TP Sage: Calcul formel en modélisation géométrique et robotique

École jeunes chercheurs en informatique mathématique

Vendredi 23 mars

Manipulation de polynômes et d'idéaux

Dans *sage*, les polynômes se manipulent en déclarant au préalable l'anneau des polynômes dans lequel on les considère. C'est la même approche qui est utilisée dans les logiciels *Singular* ou *magma* par exemple.

Dans l'anneau le plus général, le moteur de calcul principal est *Maxima*.

```
sage: var('x,y');
sage: p = x^3-y^2
sage: p.parent()
Symbolic Ring
```

Dans les anneaux plus spécifiques, les calculs sont effectués notamment par les librairies *FLINT* (anneaux en 1 variable) ou *Singular*.

```
sage: R.<x,y> = PolynomialRing (QQ,2)
sage: p = x^3-y^2
sage: p.parent()
Multivariate Polynomial Ring in x, y over Rational Field
```

Dans un anneau polynomial $R = \mathbb{Q}[x_1, ..., x_k]$, on définit naturellement *l'idéal* I engendré par une liste de polynôme $p_1, ..., p_k$:

$$I = \{q_1p_1 + \cdots + q_kp_k \mid q_1, ..., q_k \in R\}$$

En *sage*, on le construit simplement en multipliant l'anneau par la liste de polynôme:

```
sage: I = R*[p1,...,pk]
```

On notera qu'un point est solution des équations $p_1 = 0, ..., p_k = 0$ si et seulement si il annule tous les polynômes de I. Par abus de langage, on appelle *solutions* ou *zéros* de I l'ensemble des points annulant tous les polynômes de I.

Les principales fonctions sur les idéaux que nous utiliserons sont:

- *I.dimension()*: renvoie la dimension de I (par exemple 0 si I a un nombre fini de zéros)
- *I.variety()* : renvoie les zéros de I si sa dimension est 0.

- *I.elimination_ideal([x_i,...,x_j])*: renvoie l'ensemble des polynômes de I qui ne contiennent pas les variables $x_i, ..., x_j$. On notera que l'idéal d'élimination a pour solution la projection des solutions de I presque partout.
- *I.primary_decompostion()*: renvoie les composante primaires de I. Dans le cas d'un idéal engendré par un seul polynôme, cette fonction est équivalente à la factorisation.
- *I.plot()*: dans le cas où I est engendré par un polynôme et R est un anneau en 2 variables, cette fonction permet d'afficher la courbe implicite des solution réelles de I.

Exercice 1: Préliminaires

1. Construisez l'anneau de polynôme $\mathbb{Q}[x, y, z]$
2. Soient $p_1 = x^2 + y^2 + z^2 - 4$ et $p_2 = x^2y^2z^2 - 1$. Affichez les zéros de chacun de ces polynômes avec la fonction *implicit_plot3d*. Construisez I l'idéal qu'ils engendrent.
3. Construisez J l'idéal d'élimination de I où l'on a retiré la variable z.
4. La fonction *J.plot()* requiert que *J.ring()* soit un anneau en 2 variables. Modifiez J à l'aide de la fonction *change_ring* pour pouvoir afficher ses solutions.
5. Écrivez la fonction *plot_projection* qui prend en entrée un idéal I de dimension 1 et 2 variables x, y, et qui affiche la projection des solutions de I dans le plan x, y.

   Indice: si R est un anneau polynomial de sage, alors la liste de ses variables autres que x et y est donnée par

   ```
   sage: V = R.gens()
   sage: L = [v for v in V if v not in (x,y)]
   sage: R.remove_var(*L)
   ```

Mécanisme robotique

On considère maintenant le robot parallèle plan $\underline{P}R - \underline{P}RR$ en Figure 1. Les variables de poses sont x, y, articulaires sont r_1, r_2 et les variables θ_1, θ_2 sont passives. Dans la suite, on remplacera θ_1, θ_2 par t_1, t_2 tels que:

$$\begin{cases} cos(\theta_1) = \frac{1-t_1^2}{1+t_1^2} \\ sin(\theta_1) = \frac{2t_1}{1+t_1^2} \end{cases} \qquad \begin{cases} cos(\theta_2) = \frac{1-t_2^2}{1+t_2^2} \\ sin(\theta_2) = \frac{2t_2}{1+t_2^2} \end{cases}$$

On considère par ailleurs que l'angle α_1 (resp. α_2) de la prismatique de gauche (resp. de droite) avec l'horizontal vérifie $cos(\alpha_1) = -\frac{3}{5}$ et $sin(\alpha_1) = \frac{4}{5}$ (resp. $cos(\alpha_2) = \frac{3}{5}$ et $sin(\alpha_2) = \frac{4}{5}$).

Exercice 2: Analyse des singularités parallèles

1. Définissez l'anneau de polynôme à 6 variables sur le corps des rationnels.

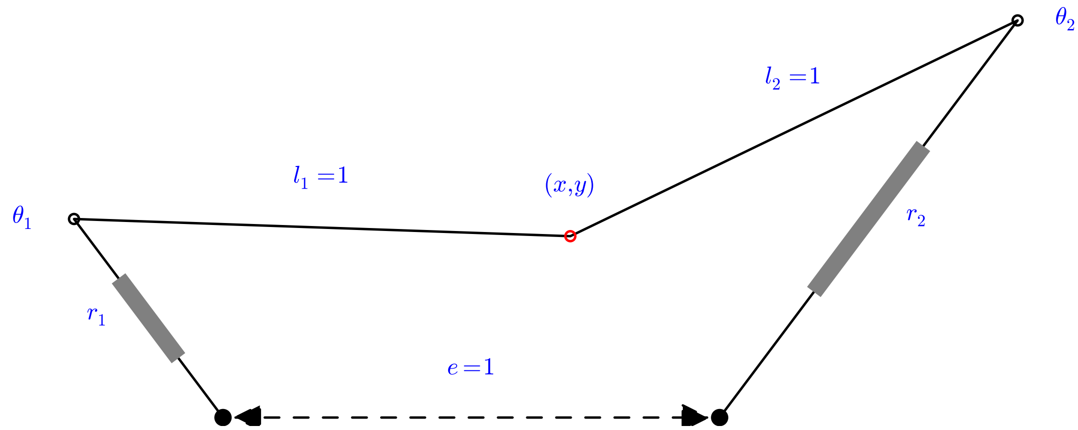

Figure 1: Robot parallèle PR-PRR

2. Chaque jambe du robot parallèle induit 2 équations de contraintes. Écrivez les 4 équations décrivant induite par la géométrie du robot sous la forme:

$$q_1 == 0, q_2 == 0, q_3 == 0, q_4 == 0$$

   Dans sage, la fonction *numerator()* permet d'éliminer le dénominateur qui dans notre cas n'est jamais nul. Stockez dans la variable *Equations* la liste des numérateurs p_i des q_i.

3. Pour le calcul des points critiques (ou singularités parallèles) de ce robot parallèle, on doit calculer le Jacobien j des équations de contraintes par rapport aux variables de pose et aux variables passives. Utilisez les fonctions *jacobian* et *det* pour le calculer.

4. On veut maintenant visualisez la projection des points critiques dans l'espace de travail (variables x, y). Utilisez la fonction *plot_projection* construite en **1.5** pour visualiser l'idéal engendré J par les p_i et par j.

5. Afin de mieux comprendre d'où viennent les différentes courbes de points critiques, décomposez l'idéal J à l'aide de la fonction *primary_decompostion*. Vous obtiendrez une liste d'idéaux, dont l'union des solutions est l'ensemble des solutions de J. Avec la fonction *elimination_ideal*, éliminez maintenant, pour chacun des idéaux primaires, les variables x, y, t_1, t_2. Vous obtiendrez ainsi les idéaux correspondants en les variables r_1 et r_2. Au vu des résultats, pouvez-vous maintenant décrire à quelles configurations du mécanismes les points critiques correspondent-ils?