

Géométrie et Commande des Drones

Professeur à l'ENSTA Bretagne, Lab-STICC,

29200 Brest, France

Mots clefs

Robotique, Modélisation, Commande, Drone

Résumé

Les drones, qu'ils soient aériens ou sous-marins, évoluent dans un monde tridimensionnel en prenant appui sur le fluide environnant pour se propulser et se diriger. Ils sont rendus autonome à travers une boucle de contrôle.

Cet article propose un ensemble d'outils géométriques facilitant la mise en équations du mouvement du drone. Cette modélisation facilitera la synthèse de contrôleurs.

Les applications traitées seront l'atterrissage autonome d'un hexarotor sur une plateforme mobile, le looping d'un hélicoptère, le suivi de chemin pour une torpille sous-marine et la stabilisation d'un quadrirotor.

Les simulations seront faites sous le langage Python et les codes sont donnés sur un site dédié.

Keywords

Robotics, Modeling, Control, Drones

Abstract

Drones, whether aerial or underwater, evolve in a three-dimensional world by relying on the surrounding fluid to move or navigate. The drone are made autonomous through a control loop.

This paper proposes a set of geometric tools to facilitate the modeling. This modelling will be used for the synthesis of controllers.

The applications treated will be the autonomous landing of a hexarotor on a mobile platform, the looping of a helicopter, the tracking of an underwater torpedo and the stabilization of a quadrotor.

The simulations will be done under the Python language and the codes are given on a dedicated website.

1 Introduction

Vu du grand public, un drone est souvent considéré comme un objet volant autostabilisé, téléopéré et muni d'une caméra. Il est aussi souvent mentionné comme un avion de guerre sans pilote avec un certain degré d'autonomie capable de procéder à des missions militaires. Pourtant, un drone n'est pas forcément volant. Il s'agit d'un robot mobile qui peut être aérien, terrestre ou sous marin, naviguant ou roulant. Un drone peut donc être considéré en première approche comme un véhicule sans personne à bord, de toute taille, avec un degré d'autonomie élevé. Il est capable de naviguer d'un point à un autre, en évitant les obstacles et avec souvent une mission à effectuer.

Différents types de missions peuvent être données à un drone, comme par exemple, la cartographie d'une zone, le transport d'un colis, la recherche d'une épave sous-marine. Les drones sont de plus en plus utilisés pour des missions longues, et pénibles, comme par exemple, certains travaux agricoles, le nettoyage (comme le passage de l'aspirateur) ou pour des missions dangereuses pour l'homme (intervention dans des zones irradiées, la recherche de personnes suite à une avalanche ou un tremblement de terre). Ils sont indispensables dans les zones impossibles d'accès pour l'homme (très grand fonds, planètes lointaines, intérieur des volcans, etc.).

Un des enjeux pour le drone est l'autonomie. Pour des tâches répétitives qui demandent une localisation précise et un cahier des charges à respecter, les drones pourront exécuter sans risque d'erreur une procédure logique en obéissant scrupuleusement au programme informatique qui le pilote. De plus, dans certains environnements, comme le fond de l'océan, ou les planètes lointaines, la téléopération est presque impossible. Il faudra donc donner à ces robots le degré d'autonomie le plus élevé possible. Un autre des défis à relever est la sécurité. Aucune erreur ne sera permise au drone. Il faut être certain qu'il ne provoquera pas de collision, que le robot ne prendra aucun risque et qu'il ne va échouer dans sa mission.

Pour développer cette autonomie, il faut une bonne compréhension de la dynamique du robot afin de pouvoir le simuler et le commander. Pour cela, il nous faut comprendre précisément comment le robot s'oriente et bouge. La géométrie non euclidienne possède un rôle fondamental dans cette représentation. Cet article propose de décrire les outils géométriques associés aux déplacements des drones et d'expliquer comment utiliser des outils afin de modéliser proprement les drones et les amener progressivement vers une autonomie complète.

2 Qu'est-ce qu'un drone ?

Un *drone* peut être défini comme un solide capable de se mouvoir dans un environnement fluide (air ou eau) de façon autonome. Ainsi, ne seront pas considérés comme des drones

- Un robot manipulateur qui est articulé [1] et qui n'est donc pas un corps solide
- les voitures autonomes [2] car elles n'évoluent pas dans un fluide
- les bateaux autonomes [3] car ils évoluent à l'interface entre deux fluides
- les satellites qui ne s'appuient pas sur un fluide.

Un drone fait donc partie de la classe des robots mobiles autonomes. Il comprend donc

- des *capteurs* qui vont collecter des informations sur son environnement et son état, comme par exemple son orientation et sa position ;
- des *actionneurs* qui vont permettre le déplacement ;
- une *intelligence* (ou régulateur), qui va chercher à calculer comment agir sur les actionneurs à partir des informations collectées afin de se déplacer suivant la direction souhaitée.

Les drones sont en constante évolution, principalement depuis les années 2000, dans le domaine militaire [4] ou maritime [5], médical, agricole, spatial, etc. C'est le cas par exemple de la chasse aux mines où des drones ratissent une zone pour détecter la présence de mines, de la recherche des boîtes noires d'un avion qui s'est abîmé en mer et de l'exploration de planètes. La robotique mobile est une discipline qui cherche à concevoir des robots capables d'aider dans ce type de tâches [6] de façon plus ou moins autonome. Elle utilise des outils des mathématiques appliquées comme l'automatique, le traitement du signal, l'informatique ou l'électronique.

Un des objectifs de cet article est de donner quelques outils géométriques qui permettent de modéliser et de contrôler des drones. Ces drones seront modélisés par des équations d'état à temps continus, c'est-à-dire, une équation

différentielle vectorielle du premier ordre de la forme

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \quad (1)$$

où $\mathbf{x}$ est le vecteur d'état et $\mathbf{u}$ est le vecteur d'entrée. Ces équations d'état sont obtenues en combinant les équations de la mécanique avec des équations cinématiques [7]. Cette étape de modélisation est fondamentale, à la fois pour simuler et comprendre les drones [8], mais également pour la conception des régulateurs. Nous allons explorer le volet tri-dimensionnel car les drones qu'ils soient avion, quadrirotor, sous marin, évoluent dans un monde à trois dimension.

Souvent, une modélisation par des équations d'état fait apparaître des paramètres constants comme des masses, des matrices d'inertie, des longueurs, des coefficients de viscosité, etc. Une étape d'identification doit alors être faite pour l'obtention d'un modèle complet. Dans ce document, nous supposons que cette étape a déjà été réalisée et que tous ces paramètres sont connus.

Pour la régulation qui va nous intéresser, nous allons supposer que l'état est connu exactement, ce qui n'est en général pas le cas. Les étapes de localisation ou même de cartographie [9] pourtant souvent nécessaires pour atteindre une autonomie complète ne seront donc pas considérées ici.

Il n'existe pas de méthodologie systématique pour modéliser ou bien commander un drone. Pourtant, un certain nombre de concepts et d'outils venant de la géométrie non-Euclidienne, comme les groupes de Lie, les algèbres de Lie ou les géodésiques reviennent souvent. Dans la section 3, nous allons rappeler certains de ces concepts, ceux qui nous semblent les plus fondamentaux. La section 4 applique ces outils pour modéliser un drone générique en le considérant comme un corps rigide capable de se mouvoir dans l'espace. La section 5 explique comment il est possible de contrôler un drone en orientation et en vitesse. Le juste niveau d'abstraction a été choisi de façon à ne pas avoir à spécifier le type de drone considéré. Les sections suivantes montrent comment s'appliquent les méthodes développées pour ce drone générique afin de traiter différentes catégories de drones. La section 6 propose un régulateur capable de stabiliser un quadrirotor et de lui faire suivre une trajectoire décrite par un champ de vecteur. La section 7 s'intéresse au contrôle d'une torpille sous-marine comprenant un propulseur et trois gouvernes. La section 8 propose un régulateur capable de commander un hélicoptère en orientation dans le but de lui faire décrire un looping à l'altitude désirée. La section 9 s'attaque au problème de l'atterrissage précis d'un hexarotor sur une plateforme mobile. La section 10 introduit dans le contexte de la commande des drones l'intérêt des géodésiques dans les espaces non-euclidiens. Il s'agit de courbes minimisant la longueur du chemin à parcourir. Cet article se termine par une conclusion qui récapitule les notions de géométrie introduites et leur intérêt pour le développement de drones fiables.

3 Orientation dans l'espace

Nous allons rappeler ici quelques notions fondamentales pour représenter l'orientation des objets dans l'espace, sachant que cette orientation évolue dans le temps.

3.1 Vecteur de rotation

Le *produit vectoriel* entre deux vecteurs $\boldsymbol{\omega}$ et $\mathbf{x} \in \mathbb{R}^3$ est classiquement défini comme suit :

$$\begin{aligned} \boldsymbol{\omega} \wedge \mathbf{x} &= \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \wedge \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_3\omega_y - x_2\omega_z \\ x_1\omega_z - x_3\omega_x \\ x_2\omega_x - x_1\omega_y \end{pmatrix} \\ &= \underbrace{\begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}}_{\boldsymbol{\omega} \wedge} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \end{aligned} \quad (2)$$

Ainsi, au vecteur $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$, on peut associer la matrice anti-symétrique $\boldsymbol{\omega} \wedge$ qui sera également dénotée $\wedge(\boldsymbol{\omega})$ pour faire apparaître l'opérateur $\wedge$.

3.2 Matrices de rotation

Pour l'orientation tridimensionnelle des robots il est essentiel d'avoir une bonne compréhension sur les matrices de rotation.

Rappelons qu'une *rotation* est une application linéaire qui est une isométrie (c'est-à-dire qu'elle préserve le produit scalaire) et qui est directe (elle ne change pas l'orientation de l'espace). Il en découle qu'une matrice de rotation $\mathbf{R}$ satisfait les deux propriétés suivantes :

$$\mathbf{R}^T \mathbf{R} = \mathbf{I} \text{ et } \det \mathbf{R} = 1. \quad (3)$$

Si $\mathbf{R}$ est une matrice de rotation qui dépend du temps t , en différentiant la relation $\mathbf{R}\mathbf{R}^T = \mathbf{I}$, nous obtenons

$$\dot{\mathbf{R}}\mathbf{R}^T + \mathbf{R}\dot{\mathbf{R}}^T = \mathbf{0}. \quad (4)$$

Ainsi $\dot{\mathbf{R}}\mathbf{R}^T$ est une matrice anti-symétrique. On peut démontrer que dans le cas où $\mathbf{R}$ est de dimensions 3×3 ,

$$\dot{\mathbf{R}}\mathbf{R}^T = \boldsymbol{\omega} \wedge \quad (5)$$

et

$$\mathbf{R}^T \dot{\mathbf{R}} = (\mathbf{R}^T \boldsymbol{\omega}) \wedge \quad (6)$$

où le vecteur $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$ est le vecteur de rotation associé.

3.3 Exponentielles et Logarithmes des matrices

Le vecteur de rotation et la matrice de rotation sont des objets de nature différente. En effet, on peut additionner des vecteurs de rotation alors que l'on peut multiplier des matrices de rotation. Ces deux objets sont liés par l'exponentielle et réciproquement le logarithme. Nous allons ici rappeler ces opérateurs dans le cas matriciel qui va nous intéresser.

Exponentielle. Soit une matrice carrée $\mathbf{M}$ de dimension n . Son exponentielle est définie par :

$$e^{\mathbf{M}} = \mathbf{I}_n + \mathbf{M} + \frac{1}{2!}\mathbf{M}^2 + \frac{1}{3!}\mathbf{M}^3 + \dots = \sum_{i=0}^{\infty} \frac{1}{i!}\mathbf{M}^i$$

où $\mathbf{I}_n$ est la matrice identité de dimension $n \times n$. Notons que $e^{\mathbf{M}}$ a la même dimension que $\mathbf{M}$. Rappelons les propriétés suivantes :

$$\begin{aligned} e^{0_n} &= \mathbf{I}_n \\ e^{\mathbf{M}} e^{\mathbf{N}} &= e^{\mathbf{M}+\mathbf{N}} \text{ (si } \mathbf{M}, \mathbf{N} \text{ commutent)} \\ \frac{d}{dt} (e^{\mathbf{M}t}) &= \mathbf{M} e^{\mathbf{M}t} \end{aligned} \quad (7)$$

où $\mathbf{M}, \mathbf{N}$ sont deux matrices de dimension $n \times n$

Logarithme. Soit $\mathbf{M}$ une matrice de dimension $n \times n$, la matrice $\mathbf{L}$ est dite logarithme de $\mathbf{M}$ si $e^{\mathbf{L}} = \mathbf{M}$. Comme c'est le cas pour les nombres complexes, la fonction exponentielle n'est pas injective et donc une matrice peut avoir plusieurs logarithmes. On peut aussi définir de façon unique le logarithme par son développement en série entière :

$$\log \mathbf{M} = \sum_{i=1}^{\infty} \frac{(-1)^{i+1}}{i} (\mathbf{M} - \mathbf{I}_n)^i. \quad (8)$$

Cette somme est convergente si $\mathbf{M}$ est proche de l'identité.

3.4 Formules de Rodrigues

En dimension 3, une matrice de rotation $\mathbf{R}$ est associée à un axe de rotation représenté par un vecteur unitaire $\mathbf{u}$ et un angle α relativement à cet axe. Le lien $\mathbf{R} \leftrightarrow (\mathbf{u}, \alpha)$ peut être formalisé par les formules de *Rodrigues*

$$\begin{aligned} (i) \quad \mathbf{R} &= \text{Exp}(\alpha \mathbf{u}) \\ (ii) \quad \alpha \mathbf{u} &= \text{Log}(\mathbf{R}) \end{aligned} \quad (9)$$

avec

$$\begin{aligned} \text{Log}(\mathbf{R}) &= \wedge^{-1}(\log \mathbf{R}) \\ \text{Exp}(\alpha \mathbf{u}) &= e^{\alpha \mathbf{u} \wedge} \end{aligned} \quad (10)$$

Dans ces formules, (ii) est la réciproque de (i). La version capitalisée *Exp*, *Log* de *exp*, *log* sont des raccourcis utilisés pour éviter l'utilisation de l'opérateur $\wedge$. Cela est illustré par la figure 1. L'ensemble des matrices de rotation dans $\mathbb{R}^3$ forme le groupe de Lie $SO(3)$ et les matrices antisymétriques de $\mathbb{R}^3$ forment l'algèbre de Lie associée. L'exponentielle et le logarithme forment un lien entre ces deux structures algébriques [10][11].

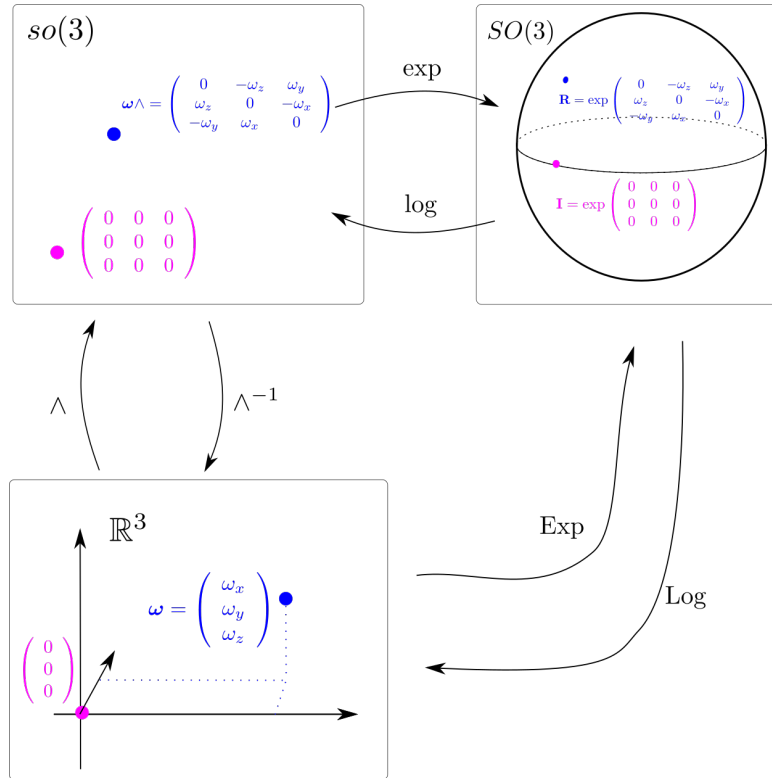


FIGURE 1 – Correspondances entre le groupe de Lie $SO(3)$ et l'algèbre de Lie $\mathbf{so}(3)$

Le vecteur ω et sa matrice associée $\wedge(\omega)$ représentent les mêmes objets écrits différemment. On aura tendance à les confondre, de même que *Exp* sera souvent confondu avec *exp*. Ainsi, par exemple, si $\mathbf{i}, \mathbf{j}, \mathbf{k}$ sont les vecteurs de la

base canonique de $\mathbb{R}^3$, on aura :

$$\begin{aligned} e^{\alpha \mathbf{i}} &= \exp \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -\alpha \\ 0 & \alpha & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix} \\ e^{\alpha \mathbf{j}} &= \exp \begin{pmatrix} 0 & 0 & \alpha \\ 0 & 0 & 0 \\ -\alpha & 0 & 0 \end{pmatrix} = \begin{pmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{pmatrix} \\ e^{\alpha \mathbf{k}} &= \exp \begin{pmatrix} 0 & -\alpha & 0 \\ \alpha & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{aligned} \quad (11)$$

Expressions analytiques

Soit une matrice de rotation $\mathbf{R}$ de $SO(3)$, son logarithme (qui est un vecteur de $so(3)$) est donné par

$$\begin{aligned} \text{Log} \mathbf{R} &= \frac{\alpha}{2 \sin \alpha} \wedge^{-1} (\mathbf{R} - \mathbf{R}^T) \\ \alpha &= \text{acos}\left(\frac{\text{tr}(\mathbf{R})-1}{2}\right) \end{aligned} \quad (12)$$

où $\text{tr}(\mathbf{R})$ désigne la trace de la matrice $\mathbf{R}$. Soit $\omega \in so(3)$, son exponentielle (qui est une matrice de rotation) est donnée par

$$\begin{aligned} \text{Exp} \omega &= \mathbf{I} + \frac{\sin \alpha}{\alpha} \omega \wedge + \frac{1 - \cos \alpha}{\alpha^2} (\omega \wedge)^2 \\ \alpha &= \|\omega\| \end{aligned} \quad (13)$$

3.5 Angles d'Euler

Pour représenter l'orientation d'un mobile dans l'espace, nous utiliserons principalement les matrices de rotation. Toutefois, il est souvent confortable d'utiliser les angles d'Euler bien que ces derniers engendrent des singularités. La relation qui relie une matrice de rotation à ses angles d'Euler est donnée par

$$\mathbf{R}_{\text{Euler}}(\varphi, \theta, \psi) = e^{\psi \mathbf{k} \wedge} e^{\theta \mathbf{j} \wedge} e^{\varphi \mathbf{i} \wedge}$$

avec $\mathbf{i} = (1, 0, 0)^T$, $\mathbf{j} = (0, 1, 0)^T$ et $\mathbf{k} = (0, 0, 1)^T$. Les angles d'Euler φ, θ, ψ sont appelés respectivement *gîte*, *assiette* et *cap*. Les termes *roulis*, *tangage*, *lacet*, souvent employés pour les désigner, sont à utiliser pour leur dérivée temporelle $\dot{\varphi}, \dot{\theta}, \dot{\psi}$. Nous éviterons d'utiliser les angles d'Euler, du fait des singularités qu'ils produisent et que l'on appelle souvent le *gimbal lock*. Cependant, pour la conception de régulateur, les angles d'Euler sont intuitifs, concrets et confortables à utiliser. Nous aurons tendance à les considérer lorsque l'orientation du drone est à mettre en lien avec le terrain. Par exemple, indiquer que le drone va vers l'ouest avec un angle d'assiette de 45% se traduit immédiatement en termes d'angles d'Euler. En revanche, les manipulations mathématiques de plus bas niveau se feront avec les matrices de rotation.

3.6 Groupes de Lie

L'ensemble des matrices de rotation de $\mathbb{R}^n$ forme un groupe relativement à la multiplication, appelé *groupe orthogonal spécial* (*spécial* car $\det \mathbf{R} = 1$, *orthogonal* car $\mathbf{R}^T \mathbf{R} = \mathbf{I}$). Il est noté $(SO(n), \cdot)$ et l'identité $\mathbf{I}$ est son élément neutre. De plus, les opérations de multiplication et d'inversion sont dérivables ce qui fait de $SO(n)$ un groupe de Lie [12] qui est une variété différentielle dans l'ensemble des matrices de $\mathbb{R}^{n \times n}$. L'équation matricielle $\mathbf{R}^T \mathbf{R} = \mathbf{I}$ peut se décomposer en $\frac{n(n+1)}{2}$ équations scalaires indépendantes. Par exemple, pour $n = 2$, nous avons $\frac{2(2+1)}{2} = 3$ équations scalaires. En effet, :

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix}^T = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \Leftrightarrow \begin{cases} a^2 + b^2 = 1 \\ ac + bd = 0 \\ c^2 + d^2 = 1 \end{cases} \quad (14)$$

Par conséquent, $SO(n)$ forme une variété différentielle de dimension $d = n^2 - \frac{n(n+1)}{2} = \frac{n(n-1)}{2}$.

- Pour $n = 1$, nous avons $d = 0$. L'ensemble $SO(1)$ est un singleton qui contient une unique rotation : $R = 1$.
- Pour $n = 2$, $d = 1$. Un seul paramètre ou angle est nécessaire pour représenter $SO(2)$.
- Pour $n = 3$, $d = 3$. Trois paramètres ou angles permettront de représenter $SO(3)$.

3.7 Evolution d'une matrice de rotation

Soit $\mathbf{R}(t)$ une matrice de rotation d'un corps mobile. On a :

$$\mathbf{R}(t + dt) = \mathbf{R}(t) \text{Exp}(dt \boldsymbol{\omega}_r(t)) \quad (15)$$

où $\boldsymbol{\omega}_r = \mathbf{R}^T \boldsymbol{\omega}$ est le vecteur de rotation exprimé dans le référentiel du mobile. C'est ce schéma d'intégration, dit *exponentiel*, qu'il nous faudra utiliser pour intégrer l'orientation du drone, plutôt qu'une méthode de Runge Kutta.

Pour montrer cette relation d'évolution, il faut partir de l'équation 6. Nous avons

$$\dot{\mathbf{R}}(t) = \mathbf{R}(t) (\boldsymbol{\omega}_r(t) \wedge) \quad (16)$$

et donc si dt est infiniment petit,

$$\begin{aligned} & \mathbf{R}(t + dt) \\ &= \mathbf{R}(t) + dt \dot{\mathbf{R}}(t) + o(dt) \\ &= \mathbf{R}(t) + dt \mathbf{R}(t) (\boldsymbol{\omega}_r(t) \wedge) + o(dt) \\ &= \mathbf{R}(t) (\mathbf{I} + dt (\boldsymbol{\omega}_r(t) \wedge) + o(dt)) \\ &= \mathbf{R}(t) \exp(dt \boldsymbol{\omega}_r(t) \wedge) + o(dt) \\ &= \mathbf{R}(t) \text{Exp}(dt \boldsymbol{\omega}_r(t)) + o(dt) \end{aligned} \quad (17)$$

Pour des raisons numériques, lorsque nous intégrons pendant une longue période, la matrice $\mathbf{R}(t)$ peut perdre la propriété $\mathbf{R}\mathbf{R}^T = \mathbf{I}$. Pour éviter cela, nous pouvons procéder à une étape de renormalisation, ce qui signifie que la matrice courante $\mathbf{R}$ doit être projetée sur $SO(3)$ qu'elle n'aurait jamais dû quitter. Pour cela, une factorisation QR peut être effectuée. Rappelons que pour une matrice carrée $\mathbf{M}$ une telle décomposition produit deux matrices $\mathbf{Q}, \mathbf{R}$ telles que $\mathbf{M} = \mathbf{Q}\mathbf{R}$ où $\mathbf{Q}$ est une matrice de rotation et $\mathbf{R}$ est triangulaire. Lorsque $\mathbf{M}$ est proche de $SO(3)$, $\mathbf{R}$ est presque diagonale et presque une matrice de rotation ce qui signifie que sur sa diagonale les valeurs sont proches de ± 1 . Le programme PYTHON génère une matrice de rotation $\mathbf{N}$ correspondant à la projection de $\mathbf{M}$ sur $SO(3)$.

```
Q,R = numpy.linalg.qr(M)
```

```
v=diag(sign(R))
```

```
N=Q@diag(v)
```

3.8 Evolution d'une transformation rigide

Le calcul de l'évolution évoqué dans la section précédente pour l'orientation s'applique également pour d'autres types de transformations pourvu que la structure de groupe de Lie soit respectée. Par exemple, les transformations rigides qui combinent translations et rotations forment un groupe de Lie puisque l'on peut les composer et les inverser. Le groupe des transformations rigides du plan est noté $SE(2)$. Pour des raisons pratiques de calcul sur ordinateur, il est préférable d'utiliser une représentation matricielle. Ainsi, la matrice

$$\mathbf{P} = \begin{pmatrix} \cos p_3 & -\sin p_3 & p_1 \\ \sin p_3 & \cos p_3 & p_2 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \quad (18)$$

correspond à une transformation rigide avec $\mathbf{t} = (p_1, p_2)$ étant la translation et p_3 l'angle de la rotation $\mathbf{R}$.

La structure de groupe de $SE(2)$ se vérifie facilement. Prenons $\mathbf{P}_1, \mathbf{P}_2$ dans $SE(2)$. La composition est interne car

$$\begin{pmatrix} \mathbf{R}_1 & \mathbf{t}_1 \\ \mathbf{0} & 1 \end{pmatrix} \begin{pmatrix} \mathbf{R}_2 & \mathbf{t}_2 \\ \mathbf{0} & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R}_1 \cdot \mathbf{R}_2 & \mathbf{R}_1 \cdot \mathbf{t}_2 + \mathbf{t}_1 \\ \mathbf{0} & 1 \end{pmatrix}. \quad (19)$$

L'inversion l'est également puisque

$$\begin{pmatrix} \mathbf{R}_1 & \mathbf{t}_1 \\ \mathbf{0} & 1 \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{R}_1^\top & -\mathbf{R}_1^\top \cdot \mathbf{t}_1 \\ \mathbf{0} & 1 \end{pmatrix}. \quad (20)$$

De plus, ces deux opérations sont différentiables, ce qui fait de $SE(2)$ un groupe de Lie. Notons que la matrice identité de $\mathbb{R}^3$ correspond à l'élément neutre de ce groupe. Le calcul de l'algèbre de Lie associée se fait en linéarisant

$$\mathbf{P}(p_1, p_2, p_3) = \begin{pmatrix} \cos p_3 & -\sin p_3 & p_1 \\ \sin p_3 & \cos p_3 & p_2 \\ 0 & 0 & 1 \end{pmatrix} \quad (21)$$

autour de la matrice identité, c'est-à-dire, autour de $\mathbf{p} = \mathbf{0}$. Nous obtenons à l'ordre 1

$$\begin{aligned} \mathbf{P}(p_1, p_2, p_3) &= p_1 \frac{\partial \mathbf{P}}{\partial p_1}(\mathbf{0}) + p_2 \frac{\partial \mathbf{P}}{\partial p_2}(\mathbf{0}) + p_3 \frac{\partial \mathbf{P}}{\partial p_3}(\mathbf{0}) \\ &= p_1 \underbrace{\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}}_{=\mathbf{E}_1} + p_2 \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}}_{=\mathbf{E}_2} + p_3 \underbrace{\begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}}_{=\mathbf{E}_3}. \end{aligned} \quad (22)$$

Les matrices $\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3$, appelées générateurs, forment la base de l'algèbre de Lie $so(3)$ associée à $SO(3)$. Notons que pour un jeu de paramètres $\mathbf{p} = (p_1, p_2, p_3)$, la transformation correspondante est

$$\mathbf{P} = \exp(p_1 \mathbf{E}_1 + p_2 \mathbf{E}_2 + p_3 \mathbf{E}_3) \quad (23)$$

Or

$$\mathbf{P} \neq \underbrace{\exp(p_1 \mathbf{E}_1) \cdot \exp(p_2 \mathbf{E}_2) \cdot \exp(p_3 \mathbf{E}_3)}_{=\mathbf{P}(p_1, p_2, p_3)} \quad (24)$$

Ainsi dans la représentation exponentielle (23), la rotation et la translation se font de façon simulatée et non pas l'une après l'autre, comme c'est le cas dans la seconde expression (24). Bien sûr, le comportement de l'expression (23) est préférable car il traduit plus la réalité du mouvement du corps rigide qui tourne et se déplace en même temps. Mais surtout, cette représentation exponentielle nous amènera à une plus grande précision lors du calcul de la trajectoire.

Pour illustrer ce gain en précision, prenons une voiture de Dubins [13]

$$\begin{cases} \dot{x}_1 &= u_1 \cdot \cos x_3 \\ \dot{x}_2 &= u_1 \cdot \sin x_3 \\ \dot{x}_3 &= u_2 \end{cases} \quad (25)$$

Le vecteur d'état $\mathbf{x} = (x_1, x_2, x_3)$, appelé *pose*, peut être représenté par un élément $\mathbf{P} \in SE(2)$. Pour l'intégration, nous pouvons donc utiliser une formule proche de celle que nous avons utilisée pour la rotation dans $SO(3)$. Ici, elle s'écrit :

$$\mathbf{P}_{t+dt} = \mathbf{P}_t \cdot e^{dt \cdot (u_1 \mathbf{E}_1 + u_2 \mathbf{E}_3)} \quad (26)$$

où $u_1 \mathbf{E}_1 + u_2 \mathbf{E}_3$ représente la variation de la pose dans le repère du robot.

Utilisons une simulation pour illustrer ce que l'on gagne en précision. Prenons les entrées et le vecteur initial suivant :

$$\begin{aligned} u_1(t) &= 1 \\ u_2(t) &= 1 \\ \mathbf{x}(0) &= (1, 0, \frac{\pi}{2})^\top \end{aligned} \quad (27)$$

et intégrons la trajectoire en prenant trois approches ; l'approche exacte, la méthode d'Euler et l'intégration exponentielle. Nous prendrons $t \in [0, 6]$ et une période d'échantillonnage $dt = 0.5$. Dans ce cas, la trajectoire exacte se calcule facilement. Elle est donnée par

$$\begin{cases} x_1(t) = \cos t \\ x_2(t) = \sin t \\ x_3(t) = \frac{\pi}{2} + t \end{cases} \quad (28)$$

Une intégration par Euler nous amène au schéma d'intégration suivant :

$$\begin{cases} x_1(t+dt) = x_1(t) + dt \cdot u_1(t) \cdot \cos x_3(t) \\ x_2(t+dt) = x_2(t) + dt \cdot u_1(t) \cdot \sin x_3(t) \\ x_3(t+dt) = x_3(t) + dt \cdot u_2(t) \end{cases} \quad (29)$$

La figure 2 (b) montre une différence notable avec la solution exacte lorsque dt est grand. En revanche, l'intégration exponentielle $\mathbf{P}_{t+dt} = \mathbf{P}_t \cdot e^{dt \cdot (u_1 E_1 + u_2 \cdot E_3)}$ illustrée par la sous figure (c) est presque identique à la solution exacte.

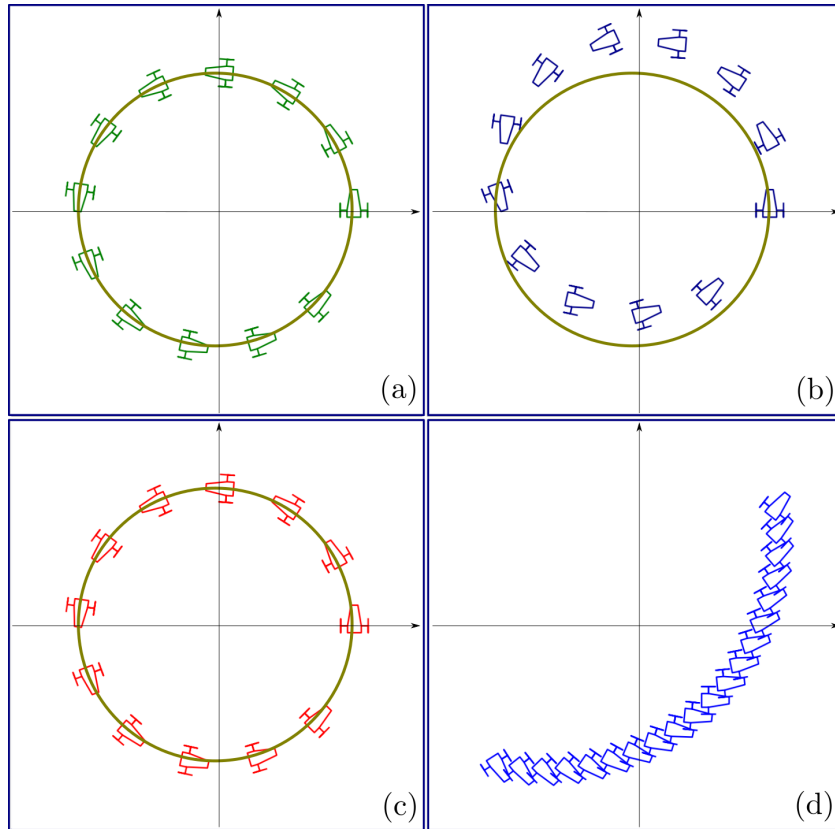


FIGURE 2 – (a) trajectoire exacte, (b) integration par Euler, (c) integration exponentielle, (d) chemin entre $\mathbf{P}_0$ et $\mathbf{P}_1$

Afin d'illustrer dans un tel contexte l'intérêt du logarithme, nous allons chercher une trajectoire qui relie deux poses. Cela signifie que nous voulons trouver un chemin, $\mathbf{P}(t), t \in [0, 1]$ dans $SE(2)$ tel que $\mathbf{P}(0) = \mathbf{P}_0, \mathbf{P}(1) = \mathbf{P}_1$, où $\mathbf{P}_0$ et $\mathbf{P}_1$ sont les deux matrices poses associées aux vecteurs $\mathbf{x}(0) = (-1, -1, -1)^T$ et $\mathbf{x}(1) = (1, 1, 1)^T$. Nous appliquons la formule d'interpolation donnée par

$$\mathbf{P}(t) = e^{\log(t \cdot \mathbf{P}_1 \mathbf{P}_0^{-1})} \cdot \mathbf{P}_0$$

Le résultat est donné par la figure 2 (d). Nous remarquons que la trajectoire n'a aucune raison d'être compatible avec un chemin possible pour la voiture de Dubins qui elle ne dérape pas latéralement. On peut montrer que la trajectoire correspondant à cette interpolation est une *géodésique*. C'est la plus courte des trajectoires dans l'espace $SE(2)$ mais par forcément la plus courte dans le monde réel euclidien.

3.9 Quelques exemples

Avant de passer à des robots tridimensionnels, nous allons ici considérer des exemples illustrant les notions définies précédemment.

3.9.1 La voiture sur la sphère

Un modèle classique de voiture évoluant sur un plan est celui de Dubin's [13] décrit par :

$$\begin{cases} \dot{x}_1 &= x_4 \cos x_3 \\ \dot{x}_2 &= x_4 \sin x_3 \\ \dot{x}_3 &= u_1 \\ \dot{x}_4 &= u_2 \end{cases} \quad (30)$$

où (x_1, x_2, x_3) est la pose, x_4 la vitesse, u_1 la vitesse de rotation et u_2 l'accélération (voir figure 3 à gauche).

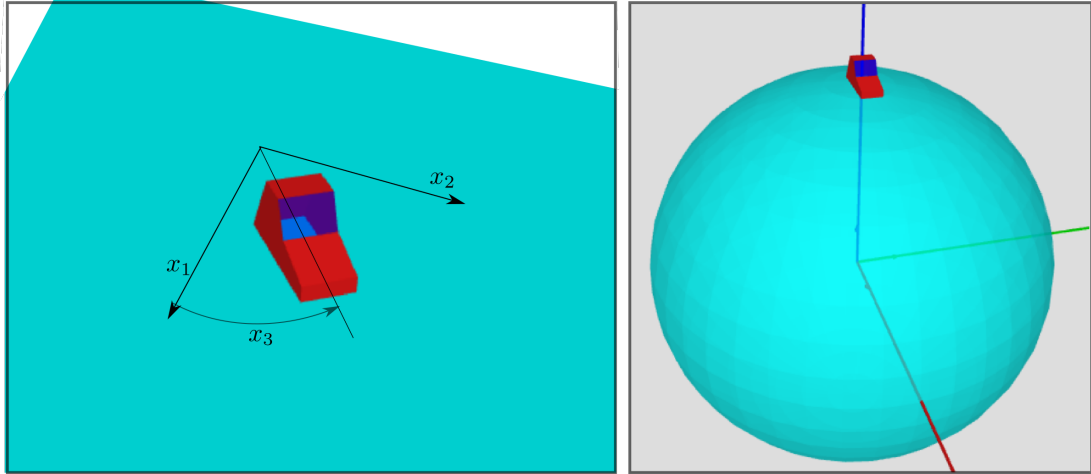


FIGURE 3 – Gauche : voiture sur le plan ; Droite : sur la sphère

Prenons par exemple l'entrée suivante

$$\mathbf{u} = \begin{pmatrix} 0.1 \cos(0.1 t) + 0.02 \\ 1 - x_4 \end{pmatrix} \quad (31)$$

avec pour état initial $\mathbf{x} = (0, 0, 0, 0)$.

Du fait que deux caps x_3^a et x_3^b sont identiques si $\frac{x_3^a - x_3^b}{2\pi} \in \mathbb{Z}$, le vecteur d'état $\mathbf{x} = (x_1, x_2, x_3, x_4)$ appartient à l'espace $\mathbb{R}^2 \times \frac{\mathbb{R}}{2\pi\mathbb{Z}} \times \mathbb{R}$. L'ensemble $\frac{\mathbb{R}}{2\pi\mathbb{Z}}$ est appelé $SO(2)$ et correspond au cercle unité. Une simulation pour $t \in [0, 200]$ est donnée par la figure 4.

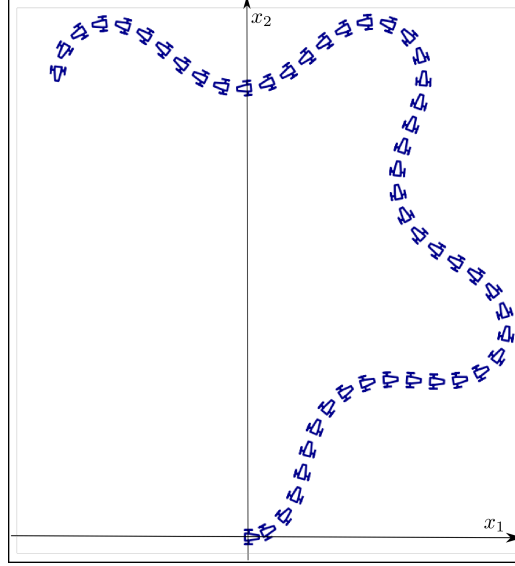


FIGURE 4 – Simulation de la voiture sur la plan. Le rectangle correspond à $[-50, 50] \times [-1, 80]$

Prenons maintenant la même voiture, avec la même entrée $\mathbf{u}$, mais qui évolue sur une sphère de rayon $r = 20\text{m}$ (voir figure 3, à droite). Comme illustré par la figure 5, à chaque pose de la voiture sur la sphère, on peut associer une orientation unique (représentée par la voiture au centre de la sphère et réciproquement). Par conséquent, l'espace d'état est donné par $SO(3) \times \mathbb{R}$.

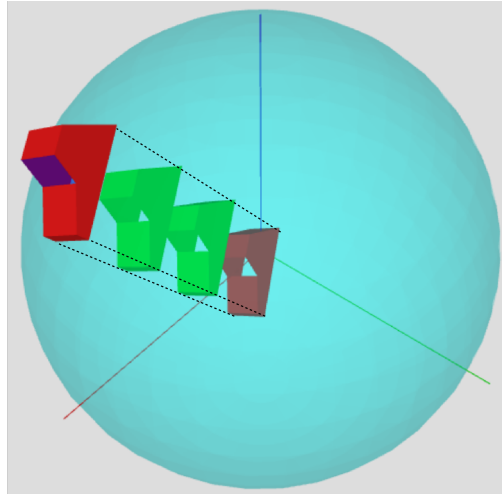


FIGURE 5 – L'ensemble des poses correspond à $SO(3)$

L'ensemble des poses pour notre voiture sur la sphère est donc le groupe de Lie $SO(3)$ et une pose peut donc être représentée par une matrice de rotation $\mathbf{R}$. Si x_4 est la vitesse, nous avons

$$\omega_r = (0, \frac{x_4}{r}, u_1). \quad (32)$$

La première composante est à zéro car il n'y a pas de roulis, la deuxième $\frac{x_4}{r}$ vient de la courbure de la sphère, et la troisième u_1 est celle que l'on commande. Ainsi l'état de la voiture est de la forme $(\mathbf{R}(t), x_4)$, et on pourra la simuler par le schéma d'intégration suivant :

$$\begin{aligned} \mathbf{R}(t + dt) &= \mathbf{R}(t) e^{dt \omega_r(t)} \\ x_4(t + dt) &= x_4(t) + dt u_2(t). \end{aligned} \quad (33)$$

La figure 6 à gauche montre qu'à l'instant initial la voiture accélère du fait de la relation $u_2 = 1 - x_4$. Sur la vue globale à droite, on observe une boucle, ce qui n'était pas le cas sur le plan. Dans cette représentation de la pose par une matrice de rotation, nous n'avons aucune singularité possible, même aux pôles, ce qui n'aurait pas été le cas avec une représentation plus classique de la sphère utilisant des angles (angles d'Euler par exemple).

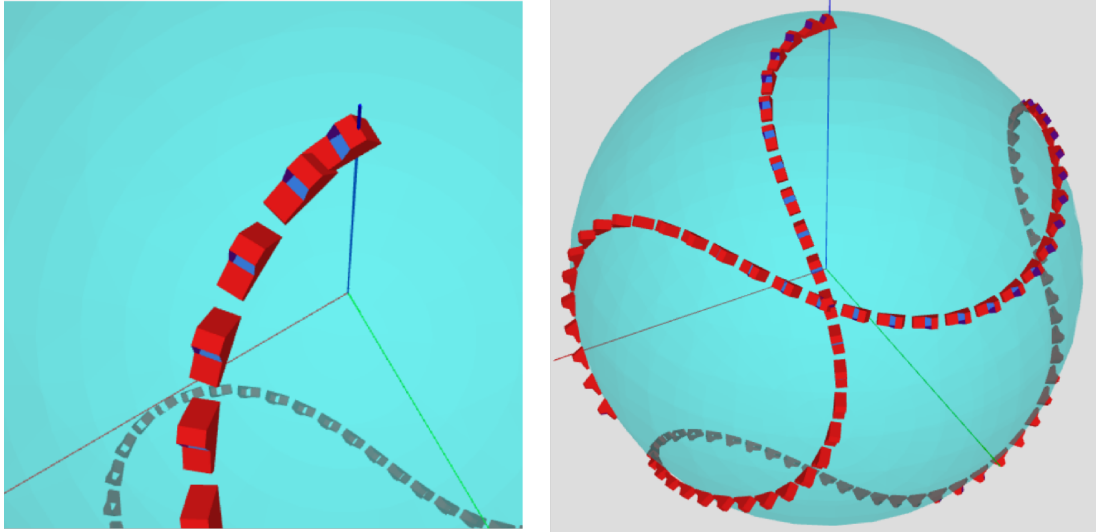


FIGURE 6 – Simulation sur la sphère. Gauche : trajectoire près du point initial ; Droite : vue globale

3.9.2 Voiture sur le tore

Considérons maintenant la voiture évoluant sur un tore paramétrisé par les deux rayons r_1, r_2 , comme illustré par la figure 7. Le tore a deux types de cercles caractéristiques : les méridiens de rayon r_2 et les parallèles. La pose peut être représentée par trois angles x_1, x_2, x_3 . Supposons que la vitesse de la voiture est égale à $1 m s^{-1}$. La voiture a donc une seule entrée u correspondant à sa vitesse de rotation scalaire relativement au sol.

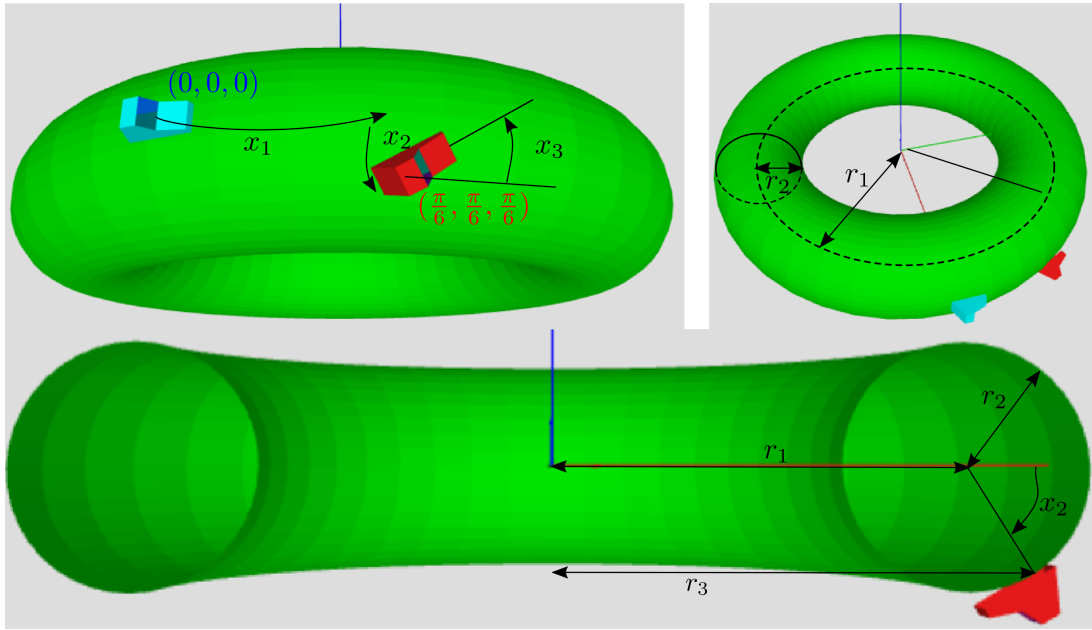


FIGURE 7 – Représentation de la pose de la voiture sur le tore

L'espace d'état est donc de la forme $\frac{\mathbb{R}}{2\pi\mathbb{Z}} \times \frac{\mathbb{R}}{2\pi\mathbb{Z}} \times \frac{\mathbb{R}}{2\pi\mathbb{Z}}$. Ainsi, une implémentation à base d'angle peut être effectuée sans singularité et sans avoir recours à $SO(3)$. Si la voiture est à la position (x_1, x_2) alors les parallèles sont des cercles de rayon $r_3 = r_1 + r_2 \cos x_2$.

Les équations d'état sont alors les suivantes :

$$\begin{cases} \dot{x}_1 = \frac{\cos x_3}{r_3} \\ \dot{x}_2 = -\frac{\sin x_3}{r_2} \\ \dot{x}_3 = u + \sin x_2 \frac{\cos x_3}{r_3} \end{cases} \quad (34)$$

Les deux premières équations sont illustrées par la figure 8, à gauche. Comme illustré par la figure 8 à droite, si à $t = 0$, l'angle de cap x_3 est nul et si $u = 0$ alors x_3 ne reste pas nul. Il augmente d'un facteur $\sin x_2$ $\dot{x}_1 = \sin x_2 \frac{\cos x_3}{r_3}$.

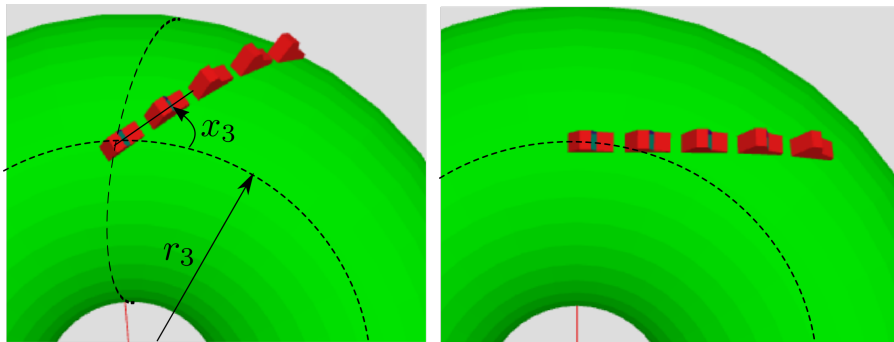


FIGURE 8 – Gauche : la voiture va tout droit avec un cap initial x_3 non nul ; Droite : La voiture va tout droit et quitte le parallèle

La figure 9 propose une simulation pour $r_1 = 10, r_2 = 6$ et $t \in [0, 100]$. L'état initial est $x_1 = x_2 = x_3 = 0$. Le contrôle est pris à $u = 0.1$. Une méthode d'intégration de type Runge-Kutta convient parfaitement pour effectuer cette simulation.

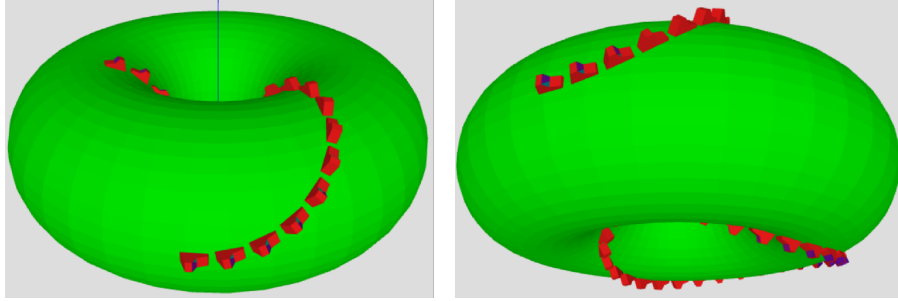


FIGURE 9 – Deux vues différentes pour la même trajectoire de la voiture

A travers cet exemple, nous avons voulu montrer que l'implémentation de l'intégration avec des orientations tridimensionnelles ne demandent pas forcément de faire appel à $SO(3)$. La méthode d'intégration est donc fortement liée à la géométrie de l'espace des poses pour le drone.

4 Modélisation d'un drone

4.1 Modélisation cinématique

Considérons un objet rigide et mobile (notre drone) dont les entrées sont les accélérations et le vecteur de rotation. Dans le cadre d'une modélisation cinématique, nous pouvons considérer que le vecteur d'état est composé

- du vecteur $\mathbf{p} = (p_x, p_y, p_z)$ qui donne les coordonnées du robot exprimées dans le repère inertiel (ici le sol) $\mathcal{R}_0$,
- l'orientation $\mathbf{R}$
- le vecteur vitesse $\mathbf{v}_r$ du mobile donné dans son propre repère $\mathcal{R}_1$.

Les entrées du système sont

- le vecteur d'accélération $\mathbf{a}_r = \mathbf{a}_{\mathcal{R}_1}$ du centre du robot donné dans $\mathcal{R}_1$ et
- le vecteur $\boldsymbol{\omega}_r = (\omega_x, \omega_y, \omega_z)$ correspondant au vecteur de rotation exprimé dans $\mathcal{R}_1$.

Il est en effet préférable d'exprimer les quantités $\mathbf{a}, \boldsymbol{\omega}, \mathbf{v}_r$, dans le repère du drone puisque qu'elles sont mesurées ainsi par les capteurs embarqués. L'évolution de la position s'écrit :

$$\dot{\mathbf{p}} = \mathbf{R} \mathbf{v}_r. \quad (35)$$

En différenciant, on obtient :

$$\ddot{\mathbf{p}} = \dot{\mathbf{R}} \mathbf{v}_r + \mathbf{R} \dot{\mathbf{v}}_r \quad (36)$$

soit en isolant $\dot{\mathbf{v}}_r$:

$$\begin{aligned} \dot{\mathbf{v}}_r &= \underbrace{\mathbf{R}^\top \ddot{\mathbf{p}}}_{\mathbf{a}_r} - \mathbf{R}^\top \dot{\mathbf{R}} \mathbf{v}_r \\ &= \mathbf{a}_r - \boldsymbol{\omega}_r \wedge \mathbf{v}_r. \end{aligned} \quad (37)$$

Or, nous avons $\mathbf{R}^\top \dot{\mathbf{R}} = \wedge (\mathbf{R}^\top \boldsymbol{\omega})$. Finalement

$$\begin{cases} \dot{\mathbf{p}} &= \mathbf{R} \mathbf{v} \\ \dot{\mathbf{R}} &= \mathbf{R} (\boldsymbol{\omega}_r \wedge) \\ \dot{\mathbf{v}}_r &= \mathbf{a}_r - \boldsymbol{\omega}_r \wedge \mathbf{v}_r \end{cases} \quad (38)$$

4.2 Avec la gravité

Dans le cas où le drone évolue dans un champ de gravité $\mathbf{g}(\mathbf{p})$ qui dépend de la position $\mathbf{p}$, l'accélération réelle $\mathbf{a}_r$ correspond à la somme de la gravité et de l'accélération mesurée $\mathbf{a}_r^{mes}$. Dans ce cas, on a

$$\begin{cases} \dot{\mathbf{p}} &= \mathbf{R} \mathbf{v}_r \\ \dot{\mathbf{R}} &= \mathbf{R}(\boldsymbol{\omega}_r \wedge) \\ \dot{\mathbf{v}}_r &= \mathbf{R}^T \mathbf{g}(\mathbf{p}) + \mathbf{a}_r^{mes} - \boldsymbol{\omega}_r \wedge \mathbf{v}_r \end{cases} \quad (39)$$

Les deux éléments qui sont mesurés avec des capteurs inertiels sont cette accélération $\mathbf{a}_r^{mes}$ (retournée par les accéléromètres) et le vecteur de rotation $\boldsymbol{\omega}_r$ (retourné par le gyroscope). Pour intégrer cette équation d'état, il faut connaître les conditions initiales et une expression pour $\mathbf{g}(\mathbf{p})$.

4.3 Centrale inertielle

Pour localiser un robot à l'estime [14], on peut se baser sur les équations (39), dites de *mécanisation*, et représentées par la figure 10. C'est ce que fait une centrale inertielle lorsqu'elle n'est pas hybridée avec un capteur externe (GPS par exemple). L'entrée du système $\mathbf{u} = (\mathbf{a}_r, \boldsymbol{\omega}_r)$ correspond au vecteur de mesure retourné par les gyromètres et les accéléromètres exprimé dans le repère du drone.

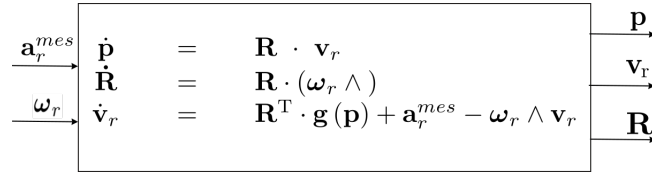


FIGURE 10 – Principe d'une centrale inertielle

Le schéma d'intégration d'Euler à suivre est le suivant

$$\begin{aligned} \mathbf{x}(t+dt) &= \mathbf{x}(t) + dt \mathbf{f}(\mathbf{x}(t), \mathbf{R}(t), \mathbf{u}(t)) \\ \mathbf{R}(t+dt) &= \mathbf{R}(t) e^{dt \boldsymbol{\omega}_r(t)} \end{aligned} \quad (40)$$

avec

$$\begin{aligned} \mathbf{x} &= (\mathbf{p}, \mathbf{v}_r) \\ \mathbf{f}(\mathbf{x}, \mathbf{R}, \mathbf{u}) &= \begin{pmatrix} \mathbf{R} \mathbf{v}_r \\ \mathbf{R}^T \mathbf{g}(\mathbf{p}) + \mathbf{a}_r^{mes} - \boldsymbol{\omega}_r \wedge \mathbf{v}_r \end{pmatrix} \end{aligned} \quad (41)$$

Notons qu'un schéma d'intégration exponentielle est utilisé pour l'orientation $\mathbf{R}(t)$. L'intégration par Euler est limitée à l'évolution de $\mathbf{x}$.

Avec un schéma d'intégration de Runge-Kutta pour l'évolution de $\mathbf{x}$, on obtient alors le schéma bloc de la Figure 11.

4.4 Modélisation dynamique

Un drone (avion, quadrirotor, sous-marin, bateau, voiture) peut être considéré comme un solide pour lequel les entrées sont les accélérations ou les vitesses de rotation. Cela nous amène alors au modèle cinématique décrit par la formule (39). Mais ces quantités cinématiques sont généralement des fonctions analytiques des forces qui sont à l'origine du mouvement du drone.

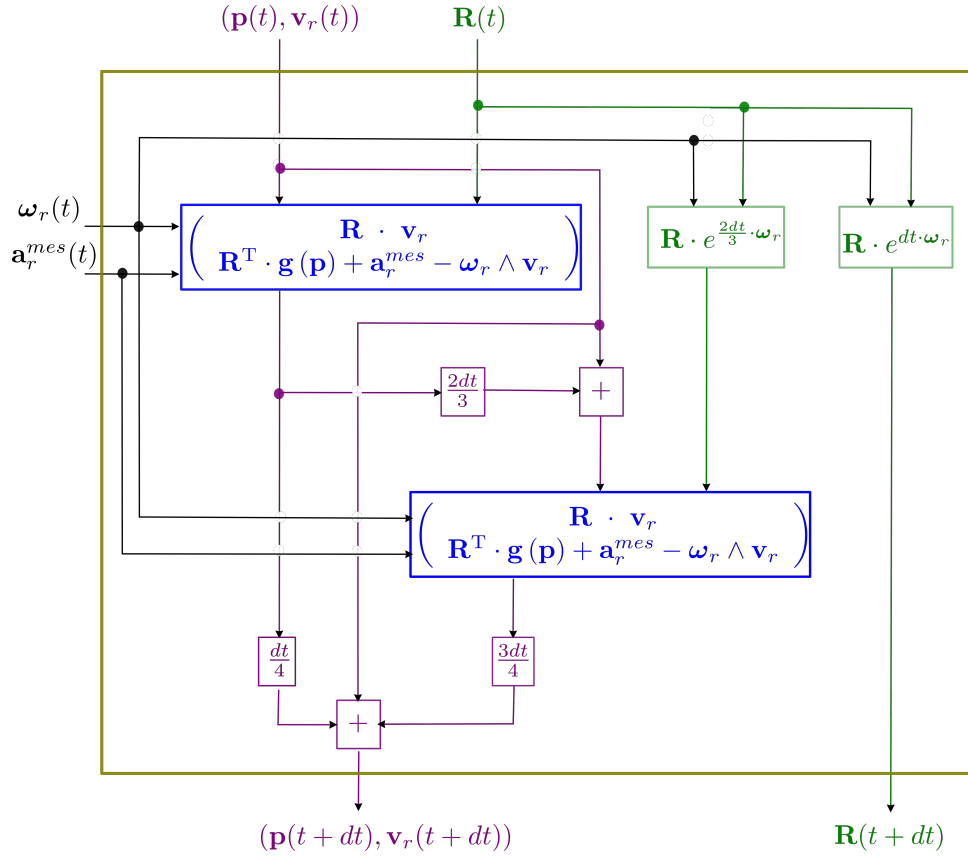


FIGURE 11 – Schéma de Runge Kutta pour l'intégration des équations de mécanisation de la centrale

Pour les robots marins et sous-marins, l'ouvrage de référence pour la modélisation est le livre de Fossen [15], mais la modélisation proposée est valide pour d'autres types de drones après quelques adaptations mineures. Le principe est de reprendre les équations de la cinématique et d'y rajouter l'interaction entre les forces et les accélérations. Ce lien se fait par la deuxième loi de Newton. Ainsi, si $\mathbf{f}$ est une force externe et si m est la masse du drone alors, vu du repère inertiel, on a

$$m\ddot{\mathbf{p}} = \mathbf{f}. \quad (42)$$

Du fait que les vitesses et les accélérations mesurées sont généralement exprimées dans le repère du drone, on préférera écrire cette relation dans son repère :

$$m\mathbf{a}_r = \mathbf{f}_r. \quad (43)$$

Le même type de relation existe pour la rotation du drone. Elle correspond à l'équation d'Euler pour les corps en rotation et est donnée par

$$\mathbf{I}\dot{\boldsymbol{\omega}}_r + \boldsymbol{\omega}_r \wedge (\mathbf{I}\boldsymbol{\omega}_r) = \boldsymbol{\tau}_r \quad (44)$$

où $\boldsymbol{\tau}_r$ est le couple appliqué sur le drone et $\boldsymbol{\omega}_r$ est sa vitesse de rotation. Les deux sont exprimées dans le repère du drone. La matrice d'inertie $\mathbf{I}$ est celle vue par le drone. On s'arrange pour que cette dernière soit diagonale en choisissant correctement le repère attaché au drone. Rappelons que la matrice d'inertie d'un corps solide qui occupe

un volume V est donnée par

$$\mathbf{I} = \int_V \rho(x, y, z) \begin{pmatrix} y^2 + z^2 & -xy & -xz \\ -xy & x^2 + z^2 & -yz \\ -xz & -yz & x^2 + y^2 \end{pmatrix} dx dy dz \quad (45)$$

où $\rho(x, y, z)$ est la densité de masse. Ainsi, les équations d'un drone peuvent s'exprimer sous la forme

$$\begin{cases} \dot{\mathbf{p}} &= \mathbf{R} \mathbf{v}_r & (i) \\ \dot{\mathbf{R}} &= \mathbf{R} (\boldsymbol{\omega}_r \wedge) & (ii) \\ \dot{\mathbf{v}}_r &= \mathbf{R}^\top \mathbf{g} + \frac{1}{m} \mathbf{f}_r - \boldsymbol{\omega}_r \wedge \mathbf{v}_r & (iii) \\ \dot{\boldsymbol{\omega}}_r &= \mathbf{I}^{-1} (\boldsymbol{\tau}_r - \boldsymbol{\omega}_r \wedge \mathbf{I} \boldsymbol{\omega}_r) & (iv) \end{cases} \quad (46)$$

où $\mathbf{p}$ est le vecteur de position, $\mathbf{R}$ est la matrice d'orientation, $\mathbf{v}_r$ est la vitesse du robot exprimée dans son propre repère, $\boldsymbol{\omega}_r$ est le vecteur de rotation du drone également exprimé dans son repère et $\mathbf{g} = (0, 0, g)$ est le vecteur de gravité. Les trois premières équations (i), (ii), (iii) sont issues du modèle cinématique (voir l'équation 39). Dans l'équation (iii), l'accélération $\mathbf{a}_r$ (exprimée dans le repère du drone) a été remplacée par $\frac{1}{m} \mathbf{f}_r$ (voir équation 43). L'équation (iv) est celle d'Euler (44) où $\boldsymbol{\tau}_r$ est le couple exercé sur le drone.

4.5 Phénomène de Dzhanibekov

Dans le cas des drones, l'interprétation de l'équation d'Euler pour les corps en rotation est fondamentale, surtout si l'on s'intéresse à l'aspect géométrique de la modélisation. En effet, dans le cas des drones, c'est en contrôlant l'orientation que l'on arrive à se déplacer dans la bonne direction.

Afin de nous focaliser sur l'orientation, nous allons considérer un solide en rotation dont le centre de gravité reste à l'origine. Dans ce cas, les équations deviennent

$$\begin{cases} \dot{\mathbf{R}} &= \mathbf{R} (\boldsymbol{\omega}_r \wedge) \\ \dot{\boldsymbol{\omega}}_r &= \mathbf{I}^{-1} (\boldsymbol{\tau}_r - \boldsymbol{\omega}_r \wedge \mathbf{I} \boldsymbol{\omega}_r) \end{cases} \quad (47)$$

Nous voulons mettre en évidence l'effet Dzhanibekov qui correspond à une instabilité dans la rotation du solide uniquement suivant le deuxième axe principal de la matrice d'inertie. Pour cela, considérons le parallélépipède de masse m et de densité uniforme représenté sur la figure 12.

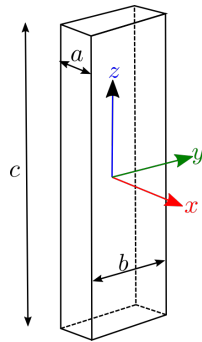


FIGURE 12 – Corps solide choisi pour illustrer l'effet Dzhanibekov

La matrice d'inertie se calcule facilement. Elle est donnée par

$$\begin{aligned} \mathbf{I} &= \frac{m}{12} \begin{pmatrix} b^2 + c^2 & 0 & 0 \\ 0 & a^2 + c^2 & 0 \\ 0 & 0 & a^2 + b^2 \end{pmatrix} \\ &= \begin{pmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{pmatrix} \end{aligned} \quad (48)$$

L'équation d'Euler $\dot{\omega}_r = -\mathbf{I}^{-1} (\omega_r \wedge (\mathbf{I} \omega_r))$ se décline ici en

$$\begin{aligned} \dot{\omega}_1 &= \frac{I_2 - I_3}{I_1} \omega_2 \omega_3 = \alpha_1 \omega_2 \omega_3 \\ \dot{\omega}_2 &= \frac{I_3 - I_1}{I_2} \omega_1 \omega_3 = \alpha_2 \omega_1 \omega_3 \\ \dot{\omega}_3 &= \frac{I_1 - I_2}{I_3} \omega_1 \omega_2 = \alpha_3 \omega_1 \omega_2 \end{aligned} \quad (49)$$

où $\omega_r = (\omega_1, \omega_2, \omega_3)$ est le vecteur de rotation exprimé dans le repère du corps. Etudions la stabilité de ces équations. Autour d'une situation d'équilibre nous avons $\dot{\omega}_1 = \dot{\omega}_2 = \dot{\omega}_3 = 0$. Si a, b, c sont différents, cela se traduit par

$$\begin{aligned} \omega_2 \omega_3 &= 0 \\ \omega_1 \omega_3 &= 0 \\ \omega_1 \omega_2 &= 0 \end{aligned} \quad (50)$$

ce qui signifie que parmi $\omega_1, \omega_2, \omega_3$ deux au moins doivent être nuls. Pour un corps solide, si $\forall i, \omega_i \neq 0$, les frictions internes au corps vont se produire jusqu'à temps d'atteindre un point d'équilibre stable.

Linéarisons autour d'un point d'équilibre $\bar{\omega}$:

$$\begin{pmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \end{pmatrix} \simeq \begin{pmatrix} \bar{\omega}_1 \\ \bar{\omega}_2 \\ \bar{\omega}_3 \end{pmatrix} + \begin{pmatrix} 0 & \alpha_1 \bar{\omega}_3 & \alpha_1 \bar{\omega}_2 \\ \alpha_2 \bar{\omega}_3 & 0 & \alpha_2 \bar{\omega}_1 \\ \alpha_3 \bar{\omega}_2 & \alpha_3 \bar{\omega}_1 & 0 \end{pmatrix} \begin{pmatrix} \omega_1 - \bar{\omega}_1 \\ \omega_2 - \bar{\omega}_2 \\ \omega_3 - \bar{\omega}_3 \end{pmatrix}. \quad (51)$$

Prenons par exemple l'équilibre correspondant à $\bar{\omega} = (\bar{\omega}_1, 0, 0)$. Par permutation, on en déduit les résultats pour les deux autres axes. On obtient

$$\begin{pmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \end{pmatrix} \simeq \begin{pmatrix} \bar{\omega}_1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & \bar{\omega}_1 \alpha_2 \\ 0 & \bar{\omega}_1 \alpha_3 & 0 \end{pmatrix} \begin{pmatrix} \omega_1 - \bar{\omega}_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}. \quad (52)$$

Le polynôme caractéristique est

$$P(s) = s(s^2 - \bar{\omega}_1^2 \alpha_2 \alpha_3). \quad (53)$$

La racine $s = 0$, associée au vecteur propre $(1, 0, 0)$ traduit la conservation du moment cinétique. Le système est instable si au moins une des racines de $P(s)$ est à partie réelle négative, c'est à dire si $\alpha_2 \alpha_3 > 0$. Or,

$$\begin{aligned} \alpha_2 \alpha_3 > 0 &\Leftrightarrow \frac{I_3 - I_1}{I_2} \frac{I_1 - I_2}{I_3} > 0 \\ &\Leftrightarrow (I_3 - I_1)(I_1 - I_2) > 0 \end{aligned} \quad (54)$$

Ce qui signifie que $I_2 < I_1 < I_3$ ou bien $I_3 < I_1 < I_2$. Si $\alpha_2 \alpha_3 < 0$ les racines sont

$$s = \pm j \bar{\omega}_1 \sqrt{-\alpha_2 \alpha_3} = \pm j \bar{\omega}_1 \sqrt{-\frac{(I_3 - I_1)(I_1 - I_2)}{I_2 I_3}} \quad (55)$$

ce qui correspond à la pulsation liée à la précession.

Nous venons donc de montrer le *théorème de l'axe intermédiaire* (ou *tennis racket theorem*) qui dit que la rotation d'un corps rigide est stable autour de ses premier et troisième axes, mais instable relativement à son deuxième axe.

Simulation. Pour la simulation, nous pouvons utiliser une méthode de Runge-Kutta pour l'évolution de ω_r et une intégration exponentielle pour l'orientation. Cela donne :

$$\begin{aligned}\omega_r(t+dt) &= \omega_r(t) + \frac{dt}{4} \mathbf{f}(\omega_r(t)) + \frac{3dt}{4} \mathbf{f}(\omega_r(t) + \frac{2dt}{3} \mathbf{f}(\omega_r(t))) \\ \mathbf{R}(t+dt) &= \mathbf{R}(t) e^{dt \omega_r(t)}\end{aligned}\tag{56}$$

avec $\mathbf{f}(\omega_r) = \mathbf{I}^{-1} (-\omega_r \wedge (\mathbf{I} \omega_r))$. Le schéma d'intégration est illustré par la figure 13.

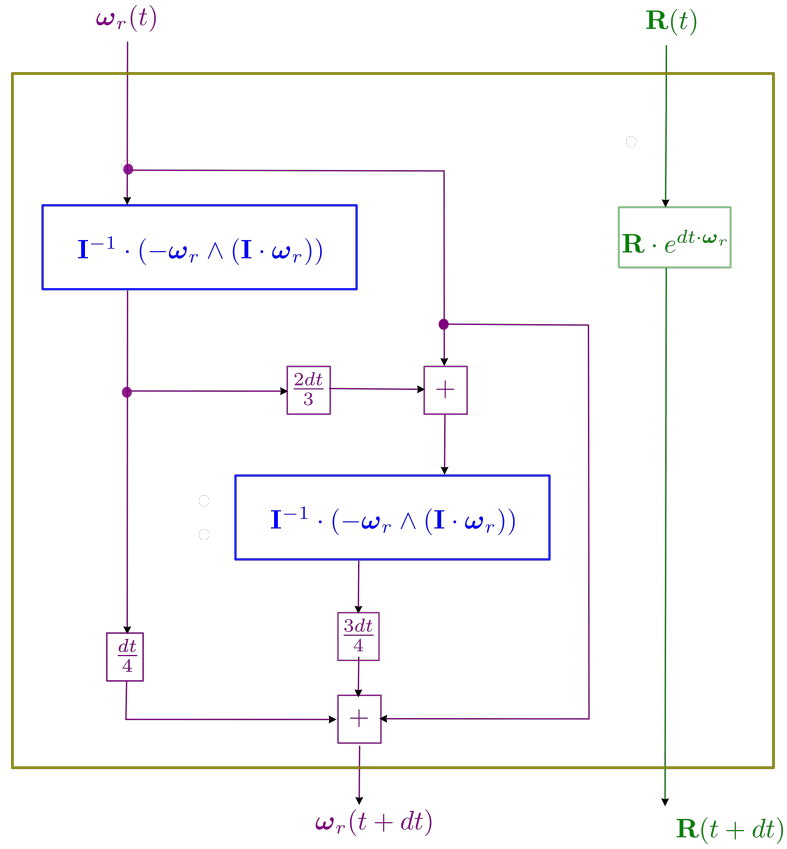


FIGURE 13 – Méthode d'intégration utilisée pour simuler la rotation du solide

Prenons $a = 0.4, b = 1, c = 3, m = 1$ afin d'illustrer la rotation instable du parallélépipède selon son deuxième axe. Sur la simulation tridimensionnelle, on peut observer le parallélépipède qui tourne et change son orientation de façon périodique. L'évolution des composantes de ω_r est donnée par la figure 14.

Le programme ainsi qu'une vidéo illustrant l'effet Dzhanibekov est disponible sur <https://www.ensta-bretagne.fr/jaulin/teching.html>

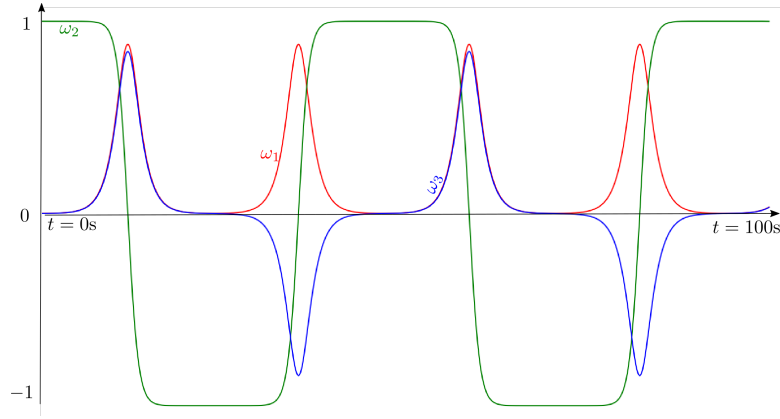


FIGURE 14 – La courbe verte montre que périodiquement le parallélépipède change radicalement son orientation relativement au second axe

4.6 Champ de vecteur d'Euler

Nous cherchons ici à donner une interprétation géométrique sur le comportement d'un solide en rotation. Nous repartons à nouveau de l'équation d'Euler pour un corps en rotation :

$$\dot{\omega}_r = -\mathbf{I}^{-1}(\omega_r \wedge (\mathbf{I}\omega_r)) \quad (57)$$

où ω_r est le vecteur de rotation exprimé dans le repère du solide et $\mathbf{I}$ est la matrice d'inertie que nous supposons donnée par

$$\mathbf{I} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix} \quad (58)$$

L'énergie cinétique donnée par

$$E_K = \frac{1}{2} \omega_r^T \mathbf{I} \omega_r \quad (59)$$

est constante. En effet,

$$\begin{aligned} \dot{E}_K &= \frac{d}{dt} \left(\frac{1}{2} \omega_r^T \mathbf{I} \omega_r \right) \\ &= \frac{1}{2} (2 \omega_r^T \mathbf{I}) \dot{\omega}_r \\ &= -\omega_r^T \mathbf{I} \mathbf{I}^{-1} (\omega_r \wedge (\mathbf{I} \omega_r)) \\ &= -\omega_r^T (\omega_r \wedge (\mathbf{I} \omega_r)) = 0 \end{aligned} \quad (60)$$

L'égalité à zéro vient du fait que ω_r et $\omega_r \wedge (\mathbf{I} \omega_r)$ sont orthogonaux.

Pour analyser l'évolution en rotation du solide, nous allons considérer une énergie constante par exemple $E_0 = 10\text{J}$. L'ensemble des vecteurs de rotation associés forme un ellipsoïde $\mathcal{E}$ sur lequel nous pouvons tracer le champ de vecteurs décrivant l'évolution de ω_r . Cela est illustré par les figures 15 et 16. En prenant $\omega_r(0) = (4, \sqrt{2}, 0)$ nous obtenons la trajectoire de la figure 15. Nous notons que $\omega_r(t)$ tourne autour de ω_1 ce qui traduit un phénomène de précession. Pour $\omega_r(0) = (0.01, \sqrt{10}, 0)$ nous observons la trajectoire de la figure 16. Le vecteur $\omega_r(t)$ alterne entre une rotation suivant ω_2 et une rotation suivant $-\omega_2$. Cette alternance traduit l'effet Dzhanibekov. Une telle trajectoire qui part d'un point d'équilibre et se déplace vers un autre points d'équilibre est appelée *orbite hétérocline*.

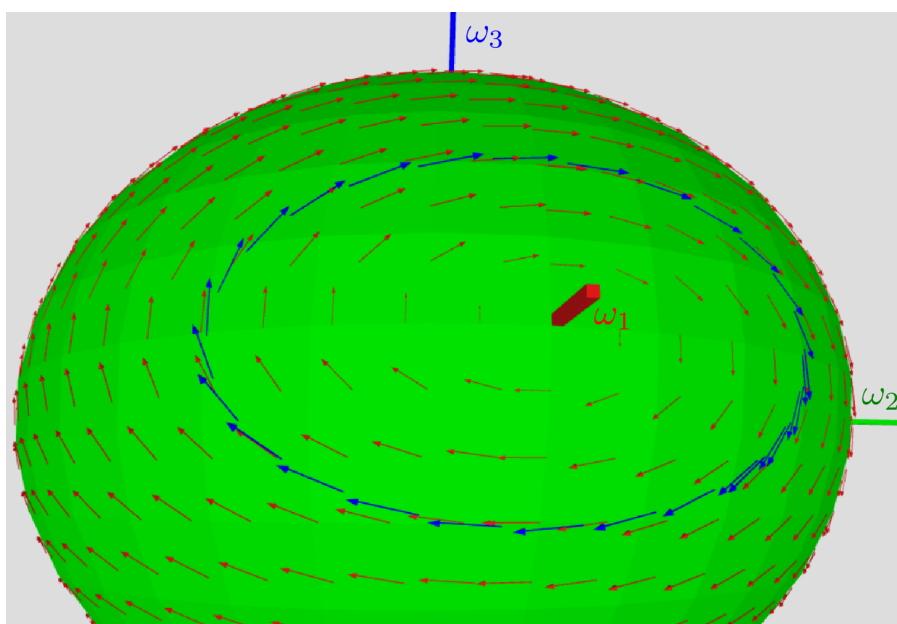


FIGURE 15 – Trajectoire traduisant une précession autour de l'axe ω_1

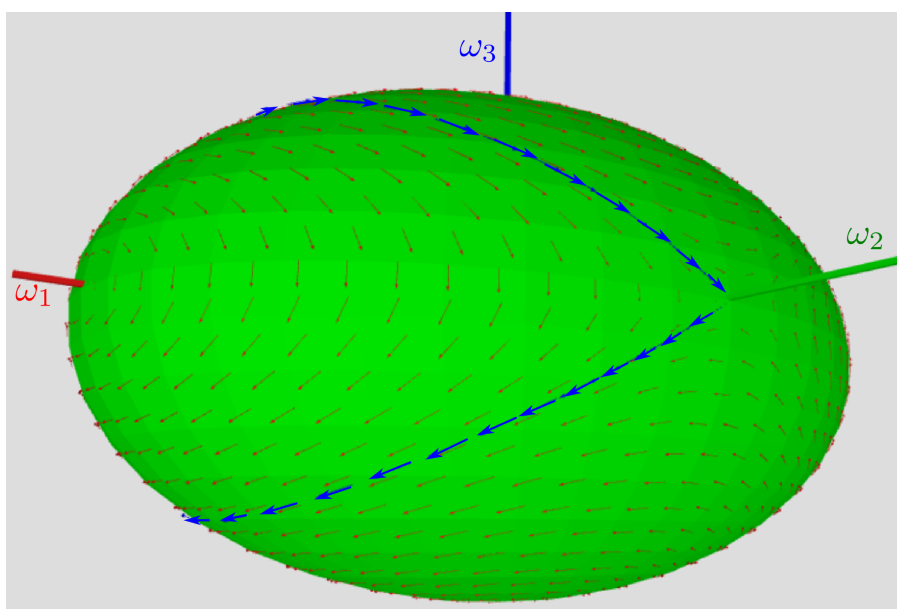


FIGURE 16 – Orbite hétérocline traduisant un effet Dzhanibekov

5 Contrôle inertiel

Ce qui distingue un véhicule d'un drone, c'est principalement l'autonomie. Cette dernière se fait par un régulateur qui est synthétisé à partir des équations du drone (46). Rappelons que ces dernières sont données par :

$$\begin{cases} \dot{\mathbf{p}} &= \mathbf{R} \mathbf{v}_r \\ \dot{\mathbf{R}} &= \mathbf{R} (\boldsymbol{\omega}_r \wedge) \\ \dot{\mathbf{v}}_r &= \mathbf{R}^\top \mathbf{g} + \frac{1}{m} \mathbf{f}_r - \boldsymbol{\omega}_r \wedge \mathbf{v}_r \\ \dot{\boldsymbol{\omega}}_r &= \mathbf{I}^{-1} (\boldsymbol{\tau}_r - \boldsymbol{\omega}_r \wedge \mathbf{I} \boldsymbol{\omega}_r) \end{cases} \quad (61)$$

Ce modèle du drone est qualifié d'*inertiel* car il ne fait intervenir que deux paramètres : la masse m et la matrice d'inertie $\mathbf{I}$, qui sont tous les deux de nature inertielle. La synthèse d'un régulateur fondé sur ce modèle ne nécessite la connaissance que de ces deux quantités et est lui aussi qualifié d'*inertiel*.

Nous cherchons à choisir les forces $\mathbf{f}_r$ et $\boldsymbol{\tau}_r$ permettant de suivre une pose virtuelle représentée par la position désirée $\mathbf{p}_d(t)$ et l'orientation désirée $\mathbf{R}_d(t)$. Nous supposons que nous avons une expression analytique pour $\mathbf{p}_d(t)$ et $\mathbf{R}_d(t)$, ce qui nous permet d'accéder à leurs dérivées $\dot{\mathbf{p}}_d, \ddot{\mathbf{p}}_d, \dot{\mathbf{R}}_d, \ddot{\mathbf{R}}_d$.

5.1 Commande en accélération

Proposons le bouclage suivant :

$$\begin{cases} \mathbf{f}_r &= m \dot{\mathbf{v}}_r^d - m \mathbf{R}^\top \mathbf{g} + m \boldsymbol{\omega}_r \wedge \mathbf{v}_r \\ \boldsymbol{\tau}_r &= \mathbf{I} \dot{\boldsymbol{\omega}}_r^d + \boldsymbol{\omega}_r \wedge \mathbf{I} \boldsymbol{\omega}_r \end{cases} \quad (62)$$

où $\dot{\mathbf{v}}_r^d$ et $\dot{\boldsymbol{\omega}}_r^d$ sont les nouvelles entrées, les valeurs désirées pour $\dot{\mathbf{v}}_r$ and $\dot{\boldsymbol{\omega}}_r$. Le système (61) se simplifie et est maintenant décrit par :

$$\begin{cases} \dot{\mathbf{p}} &= \mathbf{R} \mathbf{v}_r \\ \dot{\mathbf{R}} &= \mathbf{R} (\boldsymbol{\omega}_r \wedge) \\ \dot{\mathbf{v}}_r &= \dot{\mathbf{v}}_r^d \\ \dot{\boldsymbol{\omega}}_r &= \dot{\boldsymbol{\omega}}_r^d \end{cases} \quad (63)$$

Nous avons maintenant un modèle à contrôler qui est purement cinématique. Les paramètres $m, \mathbf{I}$ ont été supprimés par le bouclage.

5.2 Positionneur

Le *positionneur* est le régulateur qui permet de calculer l'accélération $\dot{\mathbf{v}}_r$ à appliquer au drone pour qu'il rejoigne la position désirée $\mathbf{p}_d(t)$. Définissons l'erreur

$$\mathbf{e} = \mathbf{p}_d - \mathbf{p}. \quad (64)$$

Pour annuler cette erreur en choisissant la bonne accélération, nous proposons de prendre une commande proportionnelle et dérivée. Nous demandons à l'erreur de satisfaire l'équation

$$\ddot{\mathbf{e}} + \alpha_1 \dot{\mathbf{e}} + \alpha_0 \mathbf{e} = \mathbf{0}. \quad (65)$$

où α_0, α_1 sont choisis pour assurer la convergence exponentielle vers zéro de l'erreur.

Dans le repère du drone, cette équation devient $\ddot{\mathbf{e}}_r + \alpha_1 \dot{\mathbf{e}}_r + \alpha_0 \mathbf{e}_r = \mathbf{0}$, avec

$$\begin{aligned} \mathbf{e}_r &= \mathbf{R}^\top \mathbf{e} = \mathbf{R}^\top (\mathbf{p}_d - \mathbf{p}) \\ \dot{\mathbf{e}}_r &= \mathbf{R}^\top \dot{\mathbf{e}} = \mathbf{R}^\top (\dot{\mathbf{p}}_d - \dot{\mathbf{p}}) = \mathbf{R}^\top \dot{\mathbf{p}}_d - \mathbf{v}_r \\ \ddot{\mathbf{e}}_r &= \mathbf{R}^\top \ddot{\mathbf{e}} = \mathbf{R}^\top (\ddot{\mathbf{p}}_d - \ddot{\mathbf{p}}) \\ &= \mathbf{R}^\top \ddot{\mathbf{p}}_d - \mathbf{R}^\top (\dot{\mathbf{R}} \mathbf{v}_r + \mathbf{R} \dot{\mathbf{v}}_r) \\ &= \mathbf{R}^\top \ddot{\mathbf{p}}_d - \boldsymbol{\omega}_r \wedge \mathbf{v}_r - \dot{\mathbf{v}}_r \end{aligned} \quad (66)$$

Ainsi

$$\begin{aligned}
& \ddot{\mathbf{e}}_r + \alpha_1 \dot{\mathbf{e}}_r + \alpha_0 \mathbf{e}_r = \mathbf{0} \\
\Leftrightarrow & \mathbf{R}^\top \ddot{\mathbf{p}}_d - \boldsymbol{\omega}_r \wedge \mathbf{v}_r - \dot{\mathbf{v}}_r + \alpha_1 \dot{\mathbf{e}}_r + \alpha_0 \mathbf{e}_r = \mathbf{0} \\
\Leftrightarrow & \dot{\mathbf{v}}_r = \mathbf{R}^\top \ddot{\mathbf{p}}_d - \boldsymbol{\omega}_r \wedge \mathbf{v}_r + \alpha_1 \dot{\mathbf{e}}_r + \alpha_0 \mathbf{e}_r = \mathbf{0}
\end{aligned} \tag{67}$$

L'expression du positionneur est donc

$$\dot{\mathbf{v}}_r = \mathbf{R}^\top \ddot{\mathbf{p}}_d - \boldsymbol{\omega}_r \wedge \mathbf{v}_r + \alpha_1 (\mathbf{R}^\top \dot{\mathbf{p}}_d - \mathbf{v}_r) + \alpha_0 \mathbf{R}^\top (\mathbf{p}_d - \mathbf{p}). \tag{68}$$

Cela signifie que si nous sommes capables de produire les forces et les couples qui génèrent l'accélération $\dot{\mathbf{v}}_r$, alors l'erreur $\mathbf{e} = \mathbf{p}_d - \mathbf{p}$ convergera vers zéro.

5.3 Orienteur

L'*orienteur* est le contrôleur qui nous dit comment changer le vecteur de rotation pour suivre l'orientation désirée $\mathbf{R}_d(t)$.

Dans $SO(3)$, l'erreur d'orientation peut être représentée par $\mathbf{R}_d \mathbf{R}^\top$. Pour pouvoir manipuler cette erreur, il convient de repasser dans l'algèbre de Lie qui possède une structure d'espace vectoriel. Nous définissons donc l'erreur

$$\mathbf{e} = \text{Log}(\mathbf{R}_d \mathbf{R}^\top) \tag{69}$$

où $\mathbf{e}$ correspond au vecteur de rotation qu'il faut suivre pour aller de $\mathbf{R}$ à $\mathbf{R}_d$ en 1 seconde. Pour annuler cette erreur en choisissant l'accélération angulaire $\dot{\boldsymbol{\omega}}_r^d$ adaptée, nous pouvons à nouveau considérer une commande proportionnelle et dérivée. Pour cela, nous formons la dynamique de l'erreur :

$$\ddot{\mathbf{e}} + \alpha_1 \dot{\mathbf{e}} + \alpha_0 \mathbf{e} = \mathbf{0}, \tag{70}$$

où α_0, α_1 sont choisis pour assurer la stabilité. Or,

$$\begin{aligned}
\mathbf{e} \wedge &= \text{log}(\mathbf{R}_d \mathbf{R}^\top) \\
\dot{\mathbf{e}} \wedge &= \frac{d}{dt} \text{log}(\mathbf{R}_d \mathbf{R}^\top) \\
&= \mathbf{R} \mathbf{R}_d^\top \frac{d}{dt} (\mathbf{R}_d \mathbf{R}^\top) \\
&= \mathbf{R} \mathbf{R}_d^\top (\mathbf{R}_d \dot{\mathbf{R}}^\top + \dot{\mathbf{R}}_d \mathbf{R}^\top) \\
&= \underbrace{\mathbf{R} \mathbf{R}^\top}_{=-(\boldsymbol{\omega} \wedge)} + \mathbf{R} \underbrace{\mathbf{R}_d^\top \dot{\mathbf{R}}_d}_{=(\mathbf{R}_d^\top \boldsymbol{\omega}_d) \wedge} \mathbf{R}^\top \\
&= -(\boldsymbol{\omega} \wedge) + (\mathbf{R} \mathbf{R}_d^\top \boldsymbol{\omega}_d) \wedge
\end{aligned} \tag{71}$$

Donc

$$\begin{aligned}
\mathbf{e} &= \text{Log}(\mathbf{R}_d \mathbf{R}^\top) \\
\dot{\mathbf{e}} &= -\boldsymbol{\omega} + \mathbf{R} \mathbf{R}_d^\top \boldsymbol{\omega}_d \\
\ddot{\mathbf{e}} &= -\dot{\boldsymbol{\omega}} + \dot{\mathbf{R}} \mathbf{R}_d^\top \boldsymbol{\omega}_d + \mathbf{R} \dot{\mathbf{R}}_d^\top \boldsymbol{\omega}_d + \mathbf{R} \mathbf{R}_d^\top \dot{\boldsymbol{\omega}}_d
\end{aligned} \tag{72}$$

Dans le repère du drone, l'équation $\ddot{\mathbf{e}} + \alpha_1 \dot{\mathbf{e}} + \alpha_0 \mathbf{e} = \mathbf{0}$, devient $\ddot{\mathbf{e}}_r + \alpha_1 \dot{\mathbf{e}}_r + \alpha_0 \mathbf{e}_r = \mathbf{0}$, avec

$$\begin{aligned}
\mathbf{e}_r &= \mathbf{R}^\top \mathbf{e} = \mathbf{R}^\top \text{Log}(\mathbf{R}_d \mathbf{R}^\top) \\
\dot{\mathbf{e}}_r &= \mathbf{R}^\top \dot{\mathbf{e}} = -\mathbf{R}^\top \boldsymbol{\omega} + \mathbf{R}_d^\top \boldsymbol{\omega}_d \\
\ddot{\mathbf{e}}_r &= \mathbf{R}^\top \ddot{\mathbf{e}} = \underbrace{-\mathbf{R}^\top \dot{\boldsymbol{\omega}}}_{=\dot{\boldsymbol{\omega}}_r} + \underbrace{\mathbf{R}^\top \dot{\mathbf{R}} \mathbf{R}_d^\top \boldsymbol{\omega}_d}_{=\boldsymbol{\omega}_r \wedge} + \mathbf{R}_d^\top \dot{\boldsymbol{\omega}}_d + \mathbf{R}_d^\top \boldsymbol{\omega}_d
\end{aligned} \tag{73}$$

La dynamique de l'erreur devient :

$$\begin{aligned}
\mathbf{0} &= \ddot{\mathbf{e}}_r + \alpha_1 \dot{\mathbf{e}}_r + \alpha_0 \mathbf{e}_r \\
\Leftrightarrow \mathbf{0} &= -\dot{\boldsymbol{\omega}}_r + (\boldsymbol{\omega}_r \wedge) \mathbf{R}_d^\top \boldsymbol{\omega}_d + \dot{\mathbf{R}}_d^\top \boldsymbol{\omega}_d + \mathbf{R}_d^\top \dot{\boldsymbol{\omega}}_d + \alpha_1 \dot{\mathbf{e}}_r + \alpha_0 \mathbf{e}_r \\
\Leftrightarrow \dot{\boldsymbol{\omega}}_r &= (\boldsymbol{\omega}_r \wedge) \mathbf{R}_d^\top \boldsymbol{\omega}_d + \dot{\mathbf{R}}_d^\top \boldsymbol{\omega}_d + \mathbf{R}_d^\top \dot{\boldsymbol{\omega}}_d + \alpha_1 \dot{\mathbf{e}}_r + \alpha_0 \mathbf{e}_r
\end{aligned} \tag{74}$$

Ainsi, pour suivre une orientation désirée égale à $\mathbf{R}_d(t)$ nous pouvons prendre le régulateur donné ci-dessous :

Orienteur (in : $\mathbf{R}_d, \dot{\mathbf{R}}_d, \ddot{\mathbf{R}}_d, \mathbf{R}, \boldsymbol{\omega}_r$; out : $\dot{\boldsymbol{\omega}}_r^d$)	
1	$\boldsymbol{\omega}_d = \wedge^{-1}(\dot{\mathbf{R}}_d \mathbf{R}_d^\top)$
2	$\dot{\boldsymbol{\omega}}_d = \wedge^{-1}(\ddot{\mathbf{R}}_d \mathbf{R}_d^\top + \dot{\mathbf{R}}_d \dot{\mathbf{R}}_d^\top)$
3	$\mathbf{e}_r = \mathbf{R}^\top \text{Log}(\mathbf{R}_d \mathbf{R}^\top)$
4	$\dot{\mathbf{e}}_r = -\boldsymbol{\omega}_r + \mathbf{R}_d^\top \boldsymbol{\omega}_d$
5	$\dot{\boldsymbol{\omega}}_r^d = ((\boldsymbol{\omega}_r \wedge) \mathbf{R}_d^\top + \dot{\mathbf{R}}_d^\top) \boldsymbol{\omega}_d + \mathbf{R}_d^\top \dot{\boldsymbol{\omega}}_d + \alpha_1 \dot{\mathbf{e}}_r + \alpha_0 \mathbf{e}_r$

Il faut choisir les coefficients α_0, α_1 qui assurent la stabilité de l'erreur. Cela se fait à travers le choix du polynôme caractéristique. Par exemple, si nous choisissons : $P(s) = (s + 1)^2 = s^2 + 2s + 1$ alors nous obtenons $\alpha_0 = 1$ et $\alpha_1 = 2$.

5.4 Contrôleur

Comme illustré par la figure 17, nous sommes capables de générer la force résultante $\mathbf{f}_r$ et le couple $\boldsymbol{\tau}_r$ qui permettent de suivre la position $\mathbf{p}_d(t)$ et l'orientation $\mathbf{R}_d(t)$. Pour générer $\boldsymbol{\tau}_r$, nous pouvons utiliser des propulseurs, des gouvernes, ou même des disques d'inertie fixés à l'intérieur du drone.

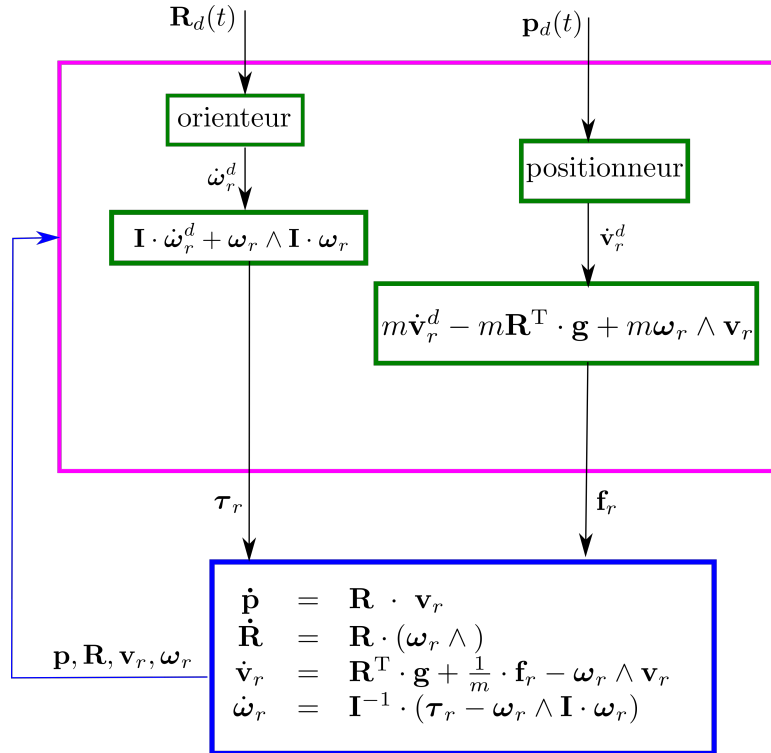


FIGURE 17 – Orienteur and positionneur

5.5 Drone bi-dimensionnel

Avant de passer aux drones tri-dimensionnels, illustrons ces notions sur un drone 2D qui évolue sur un plan (voir figure 18). L'équation (61) reste valide, mais se simplifie. En effet, nous obtenons

$$\begin{aligned}
 \mathbf{R} &= \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
 \mathbf{p} &= \begin{pmatrix} p_1 \\ p_2 \\ 0 \end{pmatrix}, \boldsymbol{\omega}_r = \begin{pmatrix} 0 \\ 0 \\ \omega \end{pmatrix} \\
 \boldsymbol{\omega}_r \wedge &= \begin{pmatrix} 0 & -\omega & 0 \\ \omega & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\
 \dot{\boldsymbol{\omega}}_r &= \begin{pmatrix} 0 \\ 0 \\ \dot{\omega} \end{pmatrix}, \mathbf{f}_r = \begin{pmatrix} f_{r1} \\ f_{r2} \\ 0 \end{pmatrix}
 \end{aligned} \tag{75}$$

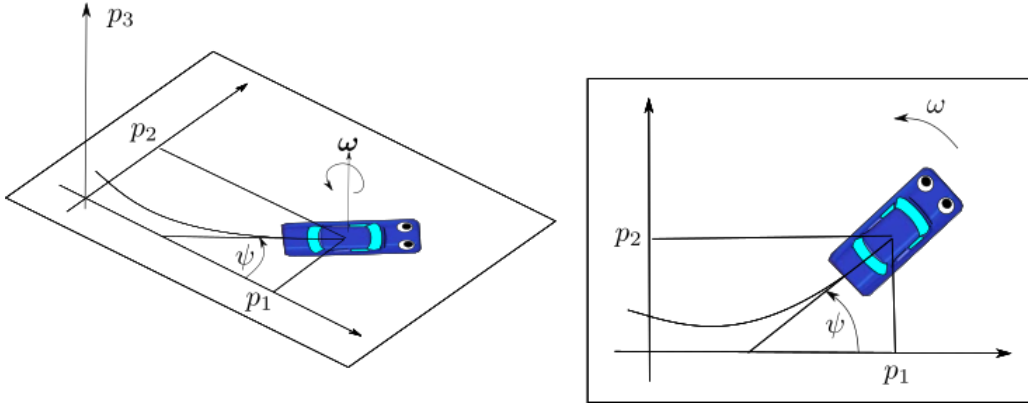


FIGURE 18 – Robot évoluant dans un plan

Si nous sélectionnons uniquement les composantes en x, y , comme sur la figure 18, droite, nous obtenons

$$\begin{cases} \dot{\mathbf{p}} &= \mathbf{R} \mathbf{v}_r \\ \dot{\mathbf{R}} &= \mathbf{R} \boldsymbol{\omega} \wedge \\ \dot{\mathbf{v}}_r &= \mathbf{R}^T \begin{pmatrix} 0 \\ g \end{pmatrix} + \frac{1}{m} \mathbf{f}_r - (\boldsymbol{\omega} \wedge) \mathbf{v}_r \\ \dot{\omega} &= \frac{\tau}{I_{33}} \end{cases} \tag{76}$$

et les vecteurs $\mathbf{v}_r, \mathbf{f}_r$ sont tous les deux bi-dimensionnels. Dans cette équation, ω est un scalaire. Nous avons

$$\boldsymbol{\omega} \wedge = \begin{pmatrix} 0 & -\omega \\ \omega & 0 \end{pmatrix}. \tag{77}$$

Il n'est alors plus nécessaire de spécifier si cette vitesse de rotation est dans le repère du drone ou celui du monde : elles sont les mêmes. De plus,

$$\mathbf{R} = \begin{pmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{pmatrix} \tag{78}$$

La seule différence avec les équations (46) et (76) réside dans le fait que l'équation d'Euler se réduit à $\dot{\omega} = \frac{\tau}{I_{33}}$. Cela est dû au fait que ω, τ sont désormais scalaires et non plus des vecteurs à trois dimensions.

Les équations pour l'orienteur et le positionneur restent valides.

6 Quadrirotor

Considérons le quadrirotor de la figure 20 que nous allons chercher à modéliser et à réguler [16]. Le drone possède quatre propulseurs qui peuvent être commandés indépendamment. Ces actionneurs vont permettre de réguler l'attitude et la position. Pour ce type de drone, on distingue classiquement quatre mouvements possibles : les gaz, le lacet, le roulis et le tangage. Pour garder le contrôle du lacet, il faut deux hélices qui tournent dans le sens horaire (hélice à pas normal) et les deux autres dans le sens antihoraire (hélices à pas inversé). Le drone matrice 600 de la figure 19 possède 6 propulseurs. Il peut pourtant être considéré comme un quadrirotor car tous ses propulseurs ont une direction verticale.



FIGURE 19 – Matrice 600 de l'équipe Robex de l'ENSTA-Bretagne

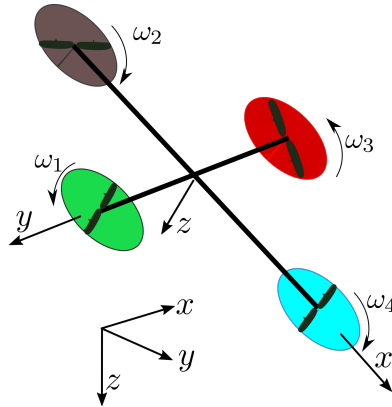


FIGURE 20 – Quadrirotor

Comme illustré par la figure, les propulseurs avant/arrière (bleu et noir), ne tournent pas dans le même sens que les propulseurs gauche/droite (rouge et vert). La valeur de la force exercée par le i ème propulseur est proportionnelle au carré de la vitesse du moteur, c'est-à-dire, égale à $\beta \omega_i |\omega_i|$, où β est le coefficient de portance. Notons δ le coefficient

de traînée et ℓ la distance entre les rotors et le centre du robot. Les forces et les couples générés sont donnés par la relation

$$\begin{pmatrix} \tau_0 \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{pmatrix} = \mathbf{B} \begin{pmatrix} \omega_1 |\omega_1| \\ \omega_2 |\omega_2| \\ \omega_3 |\omega_3| \\ \omega_4 |\omega_4| \end{pmatrix} \quad (79)$$

avec

$$\mathbf{B} = \begin{pmatrix} \beta & \beta & \beta & \beta \\ -\beta\ell & 0 & \beta\ell & 0 \\ 0 & -\beta\ell & 0 & \beta\ell \\ -\delta & \delta & -\delta & \delta \end{pmatrix} \quad (80)$$

Dans cette relation, τ_0 est la poussée résultante et τ_1, τ_2, τ_3 sont les composantes du couple résultant suivant les directions x, y, z .

Puisque la poussée résultante et le couple résultant sont donnés par

$$\mathbf{f}_r = \begin{pmatrix} 0 \\ 0 \\ -\tau_0 \end{pmatrix} \text{ et } \boldsymbol{\tau}_r = \begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{pmatrix}, \quad (81)$$

d'après (46), nous obtenons que la dynamique du drone peut être décrite par le schéma bloc de la figure 21 où $\mathbf{p}$ est la position du drone et $\mathbf{R}$ est son orientation. Notons que le schéma peut être vu comme une chaîne de relations causales reliant la vitesse des rotors ω_i à la position $\mathbf{p}$. Cette chaîne causale sera utilisée lors de la synthèse du contrôleur en utilisant la technique du backstepping [17].

Les variables d'état sont : la position $\mathbf{p}$, l'orientation $\mathbf{R}$, la vitesse $\mathbf{v}_r$ et le vecteur de rotation $\boldsymbol{\omega}_r$, ces deux derniers exprimés dans le repère du drone.

Pour contrôler ce quadrirotor, nous proposons à nouveau d'utiliser une technique par backstepping. Un régulateur correspondra à une séquence de boucles de régulation, chacune d'elle inversant un bloc de la chaîne causale.

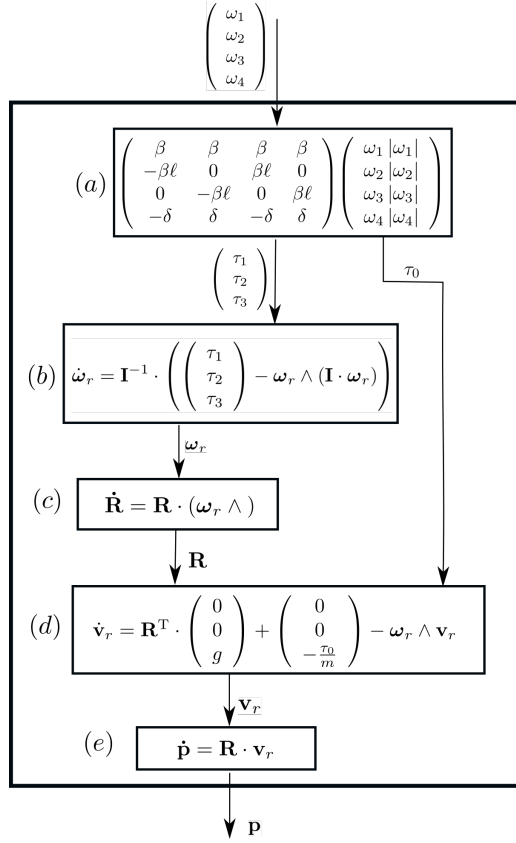


FIGURE 21 – Schéma bloc représentant la dynamique du quadrirotor. La chaîne causale relie les 5 blocs (a), (b), (c), (d), (e)

Bloc (a). Pour inverser le bloc (a), nous proposons un contrôleur feedforward avec pour nouvelle entrée $(\tau_0^d, \tau_1^d, \tau_2^d, \tau_3^d)$. L'exposant d signifie ici *désiré* puisque c'est celle que l'on souhaite pour τ_i . Ainsi, nous prendrons

$$\begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{pmatrix} = {}^a \sqrt{\mathbf{B}^{-1} \begin{pmatrix} \tau_0^d \\ \tau_1^d \\ \tau_2^d \\ \tau_3^d \end{pmatrix}} \quad (82)$$

où ${}^a \sqrt{\mathbf{v}}$ est la racine carrée algébrique prise composante par composante (par exemple ${}^a \sqrt{(-4, 16, -9)} = (-2, 4, -3)$). Avec cette boucle feedforward, les τ_i seront presque égaux aux τ_i^d . Nous disons '*presque*' car il existe forcément quelques incertitudes sur les coefficients δ, β, ℓ et quelques éléments non modélisés comme l'inertie des rotors et des pales.

Bloc (b). Pour inverser le bloc (b) nous adoptons une méthode de linéarisation par bouclage. C'est-à-dire que nous construisons un contrôleur dont l'entrée est le vecteur de rotation désiré ω_r^d et sa sortie est $\tau_{1:3}^d = (\tau_1^d, \tau_2^d, \tau_3^d)$. On demandera à ce que ω_r converge rapidement vers ω_r^d . Pour cela, prenons

$$\tau_{1:3}^d = \mathbf{I} k_1 (\omega_r^d - \omega_r) + \omega_r \wedge (\mathbf{I} \omega_r) \quad (83)$$

où k_1 est un grand gain (par exemple $k_1 = 100$). Donc

$$\begin{aligned}
 \dot{\omega}_r &= \mathbf{I}^{-1} (\tau_{1:3}^d - \omega_r \wedge (\mathbf{I} \omega_r)) \\
 &= \mathbf{I}^{-1} \tau_{1:3}^d - \mathbf{I}^{-1} (\omega_r \wedge (\mathbf{I} \omega_r)) \\
 &= k_1 \cdot (\omega_r^d - \omega_r)
 \end{aligned} \tag{84}$$

Si le gain k_1 est suffisamment grand et si ω_r^d varie lentement, alors, l'erreur $\omega_r^d - \omega_r$ converge rapidement vers zéro (en un temps approximativement égal à $\frac{1}{k_1}$ sec).

Bloc (c). Nous construisons un contrôleur ayant pour entrée $\mathbf{R}^d$ l'orientation désirée et ω_r^d pour sortie. Afin que $\mathbf{R}$ converge vers $\mathbf{R}^d$, nous prenons

$$\omega_r^d = k_2 \mathbf{R}^\top \text{Log}(\mathbf{R}^d \mathbf{R}^\top) \tag{85}$$

Le coefficient k_2 doit être choisi suffisamment grand (par exemple $k_2 = 10$), mais pas trop sinon ω_r^d aura des variations brutales qui vont perturber le régulateur en aval. Cette loi de commande suppose que les propulseurs seront suffisamment puissants pour être capables de donner l'orientation désirée en un temps égal à $\frac{1}{10}$ sec. La figure 22 illustre la séquence d'inversion proposée.

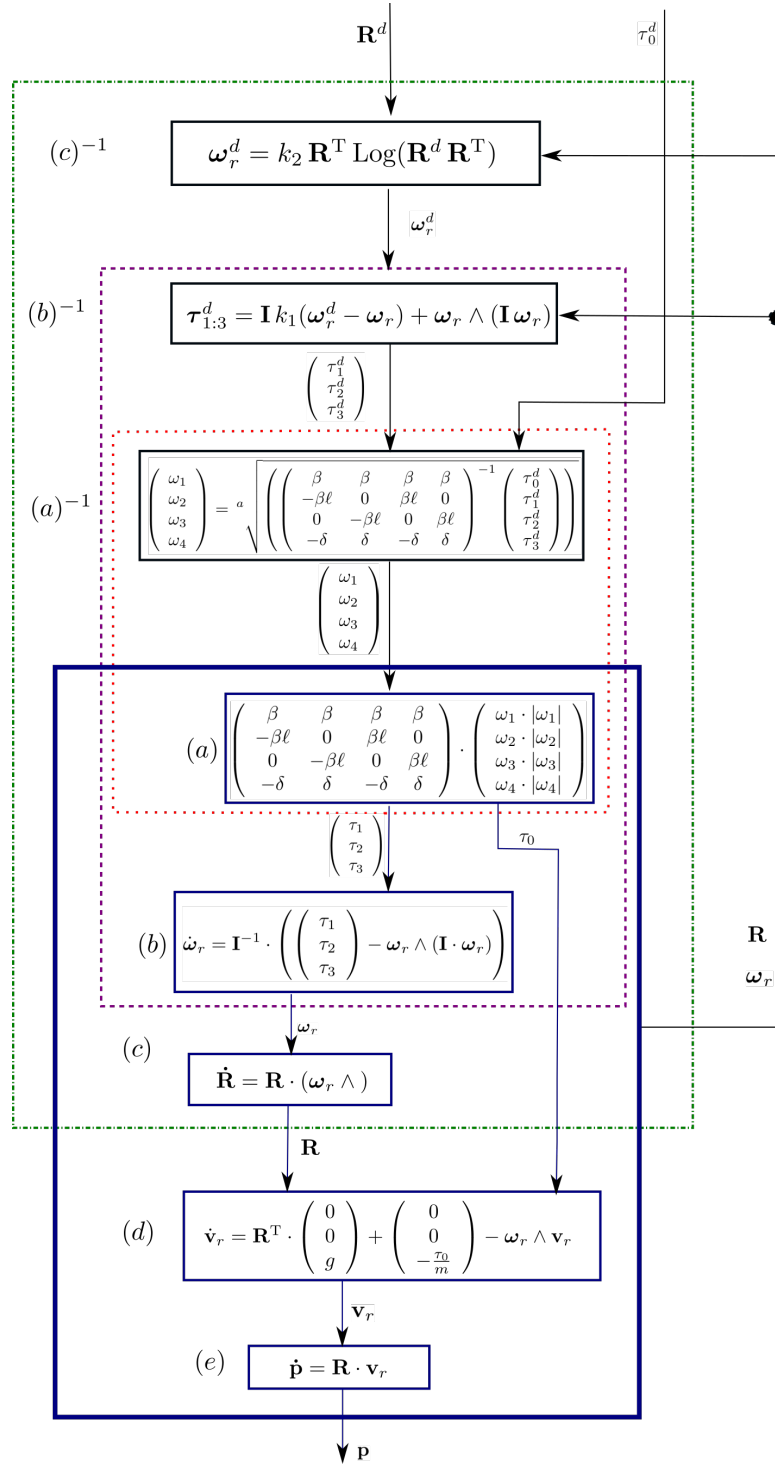


FIGURE 22 – Séquence d'inversions de type backstepping permettant une régulation étagée

Nous sommes désormais capables de donner à notre drone une vitesse et une orientation arbitraire. Nous allons chercher une dernière boucle de contrôle $(\mathbf{R}^d, \tau_0^d)$, qui permet à notre drone de suivre un champ de vecteur. Plus précisément, nous voudrions qu'il suive un chemin dont la dynamique est décrite par l'équation de Van der Pol :

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -(\alpha x_1^2 - 1)x_2 - x_1 \end{cases} \quad (86)$$

avec $\alpha = 0.001$. De plus, nous demandons au drone de rester à une altitude de 15m avec une vitesse de $v^d = 10\text{ms}^{-1}$. Enfin, nous voulons que l'avant du drone se trouve dans le sens de la trajectoire. Nous proposons le régulateur d'orientation suivant :

$$\mathbf{R}^d = \mathbf{R}_{\text{euler}}(\varphi^d, \theta^d, \psi^d) = e^{\psi^d \mathbf{k}^\wedge} e^{\theta^d \mathbf{j}^\wedge} e^{\varphi^d \mathbf{i}^\wedge} \quad (87)$$

avec

$$\begin{aligned} \varphi^d &= 0.5 \tanh(10 (\text{angle}(\mathbf{f}_{vdp}(x, y), \mathbf{R} \mathbf{v}_r))) \\ \theta^d &= -0.3 \tanh(v^d - v_{r1}) \\ \psi^d &= \text{angle}(\mathbf{i}, \mathbf{R} \mathbf{v}_r) \\ \tau_0^d &= 300 \tanh(z - z^d) + 60 v_{r3} \end{aligned} \quad (88)$$

L'équation pour τ_0^d correspond à un régulateur de type proportionnel et dérivé ($300 \tanh(z - z^d)$) avec un grand gain proportionnel permettant de lutter efficacement contre la gravité. Le terme dérivé $60 v_{r3}$ permet de limiter les oscillations verticales. Des coefficients sont réglés sur la simulation pour avoir un contrôle sur l'altitude convenable. L'équation ψ^d assure que le cap du robot se trouve suivant la direction d'avance donnée par le vecteur $\mathbf{R} \mathbf{v}_r$. Cette contrainte a du sens si le robot possède un capteur de perception comme une caméra frontale. La fonction *angle* calcule l'angle entre deux vecteurs en regardant uniquement les deux composantes horizontales. L'équation θ^d se fonde sur le fait que l'assiette régie la vitesse désirée. L'équation de gîte φ^d est motivée par la poursuite du champ de Van der Pol $\mathbf{f}_{vdp}(x, y)$: nous imposons un roulis qui correspond à l'erreur en cap dans le but de le corriger.

Une simulation est donnée sur la figure 23. Le cycle de Van der Pol est représenté en vert. Sur cette simulation, nous avons pris $m = 10$, $g = 9.81$, $\beta = 2$, $d = 1$, et $\ell = 1$, tous exprimés en unités internationales. La matrice d'inertie a été choisie égale à

$$\mathbf{I} = \begin{pmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 20 \end{pmatrix}. \quad (89)$$

Le programme PYTHON effectuant cette simulation est disponible sur <https://www.ensta-bretagne.fr/jaulin/teching.html>

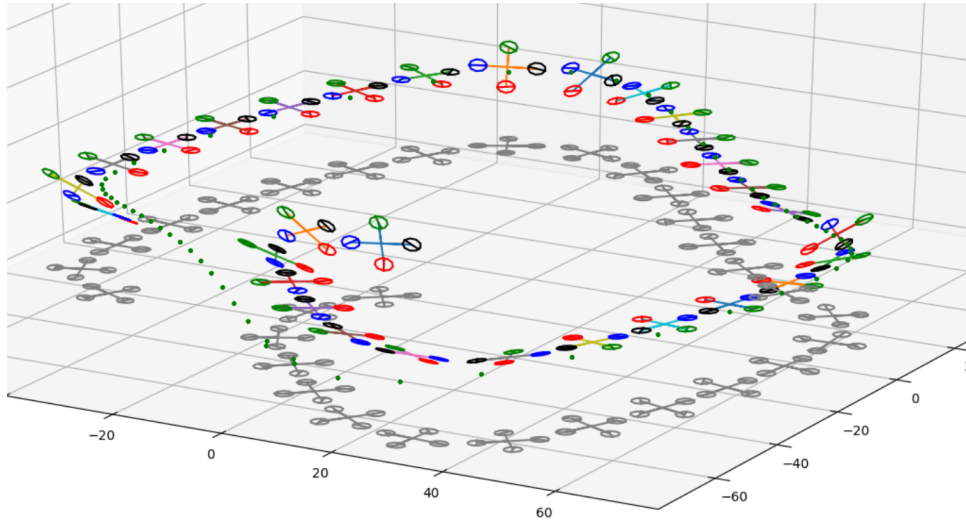


FIGURE 23 – Simulation du quadrirotor qui suit le cycle de Van der Pol

7 Torpille sous-marine

Considérons un robot sous-marin de type torpille tel que celui représenté sur la figure 24. Pour se déplacer, le drone est équipé de trois gouvernes et un propulseur. L'entrée u_0 , est la vitesse de rotation du propulseur et u_1, u_2, u_3 sont les angles des gouvernes. Nous supposons que le vecteur vitesse $\mathbf{v}_r$ est toujours suivant la direction du robot, ou de façon équivalente, le drone n'a pas de vitesse latérale.

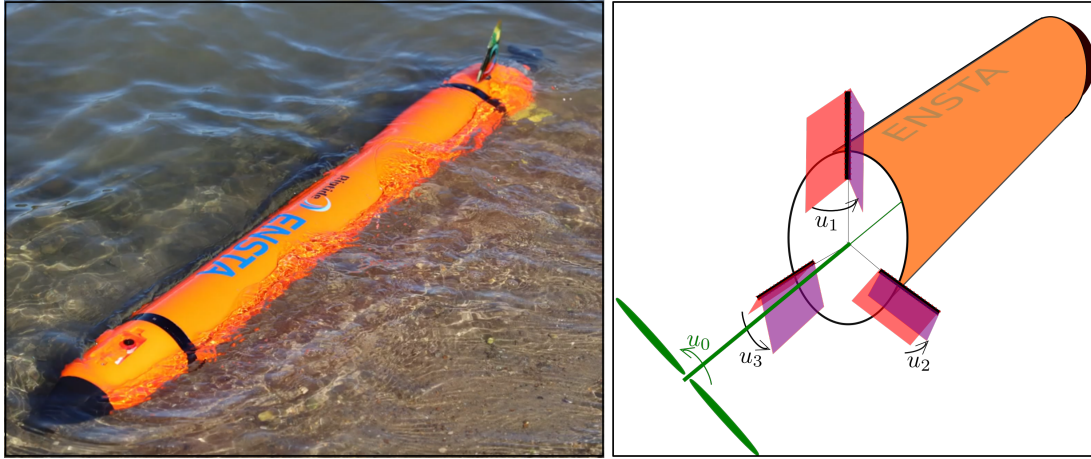


FIGURE 24 – Torpille sous-marine comprenant trois gouvernes et un propulseur

Puisque la vitesse du robot est uniquement tangentielle, les équations cinématiques du drone sont de la forme

$$\begin{cases} \dot{\mathbf{p}} &= \mathbf{R} (v \mathbf{0} \mathbf{0})^\top \\ \dot{\mathbf{R}} &= \mathbf{R} (\boldsymbol{\omega}_r \wedge) \\ \dot{v} &= a_{rx} \end{cases} \quad (90)$$

où v est la vitesse du drone, $\mathbf{R}$ est la matrice d'orientation et $\boldsymbol{\omega}_r$ est le vecteur de rotation exprimé dans le repère du drone. Dans ce modèle, les entrées sont $\boldsymbol{\omega}_r$ et l'accélération tangentielle a_{rx} .

Modèle dynamique. Nous allons maintenant proposer un modèle dynamique pour le drone où les entrées seront u_0, u_1, u_2, u_3 .

Si la torpille va tout droit et si f_x est la résultante des forces longitudinales, alors, en utilisant le principe fondamental de la dynamique, nous avons

$$f_x = (m + m_a) a_{rx} \quad (91)$$

où m_a est la masse ajoutée. Nous avons

$$m_a = k_a \rho_0 V \quad (92)$$

où V est le volume, ρ_0 est la densité de l'eau et k_a est un coefficient qui dépend de la forme du robot. Si nous supposons que le drone a une forme cylindrique, nous pouvons prendre $k_a = 1$. De plus, si nous supposons que la torpille possède une flottabilité neutre, nous avons $\rho_0 V \simeq m$ et donc $m \simeq m_a$. De plus, nous avons

$$f_x = k_p u_0^2 - \frac{1}{2} C_x S_x \rho_0 v^2 \quad (93)$$

où u_0 est la vitesse de rotation du propulseur, C_x est le coefficient de traînée longitudinale et S_x la surface frontale. Finalement, nous avons

$$a_{rx} = \frac{f_x}{m + m_a} = p_1 u_0^2 - p_2 v^2. \quad (94)$$

Les coefficients $V, k_a, \rho_0, k_p, C_x, S_x$ ne sont connus qu'approximativement. En revanche, les paramètres p_1, p_2 nous suffisent et peuvent être facilement identifiés à travers quelques expérimentations.

L'orientation du drone se modifie grâce à chaque gouverne qui génère une portance et une traînée. La traînée peut être négligée. De plus, nous pouvons supposer une réponse directe entre les angles des gouvernes et le vecteur de rotation, ce qui nous donne une équation du type :

$$\boldsymbol{\omega}_r = \begin{pmatrix} \omega_{rx} \\ \omega_{ry} \\ \omega_{rz} \end{pmatrix} = v \mathbf{B}(p_3, p_4) \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \quad (95)$$

avec

$$\begin{aligned} \mathbf{B}(p_3, p_4) &= \begin{pmatrix} p_3 & 0 & 0 \\ 0 & p_4 & 0 \\ 0 & 0 & p_4 \end{pmatrix} \cdot \begin{pmatrix} -1 & -1 & -1 \\ 0 & \sin \frac{2\pi}{3} & -\sin \frac{2\pi}{3} \\ 1 & \cos \frac{2\pi}{3} & \cos \frac{2\pi}{3} \end{pmatrix} \\ &= \begin{pmatrix} -p_3 & -p_3 & -p_3 \\ 0 & p_4 \sin \frac{2\pi}{3} & -p_4 \sin \frac{2\pi}{3} \\ p_4 & p_4 \cos \frac{2\pi}{3} & p_4 \cos \frac{2\pi}{3} \end{pmatrix} \end{aligned} \quad (96)$$

Cette relation nous dit par exemple que chaque gouverne contribue de façon équivalente au roulis ω_{rx} d'où la ligne avec les $-p_3$. Elle nous dit également que la gouverne 1 ne contribue pas au tangage d'où le zéro sur la première colonne de $\mathbf{B}(p_3, p_4)$.

Ainsi, nous pouvons considérer comme modèle dynamique pour notre torpille :

$$\begin{cases} \dot{\mathbf{p}} &= \mathbf{R} (v 0 0)^T \\ \dot{\mathbf{R}} &= \mathbf{R} \left(v \mathbf{B}(p_3, p_4) \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \right) \wedge \\ \dot{v} &= p_1 u_0^2 - p_2 v^2 \end{cases} \quad (97)$$

Les entrées du système sont u_0, u_1, u_2, u_3 .

Contrôleur. Nous voulons un régulateur pour ce système. Plus précisément, nous voulons que le système bouclé ait le vecteur de rotation désirés $\bar{\omega}_r$ et l'accélération désirée $\bar{a}_{rx}$. Pour cela, prenons

$$u_0 = \sqrt{\frac{\bar{a}_{rx} + p_2 v^2}{p_1}} \quad (98)$$

et

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \frac{1}{v} \mathbf{B}^{-1}(p_3, p_4) \bar{\omega}_r. \quad (99)$$

Ainsi, nous aurons

$$\begin{cases} \dot{\mathbf{p}} &= \mathbf{R} (v \mathbf{0} \mathbf{0})^\top \\ \dot{\mathbf{R}} &= \mathbf{R} \bar{\omega}_r \wedge \\ \dot{v} &= \bar{a}_{rx} \end{cases} \quad (100)$$

Afin d'avoir une régulation en vitesse de 1 m s^{-1} , nous pouvons prendre une accélération $\bar{a}_{rx} = 0.1(1 - v)$. Prenons $\bar{\omega}_r = (0, 0, 0.03)$ pour avoir une légère rotation suivant l'axe z . En prenant un état initial donné par $\mathbf{p}(0) = (0, 0, -25)$, $\mathbf{R}(0) = \mathbf{I}$ et $v(0) = 0.01$ et avec les paramètres dynamiques suivants : $p_1 = 1, p_2 = 2, p_3 = 1, p_4 = 4$, nous obtenons la trajectoire représentée sur la figure 25.

Le programme PYTHON effectuant cette simulation est disponible sur <https://www.ensta-bretagne.fr/jaulin/teching.html>

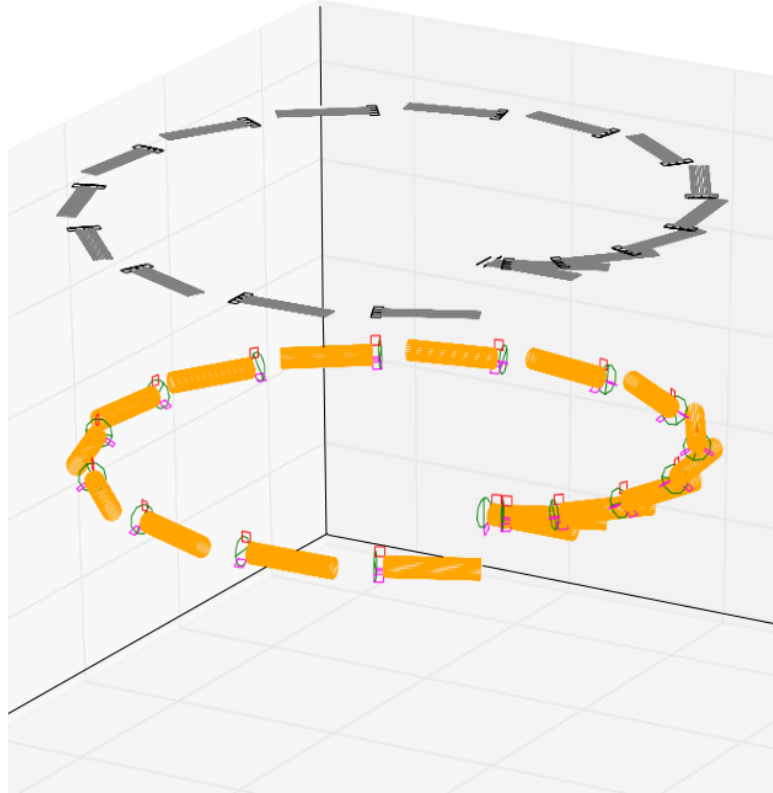


FIGURE 25 – La torpille (orange) est régulée pour tourner à gauche avec une vitesse de 1 m s^{-1} . La projection à la surface de l'eau est représentée en gris

8 Hélicoptère

La sustentation et la propulsion d'un hélicoptère sont assurées par une voilure tournante. L'hélicoptère de la figure 26 utilise un seul rotor de sustentation et un rotor anticouple situé à l'extrémité arrière du fuselage. Les pales du rotor de sustentation tournent toujours à vitesse angulaire constante, c'est la variation de leur angle d'incidence qui provoque un couple et permet de contrôler l'assiette. Pour cabrer par exemple, on modifie l'incidence des pales de sorte que la portance de celles-ci soit maximale au passage à l'avant de l'appareil, et minimale à l'arrière. Cette variation d'incidence est aussi utilisée pour les variations en gîte. Le mouvement du rotor de sustentation génère une traînée qui tend à faire tourner l'engin autour de son axe. Le rotor anticouple cherche à lutter contre cet effet indésirable.

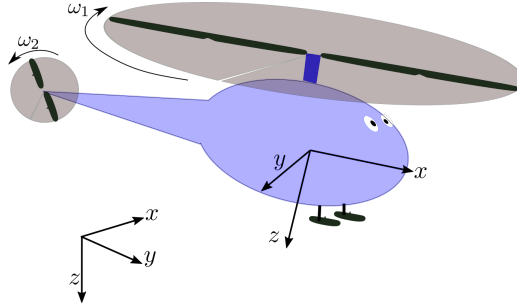


FIGURE 26 – Hélicoptère avec un rotor de sustentation et un rotor anticouple

On peut modéliser [18] l'hélicoptère par les équations d'état suivantes :

$$\begin{cases} \dot{\mathbf{p}} &= \mathbf{R} \mathbf{v}_r \\ \dot{\mathbf{R}} &= \mathbf{R} (\boldsymbol{\omega}_r \wedge) \\ \dot{\mathbf{v}}_r &= \mathbf{R}^T \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ \frac{\tau_0}{m} \end{pmatrix} - \boldsymbol{\omega}_r \wedge \mathbf{v}_r \\ \dot{\boldsymbol{\omega}}_r &= \mathbf{I}^{-1} \left(\begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{pmatrix} - \boldsymbol{\omega}_r \wedge (\mathbf{I} \boldsymbol{\omega}_r) \right) \end{cases} \quad (101)$$

où

$$\begin{pmatrix} \tau_0 \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{pmatrix} = \begin{pmatrix} \beta_1 \omega_1^2 & 0 & 0 & 0 \\ 0 & \beta_2 \omega_1^2 & 0 & 0 \\ 0 & 0 & \beta_3 \omega_1^2 & 0 \\ -\delta_1 \omega_1^2 & 0 & 0 & -\beta_4 \ell \omega_2^2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} \quad (102)$$

Dans ces équations, $\mathbf{R}$ correspond à l'orientation, $\mathbf{p}$ est la position, $\mathbf{v}_r$ la vitesse, $\boldsymbol{\omega}_r$ le vecteur de rotation, τ_0 la poussée verticale générée par le rotor de sustentation, τ_1 le couple de roulis, τ_2 le couple de tangage, τ_3 le couple de lacet, ce dernier étant généré par le rotor anticouple. Les entrées u_i jouent sur l'incidence des pales. Nous supposons ici que $\omega_1 = \omega_2 = 100$, $\beta_1 = 0.02$, $\beta_2 = \beta_3 = \frac{\beta_1}{10}$, $\beta_4 = 0.002$, $\delta_1 = \frac{\beta_1}{5}$, tous ces paramètres étant exprimés en unités internationales.

Cherchons à donner à l'hélicoptère l'orientation désirée $\mathbf{R}^d$ avec la poussée τ_0^d en utilisant à nouveau une approche

par backstepping. Si nous appliquons la boucle C_1 comme donnée sur la figure 27, nous obtenons le système $\mathcal{S}_1$

$$\begin{cases} \dot{\mathbf{p}} &= \mathbf{R} \mathbf{v}_r \\ \dot{\mathbf{R}} &= \mathbf{R} (\boldsymbol{\omega}_r \wedge) \\ \dot{\mathbf{v}}_r &= \mathbf{R}^T \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ \frac{\tau_0^d}{m} \end{pmatrix} - \boldsymbol{\omega}_r \wedge \mathbf{v}_r \\ \dot{\boldsymbol{\omega}}_r &= \mathbf{I}^{-1} \left(\begin{pmatrix} \tau_1^d \\ \tau_2^d \\ \tau_3^d \end{pmatrix} - \boldsymbol{\omega}_r \wedge (\mathbf{I} \boldsymbol{\omega}_r) \right) \end{cases} \quad (103)$$

où les nouvelles entrées sont $\tau_0^d, \tau_1^d, \tau_2^d, \tau_3^d$.

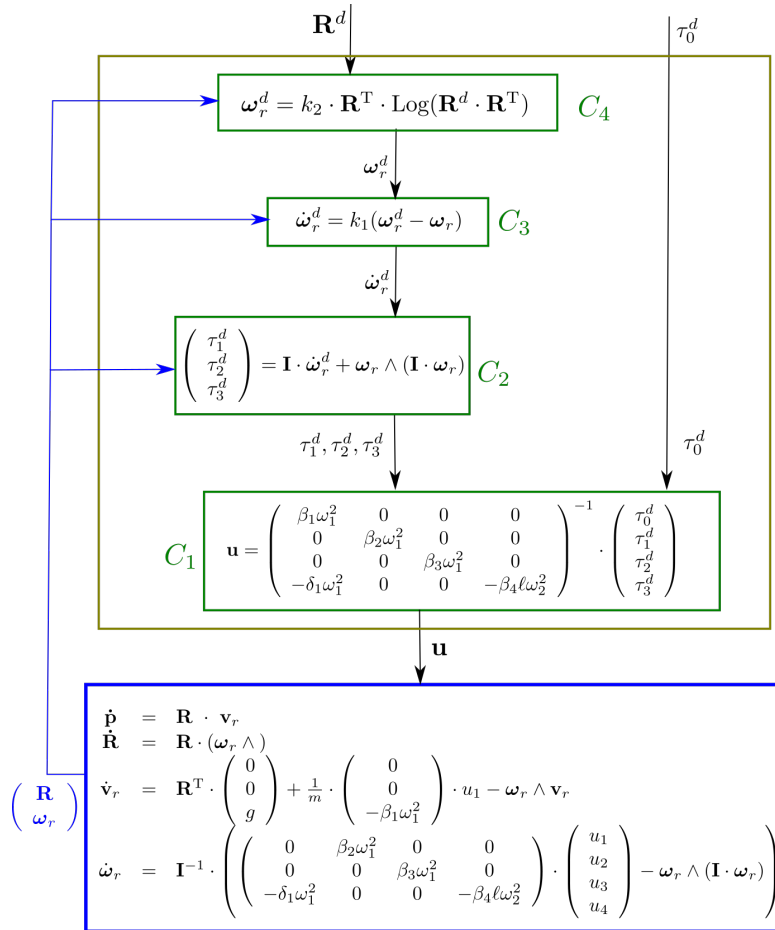


FIGURE 27 – Contrôle de l'orientation de l'hélicoptère

Si nous rajoutons la boucle de contrôle C_2 , le système se réduit à :

$$\begin{cases} \dot{\mathbf{p}} &= \mathbf{R} \mathbf{v}_r \\ \dot{\mathbf{R}} &= \mathbf{R} (\boldsymbol{\omega}_r \wedge) \\ \dot{\mathbf{v}}_r &= \mathbf{R}^\top \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -\frac{\tau_0^d}{m} \end{pmatrix} - \boldsymbol{\omega}_r \wedge \mathbf{v}_r \\ \dot{\boldsymbol{\omega}}_r &= \dot{\boldsymbol{\omega}}_r^d \end{cases} \quad (104)$$

où les nouvelles entrées sont $\dot{\boldsymbol{\omega}}_r^d$ and τ_0^d . Rajoutons la boucle C_3 . Après un régime transitoire très court, nous obtenons $\boldsymbol{\omega}_r^d = \boldsymbol{\omega}_r$ et le système se ramène à :

$$\begin{cases} \dot{\mathbf{p}} &= \mathbf{R} \mathbf{v}_r \\ \dot{\mathbf{R}} &= \mathbf{R} (\boldsymbol{\omega}_r^d \wedge) \\ \dot{\mathbf{v}}_r &= \mathbf{R}^\top \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -\frac{\tau_0^d}{m} \end{pmatrix} - \boldsymbol{\omega}_r^d \wedge \mathbf{v}_r \end{cases} \quad (105)$$

Prenons

$$\boldsymbol{\omega}_r^d = k_2 \mathbf{R}^\top \text{Log}(\mathbf{R}^d \mathbf{R}^\top). \quad (106)$$

Après un transitoire très court, nous obtenons $\mathbf{R}^d \simeq \mathbf{R}$. L'hélicoptère ainsi contrôlé possède deux entrées : l'orientation désirée $\mathbf{R}^d$ et la poussée désirée τ_0 . Il devient donc facile à piloter.

Illustrons cela à travers une manoeuvre acrobatique : le looping.

Etape 1. Nous stabilisons l'hélicoptère en orientation horizontalement et nous allons à une altitude désirée. Pour cela, nous choisissons

$$\mathbf{R}^d = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ et } \tau_0^d = \tau_0^{\max} \quad (107)$$

Une fois que l'hélicoptère a atteint l'altitude demandée, nous initialisons un chronomètre en prenant $t_0 = t$, et nous allons à l'étape 2.

Etape 2. Pour faire un looping en 2π sec, nous choisissons

$$\mathbf{R}^d = e^{(t-t_0)\mathbf{j}\wedge} \quad (108)$$

où $\mathbf{j} = (0 \ 1 \ 0)^\top$ en prenant par exemple $\tau_0^d = 0$. Dès que $t > t_0 + 2\pi$, nous allons à l'étape 3.

Etape 3. Pour stabiliser l'hélicoptère à l'horizontale à l'altitude p_3^d , nous pouvons prendre

$$\mathbf{R}^d = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ et } \tau_0^d = k_3 \tanh(p_3 - p_3^d) + k_4 v_{r3}. \quad (109)$$

Pour obtenir la simulation de la figure 28, nous avons pris pour la régulation les coefficients suivants $k_1 = 1, k_2 = 2, k_3 = 1000$ et $k_4 = 100$.

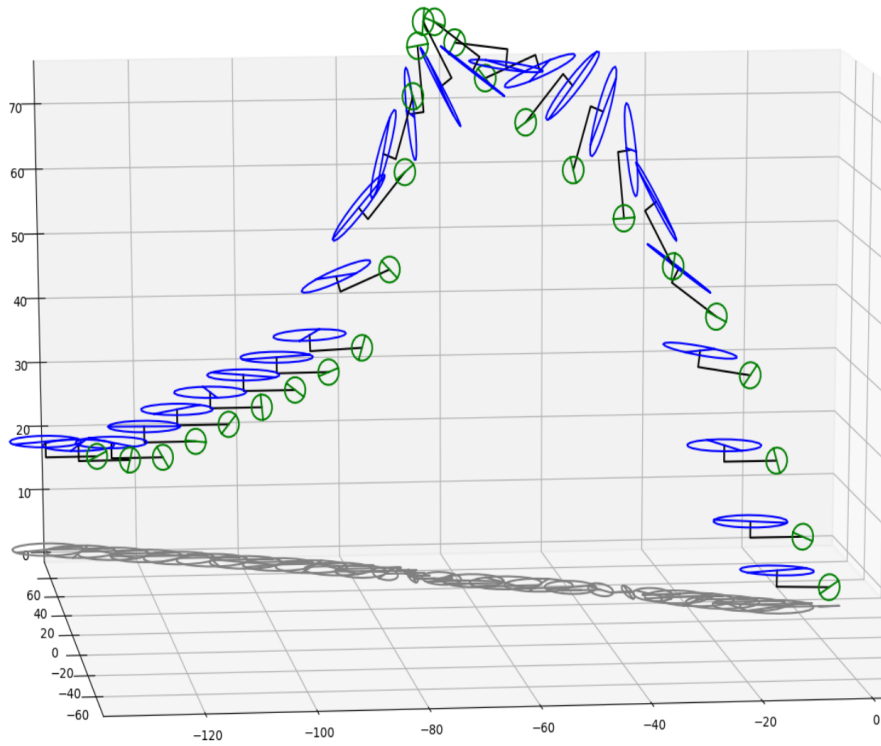


FIGURE 28 – Hélicoptère effectuant un looping

9 Hexarotor

Un hexarotor est un drone volant composé d'un corps solide et de 6 propulseurs à hélice. Chaque rotor a une position fixe $\mathbf{q}(i)$ sur le drone et une direction $\mathbf{d}(i)$. Les propulseurs produisent une force f_i qui a la direction $\mathbf{d}(i)$. Dans le repère du drone, les coordonnées $\mathbf{q}(i), \mathbf{d}(i)$ sont données par la table suivante :

i	$\mathbf{q}(i)$	$\mathbf{d}(i)$
1	$\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$
2	$\begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$
3	$\begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$
4	$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$
5	$\begin{pmatrix} \frac{1}{2} \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$
6	$\begin{pmatrix} 0 \\ \frac{1}{2} \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$

La géométrie du drone est illustrée par la figure 29. Chaque force f_i contribue à la résultante des forces par le vecteur $\mathbf{d}(i) f_i$ et contribue au couple total par le moment $\mathbf{q}(i) \wedge \mathbf{d}(i) f_i$. Nous négligerons le couple engendré par la traînée des pales.

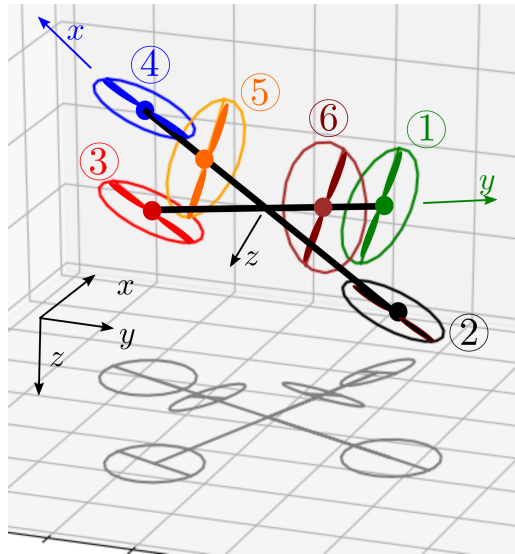


FIGURE 29 – Hexarotor avec ses six propulseurs

Cherchons à modéliser l'hexarotor. Le vecteur d'état sera composé de la position $\mathbf{p}$, de l'orientation $\mathbf{R}$, de la vitesse $\mathbf{v}_r$ et du vecteur de rotation $\boldsymbol{\omega}_r$. Les entrées du système sont $f_1, \dots, f_6$.

La force résultante exprimée dans le repère du drone est

$$\begin{aligned}\mathbf{f}_r &= \begin{pmatrix} \mathbf{d}(1) & \mathbf{d}(2) & \cdots & \mathbf{d}(6) \end{pmatrix} \begin{pmatrix} f_1 \\ \vdots \\ f_6 \end{pmatrix} \\ &= \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}}_{\mathbf{C}_f} \begin{pmatrix} f_1 \\ \vdots \\ f_6 \end{pmatrix}\end{aligned}\quad (110)$$

et le couple est donné par

$$\begin{aligned}\boldsymbol{\tau}_r &= \begin{pmatrix} \mathbf{q}(1) \wedge \mathbf{d}(1) & \cdots & \mathbf{q}(6) \wedge \mathbf{d}(6) \end{pmatrix} \begin{pmatrix} f_1 \\ \vdots \\ f_6 \end{pmatrix} \\ &= \underbrace{\begin{pmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_{\mathbf{C}_\tau} \begin{pmatrix} f_1 \\ \vdots \\ f_6 \end{pmatrix}\end{aligned}\quad (111)$$

Ainsi,

$$\begin{pmatrix} \mathbf{f}_r \\ \boldsymbol{\tau}_r \end{pmatrix} = \underbrace{\begin{pmatrix} \mathbf{C}_f \\ \mathbf{C}_\tau \end{pmatrix}}_{\mathbf{C}} \begin{pmatrix} f_1 \\ \vdots \\ f_6 \end{pmatrix}\quad (112)$$

La matrice $\mathbf{C}$ est appelée *concentrateur* ou *matrice de répartition*. D'après (46), nous obtenons que les équations d'état de l'hexarotor sont données par

$$\begin{cases} \dot{\mathbf{p}} &= \mathbf{R} \mathbf{v}_r \\ \dot{\mathbf{R}} &= \mathbf{R} (\boldsymbol{\omega}_r \wedge) \\ \dot{\mathbf{v}}_r &= \mathbf{R}^\top \mathbf{g} + \frac{1}{m} \mathbf{C}_f \begin{pmatrix} f_1 \\ \vdots \\ f_6 \end{pmatrix} - \boldsymbol{\omega}_r \wedge \mathbf{v}_r \\ \dot{\boldsymbol{\omega}}_r &= \mathbf{I}^{-1} \left(\mathbf{C}_\tau \begin{pmatrix} f_1 \\ \vdots \\ f_6 \end{pmatrix} - \boldsymbol{\omega}_r \wedge (\mathbf{I} \boldsymbol{\omega}_r) \right) \end{cases}\quad (113)$$

Cherchons maintenant à commander le drone afin qu'il puisse atterrir sur une plateforme mobile dont la position $\mathbf{p}_d(t)$ et son orientation $\mathbf{R}_d(t)$ sont connues, ainsi que leurs dérivées. Si nous voulons une force $\mathbf{f}_r$ et un couple $\boldsymbol{\tau}_r$, il faut appliquer les forces :

$$\begin{pmatrix} f_1 \\ \vdots \\ f_6 \end{pmatrix} = \mathbf{C}^{-1} \begin{pmatrix} \mathbf{f}_r \\ \boldsymbol{\tau}_r \end{pmatrix}.\quad (114)$$

En utilisant le contrôleur inertiel, nous obtenons le système bouclé de la figure 30.

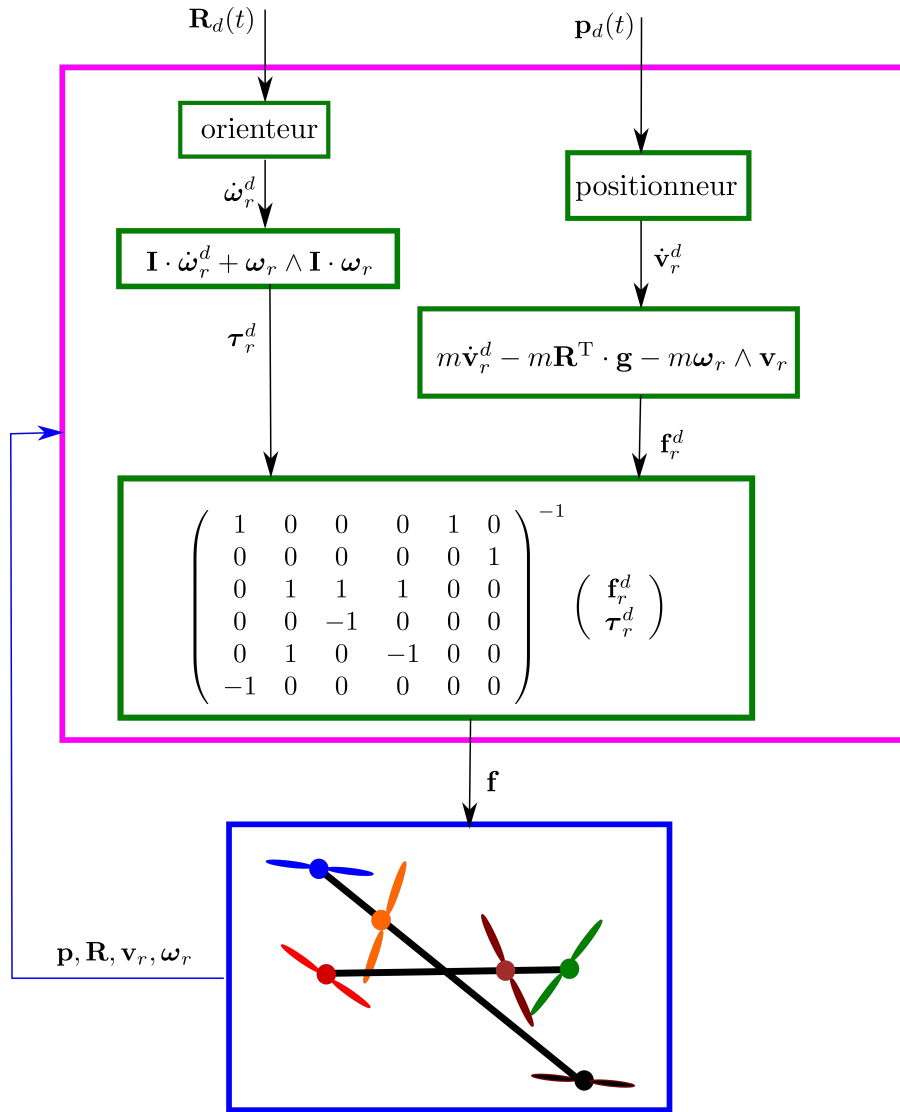


FIGURE 30 – Contrôleur pour l'hexarotor

La figure 31 illustre une situation d'atterrissage de l'hexarotor sur une plateforme mobile. Une vidéo est également disponible sur <https://youtu.be/s0HdIPY9Gj8>. La position et l'orientation de la plateforme sont données par

$$\mathbf{p}_d = \begin{pmatrix} \sin \frac{3t}{10} \\ \cos \frac{4t}{10} \\ \frac{1}{10} \sin \frac{3t}{10} \end{pmatrix}, \quad \mathbf{R}_d = \text{Exp} \begin{pmatrix} \sin t \\ \cos 2t \\ t \end{pmatrix}. \quad (115)$$

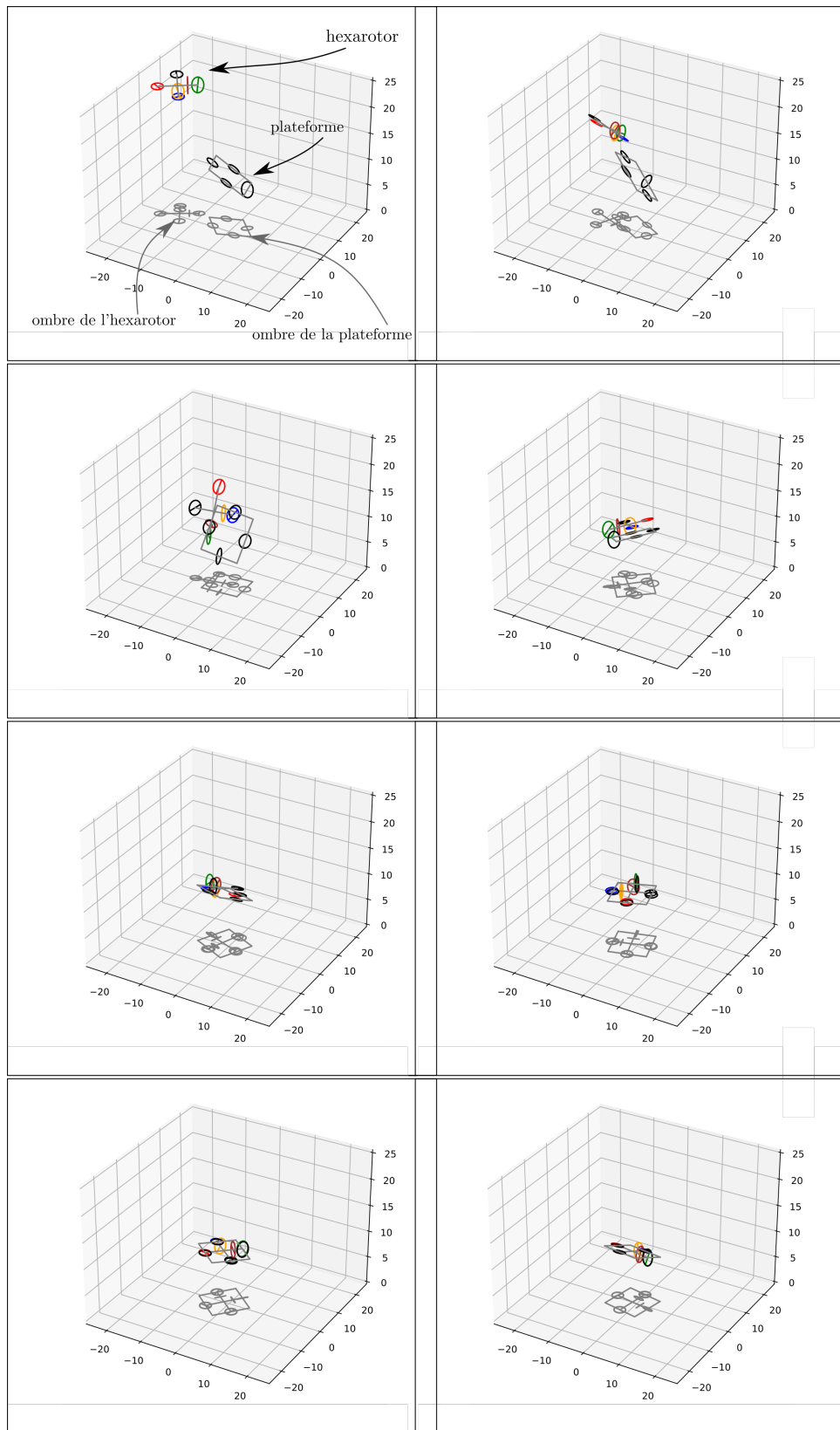


FIGURE 31 – Atterrissage sur une plateforme mobile

10 Géodésiques

Dans les paragraphes précédents, nous nous sommes focalisés sur la modélisation et la commande en orientation des drones évoluant dans l'espace. Nous avons insisté sur l'aspect géométrique qui permet d'obtenir une modélisation simple, générique et élégante. Cela a grandement simplifié la synthèse des régulateurs. Dans ce paragraphe, nous voulons montrer un autre aspect de la géométrie dans la commande des drones : les *géodésiques*.

Un drone se déplace souvent dans un environnement encombré avec des obstacles qu'il doit détecter et éviter. On cherche alors un chemin qui permette d'atteindre un objectif de façon optimale. Pour trouver de tels chemins, il existe un ensemble d'outils que l'on peut trouver dans le livre de Lavelle[19] et celui de Latombe [20]. Dans cet article, nous nous intéressons davantage à l'aspect géométrique de l'espace et nous voulons trouver le chemin le plus court pour être capable de passer d'une pose à l'autre dans l'espace des configurations. Un exemple classique est celui du chemin que doit prendre un avion pour aller de Paris à New-York. Il aura tout intérêt à rester sur la géodésique (chemin le plus court) qui relie les deux villes et qui reste sur l'ellipsoïde que forme la terre. Mais le chemin n'est pas forcément le plus court dans notre monde tridimensionnel. En effet, les distances sont à considérer dans l'espace des orientations ou plus généralement l'espace des configurations d'un robot. Et cet espace n'a rien de cartésien.

Considérons un drone qui a une orientation $\mathbf{R}_a$ et qui souhaite avoir l'orientation $\mathbf{R}_b$ et ceci en 1 seconde. Il devra décider d'un chemin $\mathbf{R}(t)$ dans $SO(3)$ tel que $\mathbf{R}(0) = \mathbf{R}_a$ et $\mathbf{R}(1) = \mathbf{R}_b$. La trajectoire de plus faible longueur (il faut voir ce chemin dans $SO(3)$ qui est un espace métrique) est donnée par

$$\mathbf{R}(t) = \exp(t \log(\mathbf{R}_b \mathbf{R}_a^T)) \mathbf{R}_a. \quad (116)$$

Cette formule se comprend relativement bien puisque $\log(\mathbf{R}_b \mathbf{R}_a^T)$ représente le vecteur de rotation qu'il nous faut suivre pour aller de $\mathbf{R}_a$ à $\mathbf{R}_b$.

10.1 Géodésique sur une variété différentielle

Considérons une variété de dimension n plongée dans $\mathbb{R}^m$, avec $m > n$. Cette variété correspond par exemple à l'espace des configurations pour notre drone.

Supposons que cette variété puisse être décrite par un ensemble d'équations de la forme

$$\mathbf{x} = \mathbf{r}(\mathbf{q}), \quad (117)$$

où $\mathbf{q} \in \mathbb{R}^n$ représente le vecteur des configurations (où le vecteur des degrés de liberté) de notre drone et $\mathbf{x} \in \mathbb{R}^m$. Dans la suite, de ce document, $\mathbf{x}$ sera un vecteur de $\mathbb{R}^3$ pour aider à la représentation graphique. Mais $\mathbf{x}$ pourrait être un vecteur appartenant à un espace de plus grande dimension dans laquelle serait plongée notre variété différentielle.

Supposons que nous puissions contrôler l'accélération $\ddot{\mathbf{q}}$ de $\mathbf{q}$ à travers un vecteur de commande $\mathbf{u}$. Nous voulons synthétiser un contrôleur (voir figure 32) qui calcule $\mathbf{u}$ tel que $\ddot{\mathbf{x}} - \mathbf{w}$ soit aussi petit que possible, où $\mathbf{w}$ est l'accélération désirée. Nous allons ensuite établir une correspondance avec la notion de géodésique qui correspond aux chemins les plus courts sur une variété différentielle [21], [22]. Ces notions pourront alors être utilisées avantageusement pour la planification de chemins pour les drones.

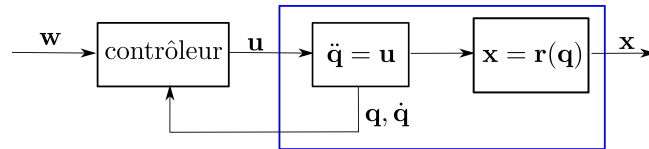


FIGURE 32 – Le contrôleur génère la commande $\mathbf{u}$ qui minimise $\|\ddot{\mathbf{x}} - \mathbf{w}\|$

Expression de l'accélération. A partir de l'équation $\mathbf{x}(t) = \mathbf{r}(\mathbf{q}(t))$, nous pouvons en déduire une expression pour $\ddot{\mathbf{x}}$ en fonction de $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$.

En effet

$$\dot{\mathbf{x}} = \underbrace{\frac{d\mathbf{r}}{d\mathbf{q}}(\mathbf{q})}_{\mathbf{J}} \dot{\mathbf{q}}. \quad (118)$$

La k ème composante de $\dot{\mathbf{x}}$ est

$$\dot{x}_k = \frac{dr_k}{d\mathbf{q}} \dot{\mathbf{q}}(t) = \sum_i \frac{\partial r_k}{\partial q_i} \dot{q}_i. \quad (119)$$

L'accélération correspondante est

$$\begin{aligned} \ddot{x}_k &= \frac{d}{dt} \left(\frac{dr_k}{d\mathbf{q}} \dot{\mathbf{q}}(t) \right) \\ &= \frac{d}{dt} \left(\frac{dr_k}{d\mathbf{q}} \right) \dot{\mathbf{q}}(t) + \frac{dr_k}{d\mathbf{q}} \ddot{\mathbf{q}}(t) \\ &= \sum_i \left(\frac{d}{dt} \frac{\partial r_k}{\partial q_i} \right) \dot{q}_i + \frac{dr_k}{d\mathbf{q}} \ddot{\mathbf{q}}(t) \end{aligned} \quad (120)$$

Or

$$\begin{aligned} \frac{d}{dt} \frac{\partial r_k}{\partial q_i} &= \frac{\partial \dot{r}_k}{\partial q_i} = \frac{\partial}{\partial q_i} \left(\sum_j \frac{\partial r_k}{\partial q_j} \dot{q}_j \right) \\ &= \sum_j \frac{\partial^2 r_k}{\partial q_i \partial q_j} \dot{q}_j \end{aligned} \quad (121)$$

Donc

$$\begin{aligned} \ddot{x}_k &= \left(\sum_i \sum_j \frac{\partial^2 r_k}{\partial q_i \partial q_j} \dot{q}_i \dot{q}_j \right) + \frac{dr_k}{d\mathbf{q}} \ddot{\mathbf{q}}(t) \\ &= \underbrace{\dot{\mathbf{q}}^T \mathbf{H}_k \dot{\mathbf{q}}}_{h_k(\mathbf{q}, \dot{\mathbf{q}})} + \frac{dr_k}{d\mathbf{q}} \ddot{\mathbf{q}} \end{aligned} \quad (122)$$

où $\mathbf{H}_k(\mathbf{q})$ est la matrice hessienne de $r_k(\mathbf{q})$.

Sous forme vectorielle (120) s'écrit

$$\ddot{\mathbf{x}} = \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{J}(\mathbf{q}) \ddot{\mathbf{q}}. \quad (123)$$

Equation de la géodésique. Nous voulons $\ddot{\mathbf{x}} = \mathbf{w}$. Or, $\mathbf{x}(t)$ doit rester sur la variété différentielle $\mathbf{r}(\mathbb{R}^n)$, et toutes les accélérations $\ddot{\mathbf{x}}$ ne sont pas autorisées. Seules les projections de $\ddot{\mathbf{x}}$ sur le plan tangent à la variété peuvent être arbitrairement choisies. Cela est illustré par la figure 33 où une voiture peut tourner à gauche ou à droite mais rester sur la surface. Ainsi, le régulateur ne pourra avoir aucun impact sur la composante de $\ddot{\mathbf{x}}$ orthogonale à la variété (flèche rouge sur la figure). Il est alors naturel de rechercher le contrôleur qui nous calcule $\mathbf{u}$ tel que $\|\ddot{\mathbf{x}} - \mathbf{w}\|$ soit le plus petit possible (voir figure 33).

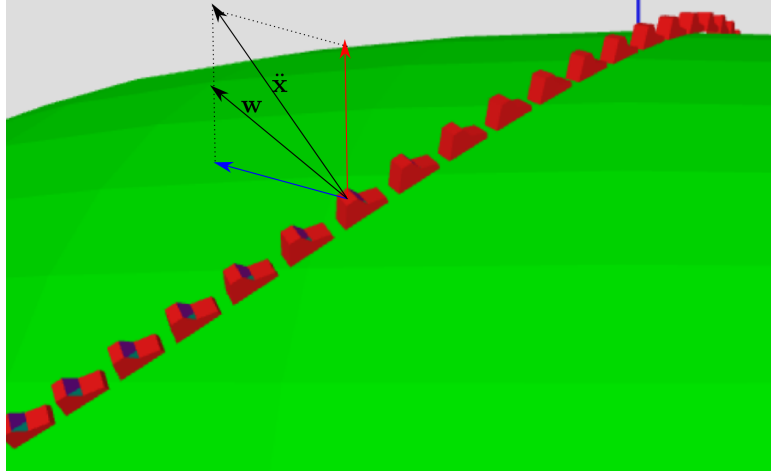


FIGURE 33 – Lorsque la voiture évolue sur la surface, elle peut avoir une action suivant la surface (action en bleu), mais ne peut pas influencer directement la composante transverse (rouge)

Pour $\mathbf{q}, \dot{\mathbf{q}}$ donnés, nous avons

$$\begin{aligned} \mathbf{u} &= \underset{\ddot{\mathbf{q}}}{\operatorname{argmin}} \|\ddot{\mathbf{x}}(\ddot{\mathbf{q}}) - \mathbf{w}\|^2 \\ &= \underset{\ddot{\mathbf{q}}}{\operatorname{argmin}} \|\mathbf{J} \ddot{\mathbf{q}} + \mathbf{h} - \mathbf{w}\|^2 \\ &= \underbrace{(\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T}_{\mathbf{J}^{-1}} (\mathbf{w} - \mathbf{h}) \end{aligned} \quad (124)$$

où $\mathbf{J}^{-1} = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T$ est l'inverse généralisée de $\mathbf{J}$ qui dépend de $\mathbf{q}$. L'erreur $\|\mathbf{J} \ddot{\mathbf{q}} + \mathbf{h} - \mathbf{w}\|^2$ est nulle lorsque $\mathbf{w}$ correspond à une accélération $\ddot{\mathbf{x}}$ faisable

Quand nous ne voulons aucune accélération suivant la surface, nous prenons $\mathbf{w} = \mathbf{0}$ et la trajectoire correspondante est appelée *géodésique*. Cherchons la géodésique qui commence en $\mathbf{q}(0) = \mathbf{q}_0$ et $\dot{\mathbf{q}}(0) = \dot{\mathbf{q}}_0$.

Puisque $\ddot{\mathbf{q}} = \mathbf{u}$, nous obtenons l'équation différentielle suivante :

$$\begin{aligned} \ddot{\mathbf{q}} &= -\mathbf{J}^{-1}(\mathbf{q}) \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) \\ \mathbf{q}(0) &= \mathbf{q}_0, \dot{\mathbf{q}}(0) = \dot{\mathbf{q}}_0 \\ \mathbf{x}(t) &= \mathbf{r}(\mathbf{q}(t)) \end{aligned} \quad (125)$$

10.2 Sur le tore

Afin d'illustrer l'application de l'équation de la géodésique, nous considérons un drone (ici un point) qui peut évoluer sur un tore décrit par l'équation

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \mathbf{r}(q_1, q_2) = \begin{pmatrix} (a_1 + a_2 \cos q_2) \cos q_1 \\ (a_1 + a_2 \cos q_2) \sin q_1 \\ a_2 \sin q_2 \end{pmatrix} \quad (126)$$

L'équation de la géodésique amène à l'équation différentielle suivante :

$$\ddot{\mathbf{q}} = \begin{pmatrix} \frac{2a_2 \dot{q}_1 \dot{q}_2 \sin q_2}{a_1 + a_2 \cos q_2} \\ -\sin q_2 \left(\frac{a_1}{a_2} + \cos q_2 \right) \dot{q}_1^2 \end{pmatrix}. \quad (127)$$

Une géodésique qui commence en $\mathbf{q}(0) = (0, 0)$, $\dot{\mathbf{q}}(0) = (0.1, 0.2)$, qui intègre l'équation en $\ddot{\mathbf{q}}$ pour $t \in [0, 50]$ est donnée sur la figure 34. Nous avons pris $a_1 = 10, a_2 = 6$.

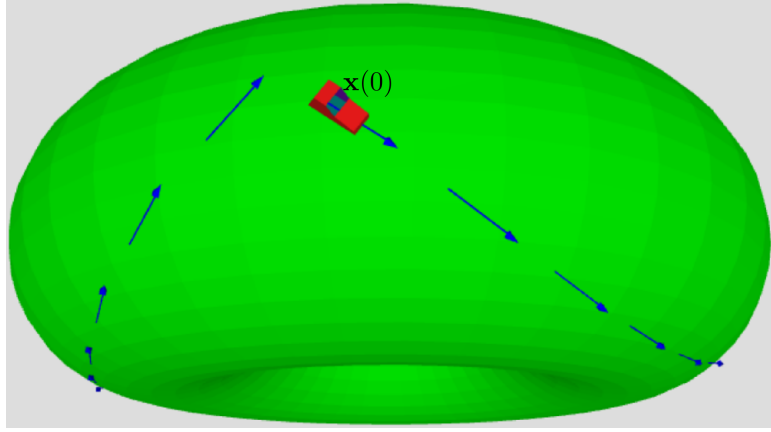


FIGURE 34 – La géodésique oscille autour de l'équateur

10.3 Sur l'ellipsoïde

Prenons maintenant l'ellipsoïde décrit par

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \mathbf{r}(q_1, q_2) = \begin{pmatrix} a_1 \cos q_2 \cos q_1 \\ a_2 \cos q_2 \sin q_1 \\ a_3 \sin q_2 \end{pmatrix} \quad (128)$$

avec $a_1 = 8, a_2 = 4, a_3 = 2$. L'équation différentielle qui nous permet de générer la géodésique est cette fois plus complexe à calculer et l'utilisation du calcul symbolique s'avère utile. Si nous connaissons les conditionales $\mathbf{q}(0), \dot{\mathbf{q}}(0)$, alors, il suffit d'intégrer cette équation différentielle pour avoir la géodésique, comme dans le cas du tore. Mais souvent, nous voulons une géodésique qui part d'une position initiale $\mathbf{q}(0)$ et qui va à une position finale $\mathbf{q}(t_f)$. Nous obtenons alors un problème aux limites de Dirichlet. Pour le résoudre, il suffit de lancer une méthode de dichotomie. Une telle méthode lance plusieurs simulations avec différents $\dot{\mathbf{q}}(0)$ en cherchant à se rapprocher de la position finale.

La figure 35 montre une géodésique qui part de $\mathbf{q}(0) = (0, 0)$ et telle que $\mathbf{q}(10) = (2, 1)$, obtenue par une telle méthode.

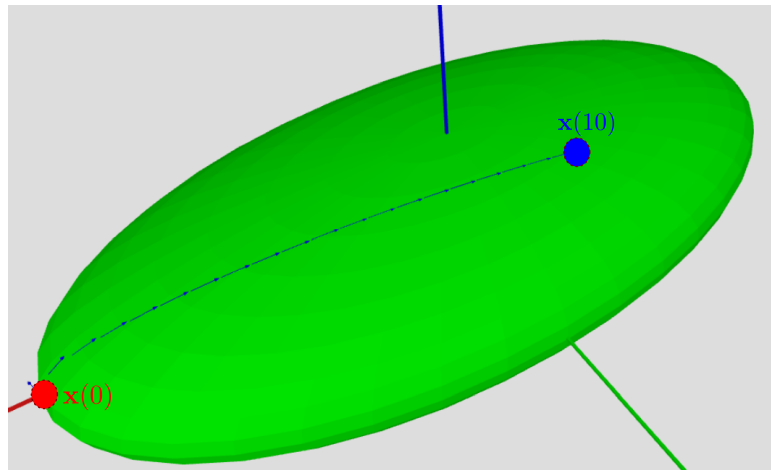


FIGURE 35 – Géodésique entre un point initial et un point final sur l'ellipsoïde

11 Conclusion

Dans cet article, nous avons donné un ensemble d'outils de nature géométrique permettant de modéliser une certaine classe de drones, avec un corps rigide, des propulseurs et des gouvernes.

Pour se déplacer dans la bonne direction, le drone doit avant tout s'orienter correctement. Pour cela, nous avons insisté sur la structure géométrique et algébrique des orientations tri-dimensionnelles représentées par l'ensemble $SO(3)$. Pour un robot mobile plus complexe avec des articulations, la posture interne évoluera dans un espace plus complexe, et les outils présentés ici devront être étendus [8].

Les modèles utilisés sont souvent simplistes car beaucoup d'effets ont été négligés, comme certains coefficients de traînée, les turbulences, les masses ajoutées qui ne sont pas forcément constantes. Cependant, ces modèles ont pour principal objectif de synthétiser un régulateur. Ainsi, le régulateur se fonde sur un modèle simple qui nous permet d'espérer que tout va bien se passer dans la réalité. Un régulateur de vitesse de voiture, par exemple, se base sur un modèle du premier ordre reliant l'accélérateur et la vitesse. Il ne prend pas en compte le nombre de personnes dans la voiture, le fait qu'il y ait une remorque ou non, etc. Pourtant, il fonctionne très bien dans la pratique. Ainsi, un modèle cinématique simple peut convenir pour la conception de la loi de contrôle, mais il est fondamental de le valider sur un modèle le plus réaliste possible en y ajoutant des incertitudes et du bruit de mesure. Enfin, seuls des expérimentations sur des drones réels permettront de valider la fiabilité et la robustesse des algorithmes développés.

Les programmes PYTHON associés aux simulations sont disponibles sur <https://www.ensta-bretagne.fr/jaulin/teching.html>.

Références

- [1] E. Dombre and W. Khalil. *Robot manipulators : modeling, performance analysis and control*. ISTE Ltd., London, 2007. :07
- [2] F. Benamar and C. Grand. Robotique mobile : conception, modélisation et commande. *Techniques de l'ingénieur*, 2016. :grand :16
- [3] F. Plumet, Y. Biere, and F. Le Bars. Les voiliers robotisés. *Techniques de l'ingénieur*, 2016. :18
- [4] R. Beard and T. McLain. *Small Unmanned Aircraft, Theory and Practice*. Princeton University Press, 2012. :McLain

- [5] V. Creuze. Robots marins et sous-marins ; perception, modélisation, commande. *Techniques de l'ingénieur*, 2014. 2
- [6] J.P. Laumond. *La robotique mobile*. Hermès, Paris, France, 2001. 2
- [7] L. Jaulin. *La Robotique Mobile, Cours et Exercices*. ISTE, 2015. 2
- [8] F. Boyer, M. Alamir, D. Chablat, W. Khalil, A. Leroyer, and P. Lemoine. Robot anguille sous-marin en 3d. In *Techniques de l'Ingénieur*, 2006. 2, 11
- [9] D. Filliat. Cartographie et localisation simultanées en robotique mobile. *Les Techniques de l'Ingénieur*, 2014. 2
- [10] J. Sola, J. Deray, and D. Atchuthan. A micro lie theory for state estimation in robotics. *arXiv :1812.01537*, 2018. 3.4
- [11] Richard M. Murray, Zexiang Li, and Shankar Sastry. *A mathematical introduction to robotics manipulation*. CRC Press, 1994. :Murray
- [12] P.J. Olver. *Applications of Lie Groups to Differential Equations*. Graduate Texts in Mathematics. Springer New York, 2012. 3.6
- [13] L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79(3) :497–516, 1957. 3.8, 3.9.1
- [14] T. Hamel and R. Mahony. Attitude estimation on $so(3)$ based on direct inertial measurements. In *Int. Conf. Robot. Automat. (ICRA 06)*, pages 2170–2175, 2006. 4.3
- [15] T. Fossen. *Marine Control Systems : Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*. Marine Cybernetics, 2002. :auv
- [16] T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski. Dynamic modelling and configuration stabilization for an x4-flyer. In *IFAC Proceedings Volumes*, volume 35, pages 217–222, 2002. 6
- [17] H.K. Khalil. *Nonlinear Systems, Third Edition*. Prentice Hall, 2002. 6
- [18] R.W. Prouty. *Helicopter Performance, Stability and Control*. Krieger Publishing Company, USA, 1986. 8
- [19] S. LaValle. *Planning algorithm*. Cambridge University Press, 2006. :06
- [20] J.C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA, 1991. 10
- [21] P. Pokorný. Geodesics revisited. *Chaotic Modeling and Simulation*, pages 281–298, 2012. 10.1
- [22] M. Berger. *Geometry I and II*. Springer-Verlag, Berlin, 1987. 10.1