Cusp Points in the Parameter Space of Degenerate 3-RPR Planar Parallel Manipulators

Montserrat Manubens  
Institut de Robòtica i Informàtica Industrial,  
CSIC - UPC,  
Llorens i Artigas, 4-6,  
08028 Barcelona, Spain  
mmanuben@iri.upc.edu

Guillaume Moroz  
INRIA Nancy-Grand Est,  
615, rue du jardin botanique,  
54600, Villers-lès-Nancy, France  
guillaume.moroz@inria.fr

Damien Chablat  
Institut de Recherche en Communications  
et Cybernétique de Nantes,  
UMR CNRS n 6597,  
1 rue de la Noé,  
44321 Nantes, France  
damien.chablat@irccyn.ec-nantes.fr

Philippe Wenger  
Institut de Recherche en Communications  
et Cybernétique de Nantes,  
UMR CNRS n 6597,  
1 rue de la Noé,  
44321 Nantes, France  
philippe.wenger@irccyn.ec-nantes.fr

Fabrice Rouillier  
INRIA Paris-Rocquencourt,  
Université Pierre et Marie Curie Paris VI  
4, place Jussieu,  
F-75005 Paris, France  
fabrice.rouillier@inria.fr

This paper investigates the conditions in the design parameter space for the existence and distribution of the cusp locus for planar parallel manipulators. Cusp points make possible non-singular assembly-mode changing motion, which increases the maximum singularity-free workspace. An accurate algorithm for the determination is proposed amending some imprecisions done by previous existing algorithms. This is combined with methods of Cylindric Algebraic Decomposition, Gröbner bases and Discriminant Varieties in order to partition the parameter space into cells with constant number of cusp points. These algorithms will allow us to classify a family of degenerate 3-RPR manipulators.

Keywords: kinematics, parallel manipulator, singularities, cusp, discriminant variety, cylindric algebraic decomposition, degenerate 3-RPR, symbolic computation.

1 Introduction

In the past, singularities were believed to physically separate the different assembly modes, meaning that for fixed joint values one could not find a path going from one assembly mode to another without crossing a singular configuration. So the interest relied on considering the widest connected non-singular domain, called aspect. Innocenti and Parenti-Castelli pointed out in [1] that non-singular changes of assembly mode are possible, and McAree and Daniel showed in [2] that such changes are possible when triple roots of the Forward Kinematic Problem (FKP) exist. In [3] Zein, Wenger and Chablat showed that for the case of 3-RPR manipulators a non-singular change of assembly mode can be accomplished by encircling a cusp point, and Husty recently proved in [4] that the generic 3-RPR parallel manipulators without joint limits always have 2 aspects.

From the algebraic point of view, the locus of cusp points can be described by means of symbolic equations. In order to avoid long symbolic-algebraic manipulations, these equations are usually solved by numerical approximation at an early stage, which may lead to small deviations that can be propagated along the process. However, there exist efficient symbolic-algebraic techniques that may leave the use of numerical methods to the last step. In particular, we will apply Gröbner bases [5] in order to adopt a more suitable equivalent system defining the same solution points.

Lazard and Rouillier introduced the mathematical notion of Discriminant Variety (DV) [6], which is a variety of codimension 1 in the chosen parameter space whose com-
plement satisfies the property that over each connected component the given system has a constant number of solutions. The complement of this DV will be partitioned into cells by a Cylindrical Algebraic Decomposition [7], also known as CAD.

This paper is intended to illustrate both the performance of the new algorithm for the determination of the locus of cusp points and its combination with the aforementioned algebraic techniques in the analysis of existence conditions and distribution along a 2-dimensional parameter space. Although the method can be applied to more general manipulators (see [8]), such performance will be exemplified on a family of degenerate 3-RPR manipulators, detailed in Section 2. The algorithm for the cusp point determination, which is one of the main contributions of the paper, is given in Section 3, where it is compared to other previous algorithms. Section 4 outlines some of the exploited algebraic objects such as the DV. In Section 5 the previous procedures are combined with the CAD to partition a 2-dimensional space with regard to the associated number of cusp points, which leads us to analyze a complete family of degenerate 3-RPR manipulators that depend on one geometric parameter. This section also illustrates some applications of the presented strategy to robot design. The paper concludes in Section 6.

2 A class of degenerate 3-RPR

Let us describe the family of manipulators on which the strategies presented along the paper will be exemplified. A general 3-RPR manipulator is a 3-degrees-of-freedom planar parallel mechanism that has two platforms connected by three RPR rods, with the prismatic joints being actuated and the revolute ones being passive. Without loss of generality we can assume the absolute reference frame to be such that the base points of the leg rods are \( A_1 = (0,0) \), \( A_2 = (A_{2x},0) \) with \( A_{2x} > 0 \), and \( A_3 = (A_{3x},A_{3y}) \). If \( B_1 \), \( B_2 \) and \( B_3 \) are the corresponding points on the moving platform, then the geometric parameters associated to this manipulator are the values \( A_{2x}, A_{3x}, A_{3y} \), the lengths \( d_1 = \|B_1B_2\| \), \( d_3 = \|B_1B_3\| \), and the angle \( \beta = B_2B_1B_3 \). The input-space is then formed by \( \rho = (\rho_1,\rho_2,\rho_3) \in \mathbb{R}^3 \), where \( \rho_i \geq 0 \) are the leg rod lengths, and the output-space is formed by the poses of the moving platform \( x = (x,y) \), where \( B_1 = (x,y) \) and \( \alpha \) is the angle of vector \( B_2 - B_1 \) relative to \( A_2 - A_1 \). We define \( (s_\alpha,c_\alpha) \), \( (s_\beta,c_\beta) \) and \( (s_{\alpha+\beta},c_{\alpha+\beta}) \) to denote the sines and cosines of \( \alpha, \beta \), and \( (\alpha+\beta) \), respectively. Then the forward kinematics of a general 3-RPR manipulator is defined by the system of equations

\[
\begin{align*}
\rho_1^2 - x^2 - y^2 &= 0, \\
(x + d_1c_\alpha - A_{2x})^2 + (y + d_1s_\alpha)^2 - \rho_1^2 &= 0, \\
(x + d_3c_{\alpha+\beta} - A_{3x})^2 + (y + d_3s_{\alpha+\beta} - A_{3y})^2 - \rho_3^2 &= 0.
\end{align*}
\]

For these manipulators, Hunt showed that the FKJ admits at most 6 assembly modes [9], and several authors [10, 11] proved independently that the system associated to the FKJ can be reduced to a polynomial of degree 6. The 3-RPR manipulators for which the degree of this characteris-tic polynomial decreases are known as analytic or degenerate [12, 13], because the Cramer system in Gosselin’s method degenerates. In this paper we will focus on a class of degenerate 3-RPR manipulators whose base and moving platforms are congruent triangles, with the moving triangle being reflected with respect to the base one, as that of Fig. 1. This class of manipulators was first studied by Wenger, Chablat and Zein in [14]. Their mathematical description requires the addition to the initial Eqn. (1) of the following geometric constraints:

\[
\begin{align*}
d_1 &= A_{2x}, \\
\cos(\beta) &= \frac{A_{3x}}{d_3}, \\
\sin(\beta) &= -\frac{A_{3y}}{d_3}.
\end{align*}
\]

Therefore, the system of equations defining this family of degenerate 3-RPR manipulators, formed by Eqns. (1) and (2), will be denoted as

\[ F(\rho, x) = 0. \]

Generically, we will refer to this system by \( F \). Whenever concrete values for the geometric parameters are considered, these will be specified. Finally, the notation \( |(\rho, x)| \) will stand for the evaluation on real values \( (\rho, x) \).

3 Cusp locus determination

In this section we describe the cuspidal locus and analyze the usual methods for their determination. After that we propose a more accurate approach and compare it to the previous ones in a simple example.

Let us assume that a specific manipulator, whose geometric parameters have been set into \( F \), has been designated. Then, we denote the associated configuration space by

\[ C(F) = \{ (\rho, x) \in \mathbb{R}^6 : F|_{(\rho, x)} = 0 \}. \]

The Jacobian matrix of \( F \) with respect to the output variables is denoted as \( J_\alpha(F) = \begin{pmatrix} \frac{dF}{d\alpha} & \frac{dF}{d\beta} & \frac{dF}{d\delta} \end{pmatrix} \). The configura-
tions where its determinant is zero are called parallel singular configurations, or type 2 singularities. On these configurations the manipulator shows a loss of control. The parallel singular locus of our manipulator is a 2-dimensional space that can be described (see [15]) as

\[ \Sigma(F) = \{(p, x) \in C(F) : J_x(F)|_{(p, x)} \text{ is rank deficient}\}. \]

For simplicity, we will refer to this set as the singular locus. With this setting we now define the cuspidal locus as

\[ \kappa(F) = \{(p, x) \in C(F) : \rho \text{ root of exact multiplicity 3 of } F\}, \]

e.g. the triple roots of the FKP. Observe that \( \kappa(F) \subseteq \Sigma(F) \), since the Jacobian \( J_x(F) \) is rank deficient on the roots of multiplicity three of \( F \). It is known that in the proximity of cusp points a non-singular change of assembly mode can be made.

Figure 2 shows a cusp point \( \kappa \) and a non-singular path connecting two different assembly modes (\( p_1 \) and \( p_3 \)). We shall note, however, that both the singular and the cusp locus are quite difficult to visualize in the 6-dimensional \((p, x)\)-space. So for mechanisms with at most one inverse kinematics solution, as is the case for the 3-RPR, we will actually project them onto the input-space instead.

### 3.1 Usual methods for the cusp computation

Let us revise the two main algorithms that have been more commonly used in the determination of the locus of cusp points for a given manipulator. The following method, introduced by Wenger and Chablat in [13], and analytically derived in [16], has been used for the degenerate 3-RPR manipulators. It was inspired on an approach developed by Hernández et al in [17] for other robots.

#### Algorithm 1 by Wenger and Chablat [13]

1. Reduce \( F \) (by successive resultants) to a single equation \( g(t) = 0 \), with \( t = \tan(\alpha/2) \) and coefficients in \( \rho \).
2. Equations of triple roots of \( g \)
   \[ G = \{g = 0, \frac{\partial^2 g}{\partial \alpha^2} = 0, \frac{\partial^3 g}{\partial \rho^3} = 0\} \]
3. Equations of strictly triple roots of \( g \)
   \[ G = G \cup \{\frac{\partial^3 g}{\partial \alpha^3} \neq 0\} \]
4. \( G \) Eliminate \( t \) from \( G \) and solve the remaining system for real values of \( \rho \).
5. Solve \( G \).

This strategy reduces the problem to the computation of the strictly triple roots of one single univariate polynomial \( g \). However, the constraint added in step 3 makes the computation quite hard, and thus this step is often removed.

Another commonly used method, described by McAree and Daniel in [2], makes use of the series expansion of \( F \).

#### Algorithm 2 by McAree and Daniel [2]

1. Series expansion of \( F \)
   \[ \Delta F = \frac{\partial^2 F}{\partial \alpha^2} \Delta x + \frac{\partial^2 F}{\partial \rho \partial \alpha} \Delta \rho + \frac{\partial^2 F}{\partial x^2} \Delta x^2 + \Delta x^T \left( \frac{\partial^2 F}{\partial \rho \partial \alpha} \right) \Delta \rho + \frac{\partial^2 F}{\partial \alpha \partial x} \Delta \alpha^T \left( \frac{\partial^2 F}{\partial \rho \partial \alpha} \right) \Delta \rho + \ldots \]
2. Compute configurations where 1st and 2nd order constraints are rank deficient, i.e. solve \( v^T \left( \frac{\partial^2 F}{\partial \alpha^2} \right) v = 0 \), where \( v \) is a unit vector in right kernel of \( \frac{\partial^2 F}{\partial \alpha^2} \), and \( u \) is a unit vector that spans left kernel.

This second strategy reduces the problem to the resolution of some quadratic equations, but it also requires to find the unit vectors \( u \) and \( v \), which may hinder the computation.

These algorithms are commonly used in the cusp locus determination. However, both have drawbacks related to the non-cuspidality of some resulting points:

- Since the polynomial \( g \) obtained by Algorithm 1 is the result of several projections, some of the obtained points may correspond to the projection of complex (not real) solutions, as we will see later on.
- Step 3 usually needs to be removed from Algorithm 1 in order to avoid slow-processing.
- Algorithm 2 does not constrain the multiplicity of the solutions to be exactly 3, so it may obtain higher multiplicity ones. Regardless of that, in [3] it is shown that additional spurious solutions may be produced for generic 3-RPR manipulators.

Therefore, both methods can only provide sufficient conditions for the cuspidal locus but not always necessary ones.

### 3.2 Improved method

Despite the fact that the formulation of the cusp locus is quite simple, the associated system of equations usually contains many equations in many unknowns, whose resolution can take long computations and even lead to abnormal termination for not too complex examples. So the methods described previously were introduced as simple, though not accurate, alternatives to the symbolic resolution. However, we can now get over some of these difficulties with current powerful symbolic algebra tools that fix the deficiencies of the algorithms detailed above.
The approach that we propose is an evolution of [18] by Moroz et al., inspired on the results of [19]. The main difference of the proposed method compared to that of [18] is the introduction of the saturation operator to remove the quadruple roots.

Algorithm 3 Proposed method

1. Equations of double roots of $F$ w.r.t. $\mathbf{p}$
   \[ D_F = F \cup \{ \det(J_4(F)) = 0 \} \]
2. Equations of triple roots of $F$ w.r.t. $\mathbf{p}$
   \[ T_F = D_F \cup \{ \det(m) : m \text{ maximal minors of } J_3(D_F) \} \]
3. Equations of quadruple roots of $F$ w.r.t. $\mathbf{p}$
   \[ Q_F = T_F \cup \{ \det(m) : m \text{ maximal minors of } J_4(T_F) \} \]
4. Saturate $T_F$ by $Q_F$
   \[ C_F = \text{sat}(T_F, Q_F) \]
5. Solve $C_F$ for real values of $(\mathbf{p}, x)$

Given the system defining the mechanism $F$, it computes iteratively the equations $T_F$ and $Q_F$ of triple and quadruple roots $\mathbf{p}$ of $F$, respectively. Then, we use saturation. Given two polynomial systems $S_1$ and $S_2$, sat($S_1, S_2$) is an algebraic operator that returns a polynomial system whose solution set is the closure of the solutions of the first system after removing those of the second one. If $V(S_i)$ denotes the solution set of $S_i$, it is satisfied that

\[ V(\text{sat}(S_1, S_2)) = V(S_1) \cup V(S_2). \]  

In general, the saturation ensures that all roots of $S_2$ are removed. However, in specific cases, some points can remain due to property of Eqn. (3) for which we can only obtain $V(S_1) \setminus V(S_2)$ instead of $V(S_1) \cup V(S_2)$, which can differ by a null-measure set that can easily be removed afterwards. Further details on the saturation and its geometric interpretation can be found in [5].

Although we are only interested in real (feasible) solutions, we shall note that the polynomial system obtained after saturating has real coefficients and thus its solution set could contain some complex (not real) roots. For this reason we need to solve the final cusp system $C_F$ in the real field. This is done by using the RootFinding Maple package.

With Algorithm 3 the previous drawbacks are amended:
- When computing the saturation of $T_F$ by $Q_F$, the points of multiplicity 4 or higher are removed, and so we can guarantee that only the cusp locus is obtained.
- By solving $C_F$ for $(\mathbf{p}, x)$, instead projecting onto the $\mathbf{p}$-space, we avoid having biased points produced by the projection of complex (not real) solutions.
- Furthermore, solving $C_F$ in the real field ensures that no other spurious complex solutions are considered.

3.3 Case study comparison

Let us now compare the performance of both Algorithm 1 without step 3 and the proposed Algorithm 3 on a simple case of degenerate 3-RPRR in order to contrast their results. However, let us clarify that both the formulation and the proposed algorithm apply to other more general manipulators (see [8]). We set the geometric parameter values $A_{2x} = 1, A_{3x} = 0, A_{3y} = 1, \beta = -\pi/2, d_1 = 1$, and $d_2 = 1$.

The characteristic polynomial for Algorithm 1 is

\[ g(t) = (p_2^4 - p_1^4)t^4 + (p_2^4 - p_1^4 - 4)t^3 + (p_2^4 - p_1^4 - 4)t^2 + p_2^2 - p_1^2 \]

After eliminating $t$ from $G$ we get $\tilde{G} = \{ P_1, P_2, P_3 \}$ as follows

\[ P_1 = p_2^4 + p_2^4 - 2p_2^2p_3^2 + 6p_1^2 - 3p_2^3 - 3p_3^2 - 12 = 0 \]
\[ 2p_1^4 + 2p_1^4 - 4p_1^2p_3^2 + 4p_1^2 + 3p_2^2 - 7p_3^2 = 16 = 0 \]
\[ (4) \quad p_1^4 + p_2^4 - 7p_1^2p_3^2 - 3p_2^3 - 3p_3^2 - 4p_1^2 - 6 = 0. \]

Observe that these equations are not independent. Indeed, $P_2$ is a combination of the other two:

\[ P_2 = \frac{(p_1^3 - p_2^3 + 1)}{3} P_1 + \frac{(p_1^3 - p_2^3 + 6)}{3} P_3. \]

Additionally, there are solutions of $\tilde{G}$ that do not correspond to the real cusp locus. For instance, if we set $p_1 = 1/3$, the system $\tilde{G}_{|p_2=1/3}$ has two solutions with both $p_2$ and $p_1$ positive. But the FKP evaluated on these two solutions only.

We now test Algorithm 3 on the same example. The equations of the cusp locus $C_F$ are:

\[ 6c_a + p_2^2 - p_1^2 = 0 \]
\[ 2x_a + s_a - 1 = 0 \]
\[ 2s_a^2 - s_a - 1 = 0 \]
\[ 2c_a s_a - c_a = 0 \]
\[ 3c_a + 3s_a + p_1^2 - p_2^2 + 1 = 0 \]
\[ 3c_a + 3s_a + x_a^2 + y_a^2 - p_3^2 + 1 = 0 \]
\[ 2x_a s_a + 2y_a - 4s_a - x - y + 2 = 0 \]
\[ 2s_a p_2^2 - 3c_a - 6s_a - 4x y - x - y = 0 \]
\[ 2x_a c_a + 2y a + c_a - 3s_a - 2 x + 1 = 0 \]
\[ 2y a + 2y a + c_a = s_a - x + y + 1 = 0 \]
\[ 4y s_a - s_a + 2(c_a - 1)p_2^2 + 4y^2 - 3y + x + 1 = 0 \]
\[ 2s_a (2y^2 - 4y - 1) - 3c_a - 4xy - 2y^2 + 3y \]
\[ -x + p_2^2 = 0 \]
\[ 6s_a (2y + 1) + 12c_a - 8y^2 x + 8y^2 + 12xy \]
\[ + x - 3y + 2x p_3^2 - 6y p_3^2 - 6p_3^2 = 8 = 0 \]
\[ 18s_a (18y - 1) + 36c_a + 32y^3 - 2y^2 + 44xy \]
\[ + 13x - 47y + 4p_2^2 - 8p_2^2 y x - 32y^2 p_3^2 \]
\[ - 6x p_3^2 - 6y p_3^2 - 32p_3^2 + 40 = 0. \]
We first compute a Gr"obner basis of the ideal \( \langle p_1, \ldots, p_m \rangle \) for any ordering, which will help us detect if the system has or has not finitely many complex solutions. If yes, then compute a so called Rational Univariate Representation (RUR) of \( \langle p_1, \ldots, p_m \rangle \) (see [21]), which is an equivalent system of the form
\[
\{ f(T) = 0, X_1 = \frac{u_1(T)}{p_1(T)}, \ldots, X_n = \frac{u_n(T)}{p_1(T)} \},
\]
where \( T \) is a new variable independent of \( X_1, \ldots, X_n \), equipped with a so called separating element (injective on the solutions of the system) \( u \in \mathbb{Q}[X_1, \ldots, X_n] \) and such that :
\[
V(p_1, \ldots, p_m) \xrightarrow{u} V(f) \xrightarrow{u^{-1}} V(p_1, \ldots, p_m) \]
defines a bijection between the (real) roots of the system and the (real) roots of the univariate polynomial \( f \).

We then solve \( f = 0 \), computing so called isolating intervals for its real roots, i.e. non-overlapping intervals with rational bounds that contain a unique real root of \( f \) (see [22]). Finally, interval arithmetic is used in order to get isolating boxes of the real roots of the system (non-overlapping products of intervals with rational bounds containing a unique real root of the system), by studying the RUR over the isolating intervals of \( f \).

In practice, we use the function RootFinding\{Isolate\} from Maple software, which performs exactly the computations described above.

4.2 Discriminant varieties
Consider now the constructible set
\[
S = \{ v \in \mathbb{C}^n : p_1(v) = 0, \ldots, p_m(v) = 0, q_1(v) > 0, \ldots, q_l(v) > 0 \},
\]
and let us assume that for almost all the parameter values this \( S \) is a finite set of points. Then, a discriminant variety of \( S \) with respect to \( \{ U_1, \ldots, U_d \} \) is a variety \( V' \subset \mathbb{C}^d \) such that over each connected open set \( U \) not intersecting \( V' \setminus V' = \emptyset \), \( S \) defines an analytic covering. In particular, the number of points of \( S \) over any point of \( \mathcal{U} \) is constant.

Discriminant varieties can be computed using basic and well-known tools from computer algebra such as Gröbner bases [6]. A full package is available in Maple software through the RootFinding\{Parametric\} package, which provides us with a polynomial \( DV(S;U_1, \ldots, U_d) \) whose associated discriminant variety is \( V' \).

4.3 Case study comparison
Let us consider again the degenerate 3-RPR with the same geometric parameter values as those specified in section 3.3, i.e. \( A_{2x} = 1, A_{3x} = 0, A_{3y} = 1, \beta = -\pi/2, d_1 = 1, \) and \( d_3 = 1 \), and consider the systems \( G \) (Eqn. 4), and \( C_f \) (Eqn. 5), obtained by Algorithm 1 without step 3 and by Algorithm 3, respectively. We will regard as a parameter one of the leg lengths \( p_1 \) of the manipulator. The discriminant
variety will provide us with a polynomial in \( p_1 \) whose roots will delimit some open intervals such that for whatever value of \( p_1 \) within one interval, the number of cusp points of \( F|_{p_1} \) will be the same. We compute the \( \text{DV} \) for each system with respect to \( p_1 \), and analyze the results.

In the first case, we get the polynomial \( \text{DV}(\tilde{G}; p_1) = p_1(\tilde{p}_1^2 - 2)(2p_1^4 + 10p_1^2 - 1) \), whose roots describing the discriminant variety are

\[
\begin{align*}
\tau_0 &= 0, \\
\tau_1 &= \frac{\sqrt{27} - 10}{2}, \\
\tau_2 &= \frac{\sqrt{2}}{2}, \\
\tau_3 &= \sqrt{2},
\end{align*}
\]

Since the number of cusp points is kept constant between two consecutive roots, we can compute the associated number of cusps by picking one single value of \( p_1 \) inside each open interval and solve \( G|_{p_1} \). In this case we obtain

- 0 cusp configurations for \( p_1 \in ]0, \tau_1[ \),
- 2 cusps configurations for \( p_1 \in [\tau_1, \tau_2[ \),
- 3 for \( p_1 \in ]\tau_2, \infty[ \).

Substituting \( p_1 = \tau_1 \) into \( \tilde{G} \) we obtain the number of cusps on the borders of the intervals.

- 0 cusps on \( p_1 = 0 \),
- 2 cusps on \( p_1 = \tau_1 \),
- 3 on \( p_1 = \tau_2 \).

In the second case, a similar analysis for \( C_F \) gives

\[
\text{DV}(C_F; p_1) = p_1(\tilde{p}_1^2 - 2)(8\tilde{p}_1^2 - 1)(2p_1^4 + 10p_1^2 - 1),
\]

which has one more root than \( \text{DV}(\tilde{G}; p_1) \)

\[
\begin{align*}
\tau_0 &= 0, \\
\tau_1 &= \frac{\sqrt{27} - 10}{2}, \\
\tau_2 &= \frac{\sqrt{2}}{4}, \\
\tau_3 &= \sqrt{2}.
\end{align*}
\]

The intervals and the numbers of cusps for \( C_F \) differ a bit from those obtained for \( G \):

- 0 cusps for \( p_1 \in ]0, \tau_1[ \),
- 0 cusps for \( p_1 \in [\tau_1, \tau_2[ \),
- 4 cusps for \( p_1 \in [\tau_2, \tau_3[ \),
- 6 cusps for \( p_1 \in [\tau_3, \infty[ \).

The results obtained in both cases are compared in Fig. 4. We can observe that the first two intervals do not exactly coincide, and that for the second system the obtained numbers of cusps appear doubled for all intervals (compared to those obtained for the first system). Both phenomena can be explained as a consequence of the projection map used to compute the system \( G \). Let us remind the reader that \( G \) is obtained after several reductions of the initial system, each of which applying also a projection on the \( p \)-space. For this, there can be complex configurations of the manipulator that project onto real roots \( p \) of \( G \). This is the case for the values of \( p_1 \in ]r_1, r_2[ \).

The same can be done for any other parameter and the same phenomena can be observed.

5 Higher-dimensional discussion by means of a CAD

By construction we know that over any connected open region not intersecting the \( \text{DV} \) the system has a constant number of real roots, for whatever chosen parameters. But if we want to discuss larger parameter spaces, then the open regions will no longer be as simple as 1-dimensional intervals. So the goal of this section is to provide an accurate description of the regions with constant number of solutions. For this we will use the Cylindrical Algebraic Decomposition (CAD) [7, 23].

5.1 The complementary of a discriminant variety

Let \( P_d \subset \mathbb{Q}[U_1, \ldots, U_d] \) be the set of polynomials describing the \( \text{DV} \). Then for each \( i = d - 1, \ldots, 0 \), we introduce a new set of polynomials \( P_i \subset \mathbb{Q}[U_1, \ldots, U_{d-i}] \) defined by a backward recursion:

- \( P_0 = \{ \text{DV}(p, U_i) \} \),
- \( P_i = \{ \text{LeadingCoefficient}(p, U_i), \text{Resultant}(p, U_j) \mid p, q \in P_{i+1} \} \)

Each \( P_i \) has an associated algebraic variety of dimension at most \( i - 1 \), \( V_i = \{ V_i[p, U_j] \} \). The \( V_i \) are used to recursively define a finite union of simply connected open subsets \( \cup_{k=1}^{P_i} U_{k} \subset \mathbb{R}^d \) of dimension \( i \) such that \( V_i \cap U_k = \emptyset \).

Before defining the sets \( U_{k,i} \), we introduce some notation: for a univariate polynomial \( p \) with \( n \) real roots, 

\[
\text{root}(p, l) = \begin{cases} 
-\infty & \text{if } l < 0, \\
\text{the } l^{th} \text{ root of } p & \text{if } 0 \leq l \leq n, \\
+\infty & \text{if } l > n.
\end{cases}
\]

Moreover, if \( p \) is a \( n \)-variate polynomial, and \( \mathbf{v} \) is \( (n-1) \)-tuple, then \( p^\mathbf{v} \) denotes the univariate polynomial where the first \( n-1 \) variables have been replaced by \( \mathbf{v} \).

The recursive process defining the \( U_{k,i} \) is the following:

- For \( i = 1 \), let \( p_1 = \prod_{p \in P_0} p \).
- Taking all \( U_{k,i} = [\text{root}(p_1, k); \text{root}(p_1, k + 1)] \) for \( k = 0, \ldots, n \), where \( n \) is the number of real roots of \( p_1 \), one gets a partition of \( \mathbb{R} \) that fits the above definition. Moreover, one can arbitrarily chose one rational point \( u_{1,k} \) in each open interval \( U_{k,i} \).
- Then, for \( i = 2, \ldots, d \), let \( p_i = \prod_{p \in P_0} p \).
- The regions \( U_{k,i} \) and the points \( u_{i,k} \) are of the form:

\[
U_{k,i} = \{ (v_1, \ldots, v_{i-1}) \mid \mathbf{v} = (v_1, \ldots, v_{i-1}) \in U_{i-1,j}, \\
v_j \in [\text{root}(p_j^\mathbf{v}, l); \text{root}(p_j^\mathbf{v}, l + 1)] \}
\]

\[
u_{i,k} = (\tilde{B}_1, \ldots, \tilde{B}_{i-1}, \tilde{B}_i), \text{ with }
\]

\[
\{ (\tilde{B}_1, \ldots, \tilde{B}_{i-1}) \} = u_{i-1,j}, \\
\{ \tilde{B}_i \in [\text{root}(p_i^\mathbf{v}, l); \text{root}(p_i^\mathbf{v}, l + 1)] \}
\]
thermore, this distribution is consistent with that obtained for the system to be solved: the cells $U_{d,k}$ are naturally grouped with regard to their number of cusps. Additionally, cells with different number of cusps are exclusively separated by curves of the discriminant variety. Furthermore, this distribution is consistent with that obtained for $d_1 = 1$ in Section 4.3, as can be seen in Fig. 7. However, here the section is divided into many more smaller intervals whose borders we cannot apriori ensure to be associated to a specific number of cusp points. In fact, let us also observe that the cells with 2 cusp points (in blue) degenerate into one single point for $d_1 = 1$. So we could claim that the case with $d_1 = 1$ is a very special degenerate 3-RPR manipulator.

5.2 Open CAD for a class of degenerate 3-RPR

Let us see the performance of this CAD on a 2-dimensional discussion. We consider now a family of degenerate 3-RPR manipulators with $A_{3x} = 1$, $A_{3y} = 0$, $A_{3z} = 1$, $\beta = -\pi/2$, and $d_1 = 1$, and regard as parameters both $\rho_1$ and $d_1$, constrained by $d_1 \geq 0$. Now, the system $C_F$ describing the cusp locus associated to this family of manipulators has 18 polynomials, its DV is plot in Fig. 5, and the polynomial $DV(C_F;\rho_1,d_1)$ factors as follows:

$$d_1 \rho_1 (d_1^2 + 1) (-4 \rho_1^6 - 12 \rho_1^4 + 27 \rho_1^2 d_1^2 + 15 \rho_1^2 - 4)$$

$$d_1 \rho_1 (d_1^2 + 1) (4 \rho_1^6 + 12 \rho_1^4 d_1^2 - 15 \rho_1^2 d_1^4 + 4 d_1^4 - 27 \rho_1^2 d_1^2)$$

$$d_1 \rho_1 (d_1^2 + 1) (256 \rho_1^6 d_1^2 + 81 \rho_1^6 d_1^4 - 288 \rho_1^4 d_1^4 + 256 \rho_1^6 - 576 \rho_1^4 d_1^2 + 256 \rho_1^2 d_1^4 + 51 \rho_1^4 d_1^4 - 16 d_1^4 - 288 \rho_1^4 + 51 \rho_1^2 d_1^4 + 81 \rho_1^2 d_1^2)).$$

The complement of this DV produces 90 cells with associated numbers of cusps varying among 0, 2, 4 and 6, as shown in Fig. 6, where black vertical lines delimit intersection points of the DV. Let us notice that although all cells must be considered disconnected, it is apparent that cells are naturally grouped with regard to their number of cusps. Additionally, cells with different number of cusps are exclusively separated by curves of the discriminant variety. Furthermore, this distribution is consistent with that obtained for $d_1 = 1$ in Section 4.3, as can be seen in Fig. 7. However, here the section is divided into many more smaller intervals whose borders we cannot apriori ensure to be associated to a specific number of cusp points. In fact, let us also observe that the cells with 2 cusp points (in blue) degenerate into one

5.3 Study of the cusp points on the borders of the CAD

At this point we can only certify the number of cusp points in the open cells. This excludes the cell borders. The union of all these borders consists of the DV plus the delimitation of intersection points of the DV. However, by definition, changes in the number of solutions can only happen on the DV. So, we just need to analyze the DV.

We could try executing a further iteration of the CAD on the DV, but the system to be solved turns out to be too complex and we cannot obtain any results after long computations. It is clear that not all points on the DV will correspond to the same number of solutions. But we can expect the number of solutions to be preserved along the DV between two consecutive auto-intersection points of the DV. And although it has not yet been proven, many tests have been run on several examples with random points on the DV and all the results confirm what the following conjecture infers.

Conjecture 1. Given a polynomial system $F$, let $V'$ be the DV of $F$ w.r.t two parameters $U_1, U_2$, and let $\mathcal{A}$ be the set of its auto-intersection points. Then, the number of solutions of $F$ is constant on each connected component of $V' \setminus \mathcal{A}$.

The following algorithm analyzes the to study the numbers of solutions on the DV based on the previous conjecture.
Instead, it should be read as follows: 

- If we pick a \((p_1,d_1)\)-point with associated number of cusps \(k\), then whatever the values \(p_2\) and \(p_3\) may take, \(C_F[p_1,p_2,d_1,p_3]\) has \(k\) solutions.

5.4 Complete analysis and applications

From the results exposed in the previous subsections and by joining all the different pieces together we obtain a complete partition of the 2-dimensional parameter space.

The execution of Algorithm 4 on \(DV(C_F;\rho_1,d_1)\) provides the distribution of cusp points shown in Fig. 8. The integral picture of the 2-dimensional distribution is given in Fig. 9. It is interesting to notice that there is a continuity on the transitions between cells having the same number of cusp points, since their common border inherits that same number of cusp points.

Observe also that this distribution has been obtained thanks to the DV associated to the chosen parameters \(d_1\) and \(\rho_1\), which depends exclusively on these two parameters. In particular, \(DV(C_F;\rho_1,d_1)\) does not depend on \(p_2\) nor \(p_3\). This tends to be erroneously interpreted as:

- If we pick a \((p_1,d_1)\)-point with associated number of cusps \(k\), then whatever the values \(p_2\) and \(p_3\) may take, \(C_F[p_1,p_2,d_1,p_3]\) has \(k\) solutions.

Instead, it should be read as follows:

- If we pick a \((p_1,d_1)\)-point with associated number of cusps \(k\) and fix these values then, among the reachable configurations there are \(k\) cuspidal ones, i.e. \(C_F[p_1,d_1]\) has \(k\) solutions.

However, the number of associated cusp points does establish a maximum of cuspidal configurations for any values \(p_2\) and \(p_3\). For example, in yellow regions we can have a maximum of 4 cuspidal configurations, but depending on the values of \(p_2\) or \(p_3\) there can even be none. In particular, for the red regions there are 0 cusp points for all possible values \(p_2\) and \(p_3\).

Some applications can be derived which may be interesting from the designer’s point of view:

- It can be helpful in deciding the most suitable architecture of the mechanism. Let us assume that we want to design a 3-RPR manipulator with some given geometric constraints such that for a specific task one of the legs has to be blocked to a fixed length \(p_1\), but the job requires a large singularity-free workspace. Therefore, we may be interested in finding a range \(\Delta d_1\) of parameter values for which the manipulator is cuspidal.

- It can also be useful for deciding the most suitable ranges of leg lengths for each possible architecture, given a specific task. For instance, let us assume that the job is set for a non-cuspidal manipulator with parameter values \(A_2x = 1, A_3x = 0, A_3y = 1, \beta = -\pi/2\), and \(d_1 = 1\), but it requires the largest possible range of the leg length \(p_1\). Then, the value \(d_1\) can be optimized with this criterion.

Figure 10 details both the optimal value \(d_1\) and the largest possible range \(\Delta p_1\) for our problem.

Let us just notice that in both cases the obtained ranges can be of varied topology (open, closed, semi-closed, open and closed, connected, or even a union of these types). This is due to the combination of both the CAD and the study of the cusp locus on the DV.

6 Conclusions

This paper has introduced both an efficient method for the computation of the cuspidal configurations of a mechanism, and a reliable algorithm that partitions a given parameter space into open regions with constant number of associated cusp points.

The first one is based on a symbolic-algebraic approach able to describe the roots of exact multiplicity 3 and a certified numerical algorithm that isolates among them the real
(i.e. not complex) ones. This symbolic-numeric approach is more efficient than other previously existing methods, which mainly relied on the approximation of roots of multiplicity at least 3 after reducing the initial system to a simpler one and projecting it onto the $\mathbf{p}$-space.

This new method is combined with some algebraic tools such as the discriminant variety (DV) and the cylindrical algebraic decomposition (CAD) in order to analyze a 2-dimensional parameter space with respect to the associated number of cusp points. This second algorithm provides a partition of the parameter space into cells with constant number of cusp points, which is certified for whatever values are picked inside each open cell but not on their borders. Cell borders are further analyzed by Algorithm 4 based on Conjecture 1, which still remains unproved.

Both algorithms have been applied to the analysis and distribution of the cusp locus for a family of degenerate 3-RP manipulators, and some applications to robot design are also derived. This does not mean that the two given algorithms are specially designed for this type of 3-RP mechanisms. Indeed, they are suitable for more general examples since they do not rely on any ad-hoc formulation. Nevertheless, for some examples the obtention of results within reasonable time may not be feasible yet, since there is an important symbolic-algebraic part, and thus the more complex the initial system is the harder it will be to compute a partition on the parameter space.

Acknowledgements

The authors would like to acknowledge the financial support of this research by ANR Project SiRoPa. The first and second authors have been supported for this research by two postdoctoral contracts at the Institut de Recherche en Communications et Cybernétique de Nantes under the sponsorship of the same project.

References


Fig. 10. Optimal $d_1$ and $\Delta p_1$ for non-cuspidal degenerate 3-RPR.
In IFToMM 13th World Congress in Mechanism and Machine Science.


