

# Algorithmes pour la planification de mouvements en robotique non-holonyme

Frédéric Jean

Unité de Mathématiques Appliquées  
ENSTA

Le 02 février 2006



## Outline

- 1 Robotique non-holonyme
  - Les systèmes non-holonomes
  - Commandabilité
  - Principes pour la planification
- 2 Méthodes de planification existantes
  - Méthodes basées sur l'algèbre de Lie
  - Équivalence des systèmes
- 3 Nos axes de recherche
  - Algorithme par approximation
  - Méthode de continuation



## Modélisation Géométrique d'un Robot

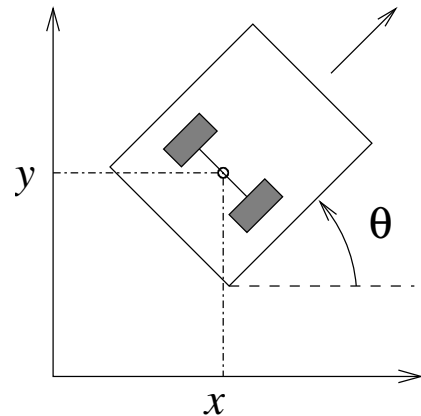
- Robot = point dans  $E$ , **espace des configurations**
- En général,  $E =$  un ouvert d'un  $\mathbb{R}^n$  ou une variété

### Exemple type

La voiture simplifiée (ou unicycle) :

$$(x, y, \theta) \in E = \mathbb{R}^2 \times S^1 \text{ ou } \mathbb{R}^3$$

- Obstacles = ensemble fermé  $O \subset E$   
 $\Rightarrow$  espace de travail =  $E \setminus O$



## Modélisation du Mouvement

Deux types de robots :

- **holonome** : pas de restriction sur le déplacement  
 $\rightarrow$  tout chemin dans  $E$  est un mouvement autorisé
- **non-holonome** : soumis à des contraintes cinématiques  
 $\rightarrow$  mouvements autorisés = chemins  $q(t)$  dans  $E$  t.q.  
 $\dot{q}(t) \in V_{q(t)} \subsetneq \mathbb{R}^n \quad \forall t$

Voiture = robot non-holonome

Contrainte de roulement sans glissement :  $\dot{x} \sin \theta - \dot{y} \cos \theta = 0$

$$\Rightarrow \dot{q} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} \in \text{Vect} \left\{ \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \end{pmatrix} \right\} \subsetneq \mathbb{R}^3$$



## Exemples de Robots Non-Holonomes

- Contraintes de roulement sans glissement :
  - robots mobiles à roues ;
  - manipulation par des mains robotisées.
- Conservation du moment angulaire :
  - robots manipulateurs flottant dans l'espace ;
  - satellites avec roues à inertie ;
  - robots astronautes, plongeurs, sauteurs (phase de vol).
- Autres sources de contraintes non-holonomes :
  - loi de Lorentz-Wong en électromagnétisme ;
  - réduction par symétrie de cinématiques holonomes.



## Problème de la Planification des Mouvements

### Problème (PPM)

Étant données deux configurations  $q_0$  et  $q_1 \in E$ ,  
trouver un chemin **faisable**  $q(t)$ ,  $t \in [0, T]$ , dans  $E$

$$\text{t.q. } q(0) = q_0 \text{ et } q(T) = q_1$$



## Point de Vue de la Théorie du Contrôle

- En général, pour  $q$  fixé,  $V_q =$  partie d'un SEV, i.e.

$$V_q = \{u_1 f_1(q) + \dots + u_m f_m(q) : u \in U \subset \mathbb{R}^m\} \quad (m < n)$$

où  $f_i : E \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$  est appelé un **champ de vecteurs**.

- $q(t)$  est un chemin faisable  $\Leftrightarrow$  il existe  $u : [0, T] \rightarrow U$  t.q.

$$\dot{q}(t) = u_1(t) f_1(q(t)) + \dots + u_m(t) f_m(q(t))$$

Chemins faisables = trajectoires du **système non-holonomie**

$$\dot{q} = u_1 f_1(q) + \dots + u_m f_m(q),$$

où  $u = (u_1, \dots, u_m) \in U$  est le **contrôle**.



## Exemple de la Voiture

Système de contrôle :

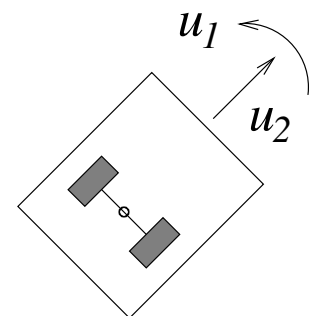
$$\dot{q} = u_1 f_1(q) + u_2 f_2(q)$$

avec les champs de vecteurs

$$f_1(q) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad f_2(q) = \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \end{pmatrix}$$

et le contrôle  $u = (u_1, u_2)$  :

$$\begin{cases} u_1 = \dot{\theta} & \text{vitesse angulaire} \\ u_2 = \pm \sqrt{\dot{x}^2 + \dot{y}^2} & \text{vitesse tangentielle} \end{cases}$$



## Reformulation du PPM

### Problème de la Planification des Mouvements

Étant données deux configurations  $q_0$  et  $q_1 \in E$ ,  
trouver une loi de contrôle  $u(t)$ ,  $t \in [0, T]$ , dans  $U$

t.q. la solution de

$$\begin{cases} \dot{q}(t) = u_1(t)f_1(q(t)) + \dots + u_m(t)f_m(q(t)) \\ q(0) = q_0 \end{cases}$$

vérifie  $q(T) = q_1$

(i.e.  $u(\cdot)$  amène le système de  $q_0$  à  $q_1$ ).



## Classe des Contrôles

- En théorie, contrôle = fonction intégrable ( $L^1$ )
- En pratique, contrôle = **fonction continue par morceaux**

Cas des extensions dynamiques :

Très souvent, le contrôle "physique" est  $\dot{u}$ , et non  $u$ .

**Exemple : contrôles "réels" de la voiture**

$$\begin{cases} \dot{u}_1 = \ddot{\theta} & \text{accélération angulaire (via le volant)} \\ \dot{u}_2 & \text{accélération tangentielle (accélérateur/frein)} \end{cases}$$

→ Même formulation pour le PPM, mais

$u(t) =$  **fonction dérivable** et contrôle réel =  $\dot{u}$ .



## Décidabilité du PPM

Dans toute la suite, **robot = système non-holonomie**

### Problème de la commandabilité

Étant données deux configurations  $q_0$  et  $q_1 \in E$ ,  
existe-t-il une loi de contrôle  $u(\cdot)$  dans  $U$   
amenant le système de  $q_0$  à  $q_1$  ?

Si c'est toujours le cas, le système est dit **commandable**.



## Crochets de Lie

### Exemple de la Voiture

Contrainte : direction transverse interdite

créneau possible  $\implies$  voiture = système commandable

### Explication

Créneau = suivre  $f_2$ , puis  $f_1$ , puis  $-f_2$ , puis  $-f_1$ .

$$\text{Or : } e^{-tf_1} e^{-tf_2} e^{tf_1} e^{tf_2}(q_0) = e^{t^2[f_1, f_2]}(q_0) + o(t^2),$$

où  $[f_1, f_2](q) = Df_2(q) \cdot f_1(q) - Df_1(q) \cdot f_2(q)$  **crochet de Lie**.

Pour la voiture,  $[f_1, f_2] = \begin{pmatrix} -\sin \theta \\ \cos \theta \\ 0 \end{pmatrix} =$  direction transverse



# Algèbre de Lie

## Définition

$\text{Lie}(q) =$  l'ensemble des combinaisons linéaires de

- $f_1(q), \dots, f_m(q),$
- tous les  $[f_i, f_j](q),$  pour  $i, j \in \{1, \dots, m\},$
- tous les  $[[f_i, f_j], f_k](q),$  pour  $i, j, k \in \{1, \dots, m\},$
- ...

est appelé **l'algèbre de Lie** du système.

Remarque :  $\text{Lie}(q)$  est un sous-espace vectoriel de  $\mathbb{R}^n.$

Pour la voiture,  $\text{Lie}(q) = \mathbb{R}^3.$



# Commandabilité

$$(\Sigma) \quad \dot{q} = u_1 f_1(q) + \dots + u_m f_m(q), \quad q \in E, \quad u \in U$$

## Théorème de Chow (1938)

Supposons  $E \subset \mathbb{R}^n$  ouvert, connexe et  $U \subset \mathbb{R}^m$  contient  $\text{Vois}(0).$

$$\text{Si } \dim \text{Lie}(q) = n \quad \forall q \in E$$

le système  $(\Sigma)$  est commandable.

Remarque : Dans ce cas, la commandabilité est vraie pour toutes les classes de contrôles denses dans  $L^1$ , par exemple  $C^\infty$  ou constants par morceaux.



# Hypothèse

Dans toute la suite, on supposera que

**la condition de Chow est satisfaite**

⇒ Le PPM a toujours une solution.

Remarque : Si la condition n'est pas satisfaite, on peut généralement s'y ramener en reparamétrant.



# Méthodes de Planification des Mouvements

Quatre idées générales pour la planification :

- mimer les crochets de Lie (créneaux !);
- se ramener à un système connu ;
- utiliser une approximation du système ;
- déformer continûment une trajectoire.

Méthodes  
existantes

Nos axes de  
recherche





# Méthodes Basées sur l'Algèbre de Lie

$$(\Sigma) \quad \dot{q} = u_1 f_1(q) + \dots + u_m f_m(q), \quad q \in E$$

## Principe

- 1 Choisir des crochets de Lie  $g_{m+1}, \dots, g_n$  de  $f_1, \dots, f_m$  t.q.

$$\text{Vect} \{f_1, \dots, f_m, g_{m+1}, \dots, g_n\}(q) = \mathbb{R}^n$$

- 2 Résoudre le PPM pour le système **holonome**

$$\dot{q} = v_1 f_1(q) + \dots + v_m f_m(q) + v_{m+1} g_{m+1}(q) + \dots + v_n g_n(q).$$

- 3 Écrire  $v(\cdot)$  en fonction du contrôle  $u(\cdot)$  du système  $(\Sigma)$ .



# Calcul de $u(t)$ en Fonction de $v(t)$

[Lafferriere/Sussmann, 1993]

- Pour  $q_1$  suffisamment proche de  $q_0$ , on a :

$$q_1 = e^{t_1 f_1} \dots e^{t_m f_m} e^{t_{m+1} g_{m+1}} \dots e^{t_n g_n}(q_0) \rightarrow v(\cdot)$$

- Pour chaque  $g_i$ ,  $e^{s g_i} = \prod_j e^{\alpha_i^j(s) f_j} + o(s)$ .

- Donc

$$\prod_{i,j} e^{\alpha_i^j(t_i) f_j}(q_0) = q_1 + o(\|t\|) := q^{(1)} \rightarrow u^{(1)}(\cdot)$$

- En itérant à partir de  $q^{(1)}$ , on obtient une suite  $q^{(k)}$ ,

$$q^{(k)} \rightarrow q_1 \quad \text{quand } k \rightarrow \infty.$$



## Exemple de la Voiture

Soient  $q_0 = 0$  et  $q_1 = (x, y, \theta)$

- Choisissons  $g_3 = [f_1, f_2]$  :  $q_1 = e^{\theta f_1} e^{x f_2} e^{y g_3}(0)$

i.e.  $v(t) = (0, 0, y)$ , puis  $(0, x, 0)$ , puis  $(\theta, 0, 0)$

- Pour  $y \geq 0$ ,  $e^{y[f_1, f_2]} = e^{-\sqrt{y}f_1} e^{-\sqrt{y}f_2} e^{\sqrt{y}f_1} e^{\sqrt{y}f_2} + o(y)$ ,

$$\Rightarrow q^{(1)} = e^{\theta f_1} e^{x f_2} e^{-\sqrt{y}f_1} e^{-\sqrt{y}f_2} e^{\sqrt{y}f_1} e^{\sqrt{y}f_2}(0) = q_1 + o(\|q_1\|)$$

$$u^{(1)}(t) = (0, \sqrt{y}), \text{ puis } (\sqrt{y}, 0), \text{ puis } \dots$$



## Calcul de $u(t)$ en Fonction de $v(t)$ – Autre Solution

[Sussmann/Liu, 1991]

Utiliser des contrôles oscillants de fréquence et amplitude  
arbitrairement grands.

### Exemple de la voiture

Prenons  $q_0 = 0$  et  $q_1 = (0, y, 0) = e^{y g_3}(0)$ , i.e.  $v(t) \equiv (0, 0, y)$ .

Appliquons  $u^N(t) = \sqrt{N}(\cos(Nt), \sin(Nt))$ . Quand  $N \rightarrow \infty$  :

$$q^N(t) \sim \left( \frac{1 - \cos(Nt)}{\sqrt{N}}, \frac{t}{2}, \frac{\sin(Nt)}{\sqrt{N}} \right) \Rightarrow q^N(2) \rightarrow q_1.$$



## Inconvénients

- Contrôles de très mauvaise qualité :
  - soit grand nombre de manœuvres et non dérivables ;
  - soit très oscillants.
- Convergence locale.
- Méthode itérative.



## Systèmes Équivalents

### Principe

Transformer  $(\Sigma)$  en un système "connu".

- Transformation par chgt de coordonnées et retour d'état :

$$z = \phi(q), \quad v = A(q)u$$

⇒ un système **équivalent** à  $(\Sigma)$  (mêmes trajectoires) :

$$\dot{z} = v_1 \tilde{f}_1(z) + \cdots + v_m \tilde{f}_m(z),$$

où  $\tilde{f} = \phi_* f$  transport de  $f$  par  $\phi$ .



## Classes de Systèmes Connus

- On sait résoudre le PPM pour :
  - les systèmes linéaires :  $\dot{x} = Ax + Bu$   
MAIS, si  $(\Sigma)$  non-holonome ( $m < n$ ) et commandable,  
 **$(\Sigma)$  n'est jamais équivalent à un syst. linéaire ;**
  - les systèmes nilpotents ;
  - les systèmes chaînés (systèmes nilpotents particuliers).



## Systèmes Nilpotents

### Définition

$(\Sigma)$  est **nilpotent d'ordre  $k$**  si tous les crochets de Lie itérés des  $f_1, \dots, f_m$  de longueur  $> k$  sont nuls.

### Intérêt :

- La “formule du créneau” est exacte : si  $g$  crochet de Lie,

$$e^{tg} = \prod_j e^{\alpha^j(t)f_{i_j}}$$

$\implies$  la méthode de planification [LafSu93] est :  
directe (pas d'itération), **exacte** et globale.



## Systèmes Nilpotents (Suite)

- Le système est polynômial et de forme triangulaire :

$$\dot{q}_i = \sum_{j=1}^m u_j \text{pol}_i(q_1, \dots, q_{i-1})$$

⇒ intégrable avec des commandes polynômiales.

### Inconvénients majeurs :

- nilpotentisation = propriété non générique ;
- pas de critère général de nilpotentisation.

[E. Cartan, Goursat]



## Nilpotentisation de la Voiture

- En appliquant la transformation  $z = \phi(q)$ ,  $v = A(q)u$  :

$$\begin{cases} z_1 = x \cos \theta + y \sin \theta \\ z_2 = \theta \\ z_3 = x \sin \theta - y \cos \theta \end{cases}, \quad \begin{cases} v_1 = u_2 - u_1 z_3 \\ v_2 = u_1 \end{cases}$$

le système s'écrit

$$\begin{cases} \dot{z}_1 = v_1 \\ \dot{z}_2 = v_2 \\ \dot{z}_3 = v_2 z_1 \end{cases} \Rightarrow \tilde{f}_1(z) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \tilde{f}_2(z) = \begin{pmatrix} 0 \\ 1 \\ z_1 \end{pmatrix}$$

- On a  $[\tilde{f}_1, \tilde{f}_2] = (0, 0, 1)$  et tous les autres crochets sont nuls.



# Systèmes Chaînés

Définition [Murray/Sastry, 1991]

$(\Sigma)$  est un **système chaîné** s'il s'écrit ( $m = 2$ ) :

$$\begin{aligned}\dot{q}_1 &= u_1 \\ \dot{q}_2 &= u_2 \\ \dot{q}_3 &= u_1 q_2 \\ &\vdots \\ \dot{q}_n &= u_1 q_{n-1}\end{aligned}$$

Intérêt :

- facile à commander (nilpotent) ;
- il existe une CNS pour la mise sous forme chaînée ;
- systèmes linéarisables dynamiquement (**systèmes plats**).



# Algorithmes par Approximation

[F. Jean, J-P. Laumond, G. Oriolo, M. Vendittelli, 2001 - 2006]

Principe

- 1 Choisir un système :  $(\hat{\Sigma}) \quad \dot{q} = \sum_{i=1}^m u_i \hat{f}_i(q), \quad \text{t.q.}$ 
  - $(\hat{\Sigma})$  est une approximation de  $(\Sigma)$  en  $q_1$ ,
  - on a une solution du PPM pour  $(\hat{\Sigma})$ .
- 2 Choisir  $u(\cdot)$  amenant  $(\hat{\Sigma})$  de  $q_0$  à  $q_1$ .
- 3 Appliquer  $u(\cdot)$  à  $(\Sigma)$  à/p de  $q_0 \rightarrow q^{(1)} := \text{Approx}(q_0, q_1)$ .
- 4 Itérer le processus  $\rightarrow q^{(k+1)} := \text{Approx}(q^{(k)}, q_1)$ .



# Convergence

Que faut-il pour obtenir un algorithme convergent ?

- **Localement** : la fonction Approx doit être contractante, i.e.

pour chaque  $q_1$ ,  $\exists \beta(q_1) > 0$  t.q.

$$d(q^{(k)}, q_1) < \beta(q_1) \implies d(q^{(k+1)}, q_1) \leq \frac{1}{2} d(q^{(k)}, q_1),$$

pour une distance  $d$  à définir.

- **Globalement** : Approx uniformément contractante  
( $\beta$  indépendant de  $q_1$ ).



# Algorithme Global

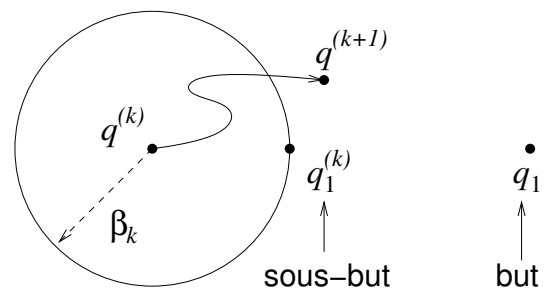
Algorithme de type **régions de confiance** :

## Étape $k$

$q^{(k)}$  = point courant

$\beta_k$  = rayon de confiance

- 1 calculer  $q_1^{(k)}$  ;
- 2  $q^{(k+1)} := \text{Approx}(q^{(k)}, q_1^{(k)})$  ;
- 3 if  $d(q^{(k+1)}, q_1^{(k)}) < \frac{1}{2} d(q^{(k)}, q_1^{(k)})$ 
  - then  $\beta_{k+1} := \beta_k$
  - else  $\beta_{k+1} := \beta_k/2$



→ algorithme convergent si Approx uniformément contractante.



## La Fonction Approx

Il reste à construire une fonction Approx unif<sup>t</sup> contractante.

Rappel : définition de  $\text{Approx}(q_0, q_1)$

① Soit un système :  $(\widehat{\Sigma}) \quad \dot{q} = \sum_{i=1}^m u_i \widehat{f}_i(q), \quad \text{t.q.}$

- $(\widehat{\Sigma})$  est une approximation de  $(\Sigma)$  en  $q_1$ ,
- on peut calculer  $u(\cdot)$  amenant  $(\widehat{\Sigma})$  de  $q_0$  à  $q_1$ .

② Appliquer  $u(\cdot)$  à  $(\Sigma) \rightarrow q^{(1)} := \text{Approx}(q_0, q_1)$ .

③ Approx unif<sup>t</sup> contractante si

$$d(q_0, q_1) < \beta \implies d(q^{(1)}, q_1) \leq \frac{1}{2} d(q_0, q_1).$$



## Approximation au 1er Ordre

• Linéarisé de  $(\Sigma)$  en  $q_1$  :  $\dot{x} = \sum_{i=1}^m v_i f_i(q_1)$

non commandable (si  $m < n$ )

• On utilise une **approximation non-holonome au 1er ordre**

[Agrachev, Hermes, Stefani, Bellaïche, Jean ... 90's]

Pour la voiture, où  $f_1 = (0, 0, 1)$  et  $f_2 = (\cos \theta, \sin \theta, 0)$  :

• linéarisé en  $q = 0$  :  $f_1(0) = (0, 0, 1), f_2(0) = (1, 0, 0)$

• approx<sup>o</sup> NH en  $q = 0$  :  $\widehat{f}_1 = (0, 0, 1), \widehat{f}_2 = (1, \theta, 0)$





## Distance de Contrôle

- **Distance de contrôle** (ou distance sous-riemannienne) :

$$d(q_0, q_1) = \inf \text{long}(u) \quad \text{parmi les } u : q_0 \rightsquigarrow q_1,$$

$$\text{où } \text{long}(u) = \int_0^T \sqrt{u_1^2(t) + \dots + u_m^2(t)} dt.$$

- Si  $(\hat{\Sigma})$  approximation NH de  $(\Sigma)$  en  $q_1$ ,

$$d(\text{Approx}(q_0, q_1), q_1) = O(d(q_0, q_1)^{1+\varepsilon})$$

(à condition que la commande  $u(\cdot)$  pour  $(\hat{\Sigma})$  ne soit pas trop grande).



## Cas Générique

### Théorème

Pour un système  $(\Sigma)$  **générique**, il existe une approximation non-holonyme  $(\hat{\Sigma})$  telle que :

- $(\hat{\Sigma})$  a une expression algébrique en fonction de  $(\Sigma)$  ;  
[ $\Rightarrow$  facile à calculer]
- $(\hat{\Sigma})$  est un système nilpotent ;  
[ $\Rightarrow$  facile à commander]
- une fonction  $\text{Approx}$  basée sur  $(\hat{\Sigma})$  est unif<sup>t</sup> contractante.  
[ $\Rightarrow$  l'algorithme global converge]



# Méthode de Continuation

## Principe

- ① choisir un contrôle  $u^0(\cdot)$  ;
- ② appliquer  $u^0(\cdot)$  à  $(\Sigma)$  à/p de  $q_0 \rightarrow \bar{q}$  ;
- ③ choisir un chemin  $\gamma(s)$ ,  $s \in [0, 1]$ , reliant  $\bar{q}$  à  $q_1$  ;
- ④ déformer  $u^0(\cdot)$  en  $u^s(\cdot)$ , qui amène  $(\Sigma)$  de  $q_0$  à  $\gamma(s)$  ;
- ⑤  $u^1(\cdot) =$  solution du PPM .



# Déformation

- **Application point final** en  $q_0$  et  $T$  :  $\text{End} : u(\cdot) \mapsto q_u(T)$   
où  $q_u(\cdot)$  trajectoire de  $(\Sigma)$  associée à  $u(\cdot)$  partant de  $q_0$ .
- $u^s(\cdot)$  défini par  $\text{End}(u^s) = \gamma(s) \Rightarrow d\text{End}(u^s) \cdot \frac{du^s}{ds} = \frac{d\gamma}{ds}(s)$ .
- Si  $d\text{End}(u^s)$  surjective,  $\exists$  un pseudo-inverse  $d\text{End}(u^s)^\#$   
 $\Rightarrow u^s =$  solution de l'équation différentielle :

$$\begin{cases} \frac{du^s}{ds} = d\text{End}(u^s)^\# \cdot \frac{d\gamma}{ds}(s) \\ u^s|_{s=0} = u^0 \end{cases} \quad s \in [0, 1]$$



## Caractéristiques

### Intérêt

- Méthode exacte, sans itération
- Contrôles obtenus très réguliers
- Très bon comportement numérique
- Permet de prendre en compte des obstacles

### Inconvénient

- Pas au point !!



## Difficulté

**Déf :**  $u$  est un **contrôle singulier** si  $d\text{End}(u)$  non surjective.

L'EDO donnant  $u^s$  n'est définie que sur les contrôles non singuliers  
⇒ phénomène d'explosion près des contrôles singuliers :

solution  $u^s$  pas toujours définie en  $s = 1$

### Programme

- Caractériser les contrôles singuliers.
- Montrer que l'EDO les évite (ou la modifier pour les éviter).



## État de l'Art

### Preuves de convergence et simulations

- Pour une classe de systèmes sans contrôles singuliers  
*[Chitour/Sussmann, 1993-2006]*
- Pour un cas particulier avec contrôles singuliers connus  
*[Alouges/Chelouah/Chitour, 2003]*

### Étude des contrôles singuliers

- Caractérisation des contrôles singuliers pour les systèmes génériques (au sens fort)  
*[Chitour/Jean/Trélat, 2006]*



Et voilà ...

