

**PLANIFICATION DE
TRAJECTOIRE D'UNE
VOITURE AVEC REMORQUE
-
UN CAS PARTICULIER**

Master Recherche Informatique et Instrumentation de
l'Image
Université de Bourgogne

par

Bastien BERRET

Encadrant : Jean-Paul GAUTHIER

Table des matières

1	Introduction	3
1.1	Problématique	3
1.2	Notions préliminaires	4
2	Etude du système cinématique	9
2.1	Mise en équation	9
2.2	Modélisation géométrique du système	11
3	Résolution du problème optimal	13
3.1	Le système nilpotent	13
3.2	Calcul direct de $\gamma(\tilde{w})$	16
3.3	Passage du système nilpotent au système cinématique	18
3.4	Etude du système nilpotent et résolution du problème	19
4	Simulation en MatLab	22
4.1	Problème de représentation	22
4.2	Algorithme	27
4.3	Illustrations, captures d'écran	28
5	Conclusion	31
6	Annexe	32

Remerciements

Je remercie le professeur Jean-Paul Gauthier, mon principal encadrant, qui m'a permis de réaliser mon stage dans les meilleures conditions possibles. Je le remercie pour sa disponibilité et ses conseils scientifiques qui m'ont permis de mener à bien cette étude.

Mes remerciements vont aussi aux personnes du laboratoire Le2i, notamment Mustapha, Mohamed, Ludovic, Ambroise avec qui j'ai toujours eu grand plaisir à discuter.

Reste à remercier mes camarades du Master 3i qui ont contribué au déroulement de ce stage dans une bonne ambiance. Jérôme, François, Vincent en particulier.

Et bien sûr, je n'oublie pas toutes les personnes qui ont pu m'aider directement ou indirectement lors de ce stage.

Chapitre 1

Introduction

1.1 Problématique

Les problèmes de planification de trajectoires ont été beaucoup étudiés dans la littérature ces dernières années. Ils mettent en oeuvre de puissants outils mathématiques et interviennent particulièrement en robotique ainsi que dans la plupart des domaines nécessitant un contrôle optimal constructif.

Notre problématique est l'étude de trajectoires pour une voiture munie d'une remorque dans un cas extrême (voir figure 1.1).

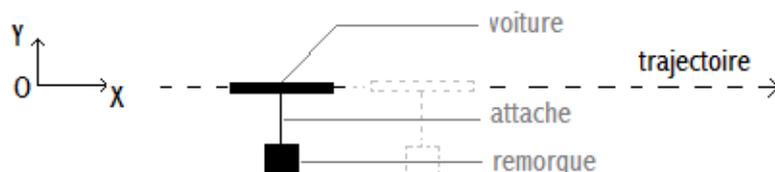


FIG. 1.1 – Le chemin non-admissible donné

Cette figure présente le chemin théorique que doit suivre la voiture et sa remorque, chemin qui n'est pas faisable a priori. Nous imposons que la voiture et sa remorque restent dans cette configuration (parallèles à (OY)) au cours du déplacement selon l'axe (OX) . Sachant que ce n'est pas réalisable, on souhaite trouver de manière formelle une trajectoire approchée que le mobile pourrait réellement suivre. Nous voulons aussi que cette trajectoire soit optimale, en un sens qui sera précisé ultérieurement.

L'étude théorique montre l'existence de trajectoires admissibles¹ répondant au problème donné. Nous allons nous efforcer de trouver la solution optimale afin de simuler et visualiser le déplacement de la voiture et de sa remorque.

Nous verrons dans les résultats (voir 4.3) l'allure de la trajectoire qui approche le mieux le chemin considéré dans la figure 1.1 et nous remarquerons que la voiture et sa remorque restent (approximativement) dans la configuration annoncée tout en se déplaçant selon (OX) .

¹ trajectoire admissible : solution du système cinématique associé au problème (trajectoire réalisable en pratique par le mobile).

En fait, ce problème se place dans un cadre plus général. Nous souhaitons trouver la trajectoire optimale reliant deux points d'un espace contenant des obstacles. Par exemple, des méthodes d'interpolation peuvent nous permettre de trouver un tel chemin. Ce problème n'étant que de la géométrie pure, nous ne le développerons pas ici. Le chemin obtenu n'est a priori pas admissible. C'est pour cela que nous cherchons à approximer ce chemin par des petits morceaux de courbes ϵ -admissibles². Ceci permet de construire un chemin admissible qui évite toujours les obstacles et relie les deux points de l'espace. Il est sous-entendu que nous essaierons d'analyser ce qui se passe quand ϵ tend vers 0 (i.e. quand l'approximation est de plus en plus précise).

Nous chercherons des solutions optimales (qui maximisent le déplacement du mobile en un temps et énergie fixés), au sens de l'entropie³. Cette partie mettra en jeu le principe du maximum de Pontrjagin⁴. Il existe des théorèmes (voir [12]) qui donnent un équivalent de l'entropie quand ϵ est au voisinage de 0. Malgré l'importance du calcul de l'entropie dans un problème de planification de trajectoire, nous ne parlerons pas de cette partie car ceci n'est pas l'objet de notre travail.

Dans un premier temps nous allons donner les définitions et les théorèmes nécessaires à la compréhension technique de ce document.

Par la suite, nous verrons que le mobile étudié peut être modélisé par un système cinématique de dimension 4 (voir partie 2, figure 2.1) et nous pouvons dès à présent préciser que nous nous placerons dans le cas du roulement sans glissement.

Ensuite nous verrons qu'il est plus avantageux de travailler dans d'autres coordonnées et cela nous amènera au système nilpotent (système possédant diverses propriétés utiles pour trouver les solutions formellement). Nous calculerons aussi l'invariant (*forme normale*) du problème qui lui est associé.

Puis, nous présenterons les calculs importants pour la simulation qui permettent de passer du système cinématique initial au système nilpotent. Une fois cela établi, nous verrons la résolution effective du problème grâce au principe du maximum de Pontrjagin.

Enfin, nous verrons la partie simulation Matlab en détail et nous exposerons les résultats obtenus.

1.2 Notions préliminaires

Nous allons maintenant définir les termes techniques que nous allons utiliser et récapituler les outils nécessaires à la bonne compréhension du document.

Notons tout d'abord que l'étude se fait a priori sur une sous-variété de dimension 4 ($\mathbb{R}^2 \times (S^1)^2$), mais comme nous ne travaillerons que localement, nous considérerons que nous sommes dans \mathbb{R}^4 . Nous omettrons donc les rappels mathématiques sur les variétés.

Néanmoins des bases de géométrie différentielle peuvent aider à la compréhension globale du rapport. Pour acquérir les définitions de bases (variétés, espace tangent, co-tangent, champs de vecteurs, courbes intégrales, géodésique etc.), on peut par exemple se référer à [9].

² courbe ϵ -admissible : courbe de longueur ϵ qui est admissible.

³ approximation de la trajectoire non-admissible par des morceaux de longueur ϵ admissibles.

⁴ Lev Pontrjagin (1908-1988) : mathématicien russe qui a beaucoup travaillé sur la théorie du contrôle optimal.

Définition 1.2.1 Une distribution est une famille de champs de vecteurs. Le rang d'une distribution est la dimension de l'espace vectoriel engendré par les champs de vecteurs en un point de l'espace ambiant.

Le rang peut donc varier suivant le point de l'espace que l'on considère.

Nous verrons dans la partie suivante que la distribution à considérer apparaît naturellement dans la modélisation cinématique du problème.

Dans notre cas, nous allons nous limiter à des **distributions de rang constant égal à 2** (on parle aussi de co-rang égal à 2 dans notre cas).

Définition 1.2.2 Une métrique sous-riemannienne sur \mathbb{R}^4 est la donnée d'un couple (Δ, g) où Δ est une distribution de \mathbb{R}^4 et g est une métrique euclidienne sur Δ .

Définition 1.2.3 Un problème de planification de trajectoire est la donnée d'un triplet $\Sigma = (\Delta, g, \Gamma)$, où (Δ, g) est une métrique sous-riemannienne et $\Gamma : [0, 1] \rightarrow \mathbb{R}^4$ est une courbe paramétrée lisse.

Remarque 1 notre problème est pertinent car $\Gamma(t)$ est transversal à $\Delta(\Gamma(t))$ pour tout t dans $[0, 1]$, i.e. $\dot{\Gamma}(t)$ n'est pas dans le plan engendré par la distribution en tout point de $\Delta(\Gamma(t))$. Si ce n'était pas le cas, le problème serait trivial aux points où $\Gamma(t)$ n'a pas la propriété de transversalité.

Définition 1.2.4 Une courbe admissible est une courbe paramétrée absolument continue $\sigma : [0, T_\sigma] \rightarrow \mathbb{R}^4$ telle que pour tout t , $\dot{\sigma}(t) \in \Delta(\sigma(t))$.

Définition 1.2.5 La longueur sous-riemannienne (d'une courbe σ admissible) associée à g est :

$$l(\sigma) = \int_0^{T_\sigma} \sqrt{g(\dot{\sigma}(s), \dot{\sigma}(s))} ds$$

Définition 1.2.6 La distance sous-riemannienne entre deux points p et q de \mathbb{R}^4 est :

$$d(p, q) = \inf \{ l(\sigma) \mid \sigma : [0, T_\sigma] \rightarrow \mathbb{R}^4 \text{ est admissible et } \sigma(0) = p, \sigma(T_\sigma) = q \}$$

Dans le but de définir l'entropie, nous introduisons le ϵ -tube T_ϵ défini ainsi :

$$T_\epsilon = \{ x \in \mathbb{R}^4 \text{ tq : } d(x, \epsilon) \leq \epsilon \}$$

Soit γ_ϵ une courbe paramétrée admissible (c'est-à-dire partout tangente à la distribution Δ). On supposera encore qu'elle est Lipschitzienne et paramétrée par la longueur d'arc. $\gamma_\epsilon : [0, t_{\gamma_\epsilon}] \rightarrow \mathbb{R}^4$ tel que $\gamma_\epsilon([0, t_{\gamma_\epsilon}]) \subset T_\epsilon$ et $\gamma_\epsilon(0) = \Gamma(0)$, $\gamma_\epsilon(t_{\gamma_\epsilon}) = \Gamma(1)$. Pour définir l'entropie, on rajoute la propriété suivante sur γ_ϵ : γ_ϵ est formée d'un nombre fini de morceaux de courbes de longueurs inférieures à ϵ , connectant les points de Γ .

Définition 1.2.7 L'entropie du système Σ est : $E(\epsilon) = \frac{1}{\epsilon} \inf l(\gamma_\epsilon)$ sur les γ_ϵ définis ci-dessus.

Cette définition de l'entropie du problème ne sera pas utilisée par la suite. Néanmoins, l'idée principale qui est d'approximer la trajectoire Γ par des petits bouts de longueurs ϵ est importante. Dès lors, il est compréhensible que la solution optimale du problème soit périodique, chose qui apparaîtra clairement lors de la simulation.

Définition 1.2.8 Le **crochet de Lie** de deux champs de vecteurs X et Y est le champ de vecteur Z défini par : $Z = [X, Y] = \sum_i Z^i \partial_i$ avec $Z^i = \sum_j X^j \partial_j Y^i - Y^j \partial_j X^i$

Remarque 2 On peut rappeler que champ de vecteur ou dérivation sont des notions équivalentes.

Théorème 1.2.1 Chow-Rashevsky. Dans M une variété connexe. Si la condition suivante (dite de Chow ou LARC) est vraie :

Les champs de vecteurs X_1, X_2, \dots, X_m et leurs crochets de Lie itérés $[X_i, X_j], [[X_i, X_j], X_k], \dots$ engendrent l'espace tangent $T_x M$ en tout point x de M .

Alors on peut toujours relier deux points de M par un chemin admissible.

Ce théorème fondamental nous indique l'existence de solutions. Nous verrons qu'il est trivialement vérifié dans notre cas (on calculera les crochets de Lie). Par contre il n'indique rien sur l'allure de la solution ni sur un quelconque caractère optimal. Pour en savoir plus, il faut maintenant introduire la notion de système nilpotent et puis le principe du maximum.

Dans notre cas, l'approximation nilpotente est faite le long de la courbe Γ , dans le tube T_ϵ . Le système nilpotent est en fait une approximation du système de départ pour lequel on garde les termes de plus bas degré. L'idée est de faire des développements limités (en tenant compte du poids des variables) et de ne garder que les premiers termes des développements. Pour formaliser cela, voici quelques définitions rigoureuses.

Notation 1 On note $L^s(p)$ le sous-espace de $T_p M$ engendré par les crochets de Lie de taille inférieure ou égale à s .

On sait qu'il existe un nombre appelé degré de non-holonomie et noté $r(p)$ vérifiant : $L^{r(p)}(p) = T_p M$. Dans notre cas, $r(p) = r$ puisque la distribution considérée est de rang constant égal à 2.

Définition 1.2.9 On appelle **poïds** de la j^{eme} coordonnée, le nombre w_j défini par : soit v_1, v_2, \dots, v_n une base adaptée au drapeau $L^1(p) \subset L^2(p) \subset \dots \subset L^r(p) = T_p M$. On défini $w_i = s$ si $v_i \in L^s(p)$ et $v_i \notin L^{s-1}(p)$

On a donc $w_1 = 1$ et $w_n = r$ (degré de non-holonomie).

Définition 1.2.10 **Système de coordonnées privilégiées.** C'est un système de coordonnées (z_1, \dots, z_n) définies le long de Γ tel que :

- (i) z_1, \dots, z_n sont linéairement adaptées au drapeau défini ci-dessus.
- (ii) Le poïds de z_j est exactement w_j .

Plaçons nous maintenant dans un tel système de coordonnées. Dès lors, le long de Γ , on peut écrire un champ de vecteur X de la façon suivante :

$$X(z) = \sum_{j=1}^n X^j(z) \partial z_j \simeq \sum_{j=1}^n \left(\sum_{\alpha} a_{\alpha,j} z^\alpha \right) \partial z_j$$

où il apparaît le développement de Taylor du champ de vecteur le long de Γ .

On associe un poïds égal à $-w_j$ aux termes ∂z_j . D'autre part, comme il est d'usage de faire pour les polynômes multivariés, on défini le degré de $a_{\alpha,j} z^\alpha$ comme :

$$\text{deg}(a_{\alpha,j} z^\alpha) = \alpha_1 w_1 + \dots + \alpha_n w_n = w(\alpha) \text{ avec } z^\alpha = z_1^{\alpha_1} \dots z_n^{\alpha_n}$$

Ayant défini ceci, on peut maintenant écrire $X(z)$ en regroupant les termes homogènes. Par exemple, les termes de degré -1 (le plus bas degré possible) seront donnés par des polynômes multivariés homogènes de degré $w_j - 1$. Et toutes les variables z_k de poïds supérieur $w_k \geq w_i$ seront exclues de ces polynômes. Ceci est très important comme nous le verrons par la suite. Les termes de degré -1 forment en fait un système triangulaire, ce qui permet de calculer les z_i en cascade, les uns après les autres. Ne considérer que les termes de degré -1 revient à considérer un système beaucoup plus commode à étudier, qui a l'avantage d'être une approximation du système initialement traité. Voici la définition rigoureuse du système nilpotent.

Définition 1.2.11 **Approximation nilpotente.** L'approximation nilpotente d'un système de champs de vecteurs X_1, \dots, X_m le long de Γ est l'ensemble des champs de vecteurs $(\hat{X}_1, \dots, \hat{X}_m)$ avec :

$$X_i = X_i^{(-1)} + X_i^{(0)} + X_i^{(1)} + X_i^{(2)} + \dots \text{ où } X_i^{(s)} \text{ est homogène de degré } s. \text{ Et } \hat{X}_i = X_i^{(-1)}.$$

L'approximation nilpotente est seulement constituée des termes homogènes de plus bas degré du développement limité autour de Γ du système différentiel considéré (une fois qu'on s'est placé dans un système de coordonnées privilégiées). Son avantage va apparaître clairement lors des calculs de la partie 4.1, notamment l'aspect triangulaire d'un tel système : on peut d'abord obtenir les variables d'ordre 0, puis celles d'ordre 1 grâce aux précédentes etc... D'autre part, ce système est tel qu'il existe une unique transformation sur les coordonnées pour passer de ce dernier au système de départ (on la trouvera explicitement). Et ainsi, toute résolution optimale sur le système nilpotent, nous fournira la solution optimale sur le système de départ. Ceci est une propriété remarquable qui découle essentiellement du fait que le système nilpotent est un développement limité particulier du système de départ.

Maintenant que nous avons précisé ce qu'est et à quoi va servir le système nilpotent, il faut parler de l'outil fondamental qui va permettre le travail d'optimisation. Cet outil est le principe du maximum. C'est une généralisation de la méthode des multiplicateurs de Lagrange. Ici, nous allons faire appel à une fonctionnelle appelée Hamiltonien. L'idée globale est de considérer un système légèrement différent du système initial, mais pour lequel nous serons capable de trouver la solution (qui sera en outre la solution optimale pour le système initial). Voici le principe du maximum de Pontrjagin dans le cas d'une métrique sous-riemannienne pour la minimisation de l'énergie :

Théorème 1.2.2 Principe du maximum de Pontrjagin. *Si une trajectoire admissible $(x(t), u(t))$ minimise l'énergie entre deux points q et q' pour le problème à temps fixé, alors il existe un relevé continu (x, φ) de x dans T^*M tel que φ ne s'annule pas, et il existe $\varphi^0 \leq 0$ tels que :*

- $\dot{x}(t) = \frac{\partial H}{\partial \varphi}$ et $\dot{\varphi}(t) = -\frac{\partial H}{\partial x}$
- $H(x(t), \varphi(t), u(t), \varphi^0) = \max\{H(x(t), \varphi(t), v(t), \varphi^0), v \in \mathbb{R}^4\}$

Une trajectoire satisfaisant les conclusions du théorème est dite extrémale. Il y a deux types d'extrémales :

- Les normales qui sont telles que $\varphi^0 < 0$
- Les anormales qui sont telles que $\varphi^0 = 0$

Nous calculerons les anormales et les normales associées à notre problème dans la suite du rapport.

Reste enfin à parler d'un outil qui nous servira pour trouver effectivement le système nilpotent. Il s'agit de la 2-forme fondamentale (*forme normale*).

Soit δ une 1-forme telle que $\Delta = \ker \delta$. On peut la choisir en imposant $\delta(\frac{d\Gamma}{dt}(t)) = 1$, pour tout t de $[0, 1]$. Cette 1-forme est définie modulo la multiplication par une fonction lisse qui prend la valeur 1 sur Γ (cela est prouvé simplement dans [7]).

On pose $\omega = d\delta$ et on considère $\tilde{\omega} = \omega|_{\Gamma}$.

Définition 1.2.12 *On appelle $\tilde{\omega}$ la 2-forme fondamentale associée à Σ .*

La forme $\tilde{\omega}$ définit aussi une matrice anti-symétrique (par rapport à la métrique g) $A(t)$ ainsi :

$$g(A(t)X, Y) = \tilde{\omega}(X, Y) \text{ où } X, Y \text{ sont dans } \Delta(\Gamma(t)).$$

Dès lors si on a une base orthonormée dans $\Delta(\Gamma(t))$, $A(t)$ sera un élément de $So(3)$ donné par sa matrice dans cette base. On note $\bar{A}(t)$ un tel élément.

L'intérêt d'une telle matrice est d'être intrinsèque au problème. C'est pour cela notamment que ses valeurs propres jouent un rôle important lors de l'étude d'un tel problème (par exemple pour l'expression de l'entropie, voir [12]).

Ayant tout l'outillage nécessaire, nous pouvons maintenant commencer l'étude du problème donné.

Chapitre 2

Etude du système cinématique

2.1 Mise en équation

Le problème de l'étude du système formé par la voiture et sa remorque commence évidemment par sa mise en équation. Précisons que nous considérons le roulement sans glissement. Voici la figure 2.1 qui représente la voiture et sa remorque (les coordonnées α et β représentent la position de la voiture dans le plan, θ et ϕ représentent respectivement l'inclinaison de la voiture et celle de la remorque par rapport à la voiture) :

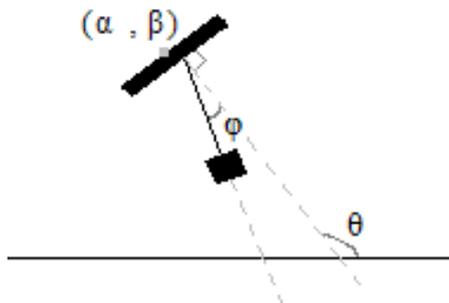


FIG. 2.1 – Configuration voiture et remorque.

Remarque 3 *Le chemin non-admissible considéré s'écrit : $\Gamma = \begin{pmatrix} \alpha = t \\ \beta = 0 \\ \theta = \pi/2 \\ \varphi = 0 \end{pmatrix}$*

Le système cinématique associé est :

$$\begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\theta} \\ \dot{\varphi} \end{pmatrix} = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \\ 0 \\ -\sin(\varphi) \end{pmatrix} * v(t) + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} * u(t) \quad (S)$$

v et u sont appelés les contrôles. Ce sont des fonctions dépendant du temps, qui indiquent à chaque instant comment le conducteur accélère (v) et tourne le volant (u).

Les trois premières lignes du système sont claires puisque la voiture avance dans la direction donnée par θ et que cet angle représente la façon dont est tourné le volant à l'instant t .

La quatrième équation sur φ est obtenue comme suit :

Effectuons un déplacement infinitésimal dans une direction donnée. Supposons que la voiture a avancé de $v dt$. Dans la figure suivante, on note (X, Y) les coordonnées de la remorque et φ l'angle que fait l'attache de la remorque avec l'horizontale (l'axe (OX)).

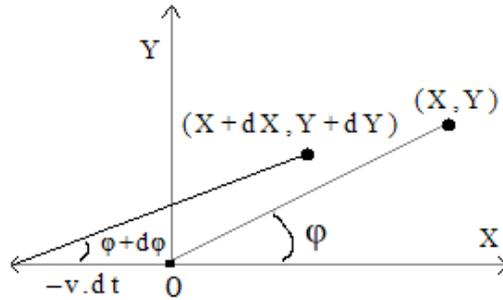


FIG. 2.2 – Déplacement infinitésimal de la remorque.

Dès lors, on a : $\begin{cases} X = L * \cos(\varphi) \\ Y = L * \sin(\varphi) \end{cases}$ avec L la longueur de l'attache remorque.

A l'instant $t + dt$, on a les équations similaires :

$$\begin{cases} X + dX = L * \cos(\varphi + d\varphi) - v * dt \\ Y + dY = L * \sin(\varphi + d\varphi) \end{cases}$$

D'où :

$$\begin{cases} dX = L * \cos(\varphi + d\varphi) - v * dt - L * \cos(\varphi) \\ dY = L * \sin(\varphi + d\varphi) - L * \sin(\varphi) \end{cases}$$

Et en simplifiant grâce aux formules classiques $\cos(a + b) = \cos(a)\cos(b) - \sin(a)\sin(b)$ et $\sin(a + b) = \sin(a)\cos(b) + \sin(b)\cos(a)$, on a :

$$\begin{cases} dX = -L * \sin(\varphi) * d\varphi - v * dt \\ dY = L * \cos(\varphi) * d\varphi \end{cases}$$

Maintenant, il suffit d'écrire que le vecteur vitesse de la remorque est colinéaire au vecteur (X, Y) (roulement sans glissement) et on obtient le résultat :

$$\begin{vmatrix} \dot{X} & X \\ \dot{Y} & Y \end{vmatrix} = 0 \iff (-L * \sin(\varphi) * \dot{\varphi} - v) * L * \sin(\varphi) = L * \cos(\varphi) * (L * \cos(\varphi) * \dot{\varphi})$$

D'où

$$\dot{\varphi} = \frac{-v * \sin(\varphi)}{L}$$

Le système cinématique étudié est donc correct (on prend une longueur d'attache remorque égale à l'unité). Il modélise parfaitement le roulement sans glissement de la voiture et de sa remorque.

On devra vérifier lors de la simulation que le vecteur déplacement de la voiture est bien toujours tangent à la trajectoire et que la remorque avance bien dans la direction de son attache. De plus, dans notre cas, on devra aussi vérifier que la voiture et sa remorque restent approximativement dans la configuration annoncée (c'est-à-dire quasiment verticales).

Nous allons maintenant voir plus précisément l'étude de ce système et notamment comment trouver les contrôles v et u optimaux permettant de résoudre le problème donné.

2.2 Modélisation géométrique du système

On peut clairement associer une distribution au système (S) (p. 9). Toute solution du système différentiel (S) sera une trajectoire associée à la distribution Δ , c'est-à-dire une courbe paramétrée dont la tangente à tout instant t sera dans le plan engendré par Δ . En fait, il y a une équivalence entre ces deux visions du problème. L'une est cinématique, l'autre est géométrique. C'est sur cette vision géométrique du problème que nous allons travailler, puisque de nombreux outils sont alors en notre possession (comme le Théorème de Chow ou encore le principe du maximum de Pontrjagin qui joue un rôle fondamental pour trouver des solutions optimales).

Dans notre cas, la distribution considérée est définie par les deux vecteurs

$$X = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \\ 0 \\ -\sin(\varphi) \end{pmatrix} \text{ et } Y = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Le crochet de Lie $Z = [X, Y]$ est le vecteur

$$Z = \begin{pmatrix} \sin(\theta) \\ -\cos(\theta) \\ 0 \\ \cos(\varphi) \end{pmatrix}$$

L'itéré d'ordre 2 $W = [X, [X, Y]]$ est le vecteur

$$W = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

A tout instant, l'espace engendré par ces quatre vecteurs est de dimension 4. Ceci prouve que la condition de Chow est vérifiée, comme nous l'avions déjà annoncé.

Mais au lieu de continuer à travailler sur ce système, on peut dès maintenant introduire un autre système, l'idée étant de redresser $\Gamma = \begin{pmatrix} \alpha = t \\ \beta = 0 \\ \theta = \pi/2 \\ \varphi = 0 \end{pmatrix}$ en se ramenant à

$$\tilde{\Gamma} = \begin{pmatrix} \tilde{\alpha} = 0 \\ \tilde{\beta} = 0 \\ \tilde{\theta} = 0 \\ \tilde{\varphi} = t \end{pmatrix} \text{ par une transformation linéaire.}$$

Ceci se fait aisément par la transformation linéaire suivante :

$$\begin{pmatrix} \tilde{\alpha} \\ \tilde{\beta} \\ \tilde{\theta} \\ \tilde{\varphi} \end{pmatrix} = \begin{pmatrix} \beta \\ \theta - \pi/2 \\ \varphi - \theta + \pi/2 \\ \alpha - (\varphi - \theta) - \pi/2 \end{pmatrix}$$

On a ainsi le système suivant :

$$\begin{pmatrix} \dot{\tilde{\alpha}} \\ \dot{\tilde{\beta}} \\ \dot{\tilde{\theta}} \\ \dot{\tilde{\varphi}} \end{pmatrix} = \begin{pmatrix} \cos(\tilde{\beta}) \\ 0 \\ -\sin(\tilde{\beta} + \tilde{\theta}) \\ -\sin(\tilde{\beta}) + \sin(\tilde{\beta} + \tilde{\theta}) \end{pmatrix} * v(t) + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} * u(t) \quad (\tilde{S})$$

Dans toute la suite, nous travaillerons donc essentiellement entre ce système et le système nilpotent. Car dans ce nouveau système, nous sommes assez proches des coordonnées privilégiées (qui présentent de nombreux avantages).

Chapitre 3

Résolution du problème optimal

3.1 Le système nilpotent

Ayant défini ce qu'était l'approximation nilpotente dans les préliminaires, il faut maintenant voir comment la trouver en pratique sachant qu'au départ, nous avons seulement le système (S) (p. 9).

Il existe de puissants théorèmes pour cela. On sait qu'il existe un système de coordonnées appelé *système de coordonnées normales* vérifiant diverses propriétés qui permet l'obtention du système nilpotent.

Théorème 3.1.1 Coordonnées normales. *Pour notre problème $(\Delta, g, \tilde{\Gamma})$, il existe un système de coordonnées $(\tilde{x}, \tilde{y}, \tilde{z}, \tilde{w})$ dans \mathbb{R}^4 défini sur un voisinage de $\tilde{\Gamma}([0, 1])$ tel que :*

- (i) $\tilde{\Gamma}(t) = (0, 0, 0, t)$, $\Delta(\tilde{\Gamma}(t)) = \ker d\omega$ ($d\omega$ forme normale) et $g_{|\tilde{\Gamma}(t)} = d\tilde{x}^2 + d\tilde{y}^2 + d\tilde{z}^2 + d\tilde{w}^2$
- (ii) Les géodésiques satisfaisant les conditions de transversalité du principe du maximum de Pontrjagin par rapport à $\tilde{\Gamma}$ sont les lignes droites passant par $\tilde{\Gamma}$, contenues dans les plans $S_{\tilde{w}_0} = \{(\tilde{x}, \tilde{y}, \tilde{z}, \tilde{w}) \mid \tilde{w} = \tilde{w}_0\}$
- (iii) Pour ϵ assez petit, le ϵ -tube est $T_\epsilon = \{q \mid d(q, \tilde{\Gamma}) = \epsilon\} \simeq \{ \|(\tilde{x}, \tilde{y}, \tilde{z}, \tilde{w})\|_2 = \epsilon \}$

Une fois les coordonnées normales $(\tilde{x}, \tilde{y}, \tilde{z}, \tilde{w})$ fixées, citons le théorème qui va nous permettre d'expliciter le système nilpotent.

Théorème 3.1.2 Forme normale. *Il existe un unique repère (F, G) orthonormé pour la métrique g vérifiant :*

- (i) Q est symétrique, $Q(0, 0, \tilde{z}, \tilde{w}) = id$, $Q(\tilde{x}, \tilde{y}, \tilde{z}, \tilde{w}) \cdot (\tilde{x}, \tilde{y}) = (\tilde{x}, \tilde{y})$
- (ii) $L(\tilde{x}, \tilde{y}, \tilde{z}, \tilde{w}) \cdot (\tilde{x}, \tilde{y}) = (0, 0)$

$$\text{avec } F = \begin{pmatrix} q_{1,1} \\ q_{2,1} \\ l_{1,1} \\ l_{2,1} \end{pmatrix}, G = \begin{pmatrix} q_{1,2} \\ q_{2,2} \\ l_{1,2} \\ l_{2,2} \end{pmatrix}, Q = \begin{pmatrix} q_{1,1} & q_{1,2} \\ q_{2,1} & q_{2,2} \end{pmatrix} \text{ et } L = \begin{pmatrix} l_{1,1} & l_{1,2} \\ l_{2,1} & l_{2,2} \end{pmatrix}$$

Et le système nilpotent est donné par : $(\dot{\tilde{x}}, \dot{\tilde{y}}, \dot{\tilde{z}}, \dot{\tilde{w}}) = F.u + G.v$

En fait, Q et L peuvent s'exprimer en fonction des valeurs propres de $\bar{A}(t)$ et grâce aussi aux conditions imposées sur les coordonnées normales.

Dans notre cas, en dimension 4, on peut trouver les premiers termes des développements limités de Q et L. On obtient (d'après [7], [12]) :

$$Q = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \text{ et } L = \begin{pmatrix} \tilde{y}/2 & -\tilde{x}/2 \\ \gamma(\tilde{w})\tilde{y}\tilde{x}/2 & -\gamma(\tilde{w})\tilde{x}^2/2 \end{pmatrix}$$

où $\gamma(\tilde{w})$ est un invariant du problème (donné par la forme normale qui est le vrai invariant fondamental du problème).

Le système nilpotent est le suivant :

$$\begin{pmatrix} \dot{\tilde{x}} \\ \dot{\tilde{y}} \\ \dot{\tilde{z}} \\ \dot{\tilde{w}} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \frac{\tilde{y}}{2} \\ \gamma(\tilde{w})\frac{\tilde{x}\tilde{y}}{2} \end{pmatrix} * v(t) + \begin{pmatrix} 0 \\ 1 \\ -\frac{\tilde{x}}{2} \\ -\gamma(\tilde{w})\frac{\tilde{x}^2}{2} \end{pmatrix} * u(t)$$

Cet invariant se calcule en utilisant le fait qu'on est dans les coordonnées normales. Ici, nous allons mener le calcul de deux façons différentes. L'idée est d'utiliser le fait que cet invariant ne dépend pas du système de coordonnées choisi. Ce calcul est fait dans la section suivante.

De plus, en posant $\hat{w} = \int \frac{ds}{\gamma(s)}$, on se ramène au système précédent en prenant $\gamma(\tilde{w}) = 1$

Le système nilpotent auquel nous arrivons est donc le système suivant (c'est celui-ci que nous prendrons dans la suite) :

$$\begin{pmatrix} \dot{\tilde{x}} \\ \dot{\tilde{y}} \\ \dot{\tilde{z}} \\ \dot{\tilde{w}} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \frac{\tilde{y}}{2} \\ \frac{\tilde{x}\tilde{y}}{2} \end{pmatrix} * v(t) + \begin{pmatrix} 0 \\ 1 \\ -\frac{\tilde{x}}{2} \\ -\frac{\tilde{x}^2}{2} \end{pmatrix} * u(t) \quad (\tilde{N})$$

Vers la fin du problème et pour la résolution effective (surtout pour simplifier les calculs et se ramener sur les équations exactes des elastica d'Euler, nous serons amenés à considérer le système suivant (qui est très proche du système (\tilde{N}) , à quelques changements de rôle près des variables x, y, z et w) :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \frac{y}{2} \\ \frac{yx}{2} \end{pmatrix} * u(t) + \begin{pmatrix} 0 \\ 1 \\ -\frac{x}{2} \\ \frac{yx}{2} \end{pmatrix} * v(t) \quad (N)$$

Nous verrons dans la suite que pour passer de ce système nilpotent au système cinématique initial, il faut travailler sur des systèmes intermédiaires et faire des développements limités. On sait qu'il existe une transformation unique (à difféomorphismes près) passant des systèmes (S) (p. 9) à (N) (p. 14) et il faudra la trouver effectivement, ce que nous ferons dans la suite.

Pour vérifier que (\tilde{N}) est bien le système nilpotent, on peut appliquer le principe du maximum de Pontrjagin et calculer les anormales pour vérifier que cela vérifie les hypothèses annoncées.

Des calculs classiques dans ce cas annoncent que les anormales sont telles que $u = 1$ et que v doit vérifier :

$$\det | F, G, [F, G], v[F, [F, G]] + [G, [F, G]] | = 0$$

avec

$$F = \begin{pmatrix} 1 \\ 0 \\ \frac{\tilde{y}}{2} \\ \frac{\tilde{x}\tilde{y}}{2} \end{pmatrix}, G = \begin{pmatrix} 0 \\ 1 \\ -\frac{\tilde{x}}{2} \\ -\frac{\tilde{x}^2}{2} \end{pmatrix}, [F, G] = \begin{pmatrix} 0 \\ 0 \\ -1 \\ -\frac{3\tilde{x}}{2} \end{pmatrix}$$

$$\text{et } v[F, [F, G]] + [G, [F, G]] = v \begin{pmatrix} 0 \\ 0 \\ 0 \\ -\frac{3}{2} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Ceci donne $v = 0$ et $u = 1$. Dès lors le système différentiel vérifié par les anormales est :

$$\begin{pmatrix} \dot{\tilde{x}} \\ \dot{\tilde{y}} \\ \dot{\tilde{z}} \\ \dot{\tilde{w}} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ -\frac{\tilde{x}}{2} \\ -\frac{\tilde{x}^2}{2} \end{pmatrix}$$

C'est donc :

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ \tilde{w} \end{pmatrix} = \begin{pmatrix} 0 \\ t \\ 0 \\ \tilde{w}_0 \end{pmatrix} \text{ car à } t=0 \text{ on est en } \begin{pmatrix} 0 \\ 0 \\ 0 \\ \tilde{w}_0 \end{pmatrix}$$

Ce qui vérifie le point (ii) du théorème, à savoir que l'on obtient bien des droites dans des plans tels que $\tilde{w} = \tilde{w}_0$. Les autres points sont aussi vérifiés : on voit aisément qu'autour de $\tilde{\Gamma} = (0, 0, 0, t)$ les crochets de Lie forment une base orthonormée.

Des approximations de F, G et [F,G] autour de $\tilde{\Gamma}$ donnent :

$$F = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, G = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, -[F, G] = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \text{ et } \dot{\tilde{\Gamma}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Ceci servira notamment pour simplifier les calculs de $\bar{A}(t)$ dans la section suivante.

3.2 Calcul direct de $\gamma(\tilde{w})$

Pour calculer effectivement la forme normale, on va mener le même calcul en se plaçant dans deux systèmes de coordonnées différents. Comme le résultat doit être identique dans les deux cas, nous obtiendrons la forme normale en identifiant.

Rappelons que l'on travaille au voisinage de $\tilde{\Gamma} : \begin{pmatrix} 0 \\ 0 \\ 0 \\ t \end{pmatrix}$

Définissons maintenant la 1-forme Υ telle que (définie car $\tilde{\Gamma}$ partout transversale à la distribution) :

$$\begin{cases} \Upsilon = 1 \text{ sur } \tilde{\Gamma} \\ \Upsilon = 0 \text{ sur } \langle F, G, [F, G] \rangle \end{cases}$$

avec

$$F = \begin{pmatrix} 1 \\ 0 \\ \frac{\tilde{y}}{2} \\ \gamma(\tilde{w})\frac{\tilde{x}\tilde{y}}{2} \end{pmatrix}, G = \begin{pmatrix} 0 \\ 1 \\ -\frac{\tilde{x}}{2} \\ -\gamma(\tilde{w})\frac{\tilde{x}^2}{2} \end{pmatrix} \text{ et } -[F, G] = \begin{pmatrix} 0 \\ 0 \\ 1 \\ \gamma(\tilde{w})\frac{3}{2}\tilde{x} \end{pmatrix}$$

Cette 1-forme peut s'écrire : $\Upsilon = a d\tilde{x} + b d\tilde{y} + c d\tilde{z} + d\tilde{w}$

En utilisant nos conditions, on obtient les équations suivantes sur a, b et c.

$$\begin{cases} a + c\frac{\tilde{y}}{2} + \gamma(\tilde{w})\frac{\tilde{x}\tilde{y}}{2} = 0 \\ b - c\frac{\tilde{x}}{2} - \gamma(\tilde{w})\frac{\tilde{x}^2}{2} = 0 \\ c + \gamma(\tilde{w})\frac{3}{2}\tilde{x} = 0 \end{cases}$$

Ceci donne :

$$c = -\gamma(\tilde{w})\frac{3}{2}\tilde{x}, a = \frac{1}{4}\tilde{x}\tilde{y}\gamma(\tilde{w}) \text{ et } b = -\frac{1}{4}\tilde{x}^2\gamma(\tilde{w}).$$

Donc :

$$\Upsilon = \frac{1}{4}\tilde{x}\tilde{y}\gamma(\tilde{w})d\tilde{x} - \frac{1}{4}\tilde{x}^2\gamma(\tilde{w})d\tilde{y} - \gamma(\tilde{w})\frac{3}{2}\tilde{x}d\tilde{z} + d\tilde{w}$$

Considérons maintenant $d\Upsilon$ (en rappelant qu'on travaille localement autour de Γ et que par conséquent on peut faire des simplifications dès qu'un terme contient un facteur \tilde{x} , \tilde{y} ou \tilde{z}).

Il reste :

$$d\Upsilon = -\frac{3}{2}\gamma(\tilde{w})d\tilde{x} \wedge d\tilde{z}$$

Les vecteurs $F, G, -[F, G]$, formant le long de $\tilde{\Gamma}$ la base canonique, il vient que l'on peut représenter la 2-forme par une matrice \bar{A} anti-symétrique.

$$\langle \bar{A}X, Y \rangle = d\Upsilon(X, Y) \text{ d'où } \bar{A} = \begin{pmatrix} 0 & 0 & \frac{3}{2}\gamma(\tilde{w}) \\ 0 & 0 & 0 \\ -\frac{3}{2}\gamma(\tilde{w}) & 0 & 0 \end{pmatrix}$$

On identifiera cette matrice avec la matrice donnée après les calculs ci-dessous.

Faisons maintenant les calculs à partir du système (\tilde{S}) (p. 12). On sait que la forme normale est un invariant du problème et que par conséquent la matrice anti-symétrique associée doit avoir les mêmes valeurs propres.

En notant X_1 et X_2 les 2 vecteurs du système (\tilde{S}), on peut calculer le crochet de Lie :

$$X_3 = [X_1, X_2] = \begin{pmatrix} \sin(\tilde{\beta}) \\ 0 \\ \cos(\tilde{\beta} + \tilde{\theta}) \\ \cos(\tilde{\beta}) - \cos(\tilde{\beta} + \tilde{\theta}) \end{pmatrix}$$

Définissons la 1-forme Ω telle que (définie car $\tilde{\Gamma}$ partout transversale à la distribution) :

$$\begin{cases} \Omega = 1 \text{ sur } \tilde{\Gamma} \\ \Omega = 0 \text{ sur } \langle X_1, X_2, X_3 \rangle \end{cases}$$

Cette 1-forme peut s'écrire :

$$\Omega = a d\tilde{\alpha} + b d\tilde{\beta} + c d\tilde{\theta} + d\tilde{\varphi}$$

En écrivant nos hypothèses, il vient $b = 0$ et le système 2x2 suivant :

$$\begin{cases} a \sin(\tilde{\beta}) + c \cos(\tilde{\theta} + \tilde{\beta}) = -(\cos(\tilde{\beta}) - \cos(\tilde{\beta} + \tilde{\theta})) \\ a \cos(\tilde{\beta}) - c \sin(\tilde{\theta} + \tilde{\beta}) = \sin(\tilde{\beta}) - \sin(\tilde{\beta} + \tilde{\theta}) \end{cases}$$

Les solutions de ce système de Cramer, sont :

$$a = \tan(\tilde{\theta}) \text{ et } c = 1 - \frac{1}{\cos(\tilde{\theta})}$$

Donc la 2-forme fondamentale (forme normale), invariant du problème peut s'écrire :

$$d\Omega = (1 + \tan^2(\tilde{\theta})) d\tilde{\alpha} \wedge d\tilde{\theta}$$

Le long de $\tilde{\Gamma}$, cela donne :

$$d\Omega = d\tilde{\alpha} \wedge d\tilde{\theta}$$

Les vecteurs X_1, X_2, X_3 , formant le long de $\tilde{\Gamma}$ une base orthonormée (la base canonique), il vient que l'on peut représenter notre 2-forme par une matrice \bar{B} anti-symétrique.

$$\langle \bar{B}X, Y \rangle = d\Omega(X, Y) \text{ d'où } \bar{B} = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Il ne reste plus qu'à identifier cette matrice avec \bar{A} pour conclure.

Nous obtenons $\gamma(\tilde{w}) = -\frac{2}{3}$ qui est donc une constante dans notre cas. La transformation utilisée pour reparamétriser la variable \tilde{w} n'est qu'une remise à l'échelle ici. Dans la suite nous travaillerons directement dans le système (\tilde{N}) (p. 14).

3.3 Passage du système nilpotent au système cinématique

Commençons d'abord par faire un développement limité du système (\tilde{S}) (p. 12) :

Le système devient :

$$\begin{pmatrix} \dot{\tilde{\alpha}} \\ \dot{\tilde{\beta}} \\ \dot{\tilde{\theta}} \\ \dot{\tilde{\varphi}} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ -\tilde{\beta} \\ \tilde{\theta} \end{pmatrix} * v(t) + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} * u(t)$$

Sur les deux premières composantes, la transformation pour passer des systèmes (\tilde{S}) (p. 12) à (\tilde{N}) (p. 14) est évidente.

Voyons comment procéder sur la troisième composante. On écrit d'abord que \tilde{z} est a priori homogène d'ordre 2 (précisons que $\tilde{\theta}$ est d'ordre 2 et que $\tilde{\varphi}$ est d'ordre 3, ce qui est immédiat à la vue du système ci-dessus, $\tilde{\alpha}$ et $\tilde{\beta}$ étant d'ordre 1).

Ecrivons donc (on omet volontairement $\tilde{\alpha}^2$ et $\tilde{\beta}^2$ qui ne jouent pas de rôle ici) :

$$\tilde{z} = a\tilde{\theta} + b\tilde{\alpha}\tilde{\beta}$$

Ce qui donne, en dérivant :

$$\dot{\tilde{z}} = a\dot{\tilde{\theta}} + b\dot{\tilde{\alpha}}\tilde{\beta} + b\tilde{\alpha}\dot{\tilde{\beta}} = -a\tilde{y}v + b\tilde{y}v + b\tilde{x}u$$

D'où en identifiant,

$$b - a = \frac{1}{2} \text{ et } b = -\frac{1}{2}$$

Ce qui donne,

$$\tilde{z} = -\tilde{\theta} - \frac{1}{2}\tilde{\alpha}\tilde{\beta}$$

Pour la dernière composante, le calcul n'est pas plus compliqué, juste plus long. Intuitivement, on peut conjecturer que \tilde{w} sera constitué de termes homogènes d'ordre 3.

On peut donc écrire a priori (sans rien négliger cette fois) :

$$\tilde{w} = \lambda\tilde{\varphi} + a\tilde{\alpha}^3 + b\tilde{\beta}^3 + c\tilde{\alpha}^2\tilde{\beta} + d\tilde{\alpha}\tilde{\beta}^2 + e\tilde{\alpha}\tilde{\theta} + f\tilde{\beta}\tilde{\theta}$$

En dérivant, puis en remplaçant, on arrive facilement à :

$$\dot{\tilde{w}} = \left(3a\tilde{x}^2 + (d-f)\tilde{y}^2 + \left(\frac{1}{2}\lambda + 2c - \frac{3}{2}e\right)\tilde{x}\tilde{y} + (-\lambda - e)\tilde{z} \right) u + \left(c\tilde{x}^2 + 3b\tilde{y}^2 + \left(2d - \frac{1}{2}f\right)\tilde{x}\tilde{y} - f\tilde{z} \right) v$$

En identifiant avec le système (\tilde{N}) (p. 14), on obtient facilement :

$$a = b = d = f = 0 \text{ et } c = -\frac{1}{2}, e = -\frac{3}{2} \text{ et } \lambda = \frac{3}{2}.$$

Pour se ramener au système nilpotent (\tilde{N}), on pose :

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ \tilde{w} \end{pmatrix} = \begin{pmatrix} \tilde{\alpha} \\ \tilde{\beta} \\ -\tilde{\theta} - \frac{1}{2}\tilde{\alpha}\tilde{\beta} \\ \frac{3}{2}\tilde{\varphi} - \frac{3}{2}\tilde{\alpha}\tilde{\theta} - \frac{1}{2}\tilde{\alpha}^2\tilde{\beta} \end{pmatrix}$$

Ensuite on passe aisément du système (\tilde{N}) (p. 14) au système (N) (p. 14) par la transformation suivante :

$$\tilde{x} = y, \tilde{y} = x, \tilde{z} = -z, \tilde{w} = -w, u \leftrightarrow v$$

En résumé, on passe des systèmes (S) (p. 9) à (\tilde{S}) (p. 12) à (\tilde{N}) (p. 14) à (N) (p. 14) par les transformations successives détaillées précédemment. Attention toutefois : nous n'arrivons pas exactement au système (S) mais à une approximation de ce système puisque nous partons du système nilpotent.

Pour faire la transformation globale, on compose les différentes transformations trouvées et on obtient :

$$\begin{pmatrix} \alpha \\ \beta \\ \theta \\ \varphi \end{pmatrix} = \begin{pmatrix} z - \frac{1}{2}xy - \frac{2}{3}w + yz - \frac{1}{6}y^2x \\ y \\ x + \frac{\pi}{2} \\ x + z - \frac{1}{2}xy \end{pmatrix}$$

et on échange les rôles de u et v .

Il est facile de vérifier qu'avec un tel changement de coordonnées, on passe du système nilpotent (N) (p. 14) à une approximation du système de départ (S) (p. 9). Une vérification sera menée dans la partie 4.1. Nous verrons dans cette partie que nous pouvons aller plus loin dans la précision et qu'au lieu de nous limiter à l'approximation à l'ordre -1, nous pouvons inclure les restes, pour avoir à la place du système nilpotent, un système qui contient le système nilpotent plus les restes. Ceci permettra de contrôler l'imprécision de nos approximations, ce qui joue un rôle important en simulation.

On remarque qu'un simple échange sur les contrôles u et v a été nécessaire, ce qui permet de résoudre le problème avec le système nilpotent et d'avoir directement comme solution les contrôles optimaux pour le système de départ. L'intérêt de connaître cette transformation est de pouvoir obtenir α , β , θ et φ grâce aux calculs sur le système nilpotent via lequel nous obtiendrons les contrôles u et v (toujours dans l'objectif d'une simulation).

3.4 Etude du système nilpotent et résolution du problème

Maintenant, nous avons le système nilpotent (N) (p. 14) à notre disposition.

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \frac{y}{2} \\ \frac{yx}{2} \end{pmatrix} * u(t) + \begin{pmatrix} 0 \\ 1 \\ -\frac{x}{2} \\ \frac{yx}{2} \end{pmatrix} * v(t)$$

Qu'on peut aussi écrire : $\dot{\xi} = F(\xi)u_1 + G(\xi).u_2$ avec $\xi = (x, y, z, w)$

Nous voulons aller de $(0, 0, 0, 0)$ à $(0, 0, 0, \hat{w})$ en un temps fixé ϵ et en minimisant \hat{w} . Pour ceci, nous allons appliquer le Principe du maximum de Pontrjagin en un temps fixé. La transversalité, plus le fait que l'hamiltonien ne dépend pas de w permettent d'écrire l'hamiltonien ainsi :

$$H = \frac{s_0}{2}y(yu - xv) + pu + qv + \frac{r}{2}(yu - xv)$$

Ici, s_0 est la variable adjointe de l'hamiltonien. $s_0 \leq 0$. On pose $\Psi(t) = (p, q, r, s_0)$, et on voit alors que $H = \Psi.F.u_1 + \Psi.G.u_2$

Nous avons déjà vu que les anormales ne font pas le travail que l'on veut, puisque ce sont des lignes droites dans des plans à $w = w_0 = cste$, alors qu'on souhaite se déplacer selon w . Ce sont donc les extrémales normales qui vont nous intéresser.

L'hamiltonien n'étant pas nul, en reparamétrant les normales par la longueur d'arc, on obtient :

$$H = \sqrt{(\Psi.F)^2 + (\Psi.G)^2}$$

et l'hamiltonien vaut 1 le long des trajectoires, si bien que $p(0)^2 + q(0)^2 = 1$.

Comme d'habitude dans cette situation, on obtient les équations suivantes :

$$\frac{d\Psi.F}{dt} = \Psi[F, G]\Psi.G$$

$$\frac{d\Psi.G}{dt} = -\Psi[F, G]\Psi.F$$

Posons $h = \Psi[F, G]$, $u = \Psi.F = \cos(\varphi)$ et $v = \Psi.G = \sin(\varphi)$ (puisque $u^2 + v^2 = 1$).

On obtient alors :

$$\dot{\varphi} = -h$$

Donc,

$$\dot{h} = -\Psi.[F, [F, G]]u - \Psi.[G, [F, G]]v = 0 + \frac{3}{2}s_0v = \frac{3}{2}s_0 \sin(\varphi)$$

Rappelons que :

$$F = \begin{pmatrix} 1 \\ 0 \\ \frac{y}{2} \\ \frac{y^2}{2} \end{pmatrix}, G = \begin{pmatrix} 0 \\ 1 \\ -\frac{x}{2} \\ -\frac{xy}{2} \end{pmatrix},$$

$$[F, G] = \begin{pmatrix} 0 \\ 0 \\ -1 \\ -\frac{3y}{2} \end{pmatrix}, [F, [F, G]] = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \text{ et } [G, [F, G]] = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -\frac{3}{2} \end{pmatrix}$$

Donc,

$$\ddot{\varphi} = -\dot{h} = -\frac{3}{2}s_0 \sin \varphi$$

Comme $s_0 \leq 0$, on change φ en $\varphi + \pi$ et on obtient :

$$\ddot{\varphi} = -\frac{3}{2}s \sin(\varphi), s \geq 0$$

Dès lors on est presque dans le cas des elastica d'Euler : la courbe a une projection dans le plan (x, y) telle que :

$$\begin{aligned}\dot{x} &= u = -\cos(\varphi) \\ \dot{y} &= v = -\sin(\varphi)\end{aligned}$$

Ainsi, suivant [1], nous avons un elastica. Comme il y a plusieurs types d'elastica, nous l'identifions en regardant les propriétés particulières qu'il vérifie. La courbe en projection sur (x, y) doit être fermée (reliant les extrémités en 0), lisse (du fait des extrémales normales) et la courbe doit délimiter un domaine dont l'aire est nulle (la variable z n'étant autre que l'aire puisque $\dot{z} = \frac{1}{2}(y\dot{x} - \dot{y}x)$). Dès lors, il n'y a qu'une seule solution possible, c'est l'elastica périodique inflexionnel. De plus, l'inflexion doit être à l'origine, ce qui implique que : $0 = \dot{\varphi}(0) = -h(0) = r + \frac{3}{2}sy(0)$. D'où $r = 0$.

La fonction $\frac{1}{2}\dot{\varphi}^2 - \frac{3}{2}s\cos(\varphi)$ est une intégrale du mouvement et l'équation sur $\ddot{\varphi}$ est réécrite en $\frac{1}{2}\dot{\varphi}^2 = -\frac{3s}{2}(\cos(\varphi_0) - \cos(\varphi))$

Par conséquent, si $k = \sin(\frac{\varphi_0}{2})$, et $K(k)$ est le quart de la période des fonctions elliptiques de Jacobi avec pour argument ν et pour module k , alors la période en $\nu = t\sqrt{\frac{3s}{2}}$ est $4K$ et celle en t est $4K\sqrt{\frac{2}{3s}} = \epsilon$. Le cas de cet elastica correspond à une valeur K telle que : $2Eam(K) = K \approx 2.32073$ obtenue pour $\varphi_0 = 130,692^\circ$ (d'après Mathematica et [8])

Eam est la fonction *Elliptique amplitude* définie ci-dessous (on en profite aussi pour introduire les fonctions sn et dn) :

Définition 3.4.1 (*Fonctions elliptiques de Jacobi*). En posant $u = \int_0^\phi \frac{d\theta}{(1-k^2\sin^2(\theta))}$, $Eam(u) = \phi$, $sn(u) = \sin(\phi)$ et $dn(u) = (1 - k^2\sin^2(\phi))^{\frac{1}{2}}$

Au final, on obtient (d'après [1] (p.403)) :

$$\begin{aligned}x(t) &= -\frac{\epsilon}{4K} \left(-\frac{4Kt}{\epsilon} + 2(Eam(\frac{4Kt}{\epsilon} + K) - Eam(K)) \right) \\ y(t) &= k\frac{\epsilon}{2K} cn(\frac{4Kt}{\epsilon} + K)\end{aligned}$$

Et on obtient les contrôles effectivement :

$$\begin{aligned}u(t) &= 1 - 2dn(K(1 + \frac{4t}{\epsilon}))^2 \\ v(t) &= -2dn(K(1 + \frac{4t}{\epsilon}))sn(K(1 + \frac{4t}{\epsilon}))sin(\frac{\varphi_0}{2})\end{aligned}$$

où $K = 2.32073$ et $\varphi_0 = 130,692^\circ$ et dn , sn étant les fonctions elliptiques de Jacobi de module $sin(\frac{\varphi_0}{2})$.

C'est grâce à ces deux fonctions de contrôles et la transformation page 19 que l'on va pouvoir effectuer la simulation sous MatLab.

Chapitre 4

Simulation en MatLab

4.1 Problème de représentation

Le problème auquel nous sommes confrontés est le problème des échelles des différentes variables. En effet, x et y sont d'ordre 1, z est d'ordre 2 et w est d'ordre 3. Pour faire face à ces différences d'échelle qui posent problème lors d'une tentative de simulation informatique, on renormalise le système en posant :

$$\hat{x}(t) = \frac{x(\epsilon t)}{\epsilon}, \hat{y}(t) = \frac{y(\epsilon t)}{\epsilon}, \hat{z}(t) = \frac{z(\epsilon t)}{\epsilon^2}, \hat{w}(t) = \frac{w(\epsilon t)}{\epsilon^3}, \hat{u}(t) = u(\epsilon t) \text{ et } \hat{v}(t) = v(\epsilon t)$$

Ceci nous permet d'avoir le système renormalisé suivant (où les variables sont maintenant de l'ordre de l'unité, sans dépendre de la grandeur ϵ) :

$$\begin{pmatrix} \dot{\hat{x}} \\ \dot{\hat{y}} \\ \dot{\hat{z}} \\ \dot{\hat{w}} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \frac{1}{2}\hat{y} \\ \frac{1}{2}\hat{y}^2 \end{pmatrix} * \hat{u}(t) + \begin{pmatrix} 0 \\ 1 \\ -\frac{1}{2}\hat{x} \\ -\frac{1}{2}\hat{x}\hat{y} \end{pmatrix} * \hat{v}(t) \quad (\hat{N})$$

On peut même aller plus loin en écrivant carrément le système exact correspondant à (S) (p. 9) dans les coordonnées normales renormalisées (en prenant en compte les restes des développements limités).

Les calculs qui vont suivre montrent par la même occasion que la transformation globale de la partie 3.3 est exacte.

Rappelons notre système de départ :

$$\begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\theta} \\ \dot{\varphi} \end{pmatrix} = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \\ 0 \\ -\sin(\varphi) \end{pmatrix} * v(t) + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} * u(t)$$

et notre changement de coordonnées :

$$\begin{pmatrix} \alpha \\ \beta \\ \theta \\ \varphi \end{pmatrix} = \begin{pmatrix} z - \frac{1}{2}xy - \frac{2}{3}w + yz - \frac{1}{6}y^2x \\ y \\ x + \frac{\pi}{2} \\ x + z - \frac{1}{2}xy \end{pmatrix} \quad u \leftrightarrow v$$

Exprimons donc le système (\hat{N}) (p.22) sans négliger de termes :

- On a :

$$\dot{\theta}(t) = u(t) = \dot{x}(t) \Rightarrow \dot{\hat{x}}(t) = \dot{x}(\epsilon t) = u(\epsilon t) = \hat{u}(t)$$

- On a :

$$\dot{\beta}(t) = \sin(\theta(t))v(t) = \dot{y}(t) \Rightarrow \dot{\beta}(t) = \cos(x(t))v(t)$$

d'où

$$\dot{\hat{y}}(t) = \cos(\epsilon \hat{x}(t))v(\epsilon t) = \cos(\epsilon \hat{x}(t))\hat{v}(t)$$

En notant

$$R = \frac{\cos(\epsilon \hat{x}(t)) - 1}{\epsilon^2}$$

On peut donc écrire :

$$\dot{\hat{y}}(t) = (1 + \epsilon^2 R)\hat{v}(t)$$

- On a :

$$\dot{\varphi}(t) = \dot{x}(t) + \dot{z}(t) - \frac{1}{2}(\dot{x}(t)y(t) + x(t)\dot{y}(t)) = -\sin(\varphi(t))v(t) + u(t)$$

On a deux termes à évaluer :

- D'une part,

$$\begin{aligned} \dot{\varphi}(\epsilon t) &= \dot{x}(\epsilon t) + \dot{z}(\epsilon t) - \frac{1}{2}(\dot{x}(\epsilon t)y(\epsilon t) + x(\epsilon t)\dot{y}(\epsilon t)) \\ &= \dot{\hat{x}}(t) + \epsilon \dot{\hat{z}}(t) - \frac{\epsilon}{2}(\dot{\hat{x}}(t)\hat{y}(t) + \hat{x}(t)\dot{\hat{y}}(t)) \end{aligned}$$

D'où

$$\dot{\varphi}(\epsilon t) = \hat{u}(t) + \epsilon \dot{\hat{z}}(t) - \frac{\epsilon}{2}(\hat{u}(t)\hat{y}(t) + \hat{x}(t)(1 + \epsilon^2 R)\hat{v}(t))$$

- D'autre part

$$\dot{\varphi}(\epsilon t) = -\sin\left(\epsilon \hat{x}(t) + \epsilon^2 \hat{z}(t) - \frac{\epsilon^2}{2}\hat{x}(t)\hat{y}(t)\right)\hat{v}(t) + \hat{u}(t)$$

Donc

$$\dot{\varphi}(\epsilon t) = \left(-\epsilon \hat{x}(t) - \epsilon^2 \hat{z}(t) + \frac{\epsilon^2}{2}\hat{x}(t)\hat{y}(t) + \epsilon^3 Q\right)\hat{v}(t) + \hat{u}(t)$$

où

$$Q = \frac{-\sin\left(\epsilon \hat{x}(t) + \epsilon^2 \hat{z}(t) - \frac{\epsilon^2}{2}\hat{x}(t)\hat{y}(t)\right) + \epsilon \hat{x}(t) + \epsilon^2 \hat{z}(t) - \frac{\epsilon^2}{2}\hat{x}(t)\hat{y}(t)}{\epsilon^3}$$

Maintenant, il ne reste plus qu'à identifier :

$$\left(-\epsilon \hat{x}(t) - \epsilon^2 \hat{z}(t) + \frac{\epsilon^2}{2}\hat{x}(t)\hat{y}(t) + \epsilon^3 Q\right)\hat{v}(t) + \hat{u}(t)$$

et

$$\hat{u}(t) + \epsilon \dot{\hat{z}}(t) - \frac{\epsilon}{2}(\hat{u}(t)\hat{y}(t) + \hat{x}(t)(1 + \epsilon^2 R)\hat{v}(t))$$

Après simplification et en exprimant tout en fonction de $\dot{\hat{z}}$, on obtient :

$$\dot{\hat{z}}(t) = \frac{1}{2}\hat{y}(t)\hat{u}(t) - \frac{1}{2}\hat{x}(t)\hat{v}(t) + \left(-\epsilon \hat{z}(t) + \frac{\epsilon}{2}\hat{x}(t)\hat{y}(t) + \epsilon^2(Q + \frac{R}{2}\hat{x}(t))\right)\hat{v}(t)$$

Au final, on pose :

$$K = \left(-\hat{z}(t) + \frac{1}{2}\hat{x}(t)\hat{y}(t) + \epsilon(Q + \frac{R}{2}\hat{x}(t)) \right) \hat{v}(t)$$

Et donc,

$$\dot{\hat{z}}(t) = \frac{1}{2}\hat{y}(t)\hat{u}(t) - \frac{1}{2}\hat{x}(t)\hat{v}(t) + \epsilon K$$

- Il reste maintenant à faire de même sur la dernière composante.

On procède toujours de la même manière : on va calculer $\dot{\alpha}(\epsilon t)$ de deux manières différentes :

◦ D'une part

$$\begin{aligned} \dot{\alpha}(t) = & \dot{z}(t) - \frac{1}{2}\dot{x}(t)y(t) - \frac{1}{2}x(t)\dot{y}(t) - \frac{2}{3}\dot{w}(t) + y(t)\dot{z}(t) + \dot{y}(t)z(t) - \\ & \frac{1}{6}(2y(t)\dot{y}(t)x(t) + y(t)^2\dot{x}(t)) \end{aligned}$$

d'où :

$$\begin{aligned} \dot{\alpha}(\epsilon t) = & \left(\epsilon\dot{z}(t) - \frac{\epsilon}{2}\hat{u}(t)\hat{y}(t) - \frac{\epsilon}{2}\hat{x}(t)(1 + \epsilon^2 R)\hat{v}(t) - \frac{2}{3}\epsilon^2\dot{w}(t) + \epsilon^2\hat{y}(t)\dot{z}(t) + \right. \\ & \left. \epsilon^2\dot{z}(t)(1 + \epsilon^2 R)\hat{v}(t) - \frac{1}{6}\left(2\epsilon^2\hat{y}(t)\hat{x}(t)(1 + \epsilon^2 R)\hat{v}(t) + \epsilon^2\hat{y}(t)^2\hat{u}(t)\right) \right) \end{aligned}$$

En utilisant le résultat qu'on vient d'avoir sur $\dot{\hat{z}}(t)$ (on voit l'intérêt de la structure triangulaire du système dans ces calculs, qui seraient beaucoup plus complexes sans cela), on écrit :

$$\begin{aligned} \dot{\alpha}(\epsilon t) = & \left(\epsilon\left(\frac{1}{2}\hat{y}(t)\hat{u}(t) - \frac{1}{2}\hat{x}(t)\hat{v}(t) + \epsilon K\right) - \frac{\epsilon}{2}\hat{u}(t)\hat{y}(t) - \frac{\epsilon}{2}\hat{x}(t)(1 + \epsilon^2 R)\hat{v}(t) - \right. \\ & \frac{2}{3}\epsilon^2\dot{w}(t) + \epsilon^2\hat{y}(t)\left(\frac{1}{2}\hat{y}(t)\hat{u}(t) - \frac{1}{2}\hat{x}(t)\hat{v}(t) + \epsilon K\right) + \epsilon^2\dot{z}(t)(1 + \epsilon^2 R)\hat{v}(t) - \\ & \left. \frac{1}{6}\left(2\epsilon^2\hat{y}(t)\hat{x}(t)(1 + \epsilon^2 R)\hat{v}(t) + \epsilon^2\hat{y}(t)^2\hat{u}(t)\right) \right) \end{aligned}$$

On voit déjà quelques termes qui se simplifient et qui s'arrangent.

◦ D'autre part :

$$\dot{\alpha}(\epsilon t) = \cos(\theta(\epsilon t))v(\epsilon t) = \cos(x(\epsilon t) + \pi/2)\hat{v}(t) = -\sin(\epsilon\hat{x}(t))\hat{v}(t) = -\epsilon\hat{x}(t)\hat{v}(t) + \epsilon^3 P$$

où

$$P = \left(\frac{\epsilon\hat{x}(t)\hat{v}(t) - \sin(\epsilon\hat{x}(t))\hat{v}(t)}{\epsilon^3} \right)$$

Dès lors, en identifiant et en simplifiant les termes qui s'annulent on obtient :

$$\begin{aligned} \frac{2\epsilon^2}{3}\dot{w}(t) &= -\epsilon^3 P - \frac{\epsilon^3}{2}R\hat{x}(t)\hat{v}(t) + \frac{1}{3}\epsilon^2\hat{y}(t)^2\hat{u}(t) - \frac{5}{6}\epsilon^2\hat{y}(t)\hat{x}(t)\hat{v}(t) + \epsilon^2\hat{z}(t)\hat{v}(t) + \\ &\quad \epsilon^3\hat{y}(t)K + \epsilon^4\left(R\hat{z}(t)\hat{v}(t) - \frac{1}{3}\hat{y}(t)\hat{x}(t)\hat{v}(t)R\right) + \epsilon^2 K \end{aligned}$$

Et, comme on s'en doute, les termes en ϵ^2 vont s'arranger pour donner ce que l'on veut : en effet,

$$\epsilon^2 K = -\hat{z}(t)\hat{v}(t)\epsilon^2 + \frac{1}{2}\hat{x}(t)\hat{y}(t)\hat{v}(t)\epsilon^2 + \epsilon^3\left(Q + \frac{R}{2}\hat{x}(t)\right)\hat{v}(t)$$

et donc,

$$\dot{w}(t) = \frac{1}{2}\hat{y}(t)^2\hat{u}(t) - \frac{1}{2}\hat{x}(t)\hat{y}(t)\hat{v}(t) + \epsilon M$$

où

$$M = -\frac{3}{2}(P - Q) + \frac{3}{2}\hat{y}(t)K + \frac{3}{2}\epsilon\left(R\hat{z}(t)\hat{v}(t) - \frac{1}{3}\hat{x}(t)\hat{y}(t)R\hat{v}(t)\right)$$

Après ces longs calculs, qui ont le mérite de valider et préciser nos formules, on peut faire le bilan :

$$\begin{aligned} \begin{pmatrix} \dot{\hat{x}} \\ \dot{\hat{y}} \\ \dot{\hat{z}} \\ \dot{\hat{w}} \end{pmatrix} &= \begin{pmatrix} \hat{u} \\ \hat{v} \\ \frac{1}{2}\hat{y}\hat{u} - \frac{1}{2}\hat{x}\hat{v} \\ \frac{1}{2}\hat{y}^2\hat{u} - \frac{1}{2}\hat{x}\hat{y}\hat{v} \end{pmatrix} + \begin{pmatrix} 0 \\ \epsilon^2 R' \\ \epsilon K \\ \epsilon M \end{pmatrix} \\ \text{avec } R &= \frac{\cos(\epsilon\hat{x})-1}{\epsilon^2} \text{ et } R' = \hat{v}R \\ Q &= \frac{-\sin\left(\epsilon\hat{x} + \epsilon^2\hat{z} - \frac{\epsilon^2}{2}\hat{x}\hat{y}\right) + \epsilon\hat{x} + \epsilon^2\hat{z} - \frac{\epsilon^2}{2}\hat{x}\hat{y}}{\epsilon^3} \\ K &= \left(-\hat{z} + \frac{1}{2}\hat{x}\hat{y} + \epsilon\left(Q + \frac{R}{2}\hat{x}\right)\right)\hat{v} \\ P &= \left(\frac{\epsilon\hat{x}\hat{v} - \sin(\epsilon\hat{x})\hat{v}}{\epsilon^3}\right) \\ M &= -\frac{3}{2}(P - Q) + \frac{3}{2}\hat{y}K + \frac{3}{2}\epsilon R\left(\hat{z} - \frac{1}{3}\hat{x}\hat{y}\right)\hat{v} \end{aligned}$$

Dans le système précédent nous voyons apparaître d'un côté le système nilpotent et de l'autre un reste qui est de plus en plus négligeable lorsque ϵ diminue. Ceci nous permettra de contrôler notre approximation et de savoir dans quelle mesure elle reste bonne. En fait, on se rendra compte dans les simulations que même avec $\epsilon = 0,9$ les résultats seront bons.

Pour simuler, nous travaillerons avec ce système qui a l'avantage d'homogénéiser les grandeurs et nous reviendrons à la configuration de départ par la transformation citée à la page 19. Nous pourrons ainsi représenter notre voiture et sa remorque à chaque instant.

Un autre problème non trivial se pose alors : comment représenter clairement le mouvement de la voiture et de sa remorque ? La solution naturelle consiste à dilater l'axe des abscisses pour voir la voiture avancer car la voiture se déplace beaucoup plus selon β que selon α . Une dilatation sur l'axe des abscisses semble donc bienvenue pour visualiser le déplacement horizontal. Sans dilatation, nous ne voyons pas nettement le déplacement sur l'axe des α sauf si nous itérons un grand nombre de fois les *boucles* (n'oublions pas que le processus est périodique et que par conséquent la voiture donne l'impression de faire des *boucles*).

Si nous dilatons l'axe des abscisses, le problème de repères orthogonaux mais non normés apparaît. La dilatation a des répercussions sur les angles et les longueurs des segments de la figure si nous ne prenons pas en compte le fait qu'on travaille alors dans des repères non orthonormés. Ce problème géométrique peut être résolu mathématiquement de façon assez simple (en prenant en compte le fait d'un repère seulement orthogonal), mais cela donne des erreurs au niveau cinématique et mécanique (glissement de la remorque notamment, alors que nous représentons un roulement sans glissement). Une dilatation de l'axe des abscisses ne convient donc pas pour la visualisation, elle donne des résultats irréalistes quand nous regardons l'animation de la voiture et sa remorque. Néanmoins, elle nous permet de distinguer clairement chaque *boucle* de la trajectoire.

En fait, le mieux reste de se placer en repère orthonormé et d'itérer de nombreuses fois les *boucles* pour voir que la voiture avance bel et bien (car rappelons que la voiture avance très peu à chaque étape et que le déplacement est presque imperceptible à des échelles normales). L'inconvénient de cette méthode est la grosse place mémoire occupée et la difficulté par la suite de créer l'animation sous forme de gif ou de video. Cependant, avec un bon choix des paramètres et de ϵ , on obtient des résultats satisfaisants comme nous le verrons sur les illustrations ci-dessous.

4.2 Algorithme

Le but de notre programme est de simuler le mouvement de notre voiture et de sa remorque quand le système est soumis aux contrôles u et v que l'on a trouvé dans l'étude ci-dessus. En entrée le programme prend donc ces deux fonctions u et v , fonctions du temps. En sortie, le programme crée une série d'images, grâce à laquelle on va voir le mouvement de la voiture et de sa remorque en deux dimensions, vue du dessus. Comme le modèle cinématique choisi est simplifié, la représentation se doit d'être aussi simple pour concorder avec le modèle. Sinon, en dessinant une *vraie* voiture, des impressions de glissement peuvent apparaître et donner lieu à un mouvement qui paraît irréaliste.

Le principe du programme Matlab que nous avons créé est le suivant.

On commence par résoudre le système différentiel renormalisé trouvé précédemment. Nous savons que nous allons obtenir la trajectoire demandée et en plus la meilleure. Bien sûr nous n'aurons qu'une approximation de la solution vue l'impossibilité de résoudre formellement un tel système différentiel. Les packages ODE permettent d'avoir aisément un solution discrétisée (aussi précise que l'on souhaite, suivant la mémoire et le temps de calcul que l'on supporte) approximant la solution réelle. Ensuite, pour chaque instant t , il suffit de créer une figure représentant la voiture et sa remorque dans la configuration à cet instant et ensuite d'exporter la figure sous forme d'image. Plus précisément, on calcule à chaque instant toutes les coordonnées des sommets de la figure (voiture et remorque) et on crée les faces des objets. Ensuite, il suffit de créer un gif animé ou une video pour visualiser l'animation sur n'importe quel ordinateur.

Le temps d'exécution est de l'ordre de quelques secondes, mais cela varie assez selon la précision et le nombre de *boucles* demandés. La résolution du système différentiel prend la majorité du temps. Le reste du temps d'exécution est occupé par la création des images à chaque instant.

Le programme complet se trouve en annexe. Des commentaires sont présents pour permettre la compréhension du code dans le détail.

4.3 Illustrations, captures d'écran

Voici quelques illustrations de nos propos.

Dans ces exemples, nous avons pris $\epsilon = 0.9$ et le temps varie de 0 à 4. (Il y a quatre boucles d'effectuées).

Nous vérifions bien que la voiture avance sur la figure 4.1 (sans tangente horizontale) avec la présence de points d'inflexion qui induisent des marches avant puis des marches arrières de la voiture. Sur la figure 4.1, l'axe des abscisses a été dilaté pour qu'on distingue bien les 4 *boucles* séparément. La même figure dans un repère orthonormé aurait donné un résultat non pertinent. On voit aussi que la voiture ne donne plus l'impression de rester verticale à cause de cette déformation abusive de la trajectoire réelle. Les problèmes discutés précédemment s'illustrent parfaitement ici.

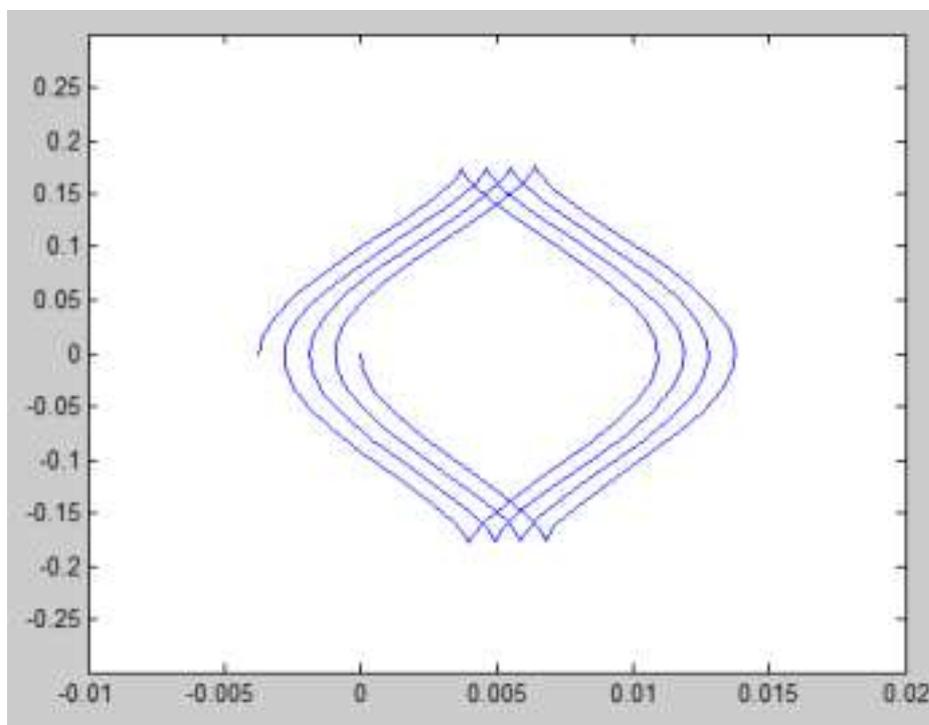


FIG. 4.1 – Le mouvement du centre des roues (α, β) .

Les figures 4.2 et 4.3 sont les valeurs des deux angles au cours du temps. Les courbes de θ et φ montrent que les valeurs de ces angles restent respectivement autour de $\pi/2$ et 0, ce qui prouve que la voiture va rester approximativement dans la configuration annoncée par Γ (défini dans la figure 1.1, page 3).

La figure 4.4 montre la trajectoire de la voiture. Elle est extraite d'un des gifs que nous avons créés. La voiture se déplace très légèrement vers la gauche de façon presque imperceptible quand il y a trop peu d'itérations. Dans cette figure, nous sommes restés en repère orthonormé pour ne pas avoir un résultat faussé. On perçoit tout de même le déplacement, indiqué principalement par l'épaisseur de la *tâche* traçant la trajectoire suivie par le mobile.

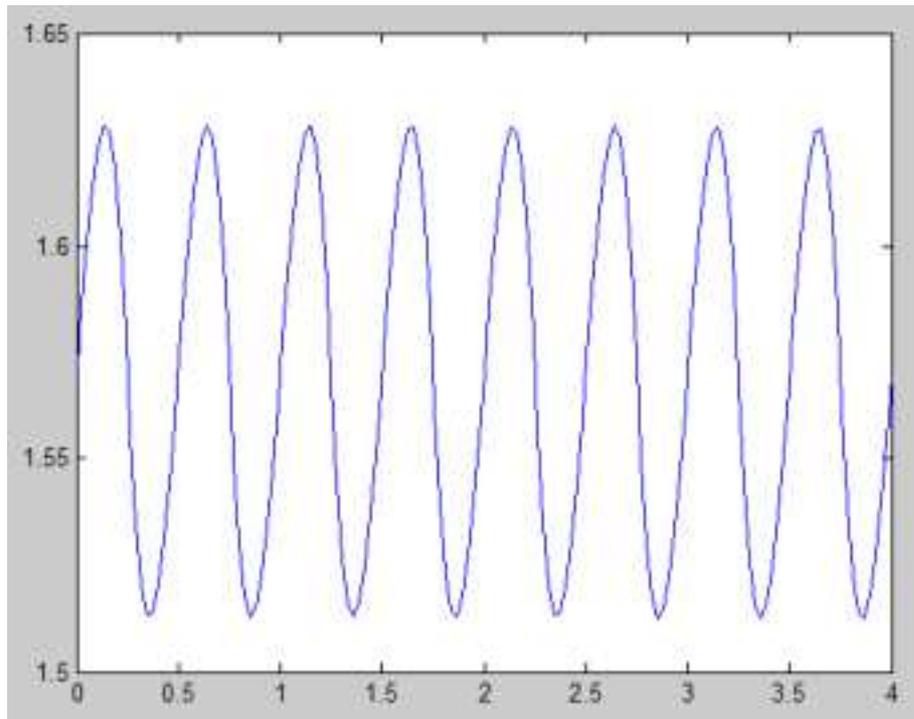


FIG. 4.2 – La valeur de θ en fonction du temps.

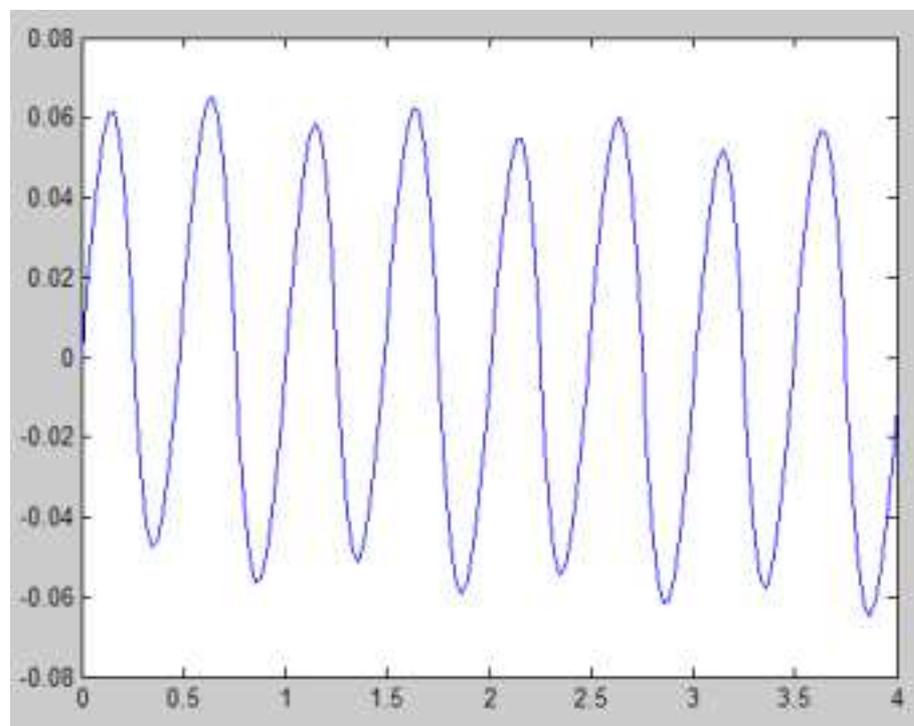


FIG. 4.3 – La valeur de φ en fonction du temps.

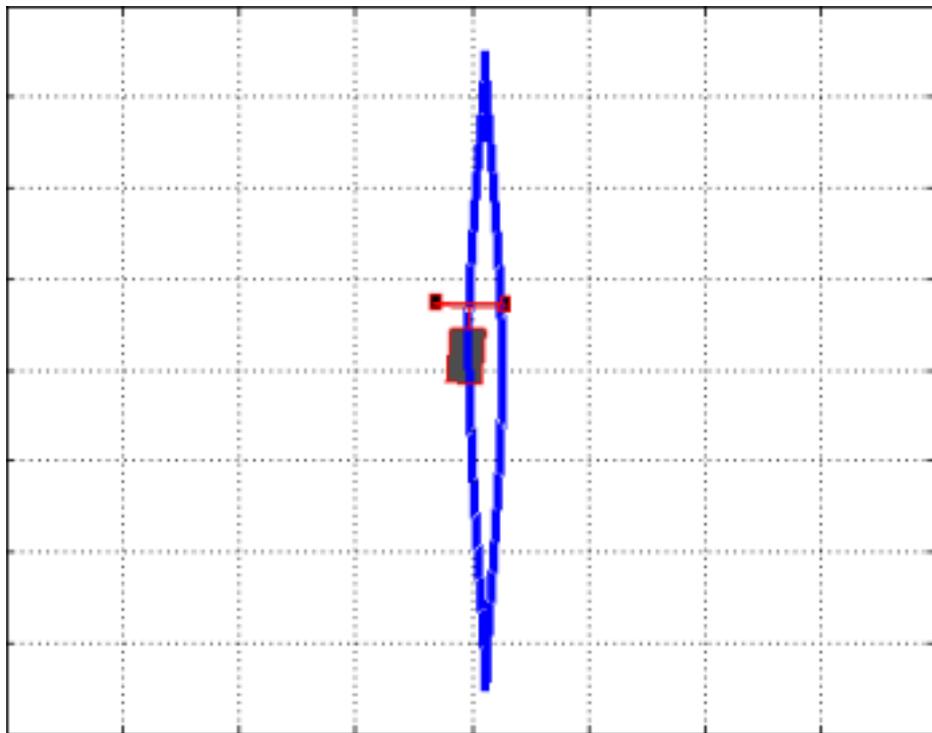


FIG. 4.4 – Exemple de figure voiture et remorque lors de son déplacement.

Chapitre 5

Conclusion

Nous nous sommes intéressés dans ce document à l'étude d'une des trajectoires les plus extrêmes pour une voiture et sa remorque. Cette étude a été menée dans le but de concevoir et visualiser concrètement les résultats obtenus par la théorie qui affirme l'existence d'une solution approchante. Le résultat des simulations concorde parfaitement avec la théorie et les contraintes que nous avons imposées. De plus, il n'était pas évident de pronostiquer l'allure de la trajectoire optimale. Cette étude permet de voir tout l'intérêt de la géométrie sous-riemannienne et du contrôle optimal dans ce type de problème. Etant encore relativement récente (milieu du 20ème siècle), cette théorie est actuellement en plein développement et de plus en plus de résultats ont été démontrés ces dernières années.

Pour généraliser cette étude, nous pourrions écrire un algorithme qui permet de traiter tout type de trajectoire (pas seulement des droites!). En fait toute courbe peut-être approchée par des segments de droites. En généralisant notre étude pour un segment quelconque de l'espace ambiant, nous pourrions traiter le problème optimal pour une trajectoire quelconque. D'ailleurs nous avons déjà spécifié que dans la pratique nous partons d'une courbe obtenue par des méthodes d'interpolation et évitant certains obstacles. Néanmoins, les calculs deviennent plus complexes puisque de nombreux paramètres entrent en jeu et l'utilisation d'un logiciel de calcul formel devient indispensable.

Nous pourrions aussi nous intéresser à l'utilisation de telles méthodes dans d'autres domaines tels que l'étude du mouvement humain. Par exemple, il serait intéressant d'étudier le lien entre l'optimisation faite par le corps humain lors d'une action de pointage complexe et l'optimisation donnée par la théorie présentée ici. Nous pourrions aussi étudier d'autres problèmes de robotique et d'automatique, notamment les bras articulés ou encore la cinématique inverse (problème aussi étudié dans le domaine de l'animation et la déformation d'objets dans des images).

Chapitre 6

Annexe

Voici le programme complet utilisé pour la simulation.

D'abord la fonction qui résout le système différentiel et trouve les solutions.

```
..... calcul_coef.m .....

function [ALPHA,BETA,THETA,PHI]=calcul_coef()

%% on fixe la valeur de epsilon
epsi='0.9'; epsi2=0.9;

%% donnée des controles
u = ['(1-2*(jacobi_DN(2.32073*(1+4*t)))^2)'];
v=['(-2*jacobi_DN(2.32073*(1+4*t))*jacobi_SN(2.32073*(1+4*t))*0.9088)'];

% initiaux instant t=0
x0 = 0; y0 = 0; z0 = 0; w0 = 0; init=[x0 y0 z0 w0];

% durée et nombre images par seconde
T=4; duree = 0:T/150:T;

%%% création du système à intégrer

R=['((cos(' epsi '*y(1))-1)/' epsi '^2)']; Q=['((-sin(' epsi '*y(1)+
epsi '^2*y(3)-1/2*' epsi '^2*y(1)*y(2))+' epsi '*y(1)+' epsi
'^2*y(3)-1/2*' epsi '^2*y(1)*y(2))/' epsi '^3)']; P=['((' epsi
'*y(1)*' v ' - sin(' epsi '*y(1))*' v ')/' epsi '^3)'];
K=['((-y(3)+1/2*y(1)*y(2)+' epsi '*(' Q '+1/2*y(1)*' R '))'*' v ')'];
M=['(-3/2*(' P '-' Q ')+3/2*y(2)*' K '+3/2*' epsi '*(' R '*y(3)*' v
'-1/3*y(1)*y(2)*' R '*' v ')')'];

fid2 = fopen('system2.m','w'); fprintf(fid2,['function
dy=system2(t,y)\ndy = zeros(4,1);\ndy(1)=' u ';' \ndy(2)=' v '+' epsi
'^2*' R '*' v ';' \ndy(3)=1/2*(y(2)*' u '-y(1)*' v ')+' epsi '*' K
';\ndy(4)=1/2*(y(2)*(y(2)*' u '-y(1)*y(2)*' v ')+' epsi '*' M ';'']);
```

```

fclose(fid2);

%%% Résolution du système avec ODE
options = odeset('RelTol',1e-8,'AbsTol',[1e-10 1e-10 1e-10 1e-10]);
[t,SOL2] = ode45(@system2,duree,init,options); X=SOL2(:,1);
Y=SOL2(:,2); Z=SOL2(:,3); W=SOL2(:,4);

%%% transformation pour donner les vrais alpha, beta, theta, phi
ALPHA=epsi2^2*Z-1/2*epsi2^2*X.*Y-2*1/3*epsi2^3*W
+epsi2^3*Y.*Z-1/6*epsi2^3*Y.*X; BETA=epsi2*Y; THETA=epsi2*X+pi/2;
PHI=epsi2*X+epsi2^2*Z-1/2*epsi2^2*X.*Y;

end

```

Voici maintenant la fonction qui crée la figure à chaque instant

```

..... compute.m .....

function compute()

clf; [ALPHA,BETA,THETA,PHI]=calcul_coef();
% paramètres d'affichage
xmin=-0.2; xmax=0.2; ymin=-0.2; ymax=0.2;

fact=xmax/ymax; %facteur d'echelle

%voiture
long2=0.002; long = 0.06 ; larg = 0.03 ;

%attache
L = 0.2*long;

%remorque
long_ = 0.5*long; larg_ = 0.5*larg;

%roues (dimensions)
roues = 0.15*larg ; roues_ = 0.15*long ;

%initiaux instant t=0
x0 = 0; y0 = 0; z0 = 0; w0 = 0; init=[x0 y0 z0 w0];

theta0=pi/2;phy0=0;

%%% création de la figure à chaque instant...
duree=length(ALPHA);

for k=1:duree-1

```

```

theta=pi/2+(THETA(k)-pi/2); phi=PHI(k); x=ALPHA(k); y=BETA(k);

% instant d'après pour orienter les roues

x_=ALPHA(k+1); y_=BETA(k+1);

norme=sqrt((cos(theta)^2/fact^2)+sin(theta)^2);

dir1_x=cos(theta)/norme; dir1_y=sin(theta)/norme;

orth1_x=-fact*sin(theta)/norme; orth1_y=cos(theta)/(fact*norme);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% listes des sommets : voiture, attache remorque...
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

s1_x = x + (larg/2)*orth1_x ; s1_y = y + (larg/2)*orth1_y ;

s2_x = x - (larg/2)*orth1_x ; s2_y = y - (larg/2)*orth1_y ;

% 3ème et 4ème sommet obtenu par translations des 1 et 2ème
s3_x = s2_x - long2*dir1_x ; s3_y = s2_y - long2*dir1_y ;

s4_y = s1_y - long2*dir1_y ; s4_x = s1_x - long2*dir1_x ;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% pour la remorque et son attache (6 sommets de +)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

norme2=sqrt((cos(theta+phi)^2/fact^2)+sin(theta+phi)^2);
at_x=cos(theta+phi)/norme2; at_y=sin(theta+phi)/norme2;

s11_x = (s4_x+s3_x)/2 - L*at_x; s11_y = (s4_y+s3_y)/2 - L*at_y;
s12_x = (s4_x+s3_x)/2; s12_y = (s4_y+s3_y)/2;

%% sommets remorque...
ath_x=-fact*sin(theta+phi)/norme2;
ath_y=cos(theta+phi)/(fact*norme2);

s13_x = s11_x - (larg_/2)*ath_x; s13_y = s11_y - (larg_/2)*ath_y ;

s14_x = s11_x + (larg_/2)*ath_x; s14_y = s11_y + (larg_/2)*ath_y ;

%k1 et k2 composantes du vecteur translation
k1_ = long_*fact*cos(theta+phi); k2_ = long_*sin(theta+phi);

% 3ème et 4ème sommet obtenu par translations des 1 et 2ème
s15_x = s14_x - long_*at_x ; s15_y = s14_y - long_*at_y ;

```

```

s16_y = s13_y - long_*at_y ; s16_x = s13_x - long_*at_x ;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% pour les roues...
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%vecteur directeur des roues * longueur roue...
n=sqrt(((x_-x)^2/fact^2+(y_-y)^2)); if n<1e-2 t1 = 0; t2 = roues_;
else t1 = roues_*(x_-x)/n; t2 = roues_*(y_-y)/n; end

%vecteur orthogonal remis à son echelle...
if n<1e-2 n1 = -fact*roues; n2 = 0; else n1 = -fact*roues*(y_-y)/n;
n2 = roues_*(x_-x)/(fact*n); end

%% génération des 4 sommets pour les roues...

%roue gauche
s5_x = s1_x + 0.5*(t1+n1) ; s5_y = s1_y + 0.5*(t2+n2) ;

s6_x = s1_x + 0.5*(t1-n1) ; s6_y = s1_y + 0.5*(t2-n2) ;

s7_x = s1_x + 0.5*(-t1-n1) ; s7_y = s1_y + 0.5*(-t2-n2) ;

s8_x = s1_x + 0.5*(-t1+n1) ; s8_y = s1_y + 0.5*(-t2+n2) ;

%roue droite
%translation roue gauche à droite
r1 = s2_x - s1_x; r2 = s2_y - s1_y;

s9_x = s5_x + r1 ; s9_y = s5_y + r2 ;

s10_x = s6_x + r1 ; s10_y = s6_y + r2 ;

s17_x = s7_x + r1 ; s17_y = s7_y + r2 ;

s18_x = s8_x + r1 ; s18_y = s8_y + r2 ;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% DESSIN %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

v = [s1_x s1_y ; s2_x s2_y ; s3_x s3_y ; s4_x s4_y ; s12_x s12_y ;
s11_x s11_y; s13_x s13_y ; s14_x s14_y ; s15_x s15_y ; s16_x s16_y ;
s5_x s5_y ; s6_x s6_y ; s7_x s7_y ; s8_x s8_y ; s9_x s9_y ; s10_x
s10_y ; s17_x s17_y ; s18_x s18_y ] ;

f = [1 2 3 4 ; 5 6 6 5 ; 7 8 9 10 ; 11 12 13 14 ; 15 16 17 18] ;

set(gca,'xlim',[xmin xmax],'ylim',[ymin ymax]);

```

```

plot(ALPHA,BETA);axis([xmin xmax ymin ymax]);

T.vertices = v ; T.faces = f; T=struct('vertices',v,'faces',f);

couleur = [ .7 .7 .7;0 0 0; .3 .3 .3 ; 0 0 0 ; 0 0 0 ];

h=patch(T,'FaceVertexCData', couleur, 'FaceColor','flat');

set(h,'EraseMode','normal','EdgeColor','r'); grid on
k2=int2str(k); % pour ordonner les images par la suite...
a = fi(k,0,10,0);k2=bin(a);

F = getframe(gca);

imwrite(F.cdata, ['images/' k2 '.png']);

clf;

end

end

```

Et bien sûr les fonctions elliptiques de Jacobi

```

..... Jacobi_DN .....

function dn_=jacobi_DN(u)

[SN,CN,DN] = ellipj(u,0.8259962710);

dn_=DN;

end

..... Jacobi_SN .....

function sn_=jacobi_SN(u)

[SN,CN,DN] = ellipj(u,0.8259962710);

sn_=SN;

end

```

Bibliographie

- [1] Love A.E.H. *A treatise on the mathematical theory of elasticity*. 1927.
- [2] Gauthier J-P. Agrachev A.A. Subriemannian metrics and isoperimetric. *Rapport de recherche du laboratoire de Topologie de l'université de Bourgogne.*, 1998.
- [3] Gauthier J-P. Agrachev A.A., Chakir El-H. Sub-riemannian metrics on \mathbb{R}^3 . *Canadian Mathematical Society, Conference Proceedings*, 25, 1998.
- [4] Gromov M.-Montgomery R. Bellaïche A., Risler J-J. Sub-riemannian geometry. *Birkhäuser Verlag*, 144, 1996.
- [5] Romero-Meléndez C. *Complexité métrique sous-riemannienne*. PhD thesis, 2004.
- [6] Charlot G. *Géométrie sous-riemannienne de contact et de quasi-contact*. PhD thesis, 2001.
- [7] Zakalyukin V. Gauthier J-P. On the codimension one motion planning problem. 2004.
- [8] Zakalyukin V. Gauthier J-P. On the motion planning problem, complexity, entropy and nonholonomic interpolation. *A paraître*, 2005.
- [9] Lafontaine J. *Introduction à la géométrie différentielle*. 1996.
- [10] Jean F. Laumond J-P. Robot motion planning and control. *LAAS report*, 1998.
- [11] Gamkrelidze R.V. Mischenko E.F. Pontryagin L.S., Boltyanskii V.G. *The Mathematical Theory of Optimal Processes*. 1964.
- [12] Monroy-Pérez F. Romero-Meléndez C., Gauthier J-P. On complexity and motion planning for co-rank one sub-riemannian metrics. *ESAIM : control, optimisation and calculus of variations*, 2005.
- [13] Jurdjevic V. *Geometric Control Theory*. 1997.