



MARMOTTES

COMMUNICATION ET SYSTÈMES

SYSTÈMES D'INFORMATIONS

DIRECTION ESPACE

Édition : 4 Date : 01/02/2002

Révision : 0 Date : 01/02/2002

Réf. : ESPACE/MS/CHOPE/MARMOTTES/DM/001

Documentation mathématique

Rédigé par : L. Maisonobe CS SI/ESPACE/MS	le :	
Validé par : O. Queyruat CS SI/ESPACE/ETS	le :	
Pour application : Ch. Hulot CS SI/ESPACE/MS	le :	

Pièces jointes :

BORDEREAU D'INDEXATION

CONFIDENTIALITÉ : NC	MOTS CLEFS : orientation, attitude, multimission, bibliothèque
-------------------------	--

TITRE DU DOCUMENT :

Documentation mathématique

AUTEUR(S) : L. Maisonobe

RÉSUMÉ :

Ce document décrit les modèles mathématiques et les méthodes de résolution utilisées par la bibliothèque multi-missions MARMOTTES et les éléments de plus bas niveau considérés comme réutilisables qui ont été regroupés dans la bibliothèque CANTOR.

DOCUMENTS RATTACHÉS : ce document vit seul		LOCALISATION :	
VOLUME : 1	NBRE TOTAL DE PAGES : 79 DONT PAGES LIMINAIRES : 5 NBRE DE PAGES SUPPL. :	DOCUMENT COMPOSITE : N	LANGUE : FR
GESTION DE CONF. : oui	RESP. GEST. CONF. : LUC MAISONOBE - CSSI		
CAUSES D'ÉVOLUTION : passage au format CS			
CONTRAT : Néant			
SYSTÈME HÔTE : Unix – L ^A T _E X 2 _ε			

DIFFUSION INTERNE

Nom	Sigle	BPi	Observations
CS SI	ESPACE/MS		2 exemplaires

DIFFUSION EXTERNE

Nom	Sigle	Observations
CNES	DTS/MPI/MS/SG	3 exemplaires

MODIFICATIONS

Éd.	Rév.	Date	Référence, Auteur(s), Causes d'évolution
1	0	28/02/95	L. Maisonobe
1	1	22/08/95	L. Maisonobe, création d'une annexe sur les prolongements
2	0	02/10/96	L. Maisonobe, description des algorithmes de recherches de zéros
2	1	13/02/97	L. Maisonobe
2	2	24/06/97	L. Maisonobe
2	3	20/05/98	L. Maisonobe, annexe sur les angles de CARDAN
3	0	31/05/99	L. Maisonobe, refonte des modèles géométriques pour traiter complètement les singularités
3	1	21/09/99	G. Filaire, L. Maisonobe, ajout des cibles nadir, position, direction et station pour les capteurs optiques, introduction des différents repères de base pour les capteurs de Cardan
3	2	07/06/00	L. Maisonobe, préparation pour diffusion publique
3	3	05/09/00	L. Maisonobe, reformatage des pages liminaires
3	4	22/11/00	L. Maisonobe, ajout des capteurs de gain d'antenne, reformulation générale pour la publication d'une note technique
3	5	26/01/01	L. Maisonobe, prise en compte de remarques de relecture
3	6	22/06/01	L. Maisonobe, remplacement de « horizon » par « limbe », mise à jour des documents de référence
3	7	17/07/01	L. Maisonobe, correction d'une coquille et de la hiérarchie des capteurs
4	0	01/02/02	L. Maisonobe, création du document CS

SOMMAIRE

GLOSSAIRE	1
DOCUMENTS DE RÉFÉRENCE	2
DOCUMENTS APPLICABLES	2
1 Introduction	3
2 Description d'orientation	5
2.1 description classique	5
2.2 inversion du problème	6
2.3 conséquences pour l'utilisateur	8
2.4 conséquences pour l'implémentation	9
2.5 limitations du modèle	9
3 Capteurs	10
3.1 capteurs géométriques	10
3.2 capteurs cinématiques	11
4 Lois de pilotage	12
4.1 contrôle par trois capteurs géométriques	12
4.2 contrôle par deux capteurs géométriques et un capteur cinématique	13
4.3 contrôle par deux capteurs cinématiques et un capteur géométrique	14
4.4 contrôle par trois capteurs cinématiques	14

5	Démarche de la résolution	15
6	Orientations respectant deux consignes identiques	16
6.1	cas de deux consignes géométriques	16
6.1.1	résolution générale	17
6.1.2	traitement des singularités	19
6.1.3	familles à un vecteur fixe	22
6.1.4	réduction du domaine de validité	23
6.1.5	synthèse	24
6.2	cas de deux consignes cinématiques	26
7	fonction d'erreur sur une consigne	27
7.1	cas d'une consigne géométrique	28
7.2	cas d'une consigne cinématique	28
8	Modélisation des capteurs	28
8.1	caractéristiques générales	29
8.2	caractéristiques des capteurs géométriques	30
8.3	caractéristiques des capteurs d'ascension droite	30
8.4	caractéristiques des capteurs de déclinaison	31
8.5	caractéristiques des capteurs optiques	32
8.6	caractéristiques des capteurs vecteurs	34
8.7	caractéristiques des capteurs plan-vecteurs (élévation)	35
8.8	caractéristiques des capteurs cartésiens	35
8.9	caractéristiques des capteurs dièdres	35
8.10	caractéristiques des capteurs Terre	36
8.11	caractéristiques des capteurs fonctions	37
8.12	caractéristiques des capteurs de gain échantillonnés	38
8.13	caractéristiques des capteurs de gain en $(\sin(\theta/\theta_0)/(\theta/\theta_0))^2$	38

8.14	caractéristiques des capteurs de gain en fonction gaussienne	39
8.15	caractéristiques des capteurs de Cardan	39
8.16	caractéristiques des capteurs cinématiques	40
8.17	caractéristiques des gyromètres intégrateurs	40
9	Conclusion	41
A	Quaternions	42
B	Méthode des trièdres	46
B.1	formalisme matriciel	46
B.2	formalisme des quaternions	46
C	Recherche de vecteurs orthogonaux	50
D	Rotation transformant un vecteur en un autre	51
E	Rotation d'axe et d'angle donnés	54
F	Intersection de cônes sur la sphère	55
G	Dérivation automatique	58
G.1	besoins	58
G.2	principe	58
G.3	formulation	58
G.3.1	initialisation	58
G.3.2	opérations arithmétiques	59
G.3.3	composition par une fonction scalaire	59
G.3.4	règles particulières	60
G.3.5	implantation informatique	60

H	Recherche des zéros d'une fonction	62
H.1	besoins	62
H.2	principes	62
H.2.1	encadrement des racines	62
H.2.2	raffinage	63
H.3	algorithme	65
H.3.1	séparation des zéros	65
H.3.2	recherche d'extremum	69
H.3.3	raffinage des zéros	71
I	rotations de Cardan	73
I.1	ordre : lacet, puis roulis, puis tangage	73
I.2	ordre : lacet, puis tangage, puis roulis	74
I.3	ordre : roulis, puis lacet, puis tangage	75
I.4	ordre : roulis, puis tangage, puis lacet	76
I.5	ordre : tangage, puis lacet, puis roulis	76
I.6	ordre : tangage, puis roulis, puis lacet	77

GLOSSAIRE

angles de **CARDAN**

Les angles de **CARDAN** permettent de définir une rotation quelconque en dimension 3 par la composition de trois rotations successives autour des trois axes canoniques dans un ordre qui varie selon les conventions : \vec{i} puis \vec{j} puis \vec{k} , ou \vec{k} puis \vec{i} puis \vec{j} , ou n'importe quel autre ordre.

angles d'**EULER**

Les angles d'**EULER** permettent de définir une rotation quelconque en dimension 3 par la composition de trois rotations successives autour de deux axes canoniques, la première et la troisième rotation se faisant autour d'un axe et la seconde rotation se faisant autour de l'autre axe. La convention la plus courante consiste à utiliser les axes \vec{k} et \vec{i} dans l'ordre : \vec{k} puis \vec{i} puis \vec{k} .

AOCS

Attitude and Orbit Control System

barbecue

Mode de pilotage consistant à pointer un axe satellite sur le Soleil et à tourner à vitesse constante autour de cet axe. Ce mode est un mode d'attente sûr pour le satellite, il est souvent utilisé comme mode de survie ou comme mode d'attente hors mission, par exemple pendant les phases de mise à poste.

CANTOR

Composants et Algorithmes Numériques Traduits sous forme d'Objets Réutilisables, bibliothèque mathématique utilisée par **MARMOTTES**

capteur

Équipement bord permettant à un satellite de mesurer son mouvement autour de son centre de gravité

mouvement de *coning*

Mode de pilotage consistant à faire parcourir à un axe satellite un cône autour du Soleil à une certaine vitesse angulaire et à tourner à la même vitesse autour de cet axe. Ce mode est très souvent utilisé pour la recherche de la Terre lors du passage d'un pointage Soleil à un pointage Terre.

MARMOTTES

Modélisation d'Attitude par Récupération des Mesures d'Orientations pour Tout Type d'Engin Spatial, bibliothèque de simulation de l'orientation des satellite

SCAO

Système de Contrôle d'Attitude et d'Orbite

DOCUMENTS DE RÉFÉRENCE

- [DR1] *MARMOTTES – manuel d'utilisation*
ESPACE/MS/CHOPE/MARMOTTES/MU/001 , édition 5.0, 01/02/2002
- [DR2] *Manuel d'utilisation de la bibliothèque CANTOR*
ESPACE/MS/CHOPE/CANTOR/MU/001 , édition 5.7, 31/01/2002
- [DR3] *Manuel d'utilisation de la bibliothèque CLUB*
SCS/CLUB/MU/2000-001 , édition 6.4, 22/06/2001
- [DR4] *Cours de technologie spatiale*
techniques et technologies des véhicules spatiaux
CNES, CEPADUES éditions, ISBN : 285428476.3 (vol. 1), 285428477.1 (vol. 2), 285428478.X (vol. 3),
Octobre 1998
- [DR5] *Iterative Optimal Orthogonalization of the Strap down Matrix*
ITZHACK Y. BAR-ITZHACK, IEEE Transactions on Aerospace and Electronic Systems – Vol AES-11,
January 1975
- [DR6] *Racines d'une équation par une méthode hybride – méthode de Brent*
J.-C. BERGÈS, DTS/MPI/MS/MN/98-092 , édition 1.0, 27/11/1998

DOCUMENTS APPLICABLES

Néant

1 Introduction

Dans toutes les phases de la mise à poste d'un satellite, de l'analyse de mission aux opérations, on est amené à tenir compte de l'orientation du satellite (pour la visibilité radio ou pour l'orientation d'un moteur d'apogée par exemple). La modélisation de l'orientation d'un satellite trois axes pose alors des problèmes nombreux, puisqu'un même satellite a plusieurs modes de pilotage que l'on modélise généralement par des algorithmes spécifiques.

Cette note décrit une méthode très générale de modélisation utilisée opérationnellement avec succès depuis plusieurs années pour les mises à postes de satellites géostationnaires ainsi que pour des analyses de missions, en orbite basse aussi bien qu'en interplanétaire. Cette modélisation permet de simuler au sol l'évolution de l'orientation des satellites. Au-delà de la simulation, de nombreux aspects liés à l'orientation ne sont pas couverts par cette méthode. Ainsi la restitution d'attitude nécessite d'ajouter du filtrage et d'identifier les paramètres figés et les paramètres libres pour les lois d'évolution, l'étude du contrôle d'attitude en lui-même nécessite la modélisation des actuateurs, enfin de nombreuses analyses nécessitent de réaliser des simulations multiples pour en extraire des résultats statistiques. La méthode présentée n'a pour vocation que de servir de base à de telles études ou développements.

Toutes les orientations pilotées par trois capteurs mesurant des angles ou des vitesses angulaires peuvent être modélisées, y compris dans les cas de contrôle hybride comme par exemple l'utilisation simultanée de deux gyromètres et d'un capteur optique. Ceci couvre directement tous les modes asservis de toutes les plateformes rencontrées par l'auteur sur de nombreux projets dans une large gamme d'orbites.

Certaines lois d'orientation font intervenir des principes plus complexes comme des rendez-vous sur une plage temporelle étendue ou des contraintes de rattachement entre modes successifs, des modes pilotés en aveugle (c'est-à-dire sans retour d'information de capteurs sur certains axes), voire même des modes non pilotés (le mode *caillou* de certaines plateformes). Ces lois ne sont pas modélisables d'une façon directe par les méthodes décrites ici. Ces méthodes peuvent cependant apporter une certaine aide en étant encapsulées au sein d'algorithmes plus spécifiques qui élaborent des consignes pour des pseudo-capteurs dédiés et leur délèguent ensuite le détail de la détermination d'orientation.

La précision de la simulation n'est limitée que par la précision des paramètres d'entrée. Si les consignes correspondent à des valeurs théoriques, l'erreur de simulation correspond à la non modélisation de l'erreur de pilotage, qui dépend entièrement du système bord de contrôle d'orientation. Si les consignes correspondent à des valeurs mesurées, l'erreur de simulation correspond à l'erreur de mesure qui dépend entièrement des capteurs. Pour réduire ces erreurs, il est nécessaire d'améliorer les paramètres d'entrée, par exemple en modélisant la dynamique ou en réalisant un filtrage sur une plage de temps ou sur un grand nombre de capteurs.

Outre leur large couverture, les algorithmes présentés apportent une très grande facilité de description des lois d'orientation à l'ingénieur. Ceci allège grandement le travail de modélisation et offre la possibilité de créer des logiciels réellement multi-missions ayant une représentation fidèle de l'orientation réelle. De nombreux avantages en résultent :

- **fiabilité** : les logiciels validés pour une plateforme le sont pour toutes ;
- **robustesse** : les cas dégradés imprévus peuvent être modélisés très rapidement pendant les opérations et sans modification de code ;
- **rapidité** : aucun développement n'est nécessaire pour implanter une loi d'orientation spécifique ;

- **représentativité** : le raffinement progressif des modèles sous-jacents permis par le potentiel de réutilisation a conduit à la prise en compte de nombreuses notions souvent négligées ;
- **répartition des compétences** : il n'est plus nécessaire de maîtriser l'ensemble des techniques liées à l'orientation pour faire des simulations fines ;
- **retour sur investissement** : les connaissances acquises portant sur des principes d'un assez haut niveau d'abstraction sont réutilisables d'un projet à l'autre.

En termes de compréhension, les principes de description d'orientation s'abstraient de la représentation mathématique. L'utilisateur final définit implicitement l'orientation par ses conséquences et n'a aucun besoin de comprendre les mécanismes sous-jacents qui vont élaborer automatiquement le modèle mathématique correspondant et le résoudre. Il ne voit que des *variables d'ingénierie* qui correspondent à ses vraies préoccupations. Ces variables correspondent d'ailleurs très souvent aux télécommandes et aux télémessures de la plateforme.

L'ensemble des algorithmes décrits dans cette note a été implémenté dans une bibliothèque qui a été validée opérationnellement depuis de nombreuses années : MARMOTTES. Cette bibliothèque est mise gratuitement à la disposition de quiconque par le CNES sur son serveur de logiciels libres¹, que ce soit pour une simple utilisation ou pour l'incorporer à des produits commerciaux (des centres de contrôle par exemple). Cette bibliothèque peut être appelée depuis des programmes en FORTRAN 77, FORTRAN 90, C (ou tout langage utilisant les mêmes règles d'édition de liens et de passages de paramètres) et C++. Des interfaces JAVA et CORBA sont envisagées. La bibliothèque est elle-même écrite en C++.

Cette note correspond à une refonte de l'ancienne documentation mathématique de la bibliothèque MARMOTTES. Elle est orientée vers les algorithmes et la méthode de résolution. Pour utiliser effectivement l'implémentation de cette méthode qui a été faite au CNES, le lecteur est invité à regarder en complément les manuels d'utilisation de la bibliothèque MARMOTTES et des bibliothèques sous-jacentes CANTOR et CLUB ([DR1], [DR2] et [DR3]). Certains détails relatifs à cette implémentation transparaissent malgré tout ici et là. Ces détails ont pour seul objectif d'attirer l'attention sur des problèmes pratiques que les équations ne montrent pas toujours comme les problèmes de singularités numériques, de précision et de performances. Ces détails peuvent être omis par les lecteurs qui ne désirent ni réaliser leur propre implémentation ni modifier la bibliothèque publique.

L'ensemble des algorithmes, la conception et la réalisation de la bibliothèque sont issus du travail de l'auteur. De nombreuses personnes ont également contribué à d'autres aspects de MARMOTTES. Geneviève CAMPAN (CNES) a soutenu le projet dès son origine et lui a donné pendant de nombreuses années les moyens d'être mené à bien. Hélène COTTET-GAUTIER (CNES) a vérifié quasiment toutes les versions de toutes les démonstrations et a contribué à nombre de réflexions concernant l'implémentation, en particulier pour la modélisation des champs de vue. François DESCLAUX (CNES) a été le premier utilisateur, il a testé de nombreuses implémentations de ces algorithmes et a contribué à des choix stratégiques et a beaucoup collaboré à la spécification des besoins. Géraldine FILAIRE (CNES) a aidé à la validation des deux grandes refontes récentes concernant l'élimination des singularités des modèles géométriques et l'introduction des informations topologiques dans le calcul des frontières de champs de vue, elle est également intervenue sur la bibliothèque elle-même, en réalisant l'implémentation complète des modèles non singuliers et en participant à la maintenance évolutive. Enfin les équipes de mise à poste de satellites géostationnaires ont validé opérationnellement l'ensemble des algorithmes et du code et ont contribué à faire de MARMOTTES un produit opérationnel fiable, elles contribuent régulièrement à la détection des bogues et à la spécification de nouvelles fonctionnalités et de nouveaux capteurs.

¹<http://logiciels.cnes.fr>

2 Description d'orientation

Les principes de description d'orientation adoptés résultent d'une inversion du problème de l'orientation.

2.1 description classique

Très souvent, on modélise l'orientation du satellite par des objets mathématiques d'assez bas niveau tels que des quaternions, des angles d'EULER ou plus souvent de CARDAN(cf. [DR4]) ou des matrices de rotations. On utilise ensuite ces objets pour convertir des directions d'un repère satellite à un repère inertiel ou réciproquement.

Cette modélisation classique engendre de nombreuses difficultés. La première est qu'elle n'est pas canonique. Prenons un exemple : le quaternion résultant de la simulation d'un simple pointage Terre parcourt une boucle fermée dans un espace en quatre dimensions au cours de l'évolution du satellite sur son orbite. Cette courbe ne peut pas être traitée manuellement et les éventuels biais n'y apparaissent pas naturellement. Pourtant, la spécification d'un pointage Terre se résume à quelques mots et éventuellement une ou deux valeurs numériques de biais. La description par quaternion introduit une explosion de la complexité.

Dans ce cas particulier, les angles de CARDAN sont bien mieux adaptés, mais leur utilité s'évanouit pour tous les autres modes de contrôle même les plus simples comme le mode *barbecue*. Leur utilisation est inextricable pour des modes comme le *coning* ou les manœuvres d'apogée par pilotage hybride géométrique/cinématique pour lesquels aucun choix de repère de base ne permet de simplifier les évolutions des angles de base.

La seconde difficulté est l'opacité des données manipulées. En reprenant notre exemple, rien ne permet de distinguer la courbe d'un pointage sans biais d'une courbe avec biais ou d'un asservissement en lacet² sur le Soleil et encore moins de retrouver ce biais. Il faut réaliser des calculs pour extraire cette information.

Une troisième difficulté est que la description est fortement dépendante de la plateforme. En effet, la plupart du temps les caractéristiques du Système de Contrôle d'Attitude et d'Orbite et les besoins de pointage sont utilisés pour calculer comment passer d'une orientation théorique de référence à l'orientation cherchée. On déduit de la séquence de rotations le quaternion, les angles ou la matrice d'orientation. Les algorithmes réalisant ces opérations dépendent d'hypothèses liées au mode de pointage utilisé et de la façon dont ils sont implémentés dans la plateforme.

À elle seule, cette dépendance limite fortement les possibilités de ce mode de description. Les algorithmes liés à la plateforme ne peuvent pas être réutilisés sur une autre. Chacun des modes devant être simulé engendre un développement spécifique, même pour un seul satellite. Seuls les cas dégradés ayant été identifiés à l'avance peuvent être pris en compte.

L'obligation de redévelopper des modèles spécifiques conduit souvent à se contenter d'une modélisation assez théorique et très simple, dans laquelle de nombreux paramètres sont omis. On voit rarement les éclipses prises en compte dans les simulations de pointage solaire ou les inhibitions des capteurs Terre modélisées selon les données constructeur. Les mésalignements de capteurs, d'antennes et d'actuateurs sont également régulièrement oubliés. Les logiciels sont souvent prévus pour fonctionner avec un seul mode dont le modèle est enfoui dans le code. Dès que l'on change une caractéristique qui n'a pas été prévue, il faut tout refaire et tout revalider, même pour des modes très proches. Un exemple classique ayant déjà été rencontré consiste à modifier l'ordre

²*yaw steering*

dans lequel les rotations doivent être appliquées ; on a même vu un satellite unique utiliser simultanément deux conventions différentes, son simulateur en utilisant bien sûr une troisième. Un autre exemple consiste simplement à permuter les axes satellites, ce qui est classique lorsque l'on passe d'un constructeur à l'autre. Le plus courant reste cependant le simple changement d'un mode.

2.2 inversion du problème

Les difficultés énoncées plus haut sont essentiellement des problèmes liés au couplage fort entre la modélisation mathématique et la description de l'orientation. Il est possible de changer de point de vue et de séparer ces deux aspects.

Considérons une boucle d'asservissement dans un Système de Contrôle d'Attitude et d'Orbite. Elle comprend des capteurs (capteur Terre, solaire ou stellaire, gyromètres) et des actionneurs (magnétocoupleurs, tuyères, roues). Les mesures réalisées par les capteurs peuvent être considérées comme résultant des modifications d'orientation engendrées par les actionneurs et les couples naturels. C'est cette vision qui est sous-jacente à la description classique. Inverser le problème revient à se placer du côté de la logique de l'asservissement : l'orientation est cette fois considérée comme le résultat de la convergence de la boucle, c'est donc elle qui découle des mesures réalisées par les capteurs, ou plus exactement des consignes imposées aux capteurs à travers l'asservissement. C'est cette vision inversée qui est sous-jacente à nos modèles.

L'orientation étant un système à trois degrés de liberté, trois valeurs numériques sont nécessaires à chaque instant, correspondant aux mesures souhaitées sur trois capteurs. L'utilisateur final n'a aucun besoin de savoir comment ces consignes sont utilisées pour calculer l'orientation, il ne voit que les consignes elles-mêmes et les capteurs. Quelques exemples classiques vont permettre d'éclairer notre propos.

En préalable aux exemples de modélisation, il est important de mentionner certaines habitudes de langage concernant les dénominations de capteurs. Les capteurs mesurant des angles autour d'axes proches de l'axe \vec{X} du satellite sont généralement qualifiés de capteurs de roulis, les capteurs mesurant des angles autour d'axes proches de l'axe \vec{Y} sont de même qualifiés de capteurs de tangage, et les capteurs mesurant des angles autour d'axes proches de l'axe \vec{Z} sont qualifiés de capteurs de lacet. Ceci ne signifie pas que ces capteurs mesurent des angles de CARDAN à partir d'un repère de référence inertiel, ils peuvent très bien mesurer par exemple un angle dièdre entre le plan (X, Z) et le plan (X, Soleil) , on parlera dans ce cas d'un capteur de roulis Soleil. Dans la pratique, il n'existe pas une définition unique et reconnue des angles de roulis, tangage et lacet, pas même dans le cadre des capteurs de CARDAN (voir l'annexe I, page 73). Ces habitudes de langage ne font qu'accentuer cette multiplicité. Ces conventions sont très classiques, elles se retrouvent en particulier dans les documents des constructeurs et dans les noms des paramètres de télémétrie générés par ces capteurs.

Un autre terme utilisé dans cette note est *pseudo-capteur*. Ce terme désigne les capteurs modélisés dans les méthodes décrites plus loin mais qui ne correspondent pas à des capteurs physiques existants. Ces pseudo-capteurs sont souvent utilisés pour faciliter la modélisation. Des exemples typiques sont les pseudo-capteurs d'ascension droite et de déclinaison ainsi que les pseudo-capteurs de CARDAN, très peu de satellites disposant de tels moyens de mesure³.

Les exemples suivants utilisent ces notions de pseudo-capteurs et de capteurs de roulis, tangage ou lacet ; ils correspondent à des modes classiques rencontrés dans plusieurs missions.

³quand les valeurs des angles de CARDAN sont disponibles dans la télémétrie, il s'agit d'une valeur élaborée à bord sur la base de plusieurs mesures élémentaires et d'un modèle d'orbite

pointage Terre avec biais : ce mode très courant en maintien à poste se décrit ainsi :

- mesure du capteur roulis Terre : 0°
- mesure du capteur tangage Terre : $+10^\circ$
- mesure du pseudo-capteur angle de CARDAN en lacet : 0°

asservissement en lacet sur le Soleil : ce mode est traditionnel sur les satellites en orbite basse ; on le désigne souvent par sa dénomination anglo-saxonne, *yaw steering*. Sa modélisation est l'une des plus simples qui soient :

- mesure du capteur roulis Terre : 0°
- mesure du capteur tangage Terre : 0°
- mesure du capteur lacet Soleil : 0°

On peut facilement introduire des biais dans l'asservissement en lacet : il suffit de changer une constante par une autre.

mouvement de *coning* : ce mode est essentiellement rencontré dans les phases de recherche Terre permettant de passer d'un pointage solaire à un pointage Terre ; il fait intervenir un pilotage hybride avec des capteurs optiques et un gyromètre :

- mesure du capteur tangage Soleil : 15°
- mesure du capteur lacet Soleil : 0°
- mesure du capteur vitesse roulis : $0,25^\circ/s$

petits mouvements autour d'une orientation de référence : le cas d'utilisation typique de ce mode consiste à observer à l'aide des paramètres de télémétrie l'orientation réelle du satellite autour d'un pointage théorique désiré (par exemple un pointage Terre). On peut utiliser directement les angles de CARDAN pour cela :

- mesure du pseudo-capteur roulis : $0,1^\circ$
- mesure du pseudo-capteur tangage : $-0,07^\circ$
- mesure du pseudo-capteur lacet : $0,2^\circ$

En fait, cette utilisation des angles de CARDAN n'est pas limitée aux petits angles. Elle perd cependant son côté didactique lorsque les angles grandissent (il faut alors bien se rappeler l'ordre des rotations, sachant qu'aucune convention n'est universelle). L'utilisateur imprudent risque par ailleurs de tomber sur les singularités inhérentes à ces angles s'il n'y prend pas garde.

Cette description très simple est facile à appréhender. Penser en termes de mesures sur les capteurs est une abstraction de plus haut niveau qu'en terme d'objet mathématique interne. Cette description est à l'usage beaucoup plus parlante que les descriptions classiques. Elle compense également les difficultés évoquées plus haut. Elle est canonique et les données manipulées sont relativement parlantes, même s'il faut parfois quelques temps pour acquérir des automatismes.

Cette description très simple est la seule chose qui soit nécessaire aux algorithmes présentés plus loin pour calculer l'orientation. Une bibliothèque a été conçue de sorte que ces algorithmes soient disponibles en boîte noire, l'utilisateur n'a donc effectivement que les noms des capteurs et les valeurs de consignes à traiter dans son code. En remontant ces noms et ces valeurs jusqu'au niveau des données d'entrées, il peut concevoir des logiciels multi-mission. La description des capteurs eux-mêmes, purement statique, est faite dans des fichiers de configuration lus directement par la bibliothèque.

La description de l'orientation par les consignes de capteurs et la bibliothèque implémentant les algorithmes de résolution à partir de cette description permettent donc de lever la dernière difficulté énoncée plus haut et d'obtenir l'indépendance par rapport à la plateforme et aux lois d'orientation.

On pourrait craindre que cette description ne soit bien adapté qu'aux quelques exemples cités ci-dessus. En sept ans d'utilisation dans divers projets, aucune limitation résultant de ce choix n'a cependant été rencontrée. De temps à autre, un projet a eu besoin d'un capteur pour lequel aucun modèle mathématique n'était disponible dans la bibliothèque où sont implémentés ces algorithmes. Le modèle du capteur a alors simplement été ajouté et rien d'autre n'a changé⁴.

Un autre facteur jouant en faveur de l'adaptabilité de la méthode est qu'outre les capteurs réels du satellite, on peut modéliser des pseudo-capteurs dans le seul but de les utiliser ensuite pour décrire des lois particulières. Ce facteur n'avait pas été prévu lors de l'élaboration des modèles, il s'est imposé de lui même au cours des divers projets. L'expérience passée a également montré que les lois d'orientation que l'on souhaite modéliser utilisaient souvent des consignes constantes dans le temps, les constantes étant d'ailleurs généralement nulles. Il faut cependant noter que ceci n'est pas vrai pour les missions d'observation de la Terre.

2.3 conséquences pour l'utilisateur

Pour l'utilisateur, la description de l'orientation fait intervenir deux types d'informations différents : la description générale des capteurs et les descriptions d'orientation par association de trois couples capteur/consigne.

Vu du côté de la simulation d'orientation, un capteur est une structure de données statique dont les composantes dépendent du type de capteur. À titre indicatif, on peut citer pour un capteur optique l'orientation par rapport au satellite, la cible (Terre, Soleil, ...), le champ de vue et les axes sensible, neutre et de visée. L'utilisateur devra définir chacun des capteurs qu'il désire faire intervenir dans ses simulations ; il ne s'agit là que d'écrire des fichiers de données, bien plus faciles à valider que du code.

Les capteurs sont souvent similaires (les capteurs solaires sont par exemple systématiquement des capteurs d'angle dièdre avec un champ de vue symétrique en double dièdre). Passer d'un satellite à l'autre se limite généralement à changer les noms des capteurs et les valeurs numériques des angles d'ouverture des dièdres et à prendre en compte les données d'alignement lorsqu'elles sont mises à jour par le constructeur. Cela arrive par exemple lorsque ce dernier réalise les mesures d'alignement au cours de l'intégration du satellite ou lorsque des capteurs solaires ont des calages différents pour des lancements en été ou en hiver.

Les couples capteur/consigne dépendent quant à eux de la phase simulée. C'est essentiellement à ce niveau que les spécificités de la mission apparaissent. Les algorithmes sous-jacents ne font aucune hypothèse sur les paramètres de description qu'on leur soumet ni sur leur succession. Si un satellite a besoin d'une simulation vraiment très spécifique, rien n'empêche de créer un logiciel pour cela qui calculera une séquence de consignes propre et délèguera la résolution aux méthodes génériques. Ceci simplifie beaucoup le programme par rapport à une simulation spécifique de bout en bout. Ce cas ne se rencontre guère que dans le cadre des missions d'observation de la Terre.

En présence de cas dégradés, le fait que toutes les descriptions soient au niveau des données permet à l'utilisateur d'intervenir rapidement et avec des risques limités. On peut par exemple faire fonctionner un logiciel conçu *a priori* pour utiliser un capteur en utilisant un autre en cas de panne du premier et de basculement sur le redondant. On peut également faire une simulation en saisissant manuellement dans un logiciel les paramètres lus sur les stations de décommutation de la télémesure pour essayer de comprendre comment se comporte le

⁴l'implémentation a été conçue selon les techniques objets avec une séparation nette entre la modélisation des capteurs et les autres composants

satellite. On peut enfin toujours intervenir sur l'ensemble des paramètres de description des capteurs qui sont *tous* accessibles dans des fichiers⁵.

2.4 conséquences pour l'implémentation

Le partage net entre les parties description et résolution ouvre la voie à une implémentation très modulaire. La réduction de la taille des composants qui résulte de cette modularité et le potentiel de réutilisation permettent d'implanter des modèles très fins et de les améliorer progressivement. L'historique de la modélisation des capteurs Terre montre clairement les gains de cette approche.

À l'origine, le modèle des capteurs Terre se contentait de réaliser une mesure d'angle dièdre et de vérifier la visibilité du limbe Terre dans le champ de vue, en tenant compte d'une logique booléenne entre les différentes zones de balayage. Après une première tentative maintenant abandonnée de modéliser un capteur spécifique à un constructeur particulier, le modèle a pris en compte la notion de champ d'inhibition pour la Lune et le Soleil. Très vite, le champ d'inhibition général a été remplacé par des champs dédiés à chaque astre. Le calcul des inhibitions a ensuite pris en compte les éclipses entre les différents corps, ce qui a permis de jouer à cache-cache avec la Lune pendant certaines opérations critiques lors d'une mise à poste. Une notion de seuil d'inhibition a été ajoutée pour la Lune, qui n'est réellement gênante qu'aux alentours de la pleine Lune, ce seuil a pendant longtemps été exprimé en terme de luminance, mais des difficultés d'homogénéisation avec les modèles constructeurs ont conduit à exprimer ce seuil en fonction de l'angle de phase de la Lune.

Cet exemple montre que l'on peut introduire des notions très fines dans les modèles, notions dont bénéficieront tous les utilisateurs ultérieurs.

Une seconde conséquence sur l'implémentation est qu'il faut prendre garde à ne pas utiliser les spécificités de tel ou tel capteur dans le cœur algorithmique, mais prévoir des pré ou post traitements gérés au niveau des capteurs (par exemple les calculs d'inhibition par les capteurs Terre).

2.5 limitations du modèle

Les exemples de consignes donnés ci-dessus montrent le principe général de la simulation : on donne les valeurs que doivent mesurer les capteurs lorsque le satellite est dans l'orientation désirée. Le Système de Contrôle d'Attitude et d'Orbite étant justement asservi par des capteurs, ceci correspond à une simulation d'un pilotage parfait, c'est-à-dire en supposant que les actionneurs (tuyères, roues) qui ne sont pas modélisés contrôlent exactement la dynamique du satellite (inerties, couples).

Les algorithmes génériques ne modélisent que la cinématique du pilotage (chaque détermination d'orientation fournit à la fois l'orientation et la rotation moyenne depuis l'état précédent). Il est cependant possible d'inclure la dynamique, par des calculs faits soit avant l'appel à la résolution (transformation des couples en consignes de vitesse sur les capteurs cinématiques à l'aide des données d'inertie), soit après l'appel (transformation de l'évolution d'orientation en couples pour déduire l'activité tuyères et roues).

Un point important à souligner est que les consignes sont considérées comme étant exprimées conformément aux conventions du capteur, et non conformément aux conventions du Système de Contrôle d'Attitude

⁵tout au moins dans l'implémentation de la méthode réalisée au CNES

et d'Orbite, en particulier en ce qui concerne le signe des mesures. Ce choix permet d'alimenter directement la bibliothèque avec le flux de télémessure pendant les opérations, sans faire intervenir de filtre. Les consignes effectivement chargées dans le Système de Contrôle d'Attitude et d'Orbite par télécommandes peuvent donc différer des consignes chargées dans la bibliothèque. Ainsi certaines plateformes adoptent la convention selon laquelle pour avoir $+15^\circ$ sur le capteur Terre, on charge -15° comme biais de consigne satellite par télécommande. Dans ce cas, la bibliothèque utilisera une valeur de consigne de $+15^\circ$ cohérente avec la mesure, pas la valeur -15° compatible avec la télécommande.

3 Capteurs

La description d'orientation présentée plus haut a besoin d'une description statique des caractéristiques de tous les capteurs utilisés.

De nombreux capteurs différents sont utilisés sur les satellites. Il s'avère cependant que l'on peut identifier deux grandes catégories, les capteurs géométriques et les capteurs cinématiques. Nous allons voir qu'il est possible d'établir un modèle général des mesures pour chacune de ces catégories, et qu'il sera possible de déterminer l'orientation en se contentant de l'information disponible dans ces modèles généraux.

3.1 capteurs géométriques

On appelle ici capteur géométrique tout capteur qui mesure la direction d'une cible définie en repère inertiel (Terre, Soleil, ...) par rapport à un repère du capteur fixe par rapport au satellite, ou au contraire qui mesure une direction de référence liée au satellite par rapport à un repère inertiel (ascension droite ou déclinaison d'un axe satellite par exemple). La mesure est généralement un angle dièdre (c'est-à-dire un angle entre plans comme

FIG. 1 – capteur solaire d'angle dièdre

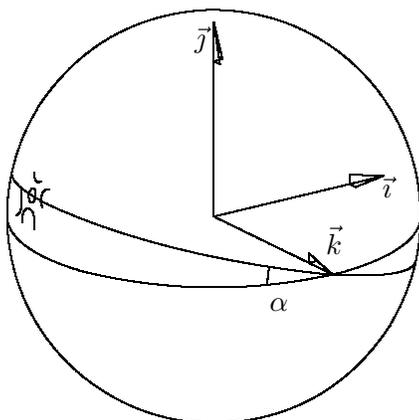
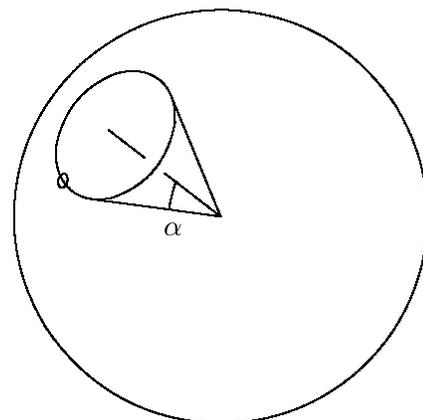


FIG. 2 – capteur d'aspect solaire



indiqué sur la figure 1) mais peut également être un angle par rapport à un axe (voir figure 2).

Les rotations de CARDAN à partir d'un repère de référence (par exemple un repère orbital local) entrent également dans la catégorie des capteurs géométriques, et l'on peut démontrer qu'ils correspondent soit à des angles dièdres soit à des angles par rapport à un axe (cf annexe I, page 73).

Si un capteur mesure un angle dièdre défini en partie par une cible inertielle (cas de la figure 1, où la cible est le Soleil), imposer une consigne consiste à obliger la cible inertielle observée à rester confinée dans un plan défini en repère satellite.

Si un capteur mesure un angle dièdre défini en partie par une direction de référence satellite (par exemple l'axe \vec{X} satellite, dont on cherche à calculer l'ascension droite à l'aide d'un pseudo-capteur d'ascension droite), imposer une consigne consiste à obliger un vecteur satellite à rester confiné dans un plan inertiel. Cette contrainte peut être retournée en remarquant qu'elle est équivalente à obliger le vecteur inertiel normal au plan inertiel à rester confiné dans le plan satellite normal à la direction de référence satellite.

Si un capteur mesure un angle entre un vecteur inertiel et un vecteur satellite, imposer la consigne consiste à obliger le vecteur inertiel à rester confiné sur un cône.

Tous ces cas peuvent être réunis en un seul si on remarque qu'un plan est un cône particulier, de demi-angle au sommet $\frac{\pi}{2}$. D'une façon générale, on modélisera une consigne de capteur géométrique par le cône de confinement d'un vecteur inertiel, cône d'axe \vec{a} défini en repère satellite et de demi-angle au sommet μ .

La modélisation adoptée suppose implicitement que chaque point du cône de confinement respecte la consigne. Dans la pratique, certains points de ce cône peuvent tomber hors du champ de vue du capteur, et devront donc être éliminés *a posteriori* (ainsi sur la figure 1 le cône de confinement du Soleil passe par $\pm\vec{k}$ et légèrement au dessous de $+\vec{i}$, alors que le champ de vue peut être limité aux alentours de $-\vec{i}$).

Si toutes les solutions sont rejetées au cours de ce post-traitement, cela signifie que l'orientation désirée n'était pas pilotable. *A contrario* il peut subsister plusieurs solutions après ce filtrage dans les cas d'utilisation de pseudo-capteurs n'ayant pas de notion de champ de vue (par exemple une mesure de l'ascension droite de la direction de poussée du moteur d'apogée). La détermination d'orientation sélectionne alors la solution *la plus proche* de l'état que l'on peut déduire d'une évolution naturelle⁶ de l'état précédent.

3.2 capteurs cinématiques

On appelle ici capteur cinématique tout capteur qui mesure le *mouvement* du satellite autour de son centre de gravité. Ces capteurs mesurent soit une vitesse instantanée autour d'un axe, soit l'intégrale au cours du temps d'une vitesse instantanée.

Il faut ici prendre garde à une subtilité : cette intégrale, même si elle est homogène à un angle, ne correspond pas forcément à un angle géométrique⁷. En effet, si l'axe autour duquel on mesure les rotations se déplace, l'intégrale correspond à une sommation d'angles non coplanaires. Ceci montre que les gyromètres intégrateurs ne peuvent pas être assimilés à des capteurs géométriques qui mesureraient un écart entre une direction de référence mémorisée et le satellite : il s'agit bien de capteurs cinématiques.

Une consigne d'orientation pour ce type de capteur consiste à imposer soit la vitesse mesurée, soit l'angle intégré. En fait, imposer l'angle revient à imposer l'intégrale de la vitesse depuis le moment de l'initialisation du zéro du gyromètre intégrateur, réalisée après l'étalonnage. Imposer l'intégrale revient à imposer la vitesse instantanée sur toute une plage de temps. Ces deux types de consigne sont donc équivalents puisque de toute

⁶l'évolution est calculée par extrapolation de la vitesse de rotation précédente en partant de l'orientation précédente, toutes ces informations étant fournies à chaque résolution

⁷on entend par angle géométrique un angle mesurable à un instant donné entre deux objets, vecteurs ou plans, à l'aide d'un compas par exemple, et indépendamment de toute référence à un état précédent

façon dans ce type de capteur, on est obligé de suivre pas à pas l'évolution de l'orientation qui dépend à chaque instant de la loi, mais également de l'orientation à l'instant précédent. On modélisera donc ces capteurs par l'axe (en repère satellite) autour duquel ils réalisent leurs mesures, et la consigne à chaque pas sera la vitesse instantanée projetée sur cet axe. Il s'agit souvent d'une consigne de vitesse constante, la constante étant non nulle en mode stabilisé par rotation, nulle en mode trois axes.

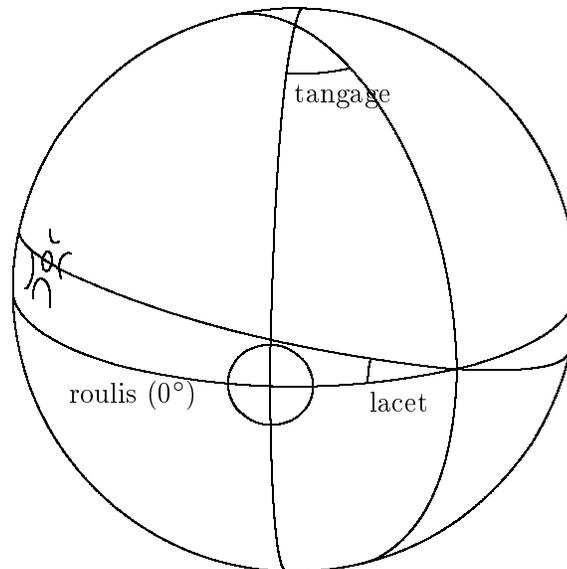
4 Lois de pilotage

Une loi de pilotage est le rassemblement à une date donnée de trois consignes, ce qui permet de fixer les trois degrés de liberté d'un solide autour de son centre de gravité. Plusieurs types de rassemblement sont possibles :

4.1 contrôle par trois capteurs géométriques

Ce mode de pilotage correspond par exemple à une manœuvre d'apogée géocentrique ou à un asservissement en lacet sur le Soleil. Dans ces deux cas une consigne en roulis est donnée au capteur Terre (souvent roulis = 0°), une consigne en tangage est donnée au capteur Terre, et une consigne en lacet est donnée au capteur solaire (voir figure 3).

FIG. 3 – contrôle par trois capteurs géométriques



Avec le modèle que nous nous sommes donnés, trouver une orientation qui respecte les trois consignes revient à trouver le changement de repères qui permet de passer du repère inertiel au repère satellite de telle façon que les trois cibles (Terre, Terre et Soleil dans notre exemple) soient sur les trois cônes de consigne tels que définis au paragraphe 3.1.

On peut exprimer mathématiquement ce changement de repères : soient \vec{v}_1 (Terre), \vec{v}_2 (Terre), \vec{v}_3 (Soleil) les vecteurs cibles connus en repère inertiel et $\vec{u}_1, \vec{u}_2, \vec{u}_3$ leurs projections en repère satellite. Le changement de

repères désiré est celui qui permet de passer d'un triplet de vecteurs à l'autre. Pour pouvoir l'exprimer, il faut trouver les vecteurs \vec{u}_i sachant que \vec{u}_1 doit appartenir au cône (\vec{a}_1, μ_1) d'axe \vec{a}_1 et de demi-angle au sommet μ_1 défini par la mesure désirée pour le premier capteur, que \vec{u}_2 doit appartenir au cône (\vec{a}_2, μ_2) d'axe \vec{a}_2 et de demi-angle au sommet μ_2 défini par la mesure désirée pour le deuxième capteur, et que \vec{u}_3 doit appartenir au cône (\vec{a}_3, μ_3) d'axe \vec{a}_3 et de demi-angle au sommet μ_3 défini par la mesure désirée pour le troisième capteur. La méthode de résolution est décrite aux sections 6.1 (page 16) et 7.1 (page 28).

4.2 contrôle par deux capteurs géométriques et un capteur cinématique

Ce mode de pilotage se rencontre lors des manœuvres d'apogée géocentriques ou lors des recherches Terre préalables à une acquisition en roulis. Dans ces deux cas une consigne en roulis est donnée au capteur Terre (souvent roulis = 0°), une consigne en tangage est donnée au capteur Terre, et une consigne en lacet est donnée au gyromètre lacet (vitesse de rotation nulle autour de l'axe de lacet pour les manœuvres d'apogée, vitesse non nulle pour la recherche Terre).

Dans les cas où la consigne de vitesse est nulle, $\dot{\theta}_L = 0$ implique $\theta_L = \int \dot{\theta}_L dt + \theta_0 = \theta_0 = cte$. L'angle constant θ_0 n'est cependant pas un angle physiquement mesurable par la séparation de deux points, on ne peut donc pas appliquer la méthode de la section précédente pour déterminer l'orientation. C'est un angle purement mathématique qui est constant, une sorte d'abscisse curviligne mesurée le long d'une courbe due au déplacement de l'axe sensible du gyromètre. Ceci est assez difficile à se représenter dans l'espace, surtout quand l'angle de tangage est non nul : il y a alors des effets de couplage entre le mouvement du satellite sur son orbite inclinée, le mouvement autour de l'axe de roulis pour conserver la Terre dans le plan (X,Z) (roulis = 0°), et le mouvement autour de l'axe de lacet qui doit rester nul malgré tous ces couplages. C'est ce type de modélisation qui a d'ailleurs motivé le développement d'une méthode de résolution générale. Dans les cas où la consigne de vitesse est non nulle, les couplages sont encore plus flagrants.

Avec le modèle que nous nous sommes donné, trouver une orientation qui respecte les trois consignes revient à trouver le changement de repères qui permet de passer du repère inertiel au repère satellite de telle façon que les deux cibles (Terre et Terre dans notre exemple) soient sur les deux cônes de consigne tels que définis au paragraphe 3.1, et que la rotation subie par le satellite depuis le pas précédent conduise à une vitesse de rotation moyenne se projetant sur l'axe sensible du gyromètre conformément à la consigne de vitesse angulaire.

On peut exprimer mathématiquement ce changement de repères : soient \vec{v}_1 (Terre) et \vec{v}_2 (Terre) les vecteurs cibles connus en repère inertiel et \vec{u}_1, \vec{u}_2 leurs projections en repère satellite. Soit $\vec{\Omega}$ le vecteur rotation instantanée correspondant à la rotation subie par le satellite depuis le pas précédent. Le changement de repères désiré est celui qui permet de passer d'un couple de vecteur à l'autre. Pour pouvoir l'exprimer, il faut trouver les vecteurs \vec{u}_i et $\vec{\Omega}$ sachant que \vec{u}_1 doit appartenir au cône (\vec{a}_1, μ_1) d'axe \vec{a}_1 et de demi-angle au sommet μ_1 défini par la mesure désirée pour le premier capteur, que \vec{u}_2 doit appartenir au cône (\vec{a}_2, μ_2) d'axe \vec{a}_2 et de demi-angle au sommet μ_2 défini par la mesure désirée pour le deuxième capteur, et que la projection de $\vec{\Omega}$ sur l'axe sensible du gyromètre doit être égale à la mesure désirée pour le gyromètre (ω_3). La méthode de résolution est décrite aux sections 6.1 (page 16) et 7.2 (page 28).

Une particularité de ce type de pilotage est qu'il dépend de l'état précédent (le capteur cinématique contrôle les changements d'orientation entre deux états du satellite). On ne peut donc calculer une orientation de ce type qu'en intégrant depuis un état initial où l'orientation était connue par d'autres moyens : orientation géométrique avant le changement de mode, ou orientation de début de manœuvre optimisée dans le cas d'une intégration à rebours de la date de début de manœuvre à la date de réorientation, justement pour trouver l'orientation

géométrique d'initialisation qui permettra au satellite d'évoluer *naturellement* jusqu'à la bonne orientation de poussée.

4.3 contrôle par deux capteurs cinématiques et un capteur géométrique

Ce mode de pilotage est typiquement utilisé à la fin des manœuvres de basculement entre deux orientations stables, les gyromètres gérant la rotation du satellite et le capteur géométrique (par exemple un capteur solaire) étant utilisé pour converger vers une direction particulière. Une utilisation un peu plus complexe concerne les phases de recherche Terre en roulis, certains satellites réglant l'ouverture du cône de recherche par la combinaison de deux vitesses angulaires plutôt que par un capteur géométrique.

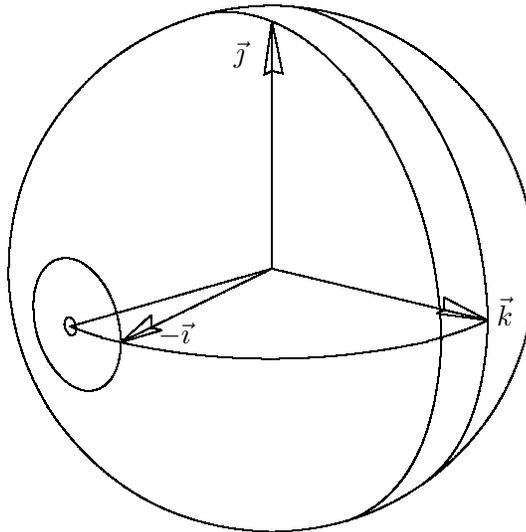
À titre d'exemple, une loi de ce type peut être pilotée par les consignes :

- lacet capteur solaire : 0° ;
- vitesse de roulis : $0,75^\circ/s$;
- vitesse de tangage : $0,2^\circ/s$.

Avec le modèle que nous nous sommes donnés, trouver une orientation qui respecte les trois consignes revient à trouver le changement de repère qui permet de passer du repère inertiel au repère satellite de telle façon que la rotation subie par le satellite depuis le pas précédent soit telle que le vecteur rotation instantanée ait pour projections sur les axes \vec{a}_1 et \vec{a}_2 les valeurs ω_1 et ω_2 ; simultanément, le vecteur \vec{v}_3 connu en repère inertiel, une fois projeté en repère satellite devra appartenir au cône d'axe \vec{a}_3 de demi-angle au sommet μ_3 .

Là encore, l'orientation ne peut se calculer qu'en intégrant depuis une orientation initiale, puisque le capteur cinématique contrôle les changements d'orientation d'un état à un autre du satellite.

FIG. 4 – contrôle par un capteur géométrique et deux capteurs cinématiques



4.4 contrôle par trois capteurs cinématiques

Ce mode de pilotage est utilisé en particulier dans les phases transitoires comme les basculements de mise en orientation de poussée, au cours desquels un gyromètre contrôle la rotation jusqu'à ce qu'un capteur géométrique

reprenne le contrôle lorsque son astre cible entre dans son champ de vue.

Avec le modèle que nous nous sommes donnés, trouver l'orientation qui respecte les trois consignes revient à faire tourner l'orientation du pas précédent autour du vecteur rotation instantanée dont les projections sur les axes $\vec{a}_1, \vec{a}_2, \vec{a}_3$ sont respectivement $\omega_1, \omega_2, \text{ et } \omega_3$. L'orientation se calcule donc par intégration pas à pas depuis une orientation initiale.

5 Démarche de la résolution

Si l'on ne respecte que deux des trois consignes d'orientation, on fixe deux degrés de liberté et le troisième reste libre. Dans la section 6, on démontre que si les deux consignes respectées sont de même type (soit géométrique, soit cinématique), alors on sait exprimer l'orientation sous forme d'une fonction à une variable $Q(\theta)$ et la rotation instantanée sous forme d'une fonction $\vec{\Omega}(\theta)$ (θ étant le degré de liberté restant), et on connaît le domaine de validité de la variable scalaire θ . Les fonctions Q et $\vec{\Omega}$ sont calculables de façon analytique et ont de bonnes propriétés de continuité et de dérivabilité à l'intérieur du domaine de validité de θ . On démontre également dans la section 6 qu'étant donnée une orientation Q , une rotation instantanée $\vec{\Omega}$ et une consigne, on sait calculer de façon analytique une fonction scalaire $f(Q, \vec{\Omega})$ qui a également de bonnes propriétés de continuité et de dérivabilité, et qui s'annule lorsque l'orientation et la rotation instantanée respectent la consigne. La démarche de la résolution est alors la suivante :

- modéliser les mesures capteurs en consignes ;
- organiser les consignes en un groupe de deux consignes identiques et une consigne unique quelconque ;
- construire les fonctions $Q(\theta)$ et $\vec{\Omega}(\theta)$ qui respectent les deux consignes identiques et calculer le domaine de validité de θ ;
- chercher tous les θ_i annulant la fonction $f(Q(\theta), \vec{\Omega}(\theta))$ dans le domaine de validité de θ , la fonction f étant celle qui est associée à la troisième consigne ;
- rejeter toutes les couples orientation et rotation instantanée $(Q(\theta_i), \vec{\Omega}(\theta_i))$ qui ne respectent pas les champs de vue des capteurs géométriques.

Après ces étapes, il ne devrait rester que l'orientation donnant les mesures capteurs fournies en entrée. S'il ne reste aucune orientation, alors l'orientation demandée n'est pas pilotable (en tout cas, pas à cet instant). S'il reste plusieurs orientations, il est vraisemblable que le mode d'orientation décrit soit réellement ambigu et présente des effets stroboscopiques. C'est en particulier le cas lorsqu'on utilise deux capteurs cinématiques et un capteur géométrique, puisqu'outre la solution naturelle il existe des solutions parasites respectant toutes les contraintes mais pour lesquelles le satellite fait un nombre entier de tours sur lui même entre deux pas autour d'un axe auquel les capteurs cinématiques sont insensibles. Dans ce cas la sélection *a posteriori* de la meilleure solution selon un critère de modification minimale de la rotation instantanée depuis l'état précédent lève correctement l'ambiguïté.

La résolution décrite précédemment peut être coûteuse en temps de calcul. Dans la pratique, l'évolution d'orientation des satellites est régulière sur de longues périodes de temps. On améliore donc les performances en tentant une simple extrapolation de l'orientation précédente (à l'aide de la rotation instantanée associée) avant d'entamer la résolution complète. Si cette extrapolation respecte les trois consignes au seuil de convergence désiré (par défaut un dixième de la précision du capteur concerné), il est inutile de lancer la résolution complète, on se contente de retourner l'orientation extrapolée. L'expérience montre que l'on gagne un facteur de l'ordre de 10 à 30 sur le temps de calcul.

6 Orientations respectant deux consignes identiques

La détermination d'orientation se fait en rajoutant les consignes les unes après les autres, pour restreindre petit à petit l'ensemble des solutions possibles.

Il s'avère qu'il faut séparer le cas des deux premières consignes du cas de la troisième. On choisit donc de traiter les deux premières en vérifiant qu'elles sont de même type (c'est toujours possible, au prix d'un éventuel réarrangement). Deux possibilités se présentent alors : les deux consignes sont soit toutes deux géométriques, soit toutes deux cinématiques.

6.1 cas de deux consignes géométriques

Lorsqu'au moins deux capteurs sur les trois sont de type géométrique, respecter les deux consignes correspondantes revient à imposer à deux vecteurs \vec{v}_1 et \vec{v}_2 connus en repère inertiel de rester sur deux cônes (\vec{a}_1, μ_1) et (\vec{a}_2, μ_2) dont les axes \vec{a}_1 et \vec{a}_2 sont connus en repère satellite. On suppose que μ_1 et μ_2 appartiennent tous deux à l'intervalle $]0; \pi[$ (si l'un des angles valait 0 ou π , la consigne correspondante fixerait deux degrés de liberté en une seule consigne ; de toutes façons, les champs de vue des capteurs limitent généralement ces angles à des valeurs de l'ordre de 60° à 120°).

On supposera ici, à chaque fois que cela sera nécessaire, que les vecteurs sont non alignés, qu'ils ne sont pas rigoureusement sur les cônes qui vont intervenir (sauf bien sûr s'ils sont définis par ces cônes), bref qu'il n'y a aucune coïncidence particulière. Tous les cas particuliers seront analysés au paragraphe suivant.

La résolution est faite dans un repère canonique⁸ tel que l'axe \vec{a}_1 du premier cône de consigne (celui sur lequel \vec{v}_1 se déplace) soit dans le plan (\vec{i}, \vec{k}) et dont l'axe \vec{k} soit l'axe \vec{a}_2 du second cône de consigne (voir figure 5). Ce repère est fixe quel que soit le choix du vecteur \vec{v}_1 sur (\vec{a}_1, μ_1) , il existe si \vec{a}_1 et \vec{a}_2 ne sont pas colinéaires. Si \vec{a}_1 et \vec{a}_2 sont colinéaires, on prend n'importe quel repère vérifiant $\vec{k} = \vec{a}_2$. À la fin des calculs, l'orientation sera donc décrite par un quaternion transformant le repère inertiel en repère canonique, une simple composition par l'inverse du quaternion canonique permettra de déterminer le vrai quaternion d'orientation, qui permet de passer du repère inertiel au repère satellite.

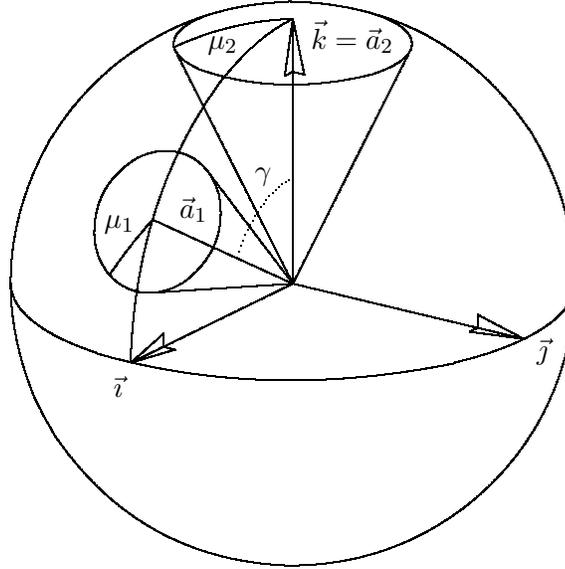
$$\vec{i} = \frac{\vec{a}_1 - (\vec{a}_1 \cdot \vec{a}_2)\vec{a}_2}{\sqrt{1 - (\vec{a}_1 \cdot \vec{a}_2)^2}} \quad \vec{j} = \frac{-\vec{a}_1 \wedge \vec{a}_2}{\sqrt{1 - (\vec{a}_1 \cdot \vec{a}_2)^2}} \quad \vec{k} = \vec{a}_2$$

Dans ce repère les vecteurs \vec{a}_1 et \vec{a}_2 s'expriment simplement (μ_1 et μ_2 sont strictement compris entre 0 et π , γ pouvant quant à lui atteindre les valeurs 0 et π) :

$$\vec{a}_1 \begin{cases} \sin \gamma = \sqrt{1 - (\vec{a}_1 \cdot \vec{a}_2)^2} \\ 0 \\ \cos \gamma = \vec{a}_1 \cdot \vec{a}_2 \end{cases} \quad \vec{a}_2 \begin{cases} 0 \\ 0 \\ 1 \end{cases}$$

⁸Les premières versions de MARMOTTES (avant la version 7.0) utilisaient un repère basé sur (\vec{v}_1, \vec{a}_2) au lieu de (\vec{a}_1, \vec{a}_2) comme décrit ici. Les équations résultantes étaient plus simples, mais il était impossible de calculer analytiquement les prolongements par continuité que nous verrons plus loin (la singularité se partageant entre le repère et les coordonnées dans le repère).

FIG. 5 – repère de résolution canonique



6.1.1 résolution générale

Le vecteur \vec{v}_2 est à la distance $\alpha_{1,2}$ de \vec{v}_1 . Cet angle est connu car on sait calculer en repère inertiel $\cos \alpha_{1,2} = \vec{v}_1 \cdot \vec{v}_2$. D'autre part, quand \vec{v}_2 respecte la seconde consigne, il est à la distance μ_2 de \vec{a}_2 .

Le vecteur \vec{v}_2 recherché est donc à l'intersection des cônes $(\vec{v}_1, \alpha_{1,2})$ et (\vec{a}_2, μ_2) , il peut être défini par un seul paramètre inconnu φ représentant l'angle de phase du vecteur autour du cône (\vec{a}_2, μ_2) :

$$\vec{v}_2 \begin{cases} \sin \mu_2 \cos \varphi \\ \sin \mu_2 \sin \varphi \\ \cos \mu_2 \end{cases}$$

Le but de la résolution va être de trouver φ en exprimant que \vec{v}_2 est également sur le cône $(\vec{v}_1, \alpha_{1,2})$.

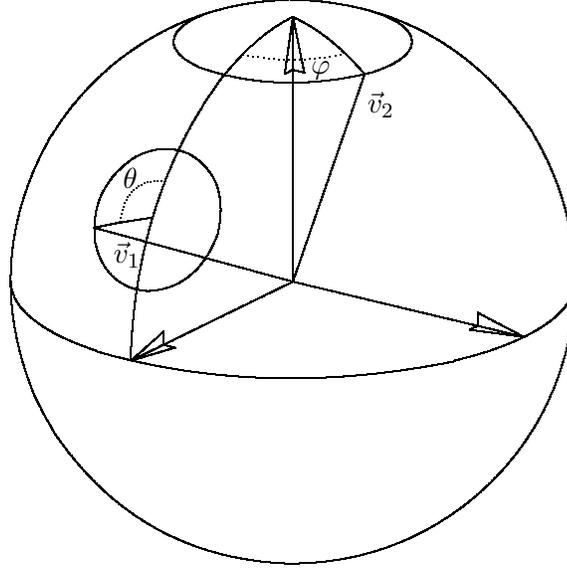
Le vecteur \vec{v}_1 a une expression plus complexe, que l'on trouve en définissant au préalable un vecteur \vec{u} sur le cône (\vec{a}_2, μ_1) et en considérant que \vec{v}_1 est l'image de \vec{u} par la rotation qui transforme \vec{a}_2 en \vec{a}_1 , c'est-à-dire la rotation d'axe \vec{j} et d'angle γ . Le vecteur \vec{u} est caractérisé par son angle de phase θ autour du cône (θ sera à terme le paramètre libre du modèle) :

$$\vec{u} \begin{cases} -\sin \mu_1 \cos \theta \\ -\sin \mu_1 \sin \theta \\ \cos \mu_1 \end{cases} \Rightarrow \vec{v}_1 \begin{cases} v_{1,x} = \sin \gamma \cos \mu_1 - \cos \gamma \sin \mu_1 \cos \theta \\ v_{1,y} = -\sin \mu_1 \sin \theta \\ v_{1,z} = \cos \gamma \cos \mu_1 + \sin \gamma \sin \mu_1 \cos \theta \end{cases}$$

Par construction le vecteur \vec{v}_2 est déjà sur le cône (\vec{a}_2, μ_2) ; on trouve l'inconnue φ en résolvant l'équation (1) qui indique que \vec{v}_2 est également sur le cône $(\vec{v}_1, \alpha_{1,2})$, avec $\alpha_{1,2}$ strictement compris entre 0 et π :

$$\begin{aligned} \cos \alpha_{1,2} &= \vec{v}_1 \cdot \vec{v}_2 \\ &= (v_{1,x} \cos \varphi + v_{1,y} \sin \varphi) \sin \mu_2 + v_{1,z} \cos \mu_2 \end{aligned} \quad (1)$$

FIG. 6 – positions des vecteurs cibles



Si $v_{1,x}$ et $v_{1,y}$ ne sont pas simultanément nuls, on peut trouver λ tel que :

$$\left. \begin{aligned} v_{1,x} &= \cos \lambda \sqrt{1 - v_{1,z}^2} \\ v_{1,y} &= \sin \lambda \sqrt{1 - v_{1,z}^2} \end{aligned} \right\} \Rightarrow \lambda = \text{atan2}(v_{1,y}, v_{1,x}) \quad (2)$$

L'équation (1) devient alors :

$$\cos(\varphi - \lambda) = \frac{\cos \alpha_{1,2} - v_{1,z} \cos \mu_2}{\sin \mu_2 \sqrt{1 - v_{1,z}^2}} \Rightarrow \left\{ \begin{aligned} \varphi_+ &= \lambda + \arccos \left(\frac{\cos \alpha_{1,2} - v_{1,z} \cos \mu_2}{\sin \mu_2 \sqrt{1 - v_{1,z}^2}} \right) \\ \varphi_- &= \lambda - \arccos \left(\frac{\cos \alpha_{1,2} - v_{1,z} \cos \mu_2}{\sin \mu_2 \sqrt{1 - v_{1,z}^2}} \right) \end{aligned} \right. \quad (3)$$

Cette équation a des solutions si la fraction définissant $\cos(\varphi - \lambda)$ est comprise entre -1 et $+1$, c'est-à-dire si :

$$\cos^2 \alpha_{1,2} + v_{1,z}^2 \cos^2 \mu_2 - 2v_{1,z} \cos \alpha_{1,2} \cos \mu_2 \leq \sin^2 \mu_2 - \sin^2 \mu_2 v_{1,z}^2 \quad (4)$$

$$\Leftrightarrow v_{1,z}^2 - 2v_{1,z} \cos \alpha_{1,2} \cos \mu_2 + \cos^2 \alpha_{1,2} - \sin^2 \mu_2 \leq 0$$

$$\Leftrightarrow (v_{1,z} - \cos \alpha_{1,2} \cos \mu_2 + \sin \alpha_{1,2} \sin \mu_2)(v_{1,z} - \cos \alpha_{1,2} \cos \mu_2 - \sin \alpha_{1,2} \sin \mu_2) \leq 0$$

$$\Leftrightarrow \cos \alpha_{1,2} \cos \mu_2 - \sin \alpha_{1,2} \sin \mu_2 \leq v_{1,z} \leq \cos \alpha_{1,2} \cos \mu_2 + \sin \alpha_{1,2} \sin \mu_2$$

$$\Leftrightarrow \cos(\alpha_{1,2} + \mu_2) \leq v_{1,z} \leq \cos(\alpha_{1,2} - \mu_2) \quad (5)$$

$$\Leftrightarrow \frac{\cos(\alpha_{1,2} + \mu_2) - \cos \gamma \cos \mu_1}{\sin \gamma \sin \mu_1} \leq \cos \theta \leq \frac{\cos(\alpha_{1,2} - \mu_2) - \cos \gamma \cos \mu_1}{\sin \gamma \sin \mu_1} \quad (6)$$

$$\Leftrightarrow c_{\min} \leq \cos \theta \leq c_{\max}$$

Dans le cas où \vec{a}_1 et \vec{a}_2 sont alignés, la condition (4) ne dépend alors plus de θ , on est obligé d'arrêter le calcul au niveau de l'expression (5). Dans ce cas, on calcule directement la valeur constante de $v_{1,z}$ et on en

déduit immédiatement soit que la condition (5) est toujours remplie c'est-à-dire que tous les θ de $-\pi$ à π sont valides, soit qu'aucun θ n'est valide.

Selon les valeurs des données initiales $\vec{v}_1 \cdot \vec{v}_2 = \cos \alpha_{1,2}$, $\vec{a}_1 \cdot \vec{a}_2 = \cos \gamma$, μ_1 et μ_2 ; c_{\min} et c_{\max} peuvent prendre à peu près n'importe quelles valeurs. On a regroupé dans la table 1 les conséquences que cela avait sur le domaine de validité de θ .

TAB. 1 – domaines de validité

c_{\min}, c_{\max}	domaine de validité de θ
$c_{\min} < c_{\max} \leq -1$	\emptyset
$c_{\min} \leq -1 \leq c_{\max} \leq +1$	$[-\pi; -\arccos(c_{\max})]$ \cup $[\arccos(c_{\max}); \pi]$
$c_{\min} \leq -1 \leq +1 \leq c_{\max}$	$[-\pi; \pi]$
$-1 \leq c_{\min} < c_{\max} \leq +1$	$[-\arccos(c_{\min}); -\arccos(c_{\max})]$ \cup $[\arccos(c_{\max}); \arccos(c_{\min})]$
$-1 \leq c_{\min} \leq +1 \leq c_{\max}$	$[-\arccos(c_{\min}); \arccos(c_{\min})]$
$+1 \leq c_{\min} < c_{\max}$	\emptyset

On dispose donc d'un domaine de validité de θ éventuellement constitué de deux intervalles disjoints, et d'un jeu de formules $\theta \rightarrow \vec{v}_1 \rightarrow \cos(\varphi - \lambda) \rightarrow \varphi \rightarrow \vec{v}_2$, c'est-à-dire $\vec{v}_2 = g(\theta)$.

Ces équations ne sont valides que si $v_{1,x}$ et $v_{1,y}$ ne s'annulent pas simultanément. En étudiant plus précisément sous quelles conditions cette annulation peut intervenir, nous allons voir qu'il est possible de réaliser un prolongement par continuité analytique qui permettra d'éviter les problèmes numériques à proximité des points litigieux.

6.1.2 traitement des singularités

Le terme $v_{1,y}$ ne s'annule que si $\theta = 0$ ou $\theta = \pi$; seules ces deux valeurs peuvent donc conduire à des singularités. Cependant cette condition n'est pas suffisante : $v_{1,y}$ s'annule à chaque fois que $\theta = 0$ ou $\theta = \pi$, mais $v_{1,x}$ peut très bien être non nul et les équations de la résolution générale précédente s'appliquer sans problème.

Ainsi la valeur $\theta = 0$ qui annule déjà $v_{1,y}$ n'annulera également $v_{1,x}$ que si $\sin(\gamma - \mu_1) = 0$, c'est-à-dire si $\gamma = \mu_1$ (on rappelle que μ_1 est strictement compris entre 0 et π , et que γ est compris largement entre 0 et π , on ne peut donc pas avoir $\gamma = \pi + \mu_1$).

De même la valeur $\theta = \pi$ qui annule déjà $v_{1,y}$ n'annulera également $v_{1,x}$ que si $\sin(\gamma + \mu_1) = 0$, c'est-à-dire si $\gamma = \pi - \mu_1$.

La valeur 0 n'appartient au domaine de validité de θ et ne correspond à une singularité que si :

$$\begin{aligned} (6) \quad &\Rightarrow \cos(\alpha_{1,2} + \mu_2) \leq \cos(\gamma - \mu_1) \leq \cos(\alpha_{1,2} - \mu_2) \\ &\Rightarrow \cos(\alpha_{1,2} + \mu_2) \leq 1 \leq \cos(\alpha_{1,2} - \mu_2) \end{aligned}$$

Seule la deuxième inégalité apporte de l'information, puisqu'elle se transforme de façon triviale en égalité : $\cos(\alpha_{1,2} - \mu_2) = 1$, d'où $\alpha_{1,2} = \mu_2$.

Observons ce qui se passe aux alentours de cette valeur critique de θ . On peut faire apparaître $\sin \frac{\theta}{2}$ explicitement dans les coordonnées de \vec{v}_1 :

$$\begin{cases} v_{1,x} = \sin \mu_1 \cos \mu_1 (1 - \cos \theta) = 2 \sin \mu_1 \cos \mu_1 \sin^2 \frac{\theta}{2} \\ v_{1,y} = -\sin \mu_1 \sin \theta = -2 \cos \frac{\theta}{2} \sin \mu_1 \sin \frac{\theta}{2} \\ v_{1,z} = \cos^2 \mu_1 + \sin^2 \mu_1 \cos \theta = 1 - \sin^2 \mu_1 (1 - \cos \theta) = 1 - 2 \sin^2 \mu_1 \sin^2 \frac{\theta}{2} \end{cases}$$

L'équation (1) peut être réécrite pour les θ non nuls, puis simplifiée par $2 \sin \mu_1 \sin \frac{\theta}{2}$, ce qui réalise un prolongement par continuité en $\theta = 0$:

$$\begin{aligned} \cos \mu_2 &= 2 \sin \mu_1 \sin \frac{\theta}{2} \sin \mu_2 \left(\cos \mu_1 \sin \frac{\theta}{2} \cos \varphi - \cos \frac{\theta}{2} \sin \varphi \right) + \left(1 - 2 \sin^2 \mu_1 \sin^2 \frac{\theta}{2} \right) \cos \mu_2 \\ \Leftrightarrow 2 \sin \mu_1 \sin \frac{\theta}{2} \left((\cos \mu_1 \sin \frac{\theta}{2} \cos \varphi - \cos \frac{\theta}{2} \sin \varphi) \sin \mu_2 - \sin \mu_1 \cos \mu_2 \sin \frac{\theta}{2} \right) &= 0 \\ \Leftrightarrow (\cos \mu_1 \sin \frac{\theta}{2} \cos \varphi - \cos \frac{\theta}{2} \sin \varphi) \sin \mu_2 &= \sin \mu_1 \cos \mu_2 \sin \frac{\theta}{2} \\ \Leftrightarrow \cos \mu_1 \sin \frac{\theta}{2} \cos \varphi - \cos \frac{\theta}{2} \sin \varphi &= \frac{\sin \mu_1 \cos \mu_2}{\sin \mu_2} \sin \frac{\theta}{2} \end{aligned}$$

Si $\mu_1 \neq \pi/2$, les deux termes $\cos \mu_1 \sin \frac{\theta}{2}$ et $-\cos \frac{\theta}{2}$ ne s'annulent pas simultanément ; on peut trouver λ_0 tel que :

$$\left. \begin{aligned} \cos \mu_1 \sin \frac{\theta}{2} &= \cos \lambda_0 \sqrt{\cos^2 \mu_1 \sin^2 \frac{\theta}{2} + \cos^2 \frac{\theta}{2}} \\ -\cos \frac{\theta}{2} &= \sin \lambda_0 \sqrt{\cos^2 \mu_1 \sin^2 \frac{\theta}{2} + \cos^2 \frac{\theta}{2}} \end{aligned} \right\} \Rightarrow \lambda_0 = \operatorname{atan2} \left(-\cos \frac{\theta}{2}, \cos \mu_1 \sin \frac{\theta}{2} \right) \quad (7)$$

L'équation devient alors :

$$\cos(\varphi - \lambda_0) = \frac{\sin \mu_1 \cos \mu_2 \sin \frac{\theta}{2}}{\sin \mu_2 \sqrt{\cos^2 \mu_1 \sin^2 \frac{\theta}{2} + \cos^2 \frac{\theta}{2}}} \Rightarrow \begin{cases} \varphi_+ = \lambda_0 + \arccos \left(\frac{\sin \mu_1 \sin \frac{\theta}{2} \cos \mu_2}{\sin \mu_2 \sqrt{1 - \sin^2 \mu_1 \sin^2 \frac{\theta}{2}}} \right) \\ \varphi_- = \lambda_0 - \arccos \left(\frac{\sin \mu_1 \sin \frac{\theta}{2} \cos \mu_2}{\sin \mu_2 \sqrt{1 - \sin^2 \mu_1 \sin^2 \frac{\theta}{2}}} \right) \end{cases} \quad (8)$$

Si $\mu_1 = \pi/2$ on ne peut pas calculer λ_0 , un traitement spécifique est nécessaire. On sait que dans ce cas $\gamma = \pi/2$ et $\alpha_{1,2} = \mu_2$. Les coordonnées de \vec{v}_1 se simplifient alors considérablement :

$$\begin{cases} v_{1,x} = 0 \\ v_{1,y} = -\sin \theta \\ v_{1,z} = \cos \theta \end{cases}$$

L'équation (1) peut être réécrite pour les θ différents de 0 puis simplifiée par $2 \sin \frac{\theta}{2}$, ce qui réalise un prolongement par continuité en $\theta = 0$:

$$\begin{aligned} & \sin \theta \sin \varphi \sin \mu_2 + (1 - \cos \theta) \cos \mu_2 = 0 \\ \Leftrightarrow & 2 \sin \frac{\theta}{2} (\cos \frac{\theta}{2} \sin \varphi \sin \mu_2 + \sin \frac{\theta}{2} \cos \mu_2) = 0 \\ \Leftrightarrow & \cos \frac{\theta}{2} \sin \varphi \sin \mu_2 + \sin \frac{\theta}{2} \cos \mu_2 = 0 \end{aligned}$$

Le domaine de validité de θ défini par l'équation (6) devient : $\cos 2\mu_2 \leq \cos \theta \leq 1$, on en déduit que si μ_2 est différent de $\pi/2$ alors θ sera toujours différent de π , dans ce cas on peut écrire :

$$\sin \varphi = -\frac{\tan \frac{\theta}{2}}{\tan \mu_2} \Rightarrow \begin{cases} \varphi_+ = \pi + \arcsin \frac{\tan \frac{\theta}{2}}{\tan \mu_2} \\ \varphi_- = -\arcsin \frac{\tan \frac{\theta}{2}}{\tan \mu_2} \end{cases} \quad (9)$$

Le cas $\mu_1 = \mu_2 = \pi/2$ est traité dans la section suivante.

La valeur π n'appartient au domaine de validité de θ et ne correspond à une singularité que si :

$$\begin{aligned} (6) \Rightarrow & \cos(\alpha_{1,2} + \mu_2) \leq \cos(\gamma + \mu_1) \leq \cos(\alpha_{1,2} - \mu_2) \\ \Rightarrow & \cos(\alpha_{1,2} + \mu_2) \leq -1 \leq \cos(\alpha_{1,2} - \mu_2) \end{aligned}$$

Seule la première inégalité apporte de l'information, puisqu'elle se transforme de façon triviale en égalité : $\cos(\alpha_{1,2} + \mu_2) = -1$, d'où $\alpha_{1,2} = \pi - \mu_2$.

Observons ce qui se passe aux alentours de cette valeur critique de θ . On peut faire apparaître $\cos \frac{\theta}{2}$ explicitement dans les coordonnées de \vec{v}_1 :

$$\begin{cases} v_{1,x} = \sin \mu_1 \cos \mu_1 (1 + \cos \theta) = 2 \sin \mu_1 \cos \mu_1 \cos^2 \frac{\theta}{2} \\ v_{1,y} = -\sin \mu_1 \sin \theta = -2 \sin \mu_1 \sin \frac{\theta}{2} \cos \frac{\theta}{2} \\ v_{1,z} = -\cos^2 \mu_1 + \sin^2 \mu_1 \cos \theta = \sin^2 \mu_1 (1 + \cos \theta) - 1 = 2 \sin^2 \mu_1 \cos^2 \frac{\theta}{2} - 1 \end{cases}$$

L'équation (1) peut être réécrite pour les θ différents de π , puis simplifiée par $2 \sin \mu_1 \cos \frac{\theta}{2}$, ce qui réalise un prolongement par continuité en $\theta = \pi$:

$$\begin{aligned} & -\cos \mu_2 = 2 \sin \mu_1 \cos \frac{\theta}{2} \sin \mu_2 \left(\cos \mu_1 \cos \frac{\theta}{2} \cos \varphi - \sin \frac{\theta}{2} \sin \varphi \right) + \left(2 \sin^2 \mu_1 \cos^2 \frac{\theta}{2} - 1 \right) \cos \mu_2 \\ \Leftrightarrow & 2 \sin \mu_1 \cos \frac{\theta}{2} \left((\cos \mu_1 \cos \frac{\theta}{2} \cos \varphi - \sin \frac{\theta}{2} \sin \varphi) \sin \mu_2 + \sin \mu_1 \cos \mu_2 \cos \frac{\theta}{2} \right) = 0 \\ \Leftrightarrow & (\cos \mu_1 \cos \frac{\theta}{2} \cos \varphi - \sin \frac{\theta}{2} \sin \varphi) \sin \mu_2 = -\sin \mu_1 \cos \mu_2 \cos \frac{\theta}{2} \\ \Leftrightarrow & \cos \mu_1 \cos \frac{\theta}{2} \cos \varphi - \sin \frac{\theta}{2} \sin \varphi = -\frac{\sin \mu_1 \cos \mu_2}{\sin \mu_2} \cos \frac{\theta}{2} \end{aligned}$$

Si $\mu_1 \neq \pi/2$, les deux termes $\cos \mu_1 \cos \frac{\theta}{2}$ et $-\sin \frac{\theta}{2}$ ne s'annulent pas simultanément ; on peut trouver λ_π tel que :

$$\left. \begin{aligned} \cos \mu_1 \cos \frac{\theta}{2} &= \cos \lambda_\pi \sqrt{\sin^2 \frac{\theta}{2} + \cos^2 \mu_1 \cos^2 \frac{\theta}{2}} \\ -\sin \frac{\theta}{2} &= \sin \lambda_\pi \sqrt{\sin^2 \frac{\theta}{2} + \cos^2 \mu_1 \cos^2 \frac{\theta}{2}} \end{aligned} \right\} \Rightarrow \lambda_\pi = \operatorname{atan2} \left(-\sin \frac{\theta}{2}, \cos \mu_1 \cos \frac{\theta}{2} \right) \quad (10)$$

L'équation devient alors :

$$\cos(\varphi - \lambda_\pi) = \frac{-\sin \mu_1 \cos \mu_2 \cos \frac{\theta}{2}}{\sin \mu_2 \sqrt{\sin^2 \frac{\theta}{2} + \cos^2 \mu_1 \cos^2 \frac{\theta}{2}}} \Rightarrow \begin{cases} \varphi_+ = \lambda_\pi + \arccos \left(\frac{-\sin \mu_1 \cos \frac{\theta}{2} \cos \mu_2}{\sin \mu_2 \sqrt{1 - \sin^2 \mu_1 \cos^2 \frac{\theta}{2}}} \right) \\ \varphi_- = \lambda_\pi - \arccos \left(\frac{-\sin \mu_1 \cos \frac{\theta}{2} \cos \mu_2}{\sin \mu_2 \sqrt{1 - \sin^2 \mu_1 \cos^2 \frac{\theta}{2}}} \right) \end{cases} \quad (11)$$

Si $\mu_1 = \pi/2$ on ne peut pas calculer λ_π , un traitement spécifique est nécessaire. On sait que dans ce cas $\gamma = \pi/2$ et $\alpha_{1,2} = \pi - \mu_2$. Les coordonnées de \vec{v}_1 se simplifient alors considérablement :

$$\begin{cases} v_{1,x} = 0 \\ v_{1,y} = -\sin \theta \\ v_{1,z} = \cos \theta \end{cases}$$

L'équation (1) peut être réécrite pour les θ différents de π puis simplifiée par $2 \sin \frac{\pi-\theta}{2}$, ce qui réalise un prolongement par continuité en $\theta = \pi$:

$$\begin{aligned} & \sin \theta \sin \varphi \sin \mu_2 - (1 + \cos \theta) \cos \mu_2 = 0 \\ \Leftrightarrow & 2 \sin \frac{\pi-\theta}{2} (\cos \frac{\pi-\theta}{2} \sin \varphi \sin \mu_2 - \sin \frac{\pi-\theta}{2} \cos \mu_2) = 0 \\ \Leftrightarrow & \cos \frac{\pi-\theta}{2} \sin \varphi \sin \mu_2 - \sin \frac{\pi-\theta}{2} \cos \mu_2 = 0 \end{aligned}$$

Le domaine de validité de θ défini par l'équation (6) devient : $-1 \leq \cos \theta \leq \cos(\pi - 2\mu_2)$, on en déduit que si μ_2 est différent de $\pi/2$ alors θ sera toujours différent de 0, dans ce cas on peut écrire :

$$\sin \varphi = \frac{\tan \frac{\pi-\theta}{2}}{\tan \mu_2} \Rightarrow \begin{cases} \varphi_+ = \pi - \arcsin \frac{\tan \frac{\pi-\theta}{2}}{\tan \mu_2} \\ \varphi_- = \arcsin \frac{\tan \frac{\pi-\theta}{2}}{\tan \mu_2} \end{cases} \quad (12)$$

Le cas $\mu_1 = \mu_2 = \pi/2$ est traité dans la section suivante.

6.1.3 familles à un vecteur fixe

Les prolongements par continuité de la section précédente ne sont pas capables de trouver *toutes* les solutions, ils se contentent d'améliorer la robustesse des calculs et de trouver les valeurs de φ qui sont la suite logique des valeurs hors alignement. En effet, ils ont été réalisés en considérant que dans des équations du type $f(\varphi, \theta) \sin \mu_1 \sin \frac{\theta}{2} = 0$ ou $f(\varphi, \theta) \sin \mu_1 \cos \frac{\theta}{2} = 0$, c'était le terme $f(\varphi, \theta)$ qui s'annulait, d'où une définition implicite d'une fonction $\varphi = g(\theta)$. Toutes les valeurs de φ entre 0 et 2π annulent cependant les équations dans le cas particulier où c'est $\sin \frac{\theta}{2}$ ou $\cos \frac{\theta}{2}$ qui s'annule. Ces solutions représentent des familles indépendantes : elles ont le même $\vec{v}_1(\theta)$ mais pas forcément⁹ le même $\vec{v}_2(\varphi)$ que les familles prolongées. Ces familles peuvent être décrites en trouvant toutes les orientations définies par \vec{v}_1 fixe et \vec{v}_2 prenant toutes les positions possibles autour de \vec{v}_1 .

⁹les valeurs de φ égales aux valeurs prolongées conduisent bien entendu à trouver certains couples (\vec{v}_1, \vec{v}_2) dans plusieurs familles, cette duplication des solutions n'a dans la pratique aucune conséquence

Un autre cas peut être modélisé de façon similaire, il s'agit du cas particulier des prolongements par continuité en 0 ou π lorsque ces points correspondent à des singularités et qu'en plus on a $\mu_1 = \mu_2 = \pi/2$. Lorsque ce cas provient d'un prolongement en 0, on a $\gamma = \mu_1$ et $\alpha_{1,2} = \mu_2$, lorsqu'il provient d'un prolongement en π , on a $\gamma = \pi - \mu_1$ et $\alpha_{1,2} = \pi - \mu_2$; ces deux cas se ramènent donc tous les deux à $\gamma = \alpha_{1,2} = \pi/2$. Dans ces conditions l'équation (1) se simplifie de façon drastique et se résoud immédiatement pour les valeurs de $\sin \theta$ non nulles, la solution se prolongeant de façon triviale pour les valeurs nulles :

$$\sin \theta \sin \varphi = 0 \Rightarrow \begin{cases} \varphi_+ = \pi \\ \varphi_- = 0 \end{cases}$$

ce qui revient à dire que l'orientation est modélisée par deux familles définies par \vec{v}_2 fixe et \vec{v}_1 prenant toutes les positions possibles autour de \vec{v}_2 . Il faut cependant prendre garde que ces familles sont bien des familles générales prolongées et qu'il faut ajouter les familles d'alignement correspondant à $\sin \theta = 0$ et φ quelconque, c'est-à-dire les familles à \vec{v}_1 fixe mentionnées au début de cette section.

Un dernier cas se modélisant encore par des familles ayant un vecteur fixe correspond à l'alignement de \vec{v}_1 et \vec{v}_2 (c'est-à-dire au cas où $\alpha_{1,2}$ vaut 0 ou π). Dans ce cas, la donnée du couple (\vec{v}_1, \vec{v}_2) ne suffit pas à déterminer l'orientation, il faut un traitement spécifique. Il est très improbable que deux astres différents soient rigoureusement alignés lorsqu'ils sont vus du satellite. Par contre, il est tout à fait concevable d'avoir deux consignes géométriques sur le même astre (par exemple, lacet Soleil = 0° , tangage Soleil = 0°) : l'alignement d'un vecteur avec lui-même est inévitable. Ce cas particulier est donc là encore très courant.

On peut tout d'abord remarquer que si $\vec{v}_2 = -\vec{v}_1$ (cas improbable de deux astres différents), on peut transformer la consigne « \vec{v}_2 est sur le cône (\vec{a}_2, μ_2) » en consigne « $-\vec{v}_2$ sur le cône $(-\vec{a}_2, \mu_2)$ ». On se ramène ainsi au cas $\vec{v}_1 = \vec{v}_2$.

Le vecteur \vec{v}_1 se trouve alors à l'intersection des cônes (\vec{a}_1, μ_1) et (\vec{a}_2, μ_2) . Si les cônes sont égaux, il y a une infinité de solutions : les consignes ne sont pas indépendantes ce qui revient à dire que le problème est mal posé. La bibliothèque de résolution doit dans ce cas générer un message d'erreur. Si les cônes n'ont pas d'intersection, les consignes sont incompatibles, l'orientation n'est pas pilotable. Dans le cas général, il y aura deux intersections, que l'on sait calculer (voir l'annexe F). On a donc là encore deux familles qui peuvent être décrites en trouvant toutes les orientations définies par \vec{v}_1 fixe.

Dans tous les cas précédents, les familles sont définies par un vecteur (\vec{v}_1 ou \vec{v}_2) et par toutes les possibilités de rotation autour de ce vecteur. Les premiers exemples définissaient ces possibilités par la donnée de l'autre vecteur, mais c'est dans la pratique inutile¹⁰, il suffit de s'assurer que l'on explore bien toutes les possibilités. On adopte donc la modélisation suivante : tous les membres de la famille i sont définis par la composition de la rotation constante transformant \vec{v}_1 en repère inertiel en \vec{v}_{1i} en repère satellite (respectivement \vec{v}_2 et \vec{v}_{2i}) et de la rotation d'angle θ' autour du vecteur \vec{v}_{1i} (en repère satellite). Le degré de liberté de ces modèles est l'angle θ' . Les formules permettant de calculer les quaternions de ces rotations sont explicitées dans les annexes D et E.

6.1.4 réduction du domaine de validité

Les modèles décrits jusqu'ici ont tous un unique degré de liberté qui est un angle, éventuellement restreint à un domaine de validité. Cette restriction correspond exclusivement aux conditions mathématiques permettant

¹⁰ce n'est de toute façon pas possible pour le dernier exemple, comme nous venons de le remarquer

de résoudre les équations. Il est cependant possible que l'on sache dès la mise en place des cônes de consigne que les cibles ne sont pas autorisées à sortir de secteurs particuliers de ces cônes de consignes. Un premier exemple est donné par les capteurs d'angles dièdres, dont les consignes définissent plutôt des demi-plans que des plans ; un autre exemple concerne les capteurs optiques dont les champs de vue limitent les portions utilisables des cônes de consigne.

Dans certains cas, ces contraintes technologiques peuvent être superposées aux contraintes mathématiques déjà vues. Cela permet alors d'éviter de perdre du temps à chercher des solutions théoriques dans certaines zones, alors que ces solutions seront rejetées dans la phase de filtrage par les contraintes technologiques postérieure à la résolution.

Les intervalles de θ autorisés sont directement liés aux contraintes du premier capteur puisqu'ils décrivent la position de \vec{v}_1 . On peut donc les filtrer par ces secteurs de contrainte ; certains intervalles (et donc certaines familles) disparaîtront complètement au travers de ce filtrage, d'autres seront découpés en plusieurs sous-intervalles (ce qui signifie qu'ils généreront plusieurs familles), d'autres enfin ne seront pas affectés. Dans la pratique cela signifie que la table 1 ne doit pas être appliquée à l'intervalle $[c_{\min}; c_{\max}]$ défini par l'équation (6), mais à une série d'intervalles $[c_{\min,1}; c_{\max,1}] \cup \dots \cup [c_{\min,n}; c_{\max,n}]$ résultant du filtrage de l'intervalle d'origine par les secteurs de consigne du premier capteur.

Le deuxième capteur a également des secteurs de contrainte qui portent sur la cible \vec{v}_2 , il est cependant plus délicat de reporter ces contraintes sur les modèles, car φ est un résultat du modèle et non une entrée comme θ . On peut cependant les utiliser dans quelques cas simples, comme par exemple lorsque \vec{v}_1 est fixe et que le modèle peut être paramétré par φ ou lorsque c'est \vec{v}_2 qui est fixe.

6.1.5 synthèse

On dispose d'une série de six modèles différents. Le premier correspond aux équations (2) et (3), il s'agit d'un modèle général qui est applicable lorsqu'aucune singularité ne peut être rencontrée au cours de la résolution. Les quatre suivants correspondent aux équations (7) et (8), à l'équation (9), aux équations (10) et (11), et à l'équation (12). Ils sont applicables lorsqu'une seule singularité doit être rencontrée, et sont valides même au niveau de la singularité grâce à un prolongement par continuité analytique. Le sixième modèle est applicable soit lorsqu'il faut introduire des familles supplémentaires en plus des prolongements, soit pour faire le prolongement lui-même lorsque certains angles sont droits, soit lorsque les cibles sont alignées. Ces modèles sont disjoints les uns des autres, il n'est donc pas utile de gérer des basculements de modèles selon la valeur de θ ¹¹. Une même résolution d'orientation doit cependant gérer plusieurs de ces modèles simultanément, on ne sait pas *a priori* laquelle des familles fournira la solution finale respectant la troisième consigne.

Toutes ces familles correspondent à des couples de vecteurs dépendant du paramètre libre θ . Les formules de l'annexe B, page 46 permettent de trouver l'orientation à partir de ces couples, pour une valeur de θ donnée. Cette orientation et l'orientation précédente permettent de calculer par différences finies la rotation moyenne depuis le pas précédent. On fait l'hypothèse que la rotation est constantes, la rotation moyenne est donc égale à la rotation instantanée. On dispose ainsi de l'état complet (orientation et rotation instantanée) pour chercher la valeur du paramètre θ qui permettra de respecter la troisième consigne.

L'algorithme suivant explicite une façon d'organiser les choix pour construire tous les modèles nécessaires à

¹¹ ces basculements poseraient d'ailleurs des problèmes numériques pour des solutions aux alentours de l'angle de basculement

une détermination d'orientation. L'utilisation de ces modèles pour faire la résolution n'est pas décrite ici. On appelle *secteur i* la série des intervalles autorisés pour le cône i .

Si $\mu_1 = \gamma = \pi/2$

Si $\mu_2 = \alpha_{1,2} = \pi/2$

Si $\vec{a}_2 \in$ secteur 1

créer la famille $\vec{v}_1 = \vec{a}_2$ fixe, restreinte par $\vec{v}_2 \in$ secteur 2

Fin si

Si $-\vec{a}_2 \in$ secteur 1

créer la famille $\vec{v}_1 = -\vec{a}_2$ fixe, restreinte par $\vec{v}_2 \in$ secteur 2

Fin si

Si $\vec{a}_1 \in$ secteur 2

créer la famille $\vec{v}_2 = \vec{a}_1$ fixe, restreinte par $\vec{v}_1 \in$ secteur 1

Fin si

Si $-\vec{a}_1 \in$ secteur 2

créer la famille $\vec{v}_2 = -\vec{a}_1$ fixe, restreinte par $\vec{v}_1 \in$ secteur 1

Fin si

Sinon

$$c_{\min} \leftarrow \cos(\alpha_{1,2} + \mu_2), \quad c_{\max} \leftarrow \cos(\alpha_{1,2} - \mu_2)$$

filtrer le domaine de θ par le secteur 1

Pour chaque intervalle du domaine de θ

Si θ passe par 0

créer la famille $\vec{v}_1 = \vec{a}_2$ fixe, restreinte par $\vec{v}_2 \in$ secteur 2

créer les familles de prolongement en 0 par l'équation (9)

Sinon

Si θ passe par π

créer la famille $\vec{v}_1 = -\vec{a}_2$ fixe, restreinte par $\vec{v}_2 \in$ secteur 2

créer les familles de prolongement en π par l'équation (12)

Sinon

créer les familles générales par les équations (2) et (3)

Fin si

Fin si

Fin pour

Fin si

Sinon

$$c_{\min} \leftarrow \frac{\cos(\alpha_{1,2} + \mu_2) - \cos \gamma \cos \mu_1}{\sin \gamma \sin \mu_1}, \quad c_{\max} \leftarrow \frac{\cos(\alpha_{1,2} - \mu_2) - \cos \gamma \cos \mu_1}{\sin \gamma \sin \mu_1}$$

filtrer le domaine de θ par le secteur 1

Pour chaque intervalle du secteur 1

Si θ passe par 0 et que $\mu_1 = \gamma$

créer la famille $\vec{v}_1 = \vec{a}_2$ fixe, restreinte par $\vec{v}_2 \in$ secteur 2

créer les familles de prolongement en 0 par les équation (7) et (8)

Sinon

Si θ passe par π et que $\mu_1 = \pi - \gamma$

créer la famille $\vec{v}_1 = -\vec{a}_2$ fixe, restreinte par $\vec{v}_2 \in$ secteur 2

créer les familles de prolongement en π par les équations (10) et (11)

Sinon

créer les familles générales par les équations (2) et (3)

Fin si

Fin si

Fin pour

Fin si

6.2 cas de deux consignes cinématiques

Lorsqu'au moins deux capteurs sur les trois sont de type cinématique, respecter les deux consignes correspondantes revient à imposer au vecteur de rotation instantanée d'avoir pour projection sur les vecteurs \vec{a}_1 et \vec{a}_2

les valeurs ω_1 et ω_2 . Les deux consignes sont indépendantes si \vec{a}_1 et \vec{a}_2 ne sont pas alignés. Dans ce cas, on peut exprimer le vecteur rotation instantanée $\vec{\Omega}$ dans le repère non orthonormé $(\vec{a}_1, \vec{a}_2, \vec{a}_1 \wedge \vec{a}_2)$.

$$\vec{\Omega} = \frac{\omega_1 - \omega_2(\vec{a}_1 \cdot \vec{a}_2)}{1 - (\vec{a}_1 \cdot \vec{a}_2)^2} \vec{a}_1 + \frac{\omega_2 - \omega_1(\vec{a}_1 \cdot \vec{a}_2)}{1 - (\vec{a}_1 \cdot \vec{a}_2)^2} \vec{a}_2 + \theta(\vec{a}_1 \wedge \vec{a}_2) \quad (13)$$

Dans cette formule, le seul degré de liberté est θ , et il peut varier de moins l'infini à plus l'infini. On peut réécrire la formule 13 sous la forme :

$$\vec{\Omega} = \vec{u} + \theta \vec{v} \quad (14)$$

où \vec{u} et \vec{v} sont des vecteurs constants orthogonaux. On a alors $\|\vec{\Omega}\|^2 = \|\vec{u}\|^2 + \theta^2 \|\vec{v}\|^2$, qui représente le carré de la vitesse de rotation instantanée. Les satellites ne tournent généralement pas très vite sur eux-mêmes, dans la pratique on peut limiter le domaine de variation de θ à l'intervalle $[-\theta_{\max}; \theta_{\max}]$ avec :

$$\theta_{\max} = \sqrt{\frac{\omega_{\max}^2 - \|\vec{u}\|^2}{\|\vec{v}\|^2}} \quad (15)$$

où ω_{\max} est une vitesse maximale de rotation. Si l'utilisateur le désire, il peut initialiser cette valeur selon son satellite, sachant que s'il ne le fait pas, la bibliothèque utilisera par défaut $\omega_{\max} = 0,4$ radians par seconde (soit environ 23 degrés par seconde). Cette valeur par défaut est un compromis entre la nécessité de couvrir les cas plausibles de satellites très rapides et le risque de générer toute une série d'orientations, solutions de la troisième consigne, selon la vitesse choisie, pour finalement rejeter toutes celles qui correspondent à des rotations plus rapides que quelques dixièmes de degré par seconde.

Pour chaque θ valide, on calcule l'orientation correspondante en supposant le vecteur de rotation constant entre le pas précédent et le pas courant, à l'aide de la relation :

$$Q(\theta, h) = Q(t)E(\theta, h) \quad (16)$$

où le quaternion $E(\theta, h)$ d'évolution de l'orientation entre t et $t + h$ est donné par :

$$E(\theta, h) = \begin{bmatrix} \cos(\frac{\omega}{2}h) \\ \frac{1}{\omega} \sin(\frac{\omega}{2}h) \vec{\Omega}(\theta) \end{bmatrix} \quad (17)$$

avec $\omega = \|\vec{\Omega}(\theta)\|$.

Dans les relations 16 et 17, le seul paramètre libre est θ , les paramètres t et h sont figés par le pas d'intégration de l'orientation imposé par le programme appelant.

7 fonction d'erreur sur une consigne

Connaissant une orientation donnée par son quaternion Q et une rotation instantanée $\vec{\Omega}$, on cherche à identifier une fonction de cette orientation et de cette rotation s'annulant lorsqu'elles respectent une consigne fixe. Pour cela, on utilise une fonction d'erreur indiquant l'écart entre une valeur obtenue et la valeur désirée.

Une méthode permettant d'annuler cette fonction d'erreur est donnée dans l'annexe H, page 62. Bien d'autres méthodes peuvent être utilisées, cela n'a pas d'importance pour l'algorithme.

7.1 cas d'une consigne géométrique

Une consigne géométrique est modélisée par un cône (\vec{a}_3, μ_3) sur lequel un vecteur \vec{v}_3 doit rester confiné, \vec{v}_3 étant la projection en repère satellite d'un vecteur \vec{u}_3 connu en repère inertiel. La fonction d'erreur est donnée par :

$$f(Q) = \arccos(Q(\vec{u}_3) \cdot \vec{a}_3) - \mu_3 \quad (18)$$

Cette fonction est continue, indéfiniment dérivable, bornée entre μ_3 et $\pi - \mu_3$, et s'annule lorsque $\vec{v}_3 = Q(\vec{u}_3)$ est sur le cône (\vec{a}_3, μ_3) . Les formules permettant de calculer $\vec{v}_3 = Q(\vec{u}_3)$ sont décrites dans l'annexe A.

7.2 cas d'une consigne cinématique

Une consigne cinématique est modélisée par un axe sensible \vec{a}_3 et une vitesse ω_3 telles que : $\vec{\Omega} \cdot \vec{a}_3 = \omega_3$ où $\vec{\Omega}$ est le vecteur vitesse de rotation instantanée du satellite.

Si on note $Q(t)$ le quaternion d'orientation au pas précédent et $Q(t+h)$ le quaternion d'orientation courant (dans la pratique, $Q(t+h)$ est calculé en fonction d'un paramètre θ et non en fonction de $t+h$ — voir section 6), alors le quaternion $E(h)$ représentant l'évolution d'orientation au cours du pas est donné par :

$$E(h) = Q^{-1}(t) (Q(t+h)) \quad (19)$$

d'où l'on déduit :

$$\vec{\Omega}(h) = \frac{2 \arccos(e_0)}{h} \frac{\vec{e}}{\|\vec{e}\|} \quad (20)$$

où e_0 et \vec{e} sont respectivement les composantes scalaire et vectorielle de $E(h)$.

La fonction d'erreur est alors :

$$f(Q(t+h)) = 2 \arccos(e_0)(\vec{e} \cdot \vec{a}_3) - h \sqrt{1 - e_0^2} \omega_3 \quad (21)$$

8 Modélisation des capteurs

La modélisation des capteurs a été présentée de façon succincte avec les principes de fonctionnement internes. Nous allons maintenant décrire la partie visible par l'utilisateur et qui devra être définie dans les fichiers de ressources : ce sont donc réellement les données d'entrée de la bibliothèque MARMOTTES.

Les nombreux capteurs utilisés sur les satellites ont souvent des caractéristiques communes, ils peuvent être hiérarchisés selon la notion classique en analyse orientée objet de généralisation¹². La figure 7 présente cette hiérarchie.

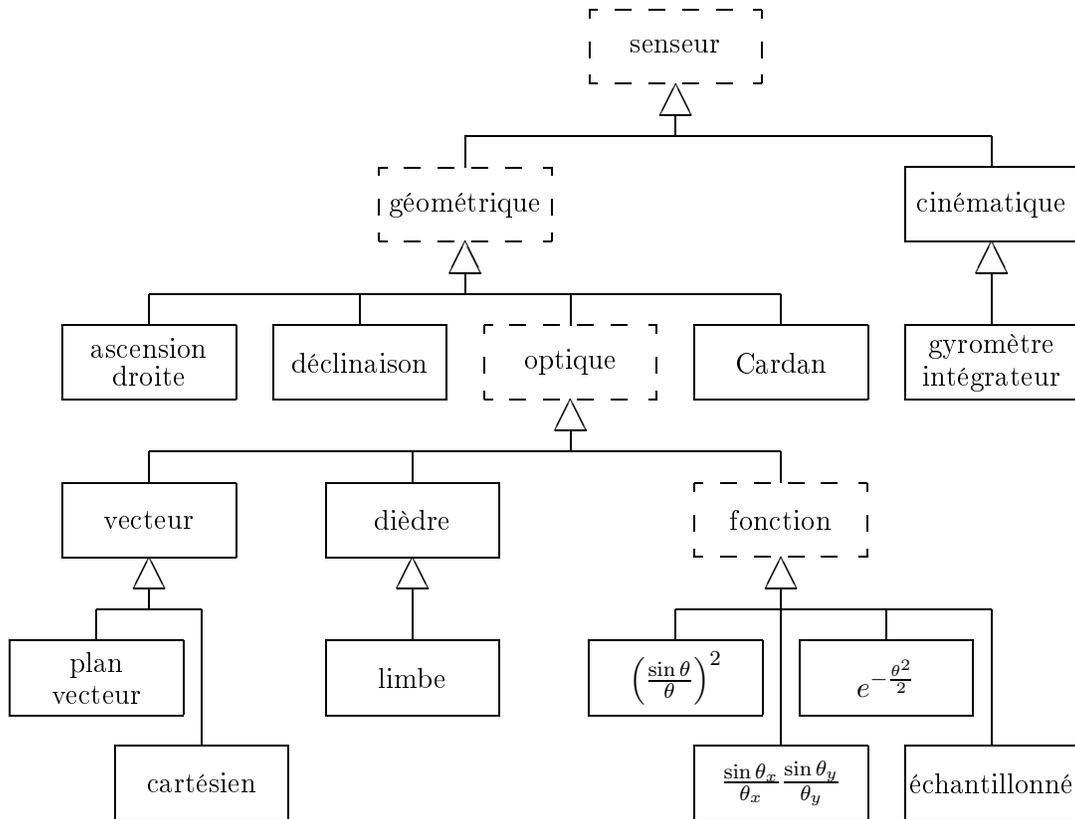
L'abstraction la plus générale correspond à la racine de l'arbre, les spécialisations les plus précises correspondant aux feuilles. Les sections suivantes décrivent les données de modélisation propre à chaque niveau. Le modèle complet d'un capteur Terre regroupe donc les données propres aux capteurs Terre, les données propres

¹²qui se retrouve au niveau de la conception dans un arbre d'héritage

aux capteurs dièdres, les données propres aux capteurs optiques, les données propres aux capteurs géométriques (il n'y en a pas) et les données générales à tous les types de capteurs.

Une remarque importante est que l'on considère dans la modélisation qu'un capteur produit une seule mesure. Pour modéliser un capteur réel produisant deux mesures (par exemple un capteur Terre mesurant le roulis et le tangage), on définira deux capteurs au sens de cette modélisation, un pour chacune des mesures.

FIG. 7 – Hiérarchie des capteurs



8.1 caractéristiques générales

Les capteurs sont des équipements de mesure implantés sur les plateformes des satellites. L'implantation se traduit au niveau du modèle par l'orientation du repère interne au capteur par rapport au repère satellite. Tous les axes seront ultérieurement décrits dans le repère capteur et c'est cette orientation qui permettra de les projeter en repère satellite. La manière la plus naturelle de définir cette orientation est de donner les coordonnées en repère satellite des axes \vec{i}_{capteur} , \vec{j}_{capteur} et \vec{k}_{capteur} , il y a bien d'autres manières de définir cette orientation, décrites dans la documentation utilisateur de la bibliothèque MARMOTTES.

Les mesures fournies par les capteurs sont des scalaires, principalement des angles ou des vitesses angulaires, mais parfois aussi des nombres sans dimension (capteurs cartésiens) ou des gains en dB. Ces scalaires ont une précision limitée, caractéristique du capteur. La précision est utilisée lors de l'évaluation de la convergence des algorithmes de résolution. L'erreur de convergence introduite par cette résolution dans un système sol se doit

d'être négligeable devant l'erreur intrinsèque du pilotage bord, le seuil de convergence doit donc être inférieur ou égal à la précision du capteur. Une convergence trop fine quant à elle augmente la durée des calculs. Le compromis utilisé par MARMOTTES est de considérer un seuil de convergence égal au dixième de la précision du capteur.

8.2 caractéristiques des capteurs géométriques

Nous avons vu que les consignes des capteurs géométriques étaient modélisées par des cônes. Il s'agit cependant là d'un calcul interne réalisé à chaque détermination d'orientation par la bibliothèque elle-même, et qui n'apparaît pas dans le fichier de description des capteurs.

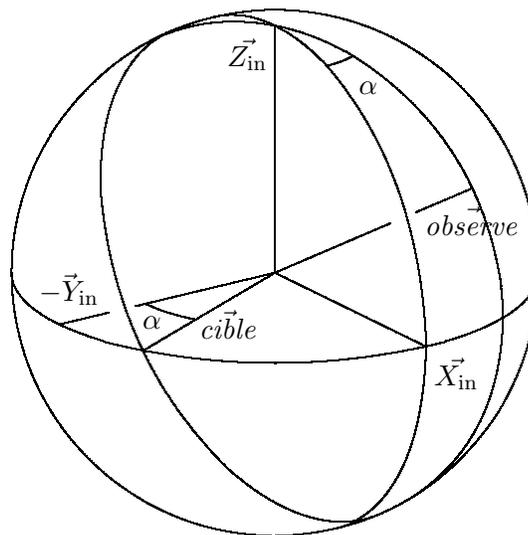
Ces capteurs n'ont pas de caractéristique propre visible pour l'utilisateur.

8.3 caractéristiques des capteurs d'ascension droite

Les capteurs *ascension* sont des pseudo-capteurs observant un vecteur défini en repère satellite (par exemple la direction de poussée d'un moteur), et mesurant son ascension droite par rapport au repère de Veis. Leur seule caractéristique spécifique est le vecteur observé (exprimé en repère capteur).

Comme nous l'avons vu à la section 3.1, page 10, on peut inverser la consigne. En effet la consigne brute consiste à obliger le vecteur *observe* à rester confiné dans un plan d'ascension droite donnée, plan obtenu en faisant tourner le plan $(\vec{X}_{in}, \vec{Z}_{in})$ d'un angle α autour de \vec{Z}_{in} . Ceci revient à dire que le vecteur *observe* doit être orthogonal à un certain vecteur *cible* défini comme étant orthogonal au plan d'ascension droite souhaité. Cette consigne est strictement équivalente à imposer au vecteur inertiel *cible* à rester confiné dans le plan orthogonal à *observe*, plan défini en repère satellite.

FIG. 8 – pseudo-capteur d'ascension droite



Cible pour la mesure α

$$cible \left\{ \begin{array}{l} \sin \alpha \\ -\cos \alpha \\ 0 \end{array} \right.$$

Cône de consigne pour la mesure α

$$\begin{aligned} a\vec{x}e &= observe \\ angle &= \frac{\pi}{2} \end{aligned}$$

Mesure produite

$$\begin{aligned} \vec{o} &= Q_{att}^{-1}(observe) \\ \alpha &= \text{atan2}(o_y, o_x) \end{aligned}$$

8.4 caractéristiques des capteurs de déclinaison

Les capteurs *déclinaison* sont des pseudo-capteurs observant un vecteur défini en repère satellite (par exemple la direction de poussée d'un moteur), et mesurant sa déclinaison par rapport au repère de Veis. Leur seule caractéristique spécifique est le vecteur observé (exprimé en repère capteur).

Comme nous l'avons vu à la section 3.1, page 10, on peut inverser la consigne. En effet la consigne brute consiste à obliger le vecteur *observe* à rester confiné dans un cône de déclinaison donnée, cône d'axe \vec{Z}_{in} . Ceci revient à dire que le vecteur *observe* doit être séparé de \vec{Z}_{in} d'un angle connu. Cette consigne est strictement équivalente à imposer au vecteur inertiel *cible* = \vec{Z}_{in} à rester confiné dans un cône d'axe *observe*, cône défini en repère satellite.

Il faut prendre garde avec les capteurs de déclinaison à ne pas utiliser des consignes valant $\pm\pi/2$. En effet le cône de consigne dégénère dans ce cas en un point situé sur l'axe des pôles, il ne s'agit donc plus d'une variété à une dimension et les algorithmes de résolution ne fonctionnent plus.

Cible pour la mesure δ

$$cible \left\{ \begin{array}{l} 0 \\ 0 \\ 1 \end{array} \right.$$

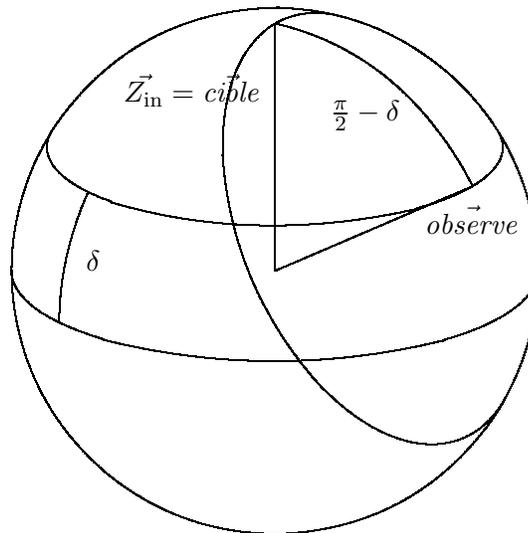
Cône de consigne pour la mesure δ

$$\begin{aligned} a\vec{x}e &= observe \\ angle &= \frac{\pi}{2} - \delta \end{aligned}$$

Mesure produite

$$\begin{aligned} \vec{o} &= Q_{att}^{-1}(observe) \\ \delta &= \arcsin(o_z) \end{aligned}$$

FIG. 9 – pseudo-capteur de déclinaison



8.5 caractéristiques des capteurs optiques

Les capteurs *optiques* sont des capteurs géométriques qui observent une cible, leurs caractéristiques spécifiques sont la cible et le champ de vue du capteur.

La liste des cibles connues de MARMOTTES à la date de rédaction de cette note est : **Soleil**, **pseudo-Soleil** (le **pseudo-Soleil** est visible même à travers la Terre et la Lune, alors que le **Soleil** est affecté par les éclipses), **Terre-Soleil** (la direction **Terre-Soleil** ne prend pas en compte la parallaxe due à la position du satellite, alors que la direction **Soleil** en tient compte), **Lune**, **Terre**, **nadir** (les points de la surface terrestre au nadir du satellite le voient exactement à leur verticale locale, en prenant l'ellipticité de la Terre en compte), **vitesse**, **moment** (*orbital*), **devant** (direction dans le plan de l'orbite perpendiculaire à la direction du centre Terre), **position** (qui indique que l'utilisateur donnera lui-même dans le code la position de la cible et qu'il faudra lui appliquer une correction de parallaxe), **direction** (qui indique que l'utilisateur donnera lui-même dans le code la direction de la cible et qu'il faudra la prendre telle qu'elle), **station** (la station elle-même est décrite dans un bloc séparé), **Polaris**, et **Canopus**¹³.

Deux cibles particulières de la liste précédente méritent une explication, les cibles **position** et **direction**. Ces cibles permettent de prendre en compte les capteurs susceptibles d'observer de nombreuses cibles différentes selon les conditions ou les besoins de l'utilisateur. La cible **position** est ainsi bien adaptée à la modélisation des pseudo-capteurs mesurant des gains d'antennes par rapport à un émetteur quelconque, éventuellement mobile (cf 8.12, 8.13 et 8.14). L'exemple type est le calcul du bilan de liaison d'un signal émis par un satellite GPS dont on recalcule la position à chaque instant. La cible **direction** est quant à elle plutôt destinée à modéliser des capteurs stellaires qui peuvent suivre des centaines voire des milliers d'étoiles différentes selon la richesse du catalogue embarqué. Pour modéliser de tels capteurs, une seule entrée dans le fichier de description est nécessaire, la cible pouvant être mise à jour à volonté pendant le déroulement du programme. Lorsqu'un tel capteur n'est utilisé qu'en temps que capteur de mesure et pas en consigne, on peut même tester de nombreuses cibles entre

¹³les deux derniers astres ont été rajoutés uniquement pour le plaisir ...

chaque détermination d'orientation, par exemple pour tester finement les étoiles visibles après un premier tri grossier.

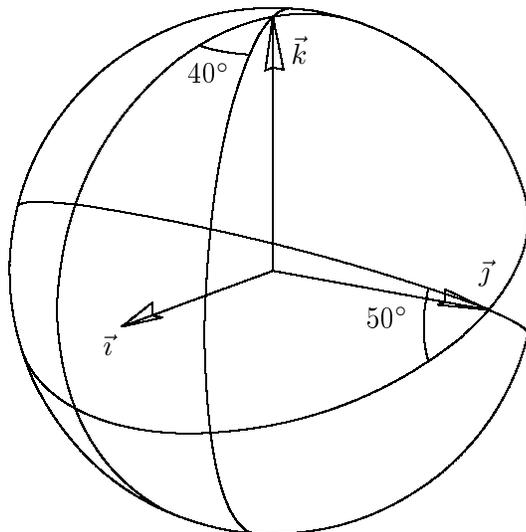
Un champ de vue est une zone de la sphère unité qui peut être simple, c'est alors un cône défini par son axe (en repère capteur, rappelons-le) et son demi-angle au sommet, ou composée à partir d'une ou de plusieurs autres zones. Dans la terminologie MARMOTTES, ces zones (simples et composées) sont dénommées des *champs*¹⁴. Les opérations de composition disponibles sont décrites dans la table 2.

TAB. 2 – opérations sur les zones géométriques de la sphère unité

nom	arguments	
intersection	champ	champ
réunion	champ	champ
différence	champ	champ
marge	scalaire	champ
déplacement	rotation	champ
balayage	rotation	champ

L'opération *marge* permet d'ajouter (ou de retirer selon le signe) une marge angulaire sur un champ. L'opération *balayage* permet de calculer le champ résultant de l'étalement de champ sur la sphère à l'aide d'une rotation (un peu à la manière d'un pinceau). Cette dernière opération est particulièrement utile pour définir les champs de vue résultant de balayages optiques de détecteurs.

FIG. 10 – champ de vue d'un capteur solaire



À titre d'exemple, voici comment on pourrait décrire le champ de vue du capteur solaire de la figure 10 :

¹⁴du nom de la classe `c++` qui les implante

dièdre d'axe \vec{k} :

intersection du cône $((\sin(20^\circ), \cos(20^\circ), 0), 90^\circ)$
avec le cône $((\sin(20^\circ), -\cos(20^\circ), 0), 90^\circ)$

dièdre d'axe \vec{j} :

intersection du cône $((\sin(25^\circ), 0, \cos(25^\circ)), 90^\circ)$
avec le cône $((\sin(25^\circ), 0, -\cos(25^\circ)), 90^\circ)$

champ de vue :

intersections des dièdres d'axes \vec{j} et \vec{k}

Cette description peut être facilement transcrite dans le fichier des capteurs, la syntaxe de ce fichier est décrite dans le manuel d'utilisation de MARMOTTES.

Dans tous les cas de champs de vue, la visibilité d'un astre considéré comme ponctuel est testée par l'inclusion du point dans le champ complet. Les capteurs *Terre*¹⁵ cependant ne se contentent pas d'observer un point de leur astre cible mais utilisent son limbe, leur champ de vue et leur logique de détermination de la visibilité sont donc plus complexes (voir section 8.10).

8.6 caractéristiques des capteurs vecteurs

Les capteurs *vecteurs* sont des capteurs optiques qui mesurent des angles entre l'astre observé et un vecteur de référence.

Le capteur est entièrement défini par l'axe de référence.

Un exemple classique concerne les satellites stabilisés par rotation, qui disposent de capteurs mesurant l'angle d'aspect solaire par rapport à l'axe de rotation. L'axe de référence est donc l'axe de rotation du satellite, qui dans la pratique dépend de l'implantation du capteur. Pour des raisons d'homogénéité la bibliothèque attend ce vecteur en repère capteur, l'utilisateur doit réaliser lui-même la projection entre les repères (ou dire à la bibliothèque que le repère capteur est similaire au repère satellite).

Le cône de consigne pour la mesure α s'exprime simplement

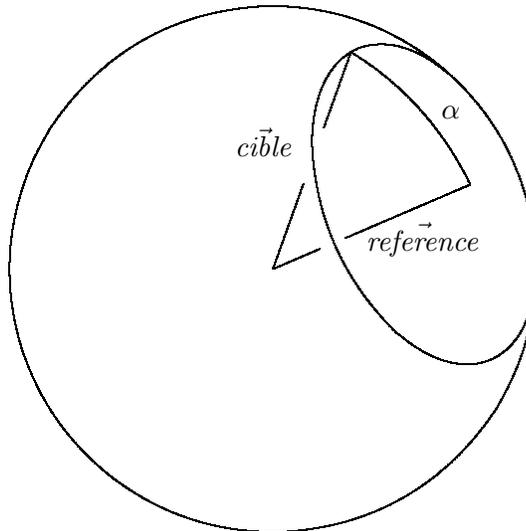
$$\begin{aligned} a\vec{x}e &= \vec{reference} \\ angle &= \alpha \end{aligned}$$

La mesure produite pour un astre cible dans la direction \vec{c} vaut

$$\begin{aligned} \vec{v} &= Q_{att}(\vec{c}) \\ \alpha &= \text{angle}(\vec{v}, \vec{reference}) \end{aligned}$$

¹⁵il s'agit là des capteurs explicitement déclarés de type *Terre*, un capteur de type *dièdre* ayant pour cible la Terre la considère ponctuelle, comme tous les autres astres

FIG. 11 – champ de vue d'un capteur vecteur



8.7 caractéristiques des capteurs plan-vecteurs (élévation)

Les capteurs *plan-vecteur*¹⁶ mesurent l'élévation par rapport à un plan. Ils sont directement issus des capteurs *vecteur*, à l'aide des règles suivantes :

$$\begin{aligned} \text{consigne}_{\text{élévation}}(\alpha) &= \text{consigne}_{\text{vecteur}}\left(\frac{\pi}{2} - \alpha\right) \\ \text{mesure}_{\text{élévation}} &= \frac{\pi}{2} - \text{mesure}_{\text{vecteur}} \end{aligned}$$

8.8 caractéristiques des capteurs cartésiens

Les capteurs *cartésiens* mesurent la projection d'une direction sur un axe. Ils sont directement issus des capteurs *vecteur*, à l'aide des règles suivantes :

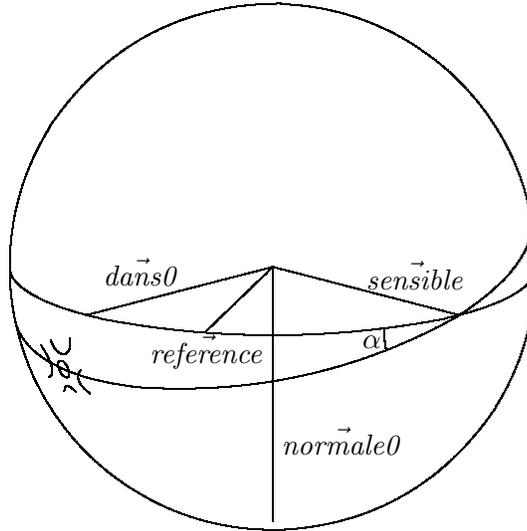
$$\begin{aligned} \text{consigne}_{\text{cartésien}}(x) &= \text{consigne}_{\text{vecteur}}(\arccos(x)) \\ \text{mesure}_{\text{cartésien}} &= \cos(\text{mesure}_{\text{vecteur}}) \end{aligned}$$

8.9 caractéristiques des capteurs dièdres

Les capteurs *dièdres* sont des capteurs optiques qui mesurent des angles entre un demi-plan de référence (passant par l'axe sensible) et un demi-plan défini par l'astre observé et l'axe sensible. Le capteur est entièrement défini par l'axe sensible et un vecteur quelconque du demi-plan de référence non aligné avec l'axe sensible. La mesure produite est un angle compris entre $-\pi$ et π , l'angle étant orienté du plan de référence vers le plan de l'astre conformément au sens de l'axe sensible (voir la figure 12, où l'angle mesuré, α , est positif).

¹⁶ce nom extrêmement laid a été choisi après bien des réflexions à la suite d'un conflit qu'introduisait le terme plus adapté *élévation*, si un lecteur désire proposer un meilleur vocable, l'auteur serait heureux de le savoir

FIG. 12 – mesure d'un capteur dièdre



Remarque : à l'intérieur de la bibliothèque, les caractéristiques **axe sensible** et **référence de zéro** sont converties dès la lecture en deux autres vecteurs **dans0** et **normale0**, plus pratiques pour les calculs. On utilise pour cela les relations :

$$\begin{aligned} \vec{normale0} &= \frac{\vec{sensible} \wedge \vec{reference0}}{\|\vec{sensible} \wedge \vec{reference0}\|^2} \\ \vec{dans0} &= \vec{normale0} \wedge \vec{sensible} \end{aligned}$$

Ces deux vecteurs permettent d'exprimer facilement le cône de consigne et de calculer une mesure :

Cône de consigne pour la mesure α

$$\begin{aligned} \vec{axe} &= \cos(\alpha)\vec{normale0} - \sin(\alpha)\vec{dans0} \\ \text{angle} &= \frac{\pi}{2} \end{aligned}$$

Mesure produite pour un astre cible dans la direction \vec{c}

$$\begin{aligned} \vec{v} &= Q_{att}(\vec{c}) \\ \alpha &= \text{atan2}(\vec{v} \cdot \vec{normale0}, \vec{v} \cdot \vec{dans0}) \end{aligned}$$

8.10 caractéristiques des capteurs Terre

Les capteurs *Terre* sont des capteurs dièdres mesurant la position de la Terre à l'aide du limbe. Ils ont la particularité de pouvoir être inhibés par la Lune ou le Soleil, l'inhibition par la Lune ne se produisant qu'en deçà d'un seuil angulaire¹⁷ paramétrable entre 0° et 180° (0° pour la pleine Lune —c'est-à-dire que la Lune

¹⁷l'angle considéré est l'angle Soleil-satellite-Lune

n'est jamais gênante— et 180° pour la nouvelle Lune —c'est-à-dire que la Lune est toujours gênante—. Les capteurs Terre ont donc les caractéristiques des capteurs dièdres (sauf la cible qui est obligatoirement la Terre), mais également un champ d'inhibition pour la Lune, un seuil de phase pour la Lune, et un champ d'inhibition pour le Soleil.

Les capteurs *Terre* n'observent pas directement le centre Terre mais plutôt les portions de son limbe qui apparaissent dans des parcelles différentes du champ de vue. Cette modélisation du limbe est faite en tenant compte de la position du satellite en connaissant la géométrie de la Terre. Dans l'implémentation actuelle réalisée au CNES, on utilise un modèle interne avec une Terre sphérique et un rayon infrarouge donné (qui est supérieur au rayon du géoïde à cause de l'effet de l'atmosphère), l'utilisateur ne paramètre rien de particulier. Il est envisagé de rendre ce modèle de corps central paramétrable afin de pouvoir utiliser de tels capteurs¹⁸ dans le cadre de missions autour d'autres corps.

De plus, pour produire des mesures globales (roulis et tangage par exemple), ces capteurs doivent combiner les informations issues de chaque parcelle. Ce principe est illustré par le champ de vue de la figure 13 : pour produire une mesure de roulis le capteur combine soit les informations provenant des parcelles 1 et 2 soit les informations provenant des parcelles 3 et 4, pour produire une mesure de tangage le capteur combine soit les informations provenant des parcelles 1 et 3 soit les informations provenant des parcelles 2 et 4. Dans le cas présenté sur la figure, le capteur roulis voit la Terre alors que le capteur tangage ne la voit pas.

Cette logique de mesure a un impact sur la visibilité de la Terre considérée comme non ponctuelle, les champs de vue des capteurs *Terre* sont donc d'un type plus riche que les champs de vue classiques définis par des champs, on les dénomme des *parcelles*¹⁹. Une parcelle peut être un simple champ ou une combinaison de parcelles à l'aide des opérations décrites dans la table 3.

TAB. 3 – opérations sur les zones logiques de la sphère unité

nom	arguments	
et	parcelle	parcelle
ou	parcelle	parcelle
marge	scalaire	parcelle

8.11 caractéristiques des capteurs fonctions

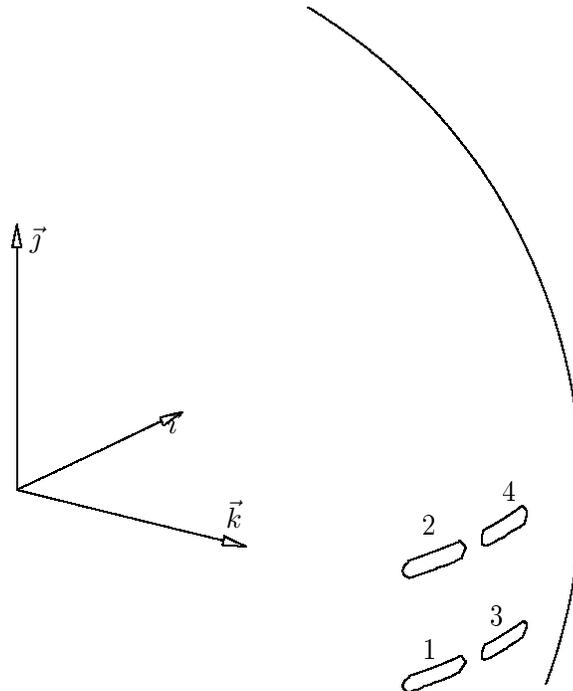
Les capteurs de *fonction* sont des pseudo-capteurs qui ne sont *a priori* pas destinés à être utilisés en consigne mais uniquement en mesure. Ils permettent de modéliser une fonction quelconque sur la sphère unité.

Ces capteurs doivent uniquement être considérés comme l'abstraction commune qui généralise les capteurs de gain d'antenne échantillonnés, en $(\sin \theta / \theta)^2$ et en $e^{-\frac{\theta^2}{2}}$.

¹⁸il faudrait alors changer leur nom ...

¹⁹du nom des classes C++ `Parcelle`, `ParcelleElementaire`, `ReunionEtParcelles`, et `ReunionOuParcelles` qui les implémentent

FIG. 13 – champ de vue d'un capteur Terre à quatre scans



8.12 caractéristiques des capteurs de gain échantillonnés

Les capteurs de gain échantillonnés modélisent des gains d'antennes pour une forme de lobe échantillonnée selon l'angle de dépointage par rapport à l'axe. Ils permettent de calculer la part liée uniquement à l'orientation dans un bilan de liaison. La forme du gain modélisée par ce senseur est une simple interpolation linéaire selon la valeur θ du dépointage entre l'axe d'antenne et la cible par rapport aux points spécifiés par l'utilisateur dans le fichier de description. En dehors de l'intervalle spécifié (c'est-à-dire entre 0 et le plus petit dépointage échantillonné et entre le plus grand dépointage échantillonné et π), le gain est considéré constant. La valeur retournée a la même dimension que ce que l'utilisateur a mis dans son fichier, il s'agit généralement directement de valeurs en dB.

La fonction n'étant pas inversible dans le cas général, on ne *peut pas utiliser ces capteurs en consigne*, ils sont destinés à être utilisés en *mesure uniquement*.

Ces capteurs sont caractérisés par l'axe d'antenne, qui correspond à la valeur maximale du gain et par le gain maximal.

8.13 caractéristiques des capteurs de gain en $(\sin(\theta/\theta_0)/(\theta/\theta_0))^2$

Les capteurs de gain en $(\sin(\theta/\theta_0)/(\theta/\theta_0))^2$ modélisent des gains d'antennes pour la forme de lobe spécifiée. Ils permettent de calculer la part liée uniquement à l'orientation dans un bilan de liaison. La forme du gain

modélisée par ce capteur est :

$$g = 10 \times \frac{\log K \left(\frac{\sin(\theta/\theta_0)}{\theta/\theta_0} \right)^2}{\log 10}$$

où θ est le dépointage entre l'axe d'antenne et la cible. La valeur retournée est donc directement une valeur en dB.

La fonction n'étant pas inversible (à moins de se restreindre au lobe primaire), on ne *peut pas utiliser ces capteurs en consigne*, ils sont destinés à être utilisés en *mesure uniquement*.

Ces capteurs sont caractérisés par l'axe d'antenne, qui correspond à la valeur maximale du gain et par le gain maximal.

8.14 caractéristiques des capteurs de gain en fonction gaussienne

Les capteurs de gain en $e^{-\frac{(\theta/\theta_0)^2}{2}}$ modélisent des gains d'antennes pour la forme de lobe spécifiée. Ils permettent de calculer la part liée uniquement à l'orientation dans un bilan de liaison. La forme du gain modélisée par ce capteur est :

$$g = 10 \times \frac{\log K e^{-\frac{(\theta/\theta_0)^2}{2}}}{\log 10}$$

où θ est le dépointage entre l'axe d'antenne et la cible. La valeur retournée est donc directement une valeur en dB.

La fonction est inversible, mais dans un souci d'homogénéité avec les autres capteurs de gain nous ont poussés à *interdire l'utilisation de ces capteurs en consigne*, ils sont destinés à être utilisés en *mesure uniquement*.

Ces capteurs sont caractérisés par l'axe d'antenne, qui correspond à la valeur maximale du gain et par le gain maximal.

8.15 caractéristiques des capteurs de Cardan

Les capteurs de *Cardan* mesurent les rotations successives qui permettent de passer d'un repère de base au repère satellite ($\vec{X}_{\text{sat}}, \vec{Y}_{\text{sat}}, \vec{Z}_{\text{sat}}$). Plusieurs repères de base sont disponibles, le repère géocentrique ($\vec{X}_{\text{in}} = \vec{s}, \vec{Y}_{\text{in}} = -\vec{w}, \vec{Z}_{\text{in}} = -\vec{q}$)²⁰, le repère orbital ($\vec{t}, \vec{n}, \vec{w}$), le repère orbital ($\vec{q}, \vec{s}, \vec{w}$), le repère inertiel de référence (dans lequel sont déjà exprimées les positions et vitesses), le repère topocentrique ($\vec{X}_{\text{in}} = -\vec{q}, \vec{Y}_{\text{in}} = \vec{E}st$), et un repère utilisateur qui peut être modifié dynamiquement entre chaque calcul.

Ces capteurs sont très utiles pour simplifier la modélisation. Ce ne sont pas systématiquement des pseudo-capteurs, certaines plateformes étant capable de calculer les angles à partir d'information provenant d'autres capteurs élémentaires et d'un modèle d'orbite et de les fournir dans la télémétrie. Il arrive d'ailleurs que dans ce cas les mesures élémentaires utilisées ne soient pas retransmises au sol, seules les angles calculés sont disponibles.

De nombreuses définitions peuvent être données pour les angles, selon l'ordre dans lequel ils sont appliqués. Si cet ordre n'a qu'une importance minimale lorsque les angles sont petits (les petites rotations sont *presque*

²⁰c'est-à-dire que lorsque les angles sont nuls, l'axe \vec{Z}_{sat} est pointé Terre, et l'axe \vec{X}_{sat} est dans le plan de l'orbite dans le sens de la vitesse

commutatives), il n'en est pas de même lorsque les rotations sont importantes. MARMOTTES supporte tous les ordres possibles.

Les seules définitions couramment acceptées par tous²¹ sont les dénominations des rotations :

TAB. 4 – rotations de CARDAN

nom	angle	axe
roulis	φ	$+\vec{X}$
tangage	θ	$+\vec{Y}$
lacet	ψ	$+\vec{Z}$

Les seules caractéristiques spécifiques des capteurs de CARDAN sont le repère de base, l'angle concerné et le jeu (c'est-à-dire l'ordre) auquel il appartient. La première caractéristique est définie dans le champ *reference* du capteur, les deux dernières caractéristiques étant réunies dans le champ *genre* où l'angle est écrit en toutes lettres et le jeu est décrit par les initiales des angles dans leur ordre d'application (RTL-roulis, RTL-tangage, RTL-lacet, TLR-tangage²², ...). Tous ces mots-clefs peuvent être modifiés par l'utilisateur.

La modélisation des consignes et le calcul des mesures pour tous les types d'angles est donnée dans l'annexe I (page 73). Ces formules sont applicables quel que soit le repère de base, il suffit en effet de calculer les coordonnées des axes canoniques du repère de base en repère inertiel (c'est-à-dire les coordonnées des vecteurs \vec{X}_{in} , \vec{Y}_{in} et \vec{Z}_{in}).

8.16 caractéristiques des capteurs cinématiques

Les capteurs *cinématiques* mesurent des vitesses de rotation instantanées autour d'un axe sensible. Cet axe est leur seule caractéristique spécifique.

8.17 caractéristiques des gyromètres intégrateurs

Les *gyromètres intégrateurs* mesurent l'intégrale des vitesses de rotation instantanées autour d'un axe sensible. Les spécificités de leur modélisation se limitent à la valeur de la mesure à une date donnée et à cette date.

L'intégration réalisée à bord par ces capteurs n'est pas complètement prédéfinie. Le centre de contrôle peut à tout moment réinitialiser la valeur intégrée par télécommande (*gyro resets*). Ces réinitialisations par le centre de contrôle sont utilisées dans les procédures nominales au cours de la mise à poste par exemple. Dans ces conditions, la simulation faite au sol doit pouvoir reproduire ce comportement. Les valeurs initiales utilisés sont simplement un angle nul à la date de la première utilisation du capteur dans la simulation, et ces valeurs sont mises à jour soit par intégration classique à chacun des pas ultérieurs (intégration réalisée directement par le modèle de capteur lorsqu'on lui transmet l'état en fin de résolution), soit à l'occasion des réinitialisations simulées.

²¹ il y a tout de même au moins une exception : les CVI Ariane 4

²² en anglais on trouvera bien-sûr : RPY-roll, RPY-pitch, RPY-yaw, PYR-pitch, ...

9 Conclusion

Les méthodes qui ont été décrites permettent de réaliser des simulateurs d'évolution d'orientation dans la composante sol des systèmes spatiaux. Elles ne répondent pas à tous les besoins liés à l'orientation des satellites. Ni les erreurs de mesures ni les actionneurs ne sont pris en compte ; les études nécessitant leur modélisation doivent donc ajouter leurs propres algorithmes, même si elles peuvent utiliser également la base présentée. Ces méthodes ne sont pas non plus utilisables à bord, elles sont essentiellement orientées vers de la simulation.

Ces méthodes s'appuient sur une description de l'orientation des satellites très simple et proche des concepts physiques familiers aux ingénieurs en mécanique spatiale. Cette description permet de séparer clairement la description et la détermination de l'orientation, ce qui permet la mise au point d'outils multi-missions.

Ces méthodes sont implémentées dans une bibliothèque validée opérationnellement. De nombreux projets en ont bénéficié depuis plusieurs années. Cette bibliothèque est diffusée librement par le CNES²³.

²³<http://logiciels.cnes.fr>

Annexes

A Quaternions

L'orientation d'un satellite est déterminée par le changement de repère qui permet d'exprimer un vecteur fixe en repère satellite connaissant ses coordonnées inertielles ou inversement. Dans \mathbb{R}^3 , il existe toujours une rotation autour d'un axe qui permet de transformer un repère en un autre, on représente donc une orientation par une rotation. Numériquement, la rotation est stockée sous forme de tableaux de nombres qui peuvent être :

- la matrice de rotation (tableau de 9 nombres vérifiant 6 relations) ;
- les angles d'Euler (3 nombres) ;
- l'axe et l'angle de rotation (4 nombres vérifiant une relation) ;
- un quaternion (4 nombres vérifiant une relation).

Dans tous les cas, l'objet mathématique créé a trois degrés de liberté, mais ils sont plus ou moins dilués dans tous les nombres qui composent l'objet. Le choix d'un formalisme ou d'un autre se fait en regardant le nombre de composants, la simplicité des opérations (composition de rotations, application d'une rotation à un vecteur, extraction de l'axe et de l'angle de rotation), l'existence de singularités.

Nous avons opté pour une représentation par quaternions principalement pour la simplicité des opérations (surtout quand on considère que nous traitons simultanément et de façon analytique les quaternions et leurs dérivées première et seconde par rapport à une variable) et la facilité d'extraction de l'axe et de l'angle de rotation. Le quaternion est ainsi à mi-chemin entre un formalisme tout linéaire (les matrices) où les éléments canoniques (axe et angle) sont cachés, et un formalisme canonique (angles d'Euler ou axe et angle de rotations) qui réclame des opérations trigonométriques directes et inverses. Un quaternion est équivalent à un axe et un angle de rotation où les calculs trigonométriques sont déjà faits.

Soit une rotation d'axe \vec{u} et d'angle θ . Si l'on considère que la rotation fait tourner des vecteurs de l'angle $+\theta$ dans un repère fixe, le quaternion associé Q est :

$$Q \begin{cases} q_0 &= \cos\left(-\frac{\theta}{2}\right) \\ q_1 &= \sin\left(-\frac{\theta}{2}\right)u_x \\ q_2 &= \sin\left(-\frac{\theta}{2}\right)u_y \\ q_3 &= \sin\left(-\frac{\theta}{2}\right)u_z \end{cases}$$

Nous définissons l'orientation par la rotation qui appliquée aux coordonnées inertielles d'un vecteur produit les coordonnées en repère satellite du même vecteur. Avec cette convention, l'angle θ utilisé dans les formules précédentes est *l'opposé* de l'angle dont tourne le satellite lorsqu'il passe d'une orientation alignée avec le repère inertiel à son orientation finale.

On trouve souvent dans la littérature une formulation sans le signe « $-$ ». Ceci est dû au fait que d'emblée les auteurs considèrent que c'est le *repère* qui tourne de θ . Les vecteurs vus dans ce repère semblent donc tourner de $-\theta$, d'où notre notation²⁴.

L'axe \vec{u} étant invariant par la rotation, les coordonnées (u_x, u_y, u_z) peuvent indifféremment être exprimées dans le repère inertiel ou dans le repère satellite.

²⁴nous présentons le quaternion d'une façon générale par les rotations et non pas seulement pour le traitement de l'orientation

On remarque que le signe de Q est indifférent ; pour inverser la rotation, il faut soit changer le signe de q_0 , soit changer les signes de q_1, q_2, q_3 .

La relation vérifiée par les composantes du quaternion est :

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$$

On notera \vec{q} le vecteur de coordonnées : $(q_1, q_2, q_3)^\top$.

Si la rotation R'' est obtenue par deux rotations successives R puis R' –c'est-à-dire que pour tout vecteur on a $R''(\vec{u}) = R'(R(\vec{u}))$ –, représentées par les quaternions Q et Q' , on calcule le quaternion Q'' de la rotation composée par :

$$Q'' = QQ' \quad \left| \begin{array}{l} q_0'' = q_0 q_0' - \vec{q} \cdot \vec{q}' \\ \vec{q}'' = q_0 \vec{q}' + q_0' \vec{q} + \vec{q} \wedge \vec{q}' \end{array} \right.$$

On remarque que cette multiplication n'est pas commutative, et que l'élément neutre (à droite comme à gauche) est le quaternion Q_I qui représente l'identité :

$$Q_I \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Attention : La multiplication des quaternions se fait en sens inverse de la multiplication des matrices : le quaternion associé à R suivie de R' est le quaternion QQ' alors que la matrice est $M'M$. C'est pour éviter toute confusion que dans la bibliothèque CANTOR on ne définit pas l'opérateur de multiplication (`operator *`) mais l'opérateur de composition (`operator ()`), qui conserve la dissymétrie intrinsèque de l'opération. Un utilisateur de CANTOR en C++ écrirait donc directement : `Q_seconde = Q_prime (Q)`.

Si \vec{u} est un vecteur dont on connaît les coordonnées, que l'on veut les coordonnées de son image \vec{v} par la rotation, et si Q est le quaternion qui représente l'orientation, alors on a :

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = (2q_0^2 - 1) \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} - 2q_0 \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} \wedge \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} + 2(q_1 u_x + q_2 u_y + q_3 u_z) \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

Remarque : Contrairement aux opérations matricielles, il est moins coûteux de composer deux rotations que d'appliquer une rotation à un vecteur.

On peut calculer la matrice M de rotation (telle que $[v] = M[u]$) à partir de Q par la relation :

$$M = \begin{bmatrix} 2(q_0^2 + q_1^2) - 1 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & 2(q_0^2 + q_2^2) - 1 & 2(q_2 q_3 + q_0 q_1) \\ 2(q_1 q_3 + q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & 2(q_0^2 + q_3^2) - 1 \end{bmatrix}$$

Pour calculer Q à partir de M , on remarque que des combinaisons des éléments diagonaux permettent d'exprimer au choix soit q_0^2 , soit q_1^2 , soit q_2^2 , soit q_3^2 . On calcule donc un des q_i et on déduit les autres des

éléments hors diagonale. Il y a donc quatre façons différentes de calculer Q (qui reviennent au même au signe près, mais nous avons vu que le signe était indifférent). On choisit le q_i que l'on va calculer en premier de façon à avoir q_i^2 suffisamment grand. En effet, la racine carrée est très sensible aux erreurs numériques lorsque son argument est proche de 0, et comme $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$, on est assuré qu'au moins un des q_i^2 sera supérieur à $\frac{1}{4}$. On calcule donc Q en suivant l'algorithme suivant :

Calculer $s_0 = M_{11} + M_{22} + M_{33}$

Si $s_0 > -0,1$

Calculer $q_0 = \frac{\sqrt{1+s_0}}{2}$

En déduire $q_1 = \frac{M_{23}-M_{32}}{4q_0}$, $q_2 = \frac{M_{31}-M_{13}}{4q_0}$, $q_3 = \frac{M_{12}-M_{21}}{4q_0}$

Sinon, calculer $s_1 = M_{11} - M_{22} - M_{33}$

Si $s_1 > -0,1$

Calculer $q_1 = \frac{\sqrt{1+s_1}}{2}$

En déduire $q_0 = \frac{M_{23}-M_{32}}{4q_1}$, $q_2 = \frac{M_{12}+M_{21}}{4q_1}$, $q_3 = \frac{M_{13}+M_{31}}{4q_1}$

Sinon, calculer $s_2 = -M_{11} + M_{22} - M_{33}$

Si $s_2 > -0,1$

Calculer $q_2 = \frac{\sqrt{1+s_2}}{2}$

En déduire $q_0 = \frac{M_{31}-M_{13}}{4q_2}$, $q_1 = \frac{M_{12}+M_{21}}{4q_2}$, $q_3 = \frac{M_{32}+M_{23}}{4q_2}$

Sinon, calculer $s_3 = -M_{11} - M_{22} + M_{33}$ (on a obligatoirement $s_3 > -0,1$)

Calculer $q_3 = \frac{\sqrt{1+s_3}}{2}$

En déduire $q_0 = \frac{M_{12}-M_{21}}{4q_3}$, $q_1 = \frac{M_{13}+M_{31}}{4q_3}$, $q_2 = \frac{M_{32}+M_{23}}{4q_3}$

Fin si

Fin si

Fin si

Dans la pratique, la matrice M peut être légèrement différente d'une matrice orthogonale (par exemple quand la matrice est écrite avec une précision limitée). Il faut alors trouver la matrice orthogonale M' la plus proche de M (par exemple au sens de la norme de Frobenius) avant d'en extraire les coefficients du quaternion. On trouve M' par la méthode itérative suivante :

$$A_0 = M$$

$$A_{n+1} = A_n - \frac{A_n M^T A_n - M}{2}$$

Le critère d'arrêt des itérations est $|f_n - f_{n-1}| < \varepsilon$, avec $f_n = \|M - A_n\|^2$.

On prend alors $M' = A_n$.

Les formules de passage des quaternions aux angles d'Euler sont :

$$q_0 = \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{\psi + \varphi}{2}\right)$$

$$q_1 = \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{\psi - \varphi}{2}\right)$$

$$q_2 = \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{\psi - \varphi}{2}\right)$$

$$q_3 = \cos\left(\frac{\theta}{2}\right) \sin\left(\frac{\psi + \varphi}{2}\right)$$

Si on s'intéresse à la cinématique de l'orientation, on a la relation suivante :

$$\dot{Q} = MQ$$

$$M = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix}$$

où (p, q, r) sont les coordonnées en repère satellite (c'est-à-dire dans le repère qui tourne) du vecteur rotation instantanée²⁵. Cette relation permet de calculer pas à pas l'évolution de Q lorsque (p, q, r) varient dans le temps. S'ils sont fixes, il est plus pratique de construire le quaternion d'évolution $E(t - t_0)$:

$$E(t - t_0) = \begin{bmatrix} \cos\left(\frac{\omega}{2}(t - t_0)\right) & & & \\ \frac{1}{\omega} \sin\left(\frac{\omega}{2}(t - t_0)\right) & \begin{bmatrix} p \\ q \\ r \end{bmatrix} & & \end{bmatrix}$$

avec $\omega = \sqrt{p^2 + q^2 + r^2}$

On a alors : $Q(t) = Q(t_0)(E(t - t_0))$ par la composition des rotations.

Remarque : La formule précédente, exprimée sous forme matricielle, donne :

$$Q(t) = G(t - t_0)Q(t_0)$$

dans laquelle $G(t - t_0)$ est l'exponentielle matricielle de M . On retrouve donc ici la simplification d'écriture apportée par les quaternions.

²⁵c'est-à-dire que (p, q, r) sont les coordonnées de l'axe de rotation instantanée tel qu'il est retourné par **Marmottes**

B Méthode des trièdres

B.1 formalisme matriciel

On connaît deux vecteurs normés \vec{v}_1 et \vec{v}_2 à la fois en repère inertiel et en repère satellite. On veut en déduire la matrice de changement de repère.

À partir des vecteurs \vec{v}_1 et \vec{v}_2 (\vec{v}_2 différent de $\pm\vec{v}_1$), on calcule le trièdre orthonormé $(\vec{v}'_1, \vec{v}'_2, \vec{v}'_3)$ par :

$$\begin{aligned}\vec{v}'_1 &= \vec{v}_1 \\ \vec{v}'_2 &= \frac{\vec{v}_1 \wedge \vec{v}_2}{\sqrt{1 - (\vec{v}_1 \cdot \vec{v}_2)^2}} \\ \vec{v}'_3 &= \vec{v}'_1 \wedge \vec{v}'_2\end{aligned}$$

Ces vecteurs ont pour coordonnées les matrices colonnes r_1, r_2, r_3 en repère satellite et s_1, s_2, s_3 en repère inertiel.

$$M = [r_1 | r_2 | r_3] \times \begin{bmatrix} s_1^\top \\ s_2^\top \\ s_3^\top \end{bmatrix} \Rightarrow [A]_{sat} = M[A]_{inertiel}$$

B.2 formalisme des quaternions

Connaissant $\vec{u}_1, \vec{u}_2, \vec{v}_1, \vec{v}_2$, trouver Q tel que $\vec{u}_1 \rightarrow \vec{v}_1$.
 $\vec{u}_2 \rightarrow \vec{v}_2$

La formule générale de calcul de $\vec{v} = Q(\vec{u})$ s'exprime $\vec{v} = 2[(\vec{q} \cdot \vec{u})\vec{q} + q_0^2\vec{u} - q_0(\vec{q} \wedge \vec{u})] - \vec{u}$, d'où :

$$\vec{v} - \vec{u} = 2[(\vec{q} \cdot \vec{u})\vec{q} - \|\vec{q}\|^2\vec{u} - q_0(\vec{q} \wedge \vec{u})] \quad (22)$$

on pose $\vec{k} = (\vec{v}_1 - \vec{u}_1) \wedge (\vec{v}_2 - \vec{u}_2)$, en appliquant deux fois l'équation 22, on trouve :

$$\begin{aligned}\frac{\vec{k}}{4} &= \|\vec{q}\|^4 \vec{u}_1 \wedge \vec{u}_2 + q_0 \|\vec{q}\|^2 \vec{u}_1 \wedge (\vec{q} \wedge \vec{u}_2) - \|\vec{q}\|^2 (\vec{q} \cdot \vec{u}_2) \vec{u}_1 \wedge \vec{q} \\ &+ q_0 \|\vec{q}\|^2 (\vec{q} \wedge \vec{u}_1) \wedge \vec{u}_2 + q_0^2 (\vec{q} \wedge \vec{u}_1) \wedge (\vec{q} \wedge \vec{u}_2) - q_0 (\vec{q} \cdot \vec{u}_2) (\vec{q} \wedge \vec{u}_1) \wedge \vec{q} \\ &- \|\vec{q}\|^2 (\vec{q} \cdot \vec{u}_1) \vec{q} \wedge \vec{u}_2 - q_0 (\vec{q} \cdot \vec{u}_1) \vec{q} \wedge (\vec{q} \wedge \vec{u}_2) + (\vec{q} \cdot \vec{u}_1) (\vec{q} \cdot \vec{u}_2) \vec{q} \wedge \vec{q}\end{aligned} \quad (23)$$

les formules du double produit vectoriel donnent :

$$\left\{ \begin{array}{ll} \vec{u}_1 \wedge (\vec{q} \wedge \vec{u}_2) &= (\vec{u}_1 \cdot \vec{u}_2)\vec{q} - (\vec{u}_1 \cdot \vec{q})\vec{u}_2 \\ (\vec{q} \wedge \vec{u}_1) \wedge \vec{u}_2 &= (\vec{u}_2 \cdot \vec{q})\vec{u}_1 - (\vec{u}_1 \cdot \vec{u}_2)\vec{q} \\ (\vec{q} \wedge \vec{u}_1) \wedge (\vec{q} \wedge \vec{u}_2) &= [(\vec{q} \wedge \vec{u}_1) \cdot \vec{u}_2]\vec{q} - [(\vec{q} \wedge \vec{u}_1) \cdot \vec{q}]\vec{u}_2 \\ &= [\vec{q} \cdot (\vec{u}_1 \wedge \vec{u}_2)]\vec{q} \\ (\vec{q} \wedge \vec{u}_1) \wedge \vec{q} &= \|\vec{q}\|^2 \vec{u}_1 - (\vec{u}_1 \cdot \vec{q})\vec{q} \\ \vec{q} \wedge (\vec{q} \wedge \vec{u}_2) &= (\vec{u}_2 \cdot \vec{q})\vec{q} - \|\vec{q}\|^2 \vec{u}_2 \end{array} \right.$$

que l'on introduit dans la relation 23, ce qui donne, d'une part, après quelques simplifications :

$$\begin{aligned} \frac{\vec{k}}{4} &= \|\vec{q}\|^4 \vec{u}_1 \wedge \vec{u}_2 + q_0 \|\vec{q}\|^2 ((\vec{u}_2 \cdot \vec{q}) \vec{u}_1 - (\vec{u}_1 \cdot \vec{q}) \vec{u}_2) \\ &\quad + \|\vec{q}\|^2 ((\vec{u}_2 \cdot \vec{q}) \vec{q} \wedge \vec{u}_1 - (\vec{u}_1 \cdot \vec{q}) \vec{q} \wedge \vec{u}_2) \\ &\quad + q_0^2 (\vec{q} \cdot (\vec{u}_1 \wedge \vec{u}_2)) \vec{q} - q_0 \|\vec{q}\|^2 ((\vec{u}_2 \cdot \vec{q}) \vec{u}_1 - (\vec{u}_1 \cdot \vec{q}) \vec{u}_2) \end{aligned} \quad (24)$$

on a d'autre part :

$$\begin{aligned} (\vec{u}_2 \cdot \vec{q}) \vec{q} \wedge \vec{u}_1 &= \vec{u}_2 \wedge [(\vec{q} \wedge \vec{u}_1) \wedge \vec{q}] + [\vec{u}_2 \cdot (\vec{q} \wedge \vec{u}_1)] \vec{q} \\ &= \vec{u}_2 \wedge [\|\vec{q}\|^2 \vec{u}_1 - (\vec{u}_1 \cdot \vec{q}) \vec{q}] + [\vec{u}_2 \cdot (\vec{q} \wedge \vec{u}_1)] \vec{q} \end{aligned}$$

d'où :

$$\begin{aligned} (\vec{u}_2 \cdot \vec{q}) \vec{q} \wedge \vec{u}_1 - (\vec{u}_1 \cdot \vec{q}) \vec{q} \wedge \vec{u}_2 &= \|\vec{q}\|^2 \vec{u}_2 \wedge \vec{u}_1 + (\vec{u}_2 \cdot (\vec{q} \wedge \vec{u}_1)) \vec{q} \\ &= \|\vec{q}\|^2 \vec{u}_2 \wedge \vec{u}_1 + \vec{q} \cdot (\vec{u}_1 \wedge \vec{u}_2) \vec{q} \end{aligned}$$

on introduit cette expression dans l'équation 24 pour obtenir :

$$\begin{aligned} \frac{\vec{k}}{4} &= \|\vec{q}\|^4 \vec{u}_1 \wedge \vec{u}_2 + \|\vec{q}\|^4 \vec{u}_2 \wedge \vec{u}_1 + \|\vec{q}\|^2 (\vec{q} \cdot (\vec{u}_1 \wedge \vec{u}_2)) \vec{q} + q_0^2 (\vec{q} \cdot (\vec{u}_1 \wedge \vec{u}_2)) \vec{q} \\ \vec{k} &= 4(\vec{q} \cdot (\vec{u}_1 \wedge \vec{u}_2)) \vec{q} \end{aligned}$$

On déduit de cette expression :

$$\begin{aligned} c &= \vec{k} \cdot (\vec{u}_1 \wedge \vec{u}_2) \\ &= 4[\vec{q} \cdot (\vec{u}_1 \wedge \vec{u}_2)]^2 \\ \Rightarrow \vec{q} &= \frac{\vec{k}}{2\sqrt{c}} \end{aligned} \quad (25)$$

Si \vec{q} n'est pas dans le plan (\vec{u}_1, \vec{u}_2) , alors c est positif et l'expression 25 est bien définie et permet de calculer \vec{q} . Le signe global du quaternion étant indifférent, le choix de $\frac{\vec{k}}{2\sqrt{c}}$ au lieu de $\frac{-\vec{k}}{2\sqrt{c}}$ n'a pas d'autre conséquence que de figer le signe de q_0 qui sera déterminé par la suite.

Pour calculer q_0 , il suffit de remarquer que :

$$\begin{aligned} \vec{v} \cdot (\vec{u} \wedge \vec{q}) &= 2q_0 (\vec{u} \wedge \vec{q})^2 \\ \Rightarrow q_0 &= \frac{\vec{v} \cdot (\vec{u} \wedge \vec{q})}{2(\vec{u} \wedge \vec{q})^2} \end{aligned} \quad (26)$$

L'expression 26 est définie lorsque \vec{u} n'est pas aligné avec \vec{q} . Si on choisit pour couple (\vec{u}, \vec{v}) l'un des couples ayant servi à déterminer \vec{q} , on sait que \vec{q} n'étant pas dans le plan (\vec{u}, \vec{u}') ne peut être aligné avec \vec{u} et donc q_0 est calculable.

Une rotation étant une isométrie, on sait que le vecteur image de $\vec{u}_3 = \frac{\vec{u}_1 \wedge \vec{u}_2}{\sqrt{1 - (\vec{u}_1 \cdot \vec{u}_2)^2}}$ par la rotation est $\vec{v}_3 = \frac{\vec{v}_1 \wedge \vec{v}_2}{\sqrt{1 - (\vec{v}_1 \cdot \vec{v}_2)^2}}$. Cette remarque permet de résoudre le cas particulier où \vec{q} est dans le plan (\vec{u}_1, \vec{u}_2) , en reprenant les calculs à partir des couples (\vec{u}_1, \vec{v}_1) et (\vec{u}_3, \vec{v}_3) . Si après ce changement, \vec{q} est encore dans le plan

(\vec{u}_1, \vec{u}_3) , alors \vec{q} était aligné avec \vec{u}_1 ; là encore on peut changer les couples de vecteurs et utiliser (\vec{u}_2, \vec{u}_3) , et (\vec{v}_2, \vec{v}_3) pour tenter de résoudre le problème. Si c persiste à être nul, on est dans le cas particulier de l'identité,

on retourne alors le quaternion $Q_I \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$.

Les formules précédentes supposent que les vecteurs $\vec{u}_1, \vec{u}_2, \vec{v}_1, \vec{v}_2$ sont normés. Il faut donc les normaliser avant d'exécuter l'algorithme.

La rotation qui amène \vec{u}_1 en \vec{v}_1 et \vec{u}_2 en \vec{v}_2 n'existe que si $\vec{u}_1 \cdot \vec{u}_2 = \vec{v}_1 \cdot \vec{v}_2$ (une rotation est une isométrie). On peut garantir cette égalité en remplaçant : \vec{v}_2 par \vec{v}_2' .

$$\vec{v}_2' = \left(\vec{u}_1 \cdot \vec{u}_2 - \vec{v}_1 \cdot \vec{v}_2 \sqrt{\frac{1 - (\vec{u}_1 \cdot \vec{u}_2)^2}{1 - (\vec{v}_1 \cdot \vec{v}_2)^2}} \right) \vec{v}_1 + \left(\sqrt{\frac{1 - (\vec{u}_1 \cdot \vec{u}_2)^2}{1 - (\vec{v}_1 \cdot \vec{v}_2)^2}} \right) \vec{v}_2$$

ce calcul doit être fait après la normalisation de la remarque précédente, et avant l'entrée dans l'algorithme.

L'algorithme ainsi sécurisé permet de calculer la rotation qui transforme \vec{u}_1 en \vec{v}_1 et \vec{v}_2 en un vecteur situé dans le plan (\vec{v}_1, \vec{v}_2) et orienté dans le même sens que \vec{v}_2 .

La résolution complète du quaternion se fait en suivant l'algorithme suivant :

$$\vec{u}_1 = \frac{\vec{u}_1}{\|\vec{u}_1\|}$$

$$\vec{u}_2 = \frac{\vec{u}_2}{\|\vec{u}_2\|}$$

$$\vec{v}_1 = \frac{\vec{v}_1}{\|\vec{v}_1\|}$$

$$\vec{v}_2 = \frac{\vec{v}_2}{\|\vec{v}_2\|}$$

$$coeff = \sqrt{\frac{1 - (\vec{u}_1 \cdot \vec{u}_2)^2}{1 - (\vec{v}_1 \cdot \vec{v}_2)^2}}$$

$$\vec{v}_2 = (\vec{u}_1 \cdot \vec{u}_2 - \vec{v}_1 \cdot \vec{v}_2 \times coeff) \vec{v}_1 + coeff \vec{v}_2$$

$$\vec{u}_{ref} = \vec{u}_1$$

$$\vec{v}_{ref} = \vec{v}_1$$

$$\vec{k} = (\vec{v}_1 - \vec{u}_1) \wedge (\vec{v}_2 - \vec{u}_2)$$

$$c = \vec{k} \cdot (\vec{u}_1 \wedge \vec{u}_2)$$

Si $(c < \varepsilon)$, alors

$$\vec{u}_3 = \frac{\vec{u}_1 \wedge \vec{u}_2}{1 - (\vec{u}_1 \cdot \vec{u}_2)^2}$$

$$\vec{v}_3 = \frac{\vec{v}_1 \wedge \vec{v}_2}{1 - (\vec{v}_1 \cdot \vec{v}_2)^2}$$

$$\vec{k} = (\vec{v}_1 - \vec{u}_1) \wedge (\vec{v}_3 - \vec{u}_3)$$

$$c = \vec{k} \cdot (\vec{u}_1 \wedge \vec{u}_3)$$

Si $(c < \varepsilon)$, alors

$$\vec{k} = (\vec{v}_2 - \vec{u}_2) \wedge (\vec{v}_3 - \vec{u}_3)$$

$$c = \vec{k} \cdot (\vec{u}_2 \wedge \vec{u}_3)$$

Si $(c < \varepsilon)$, alors

$$\vec{q} = \vec{0}$$

$$q_0 = 1$$

Retourner (q_0, \vec{q})

Fin si

$$\vec{u}_{ref} = \vec{u}_2$$

$$\vec{v}_{ref} = \vec{v}_2$$

Fin si

Fin si

$$\vec{q} = \frac{\vec{k}}{2\sqrt{c}}$$

$$q_0 = \frac{\vec{v}_{ref} \cdot (\vec{u}_{ref} \wedge \vec{q})}{2(\vec{u}_{ref} \wedge \vec{q})^2}$$

Retourner (q_0, \vec{q})

C Recherche de vecteurs orthogonaux

Connaissant un vecteur normé \vec{a} , comment trouver un vecteur \vec{b} normé orthogonal à \vec{a} ? Il existe une infinité de solutions possibles, ce qui est gênant du point de vue informatique où l'on veut une solution quelconque choisie dans cette infinité, même si du point de vue mathématique, le fait qu'il en existe au moins une soit suffisant. Les coordonnées de \vec{a} dans la base orthonormée $(\vec{i}, \vec{j}, \vec{k})$ sont :

$$\vec{a} \begin{cases} a_x \\ a_y \\ a_z \end{cases}$$

Comme \vec{a} est normé, on sait qu'au moins une des coordonnées est inférieure à $\frac{1}{\sqrt{3}}$ en valeur absolue. On choisit une coordonnée qui soit inférieure à 0,6 (supposons par exemple que ce soit a_z). On a alors $1 - a_z^2 \geq 0,64$ donc le vecteur $\vec{b} = \frac{\vec{a} \wedge \vec{k}}{\sqrt{1 - a_z^2}}$ est calculable. Ce vecteur est normé et orthogonal à \vec{a} , c'est notre solution. Si on choisit de travailler à partir de a_x , on utilise $\vec{b} = \frac{\vec{a} \wedge \vec{i}}{\sqrt{1 - a_x^2}}$, et si on choisit de travailler à partir de a_y , on utilise $\vec{b} = \frac{\vec{a} \wedge \vec{j}}{\sqrt{1 - a_y^2}}$.

D Rotation transformant un vecteur en un autre

Connaissant deux vecteurs \vec{a} et \vec{b} ($\|\vec{a}\| = \|\vec{b}\|$), comment trouver une rotation amenant \vec{a} sur \vec{b} ? Ce problème est similaire à celui des vecteurs orthogonaux (annexe C) car là encore, il faut faire un choix dans un ensemble infini de solutions. Supposons tout d'abord \vec{a} et \vec{b} non alignés. On définit le repère orthonormé $(\vec{i}, \vec{j}, \vec{k})$ par :

$$\begin{aligned}\vec{i} &= \frac{\vec{a}}{\|\vec{a}\|} \\ \vec{j} &= \frac{\|\vec{a}\|^2 \vec{b} - (\vec{a} \cdot \vec{b}) \vec{a}}{\|\vec{a}\|^3 \sqrt{1 - \frac{(\vec{a} \cdot \vec{b})^2}{\|\vec{a}\|^4}}} \\ \vec{k} &= \frac{\vec{a} \wedge \vec{b}}{\|\vec{a}\|^2 \sqrt{1 - \frac{(\vec{a} \cdot \vec{b})^2}{\|\vec{a}\|^4}}}\end{aligned}$$

Dans ce repère, les vecteurs \vec{a} et \vec{b} ont pour coordonnées :

$$\vec{a} \begin{cases} \|\vec{a}\| \\ 0 \\ 0 \end{cases} \quad \vec{b} \begin{cases} \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|} = \|\vec{b}\| \cos(\theta) \\ \|\vec{a}\| \sqrt{1 - \frac{(\vec{a} \cdot \vec{b})^2}{\|\vec{a}\|^4}} = \|\vec{b}\| \sin(\theta) \\ 0 \end{cases}$$

Une des rotations amenant \vec{a} sur \vec{b} est la rotation d'axe \vec{k} et d'angle θ .

La matrice de cette rotation en repère $(\vec{i}, \vec{j}, \vec{k})$ est :

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

On peut déduire de cette matrice une expression vectorielle traduisant la rotation d'un vecteur \vec{t} quelconque :

$$\begin{aligned}R_{\vec{k}, \theta}(\vec{t}) &= (\vec{t} \cdot \vec{k}) \vec{k} + \cos(\theta) [\vec{t} - (\vec{t} \cdot \vec{k}) \vec{k}] + \sin(\theta) [\vec{k} \wedge (\vec{t} - (\vec{t} \cdot \vec{k}) \vec{k})] \\ &= \cos(\theta) \vec{t} + (1 - \cos(\theta)) (\vec{t} \cdot \vec{k}) \vec{k} + \sin(\theta) \vec{k} \wedge \vec{t}\end{aligned}$$

d'où en exprimant $\cos(\theta)$, $\sin(\theta)$, et \vec{k} :

$$\begin{aligned}R_{\vec{a}, \vec{b}}(\vec{t}) &= \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|^2} \vec{t} \\ &+ \frac{\|\vec{a}\|^2 - \vec{a} \cdot \vec{b}}{\|\vec{a}\|^2} \frac{\vec{t} \cdot (\vec{a} \wedge \vec{b})}{\|\vec{a}\|^2 \sqrt{1 - \frac{(\vec{a} \cdot \vec{b})^2}{\|\vec{a}\|^4}}} \frac{\vec{a} \wedge \vec{b}}{\|\vec{a}\|^2 \sqrt{1 - \frac{(\vec{a} \cdot \vec{b})^2}{\|\vec{a}\|^4}}} \\ &+ \sqrt{1 - \frac{(\vec{a} \cdot \vec{b})^2}{\|\vec{a}\|^4}} \frac{\vec{a} \wedge \vec{b}}{\|\vec{a}\|^2 \sqrt{1 - \frac{(\vec{a} \cdot \vec{b})^2}{\|\vec{a}\|^4}}} \wedge \vec{t} \\ &= \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|^2} \vec{t} + \frac{\vec{t} \cdot (\vec{a} \wedge \vec{b})}{\|\vec{a}\|^2 (\|\vec{a}\|^2 + \vec{a} \cdot \vec{b})} (\vec{a} \wedge \vec{b}) + \frac{\vec{a} \wedge \vec{b}}{\|\vec{a}\|^2} \wedge \vec{t}\end{aligned}$$

On peut alors retraduire cette expression vectorielle sous forme matricielle, mais cette fois dans la base dans laquelle on connaît \vec{a} et \vec{b} :

$$[R(t)] = M[\vec{t}]$$

$$M = \frac{1}{\|\vec{a}\|^2} \begin{bmatrix} \vec{a} \cdot \vec{b} + \frac{c_x^2}{\|\vec{a}\|^2 + \vec{a} \cdot \vec{b}} & \frac{c_x c_y}{\|\vec{a}\|^2 + \vec{a} \cdot \vec{b}} - c_z & \frac{c_x c_z}{\|\vec{a}\|^2 + \vec{a} \cdot \vec{b}} + c_y \\ \frac{c_x c_y}{\|\vec{a}\|^2 + \vec{a} \cdot \vec{b}} + c_z & \vec{a} \cdot \vec{b} + \frac{c_y^2}{\|\vec{a}\|^2 + \vec{a} \cdot \vec{b}} & \frac{c_y c_z}{\|\vec{a}\|^2 + \vec{a} \cdot \vec{b}} - c_x \\ \frac{c_x c_z}{\|\vec{a}\|^2 + \vec{a} \cdot \vec{b}} - c_y & \frac{c_y c_z}{\|\vec{a}\|^2 + \vec{a} \cdot \vec{b}} + c_x & \vec{a} \cdot \vec{b} + \frac{c_z^2}{\|\vec{a}\|^2 + \vec{a} \cdot \vec{b}} \end{bmatrix}$$

où c_x , c_y et c_z sont les coordonnées de $\vec{a} \wedge \vec{b}$.

Lors du calcul de la forme vectorielle de $R_{\vec{a}, \vec{b}}(\vec{t})$, nous avons effectué une simplification par $1 - \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|^2}$. Grâce à cette simplification, la matrice M ci-dessus est également valide quand $\vec{a} = \vec{b}$, et devient la matrice identité. Le seul cas particulier à traiter est donc $\vec{a} = -\vec{b}$.

On cherche pour cela un vecteur \vec{c} orthogonal à \vec{a} (voir annexe C) et on calcule la matrice de rotation d'angle π autour de \vec{c} (voir annexe E).

Remarque : On peut indifféremment considérer dans l'expression de la matrice précédente que c'est le vecteur \vec{a} qui tourne pour venir en \vec{b} , ou que le vecteur est fixe et que le repère tourne de sorte que \vec{a} représente les coordonnées avant rotation et que \vec{b} représente les coordonnées après rotation. On a donc simplement $[\vec{b}] = M[\vec{a}]$, et c'est l'utilisateur qui donne la signification physique qu'il veut à cette expression.

Si on veut exprimer la rotation à l'aide du formalisme des quaternions, on obtient :

$$Q \begin{cases} q_0 = \cos\left(-\frac{\theta}{2}\right) \\ q_1 = \sin\left(-\frac{\theta}{2}\right)k_x \\ q_2 = \sin\left(-\frac{\theta}{2}\right)k_y \\ q_3 = \sin\left(-\frac{\theta}{2}\right)k_z \end{cases}$$

D'où, en exprimant $\cos\left(-\frac{\theta}{2}\right)$, $\sin\left(-\frac{\theta}{2}\right)$ et \vec{k} :

$$\cos\left(-\frac{\theta}{2}\right) = \sqrt{\frac{1 + \cos\theta}{2}} = \sqrt{\frac{\|\vec{a}\|^2 + \vec{a} \cdot \vec{b}}{2\|\vec{a}\|^2}}$$

$$\sin\left(-\frac{\theta}{2}\right) = \sqrt{\frac{1 - \cos\theta}{2}} = -\sqrt{\frac{\|\vec{a}\|^2 - \vec{a} \cdot \vec{b}}{2\|\vec{a}\|^2}}$$

$$\vec{k} = \frac{\vec{a} \wedge \vec{b}}{\|\vec{a}\|^2 \sqrt{1 - \frac{(\vec{a} \cdot \vec{b})^2}{\|\vec{a}\|^4}}}$$

On obtient :

$$q_0 = \sqrt{\frac{1}{2} \left(1 + \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|^2} \right)}$$

$$\begin{aligned} \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} &= \frac{-(\vec{a} \wedge \vec{b})}{\|\vec{a}\|^2 \sqrt{2 \left(1 + \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|^2}\right)}} \\ &= \frac{-(\vec{a} \wedge \vec{b})}{2q_0 \|\vec{a}\|^2} \end{aligned}$$

Cette expression est singulière si $\vec{b} = -\vec{a}$ ($q_0 = 0$). Dans ce cas, on calcule \vec{c} orthogonal à \vec{a} (voir annexe C) et on calcule le quaternion :

$$\begin{cases} q_0 &= 0 \\ \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} &= (\vec{c}) \end{cases}$$

Une application de ces formules est le calcul du quaternion qui transforme un repère initial dans lequel les coordonnées d'un vecteur sont $(v_x, v_y, v_z)^\top$ en un repère canonique où les coordonnées de ce même vecteur sont $(1, 0, 0)^\top$. On a alors :

Si $v_x = -1$

$$q_0 = 0$$

$$q_1 = 0$$

$$q_2 = 1$$

$$q_3 = 0$$

Sinon

$$q_0 = \sqrt{\frac{1+v_x}{2}}$$

$$q_1 = 0$$

$$q_2 = \frac{-v_z}{\sqrt{2(1+v_x)}}$$

$$q_3 = \frac{v_y}{\sqrt{2(1+v_x)}}$$

Fin si

E Rotation d'axe et d'angle donnés

Soit \vec{k} ($\|\vec{k}\| = 1$), l'axe de la rotation, et soit θ son angle. On considère que les vecteurs sont fixes et que c'est le repère qui tourne. La relation vectorielle suivante a été démontrée dans l'annexe D :

$$R_{\vec{k},\theta}(\vec{t}) = \cos(\theta)\vec{t} + (1 - \cos(\theta))(\vec{t} \cdot \vec{k})\vec{k} + \sin(-\theta)\vec{k} \wedge \vec{t}$$

Le signe « $-$ » provient du fait qu'ici on considère que c'est le repère qui tourne.

Cette expression se traduit sous forme matricielle : $R_{\vec{k},\theta}(\vec{t}) = M[\vec{t}]$ avec :

$$M = \begin{bmatrix} \cos \theta + (1 - \cos \theta)k_x^2 & (1 - \cos \theta)k_x k_y + \sin \theta k_z & (1 - \cos \theta)k_x k_z - \sin \theta k_y \\ (1 - \cos \theta)k_x k_y - \sin \theta k_z & \cos \theta + (1 - \cos \theta)k_y^2 & (1 - \cos \theta)k_y k_z + \sin \theta k_x \\ (1 - \cos \theta)k_x k_z + \sin \theta k_y & (1 - \cos \theta)k_y k_z - \sin \theta k_x & \cos \theta + (1 - \cos \theta)k_z^2 \end{bmatrix}$$

Si on veut exprimer la rotation à l'aide du formalisme des quaternions, on obtient :

$$\begin{aligned} q_0 &= \cos \frac{\theta}{2} \\ q_1 &= \sin \frac{\theta}{2} k_x \\ q_2 &= \sin \frac{\theta}{2} k_y \\ q_3 &= \sin \frac{\theta}{2} k_z \end{aligned}$$

Remarque : cette expression diffère de l'expression donnée à l'annexe A car cette fois θ représente l'angle dont le repère tourne (et non pas l'angle dont les vecteurs semblent tourner).

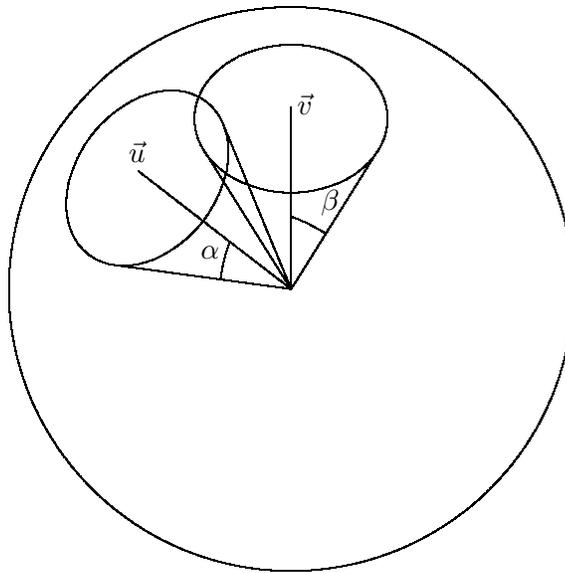
F Intersection de cônes sur la sphère

Les champs de vue des capteurs géométriques sont définis comme des combinaisons plus ou moins complexes de cônes (réunions et intersections imbriquées). De plus, dans le cas des capteurs de limbe, le bord de l'astre observé est également vu par le capteur comme un cône. Enfin, les consignes géométriques sont modélisées par des cônes.

On est donc souvent amené à rencontrer des intersections de cônes sur la sphère unité centrée sur le satellite.

Soient deux cônes (\vec{u}, α) et (\vec{v}, β) non coaxiaux (voir figure 14). On va chercher à exprimer les coordonnées

FIG. 14 – intersection de cônes



des intersections dans le repère non orthonormé $\vec{u}, \vec{v}, \vec{u} \wedge \vec{v}$ ($\vec{u} \wedge \vec{v}$ existe dès lors que les cônes ne sont pas coaxiaux).

$$\vec{p} = a\vec{u} + b\vec{v} + c\vec{u} \wedge \vec{v}$$

Les conditions

$$\vec{p} \cdot \vec{u} = \cos \alpha$$

$$\vec{p} \cdot \vec{v} = \cos \beta$$

$$\vec{p} \cdot \vec{p} = 1$$

permettent d'exprimer les coordonnées :

$$a = \frac{\cos \alpha - \vec{u} \cdot \vec{v} \cos \beta}{1 - (\vec{u} \cdot \vec{v})^2}$$

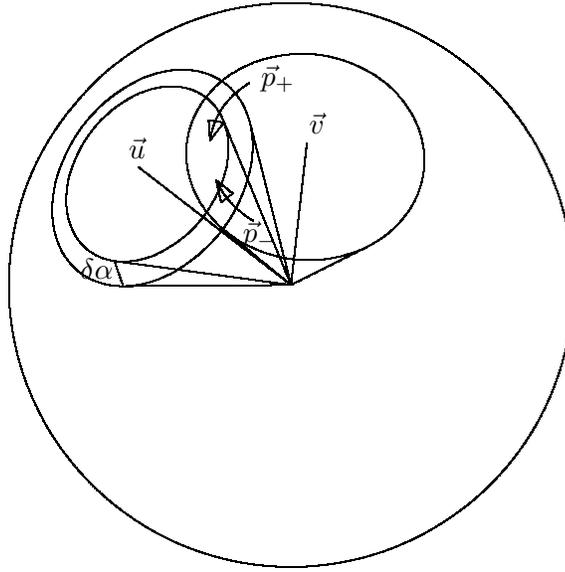
$$b = \frac{\cos \beta - \vec{u} \cdot \vec{v} \cos \alpha}{1 - (\vec{u} \cdot \vec{v})^2}$$

$$c = \pm \sqrt{\frac{1 - (a\vec{u} + b\vec{v})^2}{1 - (\vec{u} \cdot \vec{v})^2}}$$

La condition d'existence de c ($(a\vec{u} + b\vec{v}) < 1$) est la condition d'existence des intersections.

Les deux choix possibles de c donnent les deux solutions quand elles existent, symétriques par rapport au plan (\vec{u}, \vec{v}) . On notera les solutions \vec{p}_+ et \vec{p}_- selon le signe de c dans la suite de cette annexe.

FIG. 15 – variation des intersections



On cherche maintenant à savoir quelle est la partie du cône (\vec{v}, β) qui est incluse dans le cône (\vec{u}, α) et quelle est la partie qui est hors du cône. On fait donc varier l'angle d'ouverture α et on observe le sens de variation de $\vec{p}_+(\alpha)$ et $\vec{p}_-(\alpha)$ (voir figure 15).

Les dérivées des coordonnées sont :

$$\begin{aligned} \frac{\partial a}{\partial \alpha} &= \frac{-\sin \alpha}{1 - (\vec{u} \cdot \vec{v})^2} \\ \frac{\partial b}{\partial \alpha} &= \frac{(\vec{u} \cdot \vec{v}) \sin \alpha}{1 - (\vec{u} \cdot \vec{v})^2} \\ \frac{\partial [||a\vec{u} + b\vec{v}||^2]}{\partial \alpha} &= \frac{2 \sin \alpha}{1 - (\vec{u} \cdot \vec{v})^2} [a\vec{u} + b\vec{v}] \cdot [-\vec{u} + (\vec{u} \cdot \vec{v})\vec{v}] = -2a \sin \alpha \\ \frac{\partial c_+}{\partial \alpha} &= \frac{a \sin \alpha}{c_+ (1 - (\vec{u} \cdot \vec{v})^2)} \\ \frac{\partial c_-}{\partial \alpha} &= \frac{a \sin \alpha}{c_- (1 - (\vec{u} \cdot \vec{v})^2)} \end{aligned}$$

On en déduit :

$$\begin{aligned} \frac{\partial \vec{p}_+}{\partial \alpha} &= \frac{\sin \alpha}{1 - (\vec{u} \cdot \vec{v})^2} \left[-\vec{u} + (\vec{u} \cdot \vec{v})\vec{v} + \frac{a}{c_+} \vec{u} \wedge \vec{v} \right] \\ \frac{\partial \vec{p}_-}{\partial \alpha} &= \frac{\sin \alpha}{1 - (\vec{u} \cdot \vec{v})^2} \left[-\vec{u} + (\vec{u} \cdot \vec{v})\vec{v} + \frac{a}{c_-} \vec{u} \wedge \vec{v} \right] \end{aligned}$$

À ce stade, on peut remarquer que :

$$\begin{aligned}\vec{v} \wedge \vec{p}_+ &= a\vec{v} \wedge \vec{u} + b\vec{v} \wedge \vec{v} + c_+\vec{v} \wedge (\vec{u} \wedge \vec{v}) \\ &= c_+ [\vec{u} - (\vec{u} \cdot \vec{v})\vec{v}] - a\vec{u} \wedge \vec{v}\end{aligned}$$

(par application du double produit vectoriel : $\vec{a} \wedge (\vec{b} \wedge \vec{c}) = (\vec{a} \cdot \vec{c})\vec{b} - (\vec{a} \cdot \vec{b})\vec{c}$)

De même, on démontre :

$$\vec{v} \wedge \vec{p}_- = c_- [\vec{u} - (\vec{u} \cdot \vec{v})\vec{v}] - a\vec{u} \wedge \vec{v}$$

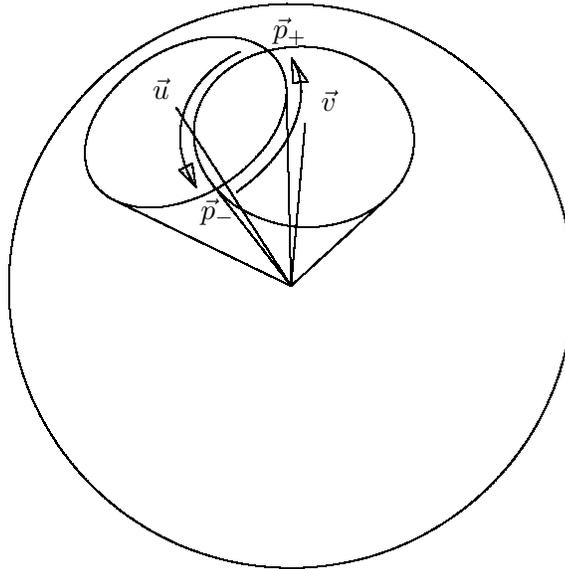
Donc :

$$\begin{aligned}\frac{\partial \vec{p}_+}{\partial \alpha} &= \frac{-\sin \alpha}{c_+} \frac{\vec{v} \wedge \vec{p}_+}{1 - (\vec{u} \cdot \vec{v})^2} \\ \frac{\partial \vec{p}_-}{\partial \alpha} &= \frac{-\sin \alpha}{c_-} \frac{\vec{v} \wedge \vec{p}_-}{1 - (\vec{u} \cdot \vec{v})^2}\end{aligned}$$

Comme $\sin(\alpha) > 0$, $c_+ > 0$, $c_- < 0$, $1 - (\vec{u} \cdot \vec{v})^2 > 0$, une diminution de l'angle d'ouverture α engendre une rotation négative de \vec{p}_- autour de \vec{v} , et une rotation positive de \vec{p}_+ autour de \vec{v} .

La partie du cône (\vec{v}, β) incluse dans le cône (\vec{u}, α) est donc la partie qui va de \vec{p}_+ vers \vec{p}_- en tournant positivement autour de \vec{v} (voir figure 16). Un raisonnement symétrique montre que la partie du cône (\vec{u}, α) incluse dans le cône (\vec{v}, β) est la partie qui va de \vec{p}_- vers \vec{p}_+ en tournant positivement autour de \vec{u} . Ce résultat

FIG. 16 – parties incluses



est vrai dans toutes les configurations de \vec{u} , \vec{v} , α , β .

G Dérivation automatique

G.1 besoins

La bibliothèque MARMOTTES est destinée à être appelée au cœur des boucles de calcul des logiciels de mécanique spatiale, qui peuvent donc demander un grand nombre de déterminations d'orientation (parfois quelques centaines de milliers d'appels). De plus, la possibilité de passer en cours de calcul, d'un type de capteur à un autre, peut faire évoluer considérablement les ordres de grandeurs manipulés (voire les unités, quand on passe d'un capteur qui mesure des angles à un capteur qui mesure des vitesses angulaires).

Ces deux points ont conduit à rejeter les méthodes de différences finies pour calculer les dérivées des fonctions dans les algorithmes de recherche de zéros.

La complexité des équations a conduit, quant à elle, à rejeter les méthodes où la dérivée symbolique était codée dans le logiciel.

MARMOTTES utilise donc un système de dérivation automatique.

G.2 principe

La dérivation automatique consiste à réaliser le calcul de la dérivée de proche en proche, à chaque opération élémentaire, en parallèle du calcul de la valeur. Ce sont donc les règles élémentaires de dérivation qui sont codées (la dérivée de la somme est la somme des dérivées,...). Ce procédé permet d'obtenir un calcul exact (à la précision de la machine près) sans avoir besoin de coder effectivement la formule symbolique, puisque l'ensemble des calculs est réalisé à chaque opération, en utilisant les valeurs et dérivées courantes.

Cette méthode est applicable quel que soit l'ordre de dérivation nécessaire, la seule contrainte étant que si l'on veut avoir l'ordre n , il faut calculer également tous les ordres de 0 à $n - 1$.

L'un des inconvénients de cette méthode est que la moindre addition devient un appel de fonction. Ceci peut être très largement compensé par l'utilisation des fonctions `inline` du langage C++ qui évitent le surcoût en temps des appels, et par la surcharge d'opérateurs, qui permet de continuer à écrire : `y = sin(x) + 3 * x`. C'est le compilateur qui se charge d'interpréter l'expression et de mettre les appels (ou le code dans le cas des fonctions `inline`) à la place des opérateurs `+` et `*` et de la fonction `sin`.

G.3 formulation

L'ensemble des calculs de fonctions explicites peut être réalisé à l'aide des formules suivantes, dans lesquelles g et h sont les données d'entrée et où f est la fonction calculée.

G.3.1 initialisation

Si la variable selon laquelle on dérive est x , on initialise le calcul en déclarant x comme fonction identité :

$$f'' = 0$$

$$\begin{aligned}f' &= 1 \\f &= x\end{aligned}$$

La constante c est initialisée comme fonction constante :

$$\begin{aligned}f'' &= 0 \\f' &= 0 \\f &= c\end{aligned}$$

G.3.2 opérations arithmétiques

L'addition de deux fonctions se fait par :

$$\begin{aligned}f'' &= g'' + h'' \\f' &= g' + h' \\f &= g + h\end{aligned}$$

La soustraction de deux fonctions se fait par :

$$\begin{aligned}f'' &= g'' - h'' \\f' &= g' - h' \\f &= g - h\end{aligned}$$

La multiplication de deux fonctions se fait par :

$$\begin{aligned}f'' &= g'' \times h' + 2g' \times h'' + g \times h'' \\f' &= g' \times h + g \times h' \\f &= g \times h\end{aligned}$$

La division de deux fonctions se fait par :

$$\begin{aligned}f'' &= \frac{[h(g'' \times h - g \times h'') - 2h''(g' \times h - g \times h')]}{h^3} \\f' &= \frac{g' \times h - g \times h'}{h^2} \\f &= \frac{g}{h}\end{aligned}$$

G.3.3 composition par une fonction scalaire

La composition par une fonction λ se fait par :

$$\begin{aligned}f'' &= \lambda'(g) \times g'' + \lambda''(g) \times g'^2 \\f' &= \lambda'(g) \times g' \\f &= \lambda(g)\end{aligned}$$

Les dérivées des fonctions usuelles sont données par le tableau :

$\lambda(x)$	$\lambda'(x)$	$\lambda''(x)$
\sqrt{x}	$\frac{1}{2\sqrt{x}}$	$\frac{-1}{4x^{3/2}}$
$\sin(x)$	$\cos(x)$	$-\sin(x)$
$\cos(x)$	$-\sin(x)$	$-\cos(x)$
$\tan(x)$	$1 + \tan^2(x)$	$2 \tan(x)[1 + \tan^2(x)]$
$\arcsin(x)$	$\frac{1}{\sqrt{1-x^2}}$	$\frac{x}{(1-x^2)^{3/2}}$
$\arccos(x)$	$\frac{-1}{\sqrt{1-x^2}}$	$\frac{-x}{(1-x^2)^{3/2}}$
$\arctan(x)$	$\frac{1}{1+x^2}$	$\frac{-2x}{(1+x^2)^2}$
e^x	e^x	e^x
$\ln(x)$	$\frac{1}{x}$	$\frac{-1}{x^2}$

G.3.4 règles particulières

Deux fonctions à deux variables sont parfois utilisées dans les calculs et nécessitent des règles particulières, qui sont implantées sous forme d'algorithmes faisant intervenir les règles précédentes. x^y est calculé par : $x^y = e^{y \ln(x)}$, $\alpha = \text{atan2}(x, y)$ est calculé par :

$$r = \sqrt{x^2 + y^2}$$

Si ($x \geq 0$)

$$\alpha = 2 \arctan\left(\frac{y}{r+x}\right)$$

Sinon

$$\beta = 2 \arctan\left(\frac{y}{r-x}\right)$$

Si ($\beta \leq 0$)

$$\alpha = -\pi - \beta$$

Sinon

$$\alpha = \pi - \beta$$

Fin si

Fin si

Il faut remarquer que le branchement conditionnel sur x ne génère aucun problème pour la dérivation. En effet, les deux formulations sont toutes deux applicables au voisinage du changement d'alternative et donnent le même résultat. Le test n'est présent que pour éviter les problèmes numériques qui se posent loin de la zone de changement. Cet algorithme renvoie un angle compris entre $-\pi$ et π , comme la fonction `atan2` des langages de programmation.

G.3.5 implantation informatique

On peut remarquer que dans les formules précédentes, le calcul des dérivées secondes était toujours présenté avant le calcul des dérivées premières, lui-même toujours exposé avant le calcul des fonctions. Ce choix a été réalisé pour le codage, car il permet en effet de rendre les algorithmes valides, même si on stocke f dans une des variables d'appel g ou h ; en effet, on écrase f'' une fois qu'on n'en a plus besoin. (g'' et h'' n'interviennent jamais dans le calcul de f' ou f), il en est de même pour f' .

Dans la pratique, MARMOTTES n'utilise que des calculs limités aux dérivées d'ordre 1. En effet, si l'utilisation de dérivation automatique d'ordre 1 permet de mettre en place des algorithmes plus rapides de recherche de zéros, la dérivation jusqu'à l'ordre 2 est considérablement plus coûteuse en temps de calcul, et le gain en nombre d'itérations n'est pas énorme, globalement l'expérience montre que l'on perd du temps. L'une des opérations les plus critiques est la multiplication, qui demande 10 opérations si l'on va jusqu'à l'ordre 2, et qui n'en demande que 4 si l'on s'arrête à l'ordre 1.

H Recherche des zéros d'une fonction

H.1 besoins

Le principe de fonctionnement de MARMOTTES consiste à fabriquer puis annuler une fonction scalaire à une variable respectant toujours deux consignes sur les trois, et s'annulant lorsque la troisième consigne est respectée.

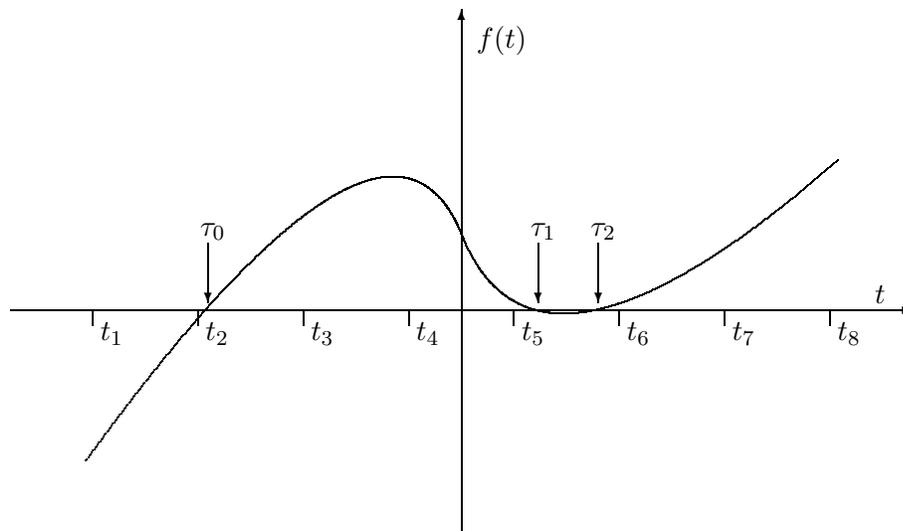
La fonction fabriquée est définie entre 0 et +1 (au prix d'un changement de variable), elle est dérivable partout sauf éventuellement aux bornes. Les implantations classiques de capteurs conduisent de plus à des fonctions relativement régulières ayant des extremums locaux bien séparés et des zéros bien séparés. La méthode présentée est en fait capable de détecter également des zéros proches les uns des autres.

H.2 principes

H.2.1 encadrement des racines

Le principe de résolution adopté consiste à identifier des intervalles monotones entourant chaque zéro, puis à rechercher le zéro correspondant.

FIG. 17 – recherche des intervalles monotones

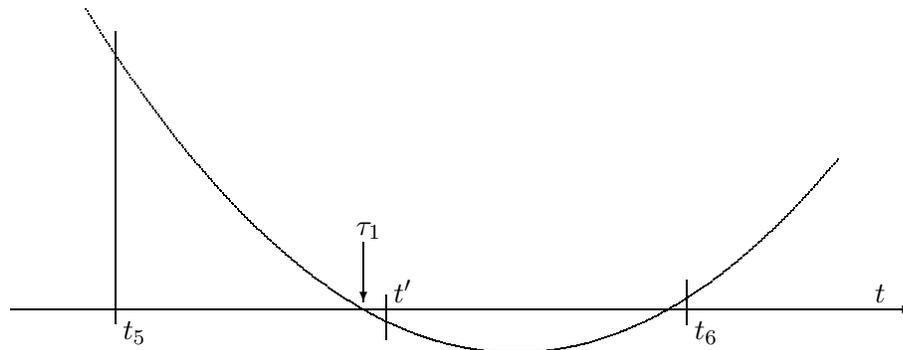


Les intervalles monotones sont trouvés en explorant l'intervalle avec un pas fixe et en calculant en chaque point la dérivée première de la fonction de façon exacte (par dérivation automatique, conformément à la méthode exposée dans l'annexe G).

Dans l'exemple de la figure 17, on voit que l'intervalle $[t_1; t_2]$ est monotone croissant mais n'encadre aucun zéro. L'intervalle $[t_2; t_3]$ est monotone croissant et encadre un zéro qui devra être recherché par la méthode décrite plus loin.

Après calcul de ce zéro, l'exploration se poursuit avec l'intervalle $[t_3; t_4]$, qui n'est pas monotone (on le détecte par le signe des dérivées), cependant la fonction croît à partir d'une valeur positive puis décroît jusqu'à une autre valeur également positive, elle ne peut donc pas s'annuler dans l'intervalle dès lors que le pas d'exploration est plus petit que l'écart entre deux extremums locaux²⁶. Un simple test sur les dérivées montre que l'intervalle $[t_4; t_5]$ est monotone décroissant, un autre test sur $f(t_5)$ montre qu'il est toujours positif ; il ne peut donc contenir aucun zéro. Quand on arrive à l'intervalle $[t_6; t_7]$, on ne peut pas savoir d'avance s'il y aura des zéros dans l'intervalle ou pas, on recherche alors l'extremum local (c'est-à-dire le zéro de la dérivée), en s'arrêtant soit lorsque cet extremum est trouvé soit quand on a un point au-dessous de l'axe, suffisant pour encadrer un zéro dans un intervalle monotone (voir la figure 18, où l'intervalle $[t_5; t']$ encadre τ_1).

FIG. 18 – arrêt de la recherche lorsqu'un zéro est encadré



Le point obtenu après convergence de l'algorithme n'a généralement pas une dérivée rigoureusement nulle. Par conséquent, pour que l'analyse du tronçon suivant fonctionne, on force le fait qu'après un maximum la fonction décroît, et qu'elle croît après un minimum, même si l'approximation de l'extremum obtenue a une dérivée du mauvais signe.

Afin d'éviter les recherches d'extremums inutiles, on fait la remarque suivante : si la fonction f est assez régulière pour ne pas avoir de grandes variations de la dérivée sur un intervalle, le cas schématisé par la figure 19 ne peut pas se produire. On ne fait donc pas de recherche d'extremum si une approximation \mathcal{P} de l'extremum paraît vraiment très loin de l'axe par rapport à la valeur observée aux bornes (voir figure 20).

H.2.2 raffinage

Une fois détecté un intervalle monotone encadrant la racine, on recherche cette racine par une méthode de Newton modifiée pour utiliser les deux bornes de l'intervalle dans une approximation cubique inverse. Cette approximation n'est pas calculable si l'une des dérivées est trop petite, on utilise alors une méthode de sécante en remplacement de la méthode de Newton modifiée pour assurer que dans tous les cas le nouveau point sera entre les anciens (voir la figure 21). Une fois la fonction évaluée sur le nouveau point, on remplace une des bornes par ce point de façon à réduire l'intervalle de recherche.

La convergence de l'algorithme est testée par une fonction définie par l'utilisateur et à qui seront fournies les bornes et les valeurs de la fonction aux bornes. Cette fonction indiquera en retour si l'une des bornes peut

²⁶il ne peut y avoir qu'un extremum par intervalle, et ici on sait qu'il s'agit d'un maximum

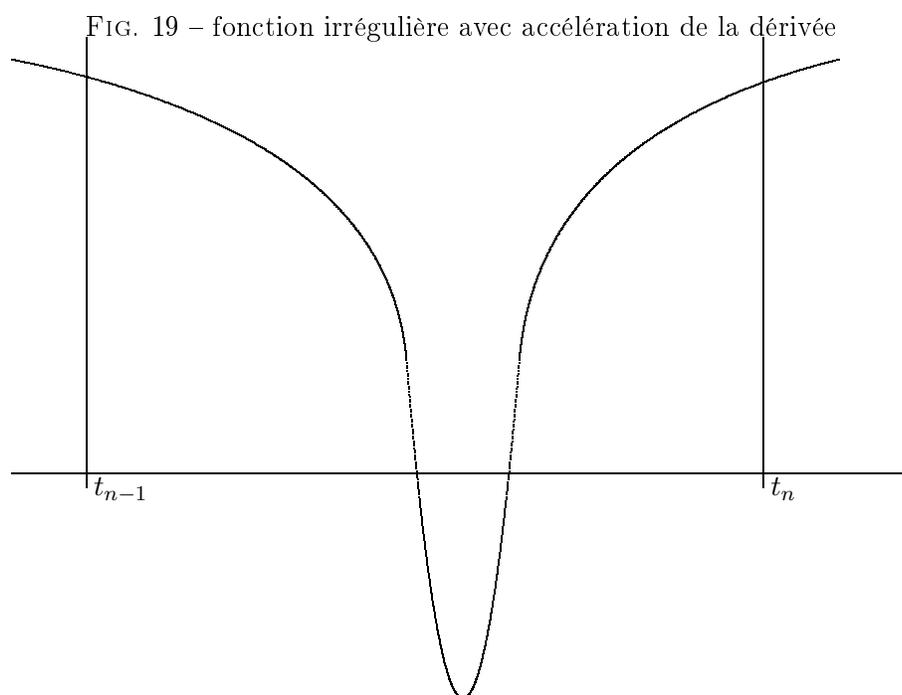


FIG. 20 – approximation d'un extremum par les bornes

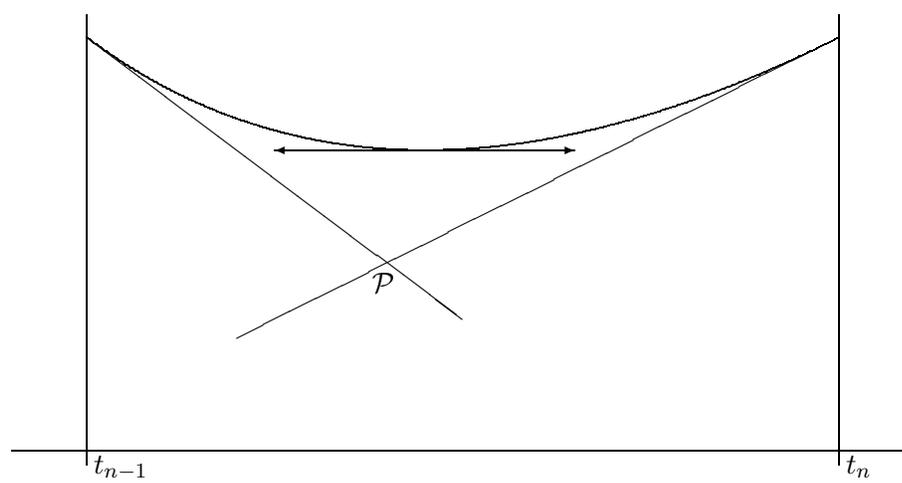
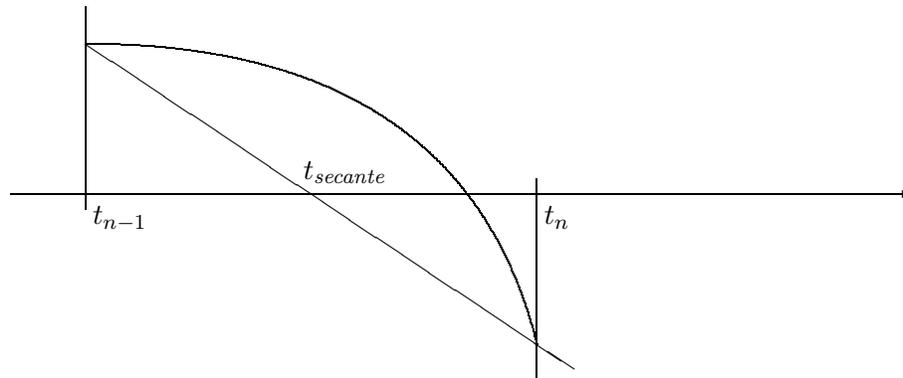


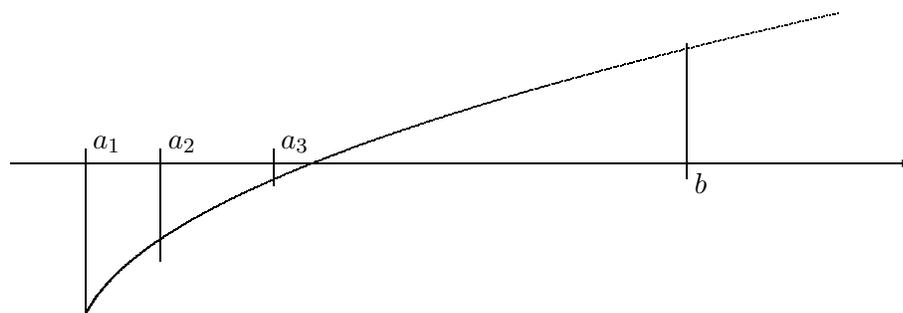
FIG. 21 – utilisation d’une sécante si la méthode de Newton modifiée n’est pas applicable



être considérée comme ayant convergé vers la solution ou si aucune n’est assez proche. L’utilisateur peut fonder sa décision sur la taille de l’intervalle (convergence en abscisse) ou sur les valeurs de la fonction (convergence en ordonnée). MARMOTTES utilise la convergence en ordonnée, le seuil de convergence étant par défaut le dixième de la précision du capteur concerné.

Si l’utilisateur utilise la convergence en abscisse, la méthode précédente peut échouer dans le cas schématisé par la figure 22, où la borne inférieure de l’intervalle converge vers la racine alors que la borne supérieure en reste toujours éloignée. Il faut donc sécuriser l’algorithme en rééquilibrant l’intervalle quand une borne est beaucoup plus proche de la racine que l’autre en ordonnée. Le rééquilibrage est réalisé en décalant toute la courbe de

FIG. 22 – convergence d’une seule borne de l’intervalle



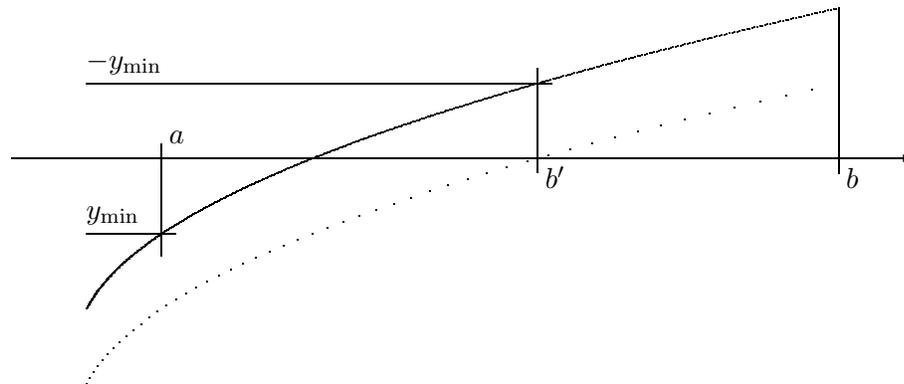
y_{\min} pour une itération, ce qui revient à chercher le point d’ordonnée $-y_{\min}$ sur la courbe originale au lieu de rechercher le point d’ordonnée 0 (voir la figure 23).

H.3 algorithme

H.3.1 séparation des zéros

L’algorithme suivant sépare les zéros par découpage de l’intervalle général, avec si nécessaire calcul des extremums locaux. Le même algorithme est utilisé que l’on dispose de la dérivée seconde ou que l’on ne dispose que de la dérivée première (la différence n’intervient que dans les recherches fines sous-jacentes décrites dans les

FIG. 23 – rééquilibrage de l'intervalle



sections H.3.2 et H.3.3). Les entrées de l'algorithme sont les bornes a et b de l'intervalle et le **pas** de découpage. Les variables **min** et **max** sont destinées à encadrer les zéros ($f(\mathbf{min}) < 0$, $f(\mathbf{max}) > 0$), elles ne correspondent donc pas à des extremums locaux mais aux extremums sur l'intervalle d'encadrement qu'elles définissent. Si les valeurs a , b , **pas**, **min** et **max** sont conservées d'un appel à l'autre, l'algorithme trouve tous les zéros de l'intervalle dans l'ordre croissant, en déplaçant la borne a .

Boucler indéfiniment

ajustement du pas en fin de parcours

Si ($a + \mathbf{pas} > b$)

$\mathbf{pas} \leftarrow \max(\varepsilon, b - a)$

Fin si

$t \leftarrow t + \mathbf{pas}$

Si ($t > b$)

échec de la recherche

Fin si

Si $f'(a)$ et $f'(t)$ sont de même signe

Si fonction croissante

Si ($f(t) \leq 0$)

$\mathbf{min} \leftarrow t$

Sinon

Si ($f(t) < 0$)

$t_0 \leftarrow$ zéro situé entre \mathbf{min} et t , *critère de convergence : fonction utilisateur*

$a \leftarrow t$

$\min \leftarrow t$

retourner t_0

Fin si

Fin si

Sinon

Si ($f(t) \geq 0$)

$\max \leftarrow t$

Sinon

Si ($f(\max) > 0$)

$t_0 \leftarrow$ zéro situé entre t et \max , *critère de convergence : fonction utilisateur*

$a \leftarrow t$

$\max \leftarrow t$

retourner t_0

Fin si

Fin si

Fin si

Sinon

il y a un extremum entre a et t

Si ($f(a) > 0$)

Si ($f(a) > 0$) et ($f(t) > 0$)

on sait d'emblée que tout l'intervalle est positif

$\max \leftarrow a$

mémoriser le fait qu'il faudra réinitialiser \max

Sinon

$\max \leftarrow$ extremum entre a et t , *critère de convergence : intervalle entièrement positif*

$t \leftarrow \max$

Fin si

Sinon

Si $(f(a) < 0)$ et $(f(t) < 0)$

on sait d'emblée que tout l'intervalle est négatif

$\text{min} \leftarrow a$

mémoriser le fait qu'il faudra réinitialiser `min`

Sinon

$\text{min} \leftarrow$ extremum entre `a` et `t`, *critère de convergence : intervalle entièrement négatif*

$t \leftarrow \text{min}$

Fin si

Fin si

à partir de là, on sait qu'il y a au plus un zéro entre `min` et `max`

Si $(f(\text{min}) \leq 0)$ et $(f(\text{max}) \geq 0)$

$t_0 \leftarrow$ zéro situé entre `min` et `max`, *critère de convergence : fonction utilisateur*

$a \leftarrow t$

Si `min` doit être réinitialisé

$\text{min} \leftarrow t$

Fin si

Si `max` doit être réinitialisé

$\text{max} \leftarrow t$

Fin si

Retourner t_0

Fin si

Si `min` doit être réinitialisé

$\text{min} \leftarrow t$

Fin si

Si `max` doit être réinitialisé

$\text{max} \leftarrow t$

Fin si

préparation de l'analyse du tronçon suivant

$a \leftarrow t$

Fin si

Fin boucle

H.3.2 recherche d'extremum

Utilisation des seules dérivées premières. Trouver l'extremum de la fonction f entre a et b lorsque l'on dispose de sa dérivée revient à calculer le zéro de la fonction f' sans connaître sa dérivée. On utilise pour cela la méthode de BRENT. Il s'agit d'une méthode hybride conjugant la bisection et la sécante, avec en amélioration supplémentaire l'utilisation d'une approximation quadratique inverse lorsque l'on a évalué assez de points distincts au cours des itérations.

Utilisation des dérivées premières et secondes. Lorsque l'on dispose des dérivées secondes, on peut utiliser une méthode de Newton modifiée pour utiliser les dérivées aux deux bornes de l'intervalle courant pour trouver l'extremum, en conservant la sécante pour sécuriser l'algorithme en cas de dérivée seconde trop faible sur une des bornes (ce qui empêche l'utilisation du Newton modifié). L'algorithme suivant trouve par cette conjugaison de méthodes l'extremum de la fonction f entre a et b lorsque l'on sait qu'il y en a un.

Boucler

$y_a \leftarrow f'(a)$

$y'_a \leftarrow f''(a)$

$y_b \leftarrow f'(b)$

$y'_b \leftarrow f''(b)$

Si ($|y_a| > 10|y_b|$)

intervalle disproportionné, on décale les ordonnées pour chercher $-y_b$ au lieu de 0

$y_a \leftarrow y_a + y_b$

$y_b \leftarrow y_b + y_b$

Sinon

Si ($|y_b| > 10|y_a|$)

intervalle disproportionné, on décale les ordonnées pour chercher $-y_a$ au lieu de 0

$y_b \leftarrow y_b + y_a$

$y_a \leftarrow y_a + y_a$

Fin si

Fin si

$$t \leftarrow a - 1$$

Si ($|y'_a| > \varepsilon$ et $|y'_b| > \varepsilon$)

les dérivées sont suffisantes pour appliquer Newton en deux points

$$r_1 \leftarrow \frac{y_a y'_a + y_b y'_b}{y'_a y'_b}$$

$$r_2 \leftarrow \frac{(b-a)(y_a + y_b)}{y_b - y_a}$$

$$t \leftarrow \frac{a y_b - b y_a - y_a y_b \frac{r_1 - r_2}{y_b - y_a}}{y_b - y_a}$$

Fin si

Si ($t < a$) ou ($t > b$)

protection contre les t hors intervalle

$$t \leftarrow \frac{a y_b - b y_a}{y_b - y_a}$$

Fin si

$$m \leftarrow \frac{a+b}{2}$$

$$\delta \leftarrow \varepsilon \times \max(1, |m|)$$

Si ($t < a + \delta$)

protection contre les t trop proches des bornes (stagnation)

Si ($b - a < 1000\delta$)

$$t \leftarrow m$$

Sinon

$$t \leftarrow 0,999a + 0,001b$$

Fin si

Sinon

Si ($t > b - \delta$)

protection contre les t trop proches des bornes (stagnation)

Si ($b - a < 1000\delta$)

$$t \leftarrow m$$

Sinon

$$t \leftarrow 0,001a + 0,999b$$

Fin si

Fin si

Fin si

évaluer $f'(t)$

Si $f'(t)$ est du même signe que $f'(a)$

$$a \leftarrow t$$

Sinon

$$b \leftarrow t$$

Fin si

Jusqu'à vérification du critère de convergence ou jusqu'à avoir $b - a \leq \varepsilon$

Si convergence atteinte

Retourner le point ayant convergé

Sinon

Retourner le dernier point calculé

Fin si

H.3.3 raffinage des zéros

Utilisation des seuls dérivées premières. La recherche des zéros avec les dérivées premières est similaire à la recherche des extremums avec les dérivées secondes, en remplaçant f' par f et f'' par f' .

Utilisation des dérivées premières et secondes. Lorsque l'on dispose des dérivées secondes, on applique une méthode de Newton améliorée par l'adjonction de la dérivée seconde, mais on se contente de l'appliquer en une seule borne²⁷.

La progression d'un point à l'autre se fait à l'aide de la formule suivante :

$$t = a - \frac{2f(a)f'(a)}{2f'^2(a) - f(a)f''(a)}$$

²⁷simplement parce que l'auteur ne s'est pas préoccupé d'étendre la méthode à deux points dans ce cas ...

La convergence de ce schéma est d'ordre 3 comme le montre l'analyse de l'erreur ε_n à chaque itération :

$$\varepsilon_{n+1} = \frac{3f''(x_0) - 2f'(x_0)f'''(x_0)}{12f'^2(x_0)}\varepsilon_n^3 + O(\varepsilon_n^4)$$

Il faut tout de même signaler que si le surcoût du calcul simultané de la fonction et de sa dérivée première est largement compensé par le gain d'une méthode de Newton (même en un seul point), les dérivées secondes n'apportent souvent pas une diminution suffisante du nombre d'évaluations de la fonction pour être globalement rentable en nombre d'opérations.

I rotations de Cardan

Les rotations de CARDAN sont des moyens pratiques de représenter l'orientation des satellites, surtout quand elle est peu éloignée d'une orientation de référence définie par un repère simple.

Elles peuvent également être utilisées pour définir des orientations sur un domaine de validité très large lorsque les consignes ne sont pas entrées directement par l'utilisateur mais calculées²⁸ ; en effet les singularités des rotations de CARDAN existent mais sont limitées à de petites zones, alors que les champs de vue, les éclipses, les alignements, ou les inhibitions obligent souvent à faire des prouesses si l'on veut utiliser un seul jeu de capteurs sur une plage de temps étendue.

Il s'avère que dans tous les cas de capteurs mesurant des angles de CARDAN, on peut trouver un vecteur particulier exprimable en repère de référence et un cône exprimé en repère satellite tels que le vecteur soit sur le cône dans l'orientation désirée. Selon les capteurs, soit le vecteur dépend de l'angle mesuré et le cône est constant, soit le vecteur est constant et le cône dépend de l'angle mesuré. On peut donc systématiquement se ramener au modèle classique des capteurs géométriques.

Les sections suivantes donnent les équations nécessaires à cette modélisation dans tous les cas possibles d'angles de CARDAN. La démarche générale est la suivante :

Pour trouver les angles à partir de l'orientation, on commence par exprimer une matrice M telle que $u_{\text{cap}} = Mu_{\text{in}}$, où u_{cap} donne les coordonnées d'un vecteur quelconque \vec{u} en repère capteur connaissant ses coordonnées u_{in} en repère de référence. Cette matrice est calculable dès lors que l'on connaît l'orientation. On remarque toujours que la matrice est inchangée par une transformation particulière²⁹, ce qui permet d'imposer qu'un des angles soit systématiquement pris entre $-\pi/2$ et $+\pi/2$, et donc permet de trouver l'angle ne connaissant que son sinus. Les autres angles se déduisent de la matrice et du premier angle. Dans la pratique, comme seuls quelques éléments matriciels sont utiles, on ne calcule *jamais* cette matrice : on se contente de calculer l'image ou l'image réciproque d'un vecteur canonique pour obtenir la ligne ou la colonne utile.

Pour la modélisation des cônes de consignes connaissant les angles, on remarque dans la matrice précédente certains éléments plus simples que les autres, et on construit des vecteurs bien choisis en repère inertiel à partir de ces coordonnées simples, vecteurs qui ne dépendent que d'un angle à chaque fois et présentent de bonnes propriétés (à savoir que leur image est sur un cône en repère capteur, que l'on projette simplement en repère satellite).

I.1 ordre : lacet, puis roulis, puis tangage

La matrice M permettant de convertir un vecteur exprimé en repère de référence en lui-même mais exprimé en repère capteur vaut :

$$M(\psi, \varphi, \theta) = \begin{bmatrix} \cos \theta \cos \psi - \sin \theta \sin \varphi \sin \psi & \cos \theta \sin \psi + \sin \theta \sin \varphi \cos \psi & -\sin \theta \cos \varphi \\ -\cos \varphi \sin \psi & \cos \varphi \cos \psi & \sin \varphi \\ \sin \theta \cos \psi + \cos \theta \sin \varphi \sin \psi & \sin \theta \sin \psi - \cos \theta \sin \varphi \cos \psi & \cos \theta \cos \varphi \end{bmatrix}$$

²⁸ par exemple par un filtre qui lisse des informations disparates sur une plage de temps

²⁹ la transformation dépend de l'ordre des rotations, mais il en existe une pour chaque ordre

M est inchangée par la transformation :

$$\begin{aligned}\psi &\rightarrow \pi + \psi \\ \varphi &\rightarrow \pi - \varphi \\ \theta &\rightarrow \pi + \theta\end{aligned}$$

il est donc licite d'imposer $\varphi \in [-\pi/2; +\pi/2]$. On peut donc déduire les angles de la matrice :

$$\begin{aligned}\psi &= \operatorname{atan2}(-M_{2,1}, M_{2,2}) \\ \varphi &= \arcsin(M_{2,3}) \\ \theta &= \operatorname{atan2}(-M_{1,3}, M_{3,3})\end{aligned}$$

Pour modéliser analytiquement les consignes connaissant les angles, on utilise la forme simple des éléments de la deuxième ligne et de la dernière colonne de M pour construire des vecteurs particuliers.

Si on connaît ψ , on trouve la consigne :

$$\vec{u}_{\text{in}} = \cos \psi \vec{X}_{\text{in}} + \sin \psi \vec{Y}_{\text{in}} \text{ doit être sur le cône } (\vec{Y}_{\text{cap}}, \frac{\pi}{2})$$

Si on connaît φ , on trouve la consigne :

$$\vec{Z}_{\text{in}} \text{ doit être sur le cône } (\vec{Y}_{\text{cap}}, \frac{\pi}{2} - \varphi)$$

Si on connaît θ , on trouve la consigne :

$$\vec{Z}_{\text{in}} \text{ doit être sur le cône } (\cos \theta \vec{X}_{\text{cap}} + \sin \theta \vec{Z}_{\text{cap}}, \frac{\pi}{2})$$

I.2 ordre : lacet, puis tangage, puis roulis

La matrice M permettant de convertir un vecteur exprimé en repère de référence en lui-même mais exprimé en repère capteur vaut :

$$M(\psi, \theta, \varphi) = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ -\cos \varphi \sin \psi + \sin \varphi \sin \theta \cos \psi & \cos \varphi \cos \psi + \sin \varphi \sin \theta \sin \psi & \sin \varphi \cos \theta \\ \sin \varphi \sin \psi + \cos \varphi \sin \theta \cos \psi & -\sin \varphi \cos \psi + \cos \varphi \sin \theta \sin \psi & \cos \varphi \cos \theta \end{bmatrix}$$

M est inchangée par la transformation :

$$\begin{aligned}\psi &\rightarrow \pi + \psi \\ \theta &\rightarrow \pi - \theta \\ \varphi &\rightarrow \pi + \varphi\end{aligned}$$

il est donc licite d'imposer $\theta \in [-\pi/2; +\pi/2]$. On peut donc déduire les angles de la matrice :

$$\begin{aligned}\psi &= \operatorname{atan2}(M_{1,2}, M_{1,1}) \\ \theta &= -\arcsin(M_{1,3}) \\ \varphi &= \operatorname{atan2}(M_{2,3}, M_{3,3})\end{aligned}$$

Pour modéliser analytiquement les consignes connaissant les angles, on utilise la forme simple des éléments de la première ligne et de la dernière colonne de M pour construire des vecteurs particuliers.

Si on connaît ψ , on trouve la consigne :

$$\vec{u}_{\text{in}} = -\sin \psi \vec{X}_{\text{in}} + \cos \psi \vec{Y}_{\text{in}} \text{ doit être sur le cône } (\vec{X}_{\text{cap}}, \frac{\pi}{2})$$

Si on connaît θ , on trouve la consigne :

$$\vec{Z}_{\text{in}} \text{ doit être sur le cône } (-\vec{X}_{\text{cap}}, \frac{\pi}{2} - \theta)$$

Si on connaît φ , on trouve la consigne :

$$\vec{Z}_{\text{in}} \text{ doit être sur le cône } (\cos \varphi \vec{Y}_{\text{cap}} - \sin \varphi \vec{Z}_{\text{cap}}, \frac{\pi}{2})$$

I.3 ordre : roulis, puis lacet, puis tangage

La matrice M permettant de convertir un vecteur exprimé en repère de référence en lui-même mais exprimé en repère capteur vaut :

$$M(\varphi, \psi, \theta) = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi \cos \varphi + \sin \theta \sin \varphi & \cos \theta \sin \psi \sin \varphi - \sin \theta \cos \varphi \\ -\sin \psi & \cos \psi \cos \varphi & \cos \psi \sin \varphi \\ \sin \theta \cos \psi & \sin \theta \sin \psi \cos \varphi - \cos \theta \sin \varphi & \sin \theta \sin \psi \sin \varphi + \cos \theta \cos \varphi \end{bmatrix}$$

M est inchangée par la transformation :

$$\varphi \rightarrow \pi + \varphi$$

$$\psi \rightarrow \pi - \psi$$

$$\theta \rightarrow \pi + \theta$$

il est donc licite d'imposer $\psi \in [-\pi/2; +\pi/2]$. On peut donc déduire les angles de la matrice :

$$\varphi = \text{atan2}(M_{2,3}, M_{2,2})$$

$$\psi = -\arcsin(M_{2,1})$$

$$\theta = \text{atan2}(M_{3,1}, M_{1,1})$$

Pour modéliser analytiquement les consignes connaissant les angles, on utilise la forme simple des éléments de la deuxième ligne et de la première colonne de M pour construire des vecteurs particuliers.

Si on connaît φ , on trouve la consigne :

$$\vec{u}_{\text{in}} = -\sin \varphi \vec{Y}_{\text{in}} + \cos \varphi \vec{Z}_{\text{in}} \text{ doit être sur le cône } (\vec{Y}_{\text{cap}}, \frac{\pi}{2})$$

Si on connaît ψ , on trouve la consigne :

$$\vec{X}_{\text{in}} \text{ doit être sur le cône } (-\vec{Y}_{\text{cap}}, \frac{\pi}{2} - \psi)$$

Si on connaît θ , on trouve la consigne :

$$\vec{X}_{\text{in}} \text{ doit être sur le cône } (\cos \theta \vec{Z}_{\text{cap}} - \sin \theta \vec{X}_{\text{cap}}, \frac{\pi}{2})$$

I.4 ordre : roulis, puis tangage, puis lacet

La matrice M permettant de convertir un vecteur exprimé en repère de référence en lui-même mais exprimé en repère capteur vaut :

$$M(\varphi, \theta, \psi) = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \varphi + \sin \psi \cos \varphi & -\cos \psi \sin \theta \cos \varphi + \sin \psi \sin \varphi \\ -\sin \psi \cos \theta & -\sin \psi \sin \theta \sin \varphi + \cos \psi \cos \varphi & \sin \psi \sin \theta \cos \varphi + \cos \psi \sin \varphi \\ \sin \theta & -\cos \theta \sin \varphi & \cos \theta \cos \varphi \end{bmatrix}$$

M est inchangée par la transformation :

$$\begin{aligned} \varphi &\rightarrow \pi + \varphi \\ \theta &\rightarrow \pi - \theta \\ \psi &\rightarrow \pi + \psi \end{aligned}$$

il est donc licite d'imposer $\theta \in [-\pi/2; +\pi/2]$. On peut donc déduire les angles de la matrice :

$$\begin{aligned} \varphi &= \text{atan2}(-M_{3,2}, M_{3,3}) \\ \theta &= \arcsin(M_{3,1}) \\ \psi &= \text{atan2}(-M_{2,1}, M_{1,1}) \end{aligned}$$

Pour modéliser analytiquement les consignes connaissant les angles, on utilise la forme simple des éléments de la dernière ligne et de la première colonne de M pour construire des vecteurs particuliers.

Si on connaît φ , on trouve la consigne :

$$\vec{u}_{\text{in}} = \cos \varphi \vec{Y}_{\text{in}} + \sin \varphi \vec{Z}_{\text{in}} \text{ doit être sur le cône } (\vec{Z}_{\text{cap}}, \frac{\pi}{2})$$

Si on connaît θ , on trouve la consigne :

$$\vec{X}_{\text{in}} \text{ doit être sur le cône } (\vec{Z}_{\text{cap}}, \frac{\pi}{2} - \theta)$$

Si on connaît ψ , on trouve la consigne :

$$\vec{X}_{\text{in}} \text{ doit être sur le cône } (\cos \psi \vec{Y}_{\text{cap}} + \sin \psi \vec{X}_{\text{cap}}, \frac{\pi}{2})$$

I.5 ordre : tangage, puis lacet, puis roulis

La matrice M permettant de convertir un vecteur exprimé en repère de référence en lui-même mais exprimé en repère capteur vaut :

$$M(\theta, \psi, \varphi) = \begin{bmatrix} \cos \psi \cos \theta & \sin \psi & -\cos \psi \sin \theta \\ -\cos \varphi \sin \psi \cos \theta + \sin \varphi \sin \theta & \cos \varphi \cos \psi & \cos \varphi \sin \psi \sin \theta + \sin \varphi \cos \theta \\ \sin \varphi \sin \psi \cos \theta + \cos \varphi \sin \theta & -\sin \varphi \cos \psi & -\sin \varphi \sin \psi \sin \theta + \cos \varphi \cos \theta \end{bmatrix}$$

M est inchangée par la transformation :

$$\begin{aligned}\theta &\rightarrow \pi + \theta \\ \psi &\rightarrow \pi - \psi \\ \varphi &\rightarrow \pi + \varphi\end{aligned}$$

il est donc licite d'imposer $\psi \in [-\pi/2; +\pi/2]$. On peut donc déduire les angles de la matrice :

$$\begin{aligned}\theta &= \operatorname{atan2}(-M_{1,3}, M_{1,1}) \\ \psi &= \arcsin(M_{1,2}) \\ \varphi &= \operatorname{atan2}(-M_{3,2}, M_{2,2})\end{aligned}$$

Pour modéliser analytiquement les consignes connaissant les angles, on utilise la forme simple des éléments de la première ligne et de la deuxième colonne de M pour construire des vecteurs particuliers.

Si on connaît θ , on trouve la consigne :

$$\vec{u}_{\text{in}} = \sin \theta \vec{X}_{\text{in}} + \cos \theta \vec{Z}_{\text{in}} \text{ doit être sur le cône } (\vec{X}_{\text{cap}}, \frac{\pi}{2})$$

Si on connaît ψ , on trouve la consigne :

$$-\vec{Y}_{\text{in}} \text{ doit être sur le cône } (-\vec{X}_{\text{cap}}, \frac{\pi}{2} - \psi)$$

Si on connaît φ , on trouve la consigne :

$$-\vec{Y}_{\text{in}} \text{ doit être sur le cône } (\cos \varphi \vec{Z}_{\text{cap}} + \sin \varphi \vec{Y}_{\text{cap}}, \frac{\pi}{2})$$

I.6 ordre : tangage, puis roulis, puis lacet

La matrice M permettant de convertir un vecteur exprimé en repère de référence en lui-même mais exprimé en repère capteur vaut :

$$M(\theta, \varphi, \psi) = \begin{bmatrix} \cos \psi \cos \theta + \sin \psi \sin \varphi \sin \theta & \sin \psi \cos \varphi & -\cos \psi \sin \theta + \sin \psi \sin \varphi \cos \theta \\ -\sin \psi \cos \theta + \cos \psi \sin \varphi \sin \theta & \cos \psi \cos \varphi & \sin \psi \sin \theta + \cos \psi \sin \varphi \cos \theta \\ \cos \varphi \sin \theta & -\sin \varphi & \cos \varphi \cos \theta \end{bmatrix}$$

M est inchangée par la transformation :

$$\begin{aligned}\theta &\rightarrow \pi + \theta \\ \varphi &\rightarrow \pi - \varphi \\ \psi &\rightarrow \pi + \psi\end{aligned}$$

il est donc licite d'imposer $\varphi \in [-\pi/2; +\pi/2]$. On peut donc déduire les angles de la matrice :

$$\begin{aligned}\theta &= \operatorname{atan2}(M_{3,1}, M_{3,3}) \\ \varphi &= -\arcsin(M_{3,2}) \\ \psi &= \operatorname{atan2}(M_{1,2}, M_{2,2})\end{aligned}$$

Pour modéliser analytiquement les consignes connaissant les angles, on utilise la forme simple des éléments de la dernière ligne et de la deuxième colonne de M pour construire des vecteurs particuliers.

Si on connaît θ , on trouve la consigne :

$$\vec{u}_{\text{in}} = \cos \theta \vec{X}_{\text{in}} - \sin \theta \vec{Z}_{\text{in}} \text{ doit être sur le cône } (\vec{Z}_{\text{cap}}, \frac{\pi}{2})$$

Si on connaît φ , on trouve la consigne :

$$-\vec{Y}_{\text{in}} \text{ doit être sur le cône } (\vec{Z}_{\text{cap}}, \frac{\pi}{2} - \varphi)$$

Si on connaît ψ , on trouve la consigne :

$$-\vec{Y}_{\text{in}} \text{ doit être sur le cône } (\cos \psi \vec{X}_{\text{cap}} - \sin \psi \vec{Y}_{\text{cap}}, \frac{\pi}{2})$$