



Développement du robot EMABOT (partie 1)

& Asservissement des moteurs

Projet long présenté par Mathieu SEILER

Projet supervisé par :

M. Couturier

M. Bataille

M. Meinouni

Sommaire

I.	Introduction.....	4
I.	Clothoïde	5
a.	Etude des courbes paramétrées dans le plan	5
i.	Domaine Curviligne	5
ii.	Domaine Temporel.....	7
b.	Application à la clothoïde.....	9
II.	Modélisation mathématique.....	11
a.	Mise en place des hypothèses.....	11
b.	Calculs.....	12
c.	Asservissement Sigma-Delta	13
III.	Adhérence	13
a.	Etude	13
i.	Modélisation plane.....	13
ii.	Modélisation 3 dimensions	14
b.	Mesure de coefficient d'adhérence de roues	16
IV.	Etude paramétrique du robot	17
a.	Etude générale	17
b.	Cas d'une trajectoire en ligne droite	19
i.	Etude des phases du robot.....	21
ii.	Conclusion	23
c.	Cas d'une Clothoïde.....	24
V.	Génération de trajectoire à partir d'une liste de points et d'orientations	25
a.	Analyse de la clothoïde	25
b.	Approximation des intégrales de Fresnel.....	30
c.	Clothoïde en S.....	33
d.	Algorithme de génération de trajectoire par "blocs"	35
i.	Principes	35
VI.	Génération de trajectoire à partir d'une liste de points et d'orientations par méthode itérative	36
ANNEXES.....		44
Annexe 1 : Roulement avec glissement		45
Annexe 2 : Programme Matlab pour dessiner une clothoïde		46
Glossaire		48
Bibliographie		49

Conclusion	50
Résumé	51

I. Introduction

Un grand problème se pose toujours quand il s'agit de connaître la position d'un objet mobile. Dans un premier temps on pense tout de suite au fameux GPS qui nous permet de connaître notre position sur la Terre à 5m prêt. C'est un bon moyen de repérage, cependant sa précision ne lui permet pas d'être utiliser dans tous les domaines (on ne peut pas penser suivre à la trace une fourmi avec GPS). Quand il s'agit de positionner un robot, on préférerait avoir une précision de l'ordre du millimètre, voire encore moins.

Pour cela il convient de préciser deux modes de repérages, le repérage absolu et le repérage relatif. On appelle repérage relatif tout repérage à partir d'un point fixe sur terre que l'on connaît, ce repérage se base en calculant ses avancées successives pour prédire sa position comme une souris de PC. On appelle repérage absolu tout repérage qui ne dépend pas de sa position antérieure, par exemple le GPS. Il y a des avantages et des inconvénients aux deux systèmes. Le repérage relatif a l'avantage d'être très précis sur des courtes distances, mais au fur et à mesure il accumule des erreurs. Le repérage absolu a l'avantage de ne pas accumuler d'erreur, mais par contre n'est pas assez précis. Il serait bon de pouvoir utiliser les avantages et les inconvénients de chacun, c'est pourquoi il existe ce qu'on appelle un Filtre de Kalman qui permet de fusionner les données perçues par un repérage relatif et un absolu pour obtenir une meilleure approximation possible de sa position. En fait dans ce cas le Filtre de Kalman se sert du repérage absolu pour re-calibrer la dérive du repérage relatif. C'est ce que nous prévoyons d'utiliser pour le robot.

Dans notre robot nous pensons utiliser un système de repérage absolu à partir de 3 balises (voir dossier balise) que nous pouvons poser sur l'aire de jeu, couplé avec un système de repérage relatif basé sur un système odométrique. Je vais donc vous expliquer le principe de l'odométrie tout au long de ce rapport.

I. Clothoïde

Dans un premier temps nous allons faire les rappels mathématiques concernant les notions de courbes paramétrées dans le plan. Puis nous verrons par la suite le cas particulier qui nous intéresse, la clothoïde.

a. Etude des courbes paramétrées dans le plan

Soit $B = (O, \vec{i}, \vec{j})$ un repère orthonormé direct de E^2 .

Soit $t \rightarrow M_{(t)} = O + \overrightarrow{F_{(t)}}$ un arc paramétré de classe C^1 sur I . $\overrightarrow{F_{(t)}} = \begin{pmatrix} x_{(t)} \\ y_{(t)} \end{pmatrix}$

On appelle abscisse curviligne l'application $t \rightarrow s_{(t)} = \int_{t_0}^t \|\overrightarrow{F_{(x)}}\| dx + s_{(t_0)}$

Soit J l'intervalle tel que $\forall t \in I, s(t) \in J$

Remarque : $\forall s \in J, ds = \|\overrightarrow{F_{(t)}}\| dt$

Nous appellerons par la suite Γ une courbe paramétrée par une abscisse curviligne ¹s.

i. Domaine Curviligne

Le vecteur $\overrightarrow{T}_{(s)} = \frac{d\overrightarrow{OM}}{ds} = \overrightarrow{F_{(s)}}$ est appelé vecteur tangent unitaire orienté à Γ au point $M_{(s)}$.

Unitaire car $\|\overrightarrow{T}_{(s)}\| = \left\| \frac{d\overrightarrow{OM}}{ds} \right\| = \left\| \frac{\sqrt{dx^2 + dy^2}}{ds} \right\| = \left\| \frac{ds}{ds} \right\| = 1$

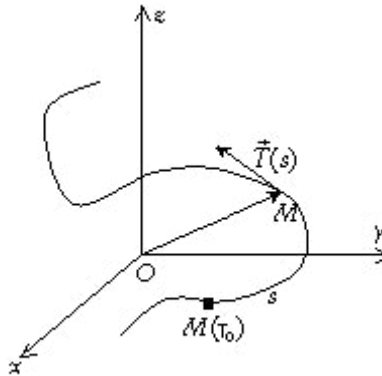


Figure 1 : Courbe Γ

Come nous avons $\forall s, \|\overrightarrow{T}_{(s)}\| = 1$, on peut donc écrire que $M_{(s)}$ a pour affixe $z = x_{(s)} + i.y_{(s)}$ et plus particulièrement $x'_{(s)} + i.y'_{(s)} = e^{i\alpha_{(s)}}$

Ce qui nous donne
$$\begin{cases} \forall s \in J, \overrightarrow{T}_{(s)} = \cos(\alpha_{(s)}) \cdot \vec{i} + \sin(\alpha_{(s)}) \cdot \vec{j} \\ \forall s \in J, \alpha_{(s)} = (\vec{i}, \overrightarrow{T}_{(s)})[2\pi] \end{cases}$$

¹ En géométrie différentielle, l'**abscisse curviligne** est une sorte de variante algébrique de la longueur d'un arc. On se donne une origine à partir de laquelle on calcule les longueurs, en les munissant d'un signe pour se situer de façon bien déterminée sur la courbe : à telle distance avant ou après le point initial. L'abscisse curviligne est donc l'analogue sur, une courbe, de l'abscisse sur une droite orientée. (Source Wikipédia)

Définition : on appelle courbure algébrique à la courbe Γ au point $M_{(s)}$, $\rho_{(s)} = \frac{d\alpha_{(s)}}{ds} = \alpha'_{(s)}$

Définition : on appelle rayon de courbure $R_{(s)} = \frac{1}{\rho_{(s)}} = \frac{1}{\alpha'_{(s)}}$

Définition : on appelle normale orientée au point $M_{(s)}$, le vecteur unitaire $\vec{N}_{(s)}$ tel que

$$(\vec{T}_{(s)}, \vec{N}_{(s)}) = +\frac{\pi}{2} [2\pi]$$

On a ainsi $(\vec{T}_{(s)}, \vec{N}_{(s)})$ une base orthonormée directe de E_2 .

$$\vec{N}_{(s)} \text{ se déduit facilement de } \vec{T}_{(s)} : \vec{T}_{(s)} = \begin{vmatrix} \cos(\alpha_{(s)}) \\ \sin(\alpha_{(s)}) \end{vmatrix}_B \xrightarrow{\text{on déduit}} \vec{N}_{(s)} = \begin{vmatrix} \cos(\alpha_{(s)} + \frac{\pi}{2}) \\ \sin(\alpha_{(s)} + \frac{\pi}{2}) \end{vmatrix}_B = \begin{vmatrix} -\sin(\alpha_{(s)}) \\ \cos(\alpha_{(s)}) \end{vmatrix}_B$$

Si $\alpha_{(s)}$ est dérivable au moins une fois, alors :

$$\vec{F}_{(s)}'' = \frac{d\vec{F}_{(s)}}{ds} = \frac{d\vec{T}_{(s)}}{ds}$$

$$\vec{F}_{(s)}'' = \begin{vmatrix} -\alpha'_{(s)} \cdot \sin(\alpha_{(s)}) \\ \alpha'_{(s)} \cdot \cos(\alpha_{(s)}) \end{vmatrix}_B = \alpha'_{(s)} \vec{N}_{(s)}$$

En exprimant en fonction du rayon de courbure

$$\vec{F}_{(s)}'' = \frac{\vec{N}_{(s)}}{R_{(s)}}$$

Récapitulons :

$$\vec{F}_{(s)} = \begin{vmatrix} \int_0^s \cos(\alpha_{(x)}) \cdot dx \\ \int_0^s \sin(\alpha_{(x)}) \cdot dx \end{vmatrix}_B$$

$$\vec{F}_{(s)} = \vec{T}_{(s)} = \begin{vmatrix} \cos(\alpha_{(s)}) \\ \sin(\alpha_{(s)}) \end{vmatrix}_B$$

$$\vec{F}_{(s)}'' = \frac{\vec{N}_{(s)}}{R_{(s)}} = \begin{vmatrix} -\alpha'_{(s)} \cdot \sin(\alpha_{(s)}) \\ \alpha'_{(s)} \cdot \cos(\alpha_{(s)}) \end{vmatrix}_B$$

ii. Domaine Temporel

Remarque : $\forall s \in J, ds = \|\vec{F}_{(t)}\| dt$

Nous appellerons par la suite Γ une courbe paramétrée par une abscisse curviligne ²s.

Le vecteur $\vec{V}_{(t)} = \frac{d\vec{OM}}{dt} = \vec{F}_{(t)}$ est appelé vecteur vitesse tangent orienté à Γ au point $M_{(t)}$.

$\vec{T}_{(t)} = \frac{\vec{F}_{(t)}}{\|\vec{F}_{(t)}\|}$ représente le vecteur unitaire directeur de $\vec{V}_{(t)}$ tel que $\vec{V}_{(t)} = V_{(t)} \cdot \vec{T}_{(t)}$

Come nous avons $\forall s, \|\vec{T}_{(t)}\| = 1$, on peut donc écrire que $\vec{T}_{(t)} = \cos(\alpha_{(t)}) \cdot \vec{i} + \sin(\alpha_{(t)}) \cdot \vec{j}$

Ainsi

$$\begin{cases} \forall s \in J, \vec{T}_{(t)} = \cos(\alpha_{(t)}) \cdot \vec{i} + \sin(\alpha_{(t)}) \cdot \vec{j} \\ \forall s \in J, \alpha_{(t)} = (\vec{i}, \vec{T}_{(t)})[2\pi] \end{cases}$$

On veut $(\vec{T}_{(t)}, \vec{N}_{(t)})$ une base orthonormée directe de E_2 .

$$\vec{N}_{(t)} \text{ se déduit facilement de } \vec{T}_{(t)} : \vec{T}_{(t)} = \begin{vmatrix} \cos(\alpha_{(t)}) \\ \sin(\alpha_{(t)}) \end{vmatrix}_B \xrightarrow{\text{on déduit}} \vec{N}_{(t)} = \begin{vmatrix} \cos(\alpha_{(t)} + \frac{\pi}{2}) \\ \sin(\alpha_{(t)} + \frac{\pi}{2}) \end{vmatrix}_B = \begin{vmatrix} -\sin(\alpha_{(t)}) \\ \cos(\alpha_{(t)}) \end{vmatrix}_B$$

Si $\alpha_{(t)}$ est dérivable au moins une fois, alors :

$$\begin{aligned} \vec{F}_{(t)}' &= \frac{d\vec{F}_{(t)}}{dt} = \frac{d(\|\vec{F}_{(t)}\| \cdot \vec{T}_{(t)})}{dt} = \frac{d\|\vec{F}_{(t)}\|}{dt} \cdot \vec{T}_{(t)} + \|\vec{F}_{(t)}\| \cdot \frac{d\vec{T}_{(t)}}{dt} \\ \vec{F}_{(t)}' &= \begin{vmatrix} \frac{d\|\vec{F}_{(t)}\|}{dt} \cdot \cos(\alpha_{(t)}) - \|\vec{F}_{(t)}\| \cdot \alpha'_{(t)} \cdot \sin(\alpha_{(t)}) \\ \frac{d\|\vec{F}_{(t)}\|}{dt} \cdot \sin(\alpha_{(t)}) + \|\vec{F}_{(t)}\| \cdot \alpha'_{(t)} \cdot \cos(\alpha_{(t)}) \end{vmatrix}_B = \frac{d\|\vec{F}_{(t)}\|}{dt} \vec{T}_{(t)} + \|\vec{F}_{(t)}\| \cdot \alpha'_{(t)} \vec{N}_{(t)} \end{aligned}$$

En exprimant en fonction du rayon de courbure

$$\vec{F}_{(t)}' = \frac{d\|\vec{F}_{(t)}\|}{dt} \vec{T}_{(t)} + \frac{\|\vec{F}_{(t)}\|}{R_{(t)}} \vec{N}_{(t)}$$

² En géométrie différentielle, l'**abscisse curviligne** est une sorte de variante algébrique de la longueur d'un arc. On se donne une origine à partir de laquelle on calcule les longueurs, en les munissant d'un signe pour se situer de façon bien déterminée sur la courbe : à telle distance avant ou après le point initial. L'abscisse curviligne est donc l'analogue sur, une courbe, de l'abscisse sur une droite orientée. (Source Wikipédia)

Récapitulons :

$$\begin{aligned} \overrightarrow{F}_{(t)} &= \int_0^t \left\| \overrightarrow{F}_{(x)} \right\| \cdot \cos(\alpha_{(x)}) \cdot dx \\ &\quad \int_0^t \left\| \overrightarrow{F}_{(x)} \right\| \cdot \sin(\alpha_{(x)}) \cdot dx \\ \overrightarrow{F}_{(t)} &= \left\| \overrightarrow{F}_{(t)} \right\| \cdot \overrightarrow{T}_{(t)} = \begin{pmatrix} \left\| \overrightarrow{F}_{(t)} \right\| \cdot \cos(\alpha_{(t)}) \\ \left\| \overrightarrow{F}_{(t)} \right\| \cdot \sin(\alpha_{(t)}) \end{pmatrix} \\ \overrightarrow{F}_{(t)}' &= \frac{d \left\| \overrightarrow{F}_{(t)} \right\|}{dt} \overrightarrow{T}_{(t)} + \frac{\left\| \overrightarrow{F}_{(t)} \right\|}{R_{(t)}} \overrightarrow{N}_{(t)} = \begin{pmatrix} \frac{d \left\| \overrightarrow{F}_{(t)} \right\|}{dt} \cdot \cos(\alpha_{(t)}) - \left\| \overrightarrow{F}_{(t)} \right\| \cdot \alpha'_{(t)} \cdot \sin(\alpha_{(t)}) \\ \frac{d \left\| \overrightarrow{F}_{(t)} \right\|}{dt} \cdot \sin(\alpha_{(t)}) + \left\| \overrightarrow{F}_{(t)} \right\| \cdot \alpha'_{(t)} \cdot \cos(\alpha_{(t)}) \end{pmatrix} \end{aligned}$$

b. Application à la clothoïde³

D'après la définition de la clothoïde, $\rho_{(s)} = \alpha'_{(s)} = \beta \cdot s$ et la vitesse est constante $\|F'_{(s)}\| = V_s$, $\|F'_{(t)}\| = V_t = V$

$\alpha_{(s)} = \beta \cdot s^2 + \alpha_0$. Supposons que $\alpha_0 = 0$.

On a donc

$$\begin{aligned}\overrightarrow{F}_{(s)} &= \begin{vmatrix} \int_0^s \cos(\frac{\beta}{2} \cdot x^2) \cdot dx \\ \int_0^s \sin(\frac{\beta}{2} \cdot x^2) \cdot dx \end{vmatrix}_B \\ \overrightarrow{F}_{(s)} = \overrightarrow{T}_{(s)} &= \begin{vmatrix} \cos(\frac{\beta}{2} \cdot s^2) \\ \sin(\frac{\beta}{2} \cdot s^2) \end{vmatrix}_B \\ \overrightarrow{F}_{(s)} = V \cdot \beta \cdot s \cdot \overrightarrow{N}_{(s)} &= \begin{vmatrix} -\beta \cdot s \cdot \sin(\frac{\beta}{2} \cdot s^2) \\ \beta \cdot s \cdot \cos(\frac{\beta}{2} \cdot s^2) \end{vmatrix}_B\end{aligned}$$

Passons dans le domaine temporel :

$$\begin{aligned}\overrightarrow{F}_{(t)} &= \begin{vmatrix} V \cdot \int_0^t \cos(\frac{\beta V^2}{2} \cdot x^2) \cdot dx \\ V \cdot \int_0^t \sin(\frac{\beta V^2}{2} \cdot x^2) \cdot dx \end{vmatrix}_B \\ \overrightarrow{F}_{(t)} = V \cdot \overrightarrow{T}_{(t)} &= \begin{vmatrix} V \cdot \cos(\frac{\beta V^2}{2} \cdot t^2) \\ V \cdot \sin(\frac{\beta V^2}{2} \cdot t^2) \end{vmatrix}_B \\ \overrightarrow{F}_{(t)} = V^3 \cdot \beta \cdot t \cdot \overrightarrow{N}_{(t)} &= \begin{vmatrix} -V^3 \cdot \beta \cdot t \cdot \sin(\frac{\beta V^2}{2} \cdot t^2) \\ V^3 \cdot \beta \cdot t \cdot \cos(\frac{\beta V^2}{2} \cdot t^2) \end{vmatrix}_B = V^2 \cdot \rho_{(t)} \cdot \overrightarrow{N}_{(t)}\end{aligned}$$

On peut remarquer que si l'on suit une trajectoire clothoïdale, on garde une vitesse constante et on ne ressent qu'une accélération radiale continue et qui augmente progressivement. C'est là tout l'intérêt des trajectoires sous forme de clothoïde, on peut passer d'une trajectoire rectiligne à un certain rayon de courbure progressivement.

Ces courbes sont très utilisées à la SNCF et pour le tracé de routes. Sur les autoroutes, la clothoïde est la trajectoire la plus « confortable » car le conducteur a juste à tourner le volant progressivement.

³ La **clothoïde** est une courbe transcendante plane dont la courbure est proportionnelle à l'abscisse curviligne. Elle est également appelée **spirale de Cornu**, en référence à Alfred Cornu, le physicien français qui l'a inventée. (Source Wikipédia)

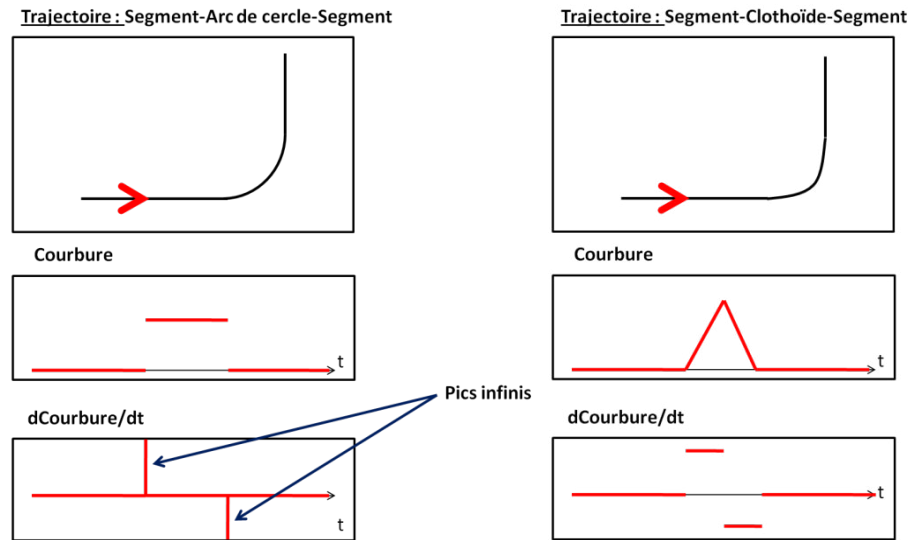


Figure 2 : Trajectoire Arc de cercle VS Clothoïde

La figure2 illustre bien les différences entre une trajectoire (à vitesse constante) formée par deux segments reliés par un arc de cercle et deux segments reliés par une clothoïde.

- Avec un arc de cercle on passe d'une courbure nulle à une courbure égale à une constante, ce qui donne un « à-coup » important sur l'accélération radiale que subit le mobile suivant la trajectoire. S'il s'agit de passagers dans un train, ils vont passer d'une accélération radiale nulle à une non nulle, ce qui n'est pas confortable et peut être dangereux (personnes qui tombent, valises qui tombent de la soute etc...).
- Avec une clothoïde, la courbure augmente progressivement puis diminue pour redevenir nulle. L'accélération radiale sera donc continue et augmentera linéairement grâce au coefficient β . Les passagers d'un train ressentiront une accélération augmentant progressivement, ce qui assure plus de confort et de sécurité.

Dans le cas de notre robot, nous souhaitons éviter d'avoir des à-coups, pour éviter tout phénomène de glissement radial non contrôlé du robot. En effet, un à-coup d'accélération à un impact sur la dynamique du robot qui va engendrer des efforts radiaux importants et le robot risque de glisser. Ces efforts vont dépendre de $\rho'_{(s)} = \alpha''_{(s)}$, or dans le cas d'un arc de cercle, $\rho'_{(s)} = \infty$ demandant des efforts infinis !(Voir partie concernant la modélisation des efforts sur le robot).

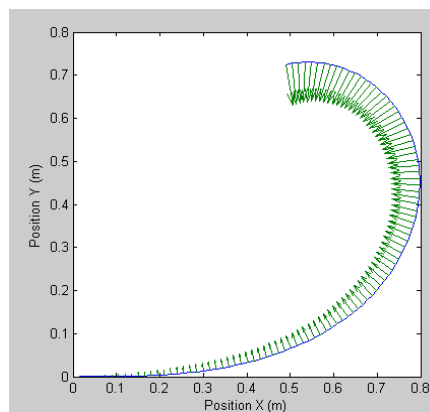


Figure 3 : Clothoïde sous Matlab

II. Modélisation mathématique

a. Mise en place des hypothèses

Nous utilisons un robot mobile avec deux roues motrices (comme 95% des robots existants).

Pourquoi ce choix ? Parce que c'est une disposition qui offre beaucoup de possibilités de mouvements simples (rotation sur soi même, avancé en ligne droite) sans avoir à utiliser beaucoup de mécanique. Nous aurions pu utiliser un robot holonome mais cependant il aurait été très complexe de créer des systèmes odométrique.

Sur la Figure 1 on peut voir la configuration en deux roues motrices du robot.

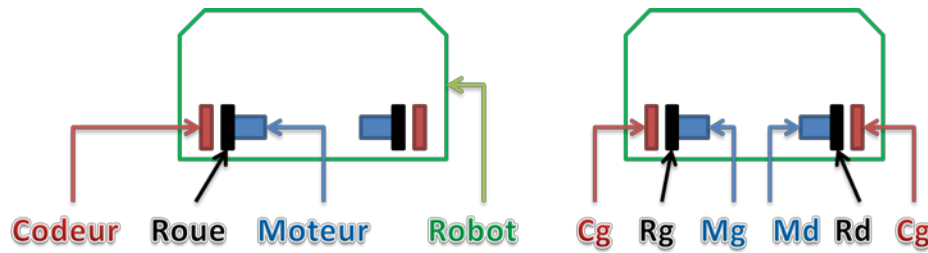


Figure 4 : Schématisation du robot

Nous ferons l'hypothèse tout le long de nos calculs que nous avons **roulement sans glissement**.

Pour la suite, nous allons énumérer ici les différentes variables en jeu :

(\vec{x}, \vec{y}) Repère galiléen lié à la table de jeu, (\vec{x}, \vec{y}) repère galiléen lié au robot

$\alpha = (\vec{y}, \vec{y})$ = Angle de positionnement du robot (rad)

r = Distance curviligne du robot parcourue (m)

D = Distance entre les deux centres des roues codeuses (m)

Φ_{roue} = Diamètre d'une roue motrice (m)

Φ_{codeur} = Diamètre d'une roue codeuse (m)

δG = Déplacement de la roue codeuse gauche pendant un instant δt (m)

$\delta \mathcal{L}_G$ = Nombre d'impulsion pour un instant δt du codeur gauche (pulse)

χ_G = Nombre d'impulsion par tour du codeur gauche (pulse/tr)

δD = Déplacement de la roue codeuse droite pendant un instant δt (m)

$\delta \mathcal{L}_D$ = Nombre d'impulsion pour un instant δt du codeur gauche (pulse)

χ_d = Nombre d'impulsion par tour du codeur droit (pulse/tr)

b. Calculs

Nous allons calculer maintenant les déplacements élémentaires qu'effectue le robot pendant un instant δt . Pour cela imaginons que δt soit très petit et nous pouvons donc supposer que le robot n'a pu se déplacer que selon une séquence linéaire (Figure 2).

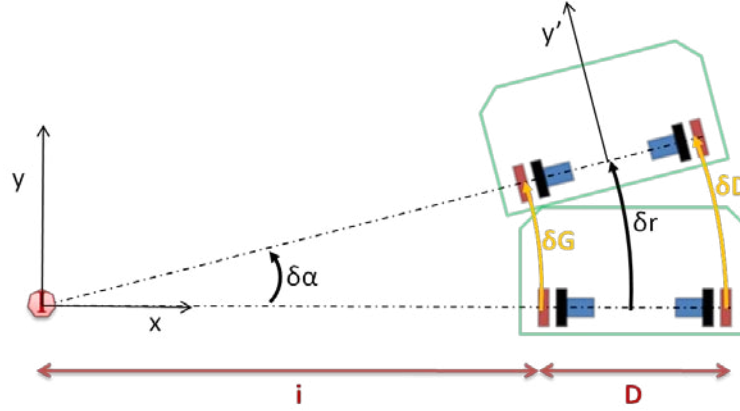


Figure 5: Formalisation mathématique

I est le centre instantané de rotation du robot.⁴

Le système odométrique nous permet de connaître seulement deux valeurs ΔG et ΔD .

Pendant le temps Δt on compte les nombres d'impulsions $\Delta \pi_G$ et $\Delta \pi_D$ sur les codeurs, on en déduit

$$\text{donc } \delta G = \frac{\Delta \pi_G}{\chi_G} * \pi * \Phi_{\text{codeur}} \text{ et } \delta D = \frac{\Delta \pi_D}{\chi_D} * \pi * \Phi_{\text{codeur}}$$

Grâce à ces deux valeurs on peut en déduire la rotation $\Delta \alpha$ ainsi que l'avancé curviligne Δr du robot :

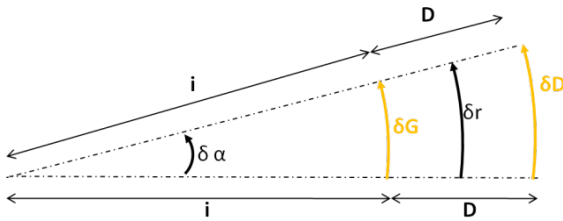


Figure 6 : trigonométrie

$$\text{On a } \delta G = \delta \alpha * i \text{ et } \delta D = \delta \alpha * (D + i)$$

$$\Rightarrow \frac{\delta D}{\delta G} = \frac{D + i}{i}$$

$$i = \frac{D \delta G}{\delta D - \delta G}$$

Remarque : Le rayon de courbure du robot s'exprime

$$R = i + \frac{D}{2} = D \left(\frac{\delta G}{\delta D - \delta G} + \frac{1}{2} \right)$$

On peut donc en déduire $\delta \alpha$

$$\delta G = \delta \alpha * i$$

$$\delta \alpha = \frac{\delta G * (D - \delta G)}{D \delta G} = \frac{\delta D - \delta G}{D} \quad \delta \alpha = \frac{\delta D - \delta G}{D}$$

Calculons à présent δr

$$\delta r = \delta \alpha * \left(i + \frac{D}{2} \right) = \frac{\delta D - \delta G}{D} * \left(\frac{D \delta G}{\delta D - \delta G} + \frac{D}{2} \right) = \delta G + \frac{\delta D - \delta G}{2} = \frac{\delta D + \delta G}{2} \quad \delta r = \frac{\delta D + \delta G}{2}$$

⁴ Le **centre instantané de rotation** (CIR) est un terme utilisé en mécanique classique et plus particulièrement en cinématique pour désigner le point autour duquel tourne un solide à un instant donné par rapport à un repère de référence.

Nous avons traité ici le cas où $\delta D > \delta G$, cependant les calculs et les résultats restent identiques si $\delta D < \delta G$.

Remarque : quand $\delta D = \delta G$ on a $i = \infty$ et aussi $R = \infty$

c. Asservissement Sigma-Delta

C'est un asservissement qui consiste à asservir le robot sur α et r .

Un asservissement sur α assure une direction contrôlée du robot tandis qu'un asservissement sur r assure une distance parcourue contrôlée. Ainsi si on asservit avec $\alpha = \text{cte}$ le robot avancera en ligne droite, et si on asservit avec $r = \text{cte}$ alors le robot tournera sur lui-même.

Ce modèle n'est pas présenté dans ce rapport intermédiaire, mais le sera lors du rapport final.

III. Adhérence

a. Etude

i. Modélisation plane

L'adhérence est un problème important pour le robot, en effet il faut éviter à tout prix que le robot dérape et ceux pour plusieurs raisons : Si on dérape, on risque de laisser une trace sur la table, ce qui nous coûterait des pénalités importantes. Mais le plus important est que si nous dérapons, nous risquons d'avoir un patinage des roues qui nous ferait faire des trajectoires « tordues » ou pire encore, si nous avons un dérapage latéral aucune des deux roues libres codeuses ne pourraient le détecter. On peut parcourir la même distance, avoir l'orientation qu'on désire mais en passant par des chemins différents qui atteignent des points différents si on a glissement. On aurait donc une position avec une erreur de position qui peut ne pas être négligeable.

Dans le cadre d'un contact ponctuel sur une surface plane et en faisant l'hypothèse d'un roulement sans glissement, on peut modéliser ce phénomène par la loi de Coulomb⁵:

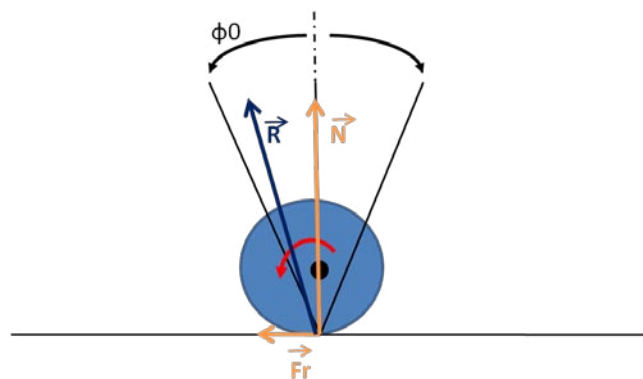


Figure 7 : Adhérence : Vue en coupe plane d'une roue

Pour rester dans le cadre d'un roulement sans glissement, il faut que la résultante \vec{R} , de l'effort normal \vec{N} du sol sur la roue et de la force d'adhérence \vec{F}_r du sol sur la roue, soit dans le cône de frottement d'adhérence. Ce cône est représenté sur la figure 4 et est caractérisé par un angle φ_0 appelé angle

⁵ En mécanique, la **loi de Coulomb**, nommée en l'honneur de Charles de Coulomb, exprime sous une forme très simplifiée l'intensité des forces de frottement qui s'exercent entre deux solides.

d'adhérence. On peut aussi utiliser le coefficient de frottement de glissement qui est défini par $f = \tan(\varphi_0)$.

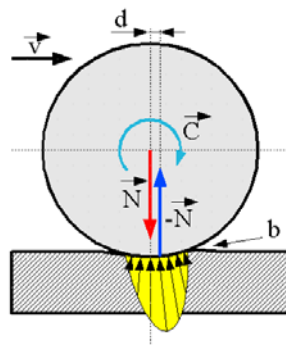
Mathématiquement on doit avoir : $\tan^{-1}\left(\frac{F_r}{N}\right) \leq \varphi_0$ ou encore $\frac{F_r}{N} \leq f$

Important : La valeur d'un coefficient de frottement ne dépend pas :

- de la nature géométrique et de l'aire de la surface de contact.
- de l'intensité de l'effort normal.

En cas de glissement, la loi de Coulomb ne s'applique pas, il se passe en réalité des phénomènes non linéaires qui dépendent du glissement défini par $g = \left| \frac{V_{robot/sol} - V_{roue/sol}}{V_{robot/sol}} \right|$ (cf. Annexe 1)

Dans cette modélisation il reste encore ce que l'on appelle la résistance au roulement. C'est un couple qui s'exerce sur la roue à cause de l'élasticité des matériaux. En effet lors de son avancement le sol (ou la roue) peut se déformer sous l'intensité des efforts en jeu. Un bourrelet se forme alors à l'avant de la roue, et à l'arrière de la roue une « marque » subsiste car le matériau constituant le sol n'a pas le temps de se relaxer complètement. On a donc une surpression des efforts à l'avant de la roue par rapport à l'arrière qui décale le point d'application de l'effort normal du sol sur la roue. Il se crée donc un couple résistant $C_r = d * N$. On appelle facteur de frottement de glissement cette valeur d , qui est une distance !



\vec{v} = vitesse du robot
 d = distance entre la résultante normale du sol et le point d'application des forces sur la roue
 b = bourrelet
 C = couple moteur appliqué à la roue

Figure 8 : Frottement de roulement (source Wikipédia)

ii. Modélisation 3 dimensions

Après avoir touché à cette modélisation plane, intéressons-nous maintenant à la modélisation 3D :

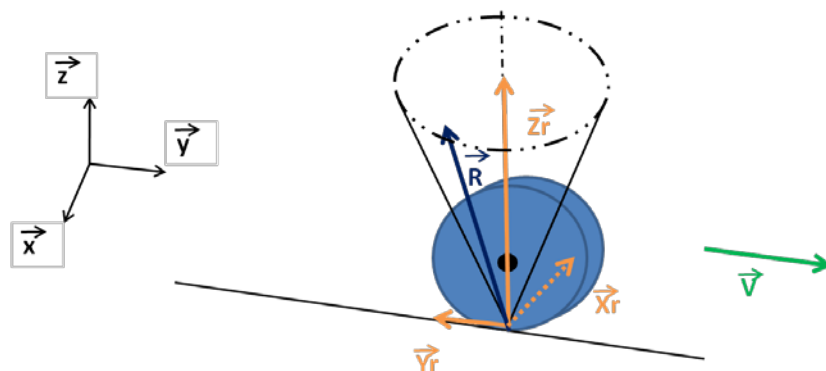


Figure 9 : Adhérence : Vue en 3 dimensions d'une roue

Cette fois-ci nous avons un vecteur $\vec{X}_r = X_r \cdot \vec{x}$ qui représente les forces de frottements transversales sur la roue (cela intervient notamment lors de suivie de trajectoires courbes par le robot).

$\vec{Y}_r = Y_r \cdot \vec{y}$ représente l'effort longitudinal des forces de frottements.

$\vec{Z}_r = Z_r \cdot \vec{z}$ représente l'effort normal de la table sur la roue.

On note alors le vecteur résultant $\vec{R} = \vec{Y}_r + \vec{X}_r + \vec{Z}_r$

Pour ne pas avoir de glissement, il ne faut toujours pas sortir du cône de frottement d'adhérence. Ce qui se traduit par : $(\vec{R}, \vec{z}) \leq \varphi_0$

On peut calculer l'angle (\vec{R}, \vec{z}) grâce au produit scalaire $\vec{R} \cdot \vec{z} = \|\vec{R}\| \|\vec{z}\| \cos(\vec{R}, \vec{z})$

$$\vec{R} \cdot \vec{z} = \vec{Y}_r \cdot \vec{z} + \vec{X}_r \cdot \vec{z} + \vec{Z}_r \cdot \vec{z} = 0 + 0 + \|\vec{Z}_r\|$$

$$\text{Finalement } \cos(\vec{R}, \vec{z}) = \frac{\|\vec{Z}_r\|}{\|\vec{R}\|} = \frac{|Z_r|}{\sqrt{Y_r^2 + X_r^2 + Z_r^2}} \leq \cos(\varphi_0)$$

/! Pour déterminer les coefficients d et φ_0 , nous procéderons à des mesures car on ne peut pas les connaître à l'avance, ce ne sont pas des données que les constructeurs donnent pour des roues de robot.

b. Mesure de coefficient d'adhérence de roues

Mise en place d'un protocole de mesure.

Cette partie n'a pas pu être traitée pour le moment elle le sera par la suite.

Sur d'autres robot $f=0.4$

IV. Etude paramétrique du robot

a. Etude générale

Hypothèses :

- On néglige le mouvement des pièces mobiles à l'intérieur du robot (\Leftrightarrow pas d'accélération de Coriolis)
- On suppose la roue uniforme, on peut alors utiliser la loi de coulomb appliquée à la roue entière.

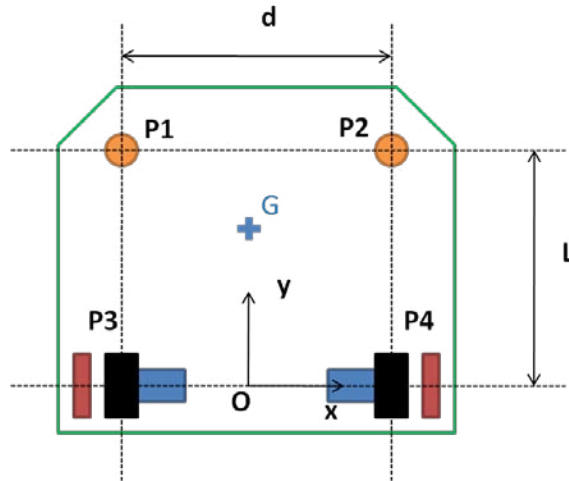


Figure 10 : Schéma du robot en vue de l'étude mécanique

Le centre de gravité se situe en $\overrightarrow{OG} = \begin{bmatrix} 0 \\ Y_g \\ Z_g \end{bmatrix}$

Graphe des liaisons :

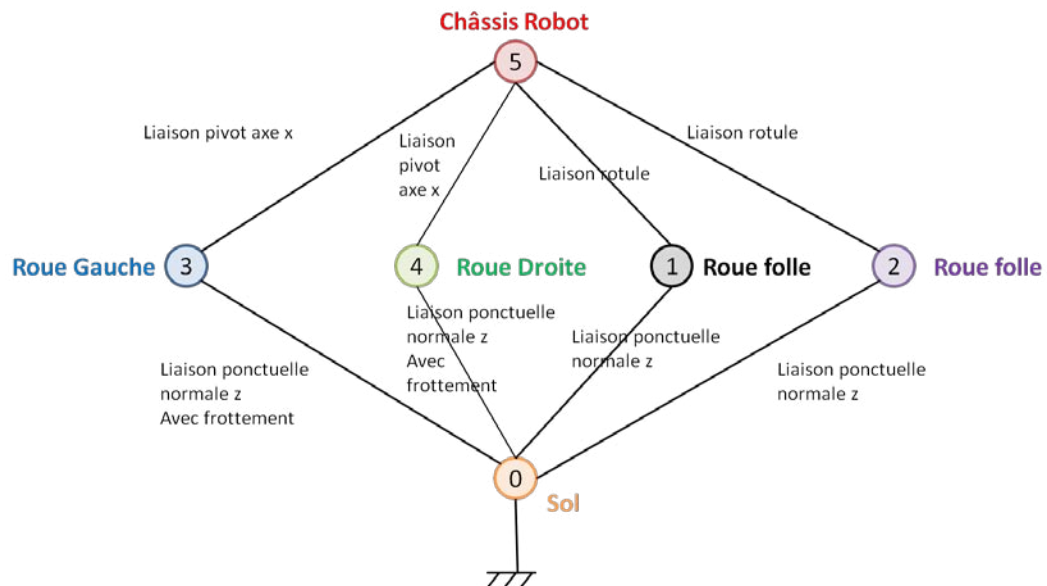


Figure 11 : Graphe des liaisons

Bilan des actions mécaniques extérieures sur le système S= {S1, S2, S3, S4, S5}

On se place dans la base $B = (\vec{x}, \vec{y}, \vec{z})$

$$\begin{aligned} \{\overrightarrow{R_{0/1}}\} &= \begin{Bmatrix} 0 & 0 \\ 0 & 0 \\ R_1 & 0 \end{Bmatrix}_B \quad \{\overrightarrow{R_{0/2}}\} = \begin{Bmatrix} 0 & 0 \\ 0 & 0 \\ R_2 & 0 \end{Bmatrix}_B \quad \{\overrightarrow{R_{0/3}}\} = \begin{Bmatrix} X_{r3} & L_3 \\ Y_{r3} & 0 \\ Z_{r3} & 0 \end{Bmatrix}_B \quad \{\overrightarrow{R_{0/4}}\} = \begin{Bmatrix} X_{r4} & L_4 \\ Y_{r4} & 0 \\ Z_{r4} & 0 \end{Bmatrix}_B \\ \{\vec{P}\} &= \begin{Bmatrix} 0 & 0 \\ 0 & 0 \\ -mg & 0 \end{Bmatrix}_B \end{aligned}$$

Remarque : Comme on l'a vu précédemment, $L_3 = d_3 * Z_{r3}$ et $L_4 = d_4 * Z_{r4}$, avec facteur de frottement de glissement. Si on choisit des pneus identiques $d_3 = d_4 = d_{fg}$

Pour appliquer le principe fondamental de la dynamique, nous devons exprimer tous ces torseurs en un même point, choisissons le point G. (On choisit ce point car cela facilite les calculs avec le moment dynamique).

$$\begin{aligned} \{\overrightarrow{R_{0/1}}\}_G &= \begin{Bmatrix} 0 & R_1 * (L - y_g) \\ 0 & R_1 * \frac{d}{2} \\ R_1 & 0 \end{Bmatrix}_B & \{\overrightarrow{R_{0/2}}\}_G &= \begin{Bmatrix} 0 & R_2 * (L - y_g) \\ 0 & -R_2 * \frac{d}{2} \\ R_2 & 0 \end{Bmatrix}_B \\ \{\overrightarrow{R_{0/3}}\}_G &= \begin{Bmatrix} X_{r3} & d_{fg} * Z_{r3} + Y_{r3} * Z_g - Z_{r3} * Y_g \\ Y_{r3} & -X_{r3} * Z_g + Z_{r3} * \frac{d}{2} \\ Z_{r3} & X_{r3} * Y_g - Y_{r3} * \frac{d}{2} \end{Bmatrix}_B \\ \{\overrightarrow{R_{0/4}}\}_G &= \begin{Bmatrix} X_{r4} & d_{fg} * Z_{r4} + Y_{r4} * Z_g - Z_{r4} * Y_g \\ Y_{r4} & -X_{r4} * Z_g - Z_{r4} * \frac{d}{2} \\ Z_{r4} & X_{r4} * Y_g + Y_{r4} * \frac{d}{2} \end{Bmatrix}_B & \{\vec{P}\}_G &= \begin{Bmatrix} 0 & 0 \\ 0 & 0 \\ -mg & 0 \end{Bmatrix}_B \end{aligned}$$

Calcul du moment cinétique :

$$\overrightarrow{\sigma_{G(S/R)}} = \overline{\overline{I_{(G,S)}}} * \overrightarrow{\Omega_{(S/R)}}$$

Avec la matrice d'inertie du robot $\overline{\overline{I_{(G,S)}}} = \begin{bmatrix} A & -F & -E \\ -F & B & -D \\ -E & -D & C \end{bmatrix}_B$ et le vecteur rotation du robot $\overrightarrow{\Omega_{(S/R)}} =$

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}_B$$

Dans le cas où le robot reste sur la table, nous avons $\omega_x = \omega_y = 0 \text{ rad/s}$, $\omega_z = \dot{\alpha}$

$$\text{Donc } \overrightarrow{\sigma_{G(S/R)}} = \begin{bmatrix} -E * \dot{\alpha} \\ -D * \dot{\alpha} \\ C * \dot{\alpha} \end{bmatrix}_B$$

On peut ainsi calculer le moment dynamique $\overrightarrow{\delta_{G(S/R)}} = \left[\frac{d\overrightarrow{G(S/R)}}{dt} \right]_R = \begin{bmatrix} -E.\ddot{\alpha} \\ -D.\ddot{\alpha} \\ C.\ddot{\alpha} \end{bmatrix}_B$

Le principe fondamental de la dynamique nous donne :

$$\{D_{(S/R)}\} = \begin{Bmatrix} m.a_x & -E.\ddot{\alpha} \\ m.a_y & -D.\ddot{\alpha} \\ m.a_z & C.\ddot{\alpha} \end{Bmatrix} = \sum \text{Torseur des actions mécaniques}$$

$$\{D_{(S/R)}\} = \{\overrightarrow{R_{0/1}}\} + \{\overrightarrow{R_{0/2}}\} + \{\overrightarrow{R_{0/3}}\} + \{\overrightarrow{R_{0/4}}\} + \{\vec{P}\}$$

$$\{D_{(S/R)}\} = \begin{Bmatrix} 0 & R_1 * (L - y_g) \\ 0 & R_1 * \frac{d}{2} \\ R_1 & 0 \end{Bmatrix}_B + \begin{Bmatrix} 0 & R_2 * (L - y_g) \\ 0 & -R_2 * \frac{d}{2} \\ R_2 & 0 \end{Bmatrix}_B$$

$$+ \begin{Bmatrix} d_{fg} * Z_{r3} + Y_{r3} * Z_g - Z_{r3} * Y_g \\ X_{r3} & -X_{r3} * Z_g + Z_{r3} * \frac{d}{2} \\ Y_{r3} & X_{r3} * Y_g - Y_{r3} * \frac{d}{2} \\ Z_{r3} & \end{Bmatrix}_B + \begin{Bmatrix} d_{fg} * Z_{r4} + Y_{r4} * Z_g - Z_{r4} * Y_g \\ X_{r4} & -X_{r4} * Z_g - Z_{r4} * \frac{d}{2} \\ Y_{r4} & X_{r4} * Y_g + Y_{r4} * \frac{d}{2} \\ Z_{r4} & \end{Bmatrix}_B = \begin{Bmatrix} 0 & 0 \\ 0 & 0 \\ -mg & 0 \end{Bmatrix}_B$$

Soit un système de 6 équations :

$$\begin{cases} m.a_x = X_{r3} + X_{r4} \\ m.a_y = Y_{r3} + Y_{r4} \\ m.a_z = R_1 + R_2 + Z_{r3} + Z_{r4} - m.g \\ -E.\ddot{\alpha} = R_1 * (L - y_g) + R_2 * (L - y_g) + d_{fg} * Z_{r3} + Y_{r3} * Z_g - Z_{r3} * Y_g + d_{fg} * Z_{r4} + Y_{r4} * Z_g - Z_{r4} * Y_g \\ -D.\ddot{\alpha} = R_1 * \frac{d}{2} - R_2 * \frac{d}{2} - X_{r3} * Z_g + Z_{r3} * \frac{d}{2} - X_{r4} * Z_g - Z_{r4} * \frac{d}{2} \\ C.\ddot{\alpha} = X_{r3} * Y_g - Y_{r3} * \frac{d}{2} + X_{r4} * Y_g + Y_{r4} * \frac{d}{2} \end{cases}$$

Le système possède plus d'inconnues (8) que d'équations(6), on ne peut pas le résoudre. Ce la vient du fait que le système est hyperstatique. Pour pouvoir résoudre, il faudrait rajouter des équations de déformations de la structure du robot ou alors rajouter des équations de trajectoire.

Cependant on peut faire quelques hypothèses simplificatrices qui vont nous permettre d'étudier le robot dans ces cas moins généraux.

On suppose connue la trajectoire que l'on veut faire au robot, c'est-à-dire que l'on connaît a_x, a_y, a_z et $\ddot{\alpha}$. $E, D, C, m, L, d, y_g, z_g, d_{fg}$ sont des constantes caractéristiques du robot supposées connues. Les inconnues à résoudre sont $X_{r3}, X_{r4}, Y_{r3}, Y_{r4}, Z_{r3}, Z_{r4}, R_1, R_2$.

b. Cas d'une trajectoire en ligne droite

Dans le cas d'une ligne droite nous avons : $a_x, a_z = 0 \text{ m.s}^{-2}$ et $\ddot{\alpha} = 0 \text{ rad.s}^{-2}$ et $X_{r3} = X_{r4} = X, Y_{r3} = Y_{r4} = Y, Z_{r3} = Z_{r4} = Z, R_1 = R_2 = R$ (Egalités qui viennent des symétries de vitesse du robot selon le plan(O, \vec{y}, \vec{z})). $d_{fg} = 0$ (On ne tient pas compte de la propriété de la roue de s'écraser pour simplifier).

On a donc :

$$\begin{cases} 0 = 2.X \\ m.a_y = 2.Y \\ 0 = 2.R + 2.Z - m.g \\ 0 = 2.R * (L - y_g) - 2.Z * Y_g + Y * Z_g \\ 0 = 0 \\ 0 = 0 \end{cases}$$

Soit :

$$\begin{cases} X = 0 \\ Y = \frac{m.a_y}{2} \\ Z = m.g.\left(1 - \frac{y_g}{L}\right) + \frac{m.a_y}{2} \cdot \frac{z_g}{L} \\ R = m.g.\frac{y_g}{L} - \frac{m.a_y}{2} \cdot \frac{z_g}{L} \end{cases}$$

i. Etude des phases du robot

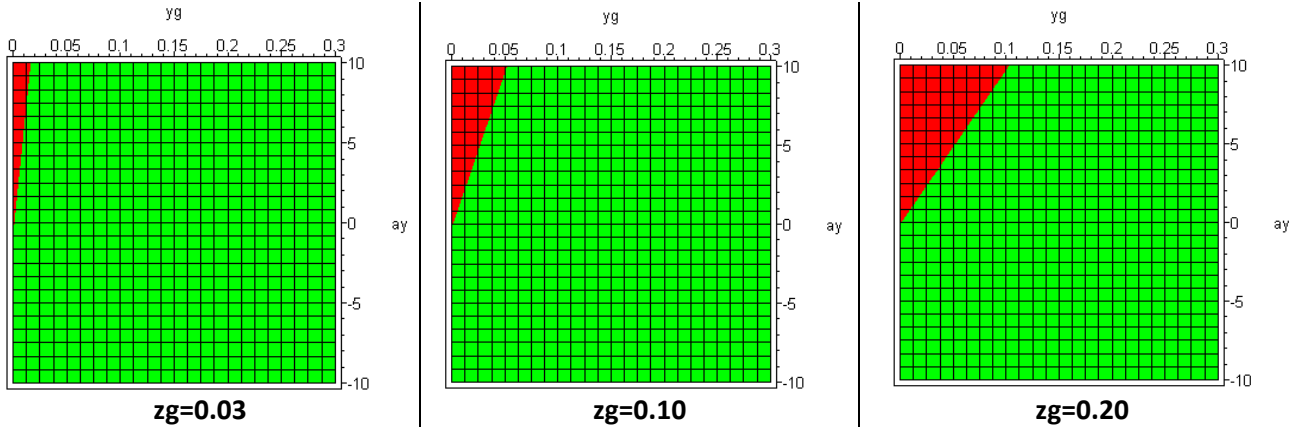
- Pour éviter tout « Wheeling » (capacité à faire une roue arrière, c'est à dire que l'avant se soulève) du robot, il faut que l'accélération $\mathbf{Y}.\vec{y}$ du robot reste inférieure à une certaine valeur. Cette valeur peut se calculer :

On cherche le point de décollement $\Leftrightarrow R = 0$

$$R = 0 = m \cdot g \cdot \frac{y_g}{L} - \frac{m \cdot a_y}{2} \cdot \frac{z_g}{L} \Rightarrow a_y = 2 \cdot g \cdot \frac{y_g}{z_g}$$

L'accélération maximale doit être de $a_y = 2 \cdot g \cdot \frac{y_g}{z_g}$

J'ai décidé de tracer les graphiques représentant en vert les zones sans décollements et en rouge les décollements.



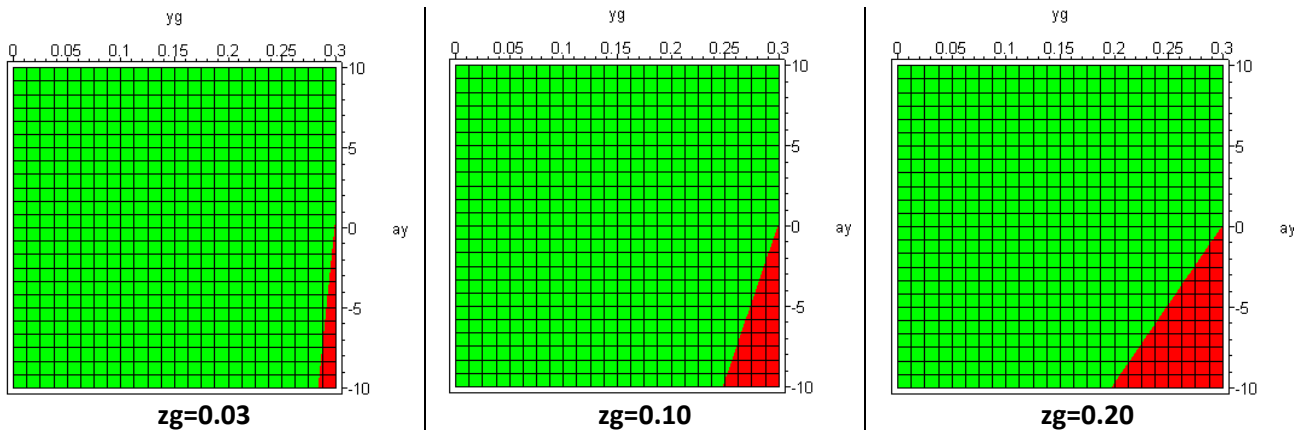
- Pour éviter tout décollement de la roue arrière au freinage on doit avoir $Z > 0$.

On cherche le point de décollement $\Leftrightarrow Z = 0$

$$Z = 0 = m \cdot g \cdot \left(1 - \frac{y_g}{L}\right) + \frac{m \cdot a_y}{2} \cdot \frac{z_g}{L} \Rightarrow a_y = -2 \cdot g \cdot \frac{L \cdot \left(1 - \frac{y_g}{L}\right)}{z_g}$$

L'accélération minimale doit être de $a_y = -2 \cdot g \cdot \frac{L \cdot \left(1 - \frac{y_g}{L}\right)}{z_g}$

J'ai décidé de tracer les graphiques représentant en vert les zones sans décollements et en rouge les décollements.



- Pour éviter tout glissement de la roue sur la table de jeu, il nous faut

$$\frac{|Z|}{\sqrt{Y^2 + X^2 + Z^2}} = \frac{\left| g \left(1 - \frac{y_g}{L} \right) + \frac{a_y z_g}{2 \cdot L} \right|}{\sqrt{\left(\frac{a_y}{2} \right)^2 + \left(g \left(1 - \frac{y_g}{L} \right) + \frac{a_y z_g}{2 \cdot L} \right)^2}} \leq \cos(\tan^{-1} f)$$

Ces deux conditions vont nous permettre de déterminer l'accélération maximale que peut avoir le robot en ligne droite.

Etudions cette condition de glissement.

Il est très difficile de voir l'impact de la position du centre de gravité sur l'accélération maximale que l'on peut obtenir. Pour cela, j'ai décidé de tracer différents graphiques afin d'avoir un résultat visible. En effet

une étude aux dérivées partielle de la fonction $f(y_g, z_g, a_y) = \frac{\left| g \left(1 - \frac{y_g}{L} \right) + \frac{a_y z_g}{2 \cdot L} \right|}{\sqrt{\left(\frac{a_y}{2} \right)^2 + \left(g \left(1 - \frac{y_g}{L} \right) + \frac{a_y z_g}{2 \cdot L} \right)^2}} = \cos(\tan^{-1} f)$

serait beaucoup trop complexe.

Une première étude pour voir l'impact de z_g sur cette fonction nous donne : (pour $y_g=0.1$, $L=0.3$, $f=0.4$)

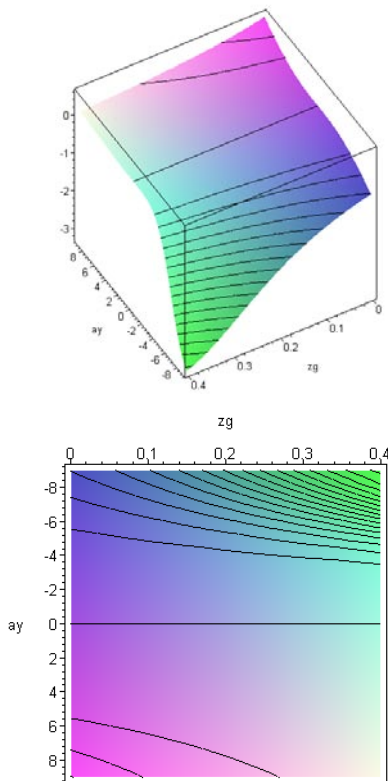


Figure 12 : Dérivée de $f(y_g, z_g, a_y)$ par rapport à z_g

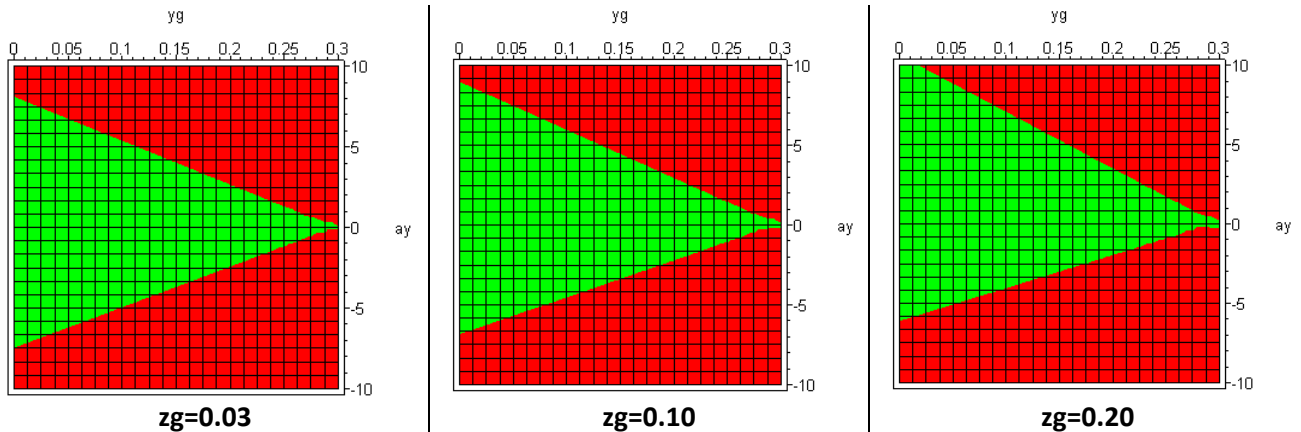
On remarque des faibles valeurs sauf dans la zone ou $a_y < -6$ et $z_g > 0.2$.

Ce qui se traduit par un impact faible de z_g dans la zone $a_y > -6$ et $z_g < 0.2$, on peut même dire que $f(y_g, z_g, a_y)$ est strictement décroissante en fonction de z_g sauf dans la zone ($a_y > 6$ et $z_g < 0.2$)

Ce graphique a été fait pour $y_g=0.1$, $L=0.3$, mais je peux conclure cette même conclusion pour d'autres valeurs de $y_g < 0.4$ et $L < 0.4$ (dimensions maximale que l'on aura pour le robot).

Cette pré-étude nous permet d'analyser l'impact de y_g et a_y en supposant que z_g a un impact peut important qui varie de manière prévisible.

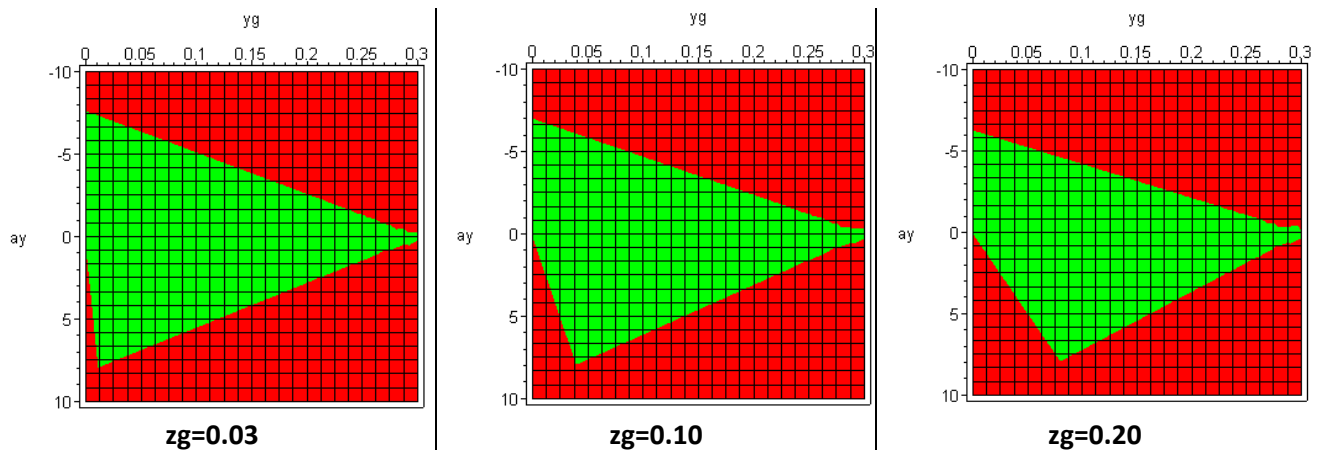
J'ai décidé de tracer les graphiques représentant les zones de fonctionnement en régime sans glissement en vert et les zones où il y a glissement en rouge.



ii. Conclusion

Comme on peut le constater, les conditions de non décollement du robot et de non-glissement ne dépendent pas de la masse du robot !

Si on regroupe les 3 conditions nous avons :



Grâce à ces graphiques on peut constater que l'on peut espérer des résultats honorables pour notre robot. Normalement le robot devrait être constitué de sorte que $y_g < 0.15$ et $z_g < 0.1$, on peut donc dire que l'accélération du robot pourra être comprise entre $-4 \text{ m/s}^2 < a_y < 5 \text{ m/s}^2$.

c. Cas d'une Clothoïde

D'après les équations sur la clothoïde nous avons :

$$\vec{a}_{(t)} = \vec{F}_{(t)} = V^3 \cdot \beta \cdot t \cdot \vec{N}_{(t)} \quad \vec{N}_{(t)} : \text{est le vecteur } \vec{x} \text{ dans le référentiel lié au robot}$$

$$\left\{ \begin{array}{l} m \cdot V^3 \cdot \beta \cdot t = X_{r3} + X_{r4} \\ 0 = Y_{r3} + Y_{r4} \\ 0 = R_1 + R_2 + Z_{r3} + Z_{r4} - m \cdot g \\ -E \cdot \beta \cdot V = R_1 * (L - y_g) + R_2 * (L - y_g) - Z_{r3} * Y_g - Z_{r4} * Y_g \\ -D \cdot \beta \cdot V = R_1 * \frac{d}{2} - R_2 * \frac{d}{2} - X_{r3} * Z_g + Z_{r3} * \frac{d}{2} - X_{r4} * Z_g - Z_{r4} * \frac{d}{2} \\ C \cdot \beta \cdot V = X_{r3} * Y_g - Y_{r3} * \frac{d}{2} + X_{r4} * Y_g + Y_{r4} * \frac{d}{2} \end{array} \right.$$

IMPOSSIBLE A RESOUDRE !! TROP D'HYPERSTATISME ! Nous avons trop d'inconnue par rapport au nombre d'équation donné par la dynamique.

Cependant je suis en contact avec M. Corn (professeur de mécanique) pour essayer de trouver une solution qui nous permette de faire des calculs.

Voici une copie du dernier mail :

« Bonjour,

Le pb que vous traitez est très intéressant.

Effectivement, tel que vous le présentez, le système méca est hyperstatique. Dans ce cas, plusieurs possibilités s'offrent à vous :

- prendre en compte les équations de déformations du châssis (ce n'est vraiment pas le plus simple !)
- revoir les liaisons afin de rendre le mécanisme isostatique (en libérant des degrés de liberté)
- reconsidérer les liaisons actuelles en intégrant la notion de jeu fonctionnel (une pivot avec du jeu n'est pas vraiment une pivot dans la réalité ...)

Le mieux, c'est peut-être qu'on regarde ça ensemble à Alès. Quand ?

Stephane CORN
Enseignant-Chercheur, Ph.D.
Mécanique des Matériaux et des Structures

ECOLE DES MINES d'ALÈS
6, Av. de Clavieres 30319 ALÈS Cedex
Tel: 04 66 78 56 29 Fax: 04 66 78 56 80
e-mail: Stephane.Corn@ema.fr »

Nous avons des pistes qui vont nous permettre de continuer les calculs.

V. Génération de trajectoire à partir d'une liste de points et d'orientations

Cette partie a été développée par nos soins lors de la mini mission mathématique.

a. Analyse de la clothoïde

L'objectif est de construire une trajectoire formée de segments et de clothoïde à partir d'une liste de points de passage.

Ci-dessous nous pouvons voir la génération d'une clothoïde « pure » (sans segment) qui passe entre deux points données A et C.

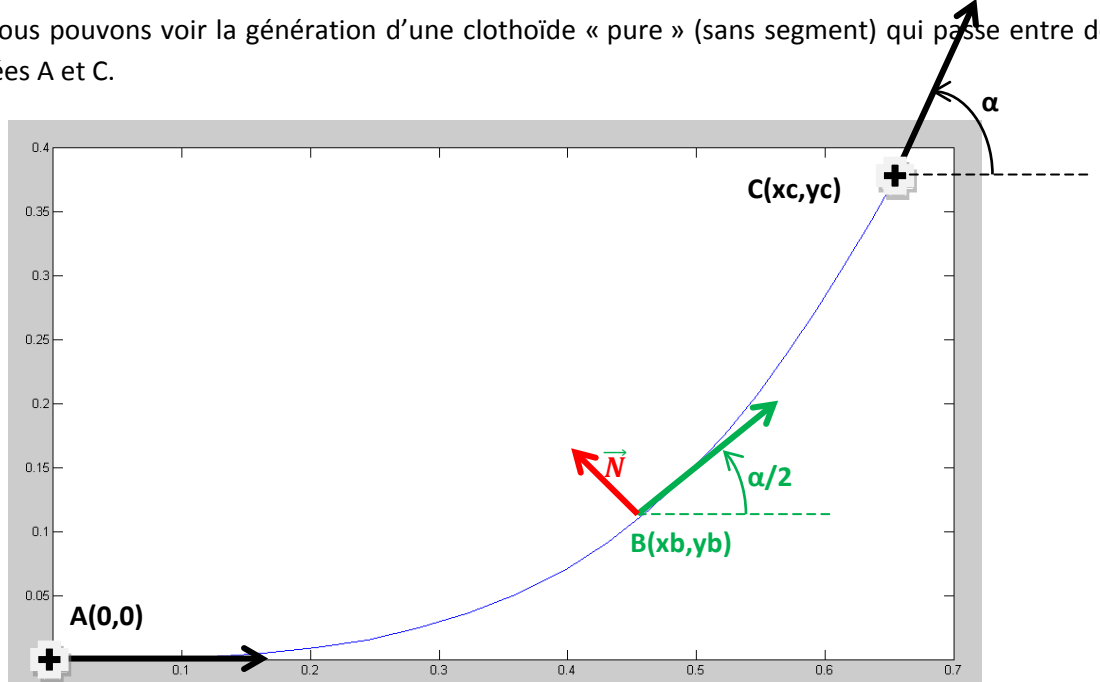


Figure 13 : Exemple de trajectoire avec deux points A et C

Pour pouvoir trouver analytiquement les propriétés de la clothoïde à obtenir pour passer par ces deux points, il nous faut simplifier les intégrales de Fresnel car elles ne permettent aucun calcul analytique.

On peut remarquer que cette trajectoire est constituée de deux morceaux de clothoïde parfaitement symétriques par rapport à (B, \vec{N}) . Ce qui explique pourquoi au point B, la tangente fait un angle de $\frac{\alpha}{2}$. Ce point B est le point où la vitesse angulaire est la plus forte (en A elle est nulle, en C aussi) cf. Figure2.

D'après ce que l'on a vu précédemment, les coordonnées du point B se calculent par :

$$x_B = \int_0^{s_B} \cos\left(\frac{\beta}{2} \cdot x^2\right) \cdot dx \quad (II.1)$$

$$y_B = \int_0^{s_B} \sin\left(\frac{\beta}{2} \cdot x^2\right) \cdot dx \quad (II.2)$$

Avec $s_B = \sqrt{\frac{\alpha}{\beta}}$ l'abscisse curviligne du point B.

$$\vec{T}_{(s_B)} = \begin{vmatrix} \cos(\frac{\alpha}{2}) \\ \sin(\frac{\alpha}{2}) \end{vmatrix}_B, \vec{N}_{(s_B)} = \begin{vmatrix} -\sin(\frac{\alpha}{2}) \\ \cos(\frac{\alpha}{2}) \end{vmatrix}_B$$

On sait que C est le symétrique de A par rapport à (B, \vec{N}) .

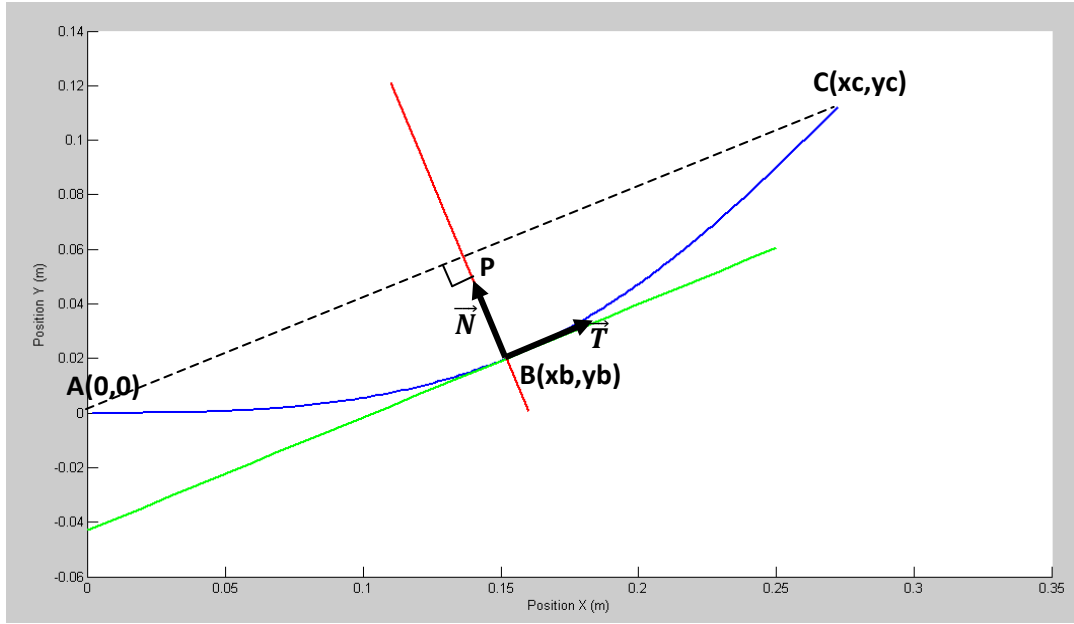


Figure 14 : Symétrie de la trajectoire par clothoïde

Posons P le projeté orthogonal de A sur (B, \vec{N}) , nous avons alors

$$\overrightarrow{AP} \cdot \vec{N} = 0 \Rightarrow -x_P \cdot \sin\left(\frac{\alpha}{2}\right) + y_P \cdot \cos\left(\frac{\alpha}{2}\right) = 0$$

Comme $P \in (B, \vec{N})$, on a :

$$y_P = \tan\left(\frac{\alpha}{2} + \frac{\pi}{2}\right) \cdot x_P + y_B - \tan\left(\frac{\alpha}{2} + \frac{\pi}{2}\right) \cdot x_B$$

Ce qui nous donne le système matriciel :

$$\begin{bmatrix} -\sin\left(\frac{\alpha}{2}\right) & \cos\left(\frac{\alpha}{2}\right) \\ -\tan\left(\frac{\alpha}{2} + \frac{\pi}{2}\right) & 1 \end{bmatrix} \cdot \begin{bmatrix} x_P \\ y_P \end{bmatrix} = \begin{bmatrix} 0 \\ y_B - \tan\left(\frac{\alpha}{2} + \frac{\pi}{2}\right) \cdot x_B \end{bmatrix} \quad (II.3)$$

Posons A et B tel quel : $A \cdot \overrightarrow{AP} = B$

Le déterminant de A est $\det(A) = -\frac{1}{\sin\left(\frac{\alpha}{2}\right)}$. L'inverse de A existe donc si $\alpha \neq 0 [2\pi]$.

Dans la suite des calculs, nous supposons que $\alpha \neq 0 [2\pi]$ car cette situation correspond au cas particulier où il n'y a pas à construire de clothoïde, c'est une ligne droite. On peut donc dire que :

$$\overrightarrow{AP} = A^{-1} \cdot B$$

D'où les coordonnées de C qui se déduisent de $\overrightarrow{AC} = 2 \cdot \overrightarrow{AP} = 2 \cdot A^{-1} \cdot B$

$$\begin{cases} x_C = 2 \cdot A^{-1} \cdot B \cdot \vec{i} \\ y_C = 2 \cdot A^{-1} \cdot B \cdot \vec{j} \end{cases} \quad (II.4)$$

Voici quelques petits exemples :

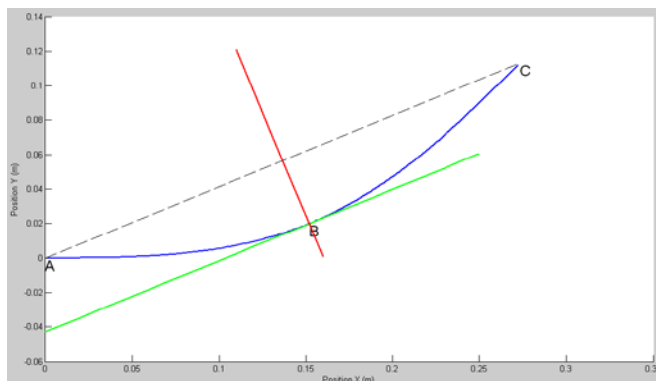


Figure 15 : $\alpha = \pi / 4$

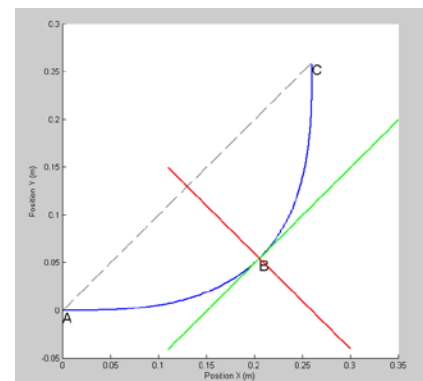


Figure 16 : $\alpha = \pi / 2$

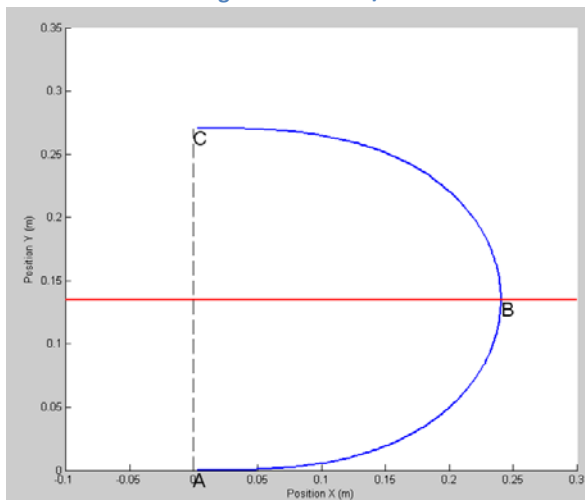


Figure 17 : $\alpha = \pi$

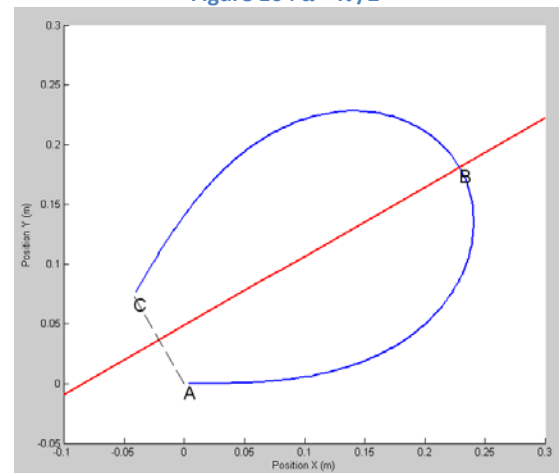


Figure 18 : $\alpha = \pi + \pi/3$

Voir Annexe 1 : Programme Matlab pour dessiner une clothoïde.

Cependant on constate que pour des angles très importants, la clothoïde fait un grand tour pour « rien ». Cela vient du fait de la commande en angle. En effet dans le cas $\alpha = \pi + 2 * \pi/3$, on observe que la clothoïde va faire une boucle, alors que la commande $\alpha = - \pi/3$ donne exactement la même chose mais en plus court

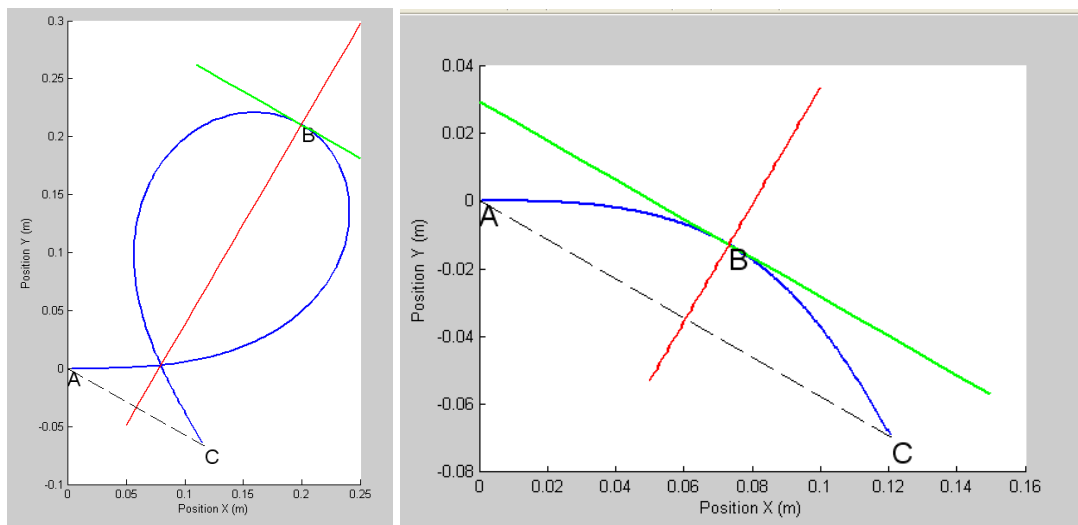


Figure 19 : $\alpha = \pi + 2 * \pi/3$ et $\alpha = \pi/3$

Dans cette première approche, nous avons réussi à fixer l'orientation finale à la sortie de la clothoïde. Nous allons maintenant étudier le comportement si nous fixons le point C, s'il existe une infinité de clothoïde qui passe par ce point ou s'il en existe une unique.

Nous avons trouvé précédemment

$$\begin{cases} x_C = 2.A^{-1}.B.\vec{i} \\ y_C = 2.A^{-1}.B.\vec{j} \end{cases}$$

Nous allons expliciter ces équations en calcul formel.

Pour cela on calcule A^{-1} :

$$A^{-1} = \frac{1}{\det(A)} \begin{bmatrix} 1 & -\cos\left(\frac{\alpha}{2}\right) \\ \tan\left(\frac{\alpha}{2} + \frac{\pi}{2}\right) & -\sin\left(\frac{\alpha}{2}\right) \end{bmatrix} = -\sin\left(\frac{\alpha}{2}\right) \begin{bmatrix} 1 & -\cos\left(\frac{\alpha}{2}\right) \\ \tan\left(\frac{\alpha}{2} + \frac{\pi}{2}\right) & -\sin\left(\frac{\alpha}{2}\right) \end{bmatrix}$$

Finalement

$$\begin{aligned} \begin{bmatrix} x_C \\ y_C \end{bmatrix} &= 2.A^{-1}.B = -2\sin\left(\frac{\alpha}{2}\right) \begin{bmatrix} 1 & -\cos\left(\frac{\alpha}{2}\right) \\ \tan\left(\frac{\alpha}{2} + \frac{\pi}{2}\right) & -\sin\left(\frac{\alpha}{2}\right) \end{bmatrix} \begin{bmatrix} y_B - \tan\left(\frac{\alpha}{2} + \frac{\pi}{2}\right).x_B \\ 0 \end{bmatrix} \\ &= -2\sin\left(\frac{\alpha}{2}\right) \begin{bmatrix} -\cos\left(\frac{\alpha}{2}\right)(y_B - \tan\left(\frac{\alpha}{2} + \frac{\pi}{2}\right).x_B) \\ -\sin\left(\frac{\alpha}{2}\right)(y_B - \tan\left(\frac{\alpha}{2} + \frac{\pi}{2}\right).x_B) \end{bmatrix} \\ \begin{bmatrix} x_C \\ y_C \end{bmatrix} &= -2\sin\left(\frac{\alpha}{2}\right) \begin{bmatrix} -\cos\left(\frac{\alpha}{2}\right) \left(\int_0^{\sqrt{\frac{\alpha}{\beta}}} \sin\left(\frac{\beta}{2}.x^2\right).dx - \tan\left(\frac{\alpha}{2} + \frac{\pi}{2}\right) \cdot \int_0^{\sqrt{\frac{\alpha}{\beta}}} \cos\left(\frac{\beta}{2}.x^2\right).dx \right) \\ -\sin\left(\frac{\alpha}{2}\right) \left(\int_0^{\sqrt{\frac{\alpha}{\beta}}} \sin\left(\frac{\beta}{2}.x^2\right).dx - \tan\left(\frac{\alpha}{2} + \frac{\pi}{2}\right) \cdot \int_0^{\sqrt{\frac{\alpha}{\beta}}} \cos\left(\frac{\beta}{2}.x^2\right).dx \right) \end{bmatrix} \quad (II.5) \end{aligned}$$

Comme on peut le voir ici, les choses se compliquent, il faut calculer une intégrale de Fresnel ce qui est n'est pas réalisable analytiquement.

b. Approximation des intégrales de Fresnel

Nous allons effectuer un développement limité sur le sin :

$$\sin(x) = x - \frac{x^3}{3!} + o(x^3)$$

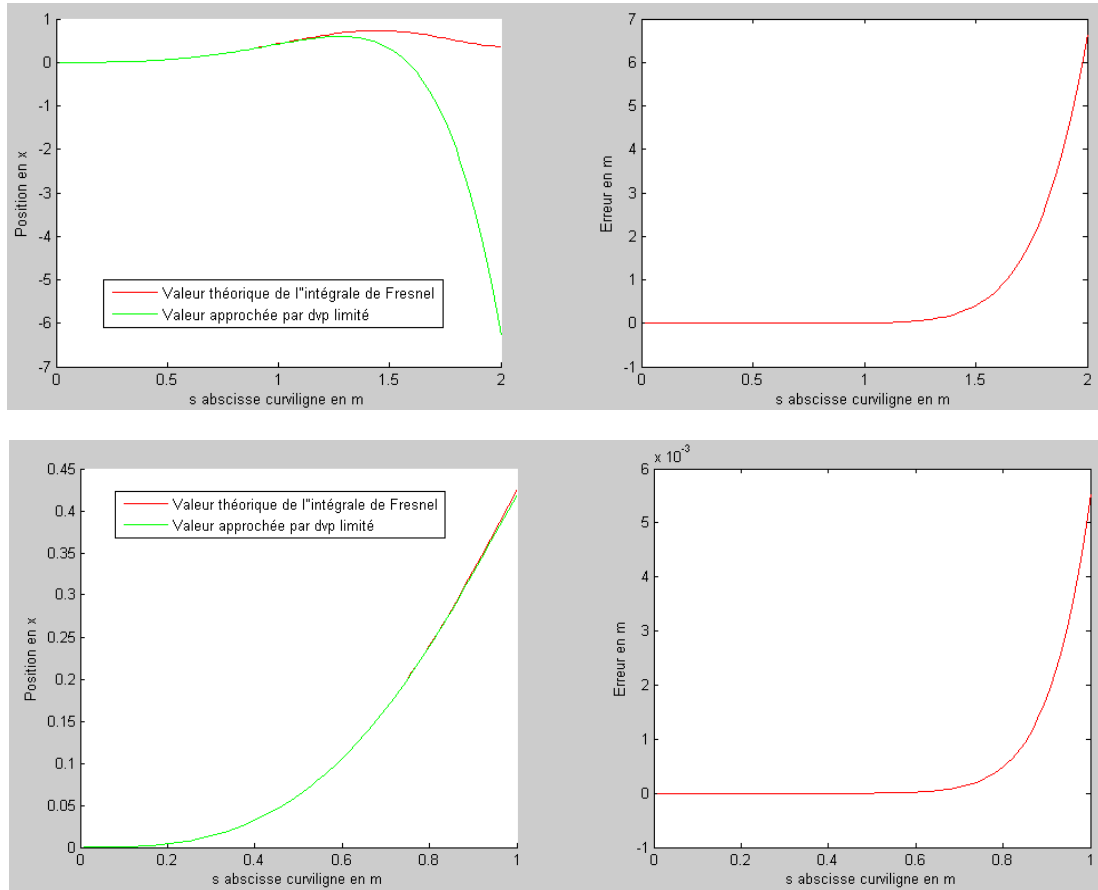
On remplace dans les équations :

$$\int_0^{\sqrt{\frac{\alpha}{\beta}}} \sin\left(\frac{\beta}{2} \cdot x^2\right) \cdot dx = \int_0^{\sqrt{\frac{\alpha}{\beta}}} \left(\frac{\beta}{2} \cdot x^2 - \frac{\left(\frac{\beta}{2}\right)^3 \cdot x^6}{3!} + o(x^3)\right) \cdot dx \approx \left[\frac{\beta}{2} \cdot \frac{x^3}{3} - \frac{\left(\frac{\beta}{2}\right)^3}{3! \cdot 7} \cdot x^7\right]_0^{\sqrt{\frac{\alpha}{\beta}}} = \frac{1}{2} \cdot \frac{\alpha \sqrt{\alpha}}{3 \cdot \sqrt{\beta}} - \left(\frac{1}{2}\right)^3 \cdot \frac{\alpha^3 \sqrt{\alpha}}{3! \cdot 7 \sqrt{\beta}} \quad (II.6)$$

Il faut maintenant savoir si l'approximation que nous faisons n'engendre pas une erreur trop importante. C'est pourquoi nous allons étudier sous Matlab la fonction suivante :

$$\int_0^s \sin\left(\frac{\beta}{2} \cdot x^2\right) \cdot dx \approx \int_0^s \left(\frac{\beta}{2} x^2 - \frac{\left(\frac{\beta}{2}\right)^3 x^6}{3!}\right) \cdot dx = \left[\frac{\beta}{2} \cdot \frac{x^3}{3} - \frac{\left(\frac{\beta}{2}\right)^3}{3! \cdot 7} x^7\right]_0^s = \frac{\beta}{2} \cdot \frac{s^3}{3} - \left(\frac{\beta}{2}\right)^3 \cdot \frac{s^7}{3! \cdot 7}$$

Nous avons :



Notre approximation polynomiale nous permet d'avoir un calcul de l'intégrale de Fresnel en sin correct tant

que $s < 0.8 \Leftrightarrow \sqrt{\frac{\alpha}{\beta}} < 0.8$

Nous allons donc effectuer un développement limité sur le cos:

$$\cos(x) = 1 - \frac{x^2}{2} + o(x^2)$$

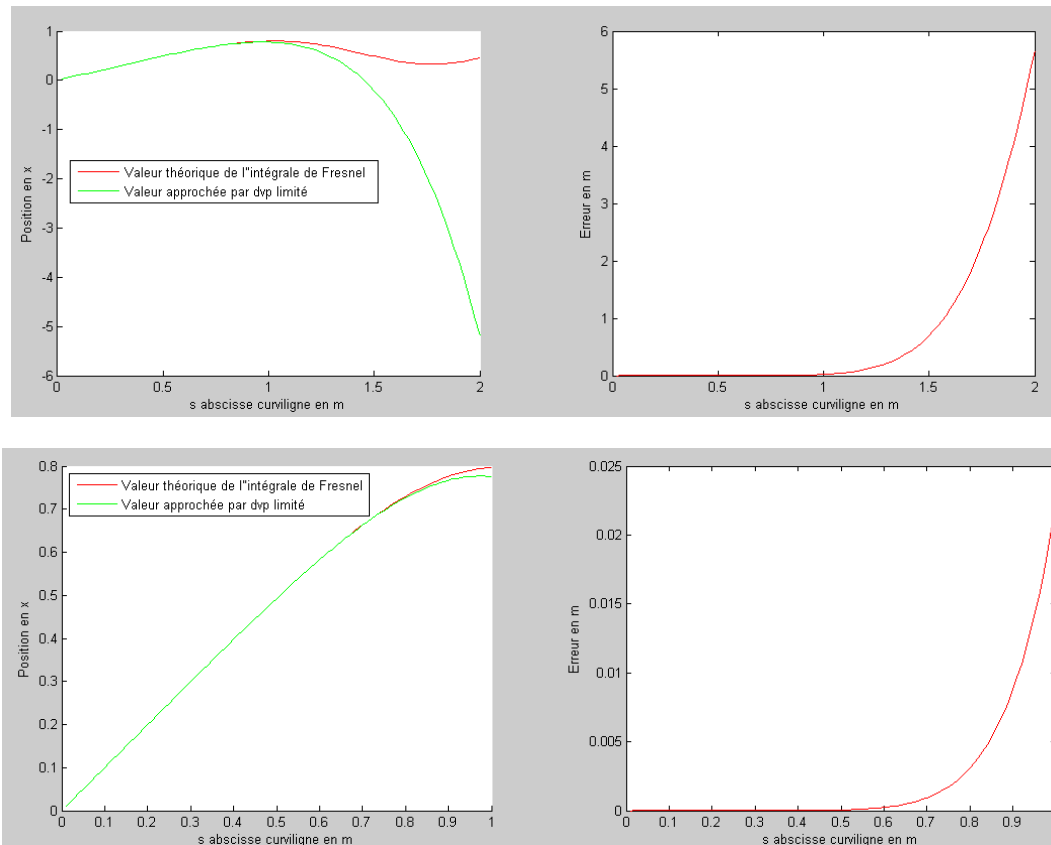
On remplace dans les équations :

$$\int_0^{\sqrt{\frac{\alpha}{\beta}}} \cos\left(\frac{\beta}{2} \cdot x^2\right) \cdot dx = \int_0^{\sqrt{\frac{\alpha}{\beta}}} \left(1 - \frac{\left(\frac{\beta}{2}\right)^2 \cdot x^4}{2} + o(x^4)\right) \cdot dx \approx \left[x - \left(\frac{\beta}{2}\right)^2 \cdot \frac{x^5}{2 \cdot 5}\right]_0^{\sqrt{\frac{\alpha}{\beta}}} = \sqrt{\frac{\alpha}{\beta}} - \left(\frac{1}{2}\right)^2 \cdot \frac{\alpha^2 \sqrt{\alpha}}{2 \cdot 5 \sqrt{\beta}} \quad (11.7)$$

Il faut maintenant savoir si l'approximation que nous faisons n'engendre pas une erreur trop importante. C'est pourquoi nous allons étudier sous Matlab la fonction suivante :

$$\int_0^s \cos\left(\frac{\beta}{2} \cdot x^2\right) \cdot dx = \int_0^s \left(1 - \frac{\left(\frac{\beta}{2}\right)^2 \cdot x^4}{2} + o(x^4)\right) \cdot dx \approx \left[x - \left(\frac{\beta}{2}\right)^2 \cdot \frac{x^5}{2 \cdot 5}\right]_0^s = s - \left(\frac{\beta}{2}\right)^2 \cdot \frac{s^5}{2 \cdot 5}$$

Nous avons :



Notre approximation polynomiale nous permet d'avoir un calcul de l'intégrale de Fresnel en sin correct tant

que $s < 0.8 \Leftrightarrow \sqrt{\frac{\alpha}{\beta}} < 0.8$

Remarque : Il va falloir trouver une autre méthode d'approximation car cette méthode amène à des puissances rapidement élevé, ce qui explique une erreur très vite croissante.

Nous avons donc après approximation :

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = -2 \sin\left(\frac{\alpha}{2}\right) \begin{bmatrix} -\cos\left(\frac{\alpha}{2}\right) \left(\frac{1}{2} \cdot \frac{\alpha\sqrt{\alpha}}{3\sqrt{\beta}} - \left(\frac{1}{2}\right)^3 \cdot \frac{\alpha^3\sqrt{\alpha}}{3! \cdot 7\sqrt{\beta}} - \tan\left(\frac{\alpha}{2} + \frac{\pi}{2}\right) \cdot \left(\sqrt{\frac{\alpha}{\beta}} - \left(\frac{1}{2}\right)^2 \cdot \frac{\alpha^2\sqrt{\alpha}}{2 \cdot 5\sqrt{\beta}} \right) \right) \\ -\sin\left(\frac{\alpha}{2}\right) \left(\frac{1}{2} \cdot \frac{\alpha\sqrt{\alpha}}{3\sqrt{\beta}} - \left(\frac{1}{2}\right)^3 \cdot \frac{\alpha^3\sqrt{\alpha}}{3! \cdot 7\sqrt{\beta}} - \tan\left(\frac{\alpha}{2} + \frac{\pi}{2}\right) \cdot \left(\sqrt{\frac{\alpha}{\beta}} - \left(\frac{1}{2}\right)^2 \cdot \frac{\alpha^2\sqrt{\alpha}}{2 \cdot 5\sqrt{\beta}} \right) \right) \end{bmatrix}$$

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} \sqrt{\beta} = 2 \sin\left(\frac{\alpha}{2}\right) \cdot \sqrt{\alpha} \left(\frac{1}{\tan\left(\frac{\alpha}{2}\right)} + \frac{\alpha}{6} - \frac{\alpha^2}{40 \cdot \tan\left(\frac{\alpha}{2}\right)} - \frac{\alpha^3}{336} \right) \begin{bmatrix} \cos\left(\frac{\alpha}{2}\right) \\ \sin\left(\frac{\alpha}{2}\right) \end{bmatrix}$$

Donc

$$\beta = \left(\frac{\sin\left(\frac{\alpha}{2}\right) \cdot \sqrt{\alpha} \left(\frac{1}{\tan\left(\frac{\alpha}{2}\right)} + \frac{\alpha}{6} - \frac{\alpha^2}{40 \cdot \tan\left(\frac{\alpha}{2}\right)} - \frac{\alpha^3}{336} \right) \cos\left(\frac{\alpha}{2}\right)}{x_c} \right)^2 = \left(\frac{\sin\left(\frac{\alpha}{2}\right) \cdot \sqrt{\alpha} \left(\frac{1}{\tan\left(\frac{\alpha}{2}\right)} + \frac{\alpha}{6} - \frac{\alpha^2}{40 \cdot \tan\left(\frac{\alpha}{2}\right)} - \frac{\alpha^3}{336} \right) \sin\left(\frac{\alpha}{2}\right)}{y_c} \right)^2 \quad (II.8)$$

Ainsi en définissant l'angle α de la trajectoire du drone, ainsi que le déplacement en x et y à effectuer, nous sommes en mesure de déterminer la clothoïde correspondante.

On remarquera que l'on peut trouver une solution seulement si $y_c = \tan\left(\frac{\alpha}{2}\right) \cdot x_c$ (II.9)

Pour exemple:

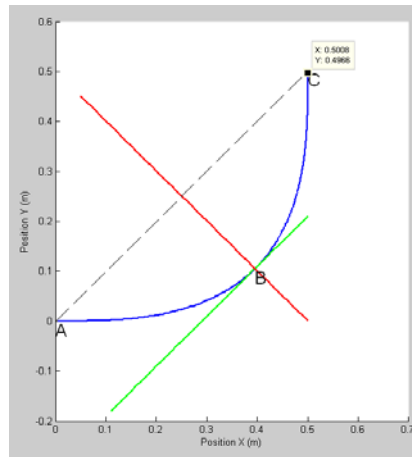


Figure 20 : $X_c=0.5$; $Y_c=0.5$; $\alpha=\pi/2$;

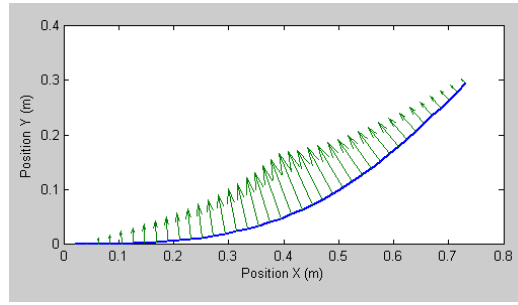


Figure 21 : Visualisation accélération clothoïde

c. Clothoïde en S

Pour la génération de trajectoire, il nous manque un élément essentiel : la clothoïde en S. En effet pour le moment aucun élément ne permet de relier deux droites parallèles. Seule une clothoïde en S peut le faire. Il s'agit du cas particulier lorsque l'on fait une symétrie par rapport au point de départ, on a :

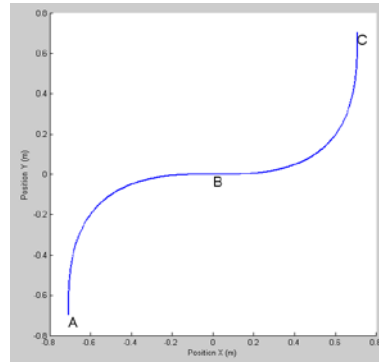


Figure 22 : Clothoïde en S

Cette courbe est donnée par

$$\vec{F}_{(s)} = \begin{cases} \int_0^s \cos(\frac{\beta}{2} \cdot x^2) \cdot dx \\ \int_0^s \sin(\frac{\beta}{2} \cdot x^2) \cdot dx \end{cases} \text{ avec } -\sqrt{\frac{\pi}{\beta}} \leq s \leq \sqrt{\frac{\pi}{\beta}} \quad (\text{II.10})$$

Comme on a par symétrie $s_a = -s_c$ on a bien $\alpha_{(s_a)} = \alpha_{(s_c)}$ donc les points A et C sont bien la liaison entre deux droites parallèles. Calculons l'espacement entre ces deux droites parallèles.

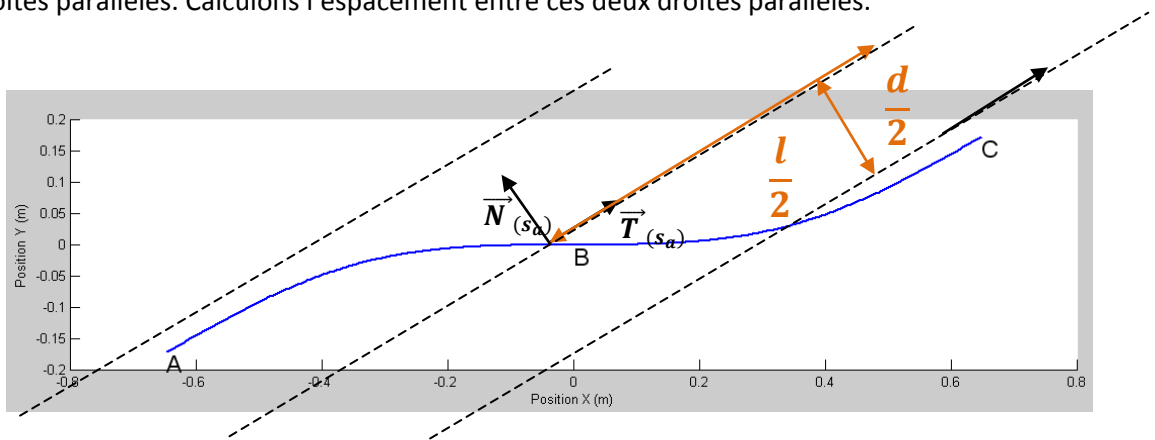


Figure 23 : para métrisation clothoïde en S

Posons le nouveau repère $R_2(B, \vec{T}_{(s_c)}, \vec{N}_{(s_c)})$, cela revient à une rotation du repère initial de $\alpha_{(s_c)}$.

Posons $\begin{bmatrix} x_{C2} \\ y_{C2} \end{bmatrix}$ les coordonnées de C dans R_2 .

$$\begin{bmatrix} x_{C2} \\ y_{C2} \end{bmatrix} = \begin{bmatrix} \frac{l}{2} \\ \frac{d}{2} \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} x_C \\ y_C \end{bmatrix}$$

En utilisant les approximations on a :

$$\begin{bmatrix} l \\ d \end{bmatrix} = 4 \sin\left(\frac{\alpha}{2}\right) \cdot \sqrt{\frac{\alpha}{\beta}} \left(\frac{1}{\tan\left(\frac{\alpha}{2}\right)} + \frac{\alpha}{6} - \frac{\alpha^2}{40 \cdot \tan\left(\frac{\alpha}{2}\right)} - \frac{\alpha^3}{336} \right) \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} \cos\left(\frac{\alpha}{2}\right) \\ \sin\left(\frac{\alpha}{2}\right) \end{bmatrix}$$

On peut trouver β pour un d et un l donné à condition que

$$l \left(-\sin(\alpha) \cdot \cos\left(\frac{\alpha}{2}\right) + \cos(\alpha) \cdot \sin\left(\frac{\alpha}{2}\right) \right) = d \left(\cos(\alpha) \cdot \cos\left(\frac{\alpha}{2}\right) + \sin(\alpha) \cdot \sin\left(\frac{\alpha}{2}\right) \right)$$

$$d = -\tan\left(\frac{\alpha}{2}\right) l \Rightarrow \boxed{\alpha = 2 * \text{atan}\left(-\frac{d}{l}\right)} \quad (II.11)$$

$$\boxed{\beta = \left(4 \sin\left(\frac{\alpha}{2}\right) \cdot \sqrt{\alpha} \left(\frac{1}{\tan\left(\frac{\alpha}{2}\right)} + \frac{\alpha}{6} - \frac{\alpha^2}{40 \cdot \tan\left(\frac{\alpha}{2}\right)} - \frac{\alpha^3}{336} \right) \cdot \cos\left(\frac{\alpha}{2}\right) \cdot \frac{1}{l} \right)^2} \quad (II.12)$$

Pour conclure, voici une portion de clothoïde en S :

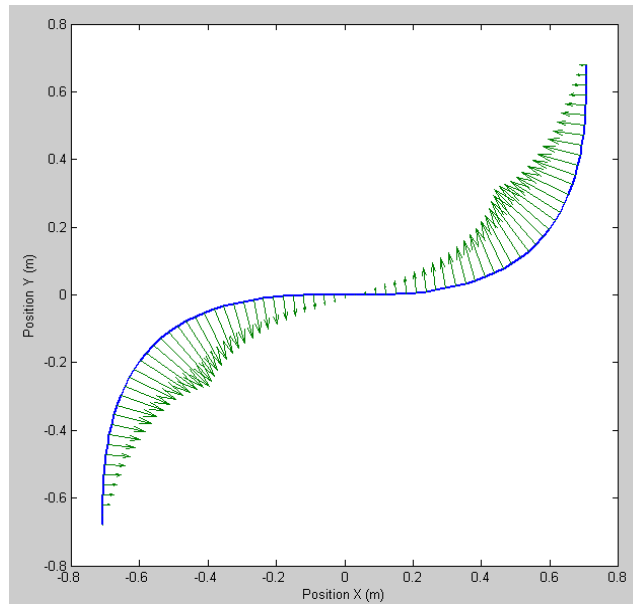


Figure 24 : Visualisation accélération clothoïde en S

Nous pouvons désormais nous intéresser à l'algorithme de génération de trajectoire.

d. Algorithme de génération de trajectoire par "blocs"

i. Principes

La génération de trajectoire doit se faire à partir d'un ensemble E de points de passages à travers lequel le drone doit obligatoirement passer.

N'ayant trouvé aucune méthode de génération de trajectoire à partir de clothoïde, nous avons du écrire de toute main celui-ci.

Principe : enchainements de clothoïdes et de lignes droites (et de clothoïdes en S si le drone change de sens de rotation).

Le tout en essayant au maximum d'utiliser le potentiel maximum du mobile. C'est-à-dire il faut essayer le plus que possible que les clothoïdes atteignent l'accélération maximale que peut supporter le mobile.

Cette accélération maximale notée $a_{tmax} = V^3 \cdot \beta_{max} \cdot t_{AB} = V^2 \cdot \beta_{max} \cdot s_B = V^2 \cdot \beta_{max} \cdot \sqrt{\frac{\alpha}{\beta_{max}}}$

D'où

$$\boxed{\beta_{max} = \frac{a_{tmax}^2}{\alpha \cdot V^4}} \quad (II.13)$$

Avec a_{tmax} : l'accélération radiale maximale admissible

V : la vitesse du mobile

Remarque : Nous n'avons pas eu le temps de terminer cet algorithme développé entièrement par nos soins.

Aperçu :

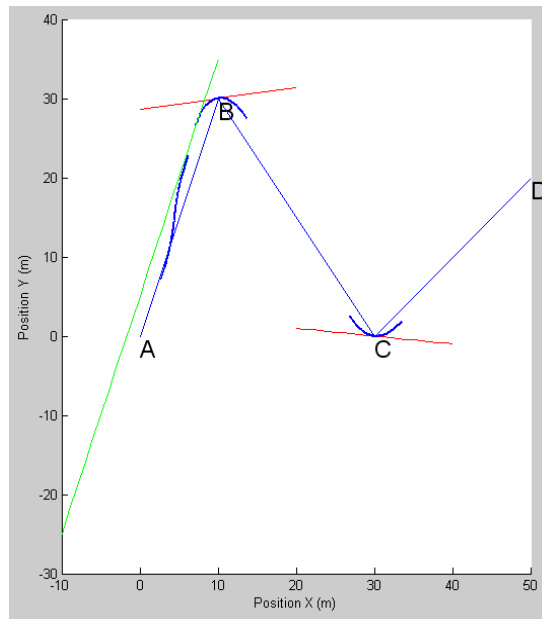


Figure 25 : trajectoire générée par "blocs"

VI. Génération de trajectoire à partir d'une liste de points et d'orientations par méthode itérative

Nous verrons dans ce chapitre un autre moyen de générer une trajectoire clothoïdale qui se fait sans utiliser de segment droit. Dans cet algorithme, il s'agit de relier 2 points orientés seulement avec 3 clothoïdes. Cet algorithme est basé sur la publication *Path generation for robot vehicles using Clothoid segments* 1990 [1].

Revenons sur la principale caractéristique de la clothoïde : son rayon de courbure proportionnel à l'abscisse curviligne : $\rho_{(s)} = \alpha'_{(s)} = \beta \cdot s + \rho_i$ (III.1)

Où β est le taux de courbure et ρ_i la courbure initiale.

Nous avons donc par intégration :

$$\alpha_{(s)} = \int_0^s \alpha'_{(\xi)} d\xi = \frac{\beta}{2} \cdot s^2 + \rho_i \cdot s + \alpha_i \quad (III.2)$$

Avec α_i l'orientation initiale de la courbe par rapport à \vec{i}

On en déduit donc les coordonnées de chaque point en fonction de son abscisse curviligne :

$$x_{(s)} = x_i + \int_0^s \cos(\alpha_{(\xi)}) \cdot d\xi = x_i + \int_0^s \cos\left(\frac{\beta}{2} \cdot \xi^2 + \rho_i \cdot \xi + \alpha_i\right) \cdot d\xi \quad (III.3)$$

$$y_{(s)} = y_i + \int_0^s \sin(\alpha_{(\xi)}) \cdot d\xi = y_i + \int_0^s \sin\left(\frac{\beta}{2} \cdot \xi^2 + \rho_i \cdot \xi + \alpha_i\right) \cdot d\xi \quad (III.4)$$

Où x_i et y_i sont les coordonnées du point de départ.

Il a été prouvé qu'avec un ensemble de 3 clothoïdes on est toujours capable de relier deux points orientés⁶.

Supposons que nous avons 3 clothoïdes qui se succèdent alors nous avons les relations de continuités :

$$x_{1(sf)} = x_{2(si)} , y_{1(sf)} = y_{2(si)} , \alpha_{1(sf)} = \alpha_{2(si)} \quad (III.5)$$

$$x_{2(sf)} = x_{3(si)} , y_{2(sf)} = y_{3(si)} , \alpha_{2(sf)} = \alpha_{3(si)} \quad (III.6)$$

Avec i en indice le numéro de la clothoïde correspondante et si et sf indiquant respectivement le début et la fin de l'abscisse curviligne.

Si l'on impose que les 3 clothoïdes possèdent le même taux de courbure et si on appelle s_1, s_2 et s_3 les longueurs curvilignes de chacune, alors le couple $\{\beta, s_1, s_2, s_3\}$ est suffisant pour caractériser la trajectoire clothoïdale entre deux points orientés.

Pour trouver ce couple il faut résoudre le système suivant grâce à (1), (2), (3), (4), (5), (6) :

$$\rho_f = \rho_i + \beta \cdot (s_1 - s_2 + s_3) \quad (III.7)$$

⁶ Ce sont des points auxquels on associe un vecteur unitaire indiquant une direction pour le sens de passage du mobile à ce point.

$$\alpha_f = \alpha_i + \rho_i \cdot (s_1 + s_2 + s_3) + \beta \cdot (s_1 \cdot s_2 - s_2 \cdot s_3 + s_1 \cdot s_3) + \frac{\beta}{2} \cdot (s_1^2 - s_2^2 + s_3^2) \quad (\text{III.8})$$

$$x_f = x_i + \int_0^{s_1} \cos(\alpha_{1(\xi)}) \cdot d\xi + \int_0^{s_2} \cos(\alpha_{2(\xi)}) \cdot d\xi + \int_0^{s_3} \cos(\alpha_{3(\xi)}) \cdot d\xi \quad (\text{III.9})$$

$$y_f = y_i + \int_0^{s_1} \sin(\alpha_{1(\xi)}) \cdot d\xi + \int_0^{s_2} \sin(\alpha_{2(\xi)}) \cdot d\xi + \int_0^{s_3} \sin(\alpha_{3(\xi)}) \cdot d\xi \quad (\text{III.10})$$

Où

$$\alpha_{1(\xi)} = \alpha_i + \rho_i \cdot \xi + \frac{\beta}{2} \cdot \xi^2$$

$$\alpha_{2(\xi)} = \alpha_i + \rho_i \cdot (s_1 + \xi) + \beta \cdot s_1 \cdot \xi + \frac{\beta}{2} \cdot (s_1^2 - \xi^2)$$

$$\alpha_{3(\xi)} = \alpha_i + \rho_i \cdot (s_1 + s_2 + \xi) + \beta \cdot (s_1 \cdot s_2 - s_2 \cdot \xi + s_1 \cdot \xi) + \frac{\beta}{2} \cdot (s_1^2 - s_2^2 + \xi^2)$$

Du fait des intégrales de Fresnel, il n'existe aucun moyen mathématique pour résoudre ce système.

Cependant il existe une méthode itérative qui permet de converger vers une solution. Voici le schéma :

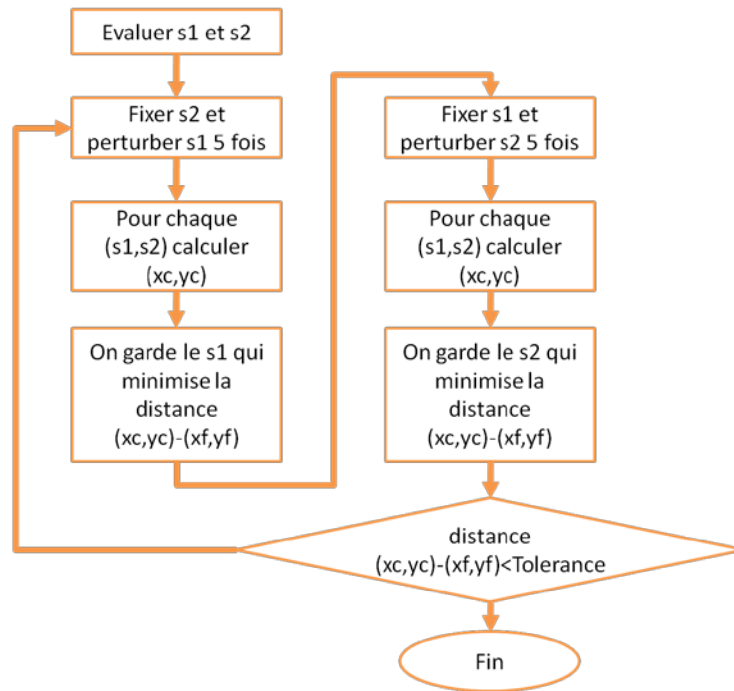


Figure 26 : Algorithme de convergence

On évalue généralement $s_1 = 1/9$ de la distance entre le point de départ et le point final et $s_2 = 1/9$ de la distance entre le point de départ et le point final.

Remarque : dans l'algorithme il faut imposer que s_1, s_2 , et s_3 soient positifs sinon la trajectoire obtenue aura des points de rebroussement.

Nous avons généré cet algorithme sous Matlab pour relier les points A (0, 0, 0°, $\rho_A = 0$) et B (14, 5, 90°, $\rho_B = 0$) avec une tolérance de 10cm :

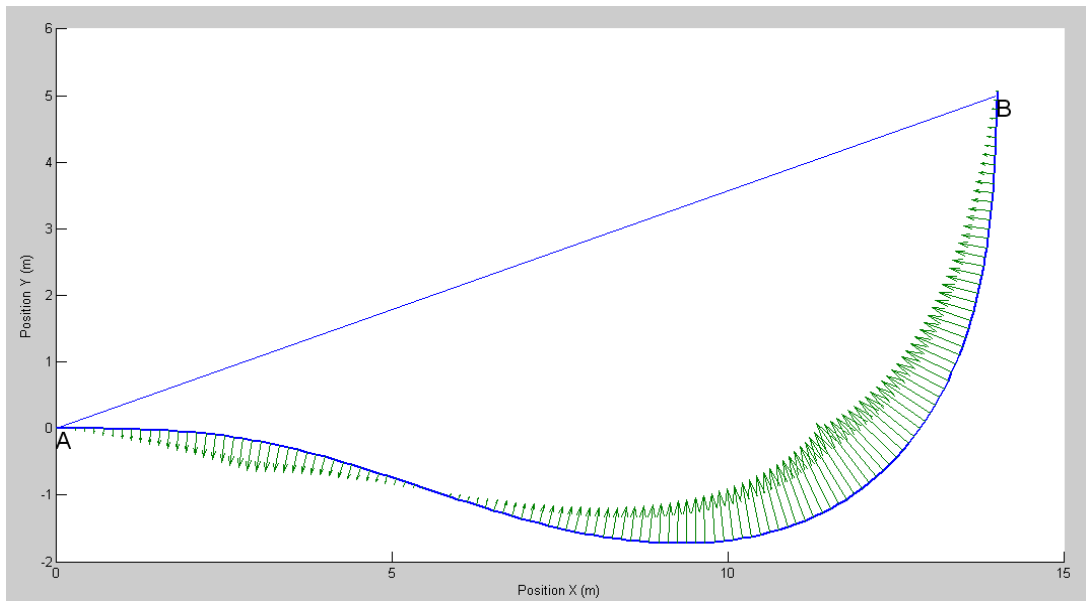


Figure 27 : Clothoïde $s_1=2.8118, s_2=9.4954, s_3=6.6836, k=-0.0427$

On peut constater que cette méthode donne de très bons résultats. Cependant cet algorithme n'est pas très performant en termes de temps et de nombre d'itérations avant d'obtenir un résultat.

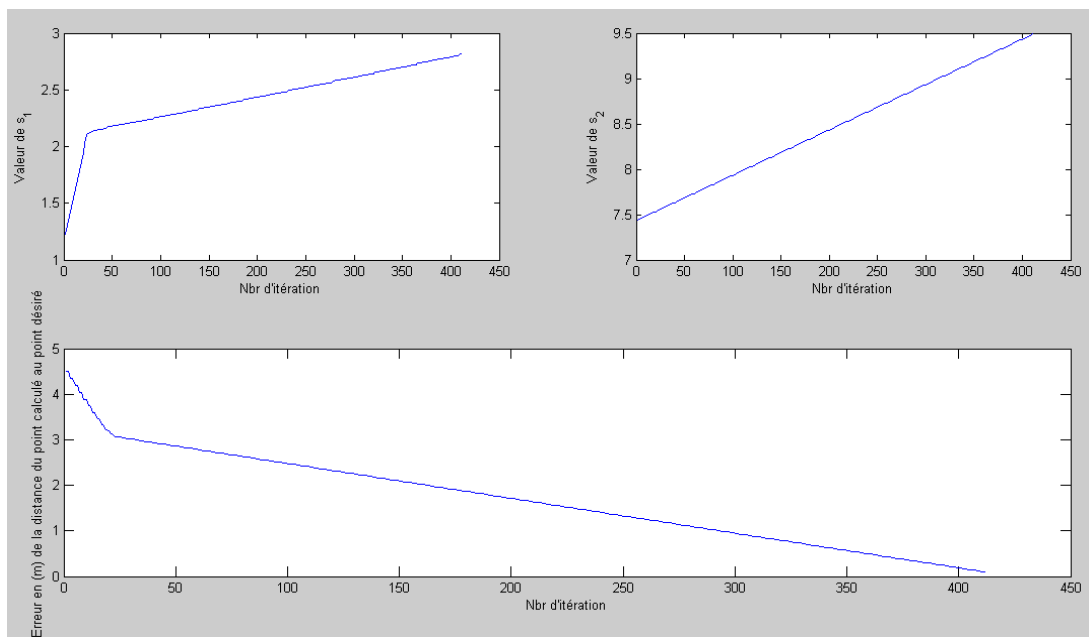


Figure 28 : Etude de performance de l'algorithme

En visualisant l'évolution de s_1, s_2 et de l'erreur commise, on peut conclure que l'algorithme n'est pas très performant.

Proposition d'amélioration : Evaluer la perturbation par rapport à l'erreur. Cette proposition vient du fait que l'on peut apercevoir que l'évolution des paramètres est presque linéaire ainsi que l'erreur commise.

Nous allons tester le même algorithme en posant que $\text{Perturbation} = \text{Erreur}^3/1000 + \text{Erreur}/10$

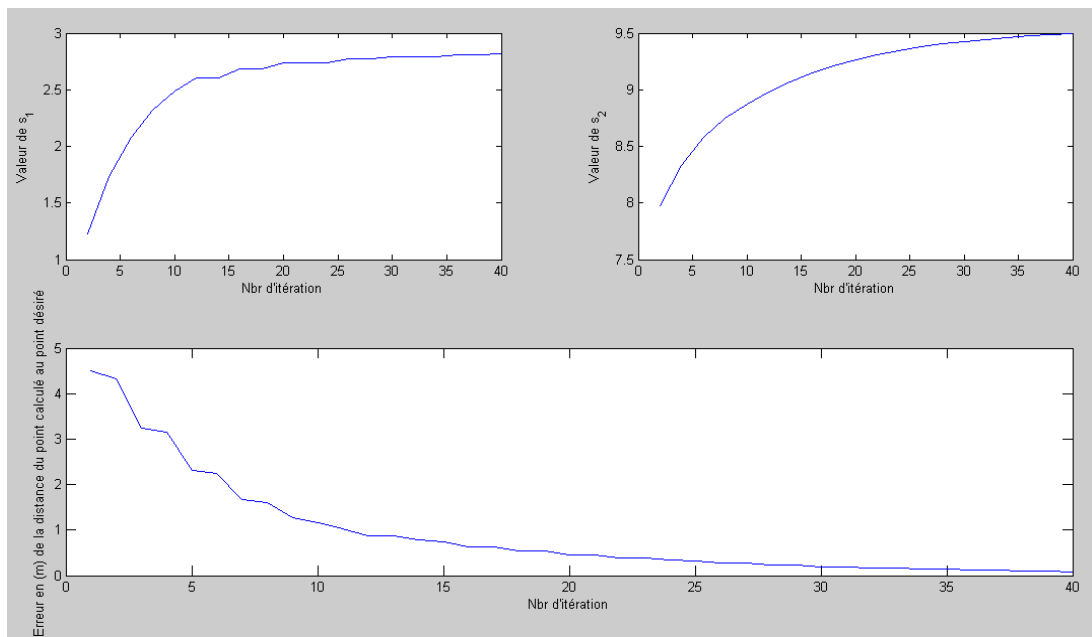


Figure 29 : Algorithme amélioré

On constate une grande amélioration ! On passe de 400 itérations à 40 itérations. Nous venons donc de voir qu'il est possible d'améliorer la convergence de cet algorithme. Cependant, ce résultat vient de « test » et n'est pas correct mathématiquement, car selon la configuration des points ce calcul de la perturbation ne permet pas de converger vers une solution.

Nous avons tenter d'étudier analytiquement afin d'écrire une fonction itérative qui converge vers la bonne solution, cependant nous ne sommes pas parvenu à finir l'étude du fait de la difficulté du problème.

Pour parer à cela, nous avons utilisé une fonction avancée d'optimisation de Matlab qui recherche le minimum d'une fonction fmincon :

```
Xc=@(x) Xi+quad(@(s) cos(Thetai+ci.*s+x(4).*s.^2./2),0,x(1))+quad(@(s)
cos(Thetai+ci.*x(1)+x(4).*x(1)^2/2+(ci+x(4).*x(1)).*s-
x(4).*s.^2./2),0,x(2))+quad(@(s)
cos(Thetai+ci.*(x(1)+x(2)+s)+x(4).*(x(1)*x(2)+x(1)*s-x(2).*s)+x(4).*(x(1)^2-
x(2)^2+s.^2)./2 ),0,x(3));

Yc=@(x) Yi+quad(@(s) sin(Thetai+ci.*s+x(4).*s.^2./2),0,x(1))+quad(@(s)
sin(Thetai+ci.*x(1)+x(4).*x(1)^2/2+(ci+x(4).*x(1)).*s-
x(4).*s.^2./2),0,x(2))+quad(@(s)
sin(Thetai+ci.*(x(1)+x(2)+s)+x(4).*(x(1)*x(2)+x(1)*s-x(2).*s)+x(4).*(x(1)^2-
x(2)^2+s.^2)./2 ),0,x(3));
distance = @(x) (sqrt((Xc(x)-Xf)^2 +(Yc(x)-Yf)^2)+abs(Thetai-
Thetaf+courbure.*(x(1)+x(2)+x(3))+x(4).*(x(1)*x(2)-
x(2)*x(3)+x(1)*x(3))+x(4).*(x(1).^2-x(2).^2+x(3).^2)./2));

%On va chercher le minimum de la fonction distance
x = fmincon(distance,X0,[-1 0 0 0;0 -1 0 0;0 0 -1 0;0 0 0 0],[-1E-2;-1E-2;-1E-
2;betamax],zeros(4,4),zeros(4,1),[1E-2, 1E-2,1E-2,-betamax],[1E3,
1E3,1E3,betamax])
```

Cette fonction $x = \text{fmincon}(\text{fonction}, x_0, A, B, A_{eq}, B_{eq}, lb, ub)$ va chercher le minimum de la fonction distance qui dépend d'un vecteur $x = [s_1, s_2, s_3, \beta]$ tel que $A \cdot x \leq B$ et $A_{eq} \cdot x = B_{eq}$ et $lb < x < ub$

Pour notre algorithme nous devons imposer $s_1, s_2, s_3 > 0$ et $-\beta_{max} < \beta < \beta_{max}$

Remarque : nous n'avons pas pu tenir compte de l'ensemble des contraintes de systèmes car il faut satisfaire aux équations (III.7) à (III.10) et il est impossible de les mettre en une seule équation à minimiser. C'est pourquoi nous avons du créer une fonction distance dans laquelle on a intégré la distance entre le point calculé et le point final souhaité ainsi que la différence d'orientation entre le point final calculé et celui souhaité. Cependant la fonction ne minimise pas très bien car cela dépend de « la taille des dimensions » en jeu, en effet nous mettons une distance en mètre et une distance en radian à minimiser, les distances en jeu ne sont pas de même échelle pour être comparées. C'est pourquoi il est souhaitable de mettre un coefficient à chacune de ses distances afin que l'algorithme converge bien vers le minimum des deux fonctions.

Si on souhaite une erreur de 0.1cm sur la position et de 0.01rad, alors on va mettre un coefficient 10 sur l'angle afin que les erreurs soient de même échelle pour l'algorithme.

Nous avons essayé cette méthode, mais le résultat ne nous convient pas encore car seule une des deux contraintes de distance est respectée.

Finalement nous avons préféré respecter la contrainte de positionnement face à l'orientation, et donc nous adaptons notre trajectoire en fonction des angles calculées même s'ils ne respectent pas forcément le choix désiré.

Nous avons alors :

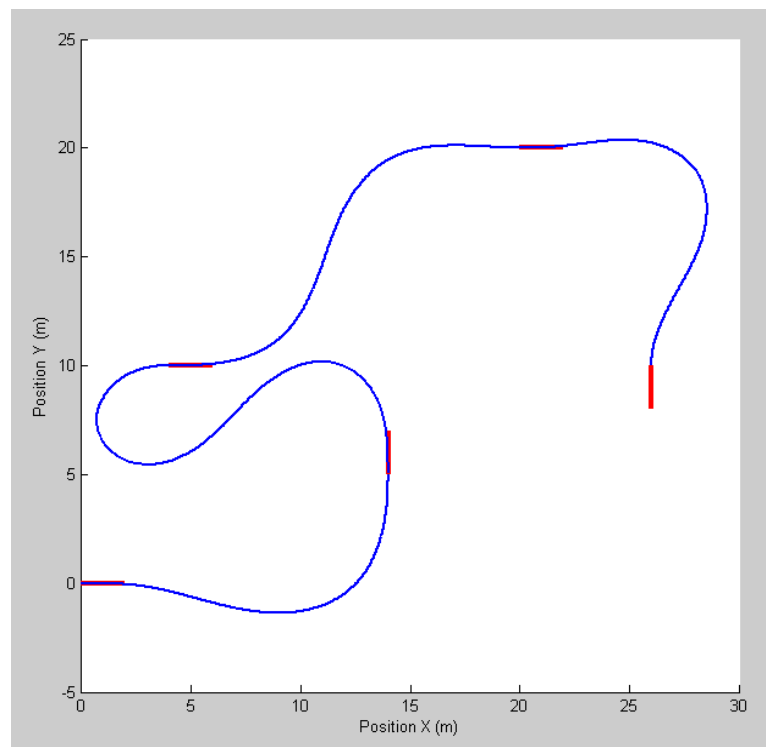


Figure 30 : Trajectoire générée par algorithme de convergence

On peut remarque que si le vecteur initial pour la fonction fmincon est mal paramétré nous pouvons avoir de « belles » courbes, pas forcément les plus optimisées en tant que trajectoire entre deux points :

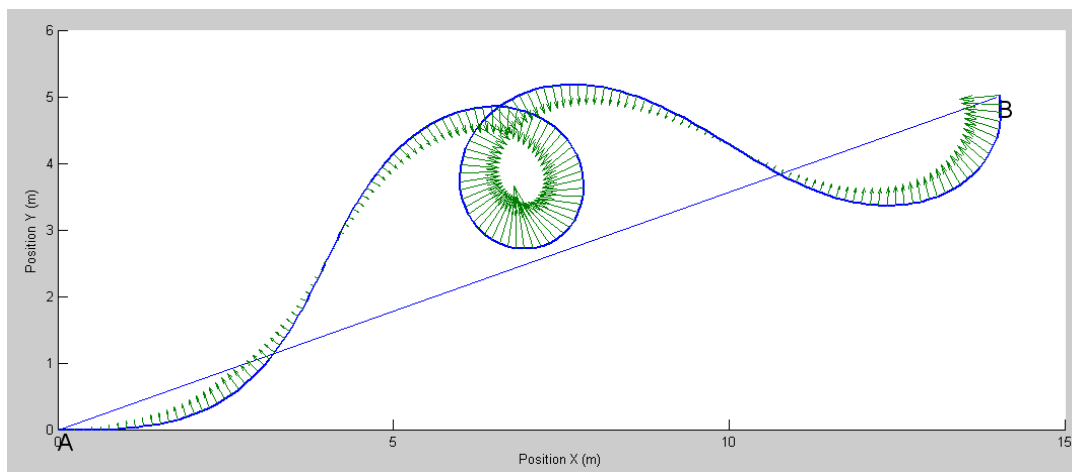


Figure 31 : Fonctions avancées de matlab

On peut constater que cette génération de trajectoire n'est pas anodine, on la retrouve sur des tracés autoroutiers comme à Ty-Pichon (France 51) que l'on peut voir sur la figure 23.



Figure 32 : Ty-Pichon 5685 47°46'48.18"N 3°20'41.44"O

Liste des figures :

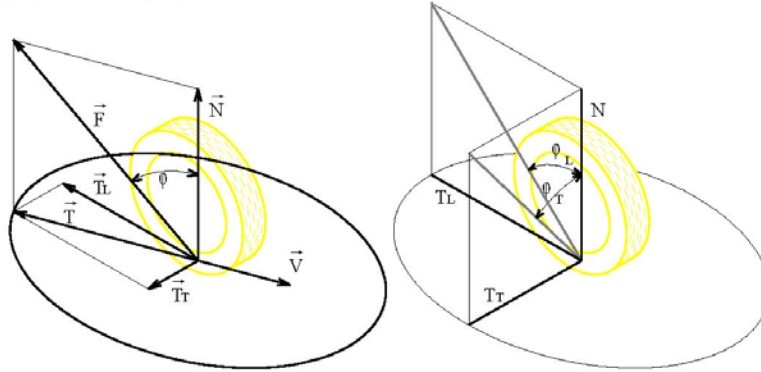
Figure 1 : Courbe Γ	5
Figure 2 : Trajectoire Arc de cercle VS Clothoïde	10
Figure 3 : Clothoïde sous Matlab.....	10
Figure 4 : Schématisation du robot	11
Figure 5: Formalisation mathématique	12
Figure 6 : trigonométrie	12
Figure 7 : Adhérence : Vue en coupe plane d'une roue	13
Figure 8 : Frottement de roulement (source Wikipédia)	14
Figure 9 : Adhérence : Vue en 3 dimensions d'une roue	14
Figure 10 : Schéma du robot en vue de l'étude mécanique	17
Figure 11 : Graphe des liaisons.....	17
Figure 12 : Dérivée de $f(yg,zg,ay)$ par rapport à zg	22
Figure 4 : Exemple de trajectoire avec deux points A et C.....	25
Figure 5 : Symétrie de la trajectoire par clothoïde.....	26
Figure 6 : $\alpha = \pi / 4$	27
Figure 7 : $\alpha = \pi / 2$	27
Figure 8 : $\alpha = \pi$	27
Figure 9 : $\alpha = \pi + \pi/3$	27
Figure 10 : $\alpha = \pi + 2 * \pi/3$ et $\alpha = \pi/3$	28
Figure 11 : $Xc=0.5 ; Yc=0.5 ; \alpha=\pi/2$;	32
Figure 12 : Visualisation accélération clothoïde.....	33
Figure 13 : Clothoïde en S.....	33
Figure 14 : para métrisation clothoïde en S	33
Figure 15 : Visualisation accélération clothoïde en S.....	34
Figure 16 : trajectoire générée par "blocs"	35
Figure 17 : Algorithme de convergence	37
Figure 18 : Clothoïde $s1=2.8118, s2=9.4954, s3=6.6836, k=-0.0427$	38
Figure 19 : Etude de performance de l'algorithme	38
Figure 20 : Algorithme amélioré.....	39
Figure 21 : Trajectoire générée par algorithme de convergence.....	40
Figure 22 : Fonctions avancées de matlab	41
Figure 23 : Ty-Pichon 5685 47°46'48.18"N 3°20'41.44"O	41

ANNEXES

Annexe 1 : Roulement avec glissement

Analyse du contact pneu/sol (ellipse de Kann).

\vec{F} est l'action du sol sur le pneu. \vec{N} sa projection normale, \vec{T} sa projection dans le plan de contact (tangente donc). Cette composante \vec{T} (du sol sur le pneu) s'oppose au déplacement (du pneu par rapport au sol) caractérisé ici par la vitesse \vec{V} . On projette \vec{T} sur les deux directions longitudinale et transversale pour introduire les facteurs d'adhérence caractéristiques μ_L et μ_T . Les composantes de T vérifient alors les relations : $|T_L| \leq \mu_L \cdot N$ et $|T_T| \leq \mu_T \cdot N$. Leurs valeurs maxi définissent l'ellipse de Kann.



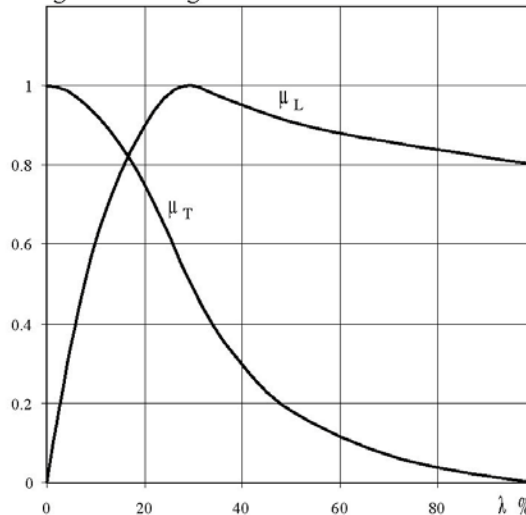
Courbes d'évolution des paramètres d'adhérence

On représente μ_L et μ_T en fonction du "pseudo-glissement", ou "glissement spécifique" défini par

$$\lambda = \frac{|V_V - V_R|}{V_V} \text{ où } V_V \text{ est la vitesse du véhicule et } V_R \text{ la vitesse circonférentielle du point théorique de}$$

contact pneu/sol. Une roue bloquée a un glissement de 100%. Une roue idéalement (?) modélisée par un contact ponctuel avec roulement sans glissement a un glissement nul, et une adhérence... nulle aussi !!!

Ce pseudo-glissement est une conséquence de la souplesse du pneumatique. En pratique, un véhicule roulant à vitesse constante a un λ d'environ 10%. Au freinage de 15 à 25 %. La présence de pluie change peu ces valeurs, par contre modifie bien sûr les valeurs des μ . μ_L maxi (à 30% de glissement) vaut 1 voire 1,2 sur chaussée sèche (pneus neufs), 0,6 sur route mouillée, 0,2 sur verglas. Il diminue de 20 (route sèche) à 30% (mouillée) par rapport à ces valeurs maxi lorsque le glissement est de 100%. Par contre μ_T , lui, se casse la figure quand le glissement augmente...



Annexe 2 : Programme Matlab pour dessiner une clothoïde

```

clc
clf
clear all
close all

figure(1)
hold on
%% Calcul de la première partie de clothoïde

alpha=-pi/3;
if alpha >=0
    positif = true;
else
    positif= false;
    alpha=-alpha;
end
alphaB=alpha/2; % Angle de rotation final

n=100;%pas
Beta =33; % caractéristique de la clothoïde
smax=sqrt(2*alphaB/Beta); %abscise curviligne pour atteindre le point B

Alpha = @(s) s.^2.*Beta/2; %Angle de direction // 1/R = Alpha'

F = @(s) cos(Alpha(s));
G = @(s) sin(Alpha(s));

%On calcule ici la première partie de clothoïde
for i=1:n
    s=i*smax/n;
    if positif
        X(i)=quad(F,0,s);
        Y(i)=quad(G,0,s);
    else
        X(i)=quad(F,0,s);
        Y(i)=-quad(G,0,s);
    end
end

%% Courbure complete avec deux clothoides

if positif
    XX=-X.*cos(alpha) - Y.*sin(alpha); %Rotation pour copier la partie
symétrique
    YY=-X.*sin(alpha) + Y.*cos(alpha);
else
    XX=-X.*cos(-alpha) - Y.*sin(-alpha);
    YY=-X.*sin(-alpha) + Y.*cos(-alpha);
end

```

```

XX=XX-XX(length(X))+X(length(X));
YY=YY-YY(length(X))+Y(length(Y));
for i=1:length(XX)
    tmp(i)=XX(length(XX)-i+1);
    tmpp(i)=YY(length(XX)-i+1);
end

X=[X tmp(2:length(tmp))];%On rajoute la partie symétrique
Y=[Y tmpp(2:length(tmpp))];

%On affiche le résultat
plot(X,Y,'LineWidth',2)
daspect([1 1 1]);
xlabel('Position X (m)')
ylabel('Position Y (m)')

if positif
    XB=quad(F,0,sqrt(2*alphaB/Beta))
    YB=quad(G,0,sqrt(2*alphaB/Beta))
    plot(0.11:0.01:0.25, tan(alphaB)*(0.11:0.01:0.25)+YB-
XB*tan(alphaB), 'g', 'LineWidth',2)
    plot(0.05:0.01:0.25, tan(alphaB+pi/2)*(0.05:0.01:0.25)+YB-
XB*tan(alphaB+pi/2), 'r', 'LineWidth',2)
    disp('Calcul des coordonnées du point C')
    A=[-sin(alphaB) cos(alphaB) ; -tan(alphaB+pi/2) 1];
    B=[0; YB-XB*tan(alphaB+pi/2)];
    AC=2.*inv(A)*B
else
    XB=quad(F,0,sqrt(2*alphaB/Beta))
    YB=-quad(G,0,sqrt(2*alphaB/Beta))
    plot(0.11:0.01:0.25, tan(-alphaB)*(0.11:0.01:0.25)+YB-XB*tan(-
alphaB), 'g', 'LineWidth',2)
    plot(0.05:0.01:0.25, tan(-alphaB+pi/2)*(0.05:0.01:0.25)+YB-XB*tan(-
alphaB+pi/2), 'r', 'LineWidth',2)
    A=[-sin(-alphaB) cos(-alphaB) ; -tan(-alphaB+pi/2) 1];
    B=[0; YB-XB*tan(-alphaB+pi/2)];
    AC=2.*inv(A)*B
end

plot([0 AC(1)],[0 AC(2)], '--k')
text(0,0,'A', 'VerticalAlignment','Top', 'FontSize',18)
text(XB,YB,'B', 'VerticalAlignment','Top', 'FontSize',18)
text(AC(1),AC(2),'C', 'VerticalAlignment','Top', 'FontSize',18)

```

Glossaire

Odométrie : L'odométrie est une technique permettant d'estimer la position d'un véhicule en mouvement. Le terme vient du grec *hodos* (voyage) et *metron* (mesure). Cette mesure de bas niveau est présente sur quasiment tous les robots mobiles, grâce à des capteurs embarqués permettant de mesurer le déplacement du robot (de ses roues).

Repérage relatif : On appelle repérage relatif tout repérage à partir d'un point fixe sur terre que l'on connaît, ce repérage se base en calculant ses avancées successives pour prédire sa position comme une souris de PC.

Repérage absolu : On appelle repérage absolu tout repérage qui ne dépend pas de sa position antérieure, par exemple le GPS

Bibliographie

1. **R.C.V.A**, Correction odométrique, <http://www.youtube.com/watch?v=hl892EU5HtA>
2. http://fr.wikibooks.org/wiki/Tribologie/Facteurs_de_frottement
3. *CRC Handbook of Chemistry and Physics, 85th Edition*, COEFFICIENT OF FRICTION ,15-50
4. *Path generation for robot vehicles using Clothoid segments 1990* , Dong Hun Shin ans Sanjiv Singh, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213
5. *Suivi de chemin par un robot*, Damien PERRITAZ, Ecole polytechnique fédérale de Lausanne 2005
6. [http://fr.wikibooks.org/wiki/Tribologie - Mod%C3%A9lisation_des_actions_de_contact](http://fr.wikibooks.org/wiki/Tribologie_-_Mod%C3%A9lisation_des_actions_de_contact)

Conclusion

Nous avons pu décider après une étude complète de la dynamique du robot, l'achat des moteurs et codeurs à acheter pour le moteur en se basant sur le couple moteur nécessaire calculé à partir de la résolution analytique de la dynamique du robot et de l'accélération maximale qu'il doit pouvoir atteindre.

Nous avons aussi développé un ensemble de lois de trajectoire basé sur les courbes clothoïdes afin de faire gagner au robot un temps crucial lors des matchs de la coupe de France de robotique.

Cependant il reste encore à écrire les algorithmes de génération de trajectoire grâce à des points de passage que la personne chargée de la stratégie de match va définir. Il reste aussi à développer un asservissement en trajectoire qui tienne en compte des limites de déplacements du robot que nous avons calculé dans ce dossier (accélération maximale etc.). Puis finalement il va falloir implémenter toute théorie sur un système réel qui est le robot une fois que l'équipe mécanique aura fini son travail.

Finalement, on peut conclure qu'il reste encore beaucoup de travail, et qu'il n'est pas encore l'heure pour se reposer.

Résumé

L'intégralité de ce dossier porte sur l'étude de l'odométrie d'un mobile afin de pouvoir utiliser celui-ci jusqu'à ses limites. C'est-à-dire que nous cherchons à l'utiliser à sa vitesse maximale sans perdre en précision sur sa position, en évitant qu'il dérape. Tout cette étude est très mathématique mais cependant donne des résultats très prometteurs. Nous espérons pouvoir présenter ce travail en l'implémentant sur le robot de la coupe de France de robotique afin de valider cette recherche.