

Perception et Navigation

Notes de cours

Philippe Mouyon
ONERA - Centre de Toulouse
Département de Contrôle des Systèmes et Dynamique du vol



Je sais où je suis car
je sais d'où je viens.

[Almanach Breton]

En navigation, plus le trait est large,
plus tu as de chance d'être dessus.

[Anonyme]



Table des matières

1	Introduction	5
1.1	Mise en garde	5
1.2	Plan de travail	6
1.3	Objectifs	8
2	Représentation du comportement d'un engin	10
2.1	Représentations de l'attitude	10
2.1.1	Représentation par les angles d'Euler	11
2.1.1.1	Formulation	11
2.1.1.2	Retrouver les angles d'Euler à partir de R . .	12
2.1.1.2.a	Cas nominal	12
2.1.1.2.b	Cas θ voisin de $\pm\pi/2$	12
2.1.2	Représentation axiale	13
2.1.2.1	Formulation	13
2.1.2.2	Retrouver angle et direction à partir de R . .	13
2.1.2.2.a	Cas nominal	13
2.1.2.2.b	Cas α petit	14
2.1.2.2.c	Cas α voisin de π	15
2.1.3	Représentation par quaternion	15
2.1.3.1	Formulation	15
2.1.3.2	Retrouver le quaternion à partir de R	16
2.1.3.3	Quaternion et angles d'Euler	17
2.1.3.4	Quaternion et représentation axiale des ro- tation	18
2.1.3.5	Formulation matricielle des quaternions . . .	19
2.2	Cinématique	20
2.2.1	Equations de la cinématique	20
2.2.1.1	Cinématique de translation	20
2.2.1.2	Cinématique de rotation avec R	21

2.2.1.3	Cinématique de rotation avec q	21
2.2.2	Schéma d'intégration numérique	22
2.2.2.1	Cinématique de translation	22
2.2.2.2	Cinématique de rotation	22
2.2.2.3	Interpolation d'attitudes	24
2.3	Validation numérique	25
2.3.1	Formules de passage entre les représentations de l'at- titude	25
2.3.1.1	Passage (ψ, θ, φ) à R	25
2.3.1.2	Passage de R à (ψ, θ, φ)	26
2.3.1.3	Passage de (α, u) à R	26
2.3.1.4	Passage de R à (α, u)	26
2.3.1.5	Passage de q à R	27
2.3.1.6	Passage de R à q	27
2.3.2	Schémas d'intégration	28
2.3.2.1	Référence des mouvements	28
2.3.2.2	Validation sur quelques trajectoires particu- lières	29
2.3.2.3	Validation sur trajectoires complexes	33
3	Capteurs	37
3.1	Introduction	37
3.2	Magnétomètre	39
3.2.1	Caractéristiques du champ magnétique terrestre	39
3.2.2	Variations spatiales du champ magnétique	40
3.2.3	Variations temporelles du champ magnétique	42
3.2.4	Quelques mesures réalisées	42
3.3	Accéléromètres et gyromètres	44
4	Estimation d'état récursive : un état de l'art méthodolo- gique	46
4.1	Introduction	46
4.2	Position du problème	47
4.2.1	Modèles stochastiques	47
4.2.2	Estimateurs d'état	48
4.3	Equations générales du filtrage	49
4.3.1	Prédiction	49
4.3.2	Correction	50
4.4	Solutions exactes	51
4.4.1	Systèmes linéaires gaussiens	51

4.4.2	Chaînes de Markov finies	52
4.4.3	Systèmes conditionnellement gaussiens	54
4.5	Approximations paramétriques	54
4.5.1	Approximation gaussienne	55
4.5.1.1	Filtre de Kalman étendu	55
4.5.1.2	Filtre de Kalman sans parfum	56
4.5.2	Somme de gaussiennes	57
4.5.2.1	Approximation multi-gaussienne	57
4.5.2.2	Equations du filtrage multi-gaussien	58
4.5.2.3	Cas de non linéarités fortes	59
4.5.2.4	Autres bases de fonctions	60
4.6	Approximations non paramétriques	61
4.6.1	Méthodes déterministes	61
4.6.2	Méthodes stochastiques	61
4.6.2.1	L'échantillonnage aléatoire	62
4.6.2.2	Le filtrage particulière	62
4.7	Systèmes hybrides	63

Chapitre 1

Introduction

1.1 Mise en garde

Ce document est la première version du support de cours. Il est donc très imparfait et je m'en excuse. Il contient dans les différents chapitres :

- Une modélisation de la cinématique des engins (en translation et en rotation), ainsi que les outils associés (matrices de rotation et quaternions en particulier, discrétisation pour la simulation).
- Une description (trop) courte et encore incomplète de la modélisation des capteurs utilisés pour la navigation.
- Une introduction (trop) générale au filtrage optimal (filtrage de Kalman) qui est l'outil utilisé pour concevoir les algorithmes de navigation à partir des modèles cinématiques.

Il manque donc en particulier à ce support des choses essentielles que nous avons vues en cours :

- Comment modéliser et simuler les incertitudes aléatoires dans les modèles.
- Une étude de la navigation inertielle pure (en translation) dans des cas particuliers importants (robot terrestre, navire, fusée).
- Une introduction à la navigation inertielle pure en translation et rotation selon les types de centrales de navigation (gyrostabilisées ou bien à composants liés).

Mais il manque aussi des choses essentielles que nous aurions du voir en cours et que nous ne verrons qu'en bureau d'étude :

- Comment on conçoit un système de navigation inertiel hybridé avec d'autres capteurs (et la nécessaire observabilité des états du modèle).
- Le réglage d'un filtre de navigation (nous l'avons vu en cours mais

beaucoup trop rapidement).

- Comment on limite la complexité d'un filtre de navigation (en transformant les équations de mesure, en simplifiant les modèles, etc...).

Le plan de travail donné ci-dessous correspond au découpage proposé initialement. En fait je n'ai pas du tout suivi le même enchainement. J'ai préféré d'abord considérer uniquement la problématique de la navigation pour les engins se déplaçant en translation uniquement. Puis ensuite seulement je suis passé au cas des engins en translation et rotation.

Ceci permet d'introduire graduellement les problèmes de représentation d'état, discrétisation, modélisation des perturbations aléatoires, algorithmes de navigation inertielle pure puis multi-senseurs.

Lors du passage en rotation on insiste sur la complexité induite par la non linéarité (singularités ou non minimalité des représentations de l'attitude, discrétisation de dynamiques non linéaires, ...).

Vous avez été nombreux à me demander des références bibliographiques. Outre les nombreux livres de Radix de la collection Cépaduès que vous devriez trouver facilement à la bibliothèque de l'école, je trouve que l'ouvrage de Titterton [1] sur les centrales inertielles à composants liés est très bien fait.

1.2 Plan de travail

1. Perception : OUTILS MATHEMATIQUES

(a) Mouvements de translation d'un engin (1 h - Philippe MOUYON)

- i. Représentation de la position
- ii. La cinématique de translation
- iii. Changement de repère
- iv. Intégration numérique
- v. Dynamique versus cinématique

(b) Mouvements de rotation d'un engin (1 h - Philippe MOUYON)

- i. Les différents paramétrages de la représentation de l'attitude
- ii. Relations de passage entre les représentations
- iii. La cinématique de rotation
- iv. Intégration numérique

- (c) La théorie du filtrage optimal (1 h - Philippe MOUYON)
 - i. Estimation d'état et fusion de données - L'approche bayésienne
 - ii. Les équations du filtrage optimal
 - iii. Cas linéaire gaussien : le filtre de Kalman
 - iv. Extensions : filtre de Kalman étendu, sans parfum
 - v. Systèmes hybrides et filtrage multi-gaussien
- (d) Mise en œuvre et réglage d'un filtre de Kalman (1 h - Philippe MOUYON)
 - i. Modèle physique et modèle de synthèse
 - ii. Initialisation du filtre, et réglages
 - iii. Modèles simplifiés : dynamique approchée, pseudo-mesures,

2. Perception : CAPTEURS PROPRIOCEPTIFS

- (a) GNSS (1 h - Mohamed SAHMOUDI)
 - i. Les mesures brutes : pseudo-distances, vitesses
 - ii. Localisation par récepteur GNSS
 - iii. Modélisation des mesures, et des pannes
- (b) IMU et AHRS (1 h - Philippe MOUYON)
 - i. Les mesures inertielles : accéléromètres et gyromètres
 - ii. Caractérisation des mesures inertielles
 - iii. Modélisation des mesures et des pannes
 - iv. Magnétomètres
 - v. Baromètres
 - vi. ?

3. LOCALISATION

- (a) La localisation inertielle pure (1 h - Philippe MOUYON)
 - i. A la surface de la terre : période de Schuller
 - ii. En vol : nécessité d'une hybridation
- (b) La localisation par mesure GPS (2 h - Mohamed SAHMOUDI)

- i. Synthèse du filtre de localisation
- ii. Filtre avec détection des arrêts
- (c) Quelques filtres hybrides (1 h - Philippe MOUYON)
 - i. Fusion Nz / baro pour l'estimation de l'altitude
 - ii. Fusion Odomètre / IMU pour les véhicules terrestres
 - iii. Fusion IMU / Vitesse GPS pour l'estimation d'attitude
 - iv. ?

4. BE NAVIGATION D'UN DRONE

- (a) Une centrale d'attitude pour drone (3 h - Philippe MOUYON)
 - i. Synthèse exacte : analyse de performance, impact des paramètres de réglages, effet d'une perte de mesure GPS.
 - ii. Synthèse sur modèle approché.

1.3 Objectifs

Les méthodes et outils présentés sont dédiés à la navigation multi-senseurs des engins, et plus particulièrement au développement d'un logiciel réalisant cette fonction. Ce travail est essentiellement justifié par deux besoins :

- **Maîtriser le code embarqué**

La maîtrise du code de navigation embarqué est nécessaire à la sécurité des engins autonomes. Les matériels du type IMU (Inertial Measurement Unit), INS (Inertial Navigation System), ou encore AHRS (Attitude and Heading Reference System) sont généralement livrés avec un logiciel de navigation. Mais on constate que les performances obtenues en situation réelle peuvent être très variables selon le domaine d'emploi, et cela sans pouvoir mettre toujours clairement en évidence la relation *performances / domaine opérationnel*. Maîtriser le code embarqué permettra de rechercher les raisons des défaillances ou dysfonctionnements constatés dans certaines situations.

- **Obtenir un code reconfigurable**

Le besoin d'un code reconfigurable peut provenir de divers objectifs d'emploi. Ainsi on voudra par exemple que le logiciel de navigation fonctionne

- sur diverses plates-formes dont les dynamiques sont hétérogènes (avion, hélicoptère, voiture)

- en utilisant des sources d'information variables (selon les équipements, mais aussi selon les défaillances des capteurs)
- et avec des objectifs d'estimation différents (attitude, vitesse, position absolue ou relative, ...).

Dans cette optique, les différentes tâches à réaliser sont les suivantes : modélisation et calibration des capteurs, modèle de simulation, méthodes de filtrage, implémentation.

Chapitre 2

Représentation du comportement d'un engin

2.1 Représentations de l'attitude

L'attitude est l'orientation du repère mobile \mathcal{R}_m par rapport au repère fixe \mathcal{R}_0 . Il existe de nombreux paramétrages pour représenter l'attitude [2]. Ceux que nous utilisons sont les suivants :

- **Matrice de rotation R**

La matrice de rotation R est la matrice des cosinus directeurs. Elle amène le repère fixe \mathcal{R}_0 sur le repère mobile \mathcal{R}_m . C'est-à-dire que si v_0 et v_m sont les coordonnées du vecteur v exprimées dans les repères fixe et mobile alors :

$$v_0 = R v_m$$

Cette matrice de rotation satisfait à l'ensemble de contraintes :

$$R^T R = I$$

On a donc 9 paramètres, et 6 contraintes algébriques, donc bien seulement 3 degrés de liberté. Tous les repères considérés ici sont directs, ce qui implique que $\det(R) = 1$.

Cette représentation n'a aucune singularité. A toute attitude correspond une et une seule matrice de rotation.

- **Angles d'Euler (ψ, θ, φ)**

On considère les 3 angles d'Euler (ψ, θ, φ) de la convention aéronautique. Ils définissent trois rotations successives amenant de \mathcal{R}_0 à \mathcal{R}_m :

$$R = R_{ZYX}(\psi, \theta, \varphi) = R_Z(\psi) R_Y(\theta) R_X(\varphi)$$

On a donc ici 3 paramètres indépendants correspondant aux 3 degrés de liberté.

Cette représentation présente deux singularités en $\theta = \pi/2$ et $\theta = -\pi/2$, où les angles ψ et φ sont indéterminés.

- **Représentation axiale (α, u)**

Puisqu'on change de repère par une rotation, il suffit de connaître l'angle α et l'axe u de la rotation pour définir l'attitude.

$$R = R(\alpha, u)$$

Le vecteur définissant l'axe de la rotation est un vecteur unitaire

$$u^T . u = 1$$

On a donc ici 4 paramètres, liés par 1 relation algébrique, donc toujours 3 degrés de liberté.

Cette représentation présente une singularité en $\alpha = 0$, où la direction u est indéterminée. Il y a aussi une singularité générale sur le signe puisque :

$$R(-\alpha, -u) = R(\alpha, u)$$

- **Quaternion q**

L'attitude peut enfin être définie par la donnée d'un quaternion vectoriel unitaire q :

$$q^T q = 1$$

On a encore 4 paramètres et 1 contrainte, donc 3 degrés de liberté.

Cette représentation présente aussi une singularité sur le signe provenant du fait que :

$$R(-q) = R(q)$$

2.1.1 Représentation par les angles d'Euler

2.1.1.1 Formulation

Selon la convention aéronautique, on passe du repère fixe au repère mobile par trois rotations successives :

$\mathcal{R}_Z(\psi)$ - la rotation d'axe Z et d'angle ψ

$\mathcal{R}_Y(\theta)$ - la rotation d'axe Y et d'angle θ

$\mathcal{R}_X(\varphi)$ - la rotation d'axe X et d'angle φ

En aéronautique l'axe X est dirigé vers l'avant, l'axe Y sur l'aile droite et l'axe Z vers le bas.

La composition des trois rotations $\mathcal{R}_Z(\psi) \mathcal{R}_Y(\theta) \mathcal{R}_X(\varphi)$ donne la matrice changement de repère R :

$$R = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \varphi - \sin \psi \cos \varphi & \cos \psi \sin \theta \cos \varphi + \sin \psi \sin \varphi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \varphi + \cos \psi \cos \varphi & \sin \psi \sin \theta \cos \varphi - \cos \psi \sin \varphi \\ -\sin \theta & \cos \theta \sin \varphi & \cos \theta \cos \varphi \end{bmatrix} \quad (2.1)$$

2.1.1.2 Retrouver les angles d'Euler à partir de R

2.1.1.2.a Cas nominal

Les angles d'Euler s'obtiennent à partir de la matrice de rotation par :

$$\begin{aligned} \psi &= \arctan(R_{21}, R_{11}) \\ \theta &= -\arcsin(R_{31}) \\ \varphi &= \arctan(R_{32}, R_{33}) \end{aligned} \quad (2.2)$$

Le calcul de ψ et φ par ces formules suppose $\theta \neq \pm\pi/2$.

2.1.1.2.b Cas θ voisin de $\pm\pi/2$

Quand $\theta = \pm\pi/2$ les angles ψ et φ sont indéterminés. Cependant dans une procédure itérative (où les angles évoluent au cours du temps) il est important d'assurer la continuité par rapport aux valeurs précédentes.

Prenons $\theta = \varepsilon \pi/2$ où $\varepsilon = \pm$. La matrice R devient :

$$R = \begin{bmatrix} 0 & -\sin(\psi - \varepsilon \varphi) & -\varepsilon \cos(\psi - \varepsilon \varphi) \\ 0 & \cos(\psi - \varepsilon \varphi) & \varepsilon \sin(\psi - \varepsilon \varphi) \\ -\varepsilon & 0 & 0 \end{bmatrix}$$

La seule information que l'on peut extraire de la matrice R est donc la combinaison linéaire $D \triangleq \psi - \varepsilon \varphi$. Cela se fera par exemple en notant que :

$$\begin{aligned} -R_{12} + \varepsilon R_{23} &= 2 \sin(\psi - \varepsilon \varphi) \\ +R_{22} + \varepsilon R_{13} &= 2 \cos(\psi - \varepsilon \varphi) \end{aligned}$$

d'où

$$D = \arctan(-R_{12} + \varepsilon R_{23}, R_{22} + \varepsilon R_{13}) \quad (2.3)$$

Introduisons $S \triangleq \psi + \varepsilon \varphi$. et notons \hat{S} la valeur précédente de S (dans une procédure itérative). On calculera par continuité :

$$\begin{aligned}\psi &= (\hat{S} + D)/2 \\ \varphi &= \varepsilon(\hat{S} - D)/2\end{aligned}\tag{2.4}$$

2.1.2 Représentation axiale

2.1.2.1 Formulation

La matrice de rotation d'angle α autour de la direction u (unitaire) s'écrit :

$$R = I + \sin \alpha u_{\times} + (1 - \cos \alpha) u_{\times}^2\tag{2.5}$$

où u_{\times} est la matrice antisymétrique associée au vecteur $u = (u_x, u_y, u_z)^T$ et définie par

$$u_{\times} = S(u) \triangleq \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix}\tag{2.6}$$

On notera vex l'opérateur inverse de S : $u = \text{vex}(u_{\times})$.

Quelques propriétés utiles de la matrice u_{\times} sont rappelées :

$$\begin{aligned}u_{\times} v &= u \times v & \text{d'où } u_{\times} u &= 0 \\ u_{\times}^2 &= u \cdot u^T - (u^T u) I & \text{d'où } \text{tr}(u_{\times}^2) &= -2 u^T u \\ u_{\times}^3 &= -(u^T u) u_{\times} & \text{d'où } R u_{\times} &= (1 - u^T u + u^T u \cos \alpha) u_{\times} + \sin \alpha u_{\times}^2\end{aligned}\tag{2.7}$$

Elles sont données pour un u quelconque, mais se simplifient grandement quand u est unitaire.

2.1.2.2 Retrouver angle et direction à partir de R

2.1.2.2.a Cas nominal

Pour retrouver l'angle et l'axe de la rotation à partir de la matrice de rotation on examine ses parties symétrique (Π_s) et antisymétrique (Π_a) de R :

$$\begin{aligned}\Pi_a(R) &\triangleq \frac{1}{2}(R - R^T) = \sin \alpha u_{\times} \\ \Pi_s(R) &\triangleq \frac{1}{2}(R + R^T) = I + (1 - \cos \alpha) u_{\times}^2\end{aligned}\tag{2.8}$$

On en déduit que :

$$\sin \alpha u = \frac{1}{2} \begin{bmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{bmatrix}\tag{2.9}$$

et comme le vecteur u cherché est unitaire, $\text{tr}(u_\times^2) = -2$, et on a donc aussi :

$$\cos \alpha = \frac{1}{2} (\text{tr } R - 1) \quad (2.10)$$

Le choix du signe de α est libre car la rotation d'angle $-\alpha$ autour de $-u$ est identique à celle d'angle α autour de u . On pourra donc calculer l'angle $\alpha \in [0, \pi]$ par la fonction arccosinus appliquée à (2.10). Puis, si $\sin \alpha \neq 0$, on trouvera la direction u en divisant (2.9) par $\sin \alpha$.

Quand l'angle est petit ou voisin de π , les formules ci-dessus sont mal conditionnées pour trouver α et u . Ces deux cas particuliers sont traités ci-après.

2.1.2.2.b Cas α petit

Dans le cas où l'angle α est petit, le manque de précision sur a n'est pas très important puisque qu'il n'y a presque pas de rotation. N'importe quelle direction convient. Cependant dans une procédure itérative (où la rotation évolue au cours du temps) il est important d'assurer la continuité de a par rapport à une valeur précédente.

Dans ce but examinons la matrice $M = R u_\times$:

$$\begin{aligned} M &\triangleq R u_\times \\ &= \cos \alpha u_\times + \sin \alpha u_\times^2 \\ &= \cos \alpha u_\times + \sin \alpha (u.u^T - u^T.u I) \\ &= \cos \alpha u_\times + \sin \alpha (u.u^T - I) \quad \text{car } u \text{ est unitaire} \end{aligned}$$

Ses parties symétrique et antisymétrique s'écrivent donc :

$$\begin{aligned} \Pi_a(R u_\times) &= \cos \alpha u_\times \\ \Pi_s(R u_\times) &= \sin \alpha (u.u^T - I) \end{aligned}$$

De façon pratique, on utilisera \hat{u} la valeur précédente de u , et on calculera :

$$\Pi_a(R \hat{u}_\times) \approx \cos \alpha u_\times \approx u_\times \quad (2.11)$$

2.1.2.2.c Cas α voisin de π

Dans le cas où l'angle α est voisin de π , il est possible de calculer les carrés des composantes de u facilement et avec précision car :

$$\begin{aligned}
\Pi_s(R) &= \frac{1}{2} (R + R^T) \\
&= I + (1 - \cos \alpha) u_\times^2 \\
&= I + (1 - \cos \alpha) \begin{pmatrix} u.u^T - u^T.u & I \end{pmatrix} \\
&= I + (1 - \cos \alpha) \begin{pmatrix} u.u^T - I \end{pmatrix} & \text{car } u \text{ est unitaire} \\
&= \cos \alpha I + (1 - \cos \alpha) u.u^T
\end{aligned}$$

On en déduit que :

$$u.u^T = (\Pi_s(R) - \cos \alpha I) / (1 - \cos \alpha) \quad (2.12)$$

Les carrés des composantes de u sont donc sur la diagonale de cette matrice. Et à partir des termes extra-diagonaux on en déduit tout le vecteur u au sens près, c'est-à-dire qu'on trouve

$$\tilde{u} = \varepsilon u \quad \text{où } \varepsilon = \pm 1 \quad (2.13)$$

La valeur de ε ne peut pas être déterminée. Cela provient du fait que les rotations d'angle π d'axe u et $-u$ sont identiques.

Cependant dans une procédure itérative (où la rotation évolue au cours du temps) il est important d'assurer la continuité de a par rapport à une valeur précédente \hat{u} . On choisira simplement pour cela

$$\varepsilon = \text{sign}(\hat{u}^T \tilde{u}) \quad (2.14)$$

assurant ainsi que le sens n'est pas modifié.

2.1.3 Représentation par quaternion

2.1.3.1 Formulation

On considère ici que l'attitude est représentée par un quaternion vectoriel unitaire :

$$q = \begin{bmatrix} s \\ x \\ y \\ z \end{bmatrix} \quad \text{avec } q^T q = 1 \quad (2.15)$$

Si on introduit le vecteur (non unitaire)

$$v = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2.16)$$

La matrice de rotation associée à q est :

$$R \triangleq I + 2s v_{\times} + 2v_{\times}^2 \quad (2.17)$$

On prendra garde à ce que R ainsi défini n'est une matrice de rotation que si q est unitaire. On peut en effet vérifier que : $R^T R = I + 4(1 - q^T q)v_{\times}^2$, qui ne vaut I que si $q^T q = 1$.

La matrice R est la rotation d'angle α et d'axe unitaire u définis par :

$$\begin{aligned} s &= \cos\left(\frac{\alpha}{2}\right) \\ v &= \sin\left(\frac{\alpha}{2}\right) u \end{aligned} \quad (2.18)$$

L'expression de R en fonction des composantes de q est polynomiale :

$$R = \begin{bmatrix} 1 - 2(y^2 + z^2) & 2(xy - sz) & 2(xz + sy) \\ 2(yx + sz) & 1 - 2(x^2 + z^2) & 2(yz - sx) \\ 2(zx - sy) & 2(zy + sx) & 1 - 2(x^2 + y^2) \end{bmatrix} \quad (2.19)$$

Quelques définitions et propriétés intéressantes des quaternions sont à noter :

$$\begin{aligned} q_1 \otimes q_1 &\triangleq \begin{bmatrix} s_1 s_2 - v_1^T v_2 \\ s_1 v_2 + s_2 v_1 + v_1 \times v_2 \end{bmatrix} && \text{Le produit de quaternion (unitaires) représente } R_1 R_2 \\ q^* &\triangleq \begin{bmatrix} s \\ -v \end{bmatrix} && \text{Le quaternion conjugué vérifie } q^* \otimes q = \begin{bmatrix} q^T q \\ 0 \end{bmatrix} \\ q^{-1} &= q^* / (q^T q) && \text{L'inverse d'un quaternion (unitaire) représente } R^{-1} \end{aligned}$$

2.1.3.2 Retrouver le quaternion à partir de R

On considère les parties symétriques et antisymétriques de R :

$$\begin{aligned} \Pi_a(R) &= 2s v_{\times} \\ \Pi_s(R) &= I + 2v_{\times}^2 \\ &= I + 2(v v^T - v^T v I) \end{aligned} \quad (2.20)$$

La première relation donne les doubles produits sx , sy et sz :

$$\begin{aligned} s x &= \frac{1}{4}(R_{32} - R_{23}) \\ s y &= \frac{1}{4}(R_{13} - R_{31}) \\ s z &= \frac{1}{4}(R_{21} - R_{12}) \end{aligned} \quad (2.21)$$

Les autres doubles produits sont obtenus en considérant les termes extra-diagonaux de la seconde relation :

$$\begin{aligned}xy &= \frac{1}{4}(R_{12} + R_{21}) \\xz &= \frac{1}{4}(R_{13} + R_{31}) \\yz &= \frac{1}{4}(R_{23} + R_{31})\end{aligned}\tag{2.22}$$

La seconde équation se réarrange sous la forme :

$$v v^T - v^T . v I = \frac{1}{2} (\Pi_s(R) - I)$$

Sur la diagonale on trouve donc :

$$\begin{aligned}-(y^2 + z^2) &= \frac{1}{2}(R_{11} - 1) \\-(x^2 + z^2) &= \frac{1}{2}(R_{22} - 1) \\-(x^2 + y^2) &= \frac{1}{2}(R_{33} - 1)\end{aligned}$$

On en déduit alors :

$$\begin{aligned}x^2 &= \frac{1}{4}(R_{11} - R_{22} - R_{33} + 1) \\y^2 &= \frac{1}{4}(-R_{11} + R_{22} - R_{33} + 1) \\z^2 &= \frac{1}{4}(-R_{11} - R_{22} + R_{33} + 1)\end{aligned}\tag{2.23}$$

Et enfin par sommation (et du fait que q est unitaire) on obtient :

$$s^2 = \frac{1}{4}(R_{11} + R_{22} + R_{33} + 1)\tag{2.24}$$

Pour extraire les composantes du quaternion sans problème numérique on procède comme suit. La valeur absolue du plus grand des carrés est choisie comme première composante. Les valeurs des autres composantes sont déduites des doubles produits par division. Le signe du quaternion n'est donc pas déterminé. Ce qui est normal puisque q et $-q$ correspondent à la même matrice de rotation. Au lieu de chercher le plus grand des carrés, on peut aussi se contenter d'en prendre un qui est supérieur à $1/4$; Il y en a nécessairement un puisque la somme des carrés vaut un. Cela évite le calcul systématique de tous les carrés et de leur classement.

2.1.3.3 Quaternion et angles d'Euler

On a vu comment passer des angles d'Euler à la matrice de rotation, et de cette rotation au quaternion. On peut passer directement des angles d'Euler au quaternion.

$$q = \begin{bmatrix} s \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos(\frac{\psi}{2}) \cos(\frac{\theta}{2}) \cos(\frac{\varphi}{2}) & + \sin(\frac{\psi}{2}) \sin(\frac{\theta}{2}) \sin(\frac{\varphi}{2}) \\ -\sin(\frac{\psi}{2}) \sin(\frac{\theta}{2}) \cos(\frac{\varphi}{2}) & + \cos(\frac{\psi}{2}) \cos(\frac{\theta}{2}) \sin(\frac{\varphi}{2}) \\ \cos(\frac{\psi}{2}) \sin(\frac{\theta}{2}) \cos(\frac{\varphi}{2}) & + \sin(\frac{\psi}{2}) \cos(\frac{\theta}{2}) \sin(\frac{\varphi}{2}) \\ \sin(\frac{\psi}{2}) \cos(\frac{\theta}{2}) \cos(\frac{\varphi}{2}) & - \cos(\frac{\psi}{2}) \sin(\frac{\theta}{2}) \sin(\frac{\varphi}{2}) \end{bmatrix} \quad (2.25)$$

On peut aussi choisir le quaternion de signe opposé, les rotations associées étant identiques.

Réciproquement, on obtient les angles d'Euler à partir du quaternion via :

$$\begin{aligned} \psi &= \arctan(2(sz + xy), s^2 + x^2 - y^2 - z^2) \\ \theta &= \arcsin(2(sy - xz)) \\ \varphi &= \arctan(2(sx + yz), s^2 + z^2 - x^2 - y^2) \end{aligned} \quad (2.26)$$

Quand θ est voisin de $\pm\pi/2$ cette formule n'est pas utilisable puisque dans ce cas les angles ψ et φ sont indéterminés. Prenons $\theta = \varepsilon\pi/2$ où $\varepsilon = \pm$. Le quaternion q devient :

$$q = \frac{\sqrt{2}}{2} \begin{bmatrix} \cos(\frac{\psi-\varepsilon\varphi}{2}) \\ -\varepsilon \sin(\frac{\psi-\varepsilon\varphi}{2}) \\ \varepsilon \cos(\frac{\psi-\varepsilon\varphi}{2}) \\ \sin(\frac{\psi-\varepsilon\varphi}{2}) \end{bmatrix} \quad (2.27)$$

La seule information que l'on peut extraire de la matrice q est donc la combinaison linéaire $D \triangleq \psi - \varepsilon\varphi$. Cela se fera par exemple en notant que :

$$\begin{aligned} z - \varepsilon x &= \sqrt{2} \sin(\frac{\psi-\varepsilon\varphi}{2}) \\ s + \varepsilon y &= \sqrt{2} \cos(\frac{\psi-\varepsilon\varphi}{2}) \end{aligned}$$

d'où

$$D = 2 \arctan(z - \varepsilon x, s + \varepsilon y) \quad (2.28)$$

Introduisons $S \triangleq \psi + \varepsilon\varphi$. et notons \hat{S} la valeur précédente de S (dans une procédure itérative). On calculera par continuité :

$$\begin{aligned} \psi &= (\hat{S} + D)/2 \\ \varphi &= \varepsilon(\hat{S} - D)/2 \end{aligned} \quad (2.29)$$

2.1.3.4 Quaternion et représentation axiale des rotation

Quelques cas particuliers sont intéressants à examiner ici.

La rotation de roulis $\mathcal{R}_X(\varphi)$ correspond à la rotation axiale $\mathcal{R}(\varphi, u)$ où u est l'axe des X . On a donc : $s = \cos(\varphi/2)$ et $v = \sin(\varphi/2) u$, soit :

$$R = \mathcal{R}_X(\varphi) \iff q = \begin{bmatrix} \cos(\varphi/2) \\ \sin(\varphi/2) \\ 0 \\ 0 \end{bmatrix}$$

De même on vérifie facilement que :

$$R = \mathcal{R}_Y(\theta) \iff q = \begin{bmatrix} \cos(\theta/2) \\ 0 \\ \sin(\theta/2) \\ 0 \end{bmatrix}$$

et

$$R = \mathcal{R}_Z(\psi) \iff q = \begin{bmatrix} \cos(\psi/2) \\ 0 \\ 0 \\ \sin(\psi/2) \end{bmatrix}$$

2.1.3.5 Formulation matricielle des quaternions

La représentation matricielle des quaternions est définie par :

$$Q^\times = A(q) = \begin{bmatrix} s & -x & -y & -z \\ x & s & -z & y \\ y & z & s & -x \\ z & -y & x & s \end{bmatrix} \quad (2.30)$$

Elle encode une représentation de l'attitude très redondante puisque l'on a 16 paramètres, mais avec des relations entre ces paramètres qui sont très simples (contrairement à la représentation par matrice de rotation). L'opérateur inverse de A est encore noté vex : $q = vex(Q_\times)$.

On notera quelques définitions et propriétés intéressantes de Q^\times :

$$\begin{aligned} Q_1^\times Q_2^\times &= Q^\times && \text{où } Q \text{ représente } q_1 \otimes q_2 \\ Q^{\times T} &= Q^{*\times} && \text{où } Q^* \text{ est la représentation matricielle de } q^* \\ Q^{\times -1} &= Q^{*\times} / q^T q && \text{car } Q^{\times T} Q^\times = q^T q I \\ Q^\times &= s I + \begin{bmatrix} 0 & -v^T \\ v & v_\times \end{bmatrix} && \text{décomposition de } Q^\times \end{aligned}$$

Pour représenter l'attitude on peut utiliser un quaternion matriciel unitaire (i.e. avec la contrainte $q^T q = 1$). Alors la composition des rotations

$R_1 R_2$ est représentée par le produit matriciel $Q_1^\times Q_2^\times$, et la rotation inverse R^{-1} est représentée par l'inversion matricielle $Q^{\times -1} = Q^{\times T} = Q^{*\times}$.

2.2 Cinématique

La cinématique se décompose en cinématique de translation et cinématique de rotation. La cinématique de translation décrit les mouvement du centre de gravité de l'engin. Paramétrée par la position X (et la vitesse de translation V) exprimée dans le repère fixe \mathcal{R}_0 , la cinématique de translation est gouvernée par une équation dynamique linéaire, donc simple à analyser.

$$X = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad V = \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \quad (2.31)$$

La cinématique de rotation décrit les mouvements de rotation de l'engin autour de son centre de gravité. Quelque soit le paramétrage de l'attitude choisi, cette cinématique est toujours gouvernée par une équation dynamique non linéaire.

Dans la suite on utilise essentiellement la représentation de l'attitude par matrice de rotation R ou par quaternion q . La vitesse de rotation ω est souvent exprimée par ses composantes (p, q, r) dans le repère mobile \mathcal{R}_m . On utilisera aussi parfois une représentation de cette vitesse par un quaternion pur non unitaire Ω .

$$\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad \Omega = \begin{bmatrix} 0 \\ \omega \end{bmatrix} \quad (2.32)$$

2.2.1 Equations de la cinématique

2.2.1.1 Cinématique de translation

La cinématique de translation est représentée dans le repère fixe par :

$$\begin{cases} \dot{X} = V \\ \dot{V} = A + G \end{cases} \quad \text{où} \quad \begin{array}{ll} X & : \text{ est la position} \\ V & : \text{ est la vitesse} \\ A & : \text{ est l'accélération spécifique} \\ G & : \text{ est l'accélération gravitationnelle} \end{array} \quad (2.33)$$

2.2.1.2 Cinématique de rotation avec R

Avec la représentation par matrice de rotation, la cinématique s'écrit :

$$\dot{R} = R \omega_{\times} \quad (2.34)$$

où ω_{\times} est la matrice antisymétrique 3×3 construite à partir du vecteur rotation ω :

$$\omega_{\times} = S(\omega) \triangleq \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \quad (2.35)$$

Rappelons que pour tous vecteurs u et v , on a $u_{\times} v = u \times v$.

On notera que la vitesse de rotation s'obtient analytiquement à partir de R et \dot{R} par :

$$\omega_{\times} = R^T \dot{R} \implies \omega = \text{vex}(R^T \dot{R}) \quad (2.36)$$

2.2.1.3 Cinématique de rotation avec q

Quand l'attitude (par rapport à un repère fixe) est représentée par un quaternion vectoriel unitaire q , la cinématique en rotation s'écrit :

$$\dot{q} = \frac{1}{2} q \otimes \Omega \quad (2.37)$$

où Ω est le quaternion pur, non unitaire, construit à partir de ω .

L'extraction de la vitesse de rotation à partir de q et \dot{q} est immédiate. Il suffit pour cela en effet de multiplier à gauche par le quaternion conjugué. Rappelons en effet que $q^{-1} = q^*/q^T q$, et que donc pour un quaternion unitaire $q^{-1} = q^*$.

$$\Omega = 2 q^* \otimes \dot{q} = 2 \begin{pmatrix} s \\ -v \end{pmatrix} \otimes \begin{pmatrix} \dot{s} \\ \dot{v} \end{pmatrix} = 2 \begin{pmatrix} \dot{s} s + \dot{v}^T v \\ s \dot{v} - \dot{s} v - v \times \dot{v} \end{pmatrix}$$

L'égalité pour la première composante exprime le fait que $0 = d/dt(q^T q)$. Les autres composantes s'écrivent :

$$\omega = s \dot{v} - \dot{s} v - v \times \dot{v} \quad (2.38)$$

On peut bien entendu aussi utiliser la représentation matricielle unitaire des quaternions pour décrire la cinématique. Parce que q^{\times} est linéaire en les composantes de q on peut intervertir la dérivation et le passage en matriciel. L'expression de la cinématique ci-dessus s'écrit donc encore :

$$\dot{q}^{\times} = \frac{1}{2} q^{\times} \Omega^{\times} \quad (2.39)$$

Dans ce cas la vitesse de rotation s'extrait par :

$$\Omega^\times = 2 q^{*\times} \dot{q}^\times \quad (2.40)$$

Cette formulation est évidemment très redondante.

2.2.2 Schéma d'intégration numérique

2.2.2.1 Cinématique de translation

La cinématique de translation est représentée dans le repère fixe par :

$$\begin{cases} \dot{X} = V \\ \dot{V} = A + G \end{cases} \quad \text{où} \quad \begin{array}{ll} X & : \text{ est la position} \\ V & : \text{ est la vitesse} \\ A & : \text{ est l'accélération spécifique} \\ G & : \text{ est l'accélération gravitationnelle} \end{array} \quad (2.41)$$

L'intégration numérique de cette équation sur un horizon Δt , en supposant l'accélération $A + G$ constante sur cet intervalle, donne :

$$\begin{cases} X_{k+1} = X_k + \Delta t V_k + \frac{\Delta t^2}{2} (A_k + G_k) \\ V_{k+1} = V_k + \Delta t (A_k + G_k) \end{cases}$$

On programmera ces équations sous une forme évitant la formation de Δt^2 :

$$\begin{cases} V_{k+1} = V_k + \Delta t (A_k + G_k) \\ X_{k+1} = X_k + \frac{\Delta t}{2} (V_k + V_{k+1}) \end{cases} \quad (2.42)$$

2.2.2.2 Cinématique de rotation

L'intégration de la cinématique en rotation est plus délicate car l'équation est non linéaire quelque soit la représentation choisie pour l'attitude. Il faut de plus traiter les éventuelles singularités et contraintes. On utilise ici la représentation par quaternion qui présente une singularité peu gênante (q et $-q$ correspondent à la même attitude) et une seule contrainte ($q^T q = 1$).

Rappelons qu'avec les quaternions la cinématique de rotation est représentée par :

$$\dot{q} = \frac{1}{2} q \otimes \Omega \quad (2.43)$$

Cette équation est linéaire en q . En faisant l'hypothèse que la vitesse de rotation ω est constante sur l'horizon d'intégration Δt on peut l'intégrer par les techniques classiques (Euler, Runge-Kutta). Néanmoins toutes ces

techniques d'intégration calculent une correction additive δq de sorte que $q_{k+1} = q_k + \delta q$. Il faut ensuite re-normaliser le quaternion, et la qualité de l'intégration n'est donc pas garantie.

Des techniques spécifiques ont été développées pour améliorer la précision.

- La première consiste à utiliser une correction multiplicative au lieu d'une correction additive. On n'écrit pas $q_{k+1} = q_k + \delta q$, mais :

$$q_{k+1} = q_k \otimes \delta q$$

où δq est la correction. De cette manière, si l'incrément δq est, comme q_k , un quaternion unitaire, alors q_{k+1} en est aussi un. Le caractère unitaire est automatiquement conservé lors de la correction. Cette méthode d'intégration permet de résoudre exactement l'équation de la cinématique quand ω est constant.

Quand ω n'est plus constant, on peut faire l'hypothèse d'une variation linéaire sur l'intervalle d'intégration. Ceci équivaut à utiliser une valeur moyenne $\omega = (\omega_{k+1} + \omega_k)/2$ pour calculer δq .

- La seconde amélioration consiste à approximer la variation de ω sur l'intervalle d'intégration par un polynôme d'ordre supérieur dans le but d'améliorer la précision du calcul de q .

La solution d'Edwards utilise une approximation polynômiale d'ordre 2 des variations de ω . Ses coefficients sont ajustés pour obtenir une précision sur le calcul de q équivalente à un développement de Taylor au troisième ordre.

Elle présente aussi l'avantage d'un surcoût calculatoire limité, et d'être bien adaptée à la technologie des gyromètres habituels. La correction d'Edwards utilise l'intégration de ω dans le repère mobile sur deux horizons consécutifs. Posons :

$$\delta A_{k+1} = \int_{t_k}^{t_{k+1}} \omega(\tau) d\tau \quad (2.44)$$

Cette intégrale est, à un coefficient Δt près, la vitesse de rotation délivrée par les gyromètres les plus courants (gyromètres intégrateurs-dérivateurs). On calcule ensuite :

$$\delta A = \delta A_{k+1} + \frac{1}{12} \delta A_k \times \delta A_{k+1} \quad (2.45)$$

Et l'incrément à utiliser est alors donné par :

$$\delta q = \begin{bmatrix} \cos(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) u \end{bmatrix} \quad \text{où} \quad \begin{cases} \theta &= ||\delta A|| \\ u &= \delta A / ||\delta A|| \end{cases} \quad (2.46)$$

2.2.2.3 Interpolation d'attitudes

Dans le cadre du filtrage pour la navigation deux opérations sont essentielles : la prédiction et la correction. L'opération de prédiction utilise l'intégration numérique. Celle de correction utilise la fusion d'information. Cette fusion peut aussi être vue parfois comme un problème d'interpolation. C'est la raison pour laquelle on aborde ici le problème de l'interpolation d'attitudes.

De nombreuses techniques spécifiques ont été développées pour interpoler des poses (attitude et translation) en particulier dans le domaine des jeux vidéos. On examine ici la plus classique d'entre-elles : l'interpolation linéaire sphérique (Slerp) [3]. Elle interpole des quaternions unitaires, et a été à la base de la majorité des outils dans ce domaine.

Il y a plusieurs manières d'introduire la Slerp. La plus directe utilise l'exponentiation des quaternions qui est définie par :

$$q = \begin{bmatrix} \cos(\frac{\alpha}{2}) \\ \sin(\frac{\alpha}{2}) u \end{bmatrix} \implies q^t = \begin{bmatrix} \cos(t \frac{\alpha}{2}) \\ \sin(t \frac{\alpha}{2}) u \end{bmatrix} \quad \forall t$$

Avec cette définition, on construit une interpolation entre deux quaternions q_1 et q_2 en posant de manière évidente :

$$q(t) = q_2^t \otimes q_1^{1-t} \quad \forall t \in [0, 1]$$

On peut montrer que cette formulation est équivalent à :

$$q(t) = \frac{\sin[(1-t)\theta]}{\sin\theta} q_1 + \frac{\sin[t\theta]}{\sin\theta} q_2 \quad \text{où} \quad \begin{cases} t \in [0, 1] \\ \theta = \arccos(q_1^T q_2) \end{cases}$$

Enfin une troisième formulation, encore équivalente aux précédentes, permet d'éviter l'apparente singularité en $\theta = 0$. C'est elle que nous utiliserons dans nos macros de calcul :

$$q(t) = \cos(t\theta) q_1 + \sin(t\theta) q_3 \quad \text{où} \quad \begin{cases} t \in [0, 1] \\ \theta = \arccos(q_1^T q_2) \\ q_3 = \tilde{q}_3 / \|\tilde{q}_3\| \\ \tilde{q}_3 = q_2 - (q_1^T q_2) q_1 \end{cases} \quad (2.47)$$

Cette dernière formulation montre aussi que dans le plan défini par (q_1, q_2) la Slerp forme une paire orthonormée (q_1, q_3) puis effectue une rotation dans ce plan d'angle proportionnel au coefficient d'interpolation t . On obtient toujours un quaternion unitaire grâce à l'orthonormalisation préalable de la base.

2.3 Validation numérique

Les transformations permettant le passage entre les diverses représentations de l'attitude ont été programmées sous la forme de fonction Matlab amenant ou partant de la matrice de rotation R . Les autres transformations s'en déduisent par composition. Les syntaxes d'appel de ces fonctions sont listées en 2.3.1. Leur fonctionnement a été validé par quelques tests élémentaires incluant les points singuliers et leurs voisinages. Il est important d'avoir en tête que dans ces zones, un effet mémoire est présent puisque les singularités des calculs sont levées en imposant la continuité de certaines grandeurs par rapport au calcul précédent.

La validation des schémas d'intégration numérique est un peu plus complexe. Elle demande la connaissance des trajectoires réelles dans l'espace d'état pour servir de référence. Nous montrons comment générer de telles trajectoires. En intégrant les accélérations de translation et les vitesses de rotation correspondantes, on peut alors évaluer la précision des schémas numériques. On a vérifié que :

- En translation, les erreurs sur la position et la vitesse sont inférieures à 0.05 m et 0.01 m/s pour des déplacements de quelques décimètres, avec des vitesses limitées à 5 m/s, et des accélérations inférieures au g .
- En rotation, les erreurs sur les angles d'Euler restent inférieures à 0.1°.

2.3.1 Formules de passage entre les représentations de l'attitude

2.3.1.1 Passage (ψ, θ, φ) à R

La macro permettant de calculer R à partir des angles d'Euler est :

```
function [M]=eul2chr(eul)
```

```
% [M] = eul2chr(eul)
```

```
%-----
```

```
% Passage des 3 angles d'Euler (convention aeronautique)
```

```
% à la matrice de changement de repere (rotation)
```

```
%-----
```

```
% eul : vecteur des angles d'Euler [psi;theta;phi]
```

```
% M      : matrice de rotation Rm/Ro = Matrice de passage de Ro à Rm
```

%-----

2.3.1.2 Passage de R à (ψ, θ, φ)

La macro permettant de calculer les angles d'Euler à partir de R est :

```
function [eul]=chr2eul(M)
```

```
% [eul] = chr2eul(M);
```

%-----

```
% Passage de la matrice de changement de repere (rotation)
```

```
% aux 3 angles d'Euler (convention aeronautique)
```

%-----

```
% M      : matrice de rotation Rm/Ro = Matrice de passage de Ro à Rm
```

```
% eul : vecteur des angles d'Euler [psi;theta;phi]
```

%-----

```
% ATTENTION:
```

```
% Si l'angle de tangage (theta) est proche de +-pi/2 on utilise la
```

```
% continuité par rapport à la valeur précédente.
```

%-----

2.3.1.3 Passage de (α, u) à R

La macro permettant de calculer R à partir de α et u est :

```
function [M] = ang2chr(alpha,u)
```

```
% [M] = ang2chr(alpha,u);
```

%-----

```
% Calcul la matrice de passage entre deux reperes
```

```
% a partir de l'angle et de la direction
```

%-----

```
% alpha : angle de la rotation
```

```
% u      : vecteur unitaire de l'axe de rotation
```

```
% M      : matrice de passage
```

%-----

```
% ATTENTION : u doit être unitaire
```

%-----

2.3.1.4 Passage de R à (α, u)

La macro permettant de calculer α et u à partir de R est :

```
function [alpha,u]=chr2ang(M)
```

```
% [alpha,u] = chr2ang(M);
```

```
%-----
```

```
% Calcul l'angle et la direction d'une rotation
```

```
% a partir de la matrice de passage entre deux reperes
```

```
%-----
```

```
% M      : matrice de passage
```

```
% alpha : angle de la rotation
```

```
% u      : vecteur unitaire de l'axe de rotation
```

```
%-----
```

```
% ATTENTION : M doit être une matrice de rotation
```

```
%-----
```

2.3.1.5 Passage de q à R

La macro permettant de calculer R à partir du quaternion q est :

```
function [M] = qua2chr(q)
```

```
% [M] = qua2chr(q);
```

```
%-----
```

```
% Calcul la matrice de passage entre deux reperes a partir du quaternion
```

```
%-----
```

```
% q : quaternion unitaire
```

```
% M : matrice de passage (rotation)
```

```
%-----
```

```
% ATTENTION : q doit etre unitaire
```

```
%-----
```

2.3.1.6 Passage de R à q

La macro permettant de calculer q à partir de R est :

```
function [q] = chr2qua(M)
```

```
% [q] = chr2qua(M);
```

```
%-----
```

```
% Calcul le quaternion associé à une matrice de passage entre deux reperes
```

```
%-----
```

```
% M : matrice de passage (rotation)
```

```
% q : quaternion unitaire
```

```
%-----
```

```
% ATTENTION : M doit etre une matrice de rotation
```

```
%-----
```

2.3.2 Schémas d'intégration

Pour valider les schémas d'intégration numérique, la difficulté est de générer des signaux cohérents : position, vitesse et accélération cohérentes en translation, et attitude et vitesse cohérentes en rotation. Par cohérence on entend par exemple que la dérivée de la position soit exactement égale à la vitesse (indépendamment de toute technique d'intégration numérique). Pour obtenir ces grandeurs cohérentes on a utilisé une expression analytique de la position dont la dérivée est elle aussi connue analytiquement.

Ensuite en intégrant numériquement les accélérations de translation et les vitesses de rotation on examine les erreurs commises par les schémas numériques.

2.3.2.1 Référence des mouvements

En translation tout est linéaire. On choisit une position évoluant de manière sinusoïdale. Il en est de même alors pour la vitesse et l'accélération. On dimensionne l'amplitude du mouvement en position pour rester dans des bornes maximales choisies pour la position, la vitesse et l'accélération.

En rotation la cinématique est non linéaire. Mais on a montré que pour calculer la vitesse de rotation analytiquement, il suffit de connaître le quaternion et sa dérivée. Rappelons l'équation (2.38) obtenue pour un quaternion vectoriel unitaire :

$$\omega = s \dot{v} - \dot{s} v + v \times \dot{v}$$

En exprimant les composantes de q sous forme analytique sinusoïdale, on obtiendra aussi une forme analytique de ω . La seule difficulté est de satisfaire la contrainte de normalité du quaternion. Pour cela on peut prendre par exemple :

$$q = \begin{pmatrix} s \\ x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos \theta_1 \\ \sin \theta_1 \cos \theta_2 \\ \sin \theta_1 \sin \theta_2 \cos \theta_3 \\ \sin \theta_1 \sin \theta_2 \sin \theta_3 \end{pmatrix} \quad (2.48)$$

On a choisi des angles θ_i variant linéairement en fonction du temps :

$$\dot{\theta}_i = \omega_i \quad (2.49)$$

Les trajectoires de référence ainsi générées permettent de vérifier la justesse des algorithmes développés pour réaliser l'intégration numérique de la cinématique.

2.3.2.2 Validation sur quelques trajectoires particulières

Pour une trajectoire en roulis seul (tonneau) on choisit $\psi = 0$ et $\theta = 0$. Le quaternion d'attitude s'écrit alors sous la forme (voir 2.25) :

$$q = \begin{bmatrix} \cos(\varphi/2) \\ \sin(\varphi/2) \\ 0 \\ 0 \end{bmatrix}$$

On génère donc une trajectoire du type tonneau en choisissant : $\omega_1 \neq 0$, $\theta_2 = 0$, et $\theta_3 = 0$. La figure 2.1 montre que l'erreur d'attitude reste nulle. En fait on a :

```
Erreur en Attitude (psi [°]) : 0
Erreur en Attitude (theta [°]) : 0
Erreur en Attitude (phi [°]) : 1.2468e-012
```

Pour une trajectoire en tangage seul (looping) on choisit $\psi = 0$ et $\varphi = 0$. Le quaternion d'attitude s'écrit alors sous la forme (voir 2.25) :

$$q = \begin{bmatrix} \cos(\theta/2) \\ 0 \\ \sin(\theta/2) \\ 0 \end{bmatrix}$$

On génère donc une trajectoire du type looping en choisissant : $\omega_1 \neq 0$, $\theta_2 = \pi/2$, et $\theta_3 = 0$. La figure 2.2 montre que l'erreur d'attitude reste quasiment nulle (à 2π près).

```
Erreur en Attitude (psi [°]) : 1.7945e-023
Erreur en Attitude (theta [°]) : 1.0928e-011
Erreur en Attitude (phi [°]) : 0
```

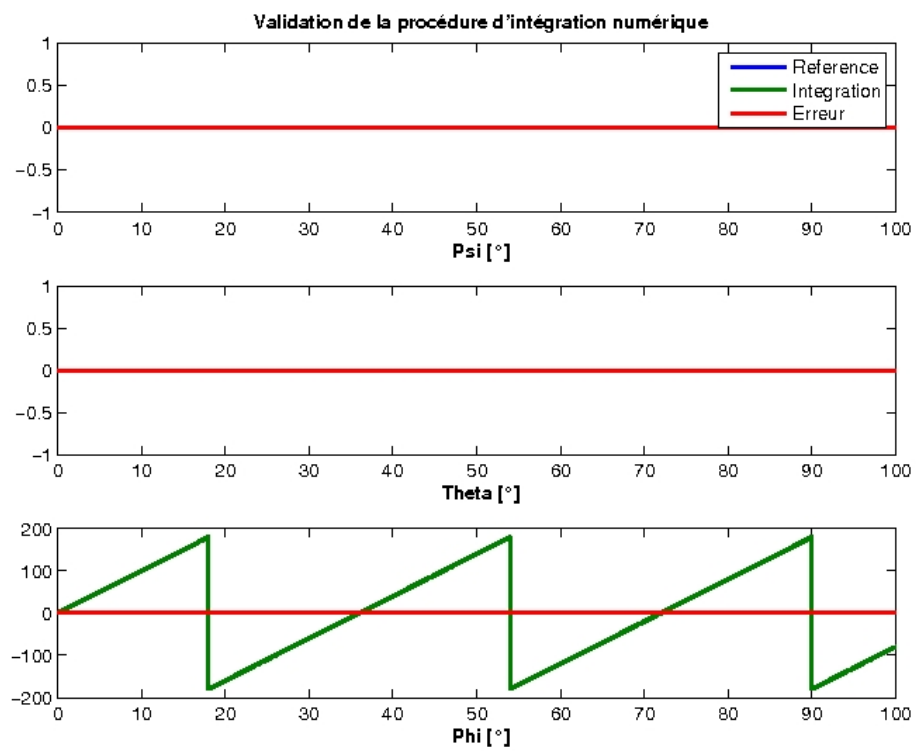


FIGURE 2.1 – Validation procédure d'intégration : Tonneau

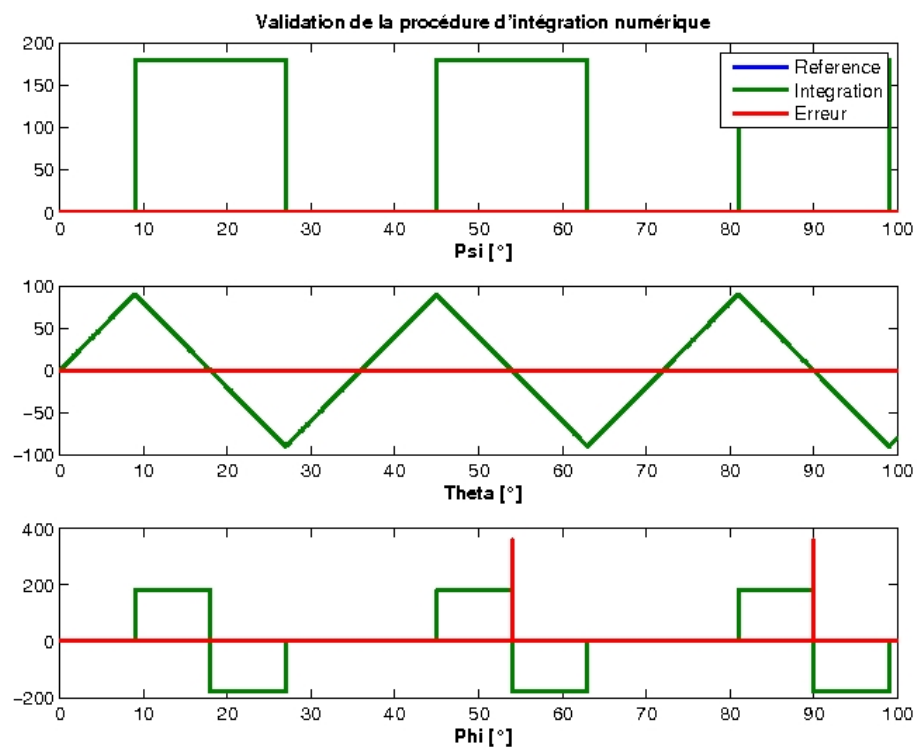


FIGURE 2.2 – Validation procédure d'intégration : Looping

Pour une trajectoire en lacet seul (virage à plat) on choisit $\theta = 0$ et $\varphi = 0$. Le quaternion d'attitude s'écrit alors sous la forme (voir 2.25) :

$$q = \begin{bmatrix} \cos(\psi/2) \\ 0 \\ 0 \\ \sin(\psi/2) \end{bmatrix}$$

On génère donc une trajectoire du type virage à plat en choisissant : $\omega_1 \neq 0$, $\theta_2 = \pi/2$, et $\theta_3 = \pi/2$. La figure 2.3 montre que l'erreur d'attitude reste quasiment nulle.

```

Erreur en Attitude (psi [°])    : 1.2468e-012
Erreur en Attitude (theta [°])  : 1.9492e-028
Erreur en Attitude (phi [°])    : 1.8503e-028

```

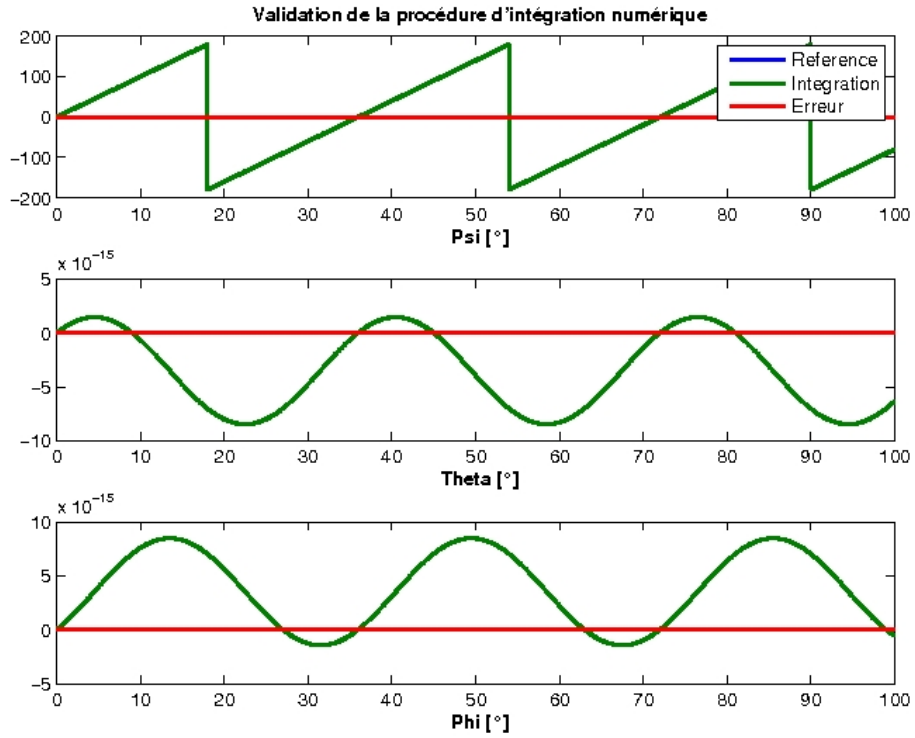


FIGURE 2.3 – Validation procédure d'intégration : Virage à plat

2.3.2.3 Validation sur trajectoires complexes

Les figures 2.4 à 2.7 montrent les résultats obtenus avec :

- en translation, des pulsations assurant des déplacements limités à 10, 20 et 30 m (sur les axes x , y et z respectivement), des vitesses limitées à 5, 5 et 1 m/s, et des accélérations limitées à 1 g sur les trois axes,
- en rotation, des pulsations de 10, 20 et 30 °/s.

Les erreurs sur la position et la vitesse de translation sont négligeables. L'erreur d'attitude (mesurée par l'angle de la rotation amenant de l'attitude intégrée à l'attitude vraie) ne dépasse pas 0.15 °. Sur les angles d'Euler elles restent en deçà de 0.1 °. Le pic d'erreur apparaissant sur l'angle de roulis correspond à un passage de -180 ° à $+180$ °. Sur les positions l'erreur est au plus de 0.05 m, et sur les vitesses de 0.01 m/s.

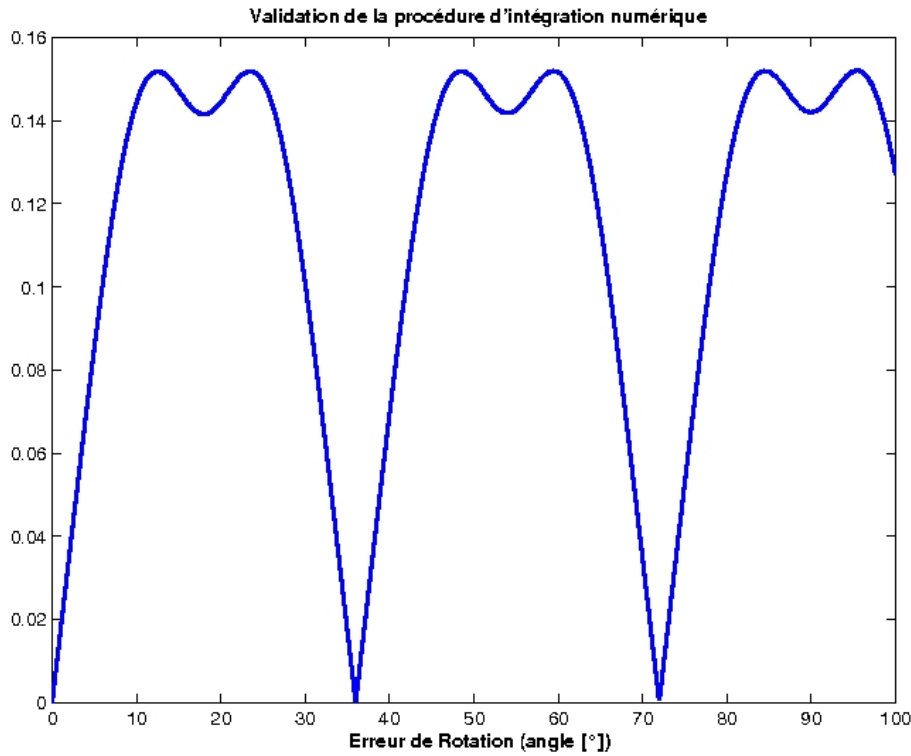


FIGURE 2.4 – Validation procédure d'intégration : 1

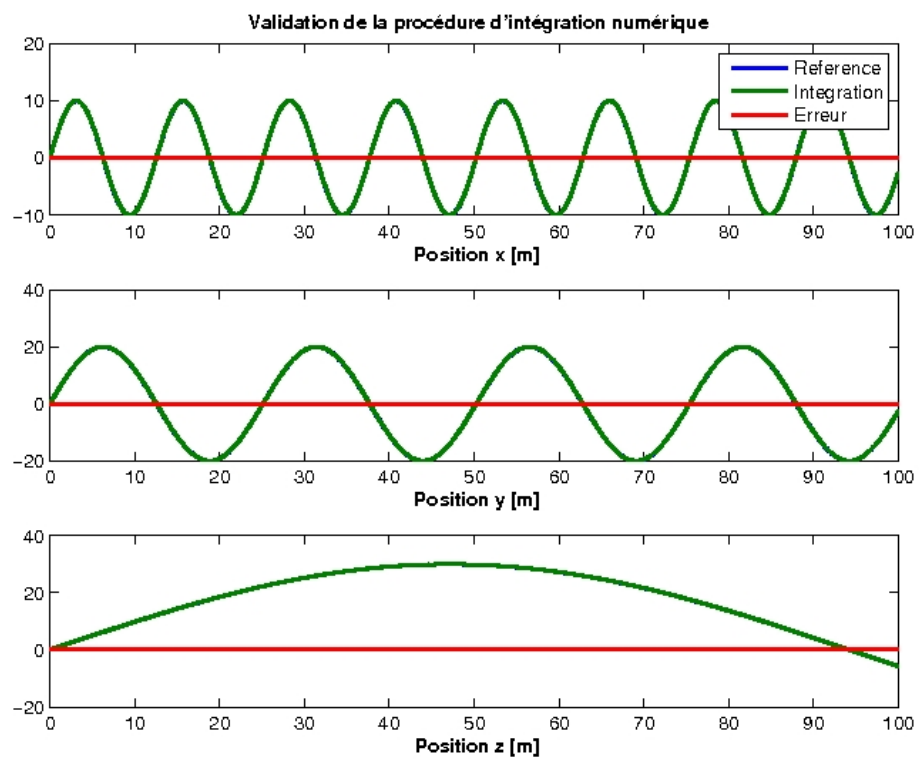


FIGURE 2.5 – Validation procédure d'intégration : 2

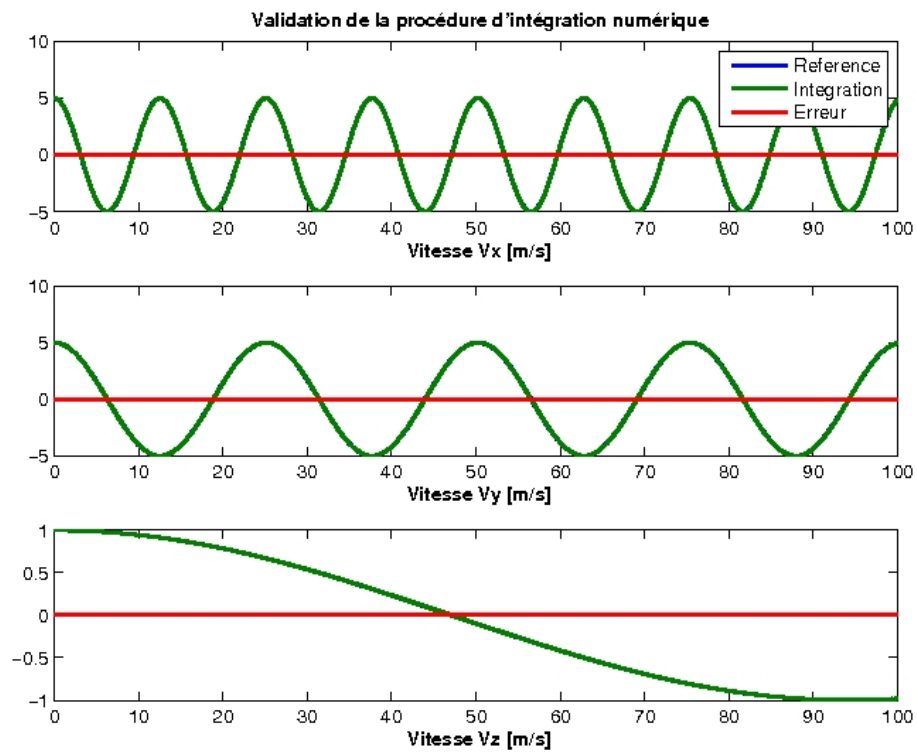


FIGURE 2.6 – Validation procédure d'intégration : 3

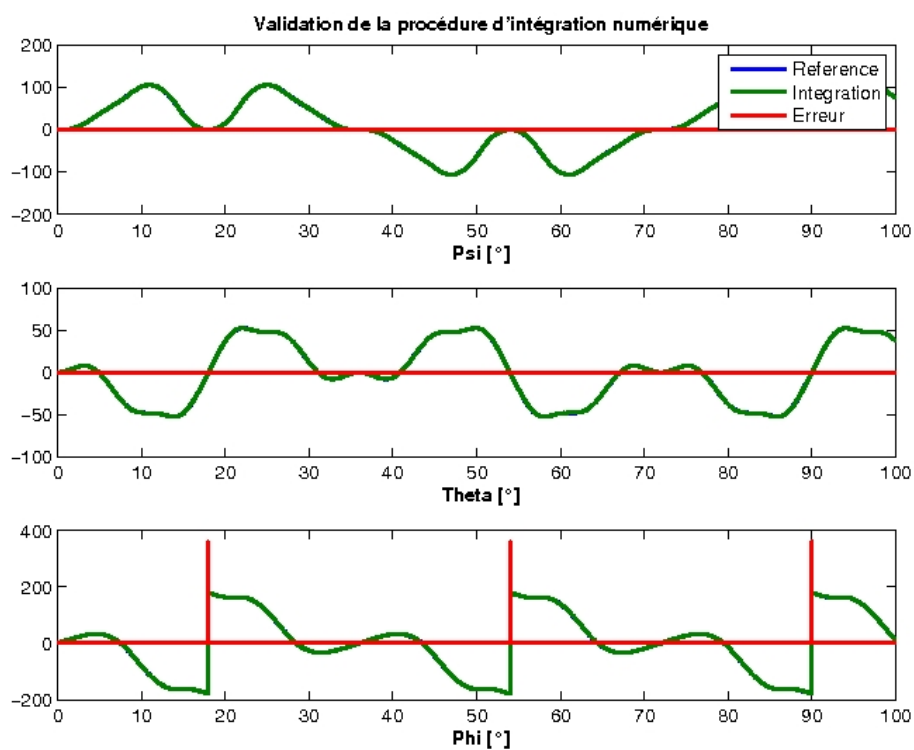


FIGURE 2.7 – Validation procédure d'intégration : 4

Chapitre 3

Capteurs

3.1 Introduction

Différents types de capteurs peuvent être utilisés pour la navigation d'un engin. Les plus courants sont : accéléromètre, gyromètre, magnétomètre, inclinomètre, baromètre, récepteur GPS, caméra, LIDAR, odomètre. Diverses classifications des capteurs peuvent être faites, en particulier selon les types de mesure délivrées, selon les technologies employées pour fabriquer ces mesures, ... On propose ici une classification selon la complexité des traitements à mettre en œuvre avant leur intégration dans un algorithme de navigation :

- Certains de ces capteurs fournissent directement une grandeur utilisable dans les algorithmes de navigation. Ainsi un accéléromètre mesure l'accélération spécifique A qui apparaît comme une entrée dans le modèle de cinématique en translation. Son intégration donne une vitesse, puis une position. De même un gyromètre mesure la vitesse de rotation ω qui apparaît comme une entrée dans le modèle de la cinématique de rotation. En l'intégrant on obtiendra une attitude.
- D'autres capteurs demandent une certaine mise en forme des mesures pour être utilisables en navigation. Ainsi la pression atmosphérique mesurée par un baromètre doit être convertie en une mesure d'altitude (souvent appelée altitude-pression) par une transformation non linéaire qui utilise un modèle d'atmosphère. Dans le même ordre d'idée, un odomètre est généralement constitué d'un compteur qui mesure la rotation des roues d'un robot. Connaissant la taille des roues, leurs positions relatives, on peut en déduire la distance parcourue, et le cap du robot. Par dérivation numérique, une vitesse peut aussi être calculée (vitesse de translation et vitesse de rotation).

- Enfin d'autres capteurs délivrent des informations tellement riches qu'il doivent impérativement être associés à un algorithme de traitement pour en extraire des grandeurs aisément intégrables dans un algorithme de navigation. On pense ici par exemple aux séquences vidéos délivrées par une caméra. Avec un algorithme de calcul de flot optique (qui est un algorithme de complexité limitée) on obtient une mesure des vitesses relatives (en translation et en rotation) par rapport à la scène, ramenée à la distance à cette scène. Avec un algorithme d'extraction de pose (plus complexe), on obtient dans certains cas directement une mesure de la position et l'attitude relative par rapport à la scène observée.

Un travail essentiel à conduire pour intégrer le plus efficacement possible les capteurs dans un algorithme de navigation est la modélisation des erreurs de mesure : facteurs d'échelle, biais, dérives, courbes caractéristiques, ... On commence en général par déterminer celles qui sont constantes (et que l'on pourra donc en général compenser à l'initialisation du filtre), et par caractériser la dynamique des autres.

A noter que certains biais peuvent évoluer assez vite lors de la montée en température de l'électronique de mesure. On déterminera alors ce temps de chauffe, et on spécifiera simplement dans le mode opératoire la nécessité d'attendre cette durée après la mise sous tension et avant la mise en marche du filtre de navigation.

On aura toujours intérêt à effectuer une analyse bibliographique. Elle apporte par exemple des informations sur :

- Les caractéristiques constructeur des capteurs.
- La modélisation des erreurs en mode de fonctionnement nominal.
- Les modes de défaillance et leurs caractéristiques.
- L'utilisation de la variance de Allan comme outil de quantification de certaines erreurs.
- Les algorithmes de pré-traitement, avant intégration des mesures dans l'algorithme de navigation.

D'un point de vue de la représentation informatique des mesures, il est souvent intéressant de considérer que chaque capteur donne, en plus de la valeur de la mesure, le domaine de confiance associé (matrice de covariance). De cette manière la fusion d'informations dont la précision évolue au cours du temps (comme par exemple une mesure de position GPS, ou de pression barométrique) ne se distinguera pas de celle d'informations dont la précision est constante (comme par exemple une mesure de vitesse de rotation). Le traitement sera alors indépendant des éventuelles instationnarités des ca-

ractéristiques statistiques de la mesure. On gagnera donc en modularité des codes développés.

Pour terminer cette introduction, quelques points particuliers :

- Accéléromètre : Le modèle de mesure est $A_m = A + b_A + bruit$, avec $\dot{b}_A = bruit$. Comparer les covariances des bruits estimées par l'analyse de variance de Allan, avec les données constructeur.
- Gyromètre : vérifier si le bruit est additif ou multiplicatif. Modèle $\Omega_m = \Omega + b_\Omega + bruit$ ou bien $\Omega_m = \Omega (1 + bruit) + b_\Omega$. Avec $\dot{b}_\Omega = bruit$.
- Magnétomètre : faire des essais sur table pour relever la caractéristique statique et son éventuelle dépendance en fonction de l'attitude.
- Inclinomètre : mesure passe-bas de l'attitude apparente. Vérifier sa dépendance par rapport aux accélérations transverses. Vérifier sa calibration aux grands angles.

3.2 Magnétomètre

En navigation les magnétomètres sont utilisés pour mesurer le champ magnétique terrestre. De ces trois composantes on déduit en particulier la direction du nord magnétique, mais pas seulement. On peut aussi théoriquement en déduire une information de latitude. C'est un capteur quasiment indispensable à la navigation de nombreux engins.

Mais tout ceci est malheureusement assez difficile à utiliser en pratique car cette mesure est très perturbée. Le champ magnétique entourant un engin est en effet souvent brouillé par les moteurs, les pièces tournantes, les parties métalliques, etc. De plus le champ magnétique terrestre fluctue lui aussi localement en fonction de la position et dans le temps. Tout ceci fait que le développement d'un magnétomètre (et de l'algorithme de pré-traitement associée) reste un enjeu fort et toujours d'actualité.

3.2.1 Caractéristiques du champ magnétique terrestre

Le champ magnétique terrestre peut être approximé (à 90%) par un champ dit géomatique, qui serait créé par aimant placé au centre de la terre et dont l'axe, passant par les deux pôles magnétiques, fait un angle de $11,5^\circ$ par rapport à l'axe de rotation de la Terre. Une mesure faite l'été 2010 a permis d'estimer que le pôle Nord magnétique était à environ 550 km du pôle Nord géographique. Il se déplace à la vitesse moyenne de 55km/an (soit

une moyenne d'environ 150m/jour ou 6m/h). Mais la position du pôle magnétique varie aussi au cours de la journée, se déplaçant ainsi de plusieurs dizaines de km autour de sa position moyenne [1]. Position des pôles magnétiques (en 2010) Pôle Nord : 85° 00 ? N, 132° 36 ? O Pôle Sud : 64° 24 ? S, 137° 18 ? E

La figure ci-dessous montre l'orientation classique du champ magnétique dans l'hémisphère Nord, et introduit les diverses grandeurs utilisées pour le caractériser. Ces sept grandeurs sont reliées entre elles par quelques expres-

sions simples :	Déclinaison (D)	$D = \tan^{-1}(Y/X)$
	Inclinaison (I)	$I = \tan^{-1}(Z/H)$
	Composante Horizontale (H)	$H = \sqrt{X^2 + Y^2}$
	Composante Nord (X)	$X = H \cos D$
	Composante Est (Y)	$Y = H \sin D$
	Intensité (F)	$F = \sqrt{X^2 + Y^2 + Z^2}$

3.2.2 Variations spatiales du champ magnétique

Le champ magnétique terrestre dépend de la latitude, de la longitude et de l'altitude. La figure ci-dessous montre ses variations spatiales relevées en 1999 par le satellite Oersted. Une cartographie encore plus précise sera réalisée par la mission Swarm du programme Earth Explorer de l'Agence spatiale européenne (ESA), qui a pour objectif principal d'étudier les variations spatiales et temporelles du champ géomagnétique, ainsi que l'environnement ionosphérique de la Terre.

Au niveau de l'équateur, l'intensité du champ magnétique terrestre varie de 24 à 45 μT (0,24 à 0,45 G) et au niveau des poles elle varie de 62 à 70 μT (0,62 à 0,7 G). Toutes ses caractéristiques peuvent être prédites en tout point du géoïde de référence (WGS84) à l'aide de modèles qui sont recalés à partir des résultats de campagnes de mesure. L'outil disponible sur le site [3] utilise le modèle IGRF-2010 (Champ Géomagnétique International de Référence, mis à jour tous les cinq ans). Actuellement, l'amplitude moyenne du champ magnétique elle est de l'ordre de 47 μT au centre de la France. Pour la ville de Toulouse, le champ peut être évalué à l'aide du modèle IGRF-2010 cité ci-avant. Il suffit d'introduire dans l'outil les coordonnées géographiques de la ville. Ces coordonnées ont été relevées sur le site [2] (coordonnées dans le repère WGS84) et sont :

Latitude : 43°36'15" Nord , soit en degrés et minutes décimales : 43°36.2556' Nord

Longitude : 1°26'37" Est, soit en degrés et minutes décimales : 1°26.6202' Est

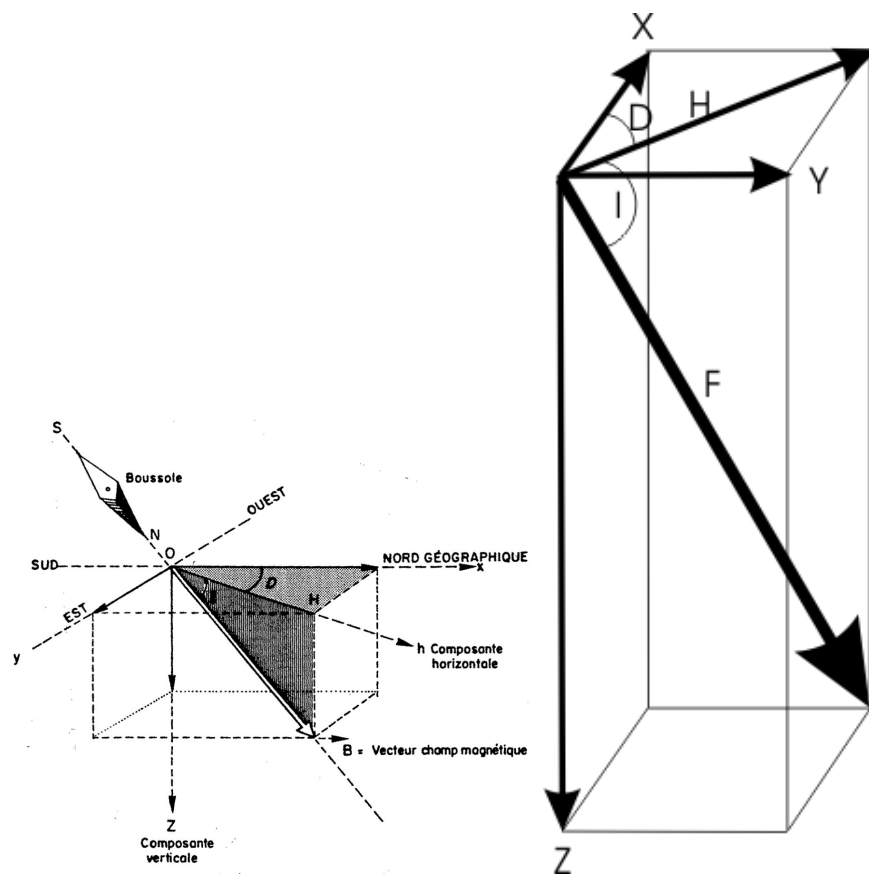


FIGURE 3.1 – Champ magnétique terrestre : inclinaison, déclinaison et composantes

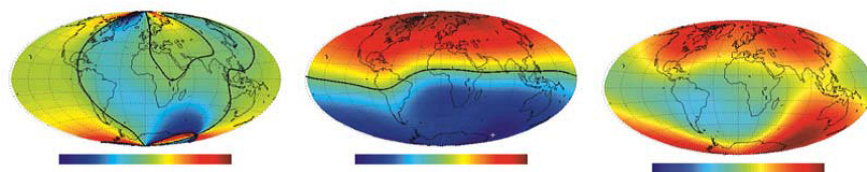


FIGURE 3.2 – Cartographie du champ magnétique terrestre (inclinaison, déclinaison, intensité) réalisée par le satellite Oersted (en 1999)

D [°]	I [°]	H [nT]	Z [nT]	F [nT]	X [nT]	Y [nT]
-0.701	58.939	23766	39458	46063	23765	-291

FIGURE 3.3 – Champ magnétique terrestre à Toulouse

Altitude par rapport au niveau de la mer : 150 m

Les composantes du champ magnétique terrestre fournies par le modèle sont données dans la table ci-dessous.

On mesure la déclinaison D et l'inclinaison I en degrés. Les autres éléments sont des mesures d'amplitude et sont exprimés en nanotesla ($1\text{ nT} = 10^{-9}\text{ Tesla}$) dans le tableau ci-dessus. On utilise aussi parfois le Gauss ($1\text{ Gauss} = 10^{-4}\text{ Tesla}$) ce qui donne un champ magnétique d'amplitude $F = 0.46063[G]$.

3.2.3 Variations temporelles du champ magnétique

Les caractéristiques du champ magnétique varient aussi au cours du temps. On a déjà vu que la position des pôles change, mais l'intensité et mêmes les diverses composantes varient dans le temps. La figure ci-dessous montre les variations de D et I sur 400 ans à Paris. Des variations importantes et rapides surviennent aussi lors d'éruptions solaires. Le champ magnétique terrestre présente en fait une vaste gamme de variations temporelles, de l'année à plusieurs millions d'années.

Sur des horizons de temps courts, des oscillations régulières des caractéristiques du champ magnétiques existent. Elles sont liées à la rotation de la terre sur elle-même et autour du soleil. En France, on mesure des variations diurnes de l'intensité approximativement sinusoïdales et d'amplitude inférieure à 20 nT (c'est-à-dire moins de 0.05%). Sur une année ces oscillations régulières sont de l'ordre de 60 nT . On voit donc que l'on peut raisonnablement considérer que les paramètres d'orientation et d'amplitude du champ magnétique terrestre restent quasiment constants pour les missions qui nous intéressent.

3.2.4 Quelques mesures réalisées

Le champ magnétique enregistré au laboratoire (essai sur table du 4 février 2014) a pour valeur moyenne $[0.32, -0.26, 0.71]$. Ceci correspond à une intensité moyenne de 0.82, c'est-à-dire bien loin de la valeur de $0.47[G]$ attendue... Lors des essais le capteur était approximativement horizontal de sorte que l'inclinaison peut être estimée par $I = \tan^{-1}(Z/\sqrt{X^2 + Y^2})$ si

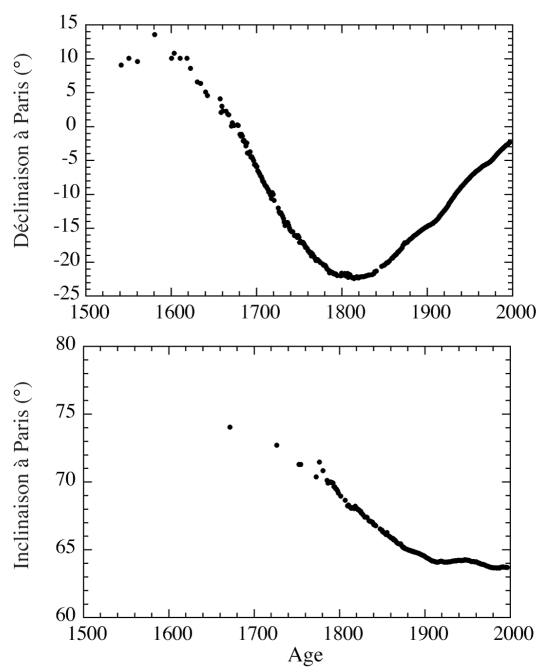


FIGURE 3.4 – Variations temporelle de D et I sur 400 ans à Paris

X,Y et Z désignent les trois composantes du champ magnétique.

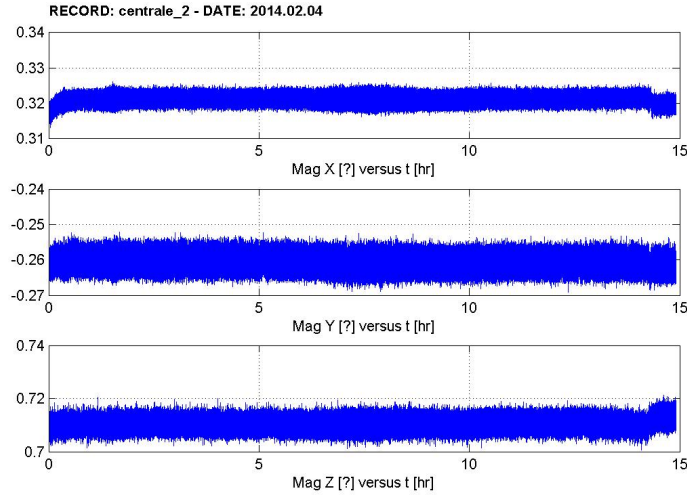


FIGURE 3.5 – Mesure du champ magnétique avec un capteur bas-coût

- [1] http://fr.wikipedia.org/wiki/Champ_magnétique_terrestre
- [2] <http://dateandtime.info>
- [3] <http://geomag.nrcan.gc.ca>

3.3 Accéléromètres et gyromètres

La navigation inertielle pure utilise les mesures accélérométriques et gyrométriques qu'elle intègre pour obtenir la vitesse et la position (en translation), ainsi que l'attitude. Cex sont donc des mesures essentielles pour de nombreux algorithmes de navigation et en particuliers pour ceux présents dans les centrales inertielles.

Les principales erreurs qui affectent ces capteurs sont les bruits de mesure et les biais. Ces biais sont souvent lentement variable, on parle alors de dérive. Pour illustrer la démarche de modélisation nous considérons le

capteur d'un accéléromètre. la mesure qu'il délivre s'écrit :

$$\begin{aligned} A_m &= A + b_A + v_A \\ \dot{b}_A &= v_{b_A} \end{aligned}$$

A est l'accélération spécifique. A_m est la mesure de cette accélération. Elle est entâchée d'un biais b_A et d'un bruit v_A (appelé bruit de mesure). Le biais évolue aléatoirement selon la dynamique décrite par la seconde équation où v_{b_A} est un autre bruit (appelé parfois bruit de dérive).

Cette modélisation assez simple et rustique des sources d'erreur est souvent suffisante pour donner des performances correctes une fois intégrer dans un algorithme de navigation.

Pour un gyromètre, le biais peut souvent être considéré comme constant et donc être estimé et éliminé au moment de la mise en marche de l'algorithme de navigation. Cela est dû au fait que la majorité des gyromètres sont des gyromètres intégrateur-dérivateur. On mesure en fait des petits changements d'attitude que l'on intègre en analogique jusqu'à obtenir une valeur assez grande par rapport au bruit (que l'intégration réduit) et au niveau de quantification du convertisseur analogique-numérique. La valeur de changement d'attitude est ensuite dérivée numériquement pour délivrer une vitesse de rotation.

C'est pour cela qu'un modèle couramment utilisé pour représenter un gyromètre est :

$$\omega_m = \omega + v_\omega$$

Chapitre 4

Estimation d'état réursive : un état de l'art méthodologique

4.1 Introduction

Ce chapitre présente un bref état de l'art des méthodes d'estimation récurives pour les systèmes stochastiques markovien. Il n'est pas exhaustif, mais indique seulement les grandes lignes d'une arborescence de méthodes. Notre objectif est essentiellement ici de positionner l'approche que nous avons développée parmi celles existantes.

Nous nous plaçons dans le cadre bayésienne, et l'on commence par décrire la problématique de l'estimation réursive et les diverses notions d'estimation couramment utilisées (4.2). Ces estimations de l'état du système se calculent à partir de la densité de probabilité de l'état conditionnée par la connaissance des mesures.

On établit ensuite (4.3) les équations générales du filtrage qui montrent que cette densité de probabilité conditionnelle peut, théoriquement, se calculer de manière réursive. Elles comportent deux étapes : la prédiction (équation de Chapman-Kolmogorov) et la correction (règle de Bayes).

On introduit ensuite les deux cas fondamentaux pour lesquels une solution exacte au problème du filtrage est disponible : le cas des chaînes de Markov finies et le cas des systèmes linéaires gaussiens (4.4). Nous aurons

recours à ces deux cas, entremêlés, pour décrire notre problématique de fusion multi-capteurs.

Dans le cas de modèles dynamiques généraux, une solution analytique exacte aux équations du filtrage n'existe pas. Des solutions numériques approchées sont alors cherchées. Elles peuvent être classées en deux grandes familles : les approximations paramétriques et les approximations non paramétriques. Chacune de ces familles est rapidement décrite (4.5 et 4.6). Pour une analyse plus exhaustive, ou pour obtenir plus de détails sur certaines familles de méthodes on pourra consulter par exemple [4, 5].

La problématique de l'estimation à partir d'un ensemble de capteurs pouvant présenter des défaillances s'inscrit dans le cadre de l'estimation pour les systèmes hybrides. Nous l'abordons pour terminer ce rapide panorama méthodologique (4.7).

4.2 Position du problème

4.2.1 Modèles stochastiques

D'une manière générale, le problème d'estimation considéré ici consiste à reconstruire la valeur x_k d'un signal aléatoire x à l'instant courant k à partir de la séquence de ses observations sur les instants antérieurs

$$Y_k = \{y_i, i = 0, \dots, k\}$$

L'évolution du signal x est supposée markovienne. Cela signifie que si on connaît x_k la valeur de x à l'instant k , alors la connaissance de x aux instants antérieurs n'apporte aucune information sur l'évolution future de x . On dit que le présent résume le passé.

Le signal de mesure y est supposé lui sans mémoire. C'est-à-dire que y_k dépend de x_k , mais pas des valeurs antérieures de x .

Le comportement d'un tel signal peut être décrit par les équations suivantes.

Système dynamique markovien :

$$\begin{array}{lcl} x_{k+1} & = & f_k(x_k, w_k) \\ y_k & = & h_k(x_k, v_k) \end{array} \quad (4.1)$$

x_k représente l'état du système à l'instant k . Sa dynamique est régie par l'équation récurrente (4.1, première équation). f_k est la fonction de transition. w_k est le bruit d'état ; Sa densité de probabilité p_k^w est supposée connue.

y_k représente l'observation (bruitée) à l'instant k . Il est lié à l'état par l'équation de mesure (4.1, deuxième équation). h_k est la fonction d'observation. v_k est le bruit de mesure ; Sa densité de probabilité p_k^v est supposée connue.

Le filtrage consiste à déterminer, à partir des mesures disponibles $Y_k = \{y_i, i = 0, \dots, k\}$ jusqu'à l'instant courant k , le meilleur estimateur \hat{x}_k de l'état x_k optimisant un critère de performance choisi. Ce filtrage est dit récursif quand le calcul de l'estimé (\hat{x}_k) à l'instant courant k ne fait intervenir que le résultat des calculs réalisés à l'instant précédent, et la dernière mesure acquise à l'instant courant (y_k). Nous nous intéressons de manière spécifique au filtrage récursif dans le but de limiter le volume de calcul à réaliser.

4.2.2 Estimateurs d'état

Nous utiliserons essentiellement deux types d'estimateurs : l'estimateur à maximum de vraisemblance et l'estimateur à minimum de variance. Leurs définitions sont données ci-dessous. On rappelle aussi la définition de l'estimateur à maximum de vraisemblance trajectorien qui est la rolls-royce des estimateurs.

- **Estimateur à maximum de vraisemblance**

Cet estimateur maximise la vraisemblance de l'état à l'instant courant étant donnée les observations sur le passé. Par définition cette vraisemblance est la probabilité conditionnelle $p(x_k|Y_k)$.

$$\boxed{\hat{x}_k = \arg \max_{x_k} p(x_k|Y_k)} \quad (4.2)$$

- **Estimateur à minimum de variance**

Cet estimateur minimise la variance de l'erreur d'estimation connaissant les mesures passées :

$$\hat{x}_k = \arg \min_{\xi_k} E (||x_k - \xi_k|| \mid Y_k)$$

On montre qu'il coïncide avec l'espérance conditionnelle $E(x_k | Y_k)$, ce qui s'écrit encore :

$$\boxed{\hat{x}_k = \int x_k p(x_k|Y_k) dx_k} \quad (4.3)$$

Ces deux estimateurs nécessitent donc, pour être calculés, la connaissance de la densité probabilité de l'état conditionnelle aux mesures $p(x_k|Y_k)$. Les équations générales du filtrage présentées ci-dessous (4.5 et 4.7) montrent que le calcul de $p(x_k|Y_k)$ s'écrit sous forme récursive. C'est pour cela que l'on parle d'estimation récursive.

- **Estimateur à maximum de vraisemblance trajectorielle**

Cet estimateur cherche à déterminer la trajectoire de l'état (notée $X_k = \{x_i, i = 0, \dots, k\}$) la plus vraisemblable au vue de la séquence d'observations ($Y_k = \{y_i, i = 0, \dots, k\}$). Il est défini par :

$$\hat{X}_k = \arg \max_{X_k} p(X_k|Y_k) \quad (4.4)$$

Ici $p(X_k|Y_k)$ est la vraisemblance conditionnelle de la trajectoire d'état X_k sachant les observations Y_k .

Contrairement aux estimateurs à maximum de vraisemblance et à minimum de variance, l'estimateur à maximum de vraisemblance trajectorielle ne s'écrit généralement pas sous forme récursive.

4.3 Equations générales du filtrage

L'objet du filtrage est de reconstruire la densité de probabilité de l'état conditionnelle aux mesures $p(x_k|Y_k)$. Cette densité sera ensuite utilisée pour construire une estimation de l'état au sens du maximum de vraisemblance ou au sens du minimum de variance.

L'utilisation de quelques rudiments de probabilité conditionnelle et en particulier du caractère markovien de x et de la règle de Bayes, nous permettent de montrer que le calcul de $p(x_k|Y_k)$ s'écrit sous une forme récursive.

4.3.1 Prédiction

Supposons connue $p(x_{k-1}|Y_{k-1})$ la densité à l'instant $k - 1$. L'étape de prédiction consiste à calculer la densité a priori $p(x_k|Y_{k-1})$, c'est-à-dire la densité de l'état à l'instant k conditionnelle aux $k - 1$ dernières observations.

L'intégration de $p(x_k|x_{k-1})$ sur l'ensemble des valeurs possibles de x_{k-1} donne $p(x_k)$. C'est le théorème des probabilités composées :

$$p(x_k) = \int p(x_k|x_{k-1}) p(x_{k-1}) dx_{k-1}$$

Appliqué à la densité conditionnelle aux mesures passées Y_{k-1} (qui est toujours une densité de probabilité), ce théorème donne :

$$p(x_k|Y_{k-1}) = \int p(x_k|x_{k-1}, Y_{k-1}) p(x_{k-1}|Y_{k-1}) dx_{k-1}$$

Le caractère markovien permet d'assurer que $p(x_k|x_{k-1}, Y_{k-1}) = p(x_k|x_{k-1})$ puisque les mesures sur le passé Y_{k-1} n'apportent pas d'information par rapport à la connaissance de l'état à l'instant $k-1$. On obtient ainsi la formule de prédiction encore appelée équation de Chapman-Kolmogorov :

$$\boxed{p(x_k|Y_{k-1}) = \int p(x_k|x_{k-1}) p(x_{k-1}|Y_{k-1}) dx_{k-1}} \quad (4.5)$$

Cette formule fait apparaître la probabilité de transition $p(x_k|x_{k-1})$. Dans la suite nous considérerons souvent des bruits d'état additifs. C'est-à-dire que l'équation d'état est de la forme : $x_{k+1} = f_k(x_k) + w_k$. La probabilité de transition s'exprime alors facilement en fonction de la densité p_k^w du bruit d'état.

$$\begin{aligned} x_{k+1} &= f_k(x_k) + w_k \\ \implies \boxed{p(x_k|x_{k-1}) &= p_{k-1}^w(x_k - f_{k-1}(x_{k-1}))} \end{aligned} \quad (4.6)$$

4.3.2 Correction

Pour obtenir $p(x_k|Y_k)$ il suffit maintenant d'appliquer la règle de Bayes. De même que précédemment, nous commençons simplement en utilisant seulement le conditionnement par rapport à la mesure y_k obtenue à l'instant courant k .

$$\begin{aligned} p(x_k|y_k) &= \frac{p(y_k|x_k) p(x_k)}{p(y_k)} \\ \text{où } p(y_k) &= \int p(y_k|x_k) p(x_k) dx_k \end{aligned}$$

Appliquons maintenant cela à la densité conditionnelle aux mesures passées Y_{k-1} . Il vient :

$$\begin{aligned} p(x_k|y_k, Y_{k-1}) &= \frac{p(y_k|x_k, Y_{k-1}) p(x_k|Y_{k-1})}{p(y_k|Y_{k-1})} \\ \text{où } p(y_k|Y_{k-1}) &= \int p(y_k|x_k, Y_{k-1}) p(x_k|Y_{k-1}) dx_k \end{aligned}$$

Là encore le caractère markovien permet de simplifier les expressions puisque : $p(y_k|x_k, Y_{k-1}) = p(y_k|x_k)$. On obtient ainsi l'équation de correction cherchée

qui permet le calcul de la densité a posteriori (i.e. après la mesure) à partir de la densité a priori (i.e. avant la mesure) :

$$\boxed{\begin{aligned} p(x_k|Y_k) &= \frac{p(y_k|x_k) p(x_k|Y_{k-1})}{p(y_k|Y_{k-1})} \\ \text{où } p(y_k|Y_{k-1}) &= \int p(y_k|x_k) p(x_k|Y_{k-1}) dx_k \end{aligned}} \quad (4.7)$$

Cette formule fait apparaître la vraisemblance des mesures $p(y_k|x_k)$. Dans la suite nous considérerons souvent des bruits de mesure additifs. C'est-à-dire que l'équation de mesure est de la forme : $y_k = h_k(x_k) + v_k$. La vraisemblance s'exprime alors facilement en fonction de la densité p_k^v du bruit de mesure.

$$\begin{aligned} y_k &= h_k(x_k) + v_k \\ \implies &\boxed{p(y_k|x_k) = p_k^v(y_k - h_k(x_k))} \end{aligned} \quad (4.8)$$

4.4 Solutions exactes

Les équations du filtrage récursif ne peuvent pas se résoudre de manière exacte sauf dans certains cas très particuliers : le cas des systèmes linéaires gaussiens, et le cas des chaînes de Markov finies.

4.4.1 Systèmes linéaires gaussiens

Dans ce cas le modèle prend la forme suivante :

$$\boxed{\begin{aligned} x_{k+1} &= F_k x_k + G_k w_k \\ y_k &= H_k x_k + v_k \end{aligned}} \quad (4.9)$$

La condition initiale suit une loi gaussienne : $x_0 \sim \mathcal{N}(m_0, P_0)$. Les bruits sont blancs gaussiens indépendants centrés : $w_k \sim \mathcal{N}(0, Q_k)$, $v_k \sim \mathcal{N}(0, R_k)$.

On montre dans ce cas que le vecteur d'état x_k reste gaussien au cours du temps et que les densités conditionnelles a priori et a posteriori sont aussi des gaussiennes. On les note :

$$\begin{aligned} p(x_k|Y_{k-1}) &= \mathcal{N}(\hat{x}_{k|k-1}, P_{k|k-1}) \\ p(x_k|Y_k) &= \mathcal{N}(\hat{x}_{k|k}, P_{k|k}) \end{aligned}$$

On remarquera que l'on utilise ici volontairement la notation \hat{x} pour désigner la moyenne des gaussiennes. En effet l'estimation a posteriori de x_k (au

sens du maximum de vraisemblance, ou du minimum de variance) est bien $\hat{x}_{k|k}$, et l'estimation a priori est bien $\hat{x}_{k|k-1}$.

Rappelons encore une fois que dans ce cadre linéaire gaussien, les estimations de l'état à minimum de variance et à maximum de vraisemblance coïncident. Elles sont toutes les deux données ici par \hat{x} .

Les équations générales du filtrage (4.5 et 4.7) se ramènent alors au calcul récursif des moyennes ($\hat{x}_{k|k-1}$ et $\hat{x}_{k|k}$) et des covariances associées ($P_{k|k-1}$ et $P_{k|k}$). Elles ont été établies par Kalman, et portent le nom de filtre de Kalman.

Filtre de Kalman

Prédiction

$$\begin{aligned}\hat{x}_{k+1|k} &= F_k \hat{x}_{k|k} \\ P_{k+1|k} &= F_k P_{k|k} F_k^T + G_k Q_k G_k^T\end{aligned}$$

Correction

$$\begin{aligned}\nu_{k+1} &= y_{k+1} - H_{k+1} \hat{x}_{k+1|k} \\ S_{k+1} &= H_{k+1} P_{k+1|k} H_{k+1}^T + R_{k+1} \\ K_{k+1} &= P_{k+1|k} H_{k+1}^T S_{k+1}^{-1} \\ \hat{x}_{k+1|k+1} &= \hat{x}_{k+1|k} + K_{k+1} \nu_{k+1} \\ P_{k+1|k+1} &= P_{k+1|k} - K_{k+1} H_{k+1} P_{k+1|k}\end{aligned}$$

Le vecteur ν_k est l'innovation, c'est-à-dire l'écart entre la valeur de la mesure et sa valeur prédite. La matrice S_k est la covariance de ν_k .

Nous avons vu ci-dessus que la matrice $P_{k|k}$ est la covariance de $x_k|Y_k$. Mais, comme $\hat{x}_{k|k} = E(x_k|Y_k)$, $P_{k|k}$ est aussi la covariance de l'erreur d'estimation $x_k - \hat{x}_{k|k}|Y_k$. Cette matrice est donc une mesure de la taille de l'erreur d'estimation.

4.4.2 Chaînes de Markov finies

On parle de chaîne de Markov finie quand le vecteur d'état x prend ses valeurs sur un ensemble discret fini $\mathcal{X} = \{e_i, i = 1, \dots, M\}$ (souvent

appelé alphabet) et que la dynamique d'évolution de x est décrite par les probabilités de transition d'une valeur à l'autre $p_k^{i,j}$:

$$p_k^{i,j} = P(x_k = e_i | x_{k-1} = e_j)$$

La densité de probabilité de transition $p(x_k | x_{k-1} = e_j)$ s'écrit donc sous forme d'un peigne de Dirac. Nous lui associons ici une équation de mesure à bruit additif pour obtenir le modèle :

$$\boxed{\begin{aligned} p(x_k | x_{k-1} = e_j) &= \sum_{i=1}^M p_k^{i,j} \delta(x_k - e_i) \\ y_k &= h_k(x_k) + v_k \end{aligned}} \quad (4.10)$$

On montre dans ce cas que les densités conditionnelles a priori et a posteriori sont aussi des peignes de Dirac centrés sur les e_j .

$$\begin{aligned} p(x_k | Y_{k-1}) &= \sum_{i=1}^M \rho_{k|k-1}^i \delta(x_k - e_i) \\ \text{où } \rho_{k|k-1}^i &= P(x_k = e_i | Y_{k-1}) \end{aligned}$$

et

$$\begin{aligned} p(x_k | Y_k) &= \sum_{i=1}^M \rho_{k|k}^i \delta(x_k - e_i) \\ \text{où } \rho_{k|k}^i &= P(x_k = e_i | Y_k) \end{aligned}$$

Avec ces expressions des densités a priori et a posteriori, les équations du filtrage (4.5 et 4.7) se ramènent alors simplement au calcul récursif des probabilités conditionnelles ($\rho_{k|k-1}^i$ et $\rho_{k|k}^i$).

Filtrage pour une chaîne de Markov finie

Prédiction

$$\rho_{k|k-1}^i = \sum_{j=1}^M p_k^{i,j} \rho_{k-1|k-1}^j$$

Correction

$$\rho_{k|k}^i = \frac{\rho_{k|k-1}^i p(y_k | x_k = e_i)}{p(y_k | Y_{k-1})}$$

$$\text{où } p(y_k | Y_{k-1}) = \sum_{j=1}^M \rho_{k|k-1}^j p(y_k | x_k = e_j)$$

$$\text{et } p(y_k | x_k = e_j) = p^v(y_k - h(e_j))$$

Dans le cadre de chaînes de Markov finies, l'estimation à minimum de variance n'a guère de sens puisque c'est une somme pondérée des valeurs possibles de l'état. On lui préférera en général l'estimation au sens du maximum de vraisemblance :

$$\hat{x}_k = e_{i^*} \quad \text{où } i^* = \arg \max_i \rho_{k|k}^i$$

4.4.3 Systèmes conditionnellement gaussiens

Un autre cas intéressant conduit à une solution exacte, il s'agit du cas des systèmes conditionnellement gaussien []. Ces systèmes sont décrits par une dynamique linéaire gaussienne connaissant l'ensemble des mesures passées. C'est-à-dire que les matrices F_k , H_k , ... dépendent de Y_{k-1} .

Dans ce cas l'état n'a plus un comportement gaussien. Mais on montre cependant que les densités conditionnelles restent des gaussiennes. Les équations du filtre de Kalman fournissent alors encore une solution exacte au problème du filtrage.

4.5 Approximations paramétriques

De nombreuses solutions approchées ont été proposées pour résoudre les équations du filtrage de manière approchée. Quand les densités conditionnelles ont une structure fixée (par exemple ce sont des lois gaussiennes), mais dépendent de paramètres (la moyenne et la variance dans notre exemple) on

parle d'approximation paramétrique. C'est ce type de solution qu'on examine ici.

Pour éviter trop de complexité dans l'exposé, nous considérons ici seulement le cas des systèmes dynamiques markovien à bruits additifs. Ils sont décrits par :

$$\boxed{\begin{array}{lcl} x_{k+1} & = & f_k(x_k) + w_k \\ y_k & = & h_k(x_k) + v_k \end{array}} \quad (4.11)$$

En filtrage on préfère souvent une description purement statistique du comportement de signal aléatoire x et de la mesure associée y . On écrit alors, en faisant apparaître les densités de probabilités (ddp) p_k^w et p_k^v des bruits :

$$\boxed{\begin{array}{lcl} p(x_{k+1}|x_k) & = & p_k^w(x_{k+1} - f_k(x_k)) \\ p(y_k|x_k) & = & p_k^v(y_k - h_k(x_k)) \end{array}} \quad (4.12)$$

4.5.1 Approximation gaussienne

La plus connue des solutions approchée est le filtrage de Kalman étendu. Il s'agit d'approximer le modèle par linéarisation autour de l'estimation courante pour pouvoir continuer d'appliquer les équations du filtre de Kalman. On cherche donc dans ce cas une approximation gaussienne des densités, bienqu'elles ne soient plus gaussiennes.

Le filtre de Kalman sans parfum recherche aussi une approximation gaussienne des densités. Il évite cependant l'utilisation de linéarisations. La transformation des moyennes et des covariances des gaussiennes à travers les non linéarités du modèle sont estimées via des échantillonnages (déterministes). On gagne ainsi généralement en précision.

4.5.1.1 Filtre de Kalman étendu

On rappelle ici les équations de ce filtre dans le cas d'un système non linéaire à bruits additifs (4.11).

Filtre de Kalman étendu

Prédiction

$$\text{avec } F_k = \frac{\partial f_k}{\partial x}(\hat{x}_{k|k})$$

$$\begin{aligned}\hat{x}_{k+1|k} &= f_k(\hat{x}_{k|k}) \\ P_{k+1|k} &= F_k P_{k|k} F_k^T + Q_k\end{aligned}$$

Correction

$$\text{avec } H_{k+1|k} = \frac{\partial h_{k+1}}{\partial x}(\hat{x}_{k+1|k})$$

$$\begin{aligned}\nu_{k+1} &= y_{k+1} - h_{k+1}(\hat{x}_{k+1|k}) \\ S_{k+1} &= H_{k+1} P_{k+1|k} H_{k+1}^T + R_{k+1} \\ K_{k+1} &= P_{k+1|k} H_{k+1}^T S_{k+1}^{-1} \\ \hat{x}_{k+1|k+1} &= \hat{x}_{k+1|k} + K_{k+1} \nu_{k+1} \\ P_{k+1|k+1} &= P_{k+1|k} - K_{k+1} H_{k+1} P_{k+1|k}\end{aligned}$$

On voit que la linéarisation permet de calculer les matrices F_k et $H_{k+1|k}$, matrices qui sont nécessaires pour évaluer l'évolution de la covariance lors des étapes de prédiction et de correction, et bien sûr aussi pour calculer le gain de Kalman K_{k+1} .

4.5.1.2 Filtre de Kalman sans parfum

On peut améliorer le filtre de Kalman étendu en calculant l'évolution de la moyenne et de la covariance par des techniques plus élaborées qu'une simple linéarisation. La transformation sans parfum [6] a été introduite dans ce but. Elle utilise un maillage déterministe d'un ellipsoïde autour de l'estimation courante (ellipsoïde défini par la covariance, centré sur la moyenne). Les images des points du maillage par une fonction non linéaire permettent d'évaluer la transformation de l'estimation et de la covariance à travers la non linéarité.

Ainsi le filtre de Kalman sans parfum [7, 8] est un filtre de Kalman étendu qui utilise cette transformation sans parfum pour mettre à jour l'estimation et la covariance sans linéariser les fonctions f et h . On gagne en précision sur l'évolution des gaussiennes utilisées pour approximer les densités a priori

et a posteriori.

L'utilisation de la transformation sans parfum n'est évidemment pas restreinte à l'amélioration du filtre de Kalman étendu.

On notera par ailleurs que cet outil évite le calcul explicite des matrices jacobiniennes de f et h .

4.5.2 Somme de gaussiennes

D'autres méthodes recherchent une approximation des densités sous des formes plus riches que de simples gaussiennes. La plus ancienne est l'approximation par des sommes de gaussiennes, approche développée dans [9]. On emploie maintenant plus volontiers le terme de mélange de gaussiennes.

4.5.2.1 Approximation multi-gaussienne

Dans ce cas la densité de probabilité a posteriori est représentée par une somme de N gaussiennes pondérées.

$$p(x_k|Y_k) = \sum_{j=1}^N \rho_k^j \mathcal{N}(x_k - \hat{x}_{k|k}^j, P_{k|k}^j) \quad (4.13)$$

Les paramètres des composantes gaussiennes \mathcal{N} sont les centres $\hat{x}_{k|k}^j$, et les covariances $P_{k|k}^j$. Les ρ^i sont les poids ou coefficients de pondération. On a bien entendu : $0 \leq \rho^i \leq 1$ et $\sum_{i=1}^N \rho^i = 1$.

L'estimation à minimum de variance (i.e. $\hat{x}_{k|k}$) s'obtient en intégrant x_k pondéré par $p(x_k|Y_k)$. De même, en intégrant l'erreur quadratique on obtient l'erreur quadratique moyenne ($P_{k|k}$). Après quelques manipulations, tout ceci s'écrit sous la forme :

$$\begin{aligned} \hat{x}_{k|k} &= \sum_{i=1}^N \rho_k^i \hat{x}_{k|k}^i \\ P_{k|k} + \hat{x}_{k|k} \hat{x}_{k|k}^T &= \sum_{i=1}^N \rho_k^i \left(P_{k|k}^i + \hat{x}_{k|k}^i (\hat{x}_{k|k}^i)^T \right) \end{aligned} \quad (4.14)$$

Reste à déterminer comment doivent évoluer les caractéristiques et poids des composantes gaussiennes au cours du temps.

4.5.2.2 Equations du filtrage multi-gaussien

On considère toujours le cas d'un système non linéaire à bruits additifs (4.11). Nous supposons que la condition initiale en x et les bruits w et v sont gaussiens. La technique s'étend naturellement au cas de bruits ou d'une condition initiale représentés par des sommes de gaussiennes.

Considérons tout d'abord la partie prédiction du filtrage. L'équation de Chapman-Kolmogorov (4.5) fait intervenir la densité a posteriori $p(x_{k-1}|Y_{k-1})$ et la matrice de transition $p(x_k|x_{k-1})$. Le premier terme est, par hypothèse, une somme de gaussiennes. Le second est une gaussienne puisque (4.12) :

$$p(x_k|x_{k-1}) = p_{k-1}^w(x_k - f_k(x_{k-1}))$$

et que p^w est une gaussienne. Le produit reste une somme de gaussiennes, et l'intégration sur x_{k-1} conduit alors à une densité a priori $p(x_k|Y_{k-1})$ qui est encore une somme de gaussiennes.

En correction, l'équation (4.7) fait aussi intervenir un produit de deux termes : la densité a priori $p(x_k|Y_{k-1})$ qui est une somme de gaussiennes et la vraisemblance de la mesure $p(y_k|x_k)$ qui est une gaussienne. En effet cette dernière s'écrit (4.12) :

$$p(y_k|x_k) = p_k^v(y_k - h_k(x_k))$$

qui est une gaussienne car p^v l'est. Le produit reste une somme de gaussiennes.

Les équations générales du filtrage montrent aussi que les caractéristiques (centres, covariances) des composantes gaussiennes des densités (a priori et a posteriori) évoluent conformément aux équations de Kalman. Les poids des composantes sont quant à eux mis à jour proportionnellement à la vraisemblance des mesures.

Filtre de Kalman multi-Gaussien (bruits faibles)

Prédiction

$$\text{avec } F_k^j = \frac{\partial f_k}{\partial x}(\hat{x}_{k|k}^j)$$

$$\begin{aligned}\hat{x}_{k+1|k}^j &= f_k(\hat{x}_{k|k}^j) \\ P_{k+1|k}^j &= F_k^j P_{k|k}^j (F_k^j)^T + Q_k \\ \rho_{k+1|k}^j &= \rho_{k|k}^j\end{aligned}$$

Correction

$$\text{avec } H_{k+1|k}^j = \frac{\partial h_{k+1}}{\partial x}(\hat{x}_{k+1|k}^j)$$

$$\begin{aligned}\nu_{k+1}^j &= y_{k+1} - h_{k+1}(\hat{x}_{k+1|k}^j) \\ S_{k+1}^j &= H_{k+1}^j P_{k+1|k}^j (H_{k+1}^j)^T + R_{k+1} \\ K_{k+1}^j &= P_{k+1|k}^j (H_{k+1}^j)^T (S_{k+1}^j)^{-1} \\ \hat{x}_{k+1|k+1}^j &= \hat{x}_{k+1|k}^j + K_{k+1}^j \nu_{k+1}^j \\ P_{k+1|k+1}^j &= P_{k+1|k}^j - K_{k+1}^j H_{k+1}^j P_{k+1|k}^j \\ \rho_{k+1|k+1}^j &= \frac{\rho_{k+1|k}^j \mathcal{N}(\nu_{k+1}^j, S_{k+1}^j)}{\sum_i \rho_{k+1|k}^i \mathcal{N}(\nu_{k+1}^i, S_{k+1}^i)}\end{aligned}$$

Du point de vue calculatoire, tout se passe donc comme si on avait N filtres de Kalman en parallèle.

4.5.2.3 Cas de non linéarités fortes

Le filtre multi-gaussien utilise plusieurs linéarisations. Pour qu'elles soient valables il convient que les covariances des bruits restent faibles. Les linéarisations doivent garder un sens sur le domaine d'incertitude autour du point de linéarisation.

Si cela n'est pas vérifié, les différentes composantes gaussiennes vont se confondre et on retrouvera asymptotiquement le comportement d'un simple filtre de Kalman étendu, éventuellement divergeant.

En fait quand les non linéarités sont fortes par rapport aux covariances

des bruits, on a intérêt pour éviter cette dégénérescence à remplacer les modèles gaussiens des bruits par des modèles multi-gaussiens, et cela même si les bruits sont réellement gaussiens. Les covariances des diverses composantes de ces approximations doivent être suffisamment petites pour que les linéarisations restent valables.

L'inconvénient de cette démarche est de faire croître (de façon exponentielle) le nombre de composantes gaussiennes au cours des itérations. Si par exemple on utilise N_w composantes pour représenter le bruit d'état w , l'étape de prédiction multiplier le nombre de composantes gaussiennes par N_w . De même si on utilise N_v composantes pour modéliser le bruit de mesure, l'étape de correction multiplier le nombre de gaussiennes par N_v . Afin de maîtriser la complexité calculatoire il est alors nécessaire d'introduire un processus de sélection. On peut par exemple éliminer les composantes de poids négligeables, ou encore regrouper certaines composantes voisines, de sorte que le nombre total de composantes gaussiennes reste identique au cours des itérations.

4.5.2.4 Autres bases de fonctions

Il est également possible de décomposer les densités de probabilité sur d'autres fonctions que des gaussiennes. En particulier quand l'espace d'état est un hypercube, on peut aussi utiliser des bases de fonctions comme les exponentielles complexes (Fourier), les polynômes d'Hermitte, ...

Nous avons vu qu'un des intérêts des sommes de gaussiennes est que lors des étapes de prédiction et de correction, les sommes de gaussiennes restent des sommes de gaussiennes. Cette propriété de maintien dans l'espace d'approximation choisi n'est pas satisfaite avec d'autres fonctions que les gaussiennes. Il faudra donc généralement mettre en place une procédure de projection sur l'espace d'approximation après ces étapes.

Une autre manière de procéder est d'utiliser différents espaces d'approximation, chacun densité (a priori, a posteriori, transition) étant approximée dans un espace dédié. Dans cet esprit de plusieurs solutions ont été proposées comme par exemple des mélanges de gaussiennes et de peignes de Dirac.

4.6 Approximations non paramétriques

On parle d'approximation non paramétrique quand les densités conditionnelles sont approchées par des sommes Dirac pondérées. On écrira par exemple la densité conditionnelle a posteriori sous la forme :

$$p(x_k|Y_k) \approx \hat{p}(x_k|Y_k) = \sum_{i=1}^N w_k^i \delta(x_k - x_k^i) \quad (4.15)$$

Les x_k^i sont les positions des Dirac et les w_k^i leurs poids ($w_k^i \geq 0$, $\sum_i w_k^i = 1$). On pourrait bien entendu dire que les x_k^i et les w_k^i sont des paramètres à estimer et que l'on a encore une approximation paramétrique, mais la qualité de l'approximation est ici plus liée au nombre de Dirac qu'à leurs caractéristiques individuelles.

La qualité de l'approximation de p par \hat{p} ne peut évidemment pas être évaluée en comparant leurs valeurs ponctuelles puisque p est généralement une fonction, alors que \hat{p} n'en est pas une ($\hat{p}(x)$ est nul pour presque toutes les valeurs de x). On examinera plutôt si les moments (moyenne et variance généralement) évalués avec \hat{p} sont proches de ceux évalués avec p . Ainsi pour la moyenne on regardera si :

$$E[x_k|Y_k] = \int x_k p(x_k|Y_k) dx_k \approx \int x_k \hat{p}(x_k|Y_k) dx_k = \sum_{i=1}^N w_k^i x_k^i$$

4.6.1 Méthodes déterministes

Certaines méthodes non paramétriques utilisent un maillage déterministe et fixe de l'espace d'état. Elles sont bien entendu restreintes aux très petites dimensions.

La première approximation considérée utilise un maillage fixe de l'espace d'état. Le filtre associé porte le nom de Point Mass Filter.

4.6.2 Méthodes stochastiques

Les méthodes stochastiques reposent sur le principe de Monte-Carlo. On réalise des tirages aléatoires et les échantillons sont moyennés pour approximer l'espérance. Le point de départ est donc l'échantillonnage aléatoire rappelé ci-après.

Appliqué au problème de l'estimation de l'état des systèmes dynamiques stochastiques, ce principe a conduit au développement du filtrage dit particulier. Ce filtrage cherche à approcher la loi conditionnelle de l'état connaissant les observations par un peigne de Diracs pondérés. Les points d'un échantillon sont appelées particules. Leurs localisations et leurs poids sont mis à jour récursivement.

4.6.2.1 L'échantillonnage aléatoire

L'intérêt de l'utilisation de tirages aléatoires provient de loi des grands nombres. Reprenons la formule d'approximation d'une densité par un peigne de Dirac pondérés (équation 4.15). Quand tous les poids sont égaux ($w_k^i = 1/N$) et si les positions des Dirac (les x_k^i) sont des échantillons de la vraie densité (i.e. des valeurs choisies au hasard selon la même loi de probabilité p) alors la loi des grands nombre assure que lorsque N est assez grand, l'approximation est correcte (l'erreur est majorée par $1/\sqrt{N}$) :

$$\left\{x_k^i\right\}_{i=1}^N \sim p(x_k|Y_k) \implies \left|E[x_k|Y_k] - \frac{1}{N} \sum_{i=1}^N x_k^i\right| \leq 1/\sqrt{N}$$

Dans notre contexte de filtrage la loi $p(x_k|Y_k)$ n'est pas connue puisque c'est le processus de filtrage qui doit la construire afin d'estimer x_k . Réaliser le tirage aléatoire des x_k^i suivant cette loi de probabilité n'est donc pas immédiat. Mais plusieurs solutions existent à ce problème : les méthodes de Monte-Carlo par chaîne de Markov (algorithme de Metropolis-Hasting, l'échantillonneur de Gibbs, ...), les méthodes d'échantillonnage pondéré, d'acceptation-rejet, etc.

Ainsi par exemple la formule d'approximation d'une densité par un peigne de Dirac pondérés (équation 4.15). correspond à ce qui est appelé un échantillonnage pondéré (*importance sampling* en anglais). Les x_k^i sont tirés aléatoirement selon une loi q connue et proche de p . Les poids sont ajustés pour tenir compte de (et réduire) l'écart entre q et p . La qualité de l'approximation est évidemment moins bonne dans ce cas.

4.6.2.2 Le filtrage particulier

Le filtrage particulier cherche à approcher la loi conditionnelle de l'état connaissant les observations par un peigne de Dirac pondéré. Les points d'un échantillon sont appelées particules. C'est sans aucun doute la méthode la plus étudiée ces vingt dernières années. Il en existe de nombreuses variantes.

Le premier type de filtre particulaire apparu dans la littérature est le filtre de Monte-Carlo pondéré, encore appelé algorithme d'échantillonnage pondéré (*importance sampling*). Cet algorithme est malheureusement instable car toutes les pondérations s'annulent sauf une qui tend vers un. Pour éviter cette dégénérescence des pondérations plusieurs solutions (ou procédures de régularisation) ont été proposées comme par exemple l'utilisation d'un facteur d'oubli diminuant l'influence des poids passés.

Une autre difficulté du filtrage particulaire est son coût calculatoire élevé. Pour le limiter une approche est de privilégier les particules de forte vraisemblance (i.e. de poids importants). Cela peut se faire en rééchantillonnant la densité (il y aura plus de particules générées sur les Dirac de poids forts). C'est ce que fait l'algorithme d'échantillonnage séquentiel (*sampling importance resampling*, *bootstrap filter* ou encore *Interacting particle filter*). La redistribution des particules induite par le rééchantillonnage présente néanmoins l'inconvénient de concentrer les particules dans une petite région de l'espace d'état. Cette dégénérescence des particules rend l'algorithme instable. Là encore des procédures de régularisation sont nécessaires.

4.7 Systèmes hybrides

La problématique applicative considérée dans notre travail consiste à réaliser l'estimation à partir de mesure provenant de capteurs dont les modes de fonctionnement peuvent varier à la suite de l'occurrence de pannes, dysfonctionnement ou plus généralement de modification des conditions de prise de mesure. L'état des capteurs est représenté par une variable discrète s qui varie aléatoirement et que nous voulons estimer en même temps que la variable continue x . Nous resterons dans le cadre d'une dynamique linéaire, et de tels modèles sont alors souvent dénomés modèles d'état à commutation (ou encore à saut).

La dynamique x est donc supposée définie par équation (linéaire) de la forme

$$\begin{aligned} x_t &\sim p(x_t | x_{t-1}, s_t) \\ z_t &\sim p(z_t | x_t, s_t) \end{aligned}$$

On notera que le mode de fonctionnement des capteurs ne joue pas seulement dans l'équation de mesure mais aussi dans l'équation d'état. Cela vient du fait que certaines composantes de x représentent l'état des capteurs.

A ce modèle il faut adjoindre un modèle d'évolution du mode de fonctionnement. Selon sa structure le modèle sera plus ou moins bien représentatif, mais aussi la résolution (approchée) des équations du filtrage sera plus ou moins complexe.

Le cas le plus simple consiste à dire que les s_t sont des variables aléatoires indépendantes de probabilités fixes $Pr(s_t = i) = \rho^i$. Ces probabilités peuvent être connues ou non, mais dans ce dernier cas il faudra bien entendu les estimer.

On peut enrichir ce modèle en supposant les probabilités variables au cours du temps par exemple selon un modèle markovien $\rho_t^i \sim p(\rho_t^i | \rho_{t-1}^i)$. Il faudra alors nécessairement les estimer.

Il semble plus réaliste de considérer comme donnée la probabilité de changer de mode de fonctionnement plutôt que celle d'être dans un mode. Dans ce cas l'évolution de s_t est représentée par sa matrice transition $P_{ij} = Pr(s_t = i | s_{t-1} = j)$. Là encore ces probabilités sont fixes, et si elles ne sont pas connues il faudra les estimer.

Dans le contexte linéaire choisi, ce type de modélisation est appelé système linéaire markovien à saut [10]. L'approximation la plus utilisée pour résoudre les équations du filtrage est alors le mélange de gaussiennes. Il en va ainsi du célèbre algorithme IMM [11].

Une lacune des modèles ci-dessus est de considérer que le mode de fonctionnement peut changer à chaque instant alors que physiquement une certaine latence est attendue. Pour la représenter une solution serait d'introduire une mémoire long terme dans le modèle d'évolution du mode s .

Mais ceci peut être antagoniste avec le souhait de construire un algorithme de filtrage qui détecte rapidement les changements de mode.

Les modélisations utilisées dans [12, 13] font appel à un hyperparamètre qui règle la dynamique de x et s et dont l'évolution crée cet effet de mémoire long terme du mode de fonctionnement. Les algorithmes de filtrage associés utilisent les outils du filtrage particulaire.

La solution que nous avons développé repose sur l'hypothèse d'une dynamique markovienne pour le couple $(x, s) : (x_t, s_t) \sim p(x_t, s_t | x_{t-1}, s_{t-1})$. Nous lui associons, comme ci-dessus, une chaîne de markov pour représenter les changements de modes $P_{ij} = Pr(s_t = i | s_{t-1} = j)$. On montre alors que $x_t \sim p(x_t | x_{t-1}, s_t, s_{t-1})$. Un effet de mémoire apparaît donc.

Mais cette expression laisse aussi suggérer une complexification de l'algo-

rythme de filtrage associé puisqu'il faudra considérer non pas la valeur du mode, mais les valeurs successives, c'est-à-dire les transitions.

Bibliographie

- [1] D. Titterton, *Strapdown Inertial navigation Technology, 2nd Edition*, The Institution of Engineering and Technology, 2004.
- [2] M. Llibre, “Représentation des attitudes”, Note technique, ONERA, février 2009.
- [3] K. Shoemake, “Animating rotation with quaternion curves”, *Computer Graphics*, vol. 19, no. 3, july 1985.
- [4] S. Haykin et al., *Kalman filtering and neural networks*, New York : Wiley, 2001.
- [5] A.s Ziadi, *Particules gaussiennes déterministes en maximum de vraisemblance non-linéaire : application au filtrage optimal des signaux radar et GPS*, PhD thesis, Université Paul Sabatier, Toulouse III, Laboratoire d’Analyse et d’Architecture des Systèmes (LAAS) - CNRS, 2007.
- [6] S. J. Julier and J. K. Uhlmann, “A new extension of the kalman filter to nonlinear systems”, in *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, 1997.
- [7] S. J. Julier and J. K. Uhlmann, “Unscented filtering and nonlinear estimation”, *proc. IEEE*, vol. 92, no. 3, pp. 401–422, March 2004.
- [8] E. Wan and R. Van der Merwe, “The unscented kalman filter for nonlinear estimation”, in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*. IEEE, 2000, pp. 153–158.
- [9] D. L. Alspach and H. W. Sorenson, “Nonlinear Bayesian estimation using Gaussian sum approximation”, *IEEE Transactions on Automatic Control*, vol. 17, no. 4, pp. 439–448, 1972.
- [10] J. Tugnait, “Adaptive estimation and identification for discrete systems with markov jump parameters”, *IEEE Trans. on Automatic Control*, vol. 27, no. 5, pp. 1054–1065, May 1982.

- [11] H. A. Blom and Y. Bar-Shalom, “The interacting multiple model algorithm for systems with Markovian switching coefficients”, *IEEE Transactions on Automatic Control*, vol. 33, no. 8, pp. 780–783, 1988.
- [12] C. Hue, J.-P. Le Cadre, and P. Pérez, “Sequential monte carlo methods for multiple target tracking and data fusion”, *IEEE Trans. on Signal Processing*, vol. 50, no. 2, pp. 309–325, February 2002.
- [13] F. Caron, M. Davy, E. Duflos, and P. Vanheeghe, “Particle filtering for multisensor data fusion with switching observation models : Application to land vehicle positioning”, *Trans. Sig. Proc.*, vol. 55, no. 6, pp. 2703–2719, June 2007.