementary approaches were used. First, the discriminant variety provides good information on the regular areas for all the existing leaves of solution. However, such a symbolic tool only certifies the theoretical robot and does not take into account the possible deformations of the robot due to the heterogeneous uncertainties related to the design (parameters) as well as the control (information of the sensors, delays). Such an unavoidable phenomenon is however considered with a semi-numerical strategy but only for the leaf of solution of interest. This second strategy relies on certifying the regularity of continuation methods using interval arithmetic with multiple precision computations. On the other hand, one ensures that the robot never leaves its certified workspace through a joint velocity limitation algorithm that also takes into account the kinematic properties of the actuators. All these tools constitute a cornerstone for a global methodology on the use of a parallel robot given an application whose flowchart is detailed in Figure 30. The decision blocs “OK” must be understood as if the obtained results are in accordance with the specifications of the robot or the requirements of the application. An important remark can be made on “No” wires of this flowchart. The first one related to the design part implies that the obtained certified joint and workspaces do not meet the initial criteria. This leads to two options:

1. either the user chooses another set of design parameters by modifying those of interest in accordance to the specifications (a sensitivity analysis using the symbolic tools can in this respect be helpful to do so, *e.g.* finding a design parameter that is critical in the size of the reachable workspace as the proximal links $\alpha_{1,i}$ for SPMs);
2. either the user concedes to downgrade the initial specifications (*e.g.* consider a smaller workspace).

This logic also applies to the control part, where the “No” wire indicates poor performance with the limitation algorithm (*e.g.* the physical characteristics of the actuators are insufficient), even though the design part is satisfied. In each case, there is a trade-off between the specifications (*e.g.* size of the workspace, expected behavior) and the actual performance. An impossible trade off simply means that the chosen robot is not suitable for the application of interest.

6.2. Future works

Part of future works can be divided into two parts. First, one will focus on a better characterization of the uncertainties by directly taking into account the propagation of the uncertainties instead of a rough and homogeneous estimation of the latter through the coefficients of the polynomials. The latter are indeed in most cases either monomials or

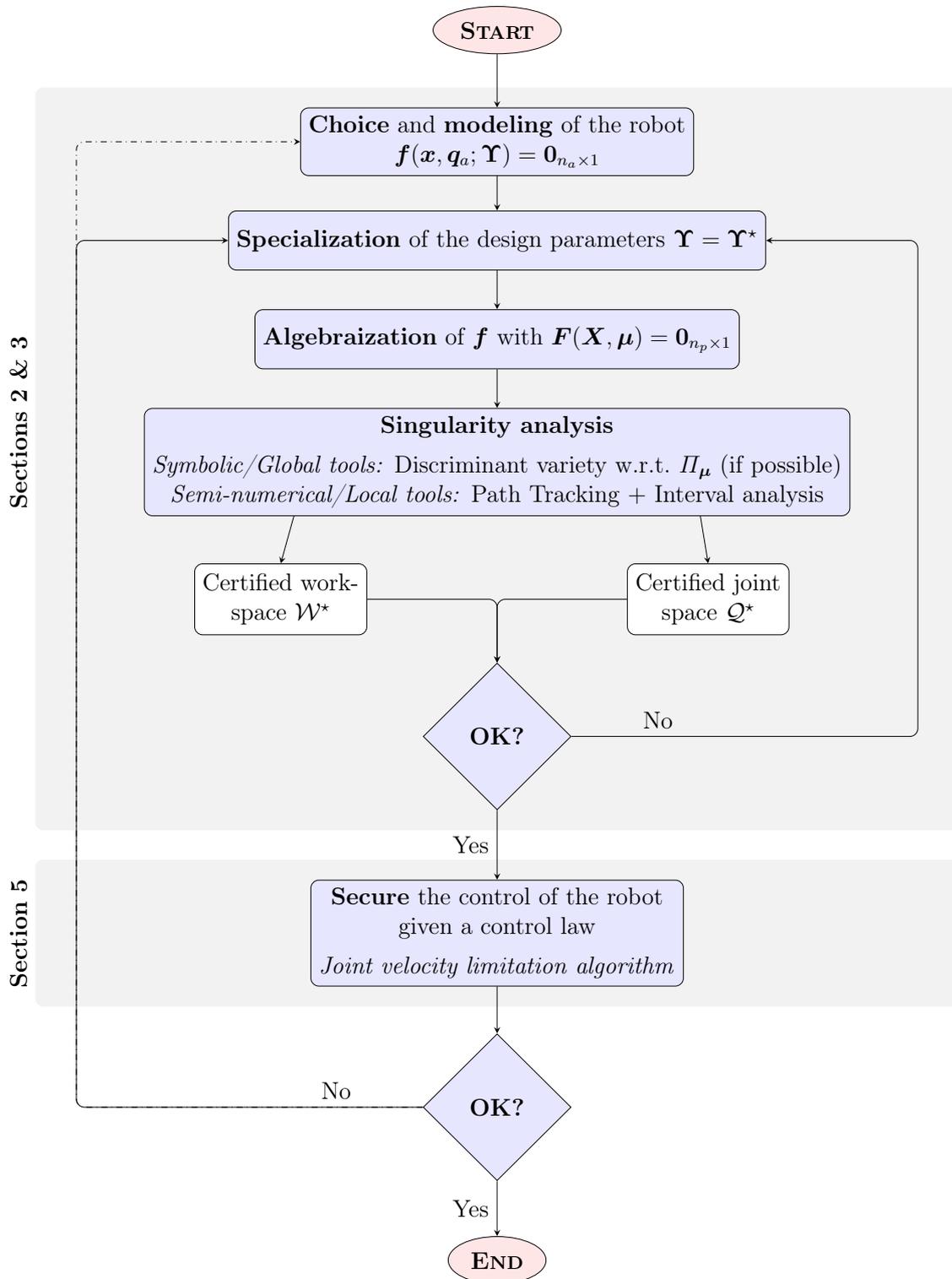


Figure 30: Methodology on the use of the use of a parallel robot given an application

products of sine or cosine of design parameters. The methodology of certification of the workspace can also be applied and adapted to more complex parallel robots such as 6-DOF 6-UPS hexapods.

Secondly, the strategy to secure the control of a robot applied to the AsyCoSPM arises some issues. As it relies on the commutation between one stop to another, smoothness is not entirely ensured especially at the vertices and bisectors of the certified joint space \mathcal{Q}^* . Moreover, further studies should address dynamic aspects related to torque management, such as considering the maximum torque limits.

Funding Data

This research was supported by ANRT CIFRE grant n°2021/1272 which funded the first author's doctoral studies.

Appendix A. Kinematic modeling of Tripods

Taking only into account the operation mode $q_z = 0$ as well as the conventions used to described the operational and active joint variables of the tripod, its geometric model writes

$$\begin{aligned} f_1 = & 4h^2q_w^4 + 12h^2q_w^2q_x^2 + 9h^2q_x^4 - 6h^2q_x^2q_y^2 + h^2q_y^4 - 4ghq_w^2 \\ & - 6ghq_x^2 + 2ghq_y^2 - 4h^2q_w^2 - 6h^2q_x^2 + 2h^2q_y^2 \\ & - 4hq_wq_yz + g^2 + 2gh + h^2 + z^2 - \rho_1^2, \end{aligned} \quad (\text{A.1a})$$

$$\begin{aligned} f_2 = & -6\sqrt{3}h^2q_w^2q_xq_y - 8\sqrt{3}h^2q_xq_y^3 + 4\sqrt{3}hq_xq_yg + 4\sqrt{3}h^2q_xq_y \\ & + 2\sqrt{3}hzq_wq_x + 4h^2q_w^4 + 3h^2q_w^2q_x^2 + 9h^2q_w^2q_y^2 + 12h^2q_x^2q_y^2 \\ & + 4h^2q_y^4 - 4h^2q_w^2 - 4h^2q_y^2 + h^2 - 4ghq_w^2 - 4ghq_y^2 \\ & + 2hq_wq_yz + 2gh + g^2 + z^2 - \rho_2^2, \end{aligned} \quad (\text{A.1b})$$

$$\begin{aligned} f_3 = & 6\sqrt{3}h^2q_w^2q_xq_y + 8\sqrt{3}h^2q_xq_y^3 - 4\sqrt{3}hq_xq_yg - 4\sqrt{3}h^2q_xq_y \\ & - 2\sqrt{3}hzq_wq_x + 4h^2q_w^4 + 3h^2q_w^2q_x^2 + 9h^2q_w^2q_y^2 + 12h^2q_x^2q_y^2 \\ & + 4h^2q_y^4 - 4h^2q_w^2 - 4h^2q_y^2 + h^2 - 4ghq_w^2 - 4ghq_y^2 \\ & + 2hq_wq_yz + 2gh + g^2 + z^2 - \rho_3^2, \end{aligned} \quad (\text{A.1c})$$

$$f_4 = q_w^2 + q_x^2 + q_y^2 - 1. \quad (\text{A.1d})$$

Appendix B. Kinematic modeling of Spherical Parallel Manipulators

Appendix B.1. Description of the end-effector orientation

The end-effector orientation is described using the Tait-Bryan ZYX rotation sequence shown in Figure B.31.

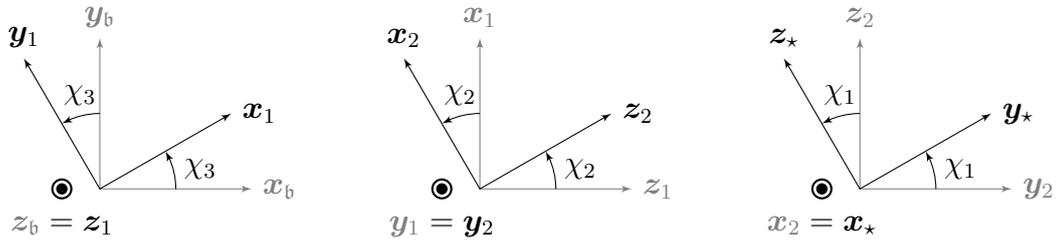


Figure B.31: Rotation order using the Euler Tait-Bryan ZYX convention

Starting from the base frame $\mathcal{F}_b := (O, \mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b)$, a rotation w.r.t. \mathbf{z}_b by a bearing angle χ_3 gives the frame $\mathcal{F}_1 := (O, \mathbf{x}_1, \mathbf{y}_1, \mathbf{z}_b)$. A second rotation w.r.t. \mathbf{y}_1 by an elevation angle χ_2 gives the frame $\mathcal{F}_2 := (O, \mathbf{x}_2, \mathbf{y}_2, \mathbf{z}_2)$. Finally, a last rotation around the \mathbf{x}_2 that is in fact the LOS axis \mathbf{x}_* gives the frame of the upper platform $\mathcal{F}_* := (O, \mathbf{x}_2, \mathbf{y}_*, \mathbf{z}_*)$.

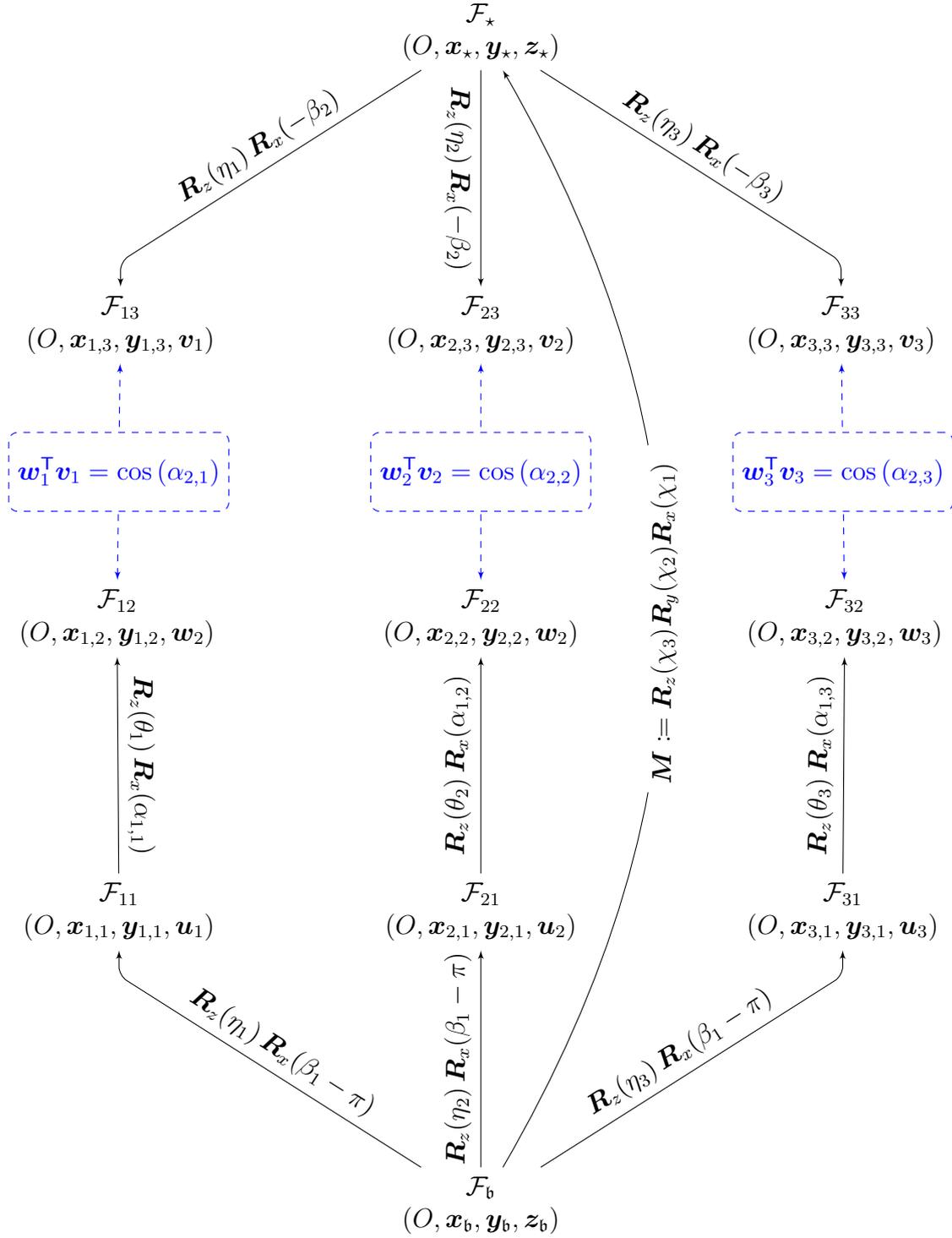


Figure B.32: Kinematic chain of a general SPM

Appendix B.2. Details on the expressions of the unit vectors

Those vectors must be expressed in the same frame. For convenience, we will express all the vectors and coordinates in the base frame $\mathcal{F}_b := (O, \mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b)$. Based on the convention used in [3] and [74], one can obtain the unit vectors \mathbf{u}_i , \mathbf{w}_i , and \mathbf{v}_i , with $i \in \llbracket 1, 3 \rrbracket$ by exploiting the kinematic chain of the robot depicted in Figure B.32, given that:

- $\mathcal{F}_\star := (O, \mathbf{x}_\star, \mathbf{y}_\star, \mathbf{z}_\star)$ is the frame attached to the upper platform;
- \mathcal{F}_{ij} is the frame attached to the j^{th} revolute joint (starting from the base) of the i^{th} kinematic chain, with $\mathbf{z}_{i,1} = \mathbf{u}_i$, $\mathbf{z}_{i,2} = \mathbf{w}_i$ et $\mathbf{z}_{i,3} = \mathbf{v}_i$ being the unit vectors normal to each revolute joint.

Given that $\mathbf{z}_b := (0, 0, 1)$, one can show that they can be written in the reference frame $\mathcal{F}_b := (O, \mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b)$ as:

$$\begin{aligned} {}^b\mathbf{u}_i &= \mathbf{R}_z(\eta_i) \mathbf{R}_x(\beta_1 - \pi) \mathbf{z}_b, \\ {}^b\mathbf{w}_i &= \mathbf{R}_z(\eta_i) \mathbf{R}_x(\beta_1 - \pi) \mathbf{R}_z(\theta_i) \mathbf{R}_x(\alpha_{1,i}) \mathbf{z}_b, \\ {}^b\mathbf{v}_i &= \mathbf{R}_z(\chi_3) \mathbf{R}_y(\chi_2) \mathbf{R}_x(\chi_1) \mathbf{R}_z(\eta_i) \mathbf{R}_x(-\beta_2) \mathbf{z}_b, \end{aligned} \tag{B.1}$$

where \mathbf{R}_x , \mathbf{R}_y and \mathbf{R}_z denote the rotation matrices around their respective local axis. The latter matrices are defined as follows:

$$\mathbf{R}_x(\chi_1) := \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\chi_1) & -\sin(\chi_1) \\ 0 & \sin(\chi_1) & \cos(\chi_1) \end{bmatrix} \tag{B.2a}$$

$$\mathbf{R}_y(\chi_2) := \begin{bmatrix} \cos(\chi_2) & 0 & \sin(\chi_2) \\ 0 & 1 & 0 \\ -\sin(\chi_2) & 0 & \cos(\chi_2) \end{bmatrix} \tag{B.2b}$$

$$\mathbf{R}_z(\chi_3) := \begin{bmatrix} \cos(\chi_3) & -\sin(\chi_3) & 0 \\ \sin(\chi_3) & \cos(\chi_3) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B.2c}$$

Remark 7. *The disposition of the pivot linkage η_i , $i \in \llbracket 1, 3 \rrbracket$ is supposed to be identical for the base and the upper platform, as in [3]. However, as our robot has coaxial input shafts ($\beta_1 = 0$), such parameters does not appear “physically” but still in the equations.*

The knowledge of these vectors is sufficient to obtain the geometric model of any SPM given its design Υ^\star , i.e. $\mathbf{f}(\boldsymbol{\chi}, \boldsymbol{\theta}; \Upsilon = \Upsilon^\star)$. The latter system and its algebraic version $\mathbf{F}(\boldsymbol{\chi}, \boldsymbol{\theta}; \Upsilon = \Upsilon^\star)$ can be obtained using symbolic computation software such as MAPLE for instance. These expressions for the CoSPM under study are tedious and therefore are not shown in this article.

References

- [1] T. Leinonen, Terminology for the theory of machines and mechanisms, *Mechanism and Machine Theory* 26 (1991) 435–539. doi:10.1016/0094-114x(91)90003-m.
- [2] J.-P. Merlet, *Parallel Robots (Second Edition)*, Springer, 2006.
- [3] A. Lê, F. Rouillier, G. Rance, D. Chablat, On the Certification of the Kinematics of 3-DOF Spherical Parallel Manipulators, *Maple Transactions* 3 (2023). doi:10.5206/mt.v3i2.15660.
- [4] W. Khalil, S. Briot, *Dynamics of Parallel Robots : From Rigid Bodies to Flexible Elements*, Springer, 2015.
- [5] R. Clavel, Conception d'un robot parallèle rapide à 4 degrés de liberté, Ph.D. thesis, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, 1991. doi:https://doi.org/10.5075/epfl-thesis-925.
- [6] T. Taunyzov, M. Rubagotti, A. Shintemirov, Constrained orientation control of a spherical parallel manipulator via online convex optimization, *IEEE/ASME Transactions on Mechatronics* 23 (2018) 252–261. doi:10.1109/TMECH.2017.2774245.
- [7] S. Ansari-Rad, M. Zarei, M. G. Tamizi, S. Mohammadi Nejadi, M. T. Masouleh, A. Kalhor, Stabilization of a Two-DOF Spherical Parallel Robot via a Novel Adaptive Approach, in: *2018 6th RSI International Conference on Robotics and Mechatronics (ICRoM)*, 2018, pp. 369–374. doi:10.1109/ICRoM.2018.8657566.
- [8] A. Lê, F. Rouillier, G. Rance, D. Chablat, Inertial Line-Of-Sight Stabilization Using a 3-DOF Spherical Parallel Manipulator with Coaxial Input Shafts, in: *11th International Symposium on Optronics in Defense & Security, Bordeaux, France, 2024*, p. 11.
- [9] C. Gosselin, J.-F. Hamel, The agile eye: a high-performance three-degree-of-freedom camera-orienting device, in: *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, 1994, pp. 781–786 vol.1. doi:10.1109/ROBOT.1994.351393.
- [10] I. Tursynbek, A. Niyetkaliye, A. Shintemirov, Computation of unique kinematic solutions of a spherical parallel manipulator with coaxial input shafts, in: *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, 2019, pp. 1524–1531. doi:10.1109/COASE.2019.8843090.
- [11] I. Tursynbek, A. Shintemirov, Infinite torsional motion generation of a spherical parallel manipulator with coaxial input axes, *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)* (2020) 1780–1785.
- [12] M. K. Masten, Inertially stabilized platforms for optical imaging systems, *IEEE Control Systems Magazine* 28 (2008) 47–64. doi:10.1109/MCS.2007.910201.
- [13] J. Hilkert, Inertially stabilized platform technology concepts and principles, *IEEE Control Systems Magazine* 28 (2008) 26–46. doi:10.1109/MCS.2007.910256.
- [14] J.-P. Merlet, Parallel robots: Open problems, in: J. M. Hollerbach, D. E. Koditschek (Eds.), *Robotics Research*, Springer London, London, 2000, pp. 27–32.
- [15] J. Wu, J. Wang, L. Wang, T. Li, Dynamics and control of a planar 3-DOF parallel manipulator with actuation redundancy, *Mechanism and Machine Theory* 44 (2009) 835–849. doi:https://doi.org/10.1016/j.mechmachtheory.2008.04.002.
- [16] C. Gosselin, *Kinematic Analysis, Optimization and Programming of Parallel Robotic Manipulators*, Ph.D. thesis, McGill University, Montréal, Canada, 1988.
- [17] S. Bai, Optimum design of spherical parallel manipulators for a prescribed workspace, *Mechanism and Machine Theory* 45 (2010) 200–211. doi:https://doi.org/10.1016/j.mechmachtheory.2009.06.007.
- [18] J. Wu, J. Wang, T. Li, L. Wang, L. Guan, Dynamic dexterity of a planar 2-DOF parallel manipulator in a hybrid machine tool, *Robotica* 26 (2008) 93–98. doi:10.1017/S0263574707003621.
- [19] J. Wu, Y. Gao, B. Zhang, L. Wang, Workspace and dynamic performance evaluation of the parallel manipulators in a spray-painting equipment, *Robotics and Computer-Integrated Manufacturing* 44 (2017) 199–207. doi:https://doi.org/10.1016/j.rcim.2016.09.002.
- [20] J.-P. Merlet, Jacobian, Manipulability, Condition Number, and Accuracy of Parallel Robots, *Journal of Mechanical Design* 128 (2005) 199–206. doi:10.1115/1.2121740.
- [21] K. C. Olds, Global Indices for Kinematic and Force Transmission Performance in Parallel Robots, *IEEE Transactions on Robotics* 31 (2015) 494–500. doi:10.1109/TR0.2015.2398632.

- [22] J. Kotlarski, T. D. Thanh, B. Heimann, T. Ortmaier, Optimization strategies for additional actuators of kinematically redundant parallel kinematic machines, 2010 IEEE International Conference on Robotics and Automation (2010) 656–661.
- [23] A. Müller, Internal Preload Control of Redundantly Actuated Parallel Manipulators—Its Application to Backlash Avoiding Control, IEEE Transactions on Robotics 21 (2005) 668–677. doi:10.1109/TR0.2004.842341.
- [24] D. Zlatanov, I. A. Bonev, C. M. Gosselin, Constraint Singularities as \mathcal{C} -Space Singularities, in: J. Lenarčič, F. Thomas (Eds.), Advances in Robot Kinematics: Theory and Applications, Springer Netherlands, Dordrecht, 2002, pp. 183–192. doi:10.1007/978-94-017-0657-5_20.
- [25] D. Chablat, R. Jha, F. Rouillier, G. Moroz, Non-singular Assembly Mode Changing Trajectories in the Workspace for the 3-RPS Parallel Robot, in: J. Lenarčič, O. Khatib (Eds.), Advances in Robot Kinematics, Springer International Publishing, Cham, 2014, pp. 149–159. doi:10.1007/978-3-319-06698-1_17.
- [26] J. B. Kuipers, Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality, Princeton University Press, 1999.
- [27] J. McCarthy, An Introduction to Theoretical Kinematics, MIT Press, 1990.
- [28] D. Cox, J. Little, D. O’Shea, Ideals, Varieties, and Algorithms. An Introduction to Computational Algebraic Geometry and Commutative Algebra (Forth Edition), Springer, 2015.
- [29] D. Cox, J. Little, D. O’Shea, Using Algebraic Geometry, Graduate Texts in Mathematics, Springer New York, 2005.
- [30] C. Gosselin, J. Angeles, Singularity analysis of closed-loop kinematic chains, IEEE Transactions on Robotics and Automation 6 (1990) 281–290. doi:10.1109/70.56660.
- [31] D. Six, Conception et commande de robots parallèles volants, Theses, École centrale de Nantes, 2018.
- [32] S. Briot, I. Bonev, D. Chablat, P. Wenger, V. Arakelian, Self-Motions of General 3-RPR Planar Parallel Robots, The International Journal of Robotics Research 27 (2008) pp. 855–866. doi:10.1177/0278364908092466.
- [33] J. Schadlbauer, M. Husty, S. Caro, P. Wenger, Self-Motions of 3-RPS Manipulators, Frontiers of Mechanical Engineering 8 (2013) 62–69. doi:10.1007/s11465-013-0366-3.
- [34] A. Rosyid, B. El-Khasawneh, A. Alazzam, Review article: Performance measures of parallel kinematics manipulators, Mechanical Sciences 11 (2020) 49–73. doi:10.5194/ms-11-49-2020.
- [35] C. Gosselin, J. Angeles, A Global Performance Index for the Kinematic Optimization of Robotic Manipulators, Journal of Mechanical Design 113 (1991) 220–226. doi:10.1115/1.2912772.
- [36] L. H. Rolland, Outils algébriques pour la résolution de problèmes géométriques et l’analyse de trajectoire de robots parallèles prévus pour des applications à haute cadence et grande précision, Ph.D. thesis, Université Henri Poincaré – Nancy 1, 2003. Thèse de doctorat dirigée par ROUILLIER, Fabrice Informatique Nancy 1 2003.
- [37] S. M. Rump, Algorithms for Verified Inclusions: Theory and Practice, in: R. E. Moore (Ed.), Reliability in Computing, Academic Press, 1988, pp. 109–126. doi:https://doi.org/10.1016/B978-0-12-505630-4.50012-2.
- [38] C. W. Wampler, A. P. Morgan, A. J. Sommese, Numerical Continuation Methods for Solving Polynomial Systems Arising in Kinematics, Journal of Mechanical Design 112 (1990) 59–68. doi:10.1115/1.2912579.
- [39] D. Lazard, F. Rouillier, Solving parametric polynomial systems, Journal of Symbolic Computation 42 (2007) 636–667. doi:https://doi.org/10.1016/j.jsc.2007.01.007.
- [40] D. Chablat, G. Moroz, F. Rouillier, P. Wenger, Using Maple to analyse parallel robots, in: J. Gerhard, I. Kotsireas (Eds.), Maple in Mathematics Education and Research, Maple in Mathematics Education and Research, Springer, Cham, 2020, pp. 50–64. doi:10.1007/978-3-030-41258-6_4.
- [41] G. Moroz, Sur la décomposition réelle et algébrique des systèmes dépendant de paramètres, Ph.D. thesis, Université Pierre et Marie Curie - Paris VI, 2008.
- [42] G. Moroz, Complexity of resolution of parametric systems of polynomial equations and inequations, CoRR abs/cs/0606031 (2006). arXiv:cs/0606031.

- [43] I. Newton, *The Method of Fluxions and Infinite Series: With Its Application to the Geometry of Curve-lines*, Henry Woodfall, John Nourse, 1736. Traduit du latin vers l'anglais par John COLSON.
- [44] E. Allgower, K. Georg, *Numerical Continuation Methods: An Introduction*, Springer Series in Computational Mathematics, Springer Berlin Heidelberg, 1990. doi:10.1007/978-3-642-61257-2.
- [45] A. Baskar, M. Plecnik, J. D. Hauenstein, C. W. Wampler, A numerical continuation approach using monodromy to solve the forward kinematics of cable-driven parallel robots with sagging cables, *Mechanism and Machine Theory* 195 (2024) 105609. doi:<https://doi.org/10.1016/j.mechmachtheory.2024.105609>.
- [46] L. V. Kantorovich, On Newton's method for functional equations, *Functional Analysis and Applied Mathematics* 59 (1948) 1237–1240.
- [47] S. Smale, The fundamental theorem of algebra and complexity theory, *Bulletin of the American Mathematical Society* 4 (1981) 1–36.
- [48] R. A. Tapia, The Kantorovich Theorem for Newton's Method, *The American Mathematical Monthly* 78 (1971) 389–392. doi:10.1080/00029890.1971.11992771.
- [49] B. Demidovitch, I. Maron, *Éléments de calcul numérique*, MIR - Moscou, 1973. Traduit du russe par V. POLONSKI.
- [50] J. M. Ortega, The Newton-Kantorovich Theorem, *The American Mathematical Monthly* 75 (1968) 658–660.
- [51] J. E. Dennis, R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM ed., Society for Industrial and Applied Mathematics, 1996. doi:10.1137/1.9781611971200.
- [52] O. Chételat, *Algorithme numérique pour les changements de coordonnées des mécanismes articulés*, Ph.D. thesis, École Polytechnique Fédérale de Lausanne (EPFL), 1997. doi:10.5075/epfl-thesis-1737.
- [53] M. Grimmer, K. Petras, N. Revol, Multiple precision interval packages: Comparing different approaches, in: R. Alt, A. Frommer, R. B. Kearfott, W. Luther (Eds.), *Numerical Software with Result Verification*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 64–90. doi:10.1007/978-3-540-24738-8_4.
- [54] R. E. Moore, *Interval analysis*, Prentice-Hall, 1966.
- [55] R. E. Moore, R. B. Kearfott, M. J. Cloud, *Introduction to Interval Analysis*, Society for Industrial and Applied Mathematics, 2009. doi:10.1137/1.9780898717716.
- [56] N. Revol, F. Rouillier, Motivations for an arbitrary precision interval arithmetic and the MPFI library, *Reliable Computing* 11 (2005) 275–290. doi:10.1007/s11155-005-6891-y.
- [57] N. Revol, *Introduction à l'arithmétique par intervalles*, Research Report RR-4297, INRIA, 2001.
- [58] O. Knüppel, Profil/bias—a fast interval library, *Computing* 53 (1994) 277–287.
- [59] J.-P. Merlet, ALIAS: an interval analysis based library for solving and analyzing system of equations, 1997. SAGA Project.
- [60] J.-P. Merlet, A C++ Algorithms Library of Interval Analysis for equation Systems, 2007. The COPRIN project.
- [61] J.-P. Merlet, Solving the forward kinematics of gough-type parallel manipulator with interval analysis, *The International Journal of Robotics Research* 23 (2004) 221–236. doi:10.1177/027836904039809.
- [62] J. van der Hoeven, *Ball arithmetic*, 2009. 33 pages.
- [63] L. Fousse, G. Hanrot, V. Lefèvre, P. Pélicier, P. Zimmermann, MPFR: A multiple-precision binary floating-point library with correct rounding, *ACM Trans. Math. Softw.* 33 (2007) 13–es. doi:10.1145/1236463.1236468.
- [64] O. Altuzarra, J. P. Merlet, Certified Kinematics Solution of 2-DOF Planar Parallel Continuum Mechanisms, in: T. Uhl (Ed.), *Advances in Mechanism and Machine Science*, Springer International Publishing, Cham, 2019, pp. 197–208.
- [65] J.-P. Merlet, Mixing neural networks, continuation and symbolic computation to solve parametric systems of non linear equations, *Neural Networks* 176 (2024) 106316. doi:<https://doi.org/10.1016/j.neunet.2024.106316>.
- [66] A. Mueller, Redundant Actuation of Parallel Manipulators, in: H. Wu (Ed.), *Parallel Manipulators*, IntechOpen, Rijeka, 2008, pp. 87–108. doi:10.5772/5427.

- [67] M. Husty, J. Schadlbauer, S. Caro, P. Wenger, The 3-RPS Manipulator Can Have Non-Singular Assembly-Mode Changes, *Mechanisms and Machine Science* 15 (2014) 339–348. doi:10.1007/978-94-007-7214-4-38.
- [68] R. Babu, R. Raju K, K. Ramji, Design for optimal performance of 3-RPS parallel manipulator using evolutionary algorithms, *Transactions of the Canadian Society for Mechanical Engineering* 37 (2013) 135–160. doi:10.1139/tcsme-2013-0009.
- [69] A. Shintemirov, A. Niyetkaliyev, M. Rubagotti, Numerical optimal control of a spherical parallel manipulator based on unique kinematic solutions, *IEEE/ASME Transactions on Mechatronics* 21 (2015) 1–1. doi:10.1109/TMECH.2015.2474707.
- [70] A. Lê, G. Rance, F. Rouillier, A. Quadrat, D. Chablat, On a Software Joint Velocity Limitation of a Spherical Parallel Manipulator with Coaxial Input Shafts, in: J. Lenarčič, M. Husty (Eds.), *Advances in Robot Kinematics 2024*, Springer Nature Switzerland, Cham, 2024, pp. 43–52.
- [71] G. Wu, S. Caro, S. Bai, J. Kepler, Dynamic modeling and design optimization of a 3-DOF spherical parallel manipulator, *Robotics Auton. Syst.* 62 (2014) 1377–1386.
- [72] G. Wu, S. Bai, Design and Kinematic Analysis of a 3-RRR Spherical Parallel Manipulator Reconfigured with Four-bar Linkages, *Robotics and Computer-Integrated Manufacturing* 56 (2019) 55–65. doi:10.1016/j.rcim.2018.08.006.
- [73] A. Pashkevich, P. Wenger, D. Chablat, Design strategies for the geometric synthesis of orthoglide-type mechanisms, *Mechanism and Machine Theory* 40 (2005) 907–930. doi:https://doi.org/10.1016/j.mechmachtheory.2004.12.006.
- [74] S. Bai, M. R. Hansen, J. Angeles, A robust forward-displacement analysis of spherical parallel robots, *Mechanism and Machine Theory* 44 (2009) 2204–2216.