

# Efficient and certified algorithms for solving polynomial system of equalities and inequalities

Applications to some problems in robotics

JNRR 2005 - Lorient

**Jean-Charles Faugère**

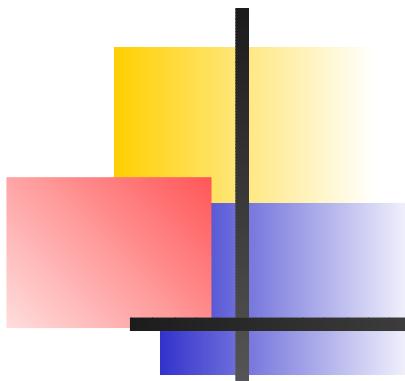
`Jean-Charles.Faugere@inria.fr` - <http://fgbrs.lip6.fr/jcf>

**Fabrice Rouillier**

`Fabrice.Rouillier@inria.fr` - <http://fgbrs.lip6.fr/~rouillie>

SALSA (INRIA) project and CALFOR (LIP6) team

*Paris, France*



# Résoudre

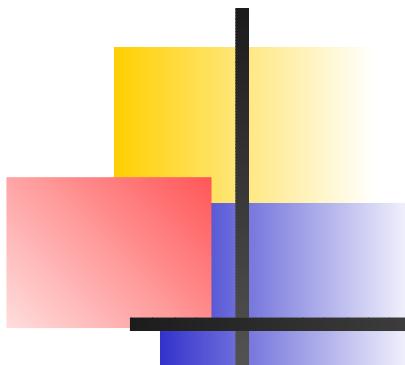
---

On cherche les **solutions** d'un système d'équations:

$$\begin{cases} f_1(x_1, \dots, x_n) = 0 \\ \dots \\ f_m(x_1, \dots, x_n) = 0 \end{cases}$$

il faut parfois rendre le problème algébrique

$$\begin{aligned} \cos(x) &\rightarrow c & c^2 + s^2 - 1 = 0 \\ \sin(x) &\rightarrow s \end{aligned}$$

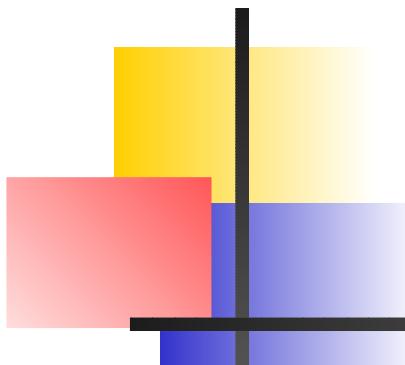


# Résoudre

---

$$f_1 = \dots = f_m = 0$$

- Calculer toutes les solutions: dans  $\mathbb{C}$
- Solutions réelles:  $\mathbb{R}$
- Solutions vérifiant des contraintes  $g(x) > 0$



# Calcul Formel

---

Les proposées sont algébriques:

- Calcul exact
- On calcule *toutes* les solutions
- Méthodes certifiées.

# The general problem

Find all solutions:

$$\left\{ \begin{array}{l} f_1(X_1, \dots, X_n) = 0 \\ f_2(X_1, \dots, X_n) = 0 \\ \vdots \\ f_m(X_1, \dots, X_n) = 0 \end{array} \right. . \quad (1)$$

Fundamental difference:

- $m \geq n$  the corresponding system has a finite number of solutions
- $m < n$  system with “parameters”

Mathematical objects:

- the ideal generated by the equations:  $I = \langle f_1, \dots, f_m \rangle$
- the roots:  $\mathcal{V}_K = \{(x_1, \dots, x_n) \in K^n, s.t. f_i(x_1, \dots, x_n) = 0\}$   
important case:  $\mathcal{V}_{\mathbb{R}}$  and  $\mathcal{V}_{\mathbb{C}}$

# Zero-dimensional systems

- Rewriting ideals (Gröbner bases)
- Transforming a zero-dimensional system into a univariate problem (Gröbner bases+RUR)
- Solving univariate problems (Uspensky)

$$1 \left\{ \begin{array}{l} f_1(X_1, \dots, X_n) = 0 \\ f_2(X_1, \dots, X_n) = 0 \\ \vdots \\ f_m(X_1, \dots, X_n) = 0 \end{array} \right.$$

$$2 \left\{ \begin{array}{l} \text{Groebner basis:} \\ (\text{ordre lexico}) \\ f_n(X_n) = 0 \\ X_{n-1} = f_{n-1}(X_n) \\ \vdots \\ X_1 = f_1(X_n) \end{array} \right. \quad (2)$$

$$3 \left\{ \begin{array}{l} \text{RUR:} \\ f(T) = 0 \\ X_{n-1} = \frac{g_{n-1}(T)}{g'(T)} \\ \vdots \\ X_1 = \frac{g_1(T)}{g'(T)} \end{array} \right. \longrightarrow$$

$$\begin{aligned} 4 \text{ Real Roots:} \\ X_n^{(1)} &\in \left[ \frac{a}{2^N}, \frac{a+1}{2^N} \right] \\ X_n^{(2)} &\in \left[ \frac{a}{2^N}, \frac{a+1}{2^N} \right] \\ &\dots \end{aligned}$$

# Algorithms

- Bad algorithms
- Bad implementations

## Inefficient Methods

For instance:

Direct Kinematic Problem (*DKP*) → Hours of CPU (Mathematica)

- Recent algorithms
- Implementations (C, Memory Management, ...)

## Efficient Methods

Direct Kinematic Problem (*DKP*) → 1 second of CPU

## Computing Groebner bases

**Definition 1.**  $G$  is a Gröbner basis of an ideal  $\mathcal{I}$ , if  $\forall f \in \mathcal{I}$  there exists  $g \in G$  s.t.  $LM(g)$  divides  $LM(f)$ .

Historical algo: Buchberger's algorithm (1965).

Recently, more efficient algorithms have been proposed:

- $F_4$  algorithm : intensive use of linear algebra methods: in short, the arbitrary choices are left to computational strategies related to classical linear algebra problems (mainly the computation of row echelon form).
- A new criterion (the  $F_5$  criterion) for detecting useless computations has been given; under some regularity conditions on the system, it is proved that the algorithm do never perform useless computations. A new algorithm named  $F_5$  has been built using these two ideas.

## Algorithms $F_5$ and $F_4$

$F_5$  constructs incrementally the following matrices in degree  $d$ :

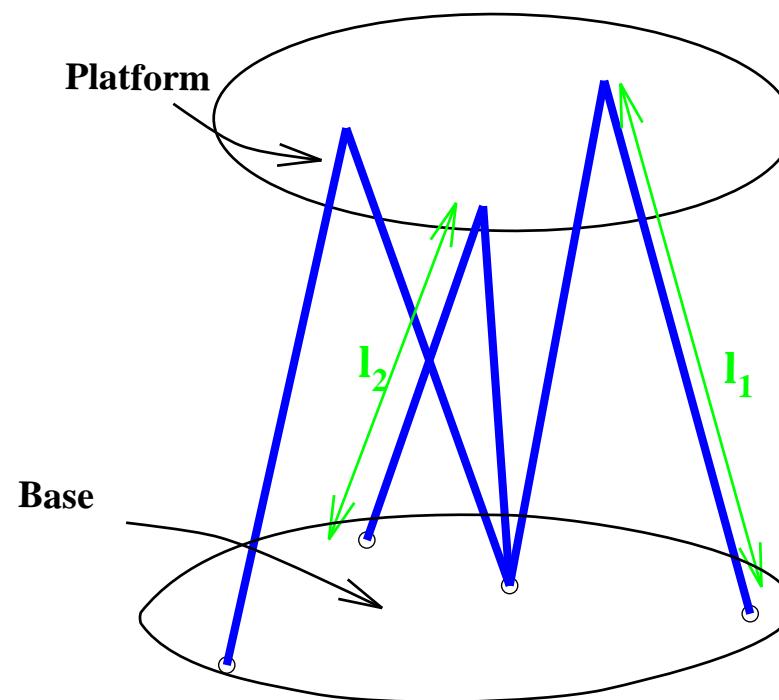
$$A_d = \begin{matrix} & m_1 > m_2 > m_3 & \dots \\ \begin{matrix} t_1 f_1 \\ t_2 f_2 \\ t_3 f_3 \\ \dots \end{matrix} & \left[ \begin{array}{cccc} \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \end{array} \right] \end{matrix}$$

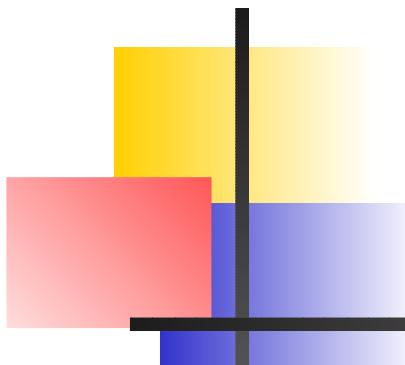
For a regular system the matrices  $A_d$  are full rank. In a second step, row echelon forms of the matrices are computed:

$$A'_d = \begin{matrix} & m_1 & m_2 & m_3 & \dots \\ \begin{matrix} t_1 f_1 \\ t_2 f_2 \\ t_3 f_3 \\ \dots \end{matrix} & \left[ \begin{array}{cccc} 1 & 0 & 0 & \dots \\ 0 & 1 & 0 & \dots \\ 0 & 0 & 1 & \dots \\ 0 & 0 & 0 & \dots \end{array} \right] \end{matrix}$$

# Parallel Robot

Dietmaier (1998): numerical global optimization program, → an example of a robot with 40 “real” roots.





# Parallel Robot

---

length of the actuators  $L_i = ||A_iB_i||$

**Displacement based equations:** let  $R_f$  (resp.  $R_m$ ) be the base Cartesian reference frame of center  $O$  (resp. reference frame of center  $C$  relative to the mobile platform).

then there exists a rotation  $\mathcal{R}$  such that :

$$\overrightarrow{OB}_{i|R_f} = \overrightarrow{OC}_{|R_f} + \mathcal{R} \cdot \overrightarrow{CB}_{i|R_m} , \quad i = 1 \dots 6 \quad (2)$$

# Parallel Robot

First idea:

- Variables:  $\overrightarrow{OC}_{|R_f} = [X, Y, Z]$  and  $\mathcal{R}$  12
- Write the corresponding equations:

$$\mathcal{R} \cdot \mathcal{R}^T = \mathcal{I}$$

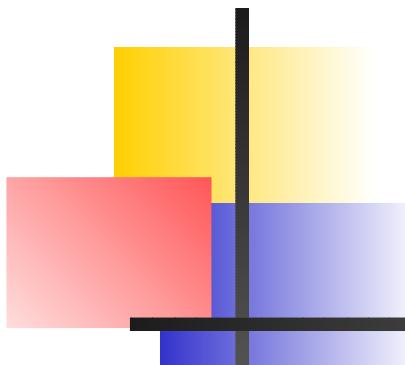
$$\overrightarrow{OB}_{i|R_f} = [X, Y, Z] + \mathcal{R} \cdot \overrightarrow{CB}_{i|R_m} , \quad i = 1 \dots 6$$

$$L_i^2 = \|A_i B_i\|^2$$

# Parallel Robot

Better idea: any rotation  $\mathcal{R}$  can be expressed using the Cayley transform:

$$\mathcal{H} = \begin{bmatrix} 0 & a & b \\ -a & 0 & c \\ -b & -c & 0 \end{bmatrix} \longrightarrow \mathcal{R} = \frac{\mathcal{I} + \mathcal{H}}{\mathcal{I} - \mathcal{H}}$$



# Parallel Robot

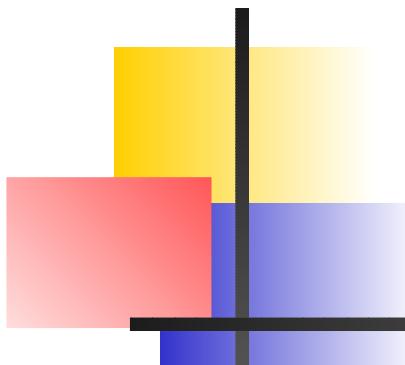
---

$$\mathcal{R} = \frac{\mathcal{I} + \mathcal{H}}{\mathcal{I} - \mathcal{H}}$$

One obtain a system depending on 6 variables  $[X, Y, Z, a, b, c]$ .

Knowing  $a, b$  and  $c$  it is obvious to recover  $[X, Y, Z]$  from a *linear* system.

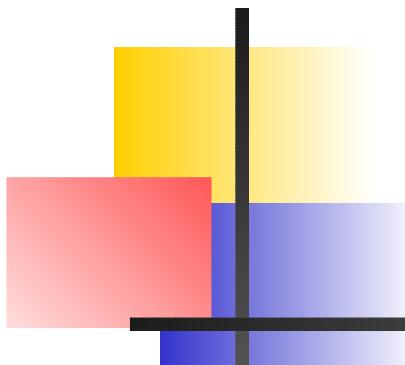
Thus it is enough to compute a Gröbner basis of the corresponding algebraic system for an ordering eliminating  $[X, Y, Z]$  (**3** variables).



# Parallel Robot

---

Isolation and certification of the real coordinates is then computed: we found **40 real roots** in approximatively **1.1 sec** (PC Intel Xeon 2.8 Ghz).

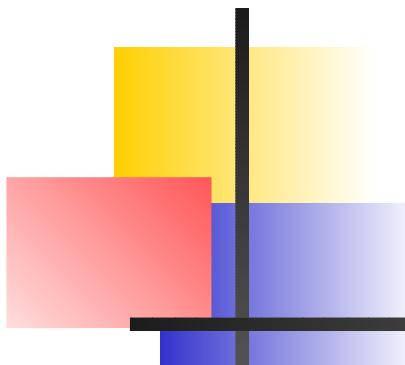


# Syst. Alg. avec paramètres

problèmes avec moins d'équations que de variables:

- $X$  variables
- $U$  paramètres

**Problème:** le nb de solutions est **infini** !



# Syst. Alg. avec paramètres

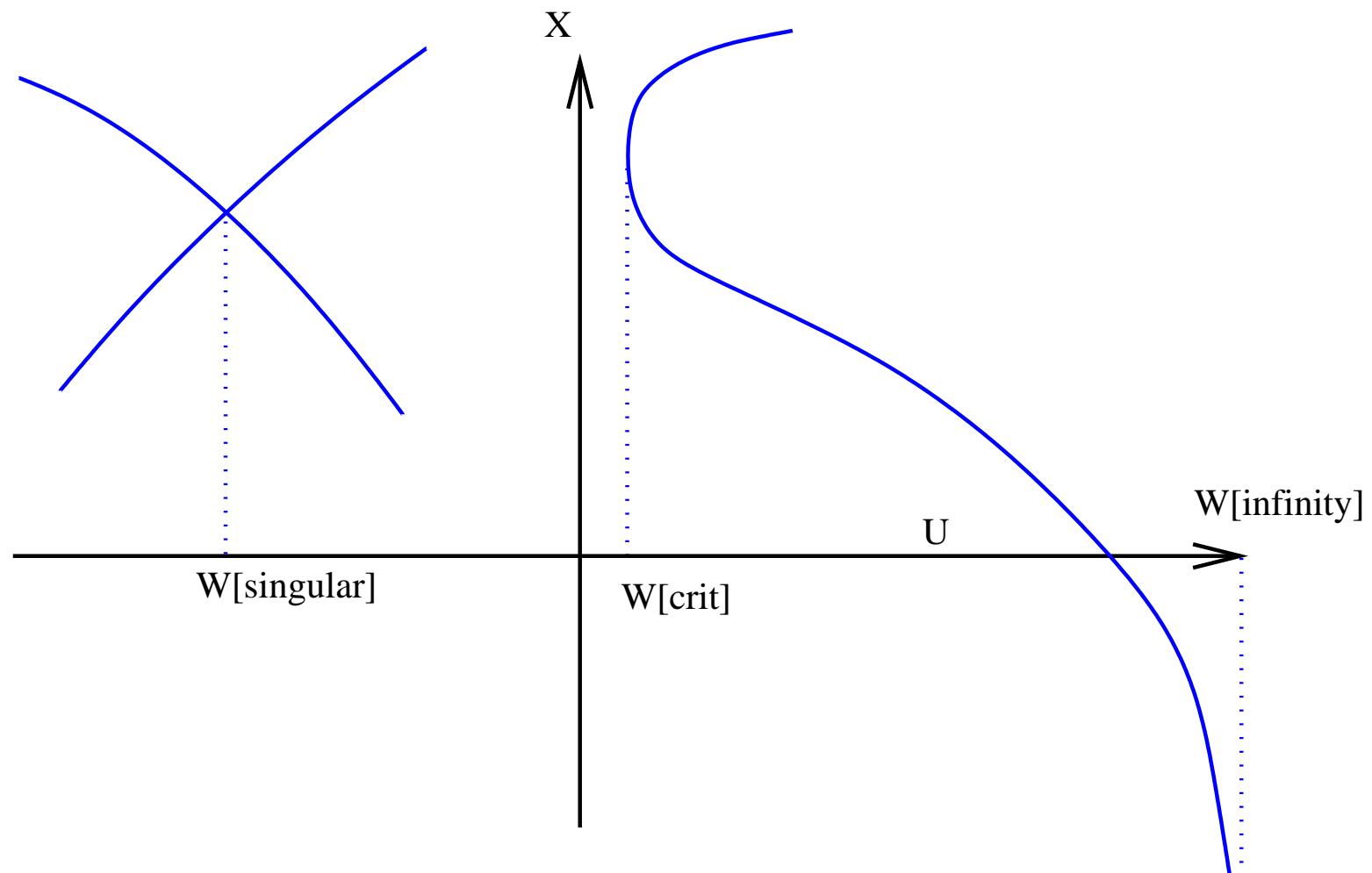
---

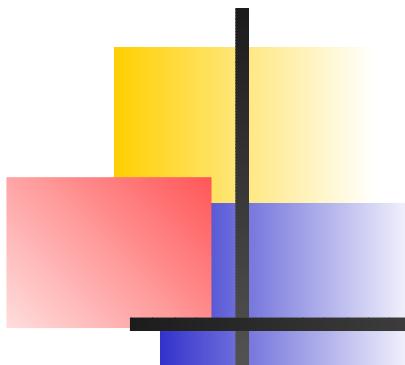
**Idée:** Calculer les points où qqc change !

**Hypothèses:** Nb fini de solutions, sans multiplicités quand on spécialise les paramètres.

**Exemple:** des points réels  $\rightarrow$  complexes  
sur un dessin: les solutions vont à l'infini . . .

# Syst. Alg. avec paramètres





# Variété discriminante

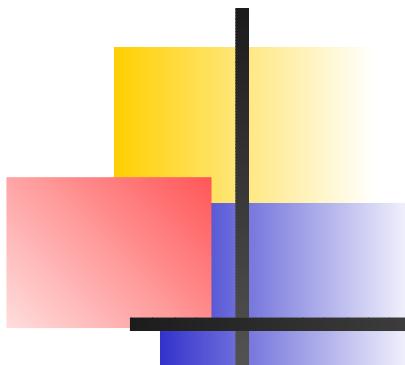
---

**Idée:** Calculer les points où qqc change !

**Objet Mathématique:**

$$\mathcal{W}_D = \mathcal{W}_\infty \cup \mathcal{W}_{\text{crit}} \cup \mathcal{W}_{\text{sing}}$$

**Théorème:**  $\mathcal{W}_\infty \cup \mathcal{W}_{\text{crit}} \cup \mathcal{W}_{\text{sing}}$  est un **fermé**.



# Variété discriminante

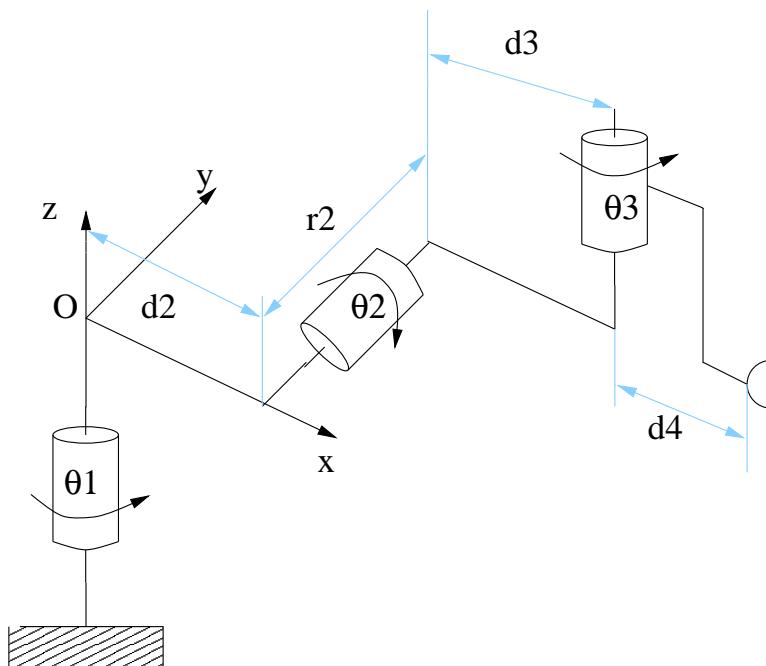
---

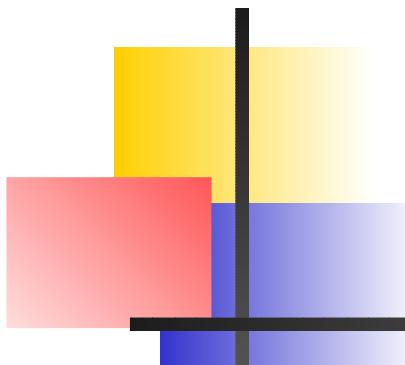
$$\mathcal{W}_D = \mathcal{W}_\infty \cup \mathcal{W}_{\text{crit}} \cup \mathcal{W}_{\text{sing}}$$

- On peut montrer que c'est l'objet le plus petit.
- On peut "lire"  $\mathcal{W}_\infty$  le résultat sur un calcul de Bases de Gröbner.
- $\mathcal{W}_{\text{crit}} \cup \mathcal{W}_{\text{sing}}$  correspond à rajouter la Jacobienne:  $\text{Jac}_X$ .

# Cuspidal robots

The goal was to compute a **classification** of 3-revolute-jointed manipulators based on the cuspidal behavior.





# Cuspidal robots

---

This ability to **change posture** without meeting a singularity is equivalent to the existence of a point s.t. a polynomial of degree 4 depending on the parameters has **a triple root**.

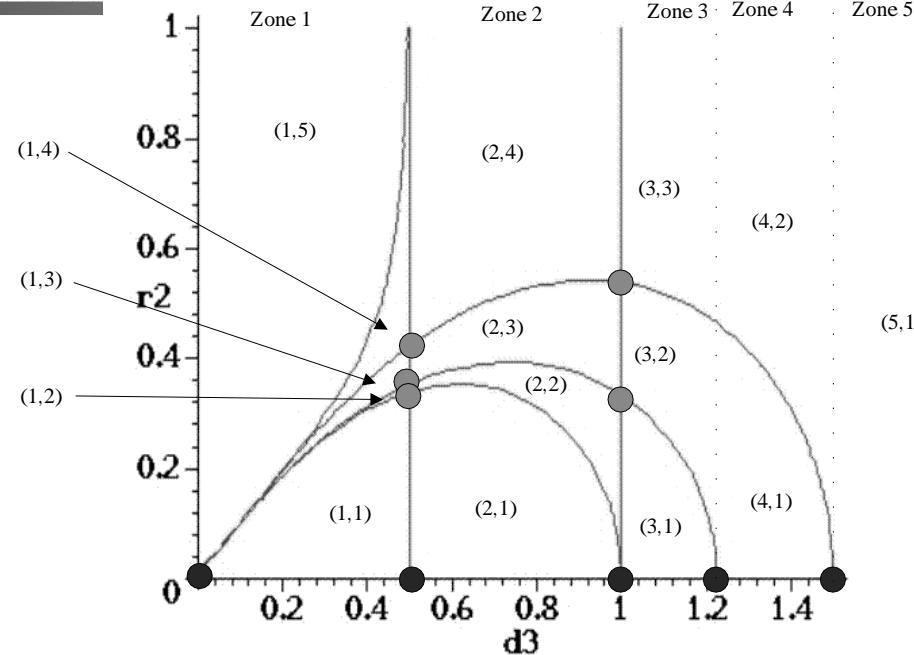
# Cuspidal robots

Cuspidal robots depends on 3 parameters  $d_4, d_3$  and  $r_2$  which are the design parameters (supposed to be positive). It is given by:

$$P(t) = at^4 + bt^3 + ct^2 + dt + e = 0, \frac{\partial P}{\partial t} = 0, \frac{\partial^2 P}{\partial t^2} = 0, d_4 > 0, d_3 > 0, r_2 > 0$$

with: 
$$\begin{cases} a &= m_5 - m_2 + m_0 \\ b &= -2m_3 + 2m_1 \\ m_0 &= -r^2 + r_2^2 + \frac{(R+1-L)^2}{4} \\ m_1 &= 2r_2d_4 + (L-R-1)d_4r_2 \\ m_2 &= (L-R-1)d_4d_3 \\ m_3 &= 2r_2d_3d_4^2 \\ m_4 &= d_4^2(r_2^2 + 1) \end{cases} \quad \begin{cases} c &= -2m_5 + 4m_4 + 2m_0 \\ d &= 2m_3 + 2m_1 \\ e &= m_5 + m_2 + m_0 \\ m_5 &= d_4^2d_3^2 \\ r^2 &= x^2 + y^2 \\ R &= r^2 + z^2 \\ L &= d_4^2 + d_3^2 + r_2^2 \end{cases}$$

# Cuspidal robots



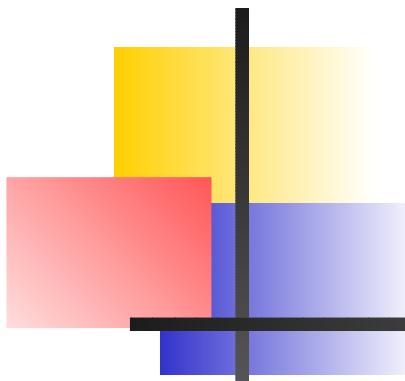
Partition of the parameters' space  $(d_3, r_2)$   
Over each open cell, there are exactly **six sheets**  
on the discriminant variety [Rouillier, Corvez].

# Cuspidal robots

$(d_3, r_2) \setminus d_4$	1	2	3	4	5	6	7
(1,1)	0	0	4	4	2	0	0
(1,2)	0	4	4	4	2	0	0
(1,3)	0	4	4	4	2	0	0
(1,4)	0	4	4	2	2	0	0
(1,5)	0	4	4	2	0	0	0
(2,1)	0	0	4	4	2	2	0
(2,2)	0	4	4	4	2	2	0
(2,3)	0	4	4	4	2	2	0

...

Number of real solutions for each cell.



# Conclusion

---

Esquissé:

- Calculer un nombre fini de solutions.
- Systèmes avec (peu) de paramètres.

En préparation:

- Calcul hybride exact/approché ( $10^{-3}$  sec)
- Systèmes sur-déterminés.