

ÉMILIE SABRIÉ

**ANALYSE D'UN MÉCANISME DE SIMULATION DE VOL
SPHÉRIQUE ET SON CONTRÔLE EN TEMPS RÉEL**

Mémoire présenté
à la Faculté des études supérieures de l'Université Laval
dans le cadre du programme de maîtrise en génie mécanique
pour l'obtention du grade de maître ès sciences (M.Sc.)

FACULTÉ DES SCIENCES ET DE GÉNIE
UNIVERSITÉ LAVAL
QUÉBEC

DÉCEMBRE 2004

Résumé

La présente recherche vise l'analyse d'un mécanisme de simulation de vol sphérique à trois degrés de liberté ainsi que le développement de son système de contrôle en temps réel. L'architecture mécanique du mécanisme en question est d'abord décrite, puis son analyse cinématique, permettant de définir l'espace atteignable ainsi que les performances de dextérité du mécanisme, est réalisée. Le développement du modèle dynamique du mécanisme par la méthode de Lagrange est ensuite fait, et la conception d'un système d'équilibrage statique partiel permettant de diminuer les efforts aux actionneurs est discutée. Par la suite, le développement et la programmation du système permettant de créer et contrôler les mouvements de la plate-forme sont décrits. Finalement, différents résultats obtenus avec le prototype fabriqué au laboratoire, mettant en pratique plusieurs aspects étudiés précédemment, sont présentés.

Avant-propos

Je tiens tout d’abord à remercier le professeur Clément Gosselin, directeur du Laboratoire de Robotique de l’Université Laval, qui après m’avoir déjà encadrée pendant mon projet de fins d’études, m’a donné la chance de poursuivre mes études de deuxième cycle sous sa supervision. C’est lui qui m’a donné le goût de la robotique, par sa passion, sa pédagogie qui rendent chaque chose si simple à comprendre et si intéressante. Ses réalisations dans le domaine de la robotique, sans parler de ses innombrables idées qui j’en suis sûre deviendront réalité dans un avenir proche, ont toujours forcé mon admiration.

Je voudrais aussi dire un grand merci à toutes les personnes qui ont été membres du laboratoire pendant les deux ans de ma maîtrise, car ils ont tous participé à cette ambiance à la fois décontractée et créative qui restera dans ma mémoire. Je voudrais remercier en particulier Simon, Boris et Thierry, les meilleurs professionnels de recherche que le laboratoire puisse avoir, qui m’ont guidé et tant appris, Lionel, Marc, et Ilian dont les conseils et l’amitié ont été d’un grand secours, Pascal et Nicolas Pouliot qui m’ont précédé dans ce projet, et desquels je me suis inspirée, et Jean Philippe, Nicolas Bouchard, Mathieu et Augustin les vaillants camarades de ma maîtrise et de mes cours. Finalement, je remercie toute ma famille et mes amis pour leur soutien et leur amour.

Table des matières

Résumé	i
Avant-propos	ii
Table des matières	iii
Liste des tableaux	vii
Liste des figures	viii
Introduction	1
1 Architecture finale du simulateur	4
1.1 Les systèmes composant un simulateur	5
1.2 Travaux antérieurs	7
1.2.1 Potentiel des mécanismes à trois degrés de liberté	7
1.2.2 Le cahier des charges	8
1.2.3 La conception	10
1.2.3.1 Le système audiovisuel	10
1.2.3.2 Le système de calcul	11
1.2.3.3 Le système mécanique	12
1.3 Conception mécanique : problématique et solutions	16
1.4 L'architecture mécanique finale	17
1.4.1 Description de l'architecture	17
1.4.2 Degrés de liberté	19

2	Analyse cinématique du simulateur	21
2.1	Notations	22
2.1.1	Définition des repères	22
2.1.2	Coordonnées cartésiennes et matrices de rotation	23
2.1.3	Notations pour le mécanisme	24
2.2	Modèle géométrique du simulateur	25
2.2.1	Problème géométrique inverse	25
2.2.2	Problème géométrique direct	29
2.3	Équations de vitesse et jacobiennes	29
2.3.1	Mécanisme 1	30
2.3.2	Mécanisme 2	32
2.3.3	Relations globales	33
2.4	Analyse cinématique globale	33
2.4.1	Espace atteignable prescrit	33
2.4.2	Dextérité	35
2.5	Cinématique des membrures	37
2.5.1	Membrures du mécanisme 1	37
2.5.2	Membrures du mécanisme 2	40
3	Analyse statique et dynamique du simulateur	42
3.1	Introduction	43
3.2	Développement du modèle dynamique	44
3.2.1	Notations supplémentaires et notions mathématiques utiles	44
3.2.2	Dynamique de la cabine	45
3.2.3	Dynamique de la plaque support	46
3.2.4	Dynamique des membrures	47
3.2.5	Modèle dynamique complet du système	49
3.3	Simplification du modèle pour le contrôle	49
3.4	Équilibrage statique partiel	51
3.4.1	Introduction	51
3.4.2	Solution mise en oeuvre	53
3.4.2.1	Équilibrage de l'effecteur	53
3.4.2.2	Optimisation de la raideur des ressorts	55
3.5	Exemples numériques et comparaison avec le modèle ADAMS	58
3.5.1	Simulation d'une manœuvre de décollage	59
3.5.2	Mouvement sinusoïdal	60
3.5.3	Comparaison des différentes approximations	62

4	Architecture du système de calcul	63
4.1	Architecture générale	64
4.1.1	Entrée du système	64
4.1.2	Principe du système	65
4.2	Architecture matérielle et communications	67
4.3	Logiciel X-Plane	70
4.4	Génération des mouvements du simulateur	71
4.4.1	Principe du système d'exploitation	72
4.4.2	L'algorithme de génération des mouvements	73
4.4.3	<i>Washout Filters</i>	75
4.5	Contrôle	78
5	Résultats expérimentaux	81
5.1	Le prototype	82
5.1.1	Système mécanique	82
5.1.2	Système de calcul	83
5.2	Méthodologie d'évaluation	85
5.2.1	Perception des mouvements	86
5.2.2	Production des résultats	87
5.2.3	Manœuvres d'évaluation	88
5.2.4	Indices de performance	90
5.3	Exemples graphiques et analyse	92
5.3.1	Manœuvre d'avion de transport commercial	92
5.3.2	Manœuvre d'avion de chasse	95
5.3.3	Analyse globale	99
	Conclusion	101
	Bibliographie	105
A	Cinématique	109
A.1	Détails de la résolution du PGD	109
A.1.1	Solutions pour l'angle α	110
A.1.2	Solutions pour l'angle β	111
A.1.3	Solutions pour l'angle γ	111
A.2	Conversions entre les conventions d'orientation	112
A.2.1	Principe de la conversion	113
A.2.2	Résolution si les inconnues sont les angles de la convention XYZ	115

A.2.3	Résolution si les inconnues sont les angles de <i>Tilt & Torsion</i> . . .	115
A.3	Cinématique de la membrure d_2	117
A.3.1	Paramètres HD de la patte 2	117
A.3.2	Angles et jacobienne passifs	118
A.3.2.1	Angles passifs	119
A.3.2.2	Cinématique	120
B	Données techniques	121
B.1	Ressorts pour le simulateur à taille réelle	121
B.2	Ressorts pour le prototype	122
B.3	Moteurs pour le prototype	123
B.4	Simulateur à six degrés de liberté	124
C	Glossaire des termes informatiques utilisés	125
D	Listes des fichiers et fonctions programmés	129
D.1	Liste des plugins développés pour X-Plane	129
D.2	Génération des mouvements du simulateur	130
D.2.1	Application Linux	130
D.2.2	Module RTLinux	132

Liste des tableaux

1.1	Résumé du cahier des charges.	10
1.2	Estimé de la distribution des masses.	15
3.1	Caractéristiques des ressorts réels.	57
3.2	Paramètres principaux (en unités SI).	59
3.3	Indicateur de l'erreur $\lambda_a(t)$	62
5.1	Indices de performance pour la manœuvre 1.	95
5.2	Indices de performance pour la manœuvre 2.	99
A.1	Paramètres de Hartenberg-Denavit pour la patte 2.	118
B.1	Caractéristiques des ressorts du prototype.	122
B.2	Débattements et limites physiques de l'architecture de l'UTIAS.	124

Liste des figures

1	Plate-forme de Gough-Stewart.	1
1.1	Sous-systèmes des simulateurs de vol (reproduit de [6]).	5
1.2	Système audiovisuel.	11
1.3	Logiciel X-Plane.	12
1.4	Moteur choisi.	13
1.5	Première architecture proposée pour le simulateur (reproduit de [7]).	14
1.6	Schéma éclaté de l'architecture finale du simulateur.	18
1.7	Schémas cinématiques simplifiés des mécanismes.	19
2.1	Orientation des corps.	22
2.2	Notations pour le mécanisme.	24
2.3	Modèle géométrique de la patte 2.	26
2.4	Les deux solutions du PGI pour la patte 1.	28
2.5	Rotations du <i>Tilt & Torsion</i> : (a) tilt, (b) torsion(reproduit de [8]).	34
2.6	Dextérité locale du mécanisme $\xi(\phi, \theta)$	36
2.7	Schéma cinématique du mécanisme 1.	38
2.8	Schéma cinématique du mécanisme 2 (vue de haut).	40
3.1	Concept d'équilibrage de simulateur (reproduit de [20]).	52
3.2	Paramètres de l'équilibrage statique.	54
3.3	Évolution de Δ en fonction de k	56
3.4	Photographie d'un des ressorts.	57
3.5	Modèle <i>ADAMS</i> du simulateur.	58
3.6	Couples obtenus pour la manœuvre de décollage.	60

3.7	Couples obtenus pour la manœuvre sinusoïdale.	61
4.1	Les commandes de vol.	64
4.2	Schéma des sous-systèmes du simulateur.	65
4.3	Schéma de la configuration matérielle du système de calcul.	67
4.4	Photographie du système de calcul.	68
4.5	Schéma de l'organisation du système d'exploitation RTlinux.	72
4.6	Schéma de l'algorithme de génération des mouvements.	74
4.7	Schéma des <i>Washout Filters</i>	77
4.8	Carte de contrôle de Delta Tau.	79
5.1	Architecture mécanique du prototype.	82
5.2	Évaluation des temps de calculs.	84
5.3	Principe de la production des résultats.	87
5.4	Forces spécifiques perçues lors de la manœuvre.	93
5.5	Déplacements angulaires de Tilt et Torsion.	94
5.6	Déplacements linéaires du simulateur UTIAS.	94
5.7	Forces spécifiques et vitesses angulaires perçues lors de la manœuvre.	96
5.8	Déplacements angulaires de Tilt et Torsion.	98
5.9	Déplacements linéaires du simulateur UTIAS.	98
A.1	Schématization de la patte 2.	117
A.2	Repères de la patte.	118
B.1	Caractéristiques données par le fournisseur.	122
B.2	Caractéristiques des moteurs.	123
B.3	Simulateur à 6 degrés de liberté de l'UTIAS.	124

Introduction

Les simulateurs de mouvement

De nos jours, le domaine de la simulation de mouvement reste la principale application commerciale des mécanismes parallèles. En particulier, la plate-forme de Gough-Stewart [1, 2], illustrée à la figure 1, est utilisée dans la très grande majorité des applications de simulateurs de vol en tant que système de mouvement à six degrés de liberté. Elle est composée de six vérins actionnés reliés à une base par des joints de cardan et reliés à une plate-forme mobile par des articulations sphériques.

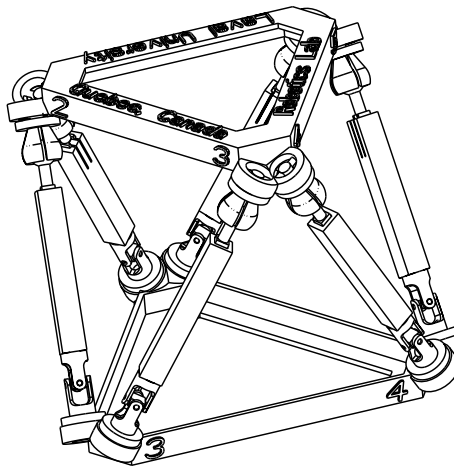


FIG. 1 – Plate-forme de Gough-Stewart.

Le principal avantage de cette architecture parallèle sur une architecture sérielle est que toutes les pattes coopèrent pour retenir les forces créées à la plate-forme. Grâce à cela, le mécanisme est plus rigide, possède un meilleur rapport charge/masse totale, et ses actionneurs sont plus petits car les vérins subissent seulement des forces de traction/compression. Ainsi, cette plate-forme peut soutenir et déplacer une reproduction complète de cabine d'avion par exemple, ce qui contribue au réalisme de la simulation. Cependant, l'espace atteignable et la dextérité globale d'un tel mécanisme sont généralement plus faibles que ceux obtenus avec un mécanisme sériel. Par ailleurs, le prix d'un tel système reste prohibitif, et limite son application à des simulateurs de vol de très haut niveau, utilisés par des organisations militaires ainsi que des compagnies d'aviation pour l'entraînement des pilotes.

Avec le développement et même la relative démocratisation des simulateurs dédiés à l'amusement et des systèmes de réalité virtuelle, il y a eu un intérêt croissant pour les simulateurs de vol plus simples, plus compacts, et peu coûteux. Une grande variété de technologies et de moyens ont été mis en oeuvre dans l'industrie pour développer des simulateurs de mouvement plus simples, et possédant un nombre de degrés de liberté réduit. Cependant, à part quelques travaux portant sur des conceptions alternatives à une plate-forme classique [3, 4, 6, 7], très peu de recherches ont été effectuées pour allier un système simple et peu coûteux à un réalisme suffisant des mouvements pour appliquer le système à l'entraînement des pilotes.

Problématique

C'est donc ce défi que s'est imposé le laboratoire de robotique de l'Université Laval en débutant son projet de simulateur de vol : concevoir un simulateur comportant un nombre restreint de degrés de liberté, possédant une architecture simple, étant très peu coûteux, et à la fois pouvant créer des sensations de mouvement assez réalistes pour être utilisé lors de l'entraînement des pilotes d'avions (du moins pour les premières phases). Pour réaliser ce défi, plusieurs étapes ont été franchies. Tout d'abord, une comparaison des sensations pouvant être créées par différentes architectures à trois degrés de liberté a été réalisée, afin de choisir le type d'architecture le plus adéquat. Ensuite, en se basant sur ces résultats, une conception générale du mécanisme a été réalisée.

C'est à ce stade du projet que s'inscrit le sujet de la présente maîtrise, qui vise la mise en pratique de ces deux aspects, c'est à dire l'aspect mécanique avec l'analyse

de l'architecture sphérique finale et le développement de son équilibrage statique, et l'aspect des sensation de mouvement avec le développement et la programmation du système permettant de créer et contrôler les mouvements de la plate-forme. Les objectifs principaux poursuivis par le sujet de cette maîtrise sont donc les suivants :

1) L'étude cinématique et dynamique du mécanisme ainsi que le développement de son système d'équilibrage statique,

2) le développement du système de génération des mouvements et de contrôle en temps réel du mécanisme.

Structure du mémoire

Le présent document traite de travaux de recherche ayant été réalisés de septembre 2002 à août 2004. Sa structure s'élabore en deux parties, en correspondance avec les deux objectifs principaux poursuivis. Le mémoire comprend au total cinq chapitres.

La première partie est décrite dans les trois premiers chapitres, et traite de l'analyse du mécanisme. Cette partie a fait l'objet d'une présentation à la conférence RoManSy portant sur la conception, la dynamique, le contrôle, et les applications en robotique, et se déroulant à l'agence spatiale canadienne en juin 2004 [5]. La seconde partie traite quant à elle du développement du système de génération de mouvement et du contrôle du mécanisme, ce qui met en pratique et utilise de nombreuses notions de la première partie. Quelques résultats obtenus avec le prototype de simulateur contrôlé en temps réel viennent conclure cette partie.

D'une manière plus spécifique, le chapitre 1 introduira, en partant d'un résumé des travaux antérieurs réalisés pour ce projet, l'architecture finale du simulateur. Le chapitre 2 sera ensuite consacré à l'analyse cinématique complète du mécanisme, puis le chapitre 3 présentera le modèle dynamique du mécanisme ainsi que son équilibrage statique. Ensuite, le développement du système de calcul du mécanisme, et plus particulièrement la génération des mouvements et le contrôle en temps réel, sera décrit dans le chapitre 4. Finalement, le chapitre 5 présentera différents résultats obtenus avec le prototype fabriqué au laboratoire, mettant en pratique différents aspects discutés dans le mémoire.

Chapitre 1

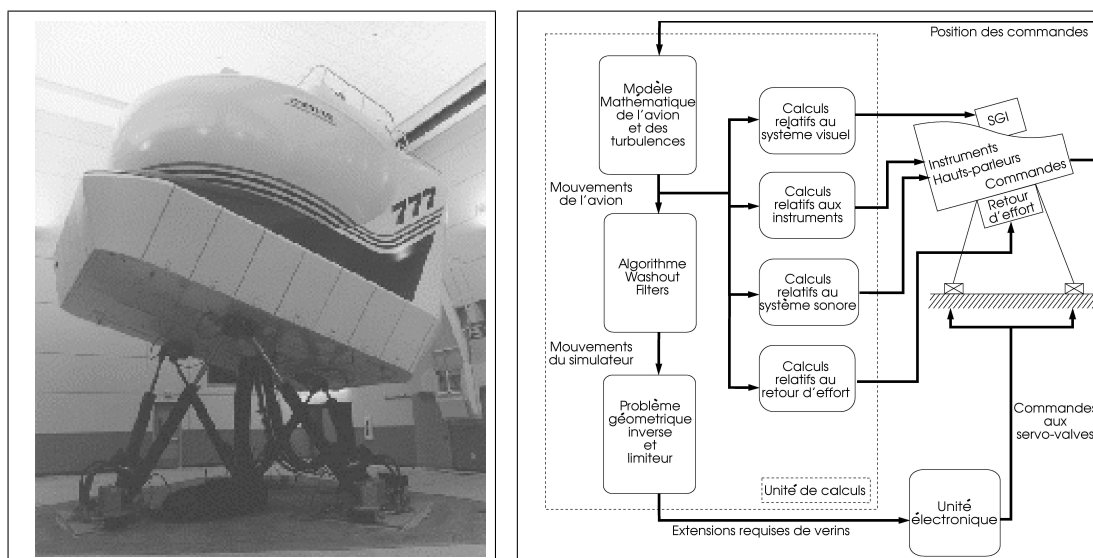
Architecture finale du simulateur

Le sujet de la présente maîtrise s'inscrit dans le cadre du projet du laboratoire de robotique de concevoir un simulateur de vol innovant. Différentes recherches réalisées pour ce projet ont permis d'aboutir à la conception finale du simulateur, aussi bien pour sa plate-forme de mouvement que pour les autres sous-systèmes du simulateur.

Dans ce chapitre, les différents sous-systèmes inhérents aux simulateurs de vol seront d'abord identifiés. Puis les travaux antérieurs réalisés au laboratoire sur ces sous-systèmes seront présentés. Enfin, à partir de la description des problèmes rencontrés et des solutions apportées, l'architecture mécanique finale sera détaillée.

1.1 Les systèmes composant un simulateur

L'identification des différents systèmes constituant un simulateur de vol moderne permet de constater que la plate-forme de génération de mouvement, bien que très importante, est appuyée par plusieurs autres sous-systèmes qui contribuent tous au but visé : le réalisme global de la simulation. Pour illustrer ces différents systèmes, la figure 1.1(a) représente un simulateur de vol moderne destiné aux appareils *Boeing 777*. Un tel système, bien que très complexe, peut être subdivisé en plusieurs sous-systèmes ce qui nous permet de mieux comprendre son fonctionnement et surtout de réaliser l'inter-dépendance qui existe entre ses composants. La figure 1.1(b) décrit schématiquement ces relations.



(a) Simulateur de B777

(b) Sous-systèmes d'un simulateur de vol.

FIG. 1.1 – Sous-systèmes des simulateurs de vol (reproduit de [6]).

Dans un avion, comme sur un navire, le capitaine est maître à bord. Le simulateur de vol ne fait pas exception à la règle et toute l'aventure de la simulation débute lorsque le pilote imprime les déplacements aux commandes de vol (palonnier, gouvernail, manette des gaz, etc.) et aux commandes de navigation (radio, feux de position, etc.). L'unité de calcul, représentée par le grand rectangle en lignes tiretées à la figure 1.1(b), entre alors en action. Le modèle mathématique de l'avion interprétera les commandes du pilote pour en déduire le comportement de l'avion, en tenant compte des effets

aérodynamiques, du comportement des moteurs, des turbulences, etc. C'est à partir d'ici que la plupart des sous-systèmes peuvent entrer en action.

Le sous-système de génération d'image (SGI) produira, par l'intermédiaire d'un ou plusieurs projecteurs, une image de synthèse représentant la scène visible du cockpit. Le SGI utilisera pour ce faire d'importantes bases de données numérisées provenant souvent de photos satellites et couvrant de vastes territoires réels.

Comme la cabine de pilotage est le plus souvent représentée intégralement dans le simulateur il est normal de prévoir un sous-système dont la fonction sera de générer la réponse de l'altimètre, du radar embarqué, des systèmes de contrôle des moteurs et de tous les autres instruments de vol. Le pilote pourra alors en tenir compte pour corriger ses manœuvres ou carrément pratiquer le vol aux instruments.

Un sous-système produira également l'environnement sonore correspondant aux conditions réelles de vol par l'intermédiaire d'un système ambiophonique. Le sifflement de l'air sur la carlingue, le bruit (normal ou non) des moteurs et l'ouverture du train d'atterrissage pourront ainsi être détectés par le pilote.

Finalement, le dernier sous-système de ce groupe est celui qui aura la fonction de générer une force de retour réaliste aux commandes que le pilote actionne.

Bien qu'à ce point de la simulation aucun mouvement n'ait été généré par le simulateur, le pilote est pourtant alimenté de plusieurs stimuli qui, par leurs effets combinés, produisent déjà une simulation très réaliste.

Les mouvements de l'avion ne pourront pas être simulés intégralement par la plateforme de mouvement qui supporte le cockpit en raison de ses limitations physiques inhérentes : elle est invariablement fixée au sol alors que l'avion ne l'est pas ! L'algorithme du *Washout Filters*¹ entre alors en action. Cet algorithme, qui sera détaillé au chapitre 4, doit générer les mouvements du simulateur en fonction des mouvements de l'avion simulé en réalisant le meilleur compromis face à cette limitation de l'espace atteignable.

Les mouvements du simulateur dictés par l'algorithme du *Washout Filters* sont ensuite convertis en mouvements des actionneurs par les calculs cinématiques, qui dépendent de l'architecture de la plateforme. Pour l'exemple illustré à la figure 1.1(b), les déplacements et les rotations imposés à la cabine sont convertis en longueurs correspondantes

¹L'expression *Washout Filters*, d'usage dans le domaine de la simulation, fait référence à un filtre de *lessivage*, qui enlèvera tout le superflu pour ne conserver que l'essentiel. Aucune expression valable ne semble faire consensus dans les milieux francophones de la simulation et ce sera donc l'expression anglaise qui sera utilisée dans ce mémoire.

pour chacun des six vérins hydrauliques qui supportent la plate-forme, puis asservis au niveau des servo-valves.

1.2 Travaux antérieurs

1.2.1 Potentiel des mécanismes à trois degrés de liberté

Comme on l'a vu en introduction, les simulateurs de vol commerciaux utilisent comme systèmes de mouvement presque exclusivement des plates-formes parallèles à six degrés de liberté de type Gough-Stewart. La première étape d'une conception alternative a donc consisté en l'étude du potentiel de plates-formes possédant un nombre de degrés de liberté restreint. Cette étape a constitué le premier objectif des recherches de Nicolas Pouliot [6]. Pour cela, il a réalisé une étude comparative de la qualité des sensations de mouvements obtenues par différents types de plates-formes simplifiées, possédant trois degrés de liberté, par rapport à celles obtenues avec une plate-forme classique à six degrés de liberté. La comparaison a été effectuée à partir de manœuvres d'avion *Boeing 747*, en utilisant un modèle du système vestibulaire humain, qui est le centre de la perception des mouvements. Les résultats de cette étude ont montrés que, pour ces manœuvres d'avion de transport, certaines architectures à trois degrés de liberté permettaient de donner au pilote des sensations de mouvement comparables à celles créées par une plate-forme classique. L'architecture la plus prometteuse fut l'architecture dite sphérique, qui possède trois degrés de liberté en rotation autour d'un point fixe.

À partir de ces résultats très encourageants, la deuxième étape du projet a consisté en l'étude de systèmes mécaniques à trois degrés de liberté pouvant servir de plate-forme de mouvement pour un simulateur de vol. Ce fut le deuxième objectif de la recherche de Nicolas Pouliot, et cinq architectures cinématiques potentielles furent analysées et optimisées. Pour une d'entre elles, un design conceptuel fut réalisé afin de démontrer la faisabilité technique d'un tel simulateur.

1.2.2 Le cahier des charges

Suite à ces études, Pascal Dufour [7] a ensuite travaillé sur la conception du simulateur. La première étape de la conception pratique fut l'identification des principales caractéristiques désirées pour le simulateur. Un cahier des charges, détaillant les caractéristiques de chacun des systèmes du simulateur fut donc rédigé par Pascal Dufour. Au cours des étapes suivantes, il fut quelque peu modifié pour réaliser des compromis suivant certains des objectifs. Voici, sous forme d'une liste de points à respecter classés selon le sous-système auquel ils font référence, la version finale du cahier des charges.

- Caractéristiques générales
 - Encombrement et transportabilité : le simulateur doit être compact, et transportable sans avoir recours à de la machinerie lourde. Son encombrement réduit doit permettre aux petites compagnies de posséder un tel simulateur. Enfin, ces deux caractéristiques doivent permettre au simulateur d'être intégré à des systèmes de réalité virtuelle et de divertissement.
 - Coût : le prix du simulateur doit permettre aux petites compagnies qui actuellement louent des simulateurs commerciaux, ainsi qu'aux écoles de pilotage, de se procurer le système.
- Système mécanique
 - Qualité de la simulation : la qualité de simulation doit être comparable aux simulateurs de vol commerciaux à six degrés de liberté, et permettre au système d'être certifié pour les premières phases d'apprentissage des pilotes.
 - Architecture : l'architecture mécanique doit rester plus simple que celle d'un simulateur classique, et sa maintenance doit être aisée.
 - Source d'énergie : la source d'énergie utilisée doit être facile d'accès (pour ne pas limiter les clients et les applications), peu encombrante et peu bruyante. Aussi son entretien et sa maintenance doivent être le plus facile possible.

- Système audiovisuel
 - Environnement visuel : la qualité de la projection des images doit être suffisante pour immerger le pilote dans l'environnement créé.
 - Environnement sonore : la qualité du système sonore doit être suffisante pour immerger le pilote dans l'environnement simulé et l'isoler des bruits extérieurs à la simulation.
- Système de calcul
 - Modèle dynamique de l'avion : le modèle mathématique de l'avion simulé doit être assez réaliste pour donner au pilote l'impression qu'il est bien aux commandes de cet avion. En particulier, ce modèle devra être assez réaliste pour passer les tests de certification réalisés par les organismes de contrôle comme la FAA (*Federal Aviation Administration*, organisme américain qui détermine les réglementations de l'aviation civile) et être approuvé pour servir au moins dans les premières phases d'entraînement des pilotes.
 - Contrôle temps réel : le temps de calcul entre le moment où le pilote imprime les mouvements aux commandes de vol et celui où le mécanisme bouge suite à cette commande doit rester imputable à l'inertie de l'appareil.
 - Synchronisation des différents systèmes : le décalage entre le système audiovisuel et le système mécanique doit rester le plus faible possible, afin d'éviter l'effet du "mal du simulateur".
 - Sécurité : étant donné qu'une personne se trouve dans le simulateur, on doit s'assurer qu'elle ne puisse en aucun cas subir de blessure. Des batteries doivent aussi être placées à l'unité de calcul pour parer à toute coupure de courant.

Ces principales caractéristiques ont été quantifiées, et les grandes lignes du cahier des charges sont résumées dans le tableau 1.1. Les quantités précédées de + ou - sont celles qui n'ont pas été chiffrées, mais plutôt où il faut faire du mieux possible.

TAB. 1.1 – Résumé du cahier des charges.

Système	Points importants	Quantité voulue
Général	Taille Masse Prix	hors tout $\leq 3m \times 3m \times 3.5m$ ≤ 500 kg $\leq 100\ 000$ \$ CAN
Mécanique	Sensation de mouvements Architecture Source d'énergie	\geq simulateur commercial + simple - bruyant - entretien
Audio visuel	Graphisme Son	+ immersion + immersion + isolement extérieur
Calculs	modèle dynamique d'avion temps réel synchronisation sécurité	réalisme \geq certification d'entraînement délai commande-mouvement ≤ 150 ms fréquence des calculs ≥ 20 Hz pas de "mal de simulateur" + grande

1.2.3 La conception

À partir de ce cahier des charges, la recherche de Pascal Dufour l'a amené à concevoir un simulateur de vol particulièrement innovant, dont les principales caractéristiques sont détaillées maintenant.

1.2.3.1 Le système audiovisuel

Le système de génération des images est constitué d'un écran sphérique et d'un projecteur achetés à la compagnie Elumens, spécialisée dans la projection sphérique sur des dômes. Il est illustré à la figure 1.2(a). Différents atouts de ce système viennent du fait que la lentille du projecteur est elle-même de forme sphérique. Ainsi, un seul projecteur est utilisé, contrairement à un système classique où plusieurs projecteurs sont nécessaires pour adapter l'image, et ce projecteur est placé directement au centre de la courbure de l'écran, donc très près par rapport à un système classique. Aussi, l'écran est relativement léger (environ 20 kg), et son diamètre d'environ 1,5 m permet au pilote d'être placé au centre de la projection, et d'être très bien immergé dans l'environnement projeté.



(a) Écran sphérique ELUMENS

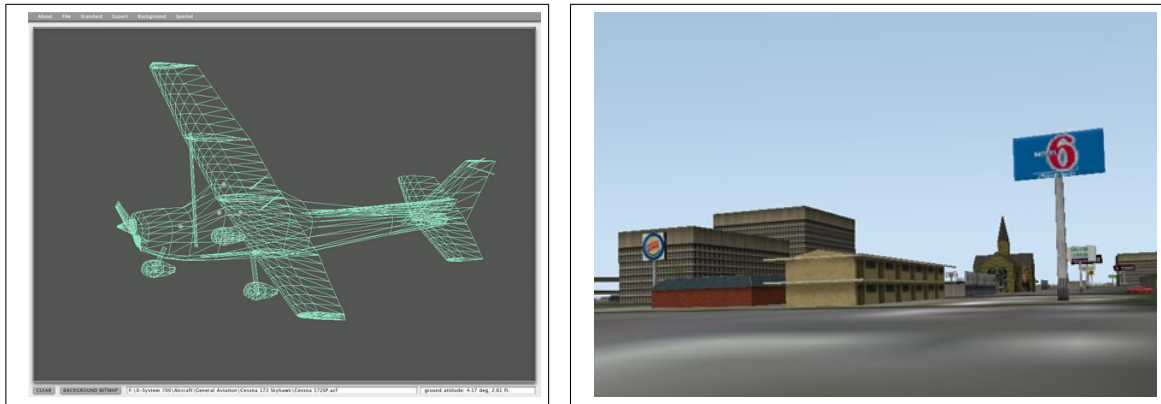
(b) chaise battlechair

FIG. 1.2 – Système audiovisuel.

Le système sonore est constitué de haut-parleurs classiques et d'un haut-parleur de graves, et agit de façon similaire à un cinéma-maison moderne. Dans le système illustré à la figure 1.2(b), ces composants sont directement intégrés à la chaise. Les haut-parleurs sont situés de part et d'autre de la tête du pilote, à la fois pour une bonne immersion, et pour isoler des bruits extérieurs. Le haut parleur de graves, quant à lui, est utilisé aussi bien pour le système sonore que pour le système de mouvement. En effet, son rôle sera à la fois de produire des sons et de faire vibrer le siège pour simuler une translation verticale. Ainsi, lors de mouvements de l'avion nécessitant des translations verticales de petite amplitude du simulateur, comme par exemple lors de turbulences, le haut-parleur de graves fera vibrer le siège.

1.2.3.2 Le système de calcul

Le modèle mathématique de l'avion, ainsi que tous les calculs relatifs aux instruments et à la génération des images et du son, sont gérés par le logiciel X-Plane, commercialisé par la compagnie Laminar Research. C'est un logiciel de simulation de



(a) Modélisation d'un avion

(b) vue extérieure

FIG. 1.3 – Logiciel X-Plane.

vol qui a fait ses preuves dans le domaine car plusieurs simulateurs commerciaux l'utilisent et sont certifiés par la FAA à différents niveaux. En particulier, les accréditations qui concernent le logiciel portent sur le modèle dynamique (pour certains avions), les commandes d'autopilotage, la lecture des instruments et les communications radio.

Dans ce logiciel, le modèle mathématique de l'avion est calculé en divisant l'avion en petits éléments (constituant un maillage de la surface de l'avion), comme illustré à la figure 1.3(a), en calculant les forces agissant sur chacun des éléments et en les intégrant sur la surface de l'appareil. Un gros avantage de ce logiciel est qu'il permet d'envoyer par le réseau les données de vol de l'avion ainsi que toutes les données nécessaires au contrôle de la plate-forme, et ce à une fréquence très élevée (environ 160 Hz), il est donc aisé par la suite de concevoir le contrôleur.

Par ailleurs, le logiciel contient des bases de données topologiques du monde entier, ses graphiques sont très réalistes (voir figure 1.3(b)), et le nombre d'avions modélisés par défaut dans le logiciel est très élevé (24 avions et 3 hélicoptères).

1.2.3.3 Le système mécanique

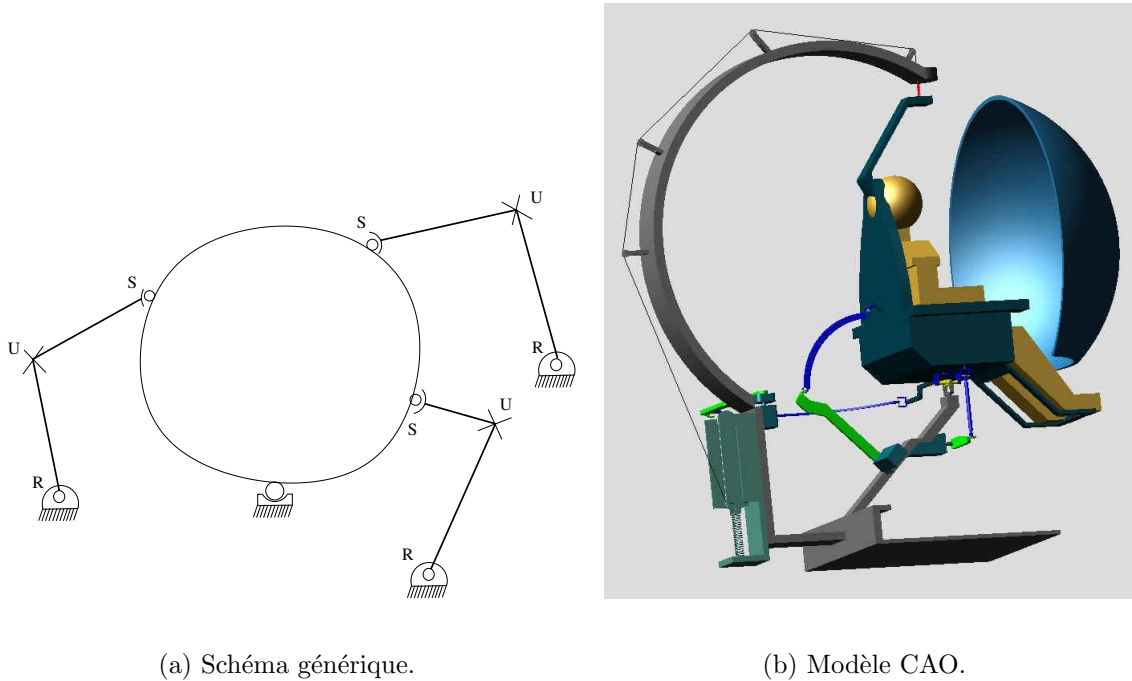
La source d'énergie choisie a été l'électricité, pour plusieurs avantages qu'elle possède sur l'énergie hydraulique utilisée pour les simulateurs classiques. D'abord, l'électricité est moins coûteuse et disponible plus facilement. Aussi, un système d'actionnement



FIG. 1.4 – Moteur choisi.

électrique est moins encombrant et nécessite moins de maintenance qu'un système hydraulique, accompagné de pompes et de réservoir, et qui doit être maintenu souvent à cause des fuites éventuelles. Cependant, la raison majeure pour laquelle les systèmes classiques utilisent une source d'énergie hydraulique est simple : elle permet une très grande puissance des actionneurs. Pour des simulateurs pesant plusieurs tonnes, un système hydraulique est alors plus approprié. Le défi majeur pour utiliser des moteurs électriques dans un simulateur est donc que ceux-ci puissent fournir des couples très élevés. Tenant compte du fait que dans un simulateur les vitesses de fonctionnement sont relativement faibles (quelques dizaines de tours par minute au maximum), ce défi est réalisé en couplant des moteurs électriques à des réducteurs. Le moteur choisi par Pascal Dufour, illustré à la figure 1.4, est un servomoteur AR-60 de la compagnie Nabtesco precision pouvant fournir un couple continu de 470 Nm, et un couple instantané de 1568 Nm, lorsqu'il est couplé à un réducteur de rapport de réduction 100. Lors de la conception finale du simulateur, après l'étude dynamique du mécanisme, il fut décidé de conserver ce choix de moteur, mais avec un réducteur ayant un rapport de réduction encore plus élevé c'est à dire 150. Ainsi l'ensemble moteur-réducteur peut fournir un couple continu de 715 Nm et un couple instantané de 1960 Nm.

Pour ce qui est de l'architecture mécanique, suivant les résultats de Nicolas Pouliot décrits à la section 1.2.1, un mécanisme à trois degrés de liberté de rotation possédant un très grand espace atteignable a été conçu. La figure 1.5 montre de façon schématique l'architecture du simulateur de vol. Il comprend, outre la cabine (qui correspond à la plate-forme mobile), trois chaînes cinématiques composées chacune de deux membrures. La première membrure de chacune de ces chaînes cinématiques, liée à la base par une



(a) Schéma générique.

(b) Modèle CAO.

FIG. 1.5 – Première architecture proposée pour le simulateur (reproduit de [7]).

articulation rotoïde, est la membrure proximale. Elle est reliée à l'autre membrure, la membrure distale, par un joint de cardan à deux degrés de liberté. Finalement la membrure distale est reliée par une articulation sphérique à la plate-forme mobile. En plus de cela, la base est reliée à la plate-forme par une articulation sphérique localisée sous le siège, qui correspond au point fixe du système : c'est le centre de rotation. Les articulations actionnées sont les articulations rotoïdes reliant les pattes à la base, par les moteurs électriques. Le mécanisme est donc parallèle et de type 3 \underline{R} US où R, U et S représentent respectivement des articulations rotoïdes, de cardan et sphériques. La lettre soulignée correspond à l'articulation actionnée. La dernière liaison sphérique sous le siège n'est pas actionnée, et permet de diminuer grandement les efforts dans les pattes actionnées tout en servant de support pour positionner les différents moteurs.

L'espace de travail prescrit du mécanisme est l'espace de travail en rotation dans lequel le mécanisme doit pouvoir bouger et être contrôlable, donc il ne doit contenir aucune singularité. Pour le simulateur de vol, et afin de donner au pilote de bonnes sensations de mouvements, il a été déterminé que le mécanisme devait pouvoir tourner de 60 degrés autour de n'importe quel axe de rotation horizontal, et de plus tourner de ± 60 degrés autour de l'axe vertical résultant de la cabine. Il a donc été naturel pour

décrire cet espace d'utiliser comme variables cartésiennes les angles de *Tilt & Torsion*, car ces critères correspondent à pouvoir effectuer 60 degrés de tilt et ± 60 degrés de torsion. Cette convention d'angles est décrite exhaustivement dans la référence [8] et sera expliquée plus en détails dans le chapitre suivant.

Dans cette application, la charge de la cabine est très grande, et réalise de grands déplacements angulaires, ce qui entraîne de très grands couples statiques aux moteurs. Pour diminuer l'importance relative des couples statiques par rapport aux couples dynamiques, un système d'équilibrage statique à été ajouté au mécanisme. On appelle un mécanisme statiquement équilibré un mécanisme pour lequel l'énergie potentielle est constante quelle que soit la configuration du mécanisme, et par conséquent aucun couple n'est nécessaire aux actionneurs pour maintenir le mécanisme en équilibre statique. Pour ce mécanisme, l'équilibrage statique est réalisé par un ressort, qui permet d'équilibrer l'énergie potentielle de gravité en stockant ou libérant de l'énergie potentielle élastique. On peut remarquer sur la figure 1.5(b) la tige circulaire qui englobe le simulateur. Cette tige sert à supporter un câble sur lequel est attaché le ressort, qui s'allonge en fonction des mouvements de la cabine.

Finalement, la distribution des masses a été estimée, et est résumée au tableau 1.2. Elle est conforme au cahier des charges.

TAB. 1.2 – Estimé de la distribution des masses.

Composante	Masse (kg)
Structure fixe	90
Pilote	100
Siège mobile	40
Visuel	40
Structure mobile (pattes)	80
Moteurs et réducteurs	110
Informatique	40
Total	500

1.3 Conception mécanique : problématique et solutions

Lors de la mise en pratique de cette conception, trois principaux problèmes sont apparus :

1. La dextérité du mécanisme :

lors de l'analyse cinématique du mécanisme, l'évolution de sa dextérité en fonction du tilt pour différentes valeurs de torsion ont été tracées. Celle-ci restait bonne quelque soit le tilt lorsqu'il n'y avait aucune torsion, c'est à dire lorsque le mécanisme était tourné seulement autour d'un axe horizontal. Par contre, les courbes ont montré que la dextérité diminuait rapidement lorsque de la torsion était ajoutée, c'est à dire lorsqu'une rotation autour de la verticale de la cabine était ajoutée au précédent mouvement. À la valeur extrême de torsion de -60 degrés, de nombreuses singularités étaient présentes, et l'espace atteignable pratique à cette valeur de torsion se limitait à 20 degrés de tilt, donc l'espace de travail prescrit ne pouvait pas être respecté.

De plus, aux points où la dextérité était particulièrement mauvaise, les couples statiques aux actionneurs étaient très élevés, et même trop élevés pour être fournis par les moteurs électriques.

2. Les débattements des articulations :

comme Pascal Dufour l'a montré lors de ses recherches, un problème qui restait non résolu était celui des très grands débattements nécessaires aux articulations. En effet, lors des essais de torsion pure, des débattements supérieurs à ± 90 degrés se sont avérés nécessaires à l'une des rotations de l'articulation sphérique de la patte arrière. Ces débattements sont bien au delà de ce qui est offert commercialement, et la conception de telles articulations est très difficile, voire impossible.

3. L'équilibrage statique :

avec les grands débattements désirés, l'équilibrage statique conçu était impossible à utiliser avec le mécanisme car la tige circulaire du système d'équilibrage entraînait en interférence avec l'écran sphérique.

On peut remarquer que ces problèmes sont tous liés aux grands débattements angulaires désirés à la plate-forme, et sont encore plus importants lorsque le mécanisme effectue des rotations combinées (comportant à la fois du tilt et de la torsion). Pour résoudre ces problèmes, la conception mécanique a alors été modifiée pour réaliser un mécanisme appelé hybride. Il est composé d'un premier mécanisme parallèle à deux degrés de liberté ayant deux pattes actionnées, mis en série avec un deuxième mécanisme parallèle à un seul degré de liberté ayant une patte actionnée.

Grâce à cette modification, la dextérité devient indépendante de la rotation du deuxième mécanisme, et reste très bonne sur tout l'espace de travail. De plus, les deux pattes du premier mécanisme n'effectuant pas cette dernière rotation, les débattements aux articulations sont grandement diminués, et sont alors réalisables.

Pour ce qui est de l'équilibrage statique, alors qu'une solution comportant les ressorts sous le siège paraissait impossible avec la première conception, désormais on peut accrocher les ressorts au premier mécanisme seulement (le centre de gravité de la cabine étant situé sur l'axe de la dernière rotation, il reste fixe lorsque le deuxième mécanisme bouge). Alors les ressorts sont situés dans l'espace libre sous le siège, et leurs mouvements sont moins importants car ils ne bougent qu'avec le premier mécanisme, donc les risques d'interférences sont réduits.

C'est cette architecture hybride qui a finalement été retenue, car elle permet de résoudre les problèmes rencontrés lors de la conception. Elle est expliquée plus en détails dans la prochaine section.

1.4 L'architecture mécanique finale

1.4.1 Description de l'architecture

La nouvelle architecture, illustrée à la figure 1.6, est composée de deux mécanismes parallèles montés en série. C'est donc un mécanisme que l'on appelle hybride, car il comprend à la fois des chaînes cinématiques ouvertes et fermées.

Le mécanisme inférieur, noté mécanisme 1, comprend trois chaînes cinématiques différentes. La première, notée patte 1, consiste en une chaîne \underline{RRU} , alors que la se-

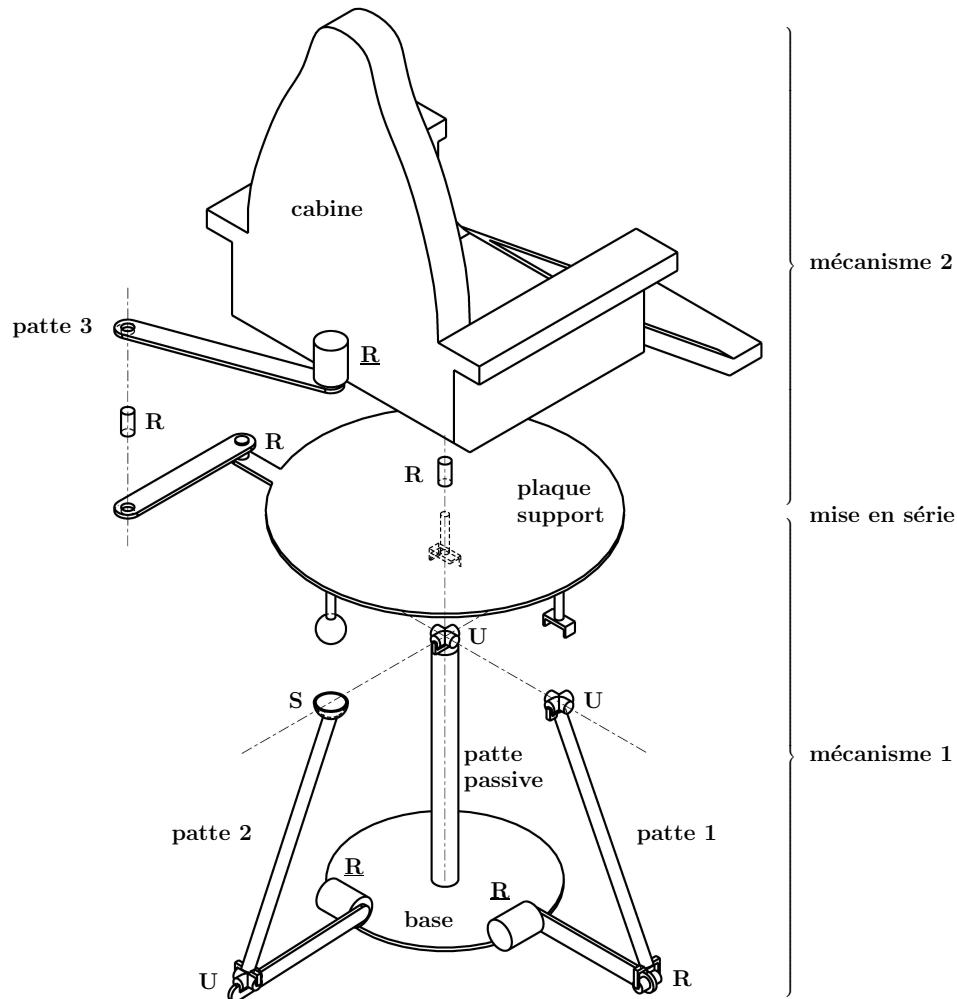


FIG. 1.6 – Schéma éclaté de l'architecture finale du simulateur.

conde, notée patte 2, consiste en une chaîne \underline{RUS} . Dans la configuration de référence illustrée, les pattes 1 et 2 sont situées dans deux plans verticaux orthogonaux.

La troisième chaîne cinématique du mécanisme 1 est appelée la patte passive, car elle n'est pas actionnée, et est équivalente à un joint de cardan. Elle est utilisée pour contraindre les mouvements du mécanisme 1 et pour diminuer les charges dans les autres pattes.

Le mécanisme supérieur, noté mécanisme 2, est un mécanisme à quatre barres plan, pour lequel la base est la plaque support, et l'effecteur est la cabine. Son articulation actionnée est celle reliant la cabine à la membrure distale de la patte 3, il s'agit donc d'une chaîne cinématique de type \underline{RRRR} .

1.4.2 Degrés de liberté

Pour déterminer le nombre de degrés de liberté d'un mécanisme, on utilise généralement la formule de Tchebychev-Grübler-Kutzbach, aussi appelée le critère de mobilité générale :

$$l = d(n - g - 1) + \sum_{i=1}^g f_i \quad (1.1)$$

où l est le degré de liberté du mécanisme, d est la dimension du système de mouvement considéré, n est le nombre de corps rigides dans la chaîne, g est le nombre d'articulations, et f_i est le nombre de degrés de liberté (ddls) permis par la i -ième articulation.

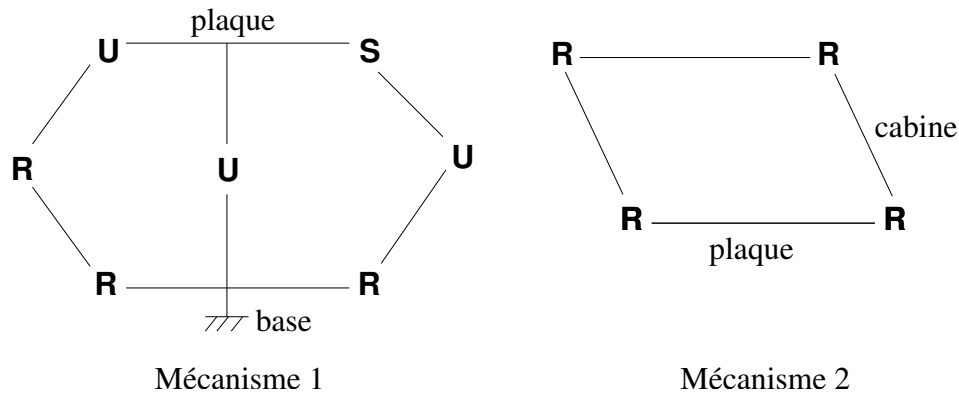


FIG. 1.7 – Schémas cinématiques simplifiés des mécanismes.

En s'appuyant sur le schéma 1.7, on a pour le mécanisme 1 :

$d = 6$ car le mécanisme est spatial,

$n = 6$ corps dans le mécanisme (correspondant au nombre de segments sur le schéma),

$g = 7$ articulations (correspondant au nombre de lettres sur le schéma),

$\sum_{i=1}^g f_i = 12$ ($R \leftrightarrow 1$ ddl, $U \leftrightarrow 2$ ddls, $S \leftrightarrow 3$ ddls).

Donc la formule (1.1) donne $l = 0$, ce qui signifie que le mécanisme ne peut pas bouger ! En réalité, le mécanisme 1 possède deux degrés de liberté en rotation, qui correspondent aux deux degrés de liberté du joint de cardan de la patte passive, et ces degrés de liberté sont obtenus en utilisant des sur-contraintes géométriques. En fait, la formule (1.1) ne dépend que de la topologie du mécanisme et ne tient pas compte de sa géométrie, donc il est possible que certaines conditions géométriques la rendent invalide (Angeles et Gosselin [9] par exemple proposent une méthode alternative).

Pour ce mécanisme en particulier, une conception particulière ajoute les contraintes géométriques au mécanisme : les joints de cardan de la patte passive et de la patte 1 sont agencés de telle façon que leurs axes restent toujours parallèles. De plus, leurs seconds axes respectifs (attachés à la plaque support) sont confondus. Cette géométrie particulière a plusieurs conséquences pour le mécanisme.

D'abord, la mobilité de la plaque support n'est pas restreinte par la patte 1, alors que cette dernière ne possède que quatre degrés de liberté, et permet donc les deux degrés de liberté souhaités.

Ensuite, le centre du joint de cardan de la patte 1 est toujours situé dans le plan défini par le centre du joint de cardan de la patte passive et la membrure proximale de la patte 1. Ainsi, la patte reste dans le même plan quelle que soit la configuration du mécanisme, et dans ce plan, la patte 1 et la plaque support sont équivalents à un mécanisme à quatre barres plan.

Pour le mécanisme 2, aucune contrainte géométrique particulière n'est appliquée, on peut donc utiliser le schéma 1.7 pour déterminer les paramètres :

$d = 3$ car le mécanisme est plan,

$n = 4$ corps dans le mécanisme (correspondant au nombre de segments sur le schéma),

$g = 4$ articulations (correspondant au nombre de lettres sur le schéma),

$\sum_{i=1}^g f_i = 4$ ($R \leftrightarrow 1$ ddl).

La formule (1.1) donne alors $l = 1$. Ce mécanisme à quatre barres a donc bien 1 degré de liberté, qui est la rotation de la cabine autour de l'axe de l'articulation rotoïde la reliant à la plaque support.

Cet axe intersecte le centre du joint de cardan de la patte passive. Ainsi, les degrés de liberté des deux mécanismes sont indépendants, et se combinent pour permettre à la cabine d'effectuer toutes les rotations possibles autour du centre du joint de cardan de la patte passive. Le système global possède donc trois degrés de liberté en rotation.

La contribution originale de la présente maîtrise dans la conception mécanique a été la conception et l'optimisation du nouvel équilibrage statique ainsi que l'analyse cinématique et dynamique du mécanisme hybride. Ces développements seront détaillés dans les chapitres 2 et 3.

Chapitre 2

Analyse cinématique du simulateur

Dans ce chapitre, les notations et conventions cinématiques sont d'abord décrites. Ensuite, le modèle géométrique du mécanisme est détaillé, et le problème géométrique inverse, qui permet de déterminer les variables articulaires en fonction des variables cartésiennes d'orientation de la cabine, est résolu. En dérivant les équations obtenues, la cinématique de la cabine, caractérisée par les matrices jacobiennes du système, est obtenue et la dextérité ainsi que l'espace de travail prescrit sont analysés. Finalement, les variables cinématiques de toutes les membrures du mécanisme, nécessaires au développement du modèle dynamique du prochain chapitre, sont calculées.

2.1 Notations

Avant de commencer le développement du modèle géométrique et de la cinématique du mécanisme, il est important de préciser les notations, ainsi que les conventions choisies pour ce mémoire. D'abord, d'une manière générale, les abréviations ainsi que les indices b, c et ps désignent respectivement la base, la cabine et la plaque support. Par ailleurs, cette section décrit quels sont les repères utilisés, quelles sont les conventions choisies pour les variables d'orientation, ainsi que les variables décrivant la géométrie et la cinématique du mécanisme.

2.1.1 Définition des repères

On définit le point O comme étant l'intersection des axes du joint de cardan passif et de la liaison rotoïde reliant la cabine à la plaque support. Ce point est le centre de rotation de la plate-forme, et est donc le point d'origine naturel pour tous les repères.

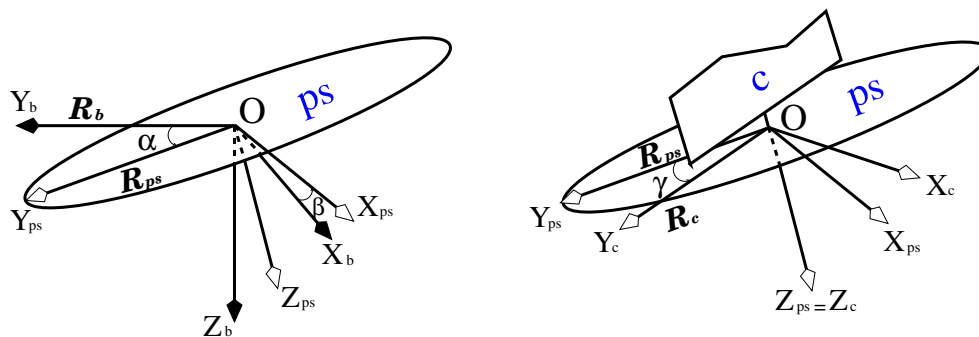


FIG. 2.1 – Orientation des corps.

Le premier repère défini, noté R_c , est lié à la cabine. Par convention, dans la configuration de référence, son axe Z_c pointe verticalement et vers le bas et son axe X_c pointe vers l'avant de la cabine. Le second repère défini, noté R_{ps} , est lié à la plaque support, et le troisième repère, noté R_b est fixe et lié à la base. Ces deux repères sont orientés de telle façon que les trois repères définis coïncident dans la configuration de référence. Les trois repères sont représentés à la figure 2.1.

2.1.2 Coordonnées cartésiennes et matrices de rotation

L'orientation de la cabine est représentée par un triplet d'angles d'Euler, selon la convention appelée XYZ . En utilisant cette convention, une rotation arbitraire de la cabine est composée de trois rotations successives autour des axes X_b , Y_{ps} , et $Z_{ps} \equiv Z_c$, d'angles respectifs α , β , et γ . Ces angles sont aussi connus sous le nom d'angles de Bryant [10], et sont illustrés à la figure 2.1.

Cette convention d'angles d'Euler présente l'avantage de décrire les angles effectivement réalisés par les articulations du mécanisme et donc permet une interprétation physique simple des équations géométriques. En effet, l'axe de rotation fixe du joint de cardan central est l'axe X_b , et son axe de rotation mobile est Y_{ps} . La plaque support ne réalise donc que les deux premières rotations. Alors, les variables cartésiennes et la matrice de rotation de ce corps sont :

$$\boldsymbol{\beta}_{ps} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (2.1)$$

$$\mathbf{Q}_{ps} = \mathbf{Q}_x(\alpha)\mathbf{Q}_y(\beta) \quad (2.2)$$

où \mathbf{Q}_x et \mathbf{Q}_y sont les matrices de rotation élémentaires respectivement autour des axes X et Y .

Finalement, le dernier axe de rotation de la cabine, qui est l'axe de rotation du mécanisme 2, correspond au dernier axe de rotation de la convention d'angles d'Euler choisie, et les variables cartésiennes et la matrice de rotation de la cabine sont :

$$\boldsymbol{\beta} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \quad (2.3)$$

$$\begin{aligned} \mathbf{Q} &= \mathbf{Q}_{ps}(\alpha, \beta)\mathbf{Q}_z(\gamma) \\ &= \begin{bmatrix} c_\beta c_\gamma & -c_\beta s_\gamma & s_\beta \\ s_\alpha s_\beta c_\gamma + c_\alpha s_\gamma & -s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & -s_\alpha c_\beta \\ -c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma & c_\alpha s_\beta s_\gamma + s_\alpha c_\gamma & c_\alpha c_\beta \end{bmatrix} \end{aligned} \quad (2.4)$$

où c_x et s_x représentent le cosinus et le sinus de l'angle x .

2.1.3 Notations pour le mécanisme

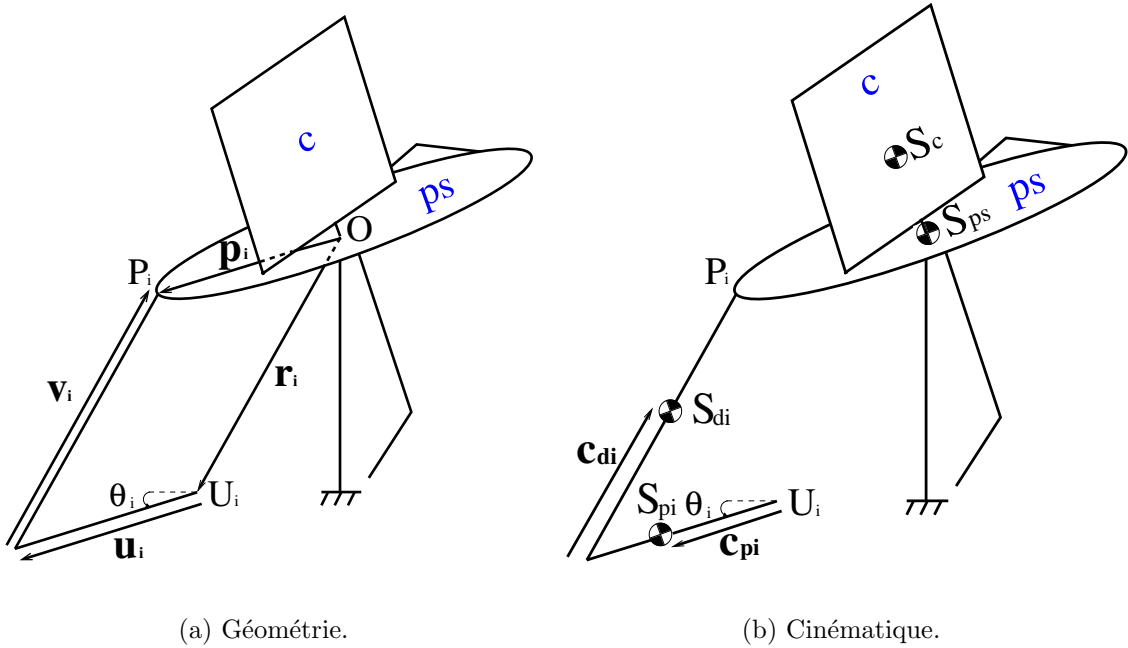


FIG. 2.2 – Notations pour le mécanisme.

La figure 2.2(a) illustre la configuration géométrique générale d'une patte. Pour chaque patte i , θ_i est la coordonnée articulaire, U_i est le point d'attache de la membrure proximale p_i à la base, et P_i est le point d'attache de la membrure distale d_i à la plaque support. Les vecteurs définis le long de la patte sont :

\mathbf{u}_i le long de la membrure p_i , avec $\|\mathbf{u}_i\| = l_{1i}$,

\mathbf{v}_i le long de la membrure d_i , avec $\|\mathbf{v}_i\| = l_{2i}$.

Aussi :

\mathbf{r}_i relie le centre de rotation O à U_i , et on a $\|\mathbf{r}_i\| = \|\mathbf{v}_i\| = l_{2i}$,

\mathbf{p}_i relie le centre de rotation O à P_i , et on a $\|\mathbf{p}_i\| = \|\mathbf{u}_i\| = l_{1i}$.

Dans la configuration de référence, on a : $\mathbf{p}_i = \mathbf{p}_{i0}$ pour $i = 1, 2, 3$, et les pattes 1 et 2 sont situées dans deux plans verticaux nommés respectivement (OY_bZ_b) et (OX_bZ_b) .

La figure 2.2(b) illustre les points et les vecteurs définis pour déterminer la cinématique des corps. Pour chaque corps j , avec $j = c, ps, p_i, d_i, i = 1, 2, 3$, on a :

S_j est le centre de masse de la membrure,

\mathbf{s}_j est le vecteur reliant O à S_j ,

Ainsi, en notant $\boldsymbol{\omega}_j$ le vecteur des vitesses de rotation du corps j , on a le torseur cinématique à six dimensions de ce corps \mathbf{t}_j , défini comme :

$$\mathbf{t}_j = \begin{bmatrix} \boldsymbol{\omega}_j \\ \dot{\mathbf{s}}_j \end{bmatrix} \quad (2.5)$$

Pour chaque membrure k , avec $k = p_i, d_i, i = 1, 2, 3$, on définit aussi à la figure 2.2(b) le vecteur \mathbf{c}_k , qui relie l'extrémité de la membrure k la plus proche de la base à son centre de masse S_k .

Dans la suite de ce mémoire, $\hat{\mathbf{v}}$ représente la matrice antisymétrique associée à tout vecteur tridimensionnel \mathbf{v} définie comme :

$$\hat{\mathbf{v}} \equiv \frac{\partial(\mathbf{v} \times \mathbf{x})}{\partial \mathbf{x}} \quad (2.6)$$

où \mathbf{x} est un vecteur à trois dimensions arbitraire non nul. Cette définition permet de donner un équivalent matriciel au produit vectoriel, car on a :

$$\hat{\mathbf{v}}\mathbf{x} = \mathbf{v} \times \mathbf{x} \quad (2.7)$$

Cette matrice est donc très utile lors de l'étude de rotations ou de la cinématique des corps rigides, où le produit vectoriel apparaît dans de nombreuses équations, car elle permet de faire disparaître cet opérateur non linéaire.

2.2 Modèle géométrique du simulateur

2.2.1 Problème géométrique inverse

Les relations qui permettent de déterminer les coordonnées articulaires en fonction des coordonnées cartésiennes correspondent à la résolution du problème géométrique inverse ou PGI. Grâce à la résolution du PGI, il est donc possible de connaître les positions angulaires des actionneurs pour une trajectoire imposée à l'effecteur. Pour le simulateur, ces relations sont très importantes, car les trajectoires imposées seront toujours des trajectoires d'orientation de la plate-forme (exprimée par les angles d'Euler

α, β, γ), et le PGI nous permet d'exprimer les angles à prescrire aux trois moteurs $(\theta_1, \theta_2, \theta_3)$. Ces relations sont donc à la fois très utiles de manière théorique pour le développement et l'étude de la cinématique du mécanisme, mais aussi en pratique pour le contrôle du simulateur, et seront codées dans le contrôleur.

En utilisant les notations géométriques précédentes, le problème géométrique inverse peut être résolu en écrivant l'équation de fermeture géométrique de chacune des chaînes cinématiques actionnées, comme illustré à la figure 2.3 pour la patte 2.

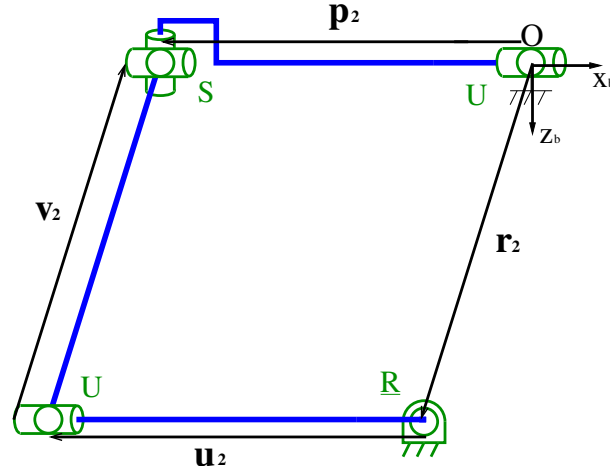


FIG. 2.3 – Modèle géométrique de la patte 2.

On a :

$$\mathbf{v}_2 = \mathbf{p}_2 - \mathbf{r}_2 - \mathbf{u}_2 \quad (2.8)$$

où \mathbf{v}_2 est placé dans le terme de gauche de l'équation car c'est le seul vecteur qui ne peut pas être exprimé simplement en fonction des coordonnées cartésiennes ou articulaires. Mais connaissant sa norme ($\|\mathbf{v}_2\| = l_{22}$), on peut faire disparaître \mathbf{v}_2 de l'équation en écrivant :

$$\begin{aligned} \mathbf{v}_2^T \mathbf{v}_2 = l_{22}^2 &= (\mathbf{p}_2 - \mathbf{r}_2 - \mathbf{u}_2)^T (\mathbf{p}_2 - \mathbf{r}_2 - \mathbf{u}_2) \\ &= (\mathbf{p}_2 - \mathbf{r}_2)^T (\mathbf{p}_2 - \mathbf{r}_2) - 2(\mathbf{p}_2 - \mathbf{r}_2)^T \mathbf{u}_2 + l_{12}^2 \end{aligned} \quad (2.9)$$

En réarrangeant les termes l'équation (2.9), on obtient :

$$2(\mathbf{p}_2 - \mathbf{r}_2)^T \mathbf{u}_2 + l_{22}^2 - l_{12}^2 - (\mathbf{p}_2 - \mathbf{r}_2)^T (\mathbf{p}_2 - \mathbf{r}_2) = 0 \quad (2.10)$$

où \mathbf{r}_2 est connu et fixe par rapport à la base, \mathbf{p}_2 peut être exprimé dans le repère R_b en fonction des coordonnées cartésiennes par :

$$\mathbf{p}_2 = \mathbf{Q}_{ps}\mathbf{p}_{20} \quad (2.11)$$

et le vecteur inconnu \mathbf{u}_2 s'écrit dans R_b comme suit :

$$\mathbf{u}_2 = \begin{bmatrix} -l_{12} \cos \theta_2 \\ 0 \\ l_{12} \sin \theta_2 \end{bmatrix} \quad (2.12)$$

En substituant l'équation (2.12) dans l'équation (2.10) et en réarrangeant les termes, on obtient une équation dont la seule inconnue est θ_2 , qui s'écrit sous la forme :

$$A_2 \cos \theta_2 + B_2 \sin \theta_2 + C_2 = 0, \quad (2.13)$$

avec :

$$\begin{aligned} A_2 &= 2(\mathbf{p}_2 - \mathbf{r}_2)^T \mathbf{e}_{12} l_{12} \\ B_2 &= 2(\mathbf{p}_2 - \mathbf{r}_2)^T \mathbf{e}_{22} l_{12} \\ C_2 &= (\mathbf{p}_2 - \mathbf{r}_2)^T (\mathbf{p}_2 - \mathbf{r}_2) - l_{22}^2 + l_{12}^2 \end{aligned}$$

où :

$$\mathbf{e}_{12} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \text{et} \quad \mathbf{e}_{22} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}.$$

On pose maintenant le changement de variable suivant :

$$t_2 = \tan \left(\frac{\theta_2}{2} \right) \quad (2.14)$$

Alors, en remplaçant $\cos \theta_2 = \frac{1-t_2^2}{1+t_2^2}$ et $\sin \theta_2 = \frac{2t_2}{1+t_2^2}$ dans l'équation (2.13), cette dernière prend la forme d'une équation quadratique en t_2 :

$$(C_2 - A_2)t_2^2 + 2B_2t_2 + (C_2 + A_2) = 0 \quad (2.15)$$

Cette équation conduit dans le cas général à deux solutions distinctes pour t_2 , et donc, en remplaçant ces solutions dans l'équation (2.14), deux solutions pour le problème géométrique de la patte 2 :

$$\theta_2 = 2 \arctan \left(\frac{B_2^2 \pm \sqrt{B_2^2 + A_2^2 - C_2^2}}{C_2 - A_2} \right) \quad (2.16)$$

Pour les pattes 1 et 3, les mêmes développements peuvent être suivis. Cependant, les coefficients sont beaucoup plus simples car ces pattes sont planes et la fermeture géométrique de leur chaîne cinématique est celle d'un mécanisme à quatre barres dans ce plan. La figure 2.4 illustre les deux solutions pour la patte 1.

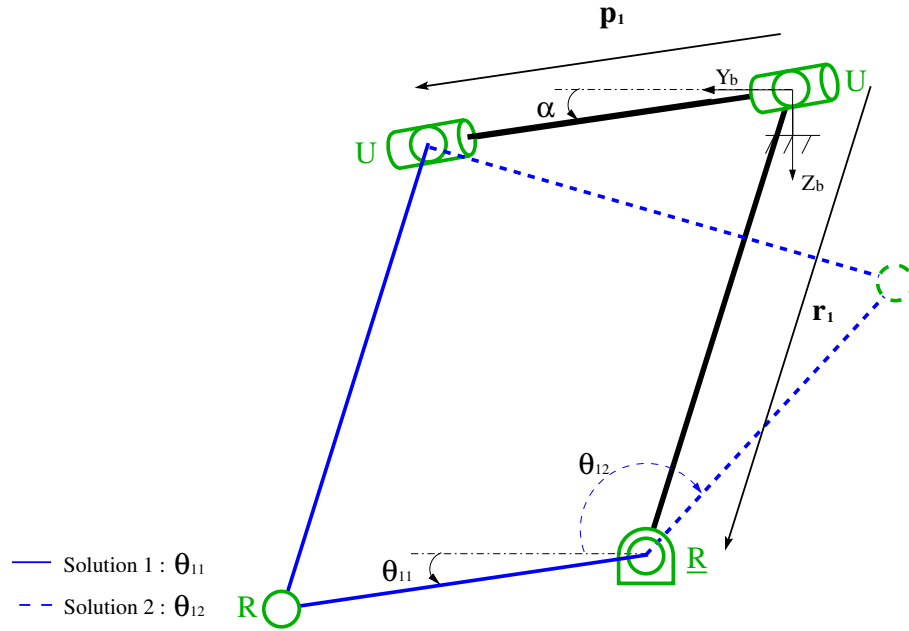


FIG. 2.4 – Les deux solutions du PGI pour la patte 1.

On voit bien que, pour \mathbf{r}_1 et \mathbf{p}_1 fixés, il existe géométriquement deux solutions, notées θ_{11} et θ_{12} sur la figure. De plus, comme les membrures opposées de cette chaîne cinématique sont de même longueur, une des deux solutions (θ_{11} sur la figure) est la solution du PGI pour laquelle la fermeture géométrique est un parallélogramme dans le plan de la patte, et c'est une solution triviale pour laquelle on a :

$$\theta_1 = \alpha \quad (2.17)$$

Et de la même façon pour la patte 3, la solution triviale est :

$$\theta_3 = \gamma \quad (2.18)$$

Finalement, le PGI possède en général huit solutions (deux solutions pour chaque coordonnée articulaire). Les solutions triviales présentées pour θ_1 et θ_3 étant celles utilisées en pratique pour le mécanisme, ce seront donc les solutions considérées dans la suite de ce mémoire.

2.2.2 Problème géométrique direct

Les relations qui permettent de déterminer les coordonnées cartésiennes en fonction des coordonnées articulaires correspondent à la résolution du problème géométrique direct ou PGD. Contrairement au cas des mécanismes sériels pour lesquels la résolution du PGD est directe, ces relations sont habituellement complexes pour les mécanismes parallèles car elles nécessitent la résolution d'équations algébriques couplées, ce qui entraîne un nombre élevé de solutions en général.

Dans le cas du simulateur sphérique hybride étudié ici, encore une fois grâce à l'architecture du mécanisme, ces équations sont presque complètement découplées, et nécessitent la résolution pour chaque angle d'Euler d'une équation quadratique similaire à l'équation (2.15). Les détails de la résolution de ces équations sont donnés en annexe A.1. On se ramène donc au même degré de complexité que le PGI, et le nombre maximum de solutions est huit. Encore une fois, une seule de ces solutions est utilisée en pratique lors des mouvements du mécanisme, et correspond à ce que l'on appelle le mode d'assemblage de ce mécanisme.

2.3 Équations de vitesse et jacobienne

Dans cette section, les équations de vitesse seront développées, ainsi que les matrices reliant les vitesses angulaires de la plate-forme aux vitesses articulaires, appelées matrices jacobienne du système. Ces différents développements seront d'abord réalisés

pour le mécanisme 1, puis pour le mécanisme 2, et enfin les résultats seront regroupés pour former la matrice jacobienne globale du simulateur.

2.3.1 Mécanisme 1

Pour déterminer les équations de vitesse du premier mécanisme, il suffit de dériver par rapport au temps les équations du problème géométrique inverse développées à la section 2.2.1.

Pour la patte 1, la dérivation de l'équation (2.17) donne de façon triviale :

$$\dot{\theta}_1 = \dot{\alpha} \quad (2.19)$$

Pour la patte 2, on dérive par rapport au temps l'équation (2.10) :

$$(\mathbf{p}_2 - \mathbf{r}_2 - \mathbf{u}_2)^T \dot{\mathbf{p}}_2 - (\mathbf{p}_2 - \mathbf{r}_2)^T \dot{\mathbf{u}}_2 = 0 \quad (2.20)$$

où $\dot{\mathbf{u}}_2$ est exprimé en dérivant par rapport au temps l'équation (2.12) :

$$\dot{\mathbf{u}}_2 = \dot{\theta}_2 \begin{bmatrix} l_{12} \sin \theta_2 \\ 0 \\ l_{12} \cos \theta_2 \end{bmatrix} = \dot{\theta}_2 \mathbf{E} \mathbf{u}_2 \quad \text{avec} \quad \mathbf{E} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad (2.21)$$

et, de manière similaire, $\dot{\mathbf{p}}_2$ est exprimé en dérivant par rapport au temps l'équation (2.11) :

$$\dot{\mathbf{p}}_2 = \dot{\mathbf{Q}}_{ps} \mathbf{p}_{20} = \boldsymbol{\omega}_{ps} \times \mathbf{p}_2 \quad (2.22)$$

où $\boldsymbol{\omega}_{ps}$ est le vecteur des vitesses angulaires de la plaque support.

En substituant les équations (2.21) et (2.22) dans l'équation (2.20), on a :

$$(\mathbf{p}_2 - \mathbf{r}_2 - \mathbf{u}_2)^T (\boldsymbol{\omega}_{ps} \times \mathbf{p}_2) = \dot{\theta}_2 (\mathbf{p}_2 - \mathbf{r}_2)^T \mathbf{E} \mathbf{u}_2 \quad (2.23)$$

Or les composants d'un produit mixte (qui est le produit scalaire dont un des vecteur est un produit vectoriel, comme dans le premier terme de l'équation précédente) peuvent

être inter-changés circulairement sans modifier le résultat. L'équation (2.23) est donc équivalente à :

$$(\mathbf{p}_2 \times (\mathbf{p}_2 - \mathbf{r}_2 - \mathbf{u}_2))^T \boldsymbol{\omega}_{ps} = (\mathbf{p}_2 - \mathbf{r}_2)^T \mathbf{E} \mathbf{u}_2 \dot{\theta}_2 \quad (2.24)$$

Il reste désormais à déterminer la relation entre le vecteur des vitesses angulaires de la plaque support $\boldsymbol{\omega}_{ps}$ et le vecteur des dérivées des angles d'Euler $\dot{\boldsymbol{\beta}}_{ps}$. Or par définition des vitesses de rotation on a :

$$\hat{\boldsymbol{\omega}}_{ps} = \dot{\mathbf{Q}}_{ps} \mathbf{Q}_{ps}^T \quad (2.25)$$

Connaissant l'expression de la matrice de rotation en fonction des angles d'Euler d'après l'équation (2.2), et donc de sa dérivée en fonction des dérivées des angles d'Euler, on peut exprimer le terme de gauche en fonction de $\boldsymbol{\beta}_{ps}$ et $\dot{\boldsymbol{\beta}}_{ps}$. Il suffit ensuite d'identifier chacun des termes de l'équation (2.25), et pour la convention des angles d'Euler XYZ choisie précédemment, on trouve la relation :

$$\boldsymbol{\omega}_{ps} = \mathbf{R}_{ps} \dot{\boldsymbol{\beta}}_{ps}, \quad \text{avec} \quad \mathbf{R}_{ps} = \begin{bmatrix} 1 & 0 \\ 0 & c_\alpha \\ 0 & s_\alpha \end{bmatrix} \quad (2.26)$$

Ainsi, l'équation de vitesse de la patte 2 s'écrit, en fonction de $\dot{\boldsymbol{\beta}}_{ps}$:

$$((\mathbf{r}_2 + \mathbf{u}_2) \times \mathbf{p}_2)^T \mathbf{R}_{ps} \dot{\boldsymbol{\beta}}_{ps} = (\mathbf{p}_2 - \mathbf{r}_2)^T \mathbf{E} \mathbf{u}_2 \dot{\theta}_2 \quad (2.27)$$

Finalement, en réunissant sous forme matricielle les équations (2.19) et (2.27), les équations de vitesses du mécanisme 1 s'écrivent :

$$\mathbf{A}_{ps} \dot{\boldsymbol{\beta}}_{ps} = \mathbf{B}_{ps} \dot{\boldsymbol{\theta}}_{ps} \quad (2.28)$$

où

$$\dot{\boldsymbol{\theta}}_{ps} = \begin{bmatrix} \dot{\theta}_1 & \dot{\theta}_2 \end{bmatrix}^T, \quad (2.29)$$

$$\mathbf{A}_{ps} = \begin{bmatrix} 1 & 0 \\ ((\mathbf{r}_2 + \mathbf{u}_2) \times \mathbf{p}_2)^T \mathbf{R}_{ps} \end{bmatrix}, \quad (2.30)$$

$$\mathbf{B}_{ps} = \begin{bmatrix} 1 & 0 \\ 0 & (\mathbf{p}_2 - \mathbf{r}_2)^T \mathbf{E} \mathbf{u}_2 \end{bmatrix}, \quad (2.31)$$

et la relation entre les vitesses angulaires de la plaque support et les vitesses articulaires est :

$$\boldsymbol{\omega}_{ps} = \mathbf{J}_{ps} \dot{\boldsymbol{\theta}}_{ps}, \quad \mathbf{J}_{ps} = \mathbf{R}_{ps} \mathbf{A}_{ps}^{-1} \mathbf{B}_{ps} \quad (2.32)$$

où \mathbf{J}_{ps} représente la matrice jacobienne du mécanisme 1. Elle est de dimension 3×2 car elle relie $\boldsymbol{\omega}_{ps}$, le vecteur des vitesses angulaires de la plaque support de dimension 3×1 , à $\dot{\boldsymbol{\theta}}_{ps}$, qui est le vecteur des vitesses articulaires de dimension 2×1 car le mécanisme 1 ne comporte que deux actionneurs.

2.3.2 Mécanisme 2

En dérivant par rapport au temps l'équation (2.18), on obtient directement la dernière équation de vitesse en fonction des dérivées des angles d'Euler :

$$\dot{\theta}_3 = \dot{\gamma} \quad (2.33)$$

Pour déterminer cette relation en fonction des vitesses angulaires de la cabine, on définit tout d'abord $\boldsymbol{\omega}_{c/ps}$ le vecteur des vitesses angulaires de la cabine par rapport à la plaque support. Exprimé dans le repère R_{ps} ou R_c , il s'écrit :

$$\left[\boldsymbol{\omega}_{c/ps} \right]_{R_{ps}} = \dot{\theta}_3 \mathbf{Z}_{ps} \quad (2.34)$$

et donc son expression dans le repère de base R_b est :

$$\begin{aligned} \boldsymbol{\omega}_{c/ps} &= \mathbf{Q}_{ps} \left[\boldsymbol{\omega}_{c/ps} \right]_{R_{ps}} = \begin{bmatrix} s_\beta \\ -s_\alpha c_\beta \\ c_\alpha c_\beta \end{bmatrix} \dot{\theta}_3 \\ &= \mathbf{J}_{c/ps} \dot{\theta}_3 \end{aligned} \quad (2.35)$$

où $\mathbf{J}_{c/ps}$ représente la matrice jacobienne du mécanisme 2. Elle est de dimension 3×1 car elle relie $\boldsymbol{\omega}_{c/ps}$, le vecteur des vitesses angulaires de la cabine par rapport à la plaque support de dimension 3×1 , à $\dot{\theta}_3$, qui est l'unique vitesse articulaire du mécanisme 2, car ce dernier ne comporte qu'un seul actionneur.

2.3.3 Relations globales

Le vecteur des vitesses angulaires totales de la cabine par rapport à la base est :

$$\boldsymbol{\omega} = \boldsymbol{\omega}_{ps} + \boldsymbol{\omega}_{c/ps} \quad (2.36)$$

Ainsi, en utilisant les relations partielles (2.32) et (2.35), on obtient la relation globale des vitesses du simulateur :

$$\boldsymbol{\omega} = \mathbf{J}\dot{\boldsymbol{\theta}} \quad (2.37)$$

avec

$$\mathbf{J} = \left[\mathbf{J}_{ps}, \mathbf{J}_{c/ps} \right], \quad \dot{\boldsymbol{\theta}} = \left[\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3 \right]^T \quad (2.38)$$

où \mathbf{J} représente la matrice jacobienne globale du simulateur, de dimension 3×3 .

2.4 Analyse cinématique globale

2.4.1 Espace atteignable prescrit

On définit un espace atteignable prescrit comme l'espace cartésien (qui est un espace en orientation pour le mécanisme considéré) dans lequel le mécanisme doit pouvoir être exploité, c'est à dire qu'il correspond à l'espace de travail pratique du mécanisme. Dans cet espace, il ne doit donc y avoir aucune interférence mécanique, aucune singularité, et le mécanisme doit rester contrôlable. Cette dernière condition est en théorie la même que l'évitement des singularités, mais en pratique elle est plus contraignante car il est nécessaire que le mécanisme soit assez éloigné d'une singularité et possède une dextérité suffisante pour être contrôlable.

On appelle w l'espace atteignable en rotation prescrit pour l'application considérée. Il a été choisit, en tenant compte de la conception initiale décrite au chapitre précédent, d'imposer au mécanisme de pouvoir tourner jusqu'à $\pm 50^\circ$ autour de tout axe horizontal et de plus $\pm 50^\circ$ autour de l'axe Z résultant. Cette limitation par rapport à la valeur

initiale (qui visait un espace de $\pm 60^\circ$ pour ces mêmes axes) est due aux problèmes rencontrés lors de la conception, et décrits au chapitre précédent. Les $\pm 50^\circ$ prescrits pour l'espace w correspondent donc à un compromis fait pour la faisabilité du système, mais w reste particulièrement étendu, et bien au delà des espaces de travail des simulateurs commerciaux actuels.

L'utilisation des angles de *Tilt & Torsion* [8] est particulièrement appropriée pour décrire cet espace atteignable prescrit w , car les limites imposées correspondent à $\pm 50^\circ$ d'angle de tilt θ (autour d'un axe horizontal orienté par l'angle ϕ), et $\pm 50^\circ$ d'angle de torsion σ . Ces angles sont décrits à la figure 2.5.

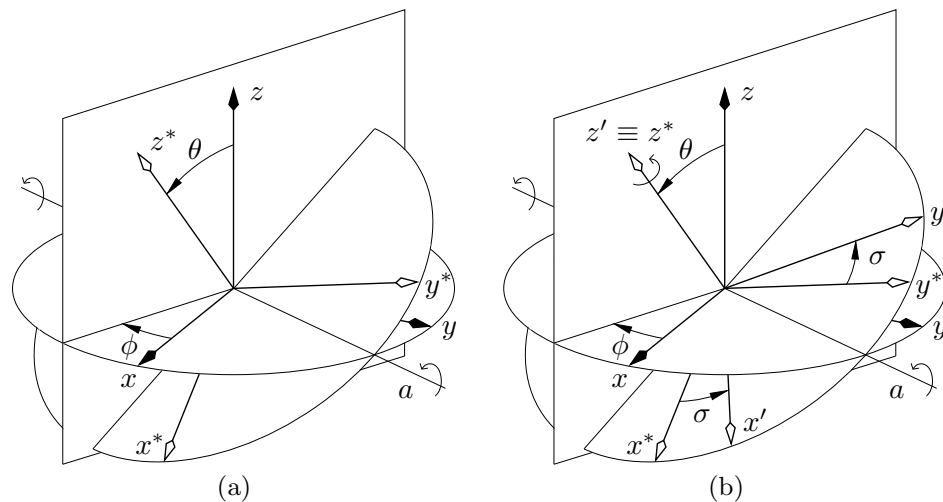


FIG. 2.5 – Rotations du *Tilt & Torsion* : (a) tilt, (b) torsion(reproduit de [8]).

Cette convention de représentation de l'orientation appelée *Tilt & Torsion* correspond à ce que l'on appelle une convention d'angles d'Euler modifiée. En effet, contrairement à une convention d'angles d'Euler classique, seulement deux étapes principales de rotation sont nécessaires pour décrire l'orientation quelconque d'un corps.

La première étape, illustrée à la figure 2.5(a), consiste à amener l'axe z du corps dans son orientation finale, c'est-à-dire z^* . La rotation nécessaire à cette étape a pour angle θ , qui est l'angle de *Tilt*. La principale différence entre cette convention et une convention d'angles d'Euler classique est l'axe autour duquel le *Tilt* est réalisé. En effet, pour une convention d'angles d'Euler classique, les axes de rotation sont toujours des axes appartenant soit au corps soit au repère inertiel. Or ici, afin de placer l'axe z dans son orientation finale, la rotation de l'angle θ est réalisée autour d'un axe a , qui est un axe horizontal n'appartenant ni au repère inertiel, ni au repère du corps. Il est orienté par l'angle ϕ dans le plan horizontal. Donc la première étape consiste en une seule rota-

tion, mais cette rotation doit être définie par deux angles : ϕ qui oriente l'axe de cette rotation, et θ qui est l'angle de cette rotation.

La seconde étape consiste à effectuer une rotation d'un angle σ autour du nouvel axe z' , comme illustré à la figure 2.5(b). Cette deuxième étape s'appelle la *Torsion*, et permet de placer les deux autres axes du corps x' et y' dans leur orientation finale.

Comme on l'a vu, les limites de l'espace atteignable prescrit w en fonction des angles de *Tilt & Torsion* sont des constantes. Cette convention permet donc de faciliter grandement la visualisation de w , ainsi que tous les calculs qui s'y rattachent. En particulier, les calculs de dextérité qui suivent sont très simplifiés. De plus, d'un point de vue pratique, les limites de l'espace w doivent être calculées par le contrôleur, et être comparées aux consignes de position cartésiennes afin de s'assurer que ces dernières restent toujours à l'intérieur de w . Il est donc clair que réaliser une telle comparaison avec une constante facilite grandement le test de contrôle.

Ainsi, les calculs cinématiques sont réalisés en utilisant la convention d'angles d'Euler XYZ, puis les calculs relatifs à l'espace de travail sont réalisés en utilisant la convention de *Tilt & Torsion*. La conversion entre ces deux représentations est simple, et détaillée en annexe A.2.

2.4.2 Dextérité

Une mesure de la précision cinématique d'un manipulateur est sa dextérité. La dextérité locale d'un mécanisme ξ est définie comme l'inverse du conditionnement κ de sa matrice jacobienne. Comme la jacobienne caractérise la transformation linéaire entre vitesses cartésiennes et articulaires du mécanisme, la dextérité correspond donc à la qualité de cette transformation. Elle est exprimée comme suit :

$$\xi = 1/\kappa(\mathbf{J}), \quad \text{avec} \quad \kappa(\mathbf{J}) = \|\mathbf{J}\| \|\mathbf{J}^{-1}\| \quad (2.39)$$

où $\|\mathbf{J}\|$ est la norme Euclidienne de la matrice jacobienne.

D'après cette définition, on a $0 < \xi \leq 1$, où une dextérité locale nulle correspond à une orientation singulière, et une dextérité locale égale à un correspond à une orientation pour laquelle la matrice jacobienne est parfaitement conditionnée.

Pour le mécanisme étudié ici, la matrice jacobienne \mathbf{J} est indépendante du troisième angle d'Euler γ d'après les équations (2.32), (2.35), et (2.38), et ne dépend donc que des angles α et β lorsqu'elle est exprimée en fonction des angles d'Euler XYZ. Par ailleurs, d'après les équations de conversion données en annexe A.2.2, ces angles α et β ne dépendent pas de la torsion σ . Donc la matrice jacobienne \mathbf{J} , et par conséquent la dextérité locale, sont aussi indépendantes de l'angle de torsion σ lorsque l'orientation est représentée par les angles de *Tilt & Torsion*. La dextérité locale peut donc être tracée comme une fonction de ϕ et θ à la figure 2.6.

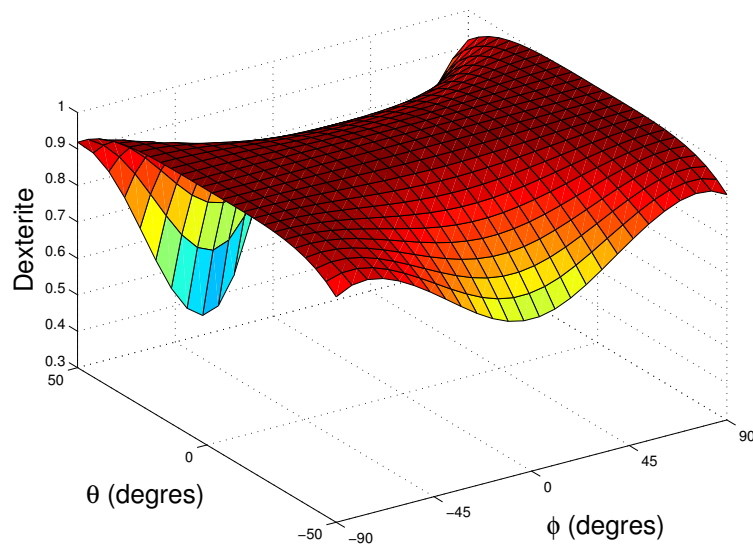


FIG. 2.6 – Dexterité locale du mécanisme $\xi(\phi, \theta)$

Cette figure montre bien que le comportement cinématique de ce mécanisme est très bon. De surcroît, les configurations pour lesquelles la dextérité est plus faible sont situées aux frontières de w . Le comportement du mécanisme autour de la configuration de référence, dans les configurations où le mécanisme se trouve le plus souvent, est donc excellent. La dextérité est même égale à un (et est donc idéale) dans la configuration de référence. Cette configuration est alors appelée une configuration isotrope.

En intégrant la dextérité locale sur w , on obtient un index de performance cinématique global, appelé la dextérité globale [11] :

$$\eta = \frac{\int_w \xi dw}{\int_w dw}, \quad 0 < \eta \leq 1 \quad (2.40)$$

Pour le mécanisme du simulateur, on trouve $\eta = 0.932$.

Finalement, ce mécanisme présente simultanément une excellente dextérité globale et un large espace atteignable pratique exempt de singularité. Cette analyse cinématique montre bien l'avantage de l'architecture mécanique hybride par rapport à la première conception où l'architecture était pleinement parallèle : alors que dans le dernier cas la dextérité était fortement dépendante de la torsion, et diminuait rapidement lorsque la torsion augmentait jusqu'à atteindre les singularités, dans le cas présent la dextérité locale est complètement indépendante de la torsion, donc la dextérité globale est très élevée et aucune singularité n'est présente dans l'espace de travail prescrit.

L'espace w prescrit est donc du point de vue cinématique aisément atteignable. De plus, grâce à la conception mécanique et aux études d'interférences mécaniques réalisées par Simon Foucault, aucune interférence mécanique n'est présente sur w , et toutes les articulations ont des débattements suffisants pour permettre au système d'adopter toutes les configurations de l'espace w .

2.5 Cinématique des membrures

Grâce à la cinématique globale du simulateur développée aux sections précédentes, on a pu déterminer les variables cinématiques de la plaque support ainsi que de la cabine en fonction des variables articulaires et de leur dérivées, et analyser l'espace de travail du simulateur. Pour pouvoir calculer le modèle dynamique complet dans le prochain chapitre, on détermine aussi dans cette section les variables cinématiques de toutes les membrures en fonction des variables articulaires et de leurs dérivées.

2.5.1 Membrures du mécanisme 1

Les membrures proximales p_1 et p_2 du mécanisme 1, illustrées à la figure 2.7, effectuent une rotation pure autour de l'axe fixe du moteur qui leur est associé, perpendiculaire au plan de la patte dans la configuration de référence. Leur vitesse angulaire autour de cet axe ω_{p_i} est donc la seule variable cinématique nécessaire au calcul du modèle dynamique, et on a simplement :

$$\omega_{p1} = \dot{\theta}_1 \mathbf{X}_b \quad (2.41)$$

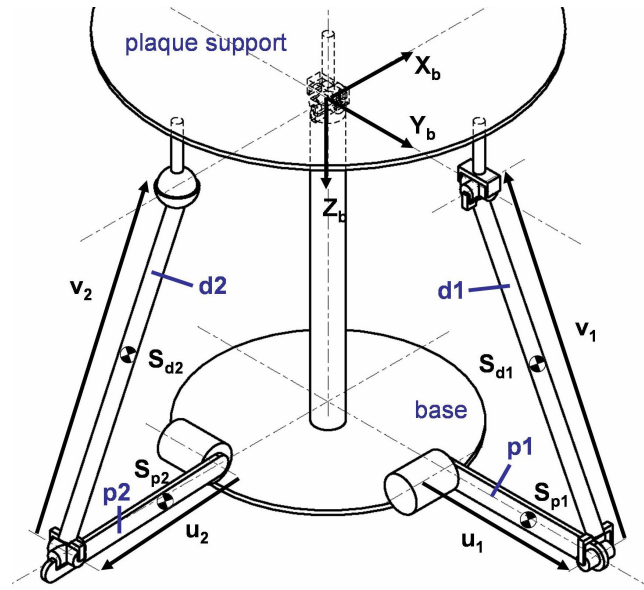


FIG. 2.7 – Schéma cinématique du mécanisme 1.

$$\omega_{p2} = \dot{\theta}_2 \mathbf{Y}_b \quad (2.42)$$

Dû au fait que la patte 1 est plane et forme un parallélogramme, la membrure d_1 ne subit quant à elle que des mouvements de translations circulaires. En effet, le vecteur \mathbf{v}_1 , qui est situé le long de d_1 (cf. figure 2.7), doit toujours rester parallèle à \mathbf{r}_1 qui forme le côté opposé du parallélogramme, alors que \mathbf{r}_1 est fixe. Le seul mouvement possible de cette membrure est donc une translation circulaire dans le plan de la patte. Alors, la vitesse de rotation de d_1 est nulle, et tous les points de la membrure ont la même vitesse, égale à la vitesse du point de connexion avec p_1 , qui est $\dot{\mathbf{u}}_1$. Le torseur cinématique de d_1 est donc simplement :

$$\mathbf{t}_{d1} = \begin{bmatrix} \mathbf{0} \\ \dot{\mathbf{u}}_1 \end{bmatrix} \quad (2.43)$$

où $\mathbf{0}$ représente le vecteur nul à trois dimensions, c'est à dire $\mathbf{0}_{3 \times 1}$.

Le mouvement de la membrure d_2 est constitué à la fois de translations et rotations, car aucune contrainte géométrique particulière ne limite ses degrés de liberté. De plus, comme elle n'est pas reliée directement aux actionneurs, ses mouvements ne s'expriment pas de manière simple en fonction des variables articulaires. Le calcul du

torseur cinématique de d_2 nécessite donc de déterminer la relation entre les variables cinématiques de d_2 et les coordonnées généralisées et leurs dérivées.

Ce calcul est réalisé en considérant la patte 2 comme un manipulateur sériel à trois degrés de liberté, utilisé pour positionner le point P_2 dans l'espace, et pour lequel d_2 est donc l'effecteur. Grâce à cet artifice de calcul, le PGI de ce manipulateur peut être résolu en utilisant le formalisme de Hartenberg-Denavit [12], qui est le formalisme utilisé classiquement pour les calculs cinématiques de manipulateurs sériels. Les variables cinématiques de d_2 sont alors exprimées en fonction de $\dot{\boldsymbol{\theta}}_2$, qui est le vecteur des vitesses aux articulations (actionnées ou non) de la patte 2. Les détails du formalisme de Hartenberg-Denavit ainsi que les calculs associés sont donnés en Annexe A.3. La relation que l'on obtient est :

$$\begin{aligned}\boldsymbol{\omega}_{d_2} &= \mathbf{J}_{d_2} \dot{\boldsymbol{\theta}}_2 \\ \dot{\mathbf{s}}_{d_2} &= \mathbf{K}_{d_2} \dot{\boldsymbol{\theta}}_2\end{aligned}\quad (2.44)$$

avec

$$\mathbf{J}_{d_2} = [\mathbf{e}_{12}, \mathbf{e}_{22}, \mathbf{e}_{32}] \quad (2.45)$$

$$\mathbf{K}_{d_2} = [\mathbf{e}_{12} \times (\mathbf{c}_{d_2} + \mathbf{u}_2), \mathbf{e}_{22} \times \mathbf{c}_{d_2}, \mathbf{e}_{32} \times \mathbf{c}_{d_2}] \quad (2.46)$$

où \mathbf{e}_{i2} , $i=1,2,3$ sont des vecteurs unitaires situés le long des articulations de la patte 2.

De plus, la matrice jacobienne de la patte 2, \mathbf{J}_2 , qui est la matrice reliant les vitesses de son effecteur, c'est à dire $\dot{\mathbf{p}}_2$, aux vitesses articulaires de la patte, c'est à dire $\dot{\boldsymbol{\theta}}_2$, est aussi calculée par cette méthode, et son calcul est détaillé en annexe A.3. Cette matrice est appelée une jacobienne passive, car elle n'implique pas seulement les vitesses des articulations actionnées, mais toutes les vitesses articulaires de la patte. L'utilité de la matrice \mathbf{J}_2 dans le présent développement est qu'elle peut être comparée à la matrice jacobienne du mécanisme 1 \mathbf{J}_{ps} afin de déterminer la relation entre $\dot{\boldsymbol{\theta}}_2$ et $\dot{\boldsymbol{\theta}}_{ps}$:

$$\dot{\boldsymbol{\theta}}_2 = \mathbf{L}_2 \dot{\boldsymbol{\theta}}_{ps}, \quad \mathbf{L}_2 = -\mathbf{J}_2^{-1} \hat{\mathbf{p}}_2 \mathbf{J}_{ps} \quad (2.47)$$

Finalement, en combinant les relations (2.44) et (2.47), on obtient l'expression du torseur cinématique de la membrure d_2 en fonction des vitesses articulaires :

$$\mathbf{t}_{d2} = \mathbf{N}_{d2} \dot{\boldsymbol{\theta}}_{ps}, \quad \mathbf{N}_{d2} = \begin{bmatrix} \mathbf{A}_{d2} \\ \mathbf{B}_{d2} \end{bmatrix} \quad (2.48)$$

avec

$$\mathbf{A}_{d2} = \mathbf{J}_{d2} \mathbf{L}_2, \quad (2.49)$$

$$\mathbf{B}_{d2} = \mathbf{K}_{d2} \mathbf{L}_2 \quad (2.50)$$

2.5.2 Membrures du mécanisme 2

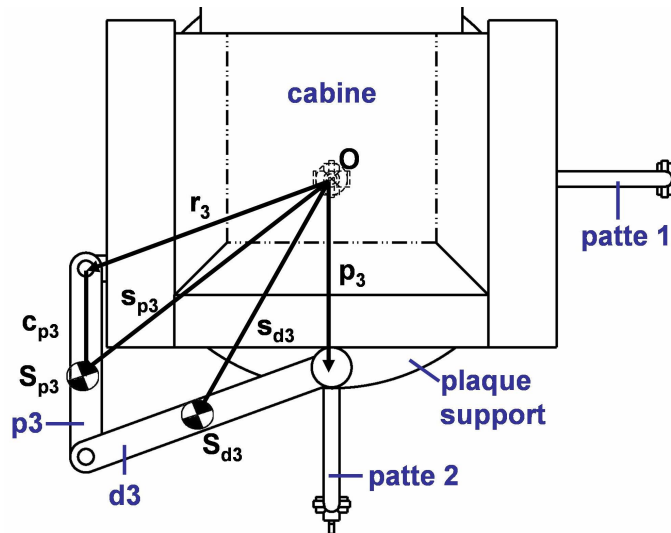


FIG. 2.8 – Schéma cinématique du mécanisme 2 (vue de haut).

Ces dérivations réalisées pour la patte 2 ne sont pas nécessaires pour la patte 3, encore une fois pour la raison que la patte est plane et que la chaîne cinématique qu'elle forme avec la cabine et la plaque support est un parallélogramme. En conséquence, le torseur cinématique de chacune des membrures de la patte 3 peut être exprimé simplement en fonction des coordonnées articulaires.

Pour la membrure proximale p_3 , on a :

$$\boldsymbol{\omega}_{p3} = \boldsymbol{\omega} = \mathbf{J} \dot{\boldsymbol{\theta}} \quad (2.51)$$

et en décomposant la vitesse de S_{p3} en vitesse relative et vitesse d'entraînement on a, en se référant à la figure 2.8 :

$$\begin{aligned}
\dot{\mathbf{s}}_{p3} &= \mathbf{v}(S_{p3} \in p3/b) = \mathbf{v}(S_{p3} \in p3/ps) + \mathbf{v}(S_{p3} \in ps/b) \\
&= \boldsymbol{\omega}_{c/ps} \times \mathbf{c}_{p3} + \boldsymbol{\omega}_{ps} \times \mathbf{s}_{p3}
\end{aligned} \tag{2.52}$$

Donc la relation entre la vitesse de S_{p3} et les vitesses articulaires s'écrit sous forme matricielle, en utilisant la notation (2.6) :

$$\dot{\mathbf{s}}_{p3} = \mathbf{B}_{p3} \dot{\boldsymbol{\theta}}, \quad \text{avec} \quad \mathbf{B}_{p3} = \begin{bmatrix} -\hat{\mathbf{s}}_{p3} \mathbf{J}_{ps}, & -\hat{\mathbf{c}}_{p3} \mathbf{J}_{c/ps} \end{bmatrix} \tag{2.53}$$

et ainsi :

$$\mathbf{t}_{p3} = \mathbf{N}_{p3} \dot{\boldsymbol{\theta}}, \quad \mathbf{N}_{p3} = \begin{bmatrix} \mathbf{J} \\ \mathbf{B}_{p3} \end{bmatrix} \tag{2.54}$$

De manière similaire pour la membrure distale $d3$, on obtient :

$$\boldsymbol{\omega}_{d3} = \boldsymbol{\omega}_{ps} = \mathbf{J}_{ps} \dot{\boldsymbol{\theta}}_{ps} \tag{2.55}$$

et en se référant à la figure 2.8 :

$$\begin{aligned}
\dot{\mathbf{s}}_{d3} &= \mathbf{v}(S_{d3} \in d3/b) = \mathbf{v}(S_{d3} \in d3/ps) + \mathbf{v}(S_{d3} \in ps/b) \\
&= \boldsymbol{\omega}_{c/ps} \times \mathbf{p}_3 + \boldsymbol{\omega}_{ps} \times \mathbf{s}_{d3}
\end{aligned} \tag{2.56}$$

Donc la relation entre la vitesse de S_{d3} et les vitesses articulaires s'écrit sous forme matricielle, en utilisant la notation (2.6) :

$$\dot{\mathbf{s}}_{d3} = \mathbf{B}_{d3} \dot{\boldsymbol{\theta}}, \quad \text{avec} \quad \mathbf{B}_{d3} = \begin{bmatrix} -\hat{\mathbf{s}}_{d3} \mathbf{J}_{ps} & -\hat{\mathbf{p}}_3 \mathbf{J}_{c/ps} \end{bmatrix} \tag{2.57}$$

et ainsi :

$$\mathbf{t}_{d3} = \mathbf{N}_{d3} \dot{\boldsymbol{\theta}}, \quad \mathbf{N}_{d3} = \begin{bmatrix} \mathbf{J}_{ps} & \mathbf{0} \\ \mathbf{B}_{d3} \end{bmatrix} \tag{2.58}$$

Ceci termine les calculs cinématiques de tous les corps composant le système. On est donc en mesure de passer à la partie suivante, la dynamique du simulateur.

Chapitre 3

Analyse statique et dynamique du simulateur

Le modèle dynamique, constitué par les équations de mouvement permettant d'exprimer les couples nécessaires aux actionneurs en fonction des positions, vitesses et accélérations du système, est présenté dans ce chapitre. Tout d'abord, le modèle est développé en utilisant la méthode de Lagrange, puis certaines simplifications du modèle sont proposées afin d'alléger l'éventuelle mise en œuvre du modèle dans le contrôleur du mécanisme. Étant donné l'importance des couples statiques dans une telle application, l'équilibrage statique partiel du mécanisme est ensuite considéré, et son influence sur le comportement statique du simulateur est étudié. Finalement, le modèle dynamique incluant l'équilibrage statique est programmé, et ses résultats numériques sont comparés au modèle virtuel obtenu en utilisant le logiciel de simulation *ADAMS* pour différentes trajectoires.

3.1 Introduction

Beaucoup d'approches différentes peuvent être utilisées pour déterminer le modèle dynamique d'un mécanisme complexe comme ce simulateur de mouvement [13, 14, 15, 16]. Par le passé, le choix d'une méthode était souvent dicté par le coût en temps de calcul de la mise en œuvre numérique du modèle, et dans les années 80 différents chercheurs ont cherché à minimiser le nombre d'opérations nécessaires à la résolution numérique du modèle. Cependant, avec l'augmentation exponentielle de la puissance de calcul des ordinateurs, le défi de la mise en œuvre en temps réel d'un modèle dynamique n'en est plus un, et de nombreux contrôleurs peuvent aujourd'hui inclure une commande dynamique sans même nécessiter la distribution en parallèle des calculs.

D'autres critères sont donc désormais utilisés pour choisir une approche de développement plutôt qu'une autre, et dans ce mémoire, la formulation lagrangienne des équations dynamiques a été choisie. Ce choix s'est basé principalement sur le fait que cette formulation permet une dérivation simple des équations, et qu'elle ne fait pas apparaître les forces de contraintes lors du développement. Ainsi, en utilisant cette méthode, on obtient directement une expression des couples nécessaires aux actionneurs en fonction des coordonnées généralisées et de leurs dérivées, ce qui est la forme la plus appropriée pour la commande. De plus, l'interprétation physique des différents termes des équations est immédiate, facilitant d'éventuelles simplifications lors de la mise en œuvre des équations dans le contrôleur.

Le développement du modèle dynamique a plusieurs objectifs dans ce projet. En premier lieu, de façon générale, il permet de comprendre le comportement dynamique du système, et de connaître l'ordre de grandeur des couples qui seront nécessaires aux actionneurs, et donc de choisir les actionneurs les plus appropriés à cette application. Ensuite, ce modèle dynamique peut être mis en œuvre dans le contrôleur afin de concevoir un schéma de contrôle en couple plus précis qu'un schéma classique de type PID (Proportionnel-Intégral-Dérivé). Finalement, la partie statique de ce modèle est exploitée lors de l'optimisation de l'équilibrage statique partiel du système.

3.2 Développement du modèle dynamique

3.2.1 Notations supplémentaires et notions mathématiques utiles

Les quantités suivantes sont définies pour chaque corps j , pour $j = ps, c, p_i, d_i$ ($i = 1, 2, 3$) :

m_j la masse,

\mathbf{I}_{j0} le tenseur d'inertie du corps j dans la configuration de référence par rapport soit au point O pour les corps c et ps , soit au point S_k pour $k = p_i, d_i$, et exprimé dans le repère R_b ,

\mathbf{I}_j le même tenseur d'inertie, mais pour une configuration arbitraire du système, Pour les membrures proximales, on définit aussi :

I_{p_i} le moment d'inertie de la membrure proximale p_i autour de l'axe de rotation de l'actionneur i .

Les équations de Lagrange sont développées ici sous l'hypothèse principale que le mécanisme est constitué de corps rigides connectés par des articulations sans frottement. L'absence de frottement aux liaisons n'est pas réalisable en pratique, mais il est très difficile de modéliser ces frottements, et encore plus d'avoir un modèle correspondant au comportement réel des forces de frottement. Avec cette hypothèse, les équations de Lagrange du mouvement sont de la forme :

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\boldsymbol{\theta}}} \right) - \frac{\partial L}{\partial \boldsymbol{\theta}} = \boldsymbol{\tau} \quad (3.1)$$

où

- $\boldsymbol{\theta}$ est le vecteur des coordonnées généralisées du système, qui sont ici les variables articulaires.
- L est le lagrangien du système : $L = T - V$ avec :
 $T(\dot{\boldsymbol{\theta}}, \boldsymbol{\theta})$ l'énergie cinétique totale du système,
 $V(\boldsymbol{\theta})$ l'énergie potentielle totale du système.
- $\boldsymbol{\tau}$ est le vecteur des couples aux actionneurs.

En utilisant le fait que l'énergie potentielle ne dépend que de la position, on peut aussi écrire ces équations sous la forme :

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\boldsymbol{\theta}}} \right) - \frac{\partial T}{\partial \boldsymbol{\theta}} + \frac{\partial V}{\partial \boldsymbol{\theta}} = \boldsymbol{\tau} \quad (3.2)$$

Dans les prochaines sections, les équations dynamiques partielles pour chaque corps, c'est à dire les expressions des couples aux actionneurs dûs à chacun des corps séparément, sont développées. Ainsi, les mouvements de chaque corps peuvent être pris en compte lors de la dérivation du Lagrangien, et les contributions des différents corps dans les équations globales de mouvement peuvent être étudiées et comparées.

3.2.2 Dynamique de la cabine

La cabine ne subit que des rotations autour du point O . Alors son énergie cinétique peut être exprimée comme une énergie cinétique de rotation seulement autour de O de la façon suivante :

$$T_c = \frac{1}{2} \boldsymbol{\omega}^T \mathbf{I}_c \boldsymbol{\omega} = \frac{1}{2} \dot{\boldsymbol{\theta}}^T \mathbf{J}^T \mathbf{I}_c \mathbf{J} \dot{\boldsymbol{\theta}} \quad (3.3)$$

où \mathbf{I}_c dépend de l'orientation de la cabine, et peut être exprimée en utilisant la matrice de rotation définie à la section 2.1.2 :

$$\mathbf{I}_c = \mathbf{Q} \mathbf{I}_{c0} \mathbf{Q}^T \quad (3.4)$$

Alors la dérivée de l'énergie cinétique de la cabine par rapport aux vitesses généralisées est :

$$\frac{\partial T_c}{\partial \dot{\boldsymbol{\theta}}} = \mathbf{J}^T \mathbf{I}_c \mathbf{J} \dot{\boldsymbol{\theta}} \quad (3.5)$$

En dérivant l'équation (3.5) par rapport au temps, on obtient :

$$\frac{d}{dt} \left(\frac{\partial T_c}{\partial \dot{\boldsymbol{\theta}}} \right) = \mathbf{J}^T \mathbf{I}_c \mathbf{J} \ddot{\boldsymbol{\theta}} + \dot{\mathbf{J}}^T \mathbf{I}_c \mathbf{J} \dot{\boldsymbol{\theta}} + \mathbf{J}^T \dot{\mathbf{I}}_c \dot{\boldsymbol{\theta}} + \mathbf{J}^T \dot{\mathbf{I}}_c \mathbf{J} \dot{\boldsymbol{\theta}} \quad (3.6)$$

De plus, d'après les relations (3.4) et (2.25) (voir aussi [17] pour de plus amples explications), on a :

$$\begin{aligned} \dot{\mathbf{I}}_c &= \hat{\boldsymbol{\omega}} \mathbf{I}_c - \mathbf{I}_c \hat{\boldsymbol{\omega}}, \\ \text{et } \hat{\boldsymbol{\omega}} \mathbf{J} \dot{\boldsymbol{\theta}} &= \hat{\boldsymbol{\omega}} \boldsymbol{\omega} = \mathbf{0} \end{aligned} \quad (3.7)$$

Grâce à ces deux relations, le dernier terme de l'équation (3.6) peut être simplifié, et celle-ci s'écrit :

$$\frac{d}{dt} \left(\frac{\partial T_c}{\partial \dot{\boldsymbol{\theta}}} \right) = \mathbf{J}^T \mathbf{I}_c \mathbf{J} \ddot{\boldsymbol{\theta}} + \dot{\mathbf{J}}^T \mathbf{I}_c \mathbf{J} \dot{\boldsymbol{\theta}} + \mathbf{J}^T \dot{\mathbf{I}}_c \dot{\boldsymbol{\theta}} + \mathbf{J}^T \hat{\boldsymbol{\omega}} \mathbf{I}_c \mathbf{J} \dot{\boldsymbol{\theta}} \quad (3.8)$$

En utilisant la relation (3.7) de nouveau, on a d'après [17] :

$$\frac{\partial T_c}{\partial \boldsymbol{\theta}} = \mathbf{J}^T \mathbf{I}_c \mathbf{J} \dot{\boldsymbol{\theta}} \quad (3.9)$$

Le poids étant la seule force conservative agissant sur le système, l'énergie potentielle de la cabine est seulement due à la gravité et s'exprime :

$$V_c = -m_c g \mathbf{Z}_b^T \mathbf{s}_c \quad (3.10)$$

où m_c inclut la masse du pilote. Alors sa dérivée par rapport aux coordonnées généralisées est :

$$\frac{\partial V_c}{\partial \boldsymbol{\theta}} = \mathbf{J}^T (m_c g \mathbf{Z}_b \times \mathbf{s}_c) \quad (3.11)$$

Finalement, en ajoutant les équations (3.8), (3.9) et (3.11), on obtient les équations dynamiques de la cabine, exprimant les couples τ_c aux actionneurs dûs à ce corps :

$$\boldsymbol{\tau}_c = \mathbf{J}^T \mathbf{I}_c \mathbf{J} \ddot{\boldsymbol{\theta}} + \mathbf{J}^T \mathbf{I}_c \dot{\mathbf{J}} \dot{\boldsymbol{\theta}} + \mathbf{J}^T \hat{\boldsymbol{\omega}} \mathbf{I}_c \mathbf{J} \dot{\boldsymbol{\theta}} + \mathbf{J}^T (m_c g \mathbf{Z}_b \times \mathbf{s}_c) \quad (3.12)$$

3.2.3 Dynamique de la plaque support

Puisque la plaque support possède elle aussi seulement des mouvements de rotation autour de O , les équations de mouvement pour ce corps sont obtenues de façon similaire aux précédentes. Par contre, seulement les actionneurs 1 et 2, et donc les couples τ_1 et τ_2 , sont affectés car la plaque support appartient au premier mécanisme, et n'affecte donc pas le troisième moteur. Le modèle dynamique de la plaque consistera seulement en deux équations (ce qui reste valable pour tous les corps du mécanisme 1). On obtient :

$$\boldsymbol{\tau}_{ps} = \mathbf{J}_{ps}^T \mathbf{I}_{ps} \mathbf{J}_{ps} \ddot{\boldsymbol{\theta}}_{ps} + \mathbf{J}_{ps}^T \mathbf{I}_{ps} \dot{\mathbf{J}}_{ps} \dot{\boldsymbol{\theta}}_{ps} + \mathbf{J}_{ps}^T \hat{\boldsymbol{\omega}}_{ps} \mathbf{I}_{ps} \mathbf{J}_{ps} \dot{\boldsymbol{\theta}}_{ps} + \mathbf{J}_{ps}^T (m_{ps} g \mathbf{Z}_b \times \mathbf{s}_{ps}) \quad (3.13)$$

3.2.4 Dynamique des membrures

La membrure p_1 tourne autour d'un axe fixe, donc on peut exprimer son énergie cinétique comme une énergie cinétique de rotation autour de cet axe. De plus, son énergie potentielle est seulement due à la gravité. Ainsi, en se référant à sa variable cinématique de l'équation (2.41), on obtient simplement le Lagrangien de cette membrure :

$$L_{p1} = \frac{1}{2} I_{p1} \dot{\theta}_1^2 - m_{p1} g \mathbf{Z}_b^T \mathbf{c}_{p1} \quad (3.14)$$

En dérivant ce Lagrangien selon l'équation (3.1), on obtient le couple au premier actionneur dû à la membrure p_1 :

$$\tau_{p1} = I_{p1} \ddot{\theta}_1 - m_{p1} g \mathbf{Y}_b^T \mathbf{c}_{p1} \quad (3.15)$$

De la même façon, le couple à l'actionneur 2 dû à la membrure p_2 est :

$$\tau_{p2} = I_{p2} \ddot{\theta}_2 + m_{p2} g \mathbf{X}_b^T \mathbf{c}_{p2} \quad (3.16)$$

Comme on l'a vu au chapitre précédent, la membrure d_1 ne réalise quant à elle que des translations circulaires, donc seule son énergie cinétique de translation peut varier, et cette dernière s'exprime de façon générale : $T_{d1} = \frac{1}{2} m_{d1} \|\dot{\mathbf{s}}_{d1}\|^2$. En substituant l'équation (2.43) dans cette relation, on obtient le Lagrangien L_{d1} de cette membrure :

$$L_{d1} = \frac{1}{2} m_{d1} l_{12}^2 \dot{\theta}_1^2 - m_{d1} g \mathbf{Z}_b^T \mathbf{u}_1 \quad (3.17)$$

La dérivation de cette expression selon l'équation (3.1) permet d'obtenir le couple à l'actionneur 1 dû à $d1$:

$$\tau_{d1} = m_{d1} l_{11}^2 \ddot{\theta}_1 - m_{d1} g l_{11} \cos \theta_1 \quad (3.18)$$

Les trois autres membrures, d_2 , p_3 et d_3 , subissent à la fois des mouvements de translation et de rotation, donc leur énergie cinétique prend la forme générale :

$$T_i = \frac{1}{2} m_i \|\dot{\mathbf{s}}_i\|^2 + \frac{1}{2} \boldsymbol{\omega}_i^T \mathbf{I}_i \boldsymbol{\omega}_i \quad (3.19)$$

On peut écrire de nouveau cette équation de manière plus compacte en utilisant le torseur cinématique \mathbf{t}_i défini au chapitre précédent, et on a alors :

$$T_i = \frac{1}{2} \mathbf{t}_i^T \mathbf{M}_i \mathbf{t}_i = \frac{1}{2} \dot{\boldsymbol{\theta}}^T \mathbf{N}_i^T \mathbf{M}_i \mathbf{N}_i \dot{\boldsymbol{\theta}} \quad (3.20)$$

pour $i = d_2, p_3, d_3$, où \mathbf{M}_i correspond à la généralisation à 6 dimensions de la matrice \mathbf{I}_i , c'est à dire :

$$\mathbf{M}_i = \begin{bmatrix} \mathbf{I}_i & \mathbf{0} \\ \mathbf{0} & m_i \mathbf{1} \end{bmatrix}$$

Ainsi, de manière similaire à l'équation (3.7) on a :

$$\dot{\mathbf{M}}_i = \boldsymbol{\Omega}_i \mathbf{M}_i - \mathbf{M}_i \boldsymbol{\Omega}_i, \quad \text{avec} \quad \boldsymbol{\Omega}_i = \begin{bmatrix} \hat{\boldsymbol{\omega}}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (3.21)$$

$$\text{et} \quad \boldsymbol{\Omega}_i \mathbf{N}_i \dot{\boldsymbol{\theta}} = \begin{bmatrix} \hat{\boldsymbol{\omega}}_i \boldsymbol{\omega}_i \\ \mathbf{0} \end{bmatrix} = \mathbf{0}_{6 \times 1}$$

Cette forme compacte de l'énergie cinétique ainsi que les similarités avec les équations de la cabine permettent de dériver le Lagrangien de manière identique à la section 3.2.2. Un indice k est ajouté aux vitesses généralisées, qui prend la valeur ps lorsque la membrure d_2 est considérée et \emptyset lorsque les membrures p_3 ou d_3 sont considérées, car le troisième actionneur n'est pas affecté par les mouvements de d_2 . On obtient :

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial T_i}{\partial \dot{\boldsymbol{\theta}}_k} \right) &= \mathbf{N}_i^T \mathbf{M}_i \mathbf{N}_i \ddot{\boldsymbol{\theta}}_k + \dot{\mathbf{N}}_i^T \mathbf{M}_i \mathbf{N}_i \dot{\boldsymbol{\theta}}_k + \mathbf{N}_i^T \mathbf{M}_i \dot{\mathbf{N}}_i \dot{\boldsymbol{\theta}}_k + \mathbf{N}_i^T \boldsymbol{\Omega}_i \mathbf{M}_i \mathbf{N}_i \dot{\boldsymbol{\theta}}_k \\ \frac{\partial T_i}{\partial \boldsymbol{\theta}_k} &= \dot{\mathbf{N}}_i^T \mathbf{M}_i \mathbf{N}_i \dot{\boldsymbol{\theta}}_k \end{aligned} \quad (3.22)$$

L'énergie potentielle de chaque membrure étant seulement due à la gravité, on a simplement $V_i = -m_i g \mathbf{Z}_b^T \mathbf{s}_i$, et

$$\frac{\partial V_i}{\partial \boldsymbol{\theta}_k} = -\mathbf{B}_i^T (m_i g \mathbf{Z}_b) \quad (3.23)$$

où les matrices \mathbf{B}_i ont été développées à la section 2.5.

Finalement, pour les corps d_2, p_3 et d_3 les équations partielles de mouvements sont :

$$\boldsymbol{\tau}_i = \mathbf{N}_i^T \mathbf{M}_i \mathbf{N}_i \ddot{\boldsymbol{\theta}}_k + \mathbf{N}_i^T \mathbf{M}_i \dot{\mathbf{N}}_i \dot{\boldsymbol{\theta}}_k + \mathbf{N}_i^T \boldsymbol{\Omega}_i \mathbf{M}_i \mathbf{N}_i \dot{\boldsymbol{\theta}}_k - \mathbf{B}_i^T (m_i g \mathbf{Z}_b) \quad (3.24)$$

3.2.5 Modèle dynamique complet du système

Finalement, en sommant les équations partielles (3.12), (3.13), (3.15), (3.16), (3.18) et (3.24) pour $i = d_2, p_3, d_3$, on obtient les équations de Lagrange du système sous la forme :

$$\boldsymbol{\tau} = \mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{g}(\boldsymbol{\theta}) \quad (3.25)$$

où $\mathbf{M}(\boldsymbol{\theta})$ est la matrice d'inertie généralisée du système, et elle est fonction seulement des coordonnées généralisées, $\mathbf{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ est le vecteur des forces de Coriolis et centrifuges (qui est une fonction quadratique des vitesses généralisées), et $\mathbf{g}(\boldsymbol{\theta})$ est le vecteur des couples dûs à la gravité.

Ces équations permettent de résoudre le problème dynamique direct, c'est à dire la détermination de l'évolution des coordonnées articulaires et de leurs dérivées lorsque les couples aux actionneurs ainsi que les conditions initiales sont connus, en considérant les équations (3.25) comme un système d'équations différentielles non-linéaires.

Ces équations permettent aussi de résoudre le problème dynamique inverse qui nous intéresse ici, c'est à dire la détermination des couples requis aux actionneurs lorsque les mouvements de la cabine sont prescrits, et peuvent donc être exploitées dans l'algorithme de contrôle du mécanisme afin de concevoir un contrôleur plus efficace. La méthode utilisée pour développer ces équations, c'est à dire calculer le comportement dynamique de chaque corps en tenant compte des spécificités des mouvements, permet de connaître les contributions de chaque corps à la partie statique et dynamique du modèle, et donc de bien comprendre l'origine des couples nécessaires aux actionneurs.

3.3 Simplification du modèle pour le contrôle

Comme on l'a vu dans les sections précédentes, certains calculs dynamiques des membrures sont compliqués, alors que leur contribution au modèle dynamique global n'est pas significative car les masses et inerties de la cabine et de la plaque support sont largement prédominantes. Ayant développé les différentes contributions séparément, on peut donc prendre avantage de ce fait en approximant le modèle précédent en négligeant et/ou simplifiant certains termes des équations lors de la mise en œuvre du modèle dans le contrôleur, permettant ainsi un algorithme de contrôle plus rapide et plus pratique.

D'abord, comme on peut le voir clairement à la section 2.5.1, il est très difficile d'obtenir les variables cinématiques de la membrure d_2 en fonction des coordonnées généralisées. De plus, son comportement dynamique ne permet aucune simplification particulière des équations (il en va de même que pour p_3 et d_3). A partir de ces remarques, on peut donc considérer différentes approximations du modèle dynamique :

1. négliger complètement d_2 .
2. remplacer d_2 par deux masses ponctuelles, valant $\frac{m_{d2}}{2}$ chacune, et situées aux extrémités de la membrure.
3. négliger complètement p_3 et d_3 .
4. négliger les mouvements translationnels de p_3 et d_3 .
5. considérer seulement la contribution en régime statique des membrures d_2 , p_3 , d_3 .

L'approximation 2 permet d'ajouter les deux masses ponctuelles à la plaque support et à la membrure p_1 respectivement en modifiant les masses, tenseurs d'inertie et centre de masse de ces corps. En conséquence, aucun calcul en temps réel n'est ajouté par cette approximation par rapport à la première approximation.

Dans la même perspective, l'approximation 4 permet de considérer p_3 comme faisant partie de la cabine, et d_3 comme faisant partie de la plaque support. Ainsi, on peut ajouter leurs tenseurs d'inertie en O ainsi que leurs masses aux corps précédents.

Différentes combinaisons de ces approximations peuvent être utilisées, et un indice de l'erreur introduite par rapport au modèle complet, basée sur son erreur relative, est calculé à la section 3.5 pour différents exemples numériques de trajectoires. Cet indice de l'erreur, pour une approximation a et une trajectoire t , est défini comme :

$$\lambda_a(t) = 100 \times moy_t \left(\frac{\|\boldsymbol{\tau} - \boldsymbol{\tau}_a\|}{\|\boldsymbol{\tau}\|} \right) \quad (3.26)$$

où $\boldsymbol{\tau}$ est le vecteur des couples aux actionneurs calculé avec le modèle complet, $\boldsymbol{\tau}_a$ est le vecteur des couples aux actionneurs calculé avec le modèle approximé avec a , et moy_t représente la valeur moyenne sur la trajectoire t .

3.4 Équilibrage statique partiel

3.4.1 Introduction

Un mécanisme est parfaitement statiquement équilibré lorsque son énergie potentielle est constante pour toutes les configurations possibles, et en conséquence aucun couple n'est nécessaire pour maintenir l'équilibre statique du mécanisme. Cet équilibrage statique, aussi désigné par le terme compensation de la gravité, peut être atteint principalement en utilisant deux méthodes. La première consiste à utiliser des contrepoids pour que le centre de masse demeure fixe dans la direction de la gravité. L'énergie potentielle de gravité reste alors constante, et le mécanisme est équilibré. Malheureusement, cette méthode nécessite souvent l'ajout de tiges, et surtout elle augmente sensiblement la masse et l'inertie du mécanisme. La seconde méthode consiste en l'utilisation de ressorts, qui permettent d'équilibrer l'énergie potentielle de gravité par de l'énergie potentielle élastique.

Pour des mécanismes en rotation, Streit et al. [18] ont d'abord déterminé les conditions d'un équilibrage parfait pour un corps en rotation autour d'un joint de cardan avec 1 et 2 ressorts, puis Gosselin [19] a généralisé ces conditions aux mécanismes parallèles à 3 degrés de liberté en rotation, et ce en utilisant un nombre quelconque de ressorts. Dans la littérature traitant de la compensation de la gravité, les simulateurs de mouvement sont souvent cités comme applications particulièrement intéressantes de l'équilibrage statique [19, 20, 21], et Gosselin [22] a même réalisé une étude spécifique à la conception de plates-formes de mouvement équilibrées dédiées à la simulation de vol. Cet intérêt pour l'équilibrage des mécanismes de simulation de vol est dû au fait que de tels systèmes doivent déplacer de très grandes charges (jusqu'à plusieurs tonnes pour des simulateurs d'entraînement). Le simple fait de maintenir la cabine en régime statique nécessite donc des couples ou forces élevés aux actionneurs, ce qui contribue aux larges coûts de fonctionnement de ces simulateurs. De ce fait, l'équilibrage statique de tels système permettrait de réduire de manière significative les coûts, et/ou de réduire la puissance et la taille des actionneurs.

Pourtant, l'équilibrage des simulateurs de mouvement reste encore aujourd'hui souvent théorique, comme l'illustre la figure 3.1, car les conditions d'équilibrage sont peu

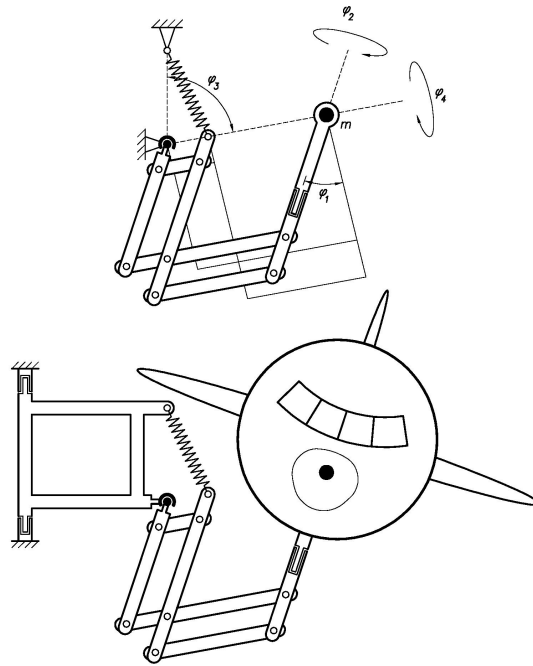


FIG. 3.1 – Concept d'équilibrage de simulateur (reproduit de [20]).

compatibles avec le peu d'espace libre et l'amplitude des efforts à compenser présents dans de tels systèmes. Une solution partielle à ce problème peut être apportée en considérant non pas un équilibrage statique parfait, mais plutôt un équilibrage partiel qui, tout en ne rencontrant pas exactement les conditions théoriques, s'en rapproche le plus possible pour permettre la conception pratique de l'équilibrage en éliminant une grande partie des couples statiques nécessaires aux actionneurs. La notion d'efficacité de l'équilibrage statique a été introduite par Johnson et Ebert-Uphoff [21], qui ont étudié différentes sources d'erreurs (non linéarité des ressorts, utilisation de poulies, fabrication) lors de la mise en œuvre de l'équilibrage d'un mécanisme parallèle à six degrés de liberté. Brinkman et Herder [23] ont considéré l'équilibrage partiel de mécanismes dû à des considérations plus générales (ressorts à longueur libre non nulle, points d'attache des ressorts ne rencontrant pas exactement les conditions géométriques prescrites à cause de contraintes de construction) et ont développé une méthode d'optimisation de la conception de tels systèmes.

3.4.2 Solution mise en oeuvre

Les nombreuses contraintes de cette application, principalement le grand espace de travail en rotation, l'utilisation d'une patte passive, et la grande charge de la cabine, laissent bien peu de place combiné à de grands couples statiques à compenser $\mathbf{g}(\boldsymbol{\theta})$. Les conditions géométriques requises pour atteindre un équilibre statique parfait, décrites dans [19], sont donc difficiles à remplir, et la réalisation pratique d'un tel équilibre s'est avérée être un problème lors de la conception de l'architecture, comme on l'a vu à la section 1.3. Un équilibre partiel a donc été développé, en deux principales étapes :

- Lors de la première étape, seules la cabine et la plaque support ont été considérées, car ces deux corps représentent à eux seuls près de 90% de la masse totale du système, et sont donc largement prédominants dans le modèle statique du simulateur. Une combinaison des deux méthodes d'équilibre a alors été appliquée, incluant un contrepoids et trois ressorts d'extension.
- Lors de la deuxième étape, le système réel a alors été considéré, et la raideur des ressorts a été optimisée pour obtenir une élimination des couples statiques la plus importante possible.

3.4.2.1 Équilibre de l'effecteur

Un contrepoids, qui est en fait le troisième actionneur du mécanisme, est utilisé pour placer le centre de masse de la cabine sur son axe Z . En conséquence, ce centre de masse reste fixe lorsque le deuxième mécanisme bouge. Cela signifie qu'aucun couple n'est requis au troisième actionneur pour maintenir la cabine en régime statique. De plus, la plaque support est symétrique par rapport à son axe Z et est homogène. Son centre de masse est donc sur le même axe que celui de la cabine. Ces deux corps peuvent donc être considérés comme un seul dans les conditions statiques, ayant pour centre de masse le barycentre des deux précédents. En régime statique, on appelle ce corps l'effecteur dans la suite de ce mémoire, \mathbf{r} désigne le vecteur reliant le point O à son centre de masse, comme illustré à la figure 3.2, et sa masse est $m_e = m_c + m_{ps}$.

Pour équilibrer ensuite l'effecteur selon les axes de rotation du premier mécanisme, le meilleur compromis trouvé est une combinaison de trois ressorts d'extension identiques situés sous l'effecteur, et reliés à celui-ci et à la base comme illustré à la figure 3.2. Les points d'attache reliant les ressorts à la base sont notés A_i et à l'effecteur B_i . Les

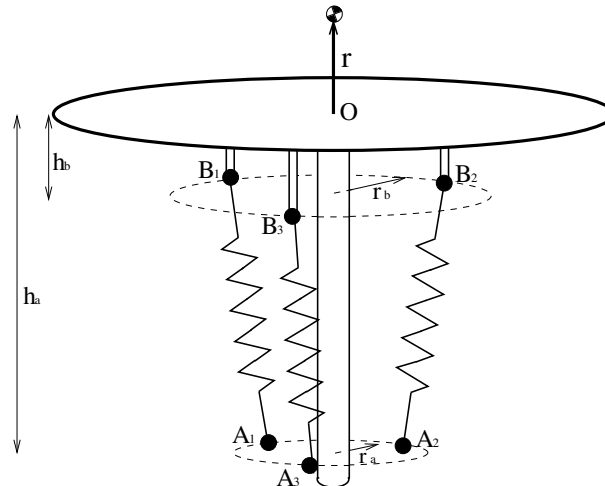


FIG. 3.2 – Paramètres de l'équilibrage statique.

vecteurs connectant O à A_i et B_i sont nommés respectivement \mathbf{a}_i et \mathbf{b}_i , pour $i = 1, 2, 3$. Dans la configuration de référence, $\mathbf{b}_i = \mathbf{b}_{i0}$, $i = 1, 2, 3$ et $\mathbf{r} = \mathbf{r}_0 = [0, 0, r_{0z}]^T$. Dans une configuration quelconque, on a donc :

$$\mathbf{b}_i = \mathbf{Q}_{ps} \mathbf{b}_{i0}, i = 1, 2, 3 \quad \text{et} \quad \mathbf{r} = \mathbf{Q}_{ps} \mathbf{r}_0 \quad (3.27)$$

Les points B_i sont situés dans un plan π_b parallèle au plan $(OX_{ps}Y_{ps})$, à une distance h_b . Ces points forment dans π_b un triangle équilatéral, dont le centroïde est situé sur l'axe Z_{ps} et dont le rayon du cercle circonscrit est r_b . De façon similaire, les points A_i sont situés dans un plan π_a parallèle au plan (OX_bY_b) , à une distance h_a . Ils constituent un triangle équilatéral dans π_a , dont le centroïde est situé sur l'axe Z_b , et dont le rayon du cercle circonscrit est r_a .

Cette configuration des ressorts ne remplit pas les conditions théoriques d'équilibrage parfait pour deux raisons :

- Si les points B_i sont répartis comme décrit plus haut, les points A_i doivent être situés sur l'axe Z_b pour atteindre l'équilibrage parfait. Mais dans notre cas ceci n'est pas possible à cause de la patte passive centrée sur cet axe. Le rayon r_a est donc non nul en pratique, mais gardé aussi petit que le système le permet sans interférences mécaniques.
- Les conditions théoriques d'équilibrage supposent l'utilisation de ressorts à longueur libre nulle, ce qui peut être réalisé en employant des poulies et des câbles,

ou différents types de systèmes de guidage (plusieurs de ces systèmes sont décrits dans [24]). Cependant, pour des raisons de simplicité, de transportabilité du mécanisme, et pour minimiser les forces de friction non modélisées, des ressorts réels avec une longueur libre non nulle ont été employés ici. Leur longueur libre l_0 a pourtant pu être réduite en appliquant une précharge à la longueur initiale L_0 .

Si les deux conditions précédentes étaient remplies, la raideur des ressorts k_{ideal} permettant d'atteindre un équilibrage parfait de l'effecteur serait, en appliquant les résultats donnés dans [19] :

$$k_{ideal} = \frac{1}{3} \frac{m_e g r_{0z}}{h_b h_a} \quad (3.28)$$

3.4.2.2 Optimisation de la raideur des ressorts

Pour le système réel, on cherche alors à maximiser l'élimination des couples statiques grâce à l'optimisation de la valeur de la raideur des ressorts, en tenant compte à la fois de la longueur libre des ressorts, des points d'attaches réels A_i et B_i , et des couples statiques dûs aux membrures. Pour cela, on calcule tout d'abord le modèle statique des ressorts. Leur énergie potentielle élastique, tenant compte de la longueur libre l_0 , est :

$$V_s = \frac{1}{2} k \sum_{i=1}^3 (l_i - l_0)^2 \quad (3.29)$$

où l_i est la longueur du ressort i :

$$l_i = \sqrt{(\mathbf{b}_i - \mathbf{a}_i)^T (\mathbf{b}_i - \mathbf{a}_i)} \quad (3.30)$$

En substituant les équations (3.27) et (3.30) dans (3.29) et en dérivant cette dernière par rapport à $\boldsymbol{\theta}_{ps}$, on obtient les couples aux actionneurs $\boldsymbol{\tau}_s$ dûs aux ressorts :

$$\boldsymbol{\tau}_s(\boldsymbol{\theta}_{ps}) = \frac{\partial V_s}{\partial \boldsymbol{\theta}_{ps}} = -\mathbf{J}_{ps}^T k \sum_{i=1}^3 \left(1 - \frac{l_0}{l_i}\right) (\mathbf{b}_i \times \mathbf{a}_i) \quad (3.31)$$

Alors, en ajoutant ces couples $\boldsymbol{\tau}_s(\boldsymbol{\theta}_{ps})$ lors du calcul des couples statiques $\mathbf{g}(\boldsymbol{\theta}_{ps})$, on obtient les couples statiques du système partiellement équilibré statiquement (la masse des ressorts a été négligée). Ainsi, on peut estimer l'amplitude des couples éliminés localement pour une configuration quelconque du système par l'indice δ_τ , qui est la

différence entre les normes des couples sans et avec l'équilibrage pour cette configuration :

$$\delta_\tau = \|\mathbf{g}\| - \|\boldsymbol{\tau}_s + \mathbf{g}\| \quad (3.32)$$

En intégrant cet indice sur l'espace de travail w , on obtient finalement une mesure globale de la performance de l'équilibrage par rapport aux couples, que l'on note Δ :

$$\Delta = \frac{\int_w \delta_\tau dw}{\int_w dw} \quad (3.33)$$

Avec cet indice, les effets de la variation de k sur les couples éliminés (tous les autres paramètres demeurant constants) peut être mise en valeur, comme illustré à la figure 3.3.

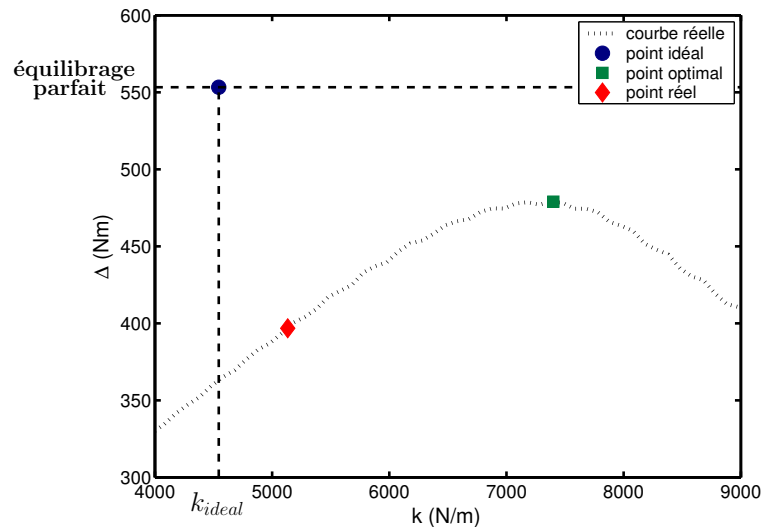


FIG. 3.3 – Évolution de Δ en fonction de k .

Malheureusement, la valeur optimale de raideur, marquée d'un carré à la figure 3.3, n'a pas pu être atteinte en pratique lors de la fabrication des ressorts. La valeur réelle de la raideur obtenue est $k = 5133$ N/m (le tableau 3.1 résume les principales caractéristiques des ressorts réels données par le fournisseur, alors que la fiche technique complète est en Annexe B.1), et la figure 3.4 montre une photographie d'un de ces ressorts. On remarque à la figure 3.3 l'importance de l'équilibrage statique. En effet, l'indice Δ correspond pour les ressorts réels à 397 Nm économisés en moyenne, ce qui est considérable, et même 479 Nm pour la valeur optimale de la raideur. On constate aussi le bénéfice de l'optimisation, car en conservant la valeur k_{ideal} , l'indice serait seulement

$\Delta = 363$ Nm pour l'équilibrage réel. On peut donc augmenter considérablement les couples économisés en considérant le système réel.

TAB. 3.1 – Caractéristiques des ressorts réels.

Caractéristique	Symbole	Valeur
raideur	k	5133 N/m
longueur initiale	L_0	0.537 m
Tension à la longueur initiale	F_0	623 N
Longueur libre résultante	l_0	0.416 m

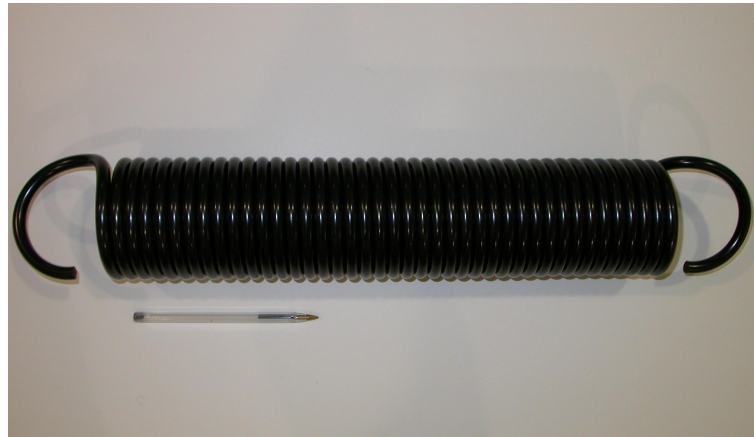


FIG. 3.4 – Photographie d'un des ressorts.

Une autre mesure de la performance de l'équilibrage est son efficacité, qui est une mesure relative de l'élimination des couples statiques, définie comme :

$$e = \frac{\int_w \left(1 - \frac{\|\boldsymbol{\tau}_s + \mathbf{g}\|}{\|\mathbf{g}\|}\right) dw}{\int_w dw} \quad (3.34)$$

Alors une efficacité égale à 1 indique un équilibrage parfait, alors que lorsque $e = 0$ il n'y a aucun équilibrage, et si $e < 0$ cela signifie que la performance du système avec un équilibrage statique partiel est pire que sans équilibrage du tout, en ce qui concerne l'amplitude des couples requis aux actionneurs. Pour le système étudié, avec les ressorts décrits précédemment, on obtient $e = 0.69$. De plus, l'efficacité calculée pour la région de l'espace de travail w où $\|\mathbf{g}\| > 100$ Nm, qui représente plus de 90% de

w , est $e_{>100} = 0.745$, ce qui montre que l'efficacité de l'équilibrage statique est meilleure lorsque les couples statiques sont élevés.

3.5 Exemples numériques et comparaison avec le modèle *ADAMS*

Deux exemples numériques sont inclus dans cette section, d'abord pour valider le modèle dynamique (incluant l'équilibrage statique) en comparant ses résultats avec ceux obtenus par le modèle *ADAMS* illustré à la figure 3.5, puis pour étudier les effets des différentes simplifications du modèle proposées.

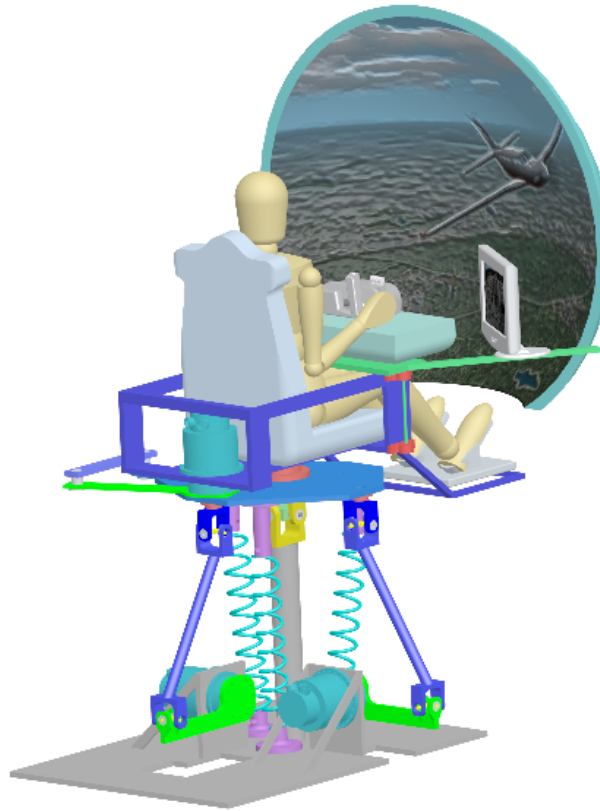


FIG. 3.5 – Modèle *ADAMS* du simulateur.

En ce qui concerne le modèle dynamique complet, les dérivées par rapport au temps des matrices \mathbf{N}_i , $i = d_2, p_3, d_3$ ont été calculées numériquement pour des raisons de simplicité.

Les paramètres principaux du système sont donnés à la table 3.2 en unités SI. La patte

2 n'apparaît pas dans ce tableau car dans le système réel les pattes 1 et 2 possèdent la même géométrie. Les tenseurs d'inertie, exprimés en kgm^2 , sont :

$$\mathbf{I}_{c0} = \begin{bmatrix} 88.1 & -1 & -11.8 \\ -1 & 121.9 & 0.4 \\ -11.8 & 0.4 & 52.5 \end{bmatrix}, \mathbf{I}_{ps0} = \begin{bmatrix} 4.4 & 0 & 0 \\ 0 & 3.63 & 0 \\ 0 & 0 & 5.75 \end{bmatrix}$$

TAB. 3.2 – Paramètres principaux (en unités SI).

Pattes							
l_{11}	l_{21}	l_{13}	l_{23}	m_{p1}	m_{d1}	m_{p3}	m_{d3}
0.36	0.692	0.39	0.536	7.63	3.23	3.55	5.45
Effecteur							
m_c		m_{ps}		$\mathbf{p}_c^T \mathbf{Z}$		$\mathbf{p}_{ps}^T \mathbf{Z}$	
220		50		-0.49		-0.115	
Ressorts							
h_b		h_a		r_b		r_a	
0.095		0.805		0.25		0.15	

3.5.1 Simulation d'une manœuvre de décollage

Une manœuvre typique de Boeing 747 a été utilisée pour tester le modèle dynamique (une définition exhaustive de cette manœuvre ainsi que de plus amples informations sont disponibles dans [6]). Elle consiste en la simulation du décollage de cet avion, suivi par une panne du moteur extérieur droit. La simulation dure 25 secondes. Les mouvements réels de l'avion sont filtrés pour respecter l'espace de travail du simulateur, en utilisant des *Washout Filters* conçus spécialement et décrits dans le prochain chapitre.

Les courbes des couples obtenus par le modèle dynamique complet et par la simulation *ADAMS* en fonction du temps sont tracées à la figure 3.6. Comme les deux courbes peuvent être difficilement distinguées, la courbe de la différence entre les deux a été ajoutée.

De manière globale, les accélérations sont de faible amplitude lors de cette trajectoire car l'avion simulé est un transporteur commercial possédant une grande inertie.

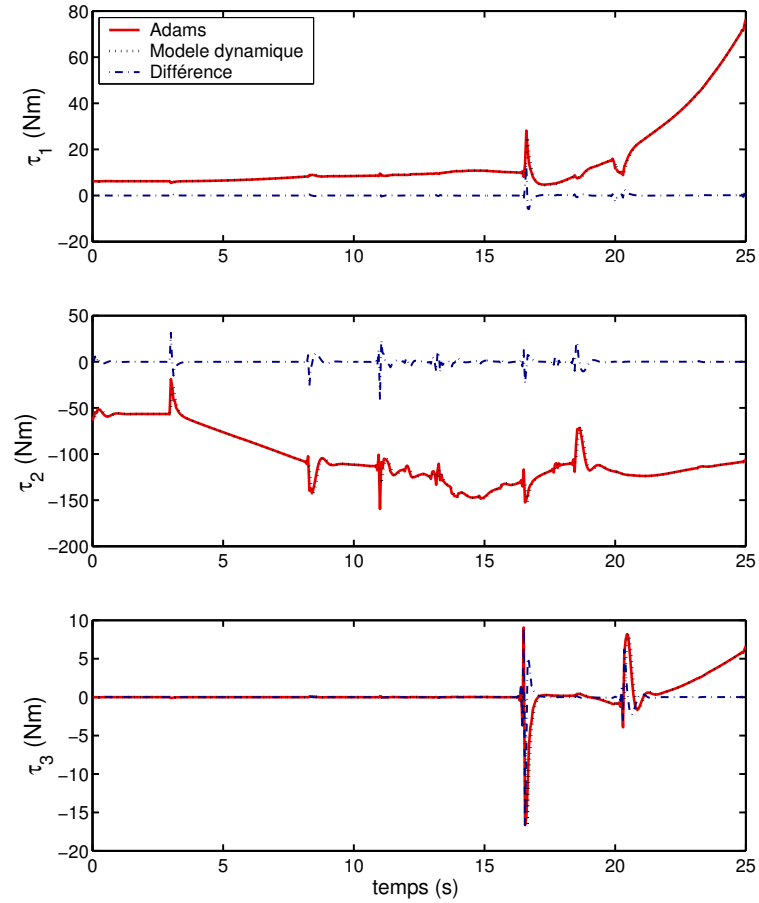


FIG. 3.6 – Couples obtenus pour la manœuvre de décollage.

La plus grande partie des couples est donc employée à supporter la charge contre la gravité. L'amplitude des couples aux actionneurs est donc relativement faible grâce à l'équilibrage statique (sans équilibrage statique τ_1 et τ_2 atteignent respectivement des amplitudes de 370 Nm et 545 Nm, comparés respectivement à 75 Nm et 159 Nm avec l'équilibrage statique).

L'inspection graphique des différentes courbes valide clairement la partie statique du modèle.

3.5.2 Mouvement sinusoïdal

Afin de tester le modèle dynamique pour des mouvements impliquant de plus grandes accélérations, une autre trajectoire cartésienne a été définie. Cette trajectoire est basée sur des variations sinusoïdales des angles d'Euler β et γ , c'est à dire :

$$\beta = \gamma = \frac{25\pi}{180} \sin(\pi t) \quad 0 \leq t \leq 2s$$

Les couples résultants sont tracés à la figure 3.7.

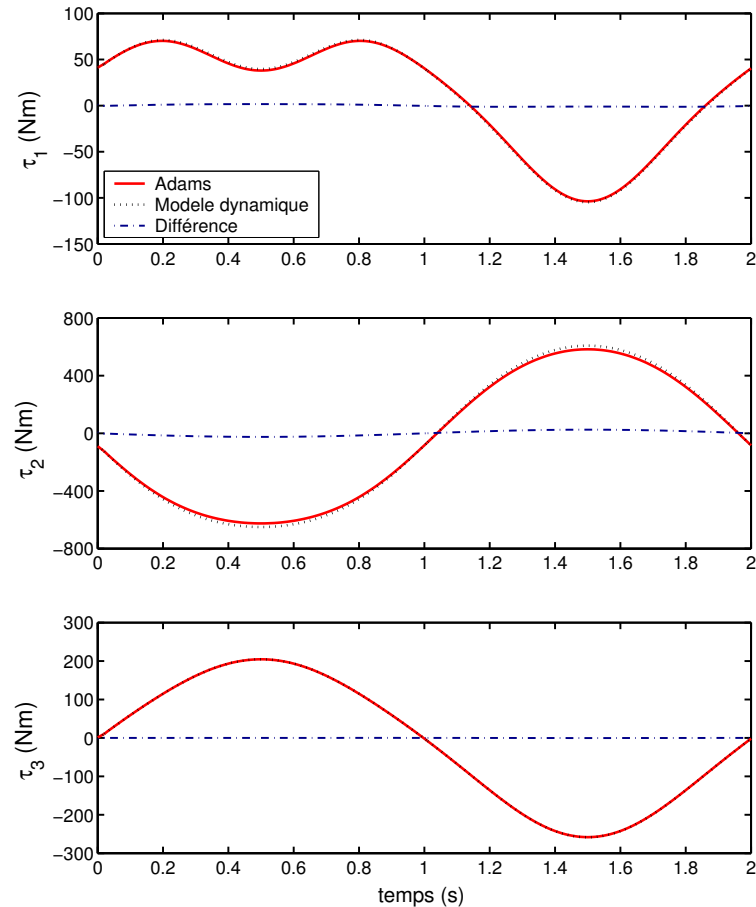


FIG. 3.7 – Couples obtenus pour la manœuvre sinusoïdale.

Même avec des accélérations de grandes amplitudes, les différences entre le modèle dynamique et la simulation de *ADAMS* restent négligeables, et principalement dues aux arrondis numériques, donc le modèle dynamique peut être validé complètement. Aussi, on peut constater avec ces deux trajectoires que l'amplitude des couples nécessaires ne dépasse jamais la capacité des moteurs couplés aux réducteurs choisis (ils peuvent fournir un couple de 715 Nm de manière continue, et de 1960 Nm instantanément).

3.5.3 Comparaison des différentes approximations

L'indice de l'erreur λ_a défini à l'équation (3.26) de la section 3.3 est calculé pour les deux trajectoires précédentes, et ce pour les différentes approximations, incluant les résultats des simulations *ADAMS*. Les valeurs obtenues sont résumées au tableau 3.3.

TAB. 3.3 – Indicateur de l'erreur $\lambda_a(t)$

approx.	1	2	3	4	5	<i>ADAMS</i>
Décollage	11.5	11.3	48.4	3.8	0.1	1.6
Sinusoïde	4.2	4.3	16.1	2.9	2.3	3.7

Comme on peut le voir clairement, les valeurs de λ_a obtenues pour la trajectoire de décollage sont bien plus élevées que celles obtenues pour la trajectoire sinusoïdale, à l'exception de l'approximation 5. En fait, comme l'équilibrage statique est basé sur l'effecteur, il diminue grandement la proportion de ce dernier dans la contribution aux couples statiques. Pour une trajectoire quasi-statique comme ce décollage, les pattes ne peuvent donc pas être négligées dans le modèle statique. En particulier, les résultats de l'approximation 3 sont les plus mauvais car l'actionneur 3 est seulement affecté par la cabine et la patte 3. Négliger complètement cette patte change de façon drastique le comportement statique et dynamique du système sur cet axe. Par ailleurs, comme le montrent les résultats de l'approximation 5, ainsi que tous les résultats de la trajectoire sinusoïdale, la contribution dynamique des membrures est bien moins critique, car les effets de l'inertie de la cabine sont pleinement ressentis.

Finalement, dans le contexte de la mise en œuvre du contrôleur, les approximations portant sur l'omission des contributions statiques et/ou de la patte 3 peuvent d'ores et déjà être écartées, alors que les approximations 4 et/ou 5 semblent particulièrement prometteuses. Leurs effets sur les temps de calcul en temps réel du modèle devront donc être étudiés en fonction de l'erreur introduite par les approximations.

Chapitre 4

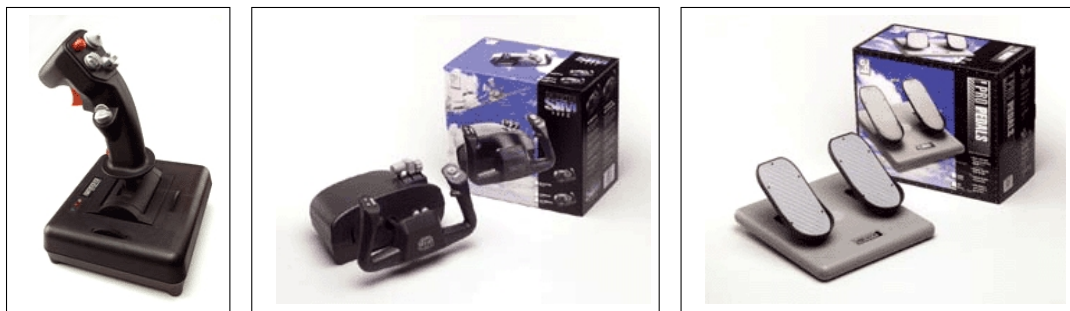
Architecture du système de calcul

Ce chapitre traite du système de calcul, qui est un sous-système clé du simulateur de mouvement car il est chargé de calculer les consignes de tous les autres sous-systèmes. D'abord, l'architecture générale est décrite à partir des principaux rôles attribués à ce système. Ensuite, la réalisation matérielle du système, ainsi que les différents moyens de communication utilisés, sont détaillés. Enfin, les principales composantes logicielles, c'est à dire la modélisation de l'avion par X-Plane, la génération des mouvements du simulateur, et son contrôle en temps réel, sont développées.

4.1 Architecture générale

4.1.1 Entrée du système

L'entrée du système de calcul est le signal des positions des commandes de vol, qui sont déterminées par le pilote. Le système des commandes de vol devait à la fois être réaliste et être suffisamment polyvalent pour pouvoir représenter les commandes de vol de tous les avions simulés.



(a) Manche.

(b) Gouvernail.

(c) Pédales.

FIG. 4.1 – Les commandes de vol.

Le choix s'est porté sur l'achat à la fois d'un manche, illustré à la figure 4.1(a), qui est représentatif d'une commande d'avion rapide comme un avion de combat, et d'un gouvernail, illustré à la figure 4.1(b), qui est plutôt représentatif d'une commande présente dans un avion de transport commercial. Finalement, le palonnier, qui est actionné avec les deux pieds du pilote, est illustré à la figure 4.1(c), et peut servir pour tous les types d'avions simulés. Il complète les commandes de vol.

À la fois le gouvernail et le palonnier peuvent avoir une deuxième configuration afin de s'adapter à la simulation de véhicules terrestres, et tiennent alors lieu respectivement de volant et de pédales de frein et d'accélération. Ces commandes sont donc flexibles, et ne limitent pas le système à la simulation de vol.

Pour l'instant, aucun retour de force n'est présent dans ces commandes car X-Plane ne permettait pas de réaliser un retour de force au moment où les commandes ont été achetées (des développements sont en cours, et peut-être que dans un avenir proche cela sera possible, et rendra la simulation encore plus réaliste). La commande entrant

dans le système de calcul est donc unidirectionnelle, et comporte deux signaux.

4.1.2 Principe du système

Le système de calcul est défini comme le sous-système du simulateur de vol dont le but est de créer, à partir des commandes de vol imprimées par le pilote, les consignes aux autres sous-systèmes (systèmes mécanique et audiovisuel). Le rôle de ce sous-système dans le système global est illustré à la figure 4.2, qui montre tous les sous-systèmes et leurs interactions, particularisés à la solution finale du simulateur. Le but du système de calcul est atteint principalement par traitement informatique des données, et ce système est donc principalement constitué d'un ensemble de composantes logicielles, distribuées sur une architecture matérielle comportant plusieurs ordinateurs selon les besoins, et inclut un ensemble de communications entre ces ordinateurs.

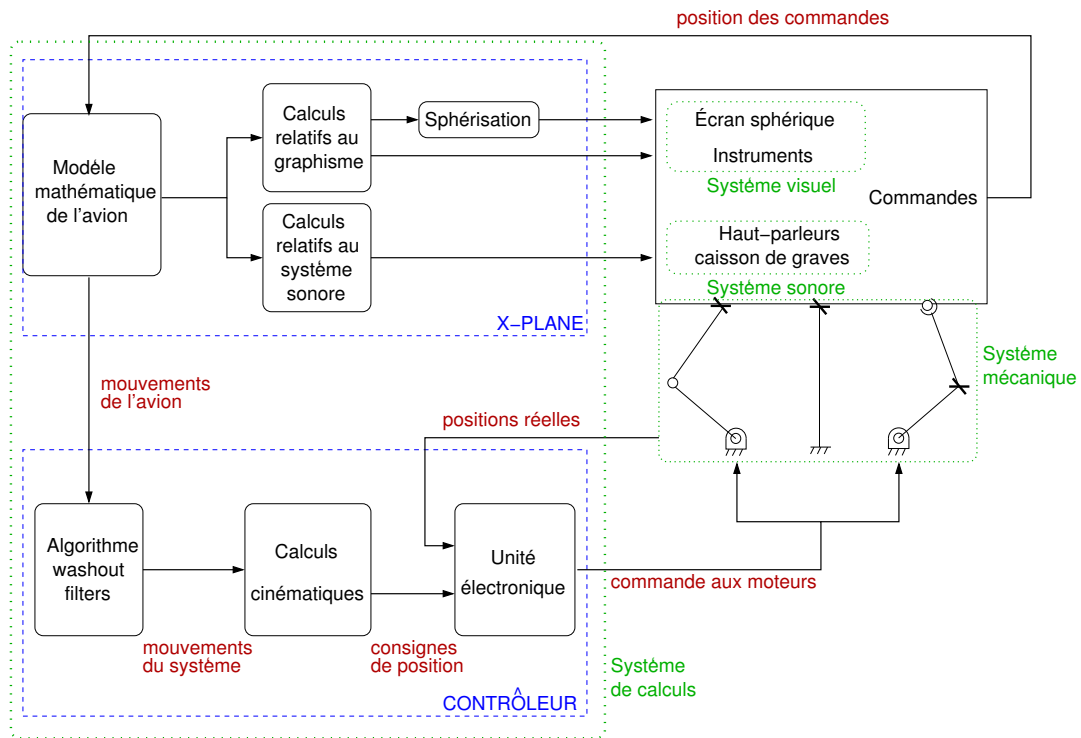


FIG. 4.2 – Schéma des sous-systèmes du simulateur.

Comme on l'a vu au premier chapitre, le sous-système de calcul avait été peu développé lors des recherches antérieures, et seul le choix du logiciel commercial X-

Plane avait été fait lors de la conception générale du simulateur. Ce logiciel correspond à la première partie du système de calcul montré à la figure 4.2, et a pour rôle de réaliser le modèle mathématique de l'avion et les calculs relatifs au son et aux graphismes. Il restait donc pour cette partie à créer les interfaces entre ce logiciel et les autres éléments. Pour cela, la structure de X-Plane autorise d'ajouter ce qu'on appelle des plugins* ¹ (dérivé de l'anglais plug-in, qui signifie branché), qui sont des ensembles de fonctions qui peuvent se greffer au logiciel initial pour en modifier certains aspects, et qui seront décrites à la section 4.3. En particulier, lors de la mise en pratique de l'interface avec le système visuel, la structure originale des calculs liés au graphisme a dû être modifiée, afin de rendre X-Plane compatible avec l'écran sphérique. En fait, un plugin permettant de modifier la vue pour la traiter et la projeter sur l'écran sphérique a été développé conjointement par les deux entreprises Elumens et Laminar Research, et a été ajouté à X-Plane. Le seul inconvénient de cette solution est la projection du tableau de bord : alors que la vue extérieure est dessinée à l'aide d'objets en trois dimensions dans X-Plane, le tableau de bord est dessiné seulement en deux dimensions, et ne peut donc pas être adapté à une projection sur l'écran sphérique. Un deuxième écran a donc été ajouté au système visuel pour dessiner le tableau de bord, et a donc nécessité la deuxième interface entre X-Plane et le système visuel montrée à la figure 4.2.

La seconde partie du système de calcul, dont le rôle est de générer les mouvements du simulateur et de les contrôler, est appelée le contrôleur à la figure 4.2. La structure finale de ce contrôleur peut être divisée en deux parties distinctes :

- La première, qui regroupe les *Washout Filters* et les calculs cinématiques, correspond à la génération des mouvements du simulateur. En effet, son rôle est de créer les consignes à fournir aux actionneurs à partir des données de vol de l'avion fournies par le logiciel X-Plane. C'est une application logicielle qui a été codée en langage C, qui sera détaillée à la section 4.4.

- La seconde, qui correspond à l'unité électronique de la figure 4.2, a pour rôle d'asservir les moteurs, et est réalisée en pratique par une carte de contrôle de la compagnie Delta Tau. Le principe de fonctionnement de cette dernière sera expliqué à la section 4.5. Grâce à cette carte, l'asservissement est facile à appréhender, et toutes les routines de sécurité sont prévues, en cas de problème aux moteurs aussi bien qu'aux consignes, garantissant la sécurité de la personne dans le simulateur.

¹Le symbole * apposé après un mot indique que celui-ci est défini à l'annexe C, qui constitue un glossaire des termes informatiques utilisés dans ce chapitre.

4.2 Architecture matérielle et communications

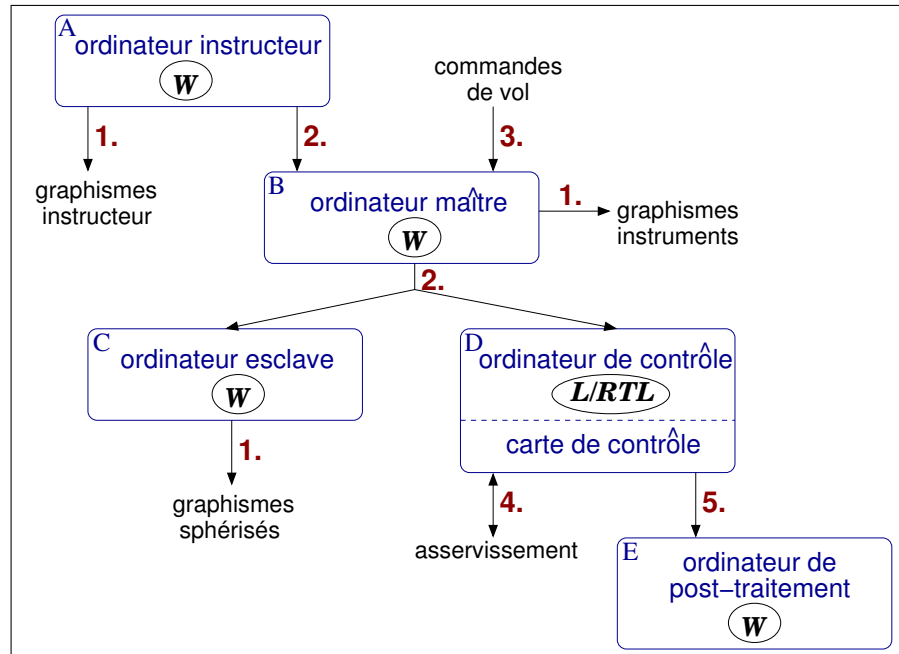


FIG. 4.3 – Schéma de la configuration matérielle du système de calcul.

La configuration matérielle du système de calcul est distribuée sur cinq ordinateurs différents, comme illustré à la figure 4.3. Sur cette figure, chaque bloc marqué d'une lettre majuscule représente un ordinateur. L'ellipse présente au centre de chaque bloc indique le système d'exploitation (SE) qui gère l'utilisation du matériel informatique de cet ordinateur, où le symbole *W* représente le SE Windows, alors que le symbole *L/RTL* représente le SE Linux comportant une partie RTLinux.

Le logiciel X-Plane est distribué sur les ordinateurs A, B et C. L'ordinateur A est optionnel, et correspond à l'ordinateur de l'instructeur, qui peut modifier les conditions de vol de l'avion à l'ordinateur B, en ajoutant des pannes à l'avion (aux moteurs, aux instruments), en modifiant les conditions météorologiques (pluie, turbulences) et plus encore. Sa sortie graphique est une carte de la région où se trouve l'avion, indiquant le trafic aérien. L'ordinateur B est appelé ordinateur maître car c'est l'ordinateur chargé de calculer le modèle dynamique de l'avion dans X-Plane, et de transmettre les données de vol aux ordinateurs C et D. C'est lui qui est aussi chargé de calculer les consignes pour la deuxième interface avec le système visuel, c'est-à-dire l'écran du tableau de bord. Ensuite, l'ordinateur C est l'ordinateur esclave car il est uniquement chargé du traitement de l'image pour la projection sphérique, et ce à partir des données de vol

fournies par B (position et orientation de l'avion). Les ordinateurs B et C sont munis de cartes graphiques très puissantes, possédant deux sorties graphiques afin que la vue extérieure et le tableau de bord soient projetés à la fois dans la cabine du simulateur et dans la station instructeur à l'extérieur.

L'ordinateur D est quant à lui l'ordinateur de contrôle, et correspond au support matériel du contrôleur. Sa structure est spéciale, et conçue pour permettre de réaliser le contrôle des mouvements du simulateur en temps réel, alors que les données lui arrivant ne le sont pas. Ce défi est réalisé en utilisant le SE Linux, dont une partie du noyau est modifiée pour en faire un système d'exploitation temps-réel*. Comme montré à la figure 4.3, il s'ajoute à ceci la carte de contrôle, qui est branchée au port PCI* de D. Cette carte possède son propre processeur, et pourrait donc être autonome, mais elle est ici branchée pour pouvoir partager une partie de sa mémoire vive* avec D, et ainsi utiliser les consignes écrites dans cette mémoire.

Finalement, l'ordinateur E est lui aussi optionnel, et est utilisé pour réaliser le post-traitement des données du contrôle. La carte de contrôle peut lui fournir toutes les données envoyées aux moteurs ainsi que les données retournées par les encodeurs. On peut donc avec E analyser et tracer entre autres les différences entre les consignes aux moteurs et leurs positions réelles, et ainsi connaître les performances de l'asservissement. E est aussi utilisé pour télécharger les programmes d'asservissement dans la carte.

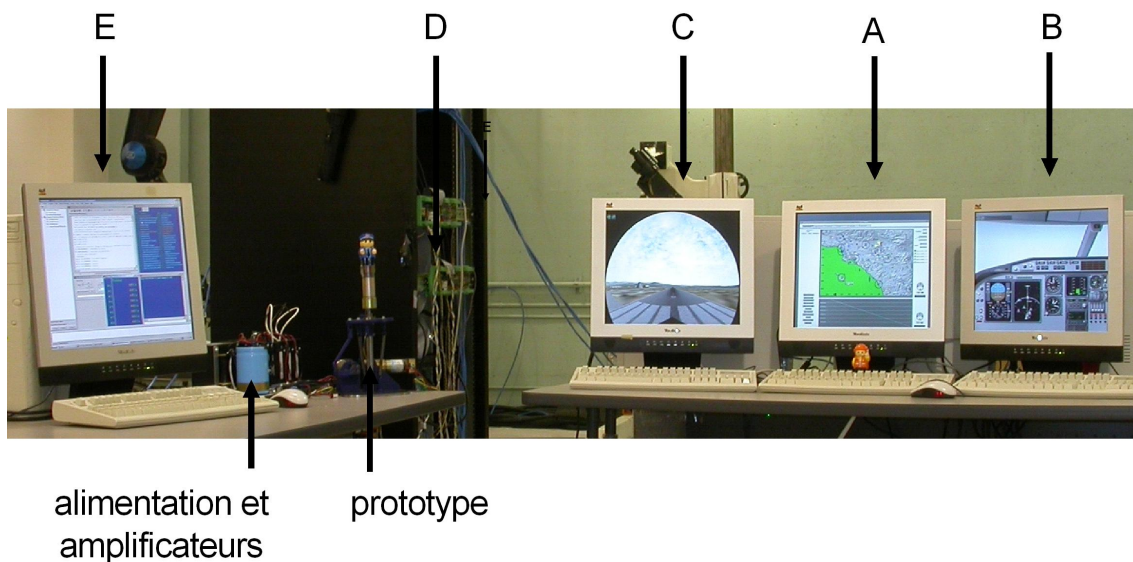


FIG. 4.4 – Photographie du système de calcul.

Le système réel est montré à la figure 4.4, où les différents ordinateurs sont identifiés. Sur cette photo, on voit à droite ce qu'on appelle la station instructeur, qui comprend les ordinateurs A, B, C ainsi que les trois sorties graphiques qui lui sont destinées (la carte de la région et une copie du système visuel présent dans le simulateur). L'ordinateur E est situé complètement à gauche de l'image, et constitue ce qu'on appelle la station du contrôleur, car c'est ici que se place la personne en charge de lancer le système et les simulations, de vérifier que tout se déroule correctement, et d'agir sur le système en cas de problème. Finalement, au milieu, on voit le cabinet de contrôle où est situé l'ordinateur D, ainsi que le système d'amplification de signal et d'alimentation d'énergie, et le prototype de taille réduite du simulateur qui a été construit au laboratoire de robotique de l'Université Laval (ce prototype sera détaillé au prochain chapitre).

Les différentes communications entre les ordinateurs sont indiquées à la figure 4.3 par les flèches marquées d'un chiffre. La plupart d'entre elles suivent des protocoles de communication bien définis et standardisés, et reconnus par les SE sans aucune programmation ni intervention spéciale (ce sont des communications "plug-and-play") :

1. correspond à une communication graphique vers un ou des écrans,
2. correspond à une communication des données par le réseau,
3. correspond à la communication des positions des commandes de vol réalisée par deux ports USB* (un pour chacune des commandes), pour laquelle la seule intervention à réaliser est la calibration des signaux dans X-Plane,
4. correspond à l'entrée/sortie des données vers le système mécanique, où la sortie est l'envoi du signal de consigne aux amplificateurs, et l'entrée est le signal des encodeurs, qui sont gérés par la carte de contrôle,
5. correspond à l'envoi des résultats de l'asservissement par le port série* RS-232 vers l'ordinateur de post-traitement afin de tracer et analyser les trajectoires, et est géré par le logiciel de post-traitement.

Parmi ces différents moyens de communication, seul celui indiqué par le chiffre 2. a été complètement développé pour les besoins de l'application, et correspond à la communication de données par messagerie via le réseau. Cette communication se fait en suivant un protocole TCP/IP*, permettant d'envoyer des messages entre applications jusqu'aux interfaces de connexion appelées des sockets*. La méthodologie pour programmer des sockets et envoyer des messages suivant ce protocole sont détaillées dans la référence [25], qui a servi de base à la programmation en C de ce système de communications.

Maintenant que l'architecture matérielle et les communications du système de calcul ont été revues, il reste à détailler quelles sont les composantes logicielles qui réalisent les calculs. Trois composantes logicielles principales sont présentes, c'est à dire le logiciel commercial X-Plane, le logiciel de génération des mouvements du simulateur, et finalement le logiciel de contrôle, et sont décrites dans les trois prochaines sections.

4.3 Logiciel X-Plane

La version de ce logiciel utilisée pour le simulateur est la version X-Plane 7.30, pour laquelle la documentation [26] donne une révision complète et détaillée de toutes les possibilités d'utilisation. Comme beaucoup de logiciels de simulation de vol, X-Plane permet à la fois de calculer le modèle mathématique de différents avions, et de dessiner l'évolution de la vue extérieure de l'avion et des indicateurs des instruments de vol. Les principales raisons qui ont permis de choisir ce logiciel plutôt qu'un autre sont les suivantes :

- X-Plane utilise un modèle dynamique des avions très réaliste, et a déjà été utilisé dans des simulateurs de vol commerciaux accrédités par la FAA (par exemple les simulateurs MOTUS de l'entreprise Fidelity Flight Simulation Inc. utilisent X-Plane 6.12 et ont une accréditation de niveau 2, dont les détails sont disponibles dans [27]),
- X-Plane possède un module instructeur de vol, qui permet à un instructeur de modifier les conditions extérieures du vol lors d'une simulation, et donc permet un entraînement du pilote aux situations d'urgence, ce qui est un des buts fixés pour ce simulateur,
- Finalement, même si on n'a pas accès au code source du logiciel, il est très facile de modifier les entrées et sorties, et d'avoir accès à de nombreuses variables. Cette possibilité est offerte par l'ajout de plugins au logiciel. Comme on l'a déjà vu, cette possibilité a rendu possible l'interface entre X-Plane et l'écran sphérique, mais permet aussi bien d'autres développements, expliqués ci-après.

Un plugin est un ensemble de fonctions compilées sous forme de DLL* (Dynamically Linked Library, c'est à dire une bibliothèque de fonctions qui sont liées au code existant de façon dynamique, lorsque le programme principal est exécuté), permettant une extension des capacités du logiciel ainsi qu'un accès à certaines de ses variables. Un plugin est différent d'un programme classique car il ne comporte pas de fonction principale, et ne peut donc pas être utilisé seul. Il doit être utilisé par le logiciel X-Plane, qui appelle en temps voulu les fonctions déclarées et définies par le plugin, et utilise ainsi le code fourni par ce plugin.

Les fonctions qui peuvent être programmées dans un plugin sont régies par le SDK* (Software Development Kit) détaillé dans [28]. Entre autre, cette documentation fournit la liste des variables qui peuvent être lues et/ou modifiées par un plugin (leur nom, leur forme, leur statut). En résumé, le SDK fournit un cadre de développement de logiciel à ajouter à X-Plane, sans pour autant permettre de modifier ni d'avoir accès au code source de X-Plane. Les différents plugins qui ont été développés pour le système de calcul ont été programmés en langage C et compilés grâce à Microsoft Visual Studio 6. Les détails de ces plugins ainsi que leurs rôles respectifs sont donnés en annexe D.1.

Ces différentes fonctionnalités rendent le logiciel très flexible d'utilisation, et permettent une reconfiguration aisée du système de calcul. En effet, en modifiant facilement les entrées/sorties du logiciel, on peut aussi bien ajouter un autre modèle mathématique d'avion et laisser X-Plane seulement réaliser les calculs graphiques (les entreprises d'aviation ont souvent leur propre modèle dynamique pour leur avion), qu'inversement utiliser X-Plane uniquement pour son modèle d'avions, couplé à un autre logiciel de calculs graphiques.

4.4 Génération des mouvements du simulateur

La seconde principale composante logicielle du système de calcul est celle de la génération des mouvements du simulateur, à partir des données de vol fournies par X-Plane. Cette composante logicielle est exécutée sur l'ordinateur D, qui comme on l'a vu est géré par un système d'exploitation Linux et RTLinux. Ce SE sera donc d'abord expliqué ici, puis on verra comment sa structure a été utilisée pour développer l'algorithme de génération de mouvements. Finalement, les *Washout Filters*, qui constituent

la partie la plus importante de cet algorithme, seront détaillés.

4.4.1 Principe du système d'exploitation

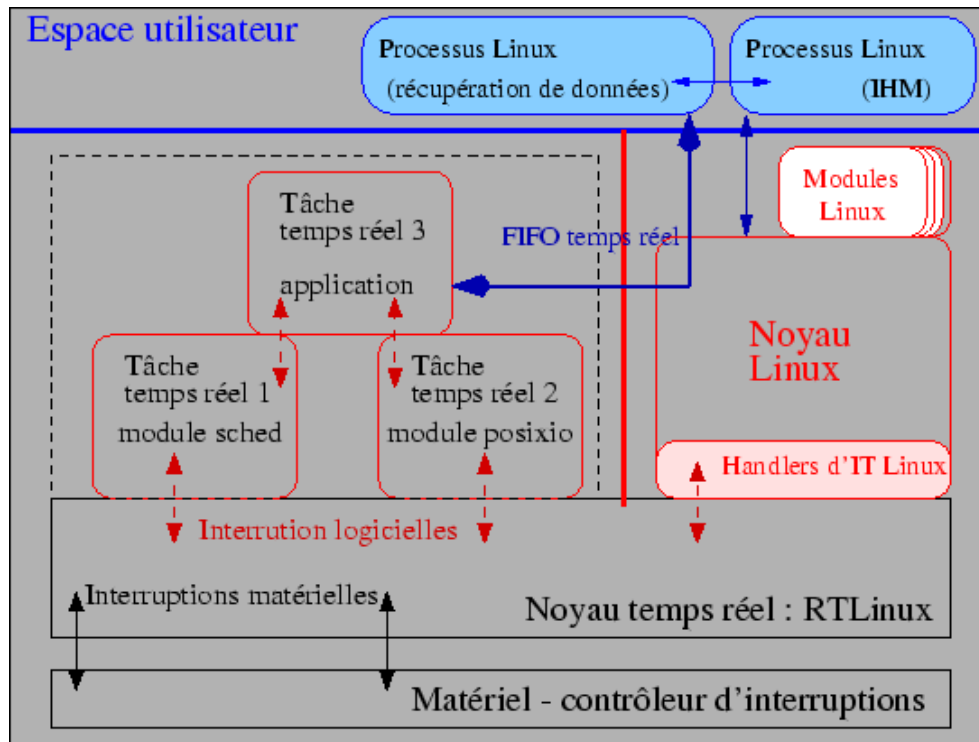


FIG. 4.5 – Schéma de l'organisation du système d'exploitation RTLinux.

Le défi de la génération des mouvements du simulateur résidait dans la génération de consignes en temps réel pour le système mécanique, sans aucune interruption possible, et ce à partir de données provenant du logiciel X-Plane, dont on ne contrôle pas la fréquence des calculs, et qui est exécuté dans un environnement Windows qui n'est pas temps-réel*. La solution à ce défi est venue en partie de la structure même du système d'exploitation de l'ordinateur dans lequel le logiciel est exécuté. Ce système d'exploitation, employé dans l'ordinateur de contrôle D, est un SE Linux dont une partie a été rendue temps-réel par RTLinux. Le principe de ce SE est illustré à la figure 4.5.

Le noyau* Linux n'est pas un noyau temps-réel, il n'était donc pas possible à partir de ce SE de réaliser le but fixé. Cependant, ce noyau peut être modifié en y ajoutant un noyau temps réel auxiliaire RTLinux. En fait, RTlinux est composé d'une mise à

jour à appliquer au noyau Linux afin d'assurer la compatibilité des deux noyaux, et d'un ensemble de modules* constituant le noyau temps-réel auxiliaire. Ainsi, le noyau classique Linux cohabite avec un noyau auxiliaire basé sur un vrai ordonnanceur temps-réel à priorités fixées. Les tâches temps réel sont gérées par ce noyau auxiliaire et le traitement des autres tâches est délégué au noyau Linux, lui-même considéré comme une tâche de plus faible priorité par le noyau RTLinux.

Un programme RTLinux doit être conforme à la structure d'un module* du noyau Linux, et est exécuté directement dans le noyau RTLinux, tout comme l'ensemble des modules définissant les règles de ce noyau. Par exemple, à la figure 4.5, les modules sched et posixio régissent respectivement les priorités des tâches temps-réel et les définitions des entrées/sorties selon les spécifications Posix, et font partie du noyau RTLinux, alors que le module développé par l'utilisateur est appelé application. La nécessité de programmer les modules temps-réel dans l'espace du noyau et non dans l'espace utilisateur est une contrainte non négligeable car elle a de fortes implications sur la facilité de programmation et les outils disponibles (voir [29] pour une explication détaillée de la programmation de modules Linux).

Pour développer un logiciel contenant à la fois des tâches temps-réel et non temps-réel comme c'est le cas pour le système de calcul, on doit donc faire appel à un processus Linux exécuté de façon classique dans l'espace utilisateur et à un module exécuté dans le noyau RTLinux, comme illustré à la figure 4.5. De par cette structure, il est nécessaire de mettre en place un système de communication entre les deux éléments de l'application. RTLinux propose donc des systèmes de communication de type FIFO* ou des mémoires partagées entre le noyau RTLinux et l'espace utilisateur.

4.4.2 L'algorithme de génération des mouvements

L'algorithme de génération des mouvements a été développé en exploitant le plus possible cette structure de système d'exploitation. En effet, pour cette application, la plupart des tâches ne sont pas temps-réel (et ne peuvent pas le devenir car on ne contrôle pas les contraintes temporelles de X-Plane), alors que la sortie doit l'être. Ceci permet une séparation logique des tâches entre l'espace utilisateur de Linux et le noyau RTLinux.

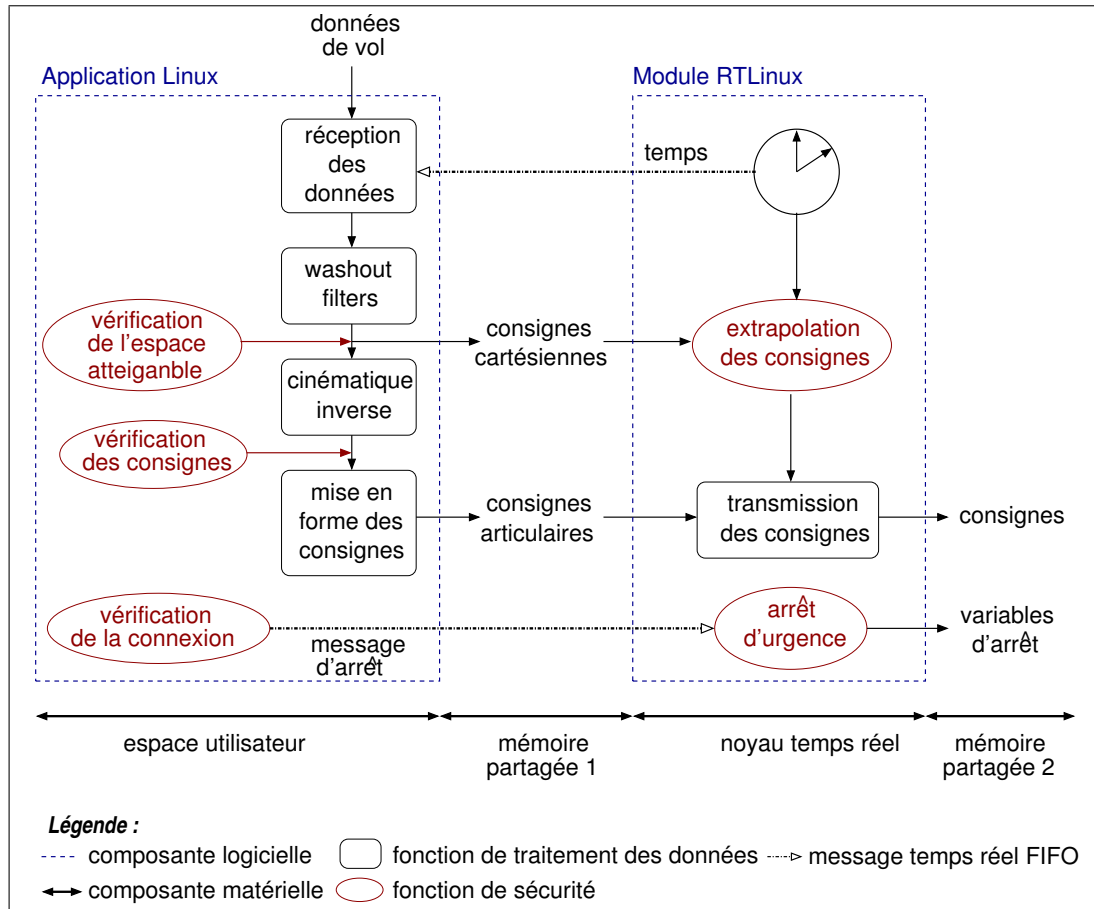


FIG. 4.6 – Schéma de l'algorithme de génération des mouvements.

La figure 4.6 schématise la structure de l'algorithme de génération des mouvements. Une première composante logicielle, appelée Application Linux, correspond à la partie logicielle de Linux, et est donc exécutée dans l'espace utilisateur. Cette composante réalise toutes les tâches n'impliquant pas d'opérations temps-réel, et réalise donc à la fois la réception des données fournies par X-Plane et leur traitement jusqu'à la mise en forme des consignes pour la commande. De plus, les routines de sécurité de cette application consistent d'abord à vérifier que le mécanisme reste toujours dans l'espace atteignable défini au chapitre 2, puis à vérifier et limiter au besoin les consignes articulaires en fonction des limites du système en positions, vitesses, et accélérations. Enfin, la dernière routine de sécurité est celle qui est chargée à chaque cycle de vérifier l'état de la connexion avec X-Plane, et d'envoyer un message d'arrêt en cas de changement brutal des coordonnées de l'avion, d'accident de l'avion, ou tout simplement d'arrêt de la connexion avec X-Plane. Les consignes cartésiennes et articulaires sont placées dans la mémoire partagée 1, accessible à partir du noyau RTLinux. La liste des fichiers

compilés pour former cette application, ainsi que les fonctions qu'ils contiennent, sont données à l'annexe D.2.1.

L'autre composante logicielle principale est le Module RTLinux, dont la tâche principale est de transmettre en temps-réel les consignes à la mémoire partagée 2. Cette tâche s'est donc vu attribué la plus grande priorité accessible, et est secondée comme on le voit à la figure 4.6 par la tâche d'extrapolation des consignes. La contrainte temporelle principale qui régit le fonctionnement de ce module est que la mémoire partagée 2 doit recevoir au moins une consigne toutes les 12 ms (cette contrainte garantit que la file des consignes fournie à la carte de contrôle ne sera jamais vide). En moyenne, X-Plane fournit des données toutes les 6 ms, mais l'écart temporel entre deux données varie, et dépasse occasionnellement cette valeur critique de 12 ms. Le premier rôle de la fonction d'extrapolation des consignes est donc, en se basant sur le chronomètre très précis de RTLinux (qui a une résolution de une nanoseconde), de calculer les écarts temporels entre les consignes. Si aucune nouvelle consigne n'est écrite dans la mémoire partagée 1 alors que 10 ms se sont écoulées depuis la dernière consigne, alors cette fonction calcule ce qu'on appelle une consigne virtuelle, qui correspond à l'extrapolation des deux dernières consignes. Finalement, cette consigne virtuelle pourra éventuellement être transmise à la place d'une consigne calculée à partir des données de X-Plane si la contrainte temporelle est atteinte.

Par ailleurs, la dernière tâche de ce module est celle d'arrêt d'urgence, qui transmet à la mémoire partagée 2 tout message nécessitant l'arrêt immédiat de la simulation. La liste des fichiers compilés pour former le module, ainsi que les fonctions qu'ils contiennent, sont données à l'annexe D.2.2.

4.4.3 *Washout Filters*

La fonction la plus importante pour la génération des mouvements du simulateur est celle des *Washout Filters* (WF), car elle doit déterminer les mouvements à appliquer au simulateur en fonction des mouvements de l'avion que l'on désire simuler. Le but principal des WF est donc de produire au pilote des sensations de mouvement les plus conformes possibles à celles ressenties dans l'avion, tout en respectant et exploitant au mieux l'espace atteignable de la plate-forme de mouvement.

Il existe de très nombreuses variations aux WF, la plus connue et la plus utilisée dans les systèmes commerciaux étant la forme dite classique [31, 32]. Pour l'application du simulateur de vol sphérique, l'algorithme utilisé correspond à une configuration classique, modifiée selon les résultats de [30] traitant de l'adaptation des filtres classiques à des systèmes sphériques à trois degrés de liberté. Les principes de base de cet algorithme sont maintenant présentés.

D'abord, et de façon générale à tous les simulateurs de mouvements, une des tâches les plus contraignantes est la simulation rigoureuse d'accélération linéaires de basses fréquences, car celles-ci ont tendance à entraîner très rapidement la plate-forme hors de son espace atteignable. Par exemple, une accélération de 1 m/s^2 selon l'axe longitudinal X et soutenue pendant 5 secondes demanderait un déplacement total de 12.5 mètres ! Partant du principe que l'on cherche à reproduire les sensations de mouvement plutôt que les mouvements eux mêmes, on peut exploiter l'accélération qui agit déjà sur le système : l'accélération gravitationnelle.

En fait, les mouvements linéaires ressentis par un être humain sont les forces spécifiques (\mathbf{f}), c'est-à-dire la différence entre le vecteur d'accélération (\mathbf{a}) et celui de l'accélération gravitationnelle (\mathbf{g}), soit : $\mathbf{f} = \mathbf{a} - \mathbf{g}$. Alors, pour l'exemple donné, il est possible d'incliner lentement la plate-forme d'un angle de 6 degrés autour de son axe Y , tout en conservant l'image projetée à l'intérieur fidèle à des conditions de vol horizontal. Dans cette situation le pilote sentira une composante X de force spécifique égale à $9.81 \cdot \sin(6^\circ) \simeq 1 \text{ m/s}^2$. En fait, même si à ce moment la composante en Z des forces spécifiques sera réduite à 9.75 m/s^2 , cette variation peut passer inaperçue dans la mesure où la vitesse de l'inclinaison est demeurée sous la valeur seuil de perception (qui est évaluée entre 0.035 et 0.05 rad/s selon les études [31, 33]). Cet artifice, connu sous le nom de verticale apparente (*Tilt-coordination*), s'avère donc très efficace, voire essentiel, pour simuler les accélérations linéaires de basses fréquences. De plus, pour un simulateur de vol sphérique, il donne la possibilité de créer des sensations de translations alors que la plate-forme ne réalise que des rotations, et permet donc au système de rivaliser avec les plates-formes à six degrés de liberté du point de vue des sensations de mouvement.

Le deuxième principe de base qui est utilisé dans les WF est d'adapter les mouvements créés à l'espace atteignable de la plate-forme, ce qui est réalisé en appliquant à la fois des gains et des filtres fréquentiels aux mouvements de l'avion. Les filtres ont pour effet de créer des mouvements qui ont tendance à ramener la plate-forme vers la position centrale de son espace atteignable (le point neutre). L'ajustement des gains,

des fréquences de coupures et de l'ordre des filtres dépend alors de l'espace atteignable de la plate-forme, et dans le cas du simulateur de vol sphérique c'est cet ajustement qui va permettre de tirer avantage du grand espace atteignable en rotation pour créer de bonnes sensations de mouvement.

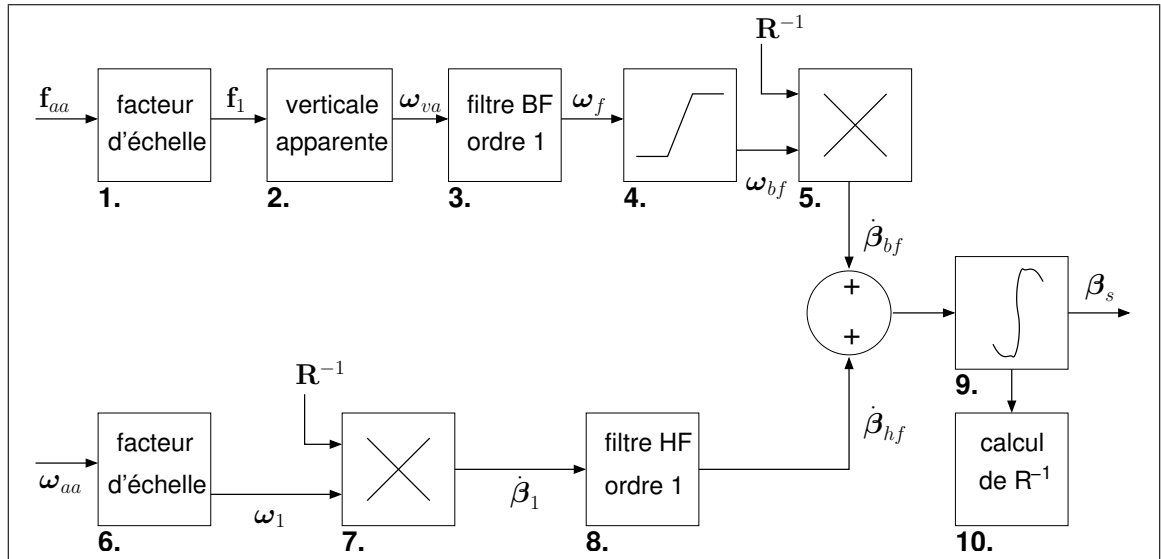


FIG. 4.7 – Schéma des *Washout Filters*.

Plus précisément, la figure 4.7 illustre schématiquement les principes de l'algorithme des WF pour ce simulateur. Il reçoit en entrée les forces spécifiques \mathbf{f}_{aa} dans l'avion évaluées à la tête du pilote, et les vitesses angulaires $\boldsymbol{\omega}_{aa}$ de l'avion (toutes deux exprimées dans un repère lié à l'avion) et produit à la sortie les angles d'Euler $\boldsymbol{\beta}_s$ que le simulateur doit adopter.

Comme on peut le voir sur la figure, l'algorithme des WF se divise principalement selon 2 axes parallèles. L'axe supérieur, constitué des blocs 1 à 5, est destiné à l'artifice de la verticale apparente. Il reçoit les valeurs de forces spécifiques \mathbf{f}_{aa} , et diminue d'abord au bloc 1 leur amplitude par un facteur d'échelle qui est un gain d'une valeur située entre 0 et 1. Ensuite, la verticale apparente en elle-même est appliquée au bloc 2, et crée à partir des forces spécifiques à l'échelle \mathbf{f}_1 les vitesses de rotations $\boldsymbol{\omega}_{va}$. Le principe de la transformation est de créer ces vitesses de rotation à partir du produit vectoriel entre la force spécifique \mathbf{f}_1 et l'accélération gravitationnelle dans le simulateur \mathbf{g}_s , selon l'équation :

$$\boldsymbol{\omega}_{va} = \frac{-\mathbf{f}_1 \times \mathbf{g}_s}{\|\mathbf{f}_1\|} \quad (4.1)$$

Étant donné que le produit vectoriel tend vers zéro avec l'angle entre les deux vecteurs, cette vitesse de rotation a pour effet de tendre à aligner \mathbf{f}_1 avec $-\mathbf{g}_s$ sans dépassement, et ainsi on tendra à faire ressentir au pilote dans le simulateur la force spécifique \mathbf{f}_1 qui est celle désirée.

On passe alors $\boldsymbol{\omega}_{va}$ dans un filtre passe-bas d'ordre 1 au bloc 3, et on limite le vecteur de vitesse de rotation ainsi obtenu $\boldsymbol{\omega}_{bf}$ à 0.05 rad/s pour qu'il reste sous le seuil de perception du pilote grâce au bloc de saturation 4. Enfin, on transforme cette vitesse de rotation en la dérivée des angles d'Euler correspondants $\dot{\boldsymbol{\beta}}_{bf}$, à l'aide de la matrice \mathbf{R}^{-1} , qui est l'inverse de la matrice \mathbf{R} définissant la relation entre les vitesses de rotation et les dérivées des angles d'Euler pour la convention utilisée $\boldsymbol{\omega} = \mathbf{R}\dot{\boldsymbol{\beta}}$ (se référer à la section 2.3.1 pour l'explication de la méthode d'obtention de cette matrice).

L'axe inférieur, composé des blocs 6 à 8, fonctionne de façon très similaire au premier mais est dédié au traitement des vitesses angulaires. Les vitesses angulaires de l'avion $\boldsymbol{\omega}_{aa}$ sont d'abord mises à l'échelle au bloc 6 de la même façon qu'en 1, puis sont transformées en dérivées d'angles d'Euler $\dot{\boldsymbol{\beta}}_1$ au bloc 7. On ne conserve finalement que les hautes fréquences de $\dot{\boldsymbol{\beta}}_1$ en passant ce vecteur dans un filtre passe-haut d'ordre 1 au bloc 8 de manière à s'assurer que le simulateur ne s'éloigne pas trop de son point central.

Finalement, la somme des dérivées d'angles d'Euler $\dot{\boldsymbol{\beta}}_{bf}$ et $\dot{\boldsymbol{\beta}}_{hf}$ est intégrée au bloc 9 pour obtenir les angles d'Euler à prescrire à la plate-forme $\boldsymbol{\beta}_s$, et la matrice \mathbf{R}^{-1} peut alors être calculée au bloc 10 pour le prochain cycle de calculs.

Comme montré à la figure 4.6, ces angles d'Euler $\boldsymbol{\beta}_s$ seront ensuite l'entrée de la fonction de cinématique inverse, où le PGI sera résolu selon les équations de la sections 2.2.1 pour déterminer les consignes articulaires à donner aux moteurs.

4.5 Contrôle

Le dernier élément logiciel du système de calcul est celui qui réalise l'asservissement des moteurs. Il est exécuté sur la carte TURBO PMAC2 PCI fournie par l'entreprise Delta Tau Data Systems Inc, illustrée à la figure 4.8.

Le guide de l'utilisateur et le guide du logiciel sont disponibles respectivement dans



FIG. 4.8 – Carte de contrôle de Delta Tau.

les documents [34] et [35]. Les raisons qui ont guidé le choix de cette carte ont été principalement la sécurité et la facilité d’appréhension. D’abord, la sécurité est un facteur très important dans cette application, car une personne sera présente dans le simulateur, et on doit donc pouvoir garantir sa sécurité. Or la carte est vraiment complète de ce point de vue là, et le logiciel permet de nombreuses routines de sécurité et d’arrêts d’urgence. Aussi, comme c’est un produit commercial utilisé en industrie, il a été testé à maintes reprises, et est sûrement plus fiable qu’un contrôleur que l’on aurait développé au laboratoire (ce qui était une autre option envisagée, et déjà mise en œuvre dans le passé). D’autre part, il est vraiment facile d’appréhender les différentes possibilités du logiciel, et seules les commandes de haut niveau sont accessibles à l’utilisateur. Il est donc aussi aisé de programmer des mouvements des moteurs, avec des commandes et une syntaxe de programmation très simples. Finalement, cette carte permet de donner les consignes en temps réel grâce à la mémoire vive partagée avec l’ordinateur de contrôle, ce qui était un pré-requis à son utilisation pour cette application.

Cependant, cette carte possède certains inconvénients qui découlent directement des avantages cités ci-dessus, car la contrepartie de ce système clef en main est qu’il laisse moins de liberté à l’utilisateur, et on ne peut pas sortir facilement du cadre pour lequel le logiciel a été développé. L’inconvénient majeur qui est apparu est que le logiciel permet seulement de contrôler les moteurs à l’aide de commandes en position, et non en couple. La conséquence directe de cette limitation est que le modèle dynamique du système n’a pas pu être inclus dans le contrôleur. La boucle de contrôle est donc seulement régie par une loi PID (Proportionnel-Intégral-Dérivé) classique en position, et les tests de modèles dynamiques proposés à la section 3.5.3 n’ont pour l’instant pas pu être réalisés.

Pour réaliser cette boucle de contrôle, un programme est téléchargé dans la carte. Il permet d'abord d'allouer des variables par leur adresse dans la mémoire vive partagée où sont écrites les consignes. Ensuite, lors de la simulation, il compare ces consignes à la position réelle des moteurs fournie par les encodeurs pour former la commande suivant la loi de contrôle PID dont les coefficients auront été préalablement calibrés. La boucle d'asservissement des moteurs est réalisée à une fréquence de 2260 Hz. Comme les consignes sont disponibles à une fréquence variable qui est en moyenne de 160 Hz environs, il faut réaliser une interpolation des consignes pour obtenir les commandes. Cette interpolation est rendue possible par le mode d'asservissement appelé PVT dans le logiciel de la carte, qui signifie Position, Vitesse, Temps. En utilisant ce mode d'asservissement, on fournit comme consigne la position, la vitesse et le pas de temps depuis la dernière consigne (qui peut varier d'une consigne à l'autre), et le logiciel se charge d'interpoler entre ces points suivant le seul polynôme d'ordre 3 qui réalise exactement les positions et vitesses désirées aux points de consignes.

Finalement, le logiciel de traitement qui est installé sur l'ordinateur E est appelé PeWin32 Pro, et sa documentation est disponible dans [36]. Il contient à la fois un éditeur de programmes, qui permet de développer les programmes de contrôle et de les télécharger dans la carte, un espace de travail qui permet de créer et gérer des projets représentant l'ensemble des configurations de la carte pour une application, et un terminal qui permet d'envoyer des commandes et de recevoir des messages d'erreur de la carte. On peut donc grâce à ce logiciel traiter les communications avec la carte qui ne sont pas temps réel. Un des accessoires de ce logiciel est aussi le module de post-traitement (PMACPlot), permettant de télécharger après une simulation les données du contrôle, et les tracer dans une fenêtre ou dans Matlab.

Ceci complète l'architecture du système de calcul, qui permet de contrôler en temps-réel les mouvements des moteurs de la plate-forme, et ce à partir des commandes données par le pilote. Les performances de ce système seront discutées dans le prochain chapitre, qui vise la mise en pratique du système au complet et son évaluation grâce au prototype qui a été construit au laboratoire de robotique de l'Université Laval.

Chapitre 5

Résultats expérimentaux

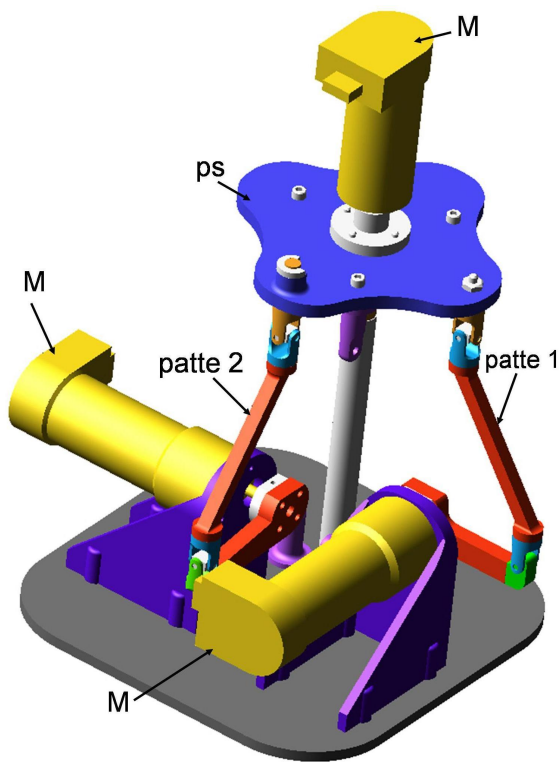
Dans ce chapitre, les aspects pratiques des systèmes mécanique et de calcul sont mis en valeur grâce à l'utilisation d'un prototype à échelle réduite du simulateur. De plus, grâce à ce prototype, les sensations de mouvement perçues lors de manœuvres d'avions simulées par X-Plane peuvent être évaluées expérimentalement, et leur qualité peut être comparée à celle obtenue avec une plate-forme classique.

Dans un premier temps, le prototype construit sera décrit. Ensuite, on verra la méthodologie utilisée pour évaluer les résultats des sensations de mouvement qu'il produit. Finalement, quelques exemples graphiques décrivant les sensations produites pour des manœuvres de deux avions différents seront discutés et analysés.

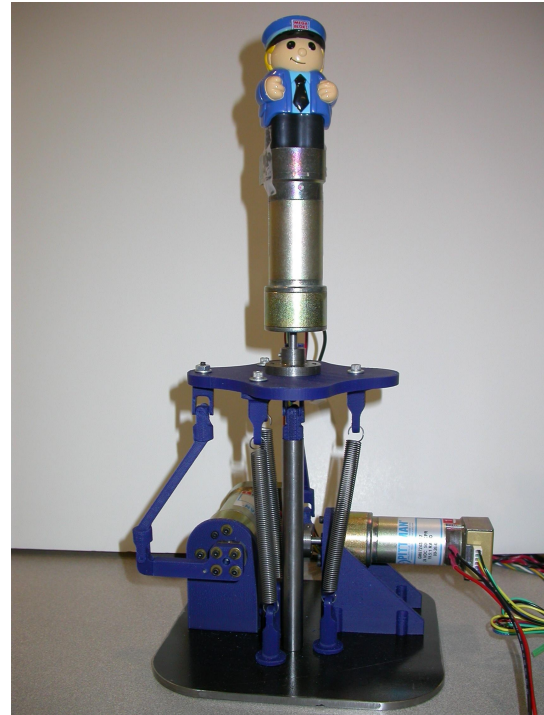
5.1 Le prototype

5.1.1 Système mécanique

Un prototype à échelle $\frac{1}{7}^e$ a été construit au laboratoire de robotique en plastique grâce au prototypage rapide [38]. La figure 5.1(a) illustre le modèle *ADAMS* du prototype, et la figure 5.1(b) montre une photographie du système construit. Les pièces en plastique ont été conçues et réalisées par Simon Foucault et Thierry Laliberté.



(a) Modèle ADAMS.



(b) Photographie.

FIG. 5.1 – Architecture mécanique du prototype.

Comme on peut le voir à la figure 5.1(a), l'architecture mécanique est similaire à celle du simulateur à taille réelle qui était illustrée à la figure 3.5. Les trois moteurs sont désignés par la lettre *M*, la plaque support est notée *ps*, et les pattes 1 et 2 ont la même géométrie réduite sept fois. La seule différence entre les deux est que le prototype ne possède pas de patte 3. En effet, le prototype n'inclut pas le système visuel ni le pilote,

et donc la cabine est composée uniquement du moteur 3. Ce moteur n'assure donc plus la fonction d'équilibrage du centre de gravité de la cabine, et de plus il n'y a aucun encombrement au centre de la plaque support. Il a donc pu être placé directement sur l'axe vertical de rotation de la cabine. Cette architecture est équivalente à celle du simulateur à taille réelle du point de vue cinématique, car comme on l'avait vu à la section 2.2.1, l'angle réalisé par le moteur θ_3 était égal à l'angle d'Euler γ réalisé au niveau de l'axe de rotation par la cabine. Ainsi, les variables cinématiques seront les mêmes, et seule la dynamique du prototype sera simplifiée par cette modification de l'architecture.

Le modèle dynamique a donc été modifié en tenant compte du fait qu'il n'y a pas de patte 3, et en changeant les masses et inerties des pièces. Encore une fois, le modèle a été validé en comparant les couples avec ceux obtenus par le modèle *ADAMS* illustré à la figure 5.1(a) pour les trajectoires test définies au chapitre 3. Grâce au modèle dynamique, les ressorts ont été calculés de la même façon qu'à la section 3.4. Étant donné que pour ce prototype les masses et inerties sont beaucoup plus faibles que pour le système à taille réelle, les ressorts possèdent une taille et une raideur standard, et ils ont donc pu être commandés sur catalogue. Leurs caractéristiques sont données en annexe B.2. Le modèle dynamique du prototype, incluant l'équilibrage statique partiel dû aux ressorts décrits ci-dessus, a aussi permis d'évaluer les couples nécessaires aux actionneurs, et de choisir les moteurs. Ce sont des moteurs DC couplés à des réducteurs de rapport 1 : 19.5, et munis d'encodeurs relatifs. Ils ont été fournis par la compagnie Pittman. Les caractéristiques de chacune des composantes de ces moteurs, ainsi que de leur assemblage sont données en annexe B.3.

5.1.2 Système de calcul

Pour le système de calcul, l'architecture décrite au chapitre précédent a été utilisée sans aucune modification. Les temps de calcul nécessaires aux différentes parties ont pu être estimés, ou majorés dans les cas où la mesure du temps était imprécise ou inaccessible. Le schéma 5.2 résume ces données.

On peut remarquer tout d'abord que le logiciel de génération des mouvements, incluant le module *RTL*Linux, permet de créer les consignes aux moteurs en un temps

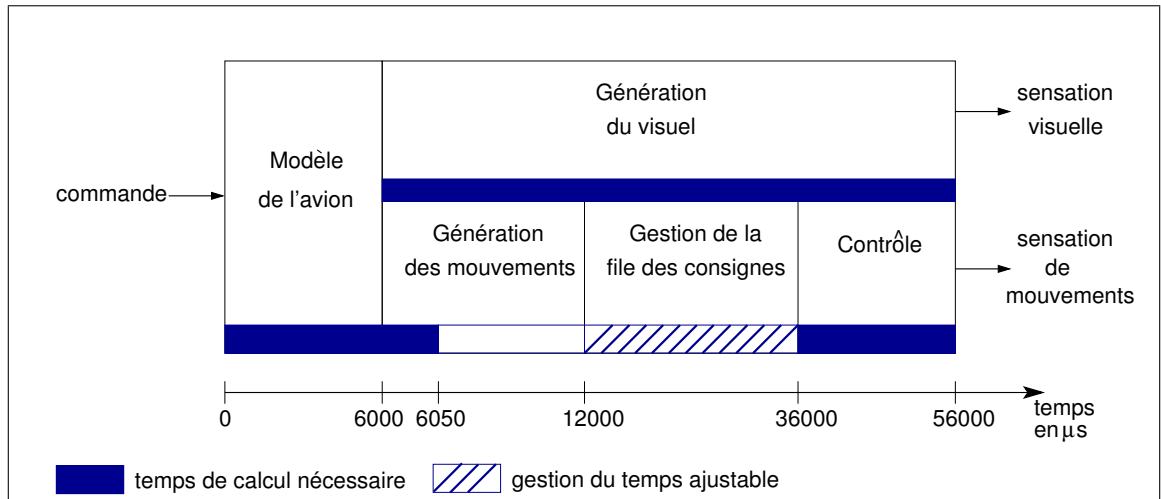


FIG. 5.2 – Évaluation des temps de calculs.

extrêmement faible. Ce temps a été mesuré en donnant comme entrée au logiciel des fichiers de données, et en mesurant grâce à l’horloge temps-réel de RTLinux le temps nécessaire au traitement de ces données, pour différents nombres de données. Le temps moyen observé de calcul est de $50 \mu s$! Cela signifie, comme on le voit à la figure 5.2, que la fréquence des calculs de ce logiciel est limitée par le modèle de l’avion de X-Plane, qui fournit, en incluant le temps d’envoi des données, une donnée de vol toutes les 6 ms en moyenne. Le logiciel de génération de mouvements est donc sous-exploité en terme de puissance de calcul, et de nombreux calculs pourraient être ajoutés sans pour autant modifier les performances du système général (comme le modèle dynamique du simulateur par exemple).

Ensuite, on voit que la composante de génération du visuel est celle qui limite la période globale de traitement d’une donnée, car son temps de calculs est de 50 ms environ. Cela est dû principalement à la fonction de traitement de l’image réalisé dans X-Plane afin de projeter l’image sur l’écran sphérique, qui nécessite une grande puissance de calcul. On peut alors ajuster la taille de la file des consignes aux moteurs en conséquence afin de synchroniser les sensations visuelles et de mouvement données au pilote, et éviter le “mal du simulateur”. Comme illustré à la figure 5.2, la file des consignes a été ajustée à 24 ms pour réaliser cette synchronisation, mais cette valeur pourra être modifiée en réalisant des tests de sensations avec des pilotes lorsque le système réel sera construit.

Finalement, les performances du système de calcul sont conformes au cahier des

charges, car on obtient un délai entre la commande les sensations correspondantes d'environ 56 ms, alors que dans un simulateur de vol en général le délai acceptable ne doit pas dépasser 150 ms [39], et peut varier quelque peu selon la dynamique de l'avion simulé (le délai est restreint par exemple pour un avion rapide ayant des mouvements brusques, pour lequel le délai sera plus difficilement imputable à l'inertie de l'appareil). Le point le plus critique pour les performances du système de calcul est la fréquence de rafraîchissement de la vue extérieure sur l'écran sphérique. En effet, on obtient une fréquence un peu en dessous de 20 Hz, ce qui est considéré comme la limite minimale de confort visuel humain. Cette fréquence reste acceptable pour un simulateur de vol, car les objets vus à l'extérieur de l'avion sont la plupart du temps éloignés. Cependant, cela pourrait devenir un problème lors de simulations de voiture par exemple, pour lesquelles les objets à l'extérieur de la cabine sont proches du pilote, et se déplacent rapidement.

5.2 Méthodologie d'évaluation

L'un des objectifs du développement du simulateur sphérique au laboratoire de robotique était de concevoir un simulateur pouvant donner des sensations de mouvement au pilote de qualité comparable à celles produites par un simulateur classique à six degrés de liberté. Une étude basée entièrement sur des simulations avait été réalisée à la première étape du projet, et avait fourni des résultats très optimistes quant à cet objectif, pour des architectures sphériques générales. Maintenant que la conception du système de calcul et de l'architecture mécanique sont dans leur phase finale, et qu'un prototype a été construit, il est possible d'évaluer de manière plus précise les performances de sensations de mouvement produites par ce simulateur, en se basant sur les mouvements réels du prototype obtenus lors de simulations de manœuvres d'avions.

Pour cela, une méthodologie de production des résultats a été mise en place. En premier lieu, comme le critère principal de l'évaluation porte sur les sensations humaines des mouvements, un modèle de la perception des mouvements par l'humain a été utilisé, et est décrit à la section 5.2.1. Ensuite, cette évaluation est faite principalement en comparant les sensations produites par le prototype avec celles qui seraient produites par une plate-forme classique à six degrés de liberté, et bien sûr celles ressenties dans l'avion simulé. On a donc simulé une plate-forme classique, et mis en place une

procédure de traitement des résultats afin de comparer les sensations de mouvement obtenues, ce qui est décrit à la section 5.2.2, pour deux manœuvres d'avions différents détaillées à la section 5.2.3. Des indices objectifs des performances, définis à la section 5.2.4, ont finalement été utilisés comme outils complémentaires de l'évaluation.

5.2.1 Perception des mouvements

Bien qu'une multitude de capteurs contribuent à la détection du mouvement chez l'humain, on s'entend généralement pour accorder un rôle prépondérant à la partie de l'oreille interne appelée le système vestibulaire [40]. De plus, il est généralement accepté que les indices de mouvement détectés par le système vestibulaire sont les composantes de la force spécifique évaluées au niveau de la tête (\mathbf{f}) et celles de la vitesse angulaire ($\boldsymbol{\omega}$) auxquelles l'humain est soumis, les premières étant perçues par les otholithes et les secondes par les canaux semi-circulaires.

Un modèle mathématique qui simule le comportement du système vestibulaire humain fut utilisé afin d'évaluer les sensations perçues par le pilote d'un avion ou d'un simulateur lors d'une manœuvre, permettant ensuite une comparaison objective de la simulation. Ce modèle vestibulaire reçoit à l'entrée les 6 composantes des vecteurs \mathbf{f} et $\boldsymbol{\omega}$ exprimées dans un repère lié au pilote (R_c pour le prototype), effectue le traitement approprié, puis produit 6 nouvelles composantes, formant ainsi la force spécifique perçue \mathbf{f}' et la vitesse angulaire perçue $\boldsymbol{\omega}'$. Ce traitement consiste principalement en un filtre des fréquences possédant une fenêtre passante s'étendant de 0.2 à 2 m/s² pour les composantes de \mathbf{f} et de 0.2 à 10 rad/s pour celles de $\boldsymbol{\omega}$. Des valeurs seuils, en deçà desquelles aucune sensation n'est perceptible, complètent ce modèle. Les valeurs utilisées varient entre 0.17 et 0.28 m/s² pour les forces spécifiques selon l'axe, et sont de 0.05 rad/s pour les vitesses angulaires. Elles ont été sélectionnées en se basant sur des études réalisées avec des pilotes effectuant des tâches de pilotage [31, 33].

Une description plus complète du modèle vestibulaire utilisé, incluant les courbes fréquentielles des réponses, peut être trouvée dans [31], alors que les détails de la programmation du modèle dans Simulink sont disponibles dans [30].

5.2.2 Production des résultats

Afin de comparer quelles sont les sensations de mouvement que le pilote percevra, que ce soit aux commandes de l'avion, du simulateur de vol sphérique de l'Université Laval, ou encore d'un simulateur classique à six degrés de liberté, une méthodologie de production des résultats a été mise en place, et est décrite à la figure 5.3. Sa réalisation pratique a été faite à l'aide de Simulink.

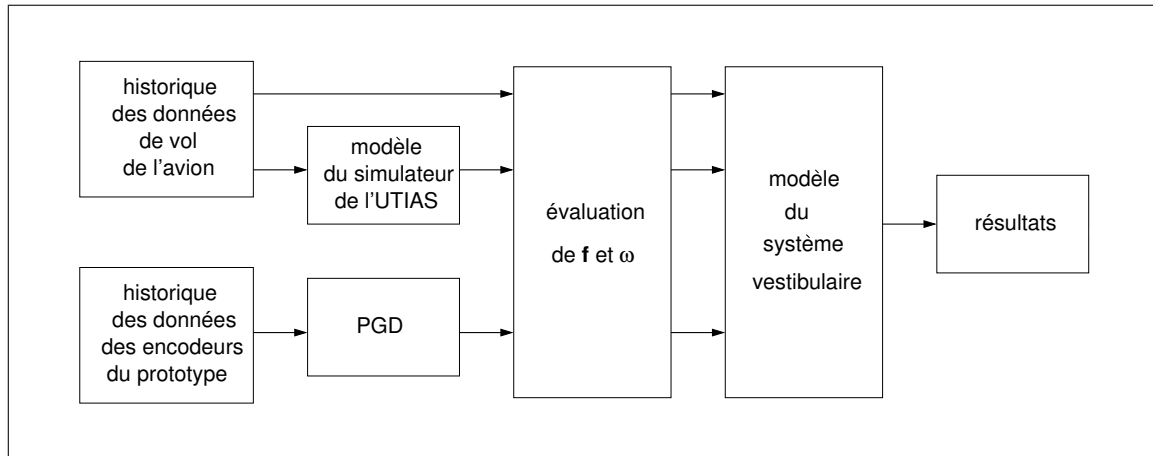


FIG. 5.3 – Principe de la production des résultats.

La première démarche consiste bien sûr à réaliser une manœuvre à l'aide des commandes de vol, et de contrôler à partir de ces commandes le prototype en temps réel. Lors de cette manœuvre, on enregistre à une fréquence constante de 20 Hz les données de vol fournies par X-Plane pour connaître les mouvements de l'avion simulé, ainsi que les données de position des moteurs du prototype fournies par les encodeurs, ce qui constitue les deux entrées à la figure 5.3.

Les données de position des moteurs sont fournies à la fonction appelée PGD, qui détermine la position cartésienne du prototype à l'aide des développements décrits à la section 2.2.2.

Quant aux données de vol de l'avion, elles sont fournies à la fonction appelée modèle du simulateur de l'UTIAS, qui est un modèle mathématique de la génération des mouvements pour la plate-forme de type Gough-Stewart à six degrés de liberté de l'UTIAS (University of Toronto Institute for Aerospace Studies). La photographie de la plate-forme, et ses débattements sont donnés en annexe B.4. La génération des mouvements pour cette plate-forme est réalisée par des *Washout Filters* classiques. Les

détails concernant le modèle ainsi que sa mise en œuvre dans Simulink sont disponibles dans [30].

Pour le prototype aussi bien que pour le modèle du simulateur classique, les gains des WF ont été posés unitaires. Le réglage des fréquences de coupures des filtres pour le WF du prototype a été de 8 rad/s pour le filtre passe-bas de l'axe de verticale apparente, et de 0.1 rad/s pour le filtre passe-haut de l'axe des rotations

Ensuite, le rôle de la fonction d'évaluation de \mathbf{f} et $\boldsymbol{\omega}$ à la figure 5.3 est de déterminer les forces spécifiques et les vitesses de rotation à la tête du pilote qui seront dans les trois cas les entrées du modèle du système vestibulaire. Pour cela, il faut connaître la distance entre le point d'évaluation des données et la tête du pilote. Dans le cas de la simulation de l'avion, les données fournies par X-Plane sont évaluées au centre de gravité de l'appareil, et X-Plane fournit pour chaque avion simulé la distance entre ce centre de gravité et la tête du pilote. Pour le simulateur de l'UTIAS, les WF sont réalisés au centre géométrique de la plate-forme, et le vecteur entre ce point et la tête du pilote est évalué à $\mathbf{d}_{UTIAS} = [-0.02, -0.465, -1.783]^T$ m dans un repère équivalent à R_c .

Finalement, pour évaluer les mouvements dans le simulateur sphérique, on utilise le vecteur reliant le centre de rotation à la tête du pilote pour le simulateur à taille réelle, qui est $\mathbf{d}_{sph} = [0.0, 0.0, -1.135]^T$ m dans le repère R_c . Ainsi, on pourra évaluer quelles seraient les sensations perçues par le pilote dans le simulateur sphérique à taille réelle pour des mouvements identiques à ceux créés par le prototype à taille réduite. Dans la suite de ce chapitre, on fera donc référence aux résultats obtenus avec le prototype aussi bien avec l'appellation simulateur sphérique que celle de prototype.

Enfin, le modèle du système vestibulaire décrit à la section précédente fournira les résultats pour la manœuvre, qui sont les forces spécifiques \mathbf{f}' et les vitesses de rotation $\boldsymbol{\omega}'$ ressenties par le pilote dans l'avion simulé, dans le simulateur à six degrés de liberté simulé, et dans le simulateur sphérique.

5.2.3 Manœuvres d'évaluation

Beaucoup de manœuvres différentes ont été réalisées avec le prototype afin de valider le système de calcul et le système mécanique. Une limitation commune à toutes ces

manœuvres est qu'elles n'ont pas été réalisées par des pilotes, mais plutôt par une novice de l'aviation : moi. Cela a eu pour conséquence de limiter à la fois la qualité des commandes appliquées, et le type de manœuvres envisageables. Deux manœuvres ont été sélectionnées ici à titre d'exemple de résultats, de manière à ce qu'elles soient les plus représentatives possible, qu'elles sollicitent tous les degrés de liberté du prototype, et qu'elles illustrent les principaux avantages et inconvénients des plates-formes de simulation de vol.

La première manœuvre est celle d'un Boeing 747, avion pour lequel des résultats de simulation avaient déjà été obtenus lors de la première étape de ce projet [6], et qui avait été modélisé avec un autre logiciel que X-Plane. Cette manœuvre permet donc de comparer les résultats obtenus lors de simulations typiques d'appareil de transport commercial possédant une grande inertie avec ceux du prototype, et donc de confirmer les performances de sensation du simulateur pour ce type d'appareil.

La seconde manœuvre est celle d'un avion de chasse de type F-22 raptor, et permet d'explorer des fréquences et amplitudes de mouvement plus élevées de l'appareil, et ainsi mettre en valeur certains avantages de l'architecture sphérique du simulateur. Ces deux manœuvres, qui sont maintenant décrites, débutent alors que l'avion est arrêté sur la piste avec ses moteurs en marche, et incluent donc un décollage impliquant à la fois des accélérations et rotations de l'appareil.

- **Manœuvre d'avion de transport commercial :**

Il s'agit de la manœuvre de Boeing 747. Lors de cette manœuvre, la manette des gaz est d'abord poussée à son maximum (100%), et lorsque les moteurs ont atteint environ 50 % de leur régime maximal, les freins sont enlevés et l'avion accélère sur la piste à partir de la 7^e seconde de la manœuvre. Lorsque la vitesse de l'avion atteint 200 nœuds (370 km/h), soit à la 43^e seconde, le train avant quitte le sol et l'avion prend son envol avec un angle de tangage qui reste inférieur à 15 degrés. Finalement, lorsque l'avion a atteint son premier palier de vol, la manette des gaz est réduite à environ 45 % de la puissance maximale à partir de la 92^e seconde, jusqu'à ce que l'appareil se stabilise dans une posture horizontale avec une vitesse à peu près constante. La manœuvre dure 118 secondes au total.

- **Manœuvre d'avion de chasse :**

Cette manœuvre simule dans un premier temps elle aussi le décollage de l'appareil. Les freins sont retirés à la 12^e seconde, et l'accélération est beaucoup plus grande

que pour la manœuvre précédente. L'avion amorce son décollage à la 36^e seconde, puis sa stabilisation vers l'horizontale à la 78^e seconde.

Dans un deuxième temps, à partir de la 143^e seconde, un double virage quasi-coordonné est amorcé en inclinant d'abord l'avion à droite, puis à gauche, et ce afin de réaliser un virage de 90 degrés. Ce virage termine la manœuvre, qui a une durée totale de 160 secondes.

5.2.4 Indices de performance

Les résultats obtenus ont été principalement étudiés visuellement. Cependant, comme cette façon de faire est nécessairement subjective, un outil complémentaire de comparaison plus objectif était requis. C'est pourquoi les *indices de performance*, introduits par [6], ont été utilisés.

Le premier de ces indices, appelé λ_1 , exprime, au moyen d'une valeur numérique unique, la moyenne des différences entre les mouvements produits dans l'avion et ceux qui le sont dans le simulateur alors que le second, λ_2 , indique plutôt la moyenne des différences entre la dérivée des mêmes mouvements. La définition formelle des ces indices est donnée par l'équation suivante dans laquelle $i \in \{1, 2\}$:

$$\lambda_i = 100 \times \left(\frac{\lambda_{if}}{a_{max}} + \frac{\lambda_{i\omega}}{\omega_{max}} \right) \quad (5.1)$$

Ainsi, il est clair que chacun des indices est une moyenne pondérée de deux sous-indices, l'un correspondant aux valeurs de forces spécifiques, l'autre aux valeurs des vitesses angulaires. De plus, comme les unités des quantités linéaires ne sont pas homogènes à celles des quantités angulaires, il a fallu normaliser ces quantités. Pour cela, elles ont été divisées par les limites supérieures correspondantes de la plate-forme classique, soit a_{max} et ω_{max} , telles que tirées du tableau B.4.

Les sous-indices qui définissent λ_1 sont alors donnés par :

$$\lambda_{1f} = \frac{1}{\mathcal{F}T} \sum_{j=0}^{\mathcal{F}T} \sqrt{(f_{xj} - f_{xj}^{sim})^2 + (f_{yj} - f_{yj}^{sim})^2 + (f_{zj} - f_{zj}^{sim})^2} \quad (5.2)$$

$$\lambda_{1\omega} = \frac{1}{\mathcal{F}T} \sum_{j=0}^{\mathcal{F}T} \sqrt{(\omega_{xj} - \omega_{xj}^{sim})^2 + (\omega_{yj} - \omega_{yj}^{sim})^2 + (\omega_{zj} - \omega_{zj}^{sim})^2} \quad (5.3)$$

où par exemple f_{kj} est la composante k de f à $t = j\Delta t$ dans l'avion, alors que f_{xj}^{sim} est la même composante, mais dans le simulateur.

Il s'agit donc d'évaluer à chaque intervalle de temps t la norme du vecteur différence des forces spécifiques (ou vitesses angulaires) et d'en faire la sommation pendant toute la durée de la manœuvre T . Cette somme est ensuite normalisée en divisant par T et par la fréquence d'échantillonnage \mathcal{F} (20 Hz)—ce qui correspond à diviser par le nombre total d'éléments intervenants dans la sommation.

Le second groupe de sous-indices, formant λ_2 , a pour but l'évaluation de la norme moyenne des différences des vecteurs contenant la dérivée des forces spécifiques (ou des vitesses angulaires) à l'instant t . On définit ces sous-indices par :

$$\lambda_{2f} = \frac{1}{(\mathcal{F}T - 1)} \sum_{j=1}^{\mathcal{F}T} \sqrt{\dot{f}_{xj}^2 + \dot{f}_{yj}^2 + \dot{f}_{zj}^2} \quad (5.4)$$

$$\lambda_{2\omega} = \frac{1}{(\mathcal{F}T - 1)} \sum_{j=1}^{\mathcal{F}T} \sqrt{\dot{\omega}_{xj}^2 + \dot{\omega}_{yj}^2 + \dot{\omega}_{zj}^2} \quad (5.5)$$

où on écrit, pour $k \in \{x, y, z\}$:

$$\dot{f}_{kj} = \frac{\Delta f_{kj}}{\Delta t} - \frac{\Delta f_{kj}^{sim}}{\Delta t} \quad (5.6)$$

$$\dot{\omega}_{kj} = \frac{\Delta \omega_{kj}}{\Delta t} - \frac{\Delta \omega_{kj}^{sim}}{\Delta t} \quad (5.7)$$

et où, par exemple f_{kj} est la composante k de f à l'instant $t = j\Delta t$ dans l'avion, alors que f_{xj}^{sim} est la même composante, mais dans le simulateur.

Ainsi, on voit que la dérivée des composantes de la force spécifique $\Delta f_{kt}/\Delta t$ (ou celle des composantes de la vitesse angulaire, $\Delta \omega_{kt}/\Delta t$) sont évaluées à l'instant t au moyen de la différence simple du voisin précédent avec Δt égal à $1/\mathcal{F} = 0.05$ sec. La sommation est ensuite faite pour toute la durée de la manœuvre et normalisée cette fois par le terme $(\mathcal{F}T - 1)$ au diviseur, la sommation débutant dans ce cas à $j = 1$.

5.3 Exemples graphiques et analyse

Suivant la méthodologie précédente, les manœuvres ont été réalisées avec le prototype, et les résultats produits. Quelques graphiques ont été sélectionnés comme exemples caractéristiques de l'ensemble des résultats obtenus. Chaque graphique montre une composante de force spécifique ou de vitesse angulaire ressentie pour la manœuvre étudiée. Pour présenter les résultats sous forme compacte, trois courbes sont placées sur chaque graphique : une ligne pleine de couleur verte montre la composante ressentie dans l'avion, une ligne pointillée rouge montre la composante ressentie dans le simulateur à six degrés de liberté, et finalement une ligne tiretée bleue montre la composante ressentie dans le simulateur sphérique.

5.3.1 Manœuvre d'avion de transport commercial

Lors de cette manœuvre, des accélérations longitudinales de basses et moyennes fréquences de relativement grande amplitude sont générées lors du décollage, alors que les vitesses de rotation de l'appareil sont de basses fréquences, et de faible amplitude. Ces dernières seront peu ou pas ressenties par le pilote, et ne sont donc pas représentées ici. La figure 5.4 illustre donc uniquement les forces spécifiques \mathbf{f}' ressenties par le pilote. Les points d'analyse les plus importants sont indiqués sur cette figure par des numéros, et sont maintenant commentés.

La figure 5.4(a) montre la composante longitudinale des forces spécifiques ressenties. On voit dans la zone marquée **1** que dès que les freins de l'appareil sont enlevés, le pilote est soumis dans l'avion à une force spécifique de grande amplitude, qui correspond à l'accélération de l'appareil sur la piste. Les deux types de simulateurs utilisent alors l'artifice de la verticale apparente pour simuler cette accélération, et la pente de la force spécifique simulée est limitée comme on le voit par la limite en vitesse de rotation de cet artifice. Arrivé à la zone marquée **2**, le simulateur sphérique prend un léger avantage sur le simulateur classique, car ce dernier réalise un dépassement de la force spécifique à simuler. Ce même phénomène se reproduit dans la zone **3**, qui correspond à la zone de décollage de l'appareil, puis dans la zone **4** lors de la stabilisation de l'appareil, zones pour lesquelles le simulateur sphérique se comporte de façon plus adéquate grâce à ses WF adaptés.

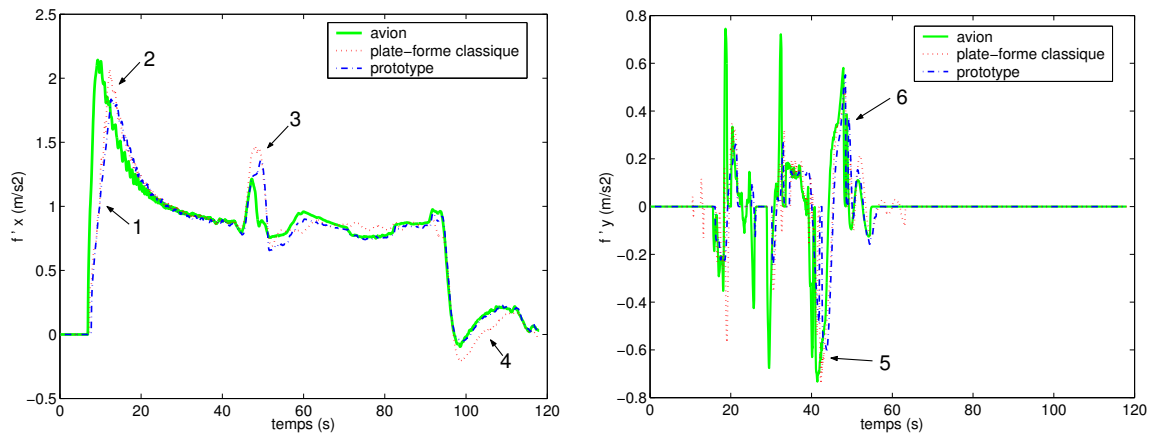
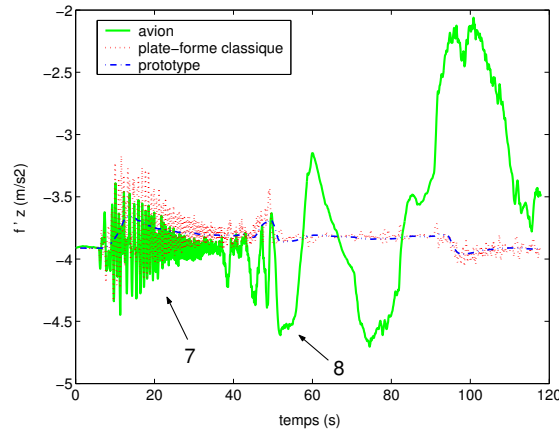
(a) Composantes selon X_c de \mathbf{f}' (b) Composantes selon Y_c de \mathbf{f}' (c) Composantes selon Z_c de \mathbf{f}'

FIG. 5.4 – Forces spécifiques perçues lors de la manœuvre.

La figure 5.4(b) montre la composante latérale des forces spécifiques ressenties, qui reste de faible amplitude pendant toute la manœuvre. La force spécifique principale à simuler est celle marquée par les zones 5 et 6, où l'avion en décollant subit un léger mouvement de roulis, qui crée une force spécifique latérale à gauche puis à droite. Cette accélération est très bien simulée par les deux types de simulateurs.

La figure 5.4(c) montre enfin la composante verticale des forces spécifiques ressenties, et illustre la difficulté des deux types de simulateurs à simuler des accélérations verticales. En effet, l'artifice de la verticale apparente étant basé sur le principe d'aligner l'accélération gravitationnelle dans la direction de la force spécifique à simuler,

cet artifice n'a aucune utilité suivant l'axe de la gravité, et donc très peu d'efficacité suivant Z_c . Cela constitue une lacune commune à tous les simulateurs de vol, qui est bien connue dans le milieu de la simulation de vol. On voit donc à la zone marquée **8** que dès que l'avion décolle, son pilote est soumis à des accélérations suivant sa verticale, mais qu'aucun des deux simulateurs n'est capable de simuler. La zone **7** montre aussi un avantage du simulateur à six degrés de liberté par rapport au simulateur sphérique. On voit que pendant la phase d'accélération, le pilote ressent dans l'avion des accélérations de hautes fréquences et de très faibles amplitudes, sûrement dues au contact des roues de l'appareil en mouvement sur la piste. Alors que le simulateur sphérique ne peut pas simuler ces accélérations, le simulateur classique le peut en se traduisant suivant son axe vertical.

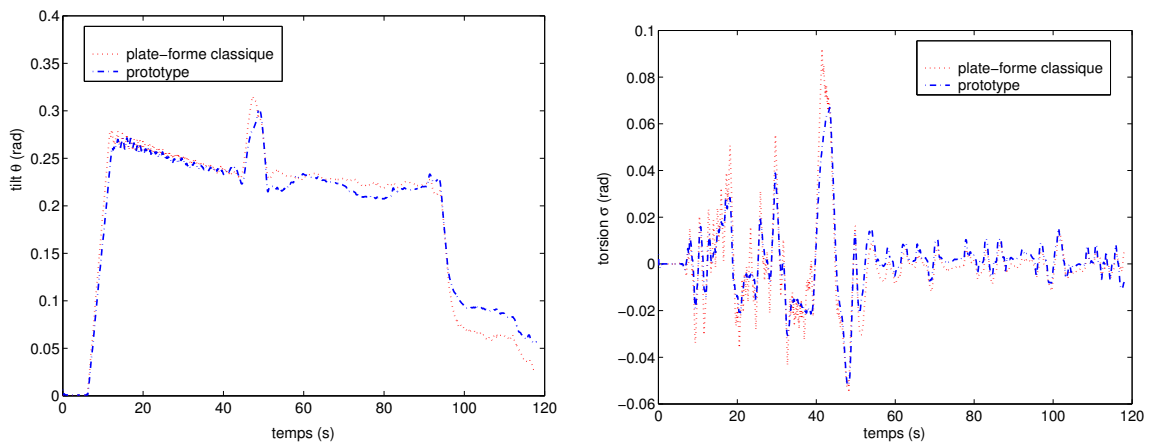
(a) Tilt θ .(b) Torsion σ .

FIG. 5.5 – Déplacements angulaires de Tilt et Torsion.

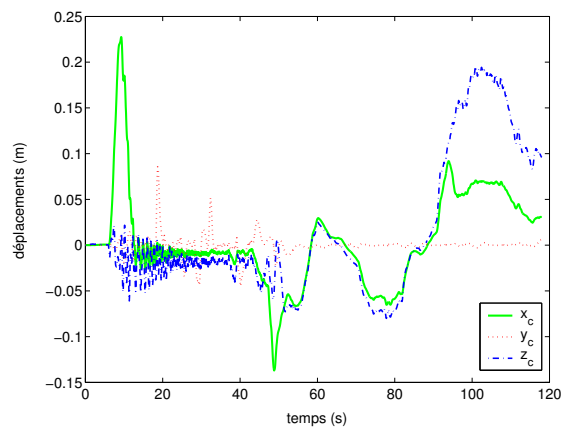


FIG. 5.6 – Déplacements linéaires du simulateur UTIAS.

Les déplacements angulaires des plates-formes durant la manœuvre sont illustrés à la figure 5.5, et exprimés en fonction des angles de *Tilt & Torsion*, alors que les déplacements linéaires pour la plate-forme classique sont donnés à la figure 5.6 dans le repère lié à la plate-forme. On remarque à la figure 5.5 que les déplacements angulaires des deux simulateurs sont comparables, les différences notables étant dues à l'adaptation des WF pour l'architecture du prototype et correspondent aux zones de différences des forces spécifiques soulignées précédemment. La figure 5.6 illustre aussi le fait que les déplacements linéaires de la plate-forme à six degrés de liberté permettent de simuler les accélérations de haute fréquences suivant l'axe Z_c , mais qu'ils ne suffisent pas du tout pour simuler les variations plus importantes de force spécifique suivant cet axe à cause de leurs faibles débattements.

Globalement, comme le montre la figure 5.4, la manœuvre est très bien simulée par les deux types de simulateurs, et les sensations produites sont véritablement de qualité comparable. Cette remarque à partir des courbes est confirmée par les indices de performance, qui sont donnés pour le prototype et le simulateur classique au tableau 5.1. D'abord, les deux indices sont très faibles pour les deux simulateurs, ce qui confirme

	Prototype	Plate-forme classique
λ_1 (%)	5.54	5.61
λ_2 (%)	5.53	4.85

TAB. 5.1 – Indices de performance pour la manœuvre 1.

la bonne qualité de simulation pour une manœuvre de ce genre. Ensuite, l'indice λ_1 est meilleur pour le prototype que pour le simulateur classique, ce qui est dû principalement à l'adaptation des WF à l'architecture, qui compense les quelques limitations dues au fait qu'il n'y a aucune translation. Finalement, l'indice λ_2 est meilleur pour le simulateur classique, ce qui est le reflet principalement de la simulation par la plate-forme à six degrés de liberté des variations hautes fréquences des accélérations suivant l'axe Z_c .

5.3.2 Manœuvre d'avion de chasse

Lors de cette manœuvre, des accélérations et vitesses de rotation comportant des composantes à plus hautes fréquences, et de plus grandes amplitudes, sont générées. En

effet, la dynamique de l'avion simulé, qui est un avion de chasse F-22, est très rapide, car l'avion est léger et très puissant. Ainsi, même si la manœuvre est de type similaire à la précédente, les défis de la simulation de tels mouvements sont plus importants pour un simulateur de vol. La figure 5.7 illustre les principaux résultats pertinents pour cette manœuvre.

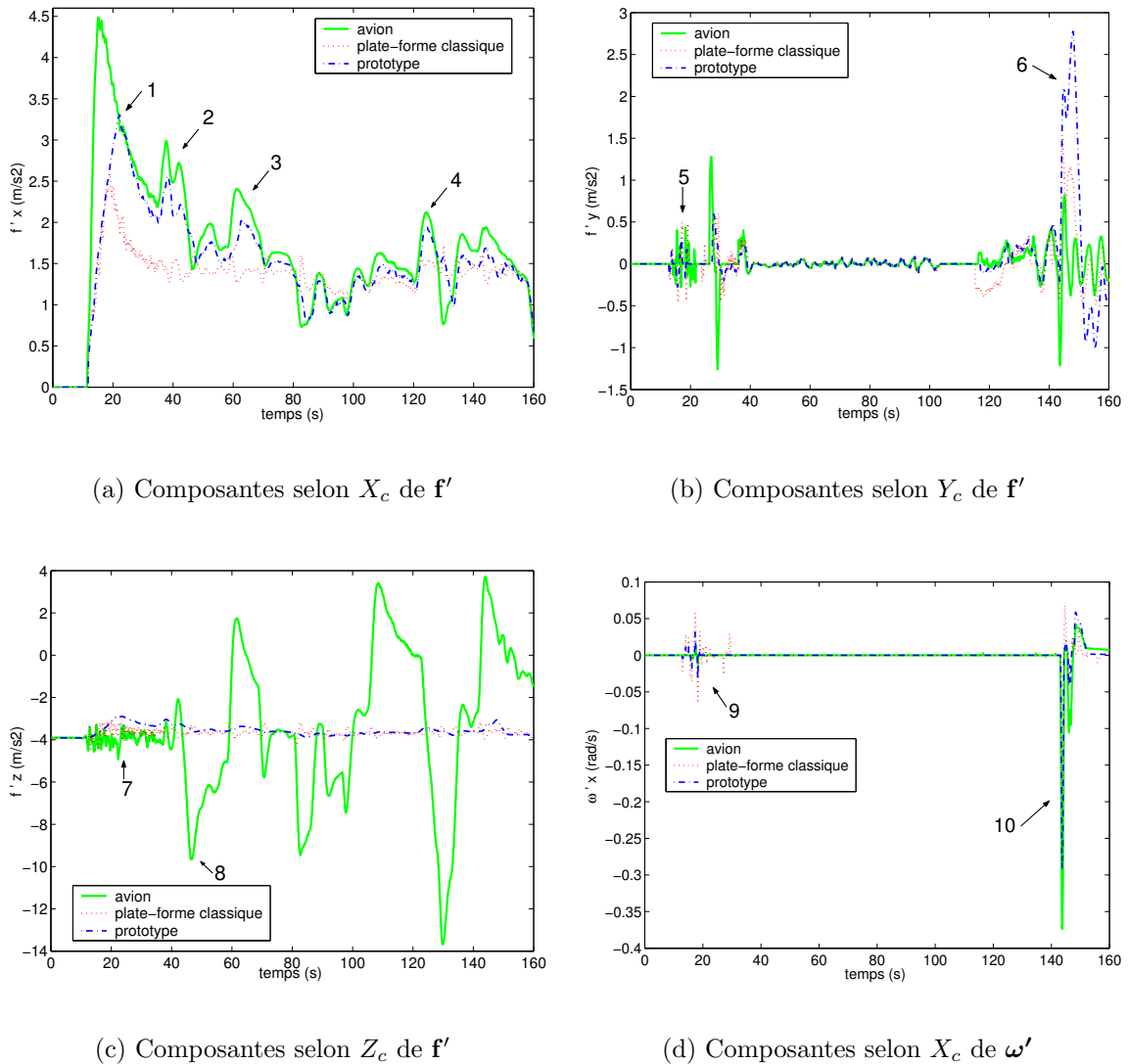


FIG. 5.7 – Forces spécifiques et vitesses angulaires perçues lors de la manœuvre.

La figure 5.7(a) montre la composante longitudinale des forces spécifiques ressenties. On voit que l'amplitude de la force spécifique ressentie par le pilote dans l'avion lors de l'accélération initiale de l'appareil est plus que doublée par rapport à la manœuvre précédente (l'accélération réelle de l'appareil dépasse $7m/s^2$). L'artifice de la verticale apparente est alors insuffisant pour simuler l'accélération, encore une fois à

cause de sa limitation en vitesse de rotation. On remarque cependant le grand avantage dans ce cas du simulateur sphérique dans la zone **1**, qui peut simuler une force spécifique de bien plus grande amplitude que le simulateur classique grâce à ses grands débattements en rotation. Ainsi, alors que le simulateur classique atteint la limite de son espace atteignable rapidement, le simulateur sphérique peut produire une sensation de force spécifique de plus en plus grande, qui atteint l'amplitude de celle ressentie dans l'avion à la 22^e seconde. Les débattements du simulateur sphérique lui permettent aussi de simuler les variations de la force spécifique longitudinales suivantes, situées aux zones **2**, **3** et **4**, car sa limite d'espace atteignable n'est pas atteinte.

Les figures 5.7(b) et 5.7(d) montrent respectivement la composante latérale des forces spécifiques ressenties, et la composante autour de l'axe longitudinal des vitesses de rotation ressenties. Ces deux courbes sont présentées ensemble car elles sont fortement couplées. D'abord, dans la zone marquée **5** sur la courbe de f'_y et **9** sur la courbe de ω'_x , un léger mouvement de roulis entraîne des sensations de forces spécifiques latérales, qui sont bien simulées par les deux types de simulateurs.

Par contre, lors du virage quasi-coordonné, correspondant à la zone marquée **6** sur la courbe de f'_y et **10** sur la courbe de ω'_x , un inconvénient du simulateur sphérique est souligné. En effet, comme ce simulateur ne peut réaliser que des rotations, les sensations de mouvement linéaires et de rotations sont nécessairement couplées, et alors que la vitesse de rotation est beaucoup mieux simulée que par le simulateur classique, elle s'accompagne d'une sensation de force spécifique latérale parasite, d'amplitude considérable.

Finalement, la figure 5.7(c) montre la composante verticale des forces spécifiques ressenties. Les courbes présentent les mêmes allures que pour la manœuvre précédente sauf que l'amplitude des forces spécifiques de l'avion est plus élevée, et les mêmes remarques peuvent être faites. Dans la zone **7**, le simulateur classique est capable de simuler les forces spécifiques de hautes fréquences et de faible amplitude, alors que dans la zone **8** aucun des deux simulateurs ne simule les forces spécifiques de grande amplitude.

Les déplacements angulaires des plates-formes durant la manœuvre sont illustrés à la figure 5.8, et exprimés en fonction des angles de *Tilt* et *Torsion*, alors que les déplacements linéaires pour la plate-forme classique sont donnés à la figure 5.9 dans le repère lié à la plate-forme. On remarque à la figure 5.8 que pour cette manœuvre,

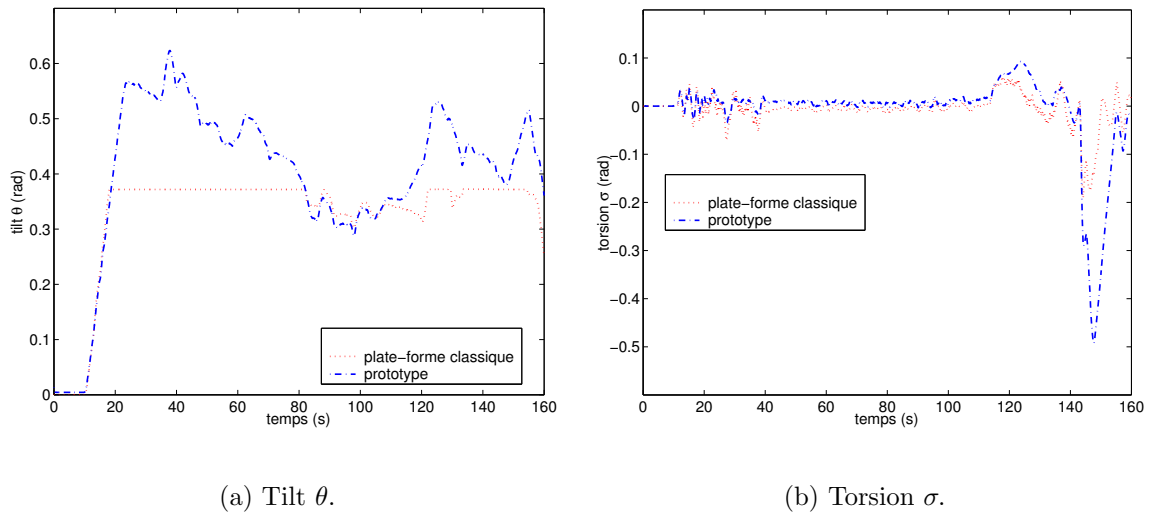


FIG. 5.8 – Déplacements angulaires de Tilt et Torsion.

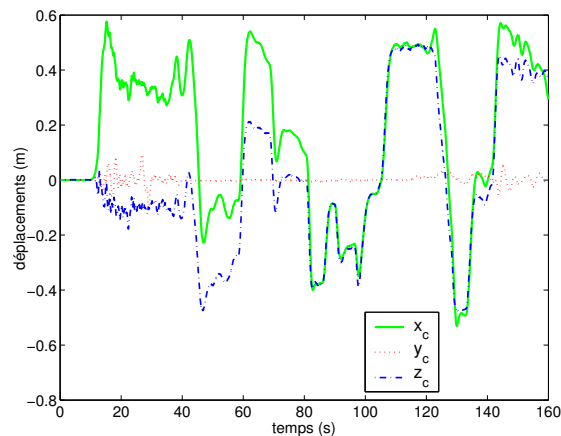


FIG. 5.9 – Déplacements linéaires du simulateur UTIAS.

comme on a déjà pu le constater en étudiant les courbes précédentes, les déplacements angulaires autorisés par le prototype sont un avantage majeur, et on peut constater que la limite en tilt n'est même pas atteinte lors de cette manœuvre, et donc les capacités de ce simulateur sont encore au delà des mouvements réalisés ici. La courbe de torsion montre aussi que les grands débattements sont aussi efficaces lors de la simulation du virage. Au contraire, les figures 5.8(a) et 5.9 montrent que les débattements angulaires de la plate-forme classique sont trop faibles pour simuler adéquatement cette manœuvre, alors que les déplacements linéaires sont tout juste suffisants pour les axes X_c et Z_c .

Globalement, la manœuvre représente un défi pour la simulation, car elle comporte

des accélérations et vitesses de rotation à la fois de grande amplitude et de haute fréquence, elle est donc bien moins bien simulée que la manœuvre précédente par les deux simulateurs. Cette difficulté se reflète dans les résultats des indices de performance donnés au tableau 5.2, qui sont plus élevés. L'indice λ_1 est meilleur pour le prototype

	Prototype	Plate-forme classique
λ_1 (%)	25.95	27.10
λ_2 (%)	10.81	12.23

TAB. 5.2 – Indices de performance pour la manœuvre 2.

que pour le simulateur classique, mais la différence entre les deux n'est pas aussi élevée qu'on aurait pu le penser à cause des forces spécifiques parasites qui ont été générées suivant l'axe Y_c par le prototype. Par contre, cette fois, l'indice λ_2 est aussi meilleur pour le prototype, en partie à cause du fait que les dérivées des mouvements ressentis sont fortement erronées pour le simulateur classique lorsque ce dernier atteint les limites de son espace atteignable.

5.3.3 Analyse globale

Suite aux commentaires formulés pour les deux manœuvres étudiées, qui sont représentatives des manœuvres générales qui peuvent être réalisées par un simulateur de vol, plusieurs remarques globales peuvent être faites.

D'abord, pour un avion de transport commercial, pour lequel les accélérations et rotations à simuler sont d'amplitudes et de fréquences relativement faibles, le simulateur sphérique représente une alternative viable aux systèmes classiques possédant 6 degrés de liberté car la simulation offerte par ce système simplifié est d'une qualité comparable, voire meilleure grâce à ses WF adaptés pour son architecture. Le simulateur génère globalement des résultats similaires à ceux de la plate-forme classique simulée, en raison de la composition première en basses fréquences des mouvements, qui peuvent être adéquatement simulés par l'artifice de la verticale apparente.

L'inconvénient souligné de la plate-forme sphérique est son incapacité à simuler des accélérations linéaires de hautes fréquences, car ces dernières ne peuvent pas être

produites indépendamment en utilisant seulement des rotations. Néanmoins, les répercussions de ce désavantage pourront être diminuées dans le système à taille réelle par l'utilisation du caisson de graves, qui pourra produire des vibrations simulant ces accélérations.

Ensuite, pour des manœuvres nécessitant des mouvements de grande amplitude et de fréquence plus élevée, comme cela pourrait être le cas aussi pour des avions de type Cessna ou encore des hélicoptères, l'exploitation du grand espace atteignable de la plate-forme sphérique présente un avantage considérable sur une plate-forme classique, dont les débattements en rotation sont faibles. En effet, ces débattements permettent de simuler à la fois des accélérations linéaires de grande amplitude et des vitesses de rotation dans une gamme de fréquences plus grande. Il est cependant nécessaire de limiter la gamme de ces fréquences, car comme on l'a constaté lors de la deuxième manœuvre, la simulation de vitesses de rotation de haute fréquence et grande amplitude entraîne des forces spécifiques parasites. En effet, comme le simulateur possède seulement trois degrés de liberté en rotation, le pilote peut ressentir certaines forces spécifiques de hautes fréquences, mais ces sensations demeurent généralement couplées aux accélérations angulaires et ne peuvent donc pas être produites de façon indépendante.

Cependant, les manœuvres ont été réalisées ici avec le même réglage des filtres pour la comparaison, alors qu'il serait plus judicieux de réaliser un réglage selon la dynamique de l'avion simulé, et même idéalement basé sur les sensations réelles de pilotes. Par exemple, on pourrait minimiser les forces spécifiques parasites dues à des vitesses de rotation de grande amplitude et de hautes fréquences en limitant un peu plus la fréquence de coupure du filtre HF appliqué aux vitesses de rotation, ou en appliquant un facteur d'échelle inférieur à un pour ces vitesses de rotation lorsqu'un avion de ce type est simulé.

Conclusion

Les travaux décrits dans ce mémoire visaient deux objectifs principaux. La présente conclusion a pour but de résumer les conclusions partielles qui furent émises en regard de chacun de ces objectifs et de présenter quelques travaux futurs et perspectives nouvelles qui pourraient désormais être envisagés.

Résumé des travaux

Les trois premiers chapitres ont permis de présenter les travaux réalisés pour répondre au premier objectif, soit l'analyse du mécanisme de simulation de mouvement. Après une revue des travaux précédents, et un aperçu des modifications apportées à la conception mécanique du mécanisme, le premier chapitre a détaillé l'architecture mécanique finale du simulateur de vol. La cinématique du mécanisme a ensuite été étudiée au chapitre 2. L'analyse cinématique a montré que ce mécanisme hybride à trois degrés de liberté en rotation remplit les conditions cinématiques du cahier des charges, car son espace atteignable est très étendu, et sa dextérité sur cet espace atteignable est très bonne. Puis le modèle dynamique du mécanisme a été obtenu au chapitre 3 en utilisant la méthode de Lagrange, et un système d'équilibrage statique partiel a été développé. Ce dernier entraîne l'élimination en moyenne sur l'espace atteignable de près de 70% des couples aux actionneurs nécessaires au maintien statique du mécanisme, et permet donc l'utilisation des moteurs électriques qui était souhaitée dans le cahier des charges. Le modèle dynamique, incluant l'équilibrage statique, a finalement été validé

par comparaison avec le modèle ADAMS du mécanisme pour différentes trajectoires. Ces analyses permettent d'affirmer que l'architecture mécanique de ce simulateur de vol possède les propriétés requises par le cahier des charges.

Le deuxième objectif, portant sur le développement du système de génération des mouvements et de contrôle, a été traité dans les deux derniers chapitres. Le chapitre 4 a d'abord résumé les développements matériels et logiciels réalisés pour concevoir ce système. Finalement, le chapitre 5 a montré quelques résultats expérimentaux obtenus avec un prototype à taille réduite, qui ont confirmé d'une part que le système de calcul fonctionne en pratique tel que désiré, puis que ce simulateur de mouvement peut produire des sensations de mouvement au pilote de très bonne qualité, voire de qualité comparable aux sensations produites par un simulateur de mouvement classique de type Gough-Stewart à six degrés de liberté.

Travaux futurs

Les prochaines étapes à franchir dans ce projet vont mener à la construction du simulateur de mouvement à taille réelle. Pour son système mécanique, toutes les composantes achetées chez des fournisseurs (les ressorts, les moteurs, les amplificateurs) sont déjà en place. La dernière étape, et non la moindre, est donc le calcul de résistance des matériaux pour les pièces constituant les membrures des pattes, afin de concevoir la forme finale de ces pièces et les fabriquer à l'université. Ce travail est réalisé en ce moment même par Simon Foucault.

Pour son système de calcul, tout le système qui sera utilisé pour le simulateur à taille réelle a déjà été mis en pratique, et toute l'architecture matérielle est déjà présente (ordinateurs, carte de contrôle). Pour adapter ce système au simulateur à taille réelle, il faudra tout de même changer différentes variables dans le logiciel de contrôle (taille des membrures pour le PGI, performances des moteurs), et il reste aussi à développer l'alimentation en courant.

Une fois le simulateur construit, il sera primordial de réaliser des évaluations avec des pilotes pour prouver complètement ce concept et valider les études de sensations des mouvements dans un tel simulateur. En pratique, de telles évaluations ont été réalisées [41], et ont montré que même si les perceptions changent d'un pilote à l'autre, leur jugement permet de valider ou de réfuter globalement ces études de sensations des mouvements. Des méthodes de réglage fin des filtres WF en utilisant les évaluations des

pilotes ont même été développées [42, 43], et pourraient être utilisées à ce stade.

Différentes améliorations au système prévu peuvent aussi être d'ores et déjà envisagées. Par exemple, l'interface de l'utilisateur avec le logiciel de génération des mouvements est pour l'instant sommaire, c'est à dire que l'utilisateur dispose uniquement d'une fenêtre de commande. Afin de rendre cette interface plus interactive, et plus transparente pour l'utilisateur, il serait possible de réaliser une interface plus évoluée et graphique, comme cela se fait pour les logiciels commerciaux. Par ailleurs, si la carte de contrôle est changée dans l'avenir, alors le modèle dynamique développé pourra aisément être programmé afin d'améliorer les performances de l'asservissement. En effet, pour l'instant la fréquence d'exécution du système de calcul est limitée par les données de X-Plane et non par les calculs du logiciel de génération des mouvements, on peut donc augmenter de beaucoup ces calculs réalisés sans pour autant diminuer la fréquence d'exécution.

Perspectives d'exploitation du simulateur

Une fois construit, le simulateur de mouvement possédera un très grand potentiel d'exploitation, et pourra être utilisé aussi bien pour les premières phases d'entraînement des pilotes qu'en tant que simulateur de recherche.

Dans le premier cas, un travail sur l'évolution des certifications des simulateurs reste à faire pour un jour convaincre l'industrie de la pertinence d'un tel projet, et des avantages d'un tel système.

Dans le deuxième cas, de nombreuses perspectives sont offertes grâce entre autres à la grande flexibilité et reconfigurabilité du système de calcul développé. D'abord, comme on vient de le mentionner, les capacités de calcul du logiciel de génération des mouvements sont très grandes, et permettent d'envisager de nombreuses modifications. En particulier, l'amélioration des WF reste aujourd'hui encore un intérêt de recherche, et de nombreuses alternatives aux filtres classiques peuvent être étudiées. Par exemple, il est possible de réaliser l'adaptation des coefficients des filtres en temps réel, on peut aussi envisager le choix de ces coefficients par des réseaux de neurones (qui pourraient être entraînés sur la base des modèles de perception des mouvements), ou bien modifier le schéma de l'algorithme en incluant les WF dans la boucle d'asservissement avec le modèle dynamique, et ainsi réaliser un contrôle robuste [44] du

mécanisme.

Ensuite, comme le modèle dynamique de l'avion et les calculs graphiques pour le système visuel ont été distribués sur deux différents ordinateurs, le simulateur peut s'adapter à différents couplages de ces deux éléments. En outre, le simulateur peut être utilisé comme outil de recherche sur les modèles dynamiques d'avion, en couplant un modèle dynamique d'avion différent de ceux fournis par X-Plane avec les calculs graphiques réalisés par X-Plane (cette possibilité est aussi très intéressante pour une application d'entraînement des pilotes car les compagnies d'aviation ont très souvent leurs propres modélisations pour leurs avions).

Pour la même raison, le simulateur peut aussi être exploité comme outil de recherche sur le système visuel, en couplant le modèle dynamique de l'avion avec un autre système de calcul graphique et/ou en modifiant la projection des images. Cela permettrait par exemple de réaliser des études sur l'influence des sensations visuelles sur les sensations de mouvement, ou sur l'importance relative de ces deux sensations, sujet qui reste à ce jour discuté.

Bibliographie

- [1] Gough, V.E. et Whitehall, S.G., 1962, "Universal tire test machine," *Proceeding 9th Int. Technical Congress F.I.S.I.T.A.*, vol. 117, pp. 117-135.
- [2] Stewart, D., 1965, "A platform with six degrees of freedom," *Proceeding of the Institution of Mechanical Engineers*, Vol. 180, No. 15, pp. 371-378.
- [3] Nahon, M., Ricard, R., Gosselin, C. M., 1995, "A Comparison of Flight Simulator Motion-Base Architectures," *Proceedings of the Confederation of European Aerospace Societies Symposium on Simulation Technologies*, Delft, the Netherlands, October 30th-November 1st, pp. 1-16.
- [4] Ben-Horin, R., Shoham, M., 1996, "Construction of a Six-Degree-of-Freedom Parallel Manipulator with Three Planarly Actuated Links," *Proceedings of the ASME DETC*, Irvine, California, August 18-22.
- [5] Sabrié, É., Dufour, P., Gosselin, C. M., 2004, "Kinematic and Dynamic Analysis of a New Rotational Motion Simulator," *Proceedings of the RoManSy 04 conference*, Saint Hubert, Canada, June 14-18.
- [6] Pouliot, N., 1997, "Analyse, optimisation et conception de mécanismes de simulation de mouvement à trois degrés de liberté," Mémoire de maîtrise, Département de génie mécanique, Faculté des sciences et de génie, Université Laval, Québec, Canada.
- [7] Dufour, P., 2002, "Analyse cinématique, dynamique et équilibrage statique d'un mécanisme parallèle sphérique à 3 degrés de liberté pour une installation de simu-

- lation de vol,” Mémoire de maîtrise, Département de génie mécanique, Faculté des sciences et de génie, Université Laval, Québec, Canada.
- [8] Bonev, I., Zlatanov, D. et Gosselin, C.M., 2002, “Advantages of Modified Euler Angles in the Design and Control of PKMs,” *Parallel Kinematic Machines International Conference (PKS 2002)*, Chemnitz, Germany, pp. 171-188.
- [9] Angeles, J., Gosselin, C. M., 1991, “Détermination du degré de liberté des chaînes cinématiques,” *Transactions de la Société Canadienne de Génie Mécanique*, vol. 14, no. 4, pp. 219-226.
- [10] Nikravesh, P. E., 1988, “Computer-Aided Analysis of Mechanical Systems,” Englewood Cliffs, Prentice-Hall, pp. 351.
- [11] Gosselin, C. M., Angeles, J., 1991, “A global performance index for the kinematic optimization of robotic manipulators,” *ASME Journal of Mechanical Design*, vol. 113, no. 3, pp. 220-226.
- [12] Hartenberg, R. S., Denavit, J., 1955, “A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices,” *Journal of Applied Mechanics*, vol. 22, no. 2, pp. 215-221.
- [13] Dasgupta, B., Mruthyunjaya, T. S., 1998, “Newton-Euler formulation for the Inverse Dynamics of the Stewart Platform Manipulator,” *Mechanism and Machine Theory*, vol. 33, no. 8, pp. 1135-1152.
- [14] Di Gregorio, R., Parenti-Castelli, V., 2002, “Dynamics of a Class of Parallel Wrist,” *Proceedings of the ASME DETC*, Montreal, Canada, September 29 - October 2.
- [15] Wang, J., Gosselin, C. M., 1998, “A new Approach for the Dynamic Analysis of Parallel Manipulators,” *Multibody System Dynamics*, vol. 2, no. 3, pp. 317-334.
- [16] Miller, K., 2001, “Dynamics of the new UWA Robot,” *Proceedings of Australian Conference on Robotics and Automation*, Sydney, Australie, November 14-15.
- [17] Angeles, J., 1997, “Fundamentals of Robotic Mechanical Systems,” Springer-Verlag, New York (USA).
- [18] Streit, D. A., Chung, H., Gilmore, B. J., 1991, “Perfect Equilibrators for Rigid Body Spatial Rotations About a Hooke’s Joint,” *Journal of Mechanical Design*, vol. 113, pp. 500-507.
- [19] Gosselin, C. M., 1999, “Static Balancing of Spherical 3-DoF Parallel Mechanisms and Manipulators,” *The International Journal of Robotics Research*, vol. 18, no. 8, pp. 819-829.

- [20] Herder, J. L., 2002, "Some considerations regarding Statically Balanced Parallel Mechanisms," *Proceedings of the Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators*, Québec, Canada, October 3-4 .
- [21] Johnson, K., Ebert-Uphoff, I., 2000, "Development of a Statically-Balanced Parallel Platform Mechanism with Six Degrees-of-Freedom," *Proceedings of the Parallel Kinematic Machines Conference*.
- [22] Gosselin, C. M., Wang, J., 1998, "On the Design of Statically Balanced Motion Bases for Flight Simulators," *Proceedings of the AIAA Modeling and Simulation Technologies Conference and Exhibit*, Boston, USA, August 10-12.
- [23] Brinkman, M. L., Herder, J. L., 2002, "Optimizing a Balanced Spring Mechanism," *Proceedings of the ASME DETC*, Montreal, Canada, September 29 - October 2.
- [24] Herder, J. L., 2001, "Energy-free Systems : Theory, Conception and Design of Statically Balanced Spring Mechanisms," *PhD-thesis*, Delft University of Technology, Delft, The Netherlands.
- [25] Quinn, B., Shute D., 1996, "Windows Sockets Network Programming," Addison-Wesley, Boston (USA).
- [26] Gondola T., Barbour, S., Chandler J., 2002, —Site internet—, "*X-Plane 7 Review*", http://www.digital-flight.com/reviews/X%20Plane%207/x_plane_7_review.htm
- [27] Fidelity Flight Simulation Inc., 2002, —Site internet—, "*Accréditation FAA pour les simulateurs de vol Motus*", http://www.flightmotion.com/docs/faa_approval.htm
- [28] Supnik B., Barbour, S., 2003, —Site internet—, "*X-Plane SDK Documentation*", <http://xsquawkbox.net/xpsdk/docs/index.html>
- [29] Salzman P. J., Pomerantz, O., 2003, —Site internet—, "*The Linux Module Kernel Programming Guide 2.4*", <http://www.tldp.org/LDP/lkmpg/lkmpg.pdf>
- [30] Sabrié, É., 2002, "Adaptation de Washout Filters à un simulateur de vol à trois degrés de liberté," Mémoire de projet de fin d'études, Laboratoire de robotique, Université Laval, Québec, Canada.
- [31] Reid, L. D., Nahon, M. A., 1985, "Flight Simulation Motion-Base Drive Algorithms : Part 1 - Developing and Testing the Equations", *UTIAS* No. 296.
- [32] Reid, L. D., Nahon, M. A., 1986, "Flight Simulation Motion-Base Drive Algorithms : Part 2 - Selecting the System Parameters", *UTIAS* No. 296.
- [33] Groen E., Valenti Clari M., Hosman R., 2001, "Evaluation of Perceived Motion During a Simulated Takeoff Run", *Journal of Aircraft*, Vol. 38, No 4.
- [34] Delta Tau Data Systems, mai 2004, "Turbo PMAC user manual".

- [35] Delta Tau Data Systems, mai 2004, “Turbo PMAC/PMAC2 Software Reference Manual”.
- [36] Delta Tau Data Systems, janvier 2003, “Pewin32 Pro Software Reference Manual”.
- [37] Lassure, C., Grant, M., 2002, “Dictionnaire anglais-français pour l’informaticien et l’internaute,” Édition Ellipses, Paris (France).
- [38] Laliberté, T, Gosselin, C. M., Côté, G., 2001, “A rapid prototyping framework for fast and cost-effective design of robotic mechanism prototypes ”, *IEEE Robotics and Automation Magazine* , Vol. 8, No 3, pp. 43-52.
- [39] Johansson, M., Nordin, J., 2002, “A Survey of Driving Simulators and their Suitability for Testing Volvo Cars,” Mémoire de maîtrise, Département de machines et systèmes des véhicules, Université Chalmers, Göteborg, Suède.
http://www.mvs.chalmers.se/Exjobb/Reports/exjobb_driving_simulator.pdf
- [40] Gum, D. R., 1973, “Modeling of the Human Force and Motion-Sensing Mechanisms”, *AFHRL-TR-72-54*.
- [41] Reid, L. D., Nahon, M. A., 1986, “Flight Simulation Motion-Base Drive Algorithms : Part 3 - Pilot evaluations”, *UTIAS* No. 296.
- [42] Grant, P., Reid, L., 1997, “Motion Washout Filter Tuning : Rules and Requirements”, *Journal of Aircraft*, Vol. 34, No 2, pp. 145-151
- [43] Grant, P., Reid, L., 1997, “PROTEST : An Expert System for Tuning Simulator Washout Filters”, *Journal of Aircraft*, Vol. 34, No 2, pp. 152-159.
- [44] Idan, M., Nahon, M. A., 1999, “Off-Line Comparison of Classical and Robust Flight Simulator Motion Control”, *Journal of Guidance, Control, and Dynamics*, Vol. 22, No 5.

Annexe A

Cinématique

A.1 Détails de la résolution du PGD

La résolution du PGD consiste à déterminer, pour des variables articulaires données $(\theta_1, \theta_2, \theta_3)$, quelles valeurs peuvent prendre les variables cartésiennes (α, β, γ) . Pour cela, on utilise le même principe que celui mis en œuvre lors de la résolution du PGI, c'est à dire que l'on écrit pour chaque patte la fermeture de la chaîne cinématique.

A.1.1 Solutions pour l'angle α

L'angle α est résolu en écrivant la fermeture de la première chaîne cinématique, pour laquelle on a :

$$\begin{aligned} \mathbf{v}_1^T \mathbf{v}_1 &= l_{21}^2 = (\mathbf{p}_1 - \mathbf{r}_1 - \mathbf{u}_1)^T (\mathbf{p}_1 - \mathbf{r}_1 - \mathbf{u}_1) \\ &= (\mathbf{u}_1 + \mathbf{r}_1)^T (\mathbf{u}_1 + \mathbf{r}_1) - 2\mathbf{p}_1^T (\mathbf{u}_1 + \mathbf{r}_1) + l_{11}^2 \end{aligned} \quad (\text{A.1})$$

Cette fois-ci, contrairement au PGI, \mathbf{u}_1 est connu en fonction des données :

$$\mathbf{u}_1 = \begin{bmatrix} 0 \\ l_{11} \cos \theta_1 \\ l_{11} \sin \theta_1 \end{bmatrix}, \quad (\text{A.2})$$

alors que \mathbf{p}_1 est une fonction de l'inconnue α :

$$\mathbf{p}_1 = \begin{bmatrix} 0 \\ l_{11} \cos \alpha \\ l_{11} \sin \alpha \end{bmatrix}. \quad (\text{A.3})$$

En substituant l'équation (A.3) dans l'équation (A.1) et en réarrangeant les termes, on obtient une équation dont la seule inconnue est α , qui s'écrit sous la forme :

$$D_1 \cos \alpha + E_1 \sin \alpha + F_1 = 0, \quad (\text{A.4})$$

avec :

$$\begin{aligned} D_1 &= 2(\mathbf{u}_1 + \mathbf{r}_1)^T \mathbf{f}_{11} l_{11} \\ E_1 &= 2(\mathbf{u}_1 + \mathbf{r}_1)^T \mathbf{f}_{21} l_{11} \\ F_1 &= (\mathbf{u}_1 + \mathbf{r}_1)^T (\mathbf{u}_1 + \mathbf{r}_1) - l_{21}^2 + l_{11}^2 \end{aligned}$$

où :

$$\mathbf{f}_{11} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \quad \text{et} \quad \mathbf{f}_{21} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}.$$

En utilisant le même développement qu'à la section 2.2.1, on obtient alors deux solutions pour l'angle α , qui sont :

$$\alpha = 2 \arctan \left(\frac{E_1^2 \pm \sqrt{E_1^2 + D_1^2 - F_1^2}}{F_1 - D_1} \right) \quad (\text{A.5})$$

Une de ces deux solutions correspond bien à la solution triviale pour laquelle $\alpha = \theta_1$.

A.1.2 Solutions pour l'angle β

On procède cette fois-ci à la fermeture géométrique de la deuxième chaîne cinématique, pour laquelle l'inconnue est \mathbf{p}_2 , qui dépend à la fois de α et β . Pour extraire l'inconnue β , on projette ce vecteur dans un repère qu'on nomme R_α tourné de l'angle α autour de X_b par rapport au repère R_b . Son expression dans ce repère est alors :

$$[\mathbf{p}_2]_{R_\alpha} = \begin{bmatrix} -l_{12} \cos \beta \\ 0 \\ l_{12} \sin \beta \end{bmatrix}. \quad (\text{A.6})$$

Alors, en utilisant les mêmes développements que précédemment, on a les solutions pour l'angle β :

$$\beta = 2 \arctan \left(\frac{E_2^2 \pm \sqrt{E_2^2 + D_2^2 - F_2^2}}{F_2 - D_2} \right) \quad (\text{A.7})$$

où

$$\begin{aligned} D_2 &= 2(\mathbf{Q}_x(\alpha)^T(\mathbf{u}_2 + \mathbf{r}_2))^T \mathbf{f}_{12} l_{12} \\ E_2 &= 2(\mathbf{Q}_x(\alpha)^T(\mathbf{u}_2 + \mathbf{r}_2))^T \mathbf{f}_{22} l_{12} \\ F_2 &= (\mathbf{u}_2 + \mathbf{r}_2)^T (\mathbf{u}_2 + \mathbf{r}_2) - l_{22}^2 + l_{12}^2 \end{aligned}$$

avec :

$$\mathbf{f}_{12} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \text{et} \quad \mathbf{f}_{22} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}.$$

A.1.3 Solutions pour l'angle γ

Finalement, la fermeture géométrique de la troisième chaîne cinématique, pour laquelle l'inconnue est \mathbf{p}_3 , permet de résoudre l'angle γ . Pour extraire l'angle γ , on projette cette fois-ci \mathbf{p}_3 dans le repère R_{ps} :

$$[\mathbf{p}_3]_{R_{ps}} = \begin{bmatrix} -l_{13} \cos \gamma \\ -l_{13} \sin \gamma \\ 0 \end{bmatrix}. \quad (\text{A.8})$$

Alors les solutions pour l'angle γ sont :

$$\gamma = 2 \arctan \left(\frac{E_3^2 \pm \sqrt{E_3^2 + D_3^2 - F_3^2}}{F_3 - D_3} \right) \quad (\text{A.9})$$

où

$$\begin{aligned} D_3 &= 2(\mathbf{Q}_{ps}^T(\mathbf{u}_3 + \mathbf{r}_3))^T \mathbf{f}_{13} l_{13} \\ E_3 &= 2(\mathbf{Q}_{ps}^T(\mathbf{u}_3 + \mathbf{r}_3))^T \mathbf{f}_{23} l_{13} \\ F_3 &= (\mathbf{u}_3 + \mathbf{r}_3)^T (\mathbf{u}_3 + \mathbf{r}_3) - l_{23}^2 + l_{13}^2 \end{aligned}$$

avec :

$$\mathbf{f}_{13} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \text{et} \quad \mathbf{f}_{23} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

A.2 Conversions entre les conventions d'orientation

Deux différentes conventions sont utilisées pour décrire l'orientation du mécanisme dans ce mémoire :

- La convention d'Euler XYZ

Elle est utilisée pour décrire le problème géométrique inverse et la cinématique du simulateur car cette convention correspond aux angles réellement réalisés par les articulations du mécanisme. Ainsi, les équations de position et de vitesse sont très simples, et en grande partie découplées.

- La convention de *Tilt & Torsion*

Elle est utilisée pour décrire l'espace de travail prescrit du simulateur, car les frontières de cet espace de travail correspondent à des valeurs constantes des angles de *Tilt & Torsion*.

On peut donc tirer avantage des deux différentes conventions, mais il faut connaître quelle est la conversion entre ces deux conventions pour une même orientation du mécanisme. Cette conversion a été calculée à l'aide du logiciel MAPLE, et le programme utilisé est commenté dans cette section.

A.2.1 Principe de la conversion

La matrice de rotation Q est indépendante de la représentation de l'orientation choisie, elle doit donc avoir les mêmes termes pour une orientation donnée, peu importe la convention choisie. Il suffit d'égaliser les termes de cette matrice obtenue par les deux conventions pour une orientation arbitraire pour connaître la conversion entre les deux représentations.

Soit $Q1$ la matrice de rotation en fonction des angles de *Tilt & Torsion* :

```
> Q1 := matrix(3,3,[cos(phi)*cos(theta)*cos(sigma-phi)-sin(phi)*sin(sigma-phi)
, -cos(phi)*cos(theta)*sin(sigma-phi)-sin(phi)*cos(sigma-phi),
cos(phi)*sin(theta),
sin(phi)*cos(theta)*cos(sigma-phi)+cos(phi)*sin(sigma-phi),
-sin(phi)*cos(theta)*sin(sigma-phi)+cos(phi)*cos(sigma-phi),
sin(phi)*sin(theta), -sin(theta)*cos(sigma-phi),
sin(theta)*sin(sigma-phi), cos(theta)]);
```

$$Q1 := \begin{bmatrix} \cos(\phi) \cos(\theta) \cos(-\sigma + \phi) + \sin(\phi) \sin(-\sigma + \phi), \\ \cos(\phi) \cos(\theta) \sin(-\sigma + \phi) - \sin(\phi) \cos(-\sigma + \phi), \cos(\phi) \sin(\theta) \\ \sin(\phi) \cos(\theta) \cos(-\sigma + \phi) - \cos(\phi) \sin(-\sigma + \phi), \\ \sin(\phi) \cos(\theta) \sin(-\sigma + \phi) + \cos(\phi) \cos(-\sigma + \phi), \sin(\phi) \sin(\theta) \\ [-\sin(\theta) \cos(-\sigma + \phi), -\sin(\theta) \sin(-\sigma + \phi), \cos(\theta)] \end{bmatrix}$$

Soit $Q2$ la matrice de rotation en fonction des angles de la convention XYZ. $Q2$ est le produit des trois matrices de rotation élémentaires autour des axes X, Y, et Z d'angles respectifs α , β et $\gamma2$:

```
> Qalpha:= matrix(3,3,[1,0,0,0,cos(alpha),-sin(alpha),0,sin(alpha),cos(alpha)]):
> Qbeta:= matrix(3,3,[cos(beta),0,sin(beta),0,1,0,-sin(beta),0,cos(beta)]):
> Qgamma2:=matrix(3,3,[cos(gamma2),-sin(gamma2), 0,sin(gamma2),cos(gamma2),0,0,0,1])
```

> Q2:=evalm(Qalpha*Qbeta*Qgamma2);

Q2 :=

[cos(β) cos(γ 2) , -cos(β) sin(γ 2) , sin(β)]

[sin(α) sin(β) cos(γ 2) + cos(α) sin(γ 2) , -sin(α) sin(β) sin(γ 2) + cos(α) cos(γ 2) ,
-sin(α) cos(β)]

[-cos(α) sin(β) cos(γ 2) + sin(α) sin(γ 2) , cos(α) sin(β) sin(γ 2) + sin(α) cos(γ 2) ,
cos(α) cos(β)]

Pour connaître la conversion entre les deux conventions, on égale chacun des termes des matrices Q1 et Q2 :

> Eq1:=cos(phi)*cos(theta)*cos(sigma-phi)-sin(phi)*sin(sigma-phi)

=cos(beta)*cos(gamma2):

> Eq2:=sin(phi)*cos(theta)*cos(sigma-phi)+cos(phi)*sin(sigma-phi)

=sin(alpha)*sin(beta)*cos(gamma2)+cos(alpha)*sin(gamma2):

> Eq3:=-sin(theta)*cos(sigma-phi)

=-cos(alpha)*sin(beta)*cos(gamma2)+sin(alpha)*sin(gamma2):

> Eq4:=-cos(phi)*cos(theta)*sin(sigma-phi)-sin(phi)*cos(sigma-phi)

=-cos(beta)*sin(gamma2):

> Eq5:=-sin(phi)*cos(theta)*sin(sigma-phi)+cos(phi)*cos(sigma-phi)

=-sin(alpha)*sin(beta)*sin(gamma2)+cos(alpha)*cos(gamma2):

> Eq6:=sin(theta)*sin(sigma-phi)

=cos(alpha)*sin(beta)*sin(gamma2)+sin(alpha)*cos(gamma2):

> Eq7:=cos(phi)*sin(theta)=sin(beta):

> Eq8:=sin(phi)*sin(theta)=-sin(alpha)*cos(beta):

> Eq9:=cos(theta)=cos(alpha)*cos(beta):

On obtient don un ensemble de 9 équations que l'on appelle Eq :

> Eq:={Eq1,Eq2,Eq3,Eq4,Eq5,Eq6,Eq7,Eq8,Eq9}:

A.2.2 Résolution si les inconnues sont les angles de la convention XYZ

> Inconnues:={alpha,beta,gamma2}:

On résoud les inconnues grâce au système d'équations redondant :

> Sol:=simplify(solve(Eq,Inconnues),trig);

$$\begin{aligned} Sol := & \{ \beta = \arctan(\cos(\phi) \sin(\theta), \%1), \alpha = \arctan\left(-\frac{\sin(\phi) \sin(\theta)}{\%1}, \frac{\cos(\theta)}{\%1}\right), \\ \gamma2 = & \arctan\left(\frac{(\cos(\phi)^2 \cos(\theta) \sin(\sigma) - \cos(\phi) \cos(\theta) \cos(\sigma) \sin(\phi) + \sin(\phi) \cos(\sigma) \cos(\phi) + \sin(\sigma) - \sin(\sigma) \cos(\phi)^2)/\%1, (\cos(\phi)^2 \cos(\theta) \cos(\sigma) + \cos(\phi) \cos(\theta) \sin(\sigma) \sin(\phi) - \sin(\phi) \sin(\sigma) \cos(\phi) + \cos(\sigma) - \cos(\phi)^2 \cos(\sigma))/\%1}{\%1}, \right. \\ \%1 := & \left. \text{RootOf}(-Z^2 - 1 + \cos(\phi)^2 - \cos(\theta)^2 \cos(\phi)^2, \text{label} = _L2) \right\} \end{aligned}$$

A.2.3 Résolution si les inconnues sont les angles de *Tilt & Torsion*

> Inconnues:={phi,theta,sigma}:

On résoud les inconnues grâce au système d'équations redondant :

> Sol:=simplify(solve(Eq,Inconnues),trig);

$$\begin{aligned} Sol := & \left\{ \theta = \arctan(\%1, \cos(\alpha) \cos(\beta)), \sigma = \arctan\left(\frac{\cos(\alpha) \sin(\gamma2) + \cos(\beta) \sin(\gamma2) + \sin(\alpha) \sin(\beta) \cos(\gamma2)}{\cos(\alpha) \cos(\beta) + 1}, \right. \right. \\ & \left. \left. -\frac{-\cos(\alpha) \cos(\gamma2) - \cos(\beta) \cos(\gamma2) + \sin(\alpha) \sin(\beta) \sin(\gamma2)}{\cos(\alpha) \cos(\beta) + 1}\right), \right. \\ \phi = & \left. \arctan\left(-\frac{\sin(\alpha) \cos(\beta)}{\%1}, \frac{\sin(\beta)}{\%1}\right)\right\} \\ \%1 := & \text{RootOf}(-Z^2 - 1 + \cos(\alpha)^2 \cos(\beta)^2, \text{label} = _L1) \end{aligned}$$

On remarque que pour cette résolution, contrairement à la précédente, le dénominateur appelé %1 peut s'annuler. En fait, le dénominateur s'annule lorsque simultanément α et β sont nuls, ce qui correspond à une valeur nulle de tilt θ . On se trouve alors dans une configuration que l'on appelle une singularité de la représentation par les angles de *Tilt & Torsion*. En effet, dans une telle configuration où le tilt est nul, le premier angle ϕ est indéterminé car cet angle est utilisé pour placer l'axe de rotation du tilt.

Si la rotation du tilt est nulle, alors cet angle ϕ peut prendre n'importe quelle valeur, et n'a aucun effet. En pratique, lors d'une trajectoire, on donnera à ϕ la même valeur qu'au point précédent dans un tel cas, afin d'éviter les discontinuités.

Ce phénomène n'apparaît pas avec la convention d'angles d'Euler XYZ car la singularité de représentation de cette convention est située hors de l'espace atteignable du mécanisme (mais elle existe aussi).

A.3 Cinématique de la membrure d_2

Pour déterminer les variables cinématiques de cette membrure en fonction des variables articulaires, on fait appel dans un premier temps aux techniques de résolutions cinématiques bien connues pour les manipulateurs sériels. On considère donc la patte 2 comme un manipulateur sériel à trois degrés de liberté dont l'effecteur est d_2 , utilisé pour placer le point P_2 dans l'espace. On résout les équations de position et de vitesse en faisant appel au formalisme de Hartenberg-Denavit, qui permet une notation ainsi qu'un développement mathématique systématiques.

A.3.1 Paramètres HD de la patte 2

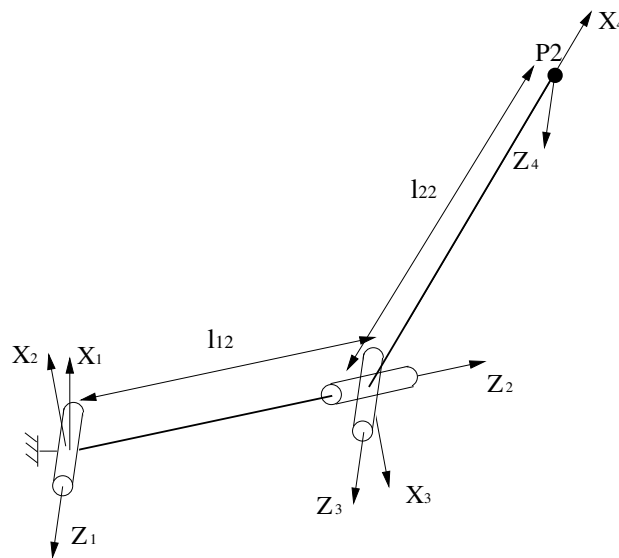


FIG. A.1 – Schématisation de la patte 2.

En utilisant la notation de Hartenberg-Denavit, on numérote les membres de 0 à 3 :

- le membre 0 est la base,
- le membre 1 est la membrure proximale de la patte,
- le membre 2 est le membre situé entre les deux rotoïdes du joint de Cardan (qui n'est pas physique ici),
- le membre 3 est la membrure distale.

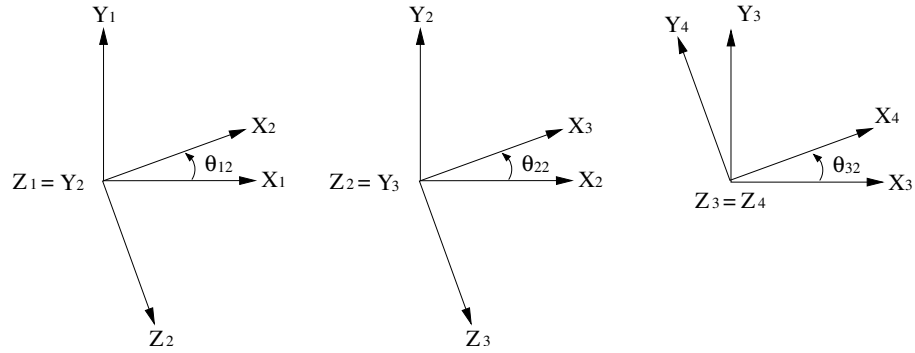


FIG. A.2 – Repères de la patte.

On lie ensuite, comme montré à la figure A.1, un repère cartésien (X_j, Y_j, Z_j) au $(j-1)$ ième membre, et ce suivant une convention prédéfinie. L'orientation de ces repères est illustrée à la figure A.2.

Grâce à cette modélisation de la patte 2, on peut écrire les paramètres de Hartenberg-Denavit de cette chaîne cinématique ouverte :

j	a_j	b_j	α_j	θ_j
1	0	0	$\frac{\pi}{2}$	θ_{12}
2	0	l_{12}	$\frac{\pi}{2}$	θ_{22}
3	l_{22}	0	0	θ_{32}

TAB. A.1 – Paramètres de Hartenberg-Denavit pour la patte 2.

A.3.2 Angles et jacobienne passifs

En utilisant les paramètres de Hartenberg-Denavit, on peut décrire la position et l'orientation de chacune des membrures, et ainsi développer de manière systématique le problème géométrique inverse et les relations de vitesses de la patte.

A.3.2.1 Angles passifs

On appelle \mathbf{Q}_j la matrice de rotation du j -ième au $(j+1)$ -ième repère, et \mathbf{a}_j le vecteur position reliant l'origine du j -ième repère à celle du $(j+1)$ -ième. Exprimées dans le j -ième repère, ces quantités s'expriment en fonction des paramètres de Hartenberg-Denavit de la façon suivante :

$$[\mathbf{Q}_j]_j = \begin{bmatrix} c_{\theta_j} & -c_{\alpha_j} s_{\theta_j} & s_{\alpha_j} s_{\theta_j} \\ s_{\theta_j} & c_{\alpha_j} c_{\theta_j} & -s_{\alpha_j} c_{\theta_j} \\ 0 & s_{\alpha_j} & c_{\alpha_j} \end{bmatrix} \quad (\text{A.10})$$

et

$$[\mathbf{a}_j]_j = \begin{bmatrix} a_j c_{\theta_j} \\ a_j s_{\theta_j} \\ b_j \end{bmatrix} \quad (\text{A.11})$$

D'après cette définition des vecteurs \mathbf{a}_j , leur somme est égale au vecteur $\mathbf{p}_2 - \mathbf{r}_2$. En écrivant cette égalité dans le repère 2, on a :

$$\mathbf{Q}_2 [\mathbf{a}_3]_3 + [\mathbf{a}_2]_2 = \mathbf{w}, \quad \text{avec} \quad \mathbf{w} = \mathbf{Q}_1^T (\mathbf{p}_2 - \mathbf{r}_2) \quad (\text{A.12})$$

où $\mathbf{p}_2 - \mathbf{r}_2$ correspond au vecteur reliant l'origine du repère 1 O_1 au point P_2 exprimé dans le repère 1, et \mathbf{Q}_1 n'est fonction que de θ_2 , qui est la position articulaire, donc le vecteur \mathbf{w} de l'équation (A.12) est connu.

Le terme de gauche de cette équation s'exprime lui en fonction des angles passifs :

$$\mathbf{Q}_2 [\mathbf{a}_3]_3 = \begin{bmatrix} l_{22} \cos \theta_{22} \cos \theta_{32} \\ l_{22} \sin \theta_{22} \cos \theta_{32} \\ l_{22} \sin \theta_{32} \end{bmatrix} \quad (\text{A.13})$$

et

$$[\mathbf{a}_2]_2 = \begin{bmatrix} 0 \\ 0 \\ l_{12} \end{bmatrix} \quad (\text{A.14})$$

En remplaçant les expressions (A.13) et (A.14) dans la relation (A.12), on peut résoudre les deux angles passifs de la patte 2 en fonction des données :

$$\theta_{32} = \arcsin \left(\frac{\mathbf{w}^T \mathbf{Z}_2}{l_{22}} \right) \quad (\text{A.15})$$

et

$$\theta_{22} = \arctan 2 \left(\frac{\mathbf{w}^T \mathbf{Y}_2}{l_{22} \cos \theta_{32}}, \frac{\mathbf{w}^T \mathbf{X}_2}{l_{22} \cos \theta_{32}} \right) \quad (\text{A.16})$$

A.3.2.2 Cinématique

Une fois les angles passifs déterminés, les relations de vitesses sont obtenues en utilisant les résultats sur les manipulateurs sériels, et on a :

$$[\dot{\mathbf{p}}_2]_1 = \mathbf{J}_2 \dot{\boldsymbol{\theta}}_2 \quad (\text{A.17})$$

avec

$$\mathbf{J}_2 = \left[\mathbf{e}_1 \times \mathbf{b}_1, \mathbf{e}_2 \times \mathbf{b}_2, \mathbf{e}_3 \times \mathbf{b}_3 \right] \quad (\text{A.18})$$

où

$$\mathbf{e}_j = [\mathbf{Z}_j]_1 \quad \text{et} \quad \mathbf{b}_j = \sum_{k=j}^3 [\mathbf{a}_k]_1 \quad j = 1, 2, 3 \quad (\text{A.19})$$

Considérant désormais l'effecteur de ce manipulateur, on détermine ses paramètres cinématiques en fonction des variables articulaires $\dot{\boldsymbol{\theta}}_2$ en utilisant les relations :

$$\begin{aligned} \boldsymbol{\omega}_{d2} &= \mathbf{J}_{d2} \dot{\boldsymbol{\theta}}_2 \\ \dot{\mathbf{s}}_{d2} &= \mathbf{K}_{d2} \dot{\boldsymbol{\theta}}_2 \end{aligned} \quad (\text{A.20})$$

avec

$$\mathbf{J}_{d2} = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3] \quad (\text{A.21})$$

$$\mathbf{K}_{d2} = [\mathbf{e}_1 \times (\mathbf{c}_{d2} + \mathbf{u}_2), \mathbf{e}_2 \times \mathbf{c}_{d2}, \mathbf{e}_3 \times \mathbf{c}_{d2}] \quad (\text{A.22})$$

Annexe B

Données techniques

B.1 Ressorts pour le simulateur à taille réelle

Les ressorts pour le simulateur ont été conçus sur mesure pour cette application par l'entreprise Ressorts Liberté située à Montmagny. Les caractéristiques obtenues pour ces ressorts sont détaillées à la figure B.1, où les unités de chaque donnée sont indiquées entre parenthèses.

SMI SPRING DESIGNER VERSION 2.0 HARDCOPY PRINT
To place your company name and other customizing data here, run SDSETUP

UNIVERSITE LAVAL

Extension	Cylind	Close Wound	Chrome Silicon	Round
Wire Dia (mm)	9.9822	Mean Dia (mm)	78.9180	Active Coils 39.0000 Total Coils 39.0000 Inactive Coils 0.0000
Rate (N/mm)	5.1336	Inside Dia (mm)	68.9357	
Spring Index C	7.9059	Outside Dia (mm)	88.9002	
Nat Freq (Hz)	7.2503			
Fatigue (cycles)	N/A	Body Length (mm)	399.2888	Addl Feed (mm) 0.0000
No Peen (cycles)	< 100K	Hook Length (mm)	137.4143	Devel Lngth (mm) 10165.05 Weight (kg) 6.2538

	Free	Point 1	Point 2	Hooks at Max Load
Available IT (N)	644.8500			
Load (N)	622.7200	742.3165	2693.074	Load (N) 2693.074
Length (mm)	536.7031	560.0000	940.0000	R1 (mm) 39.4590
Deflection (mm)		23.2969	403.2969	R2 (mm) 14.9733
Free Len / Deflection		23.0375	1.3308	
UNK Stress (MPa)	126	150	544	Bend (MPa) 1236
UNK Stress % of MTS	7.7	9.1	33.2	Bend % MTS 75.4
CORR Stress (MPa)	149	178	646	Tors (MPa) 748
CORR Stress % of MTS	9.1	10.9	39.4	Tors % MTS 45.6

NORMAL MANUFACTURING TOLERANCES: (All other dimensions ref. only)
Coil Diameter: +/- 0.70 (mm)
End (hook) planes: Consult your springmaker.

NOTES:
Hook radius R1 based on mean diameter
Hook radius R2 assumed to be 1.5 wire diameters
Excessive hook bending stress
Excessive hook torsional stress

FIG. B.1 – Caractéristiques données par le fournisseur.

B.2 Ressorts pour le prototype

Les ressorts d'extension pour le prototype en plastique, qui possèdent une taille et une raideur beaucoup plus classiques que ceux nécessaires pour le simulateur à taille réelle, ont pu être commandé sur catalogue à la compagnie Associated Spring Raymond. Les caractéristiques de ces ressorts sont résumées au tableau B.2.

Diamètre extérieur D (mm)	Diamètre du fil ϕ (mm)	Longueur libre l_0 (mm)	Tension initiale F_0 (N)	Longueur maximale L_1 (mm)	Tension à L1 F_1 (N)	Raideur k (N/mm)
9.14	0.79	76.20	1.47	246.89	16.41	0.088

TAB. B.1 – Caractéristiques des ressorts du prototype.

B.3 Moteurs pour le prototype



GM8724S017

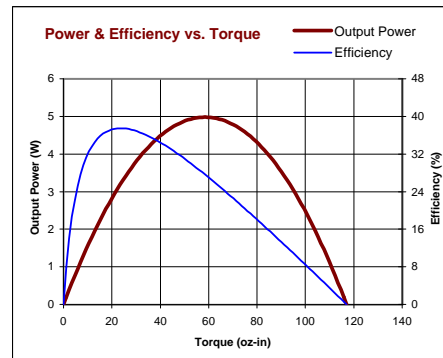
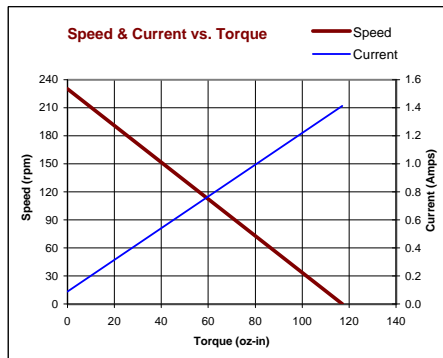
Lo-Cog® DC Servo Gearmotor



Assembly Data	Symbol	Units	Value
Reference Voltage	E	V	24
No-Load Speed	S _{NL}	rpm (rad/s)	230 (24.1)
Continuous Torque (Max.) ¹	T _C	oz-in (N-m)	42 (2.9E-01)
Peak Torque (Stall) ²	T _{PK}	oz-in (N-m)	117 (8.3E-01)
Weight	W _M	oz (g)	11.3 (320)
Motor Data			
Torque Constant	K _T	oz-in/A (N-m/A)	6.18 (4.36E-02)
Back-EMF Constant	K _E	V/krpm (V/rad/s)	4.57 (4.36E-02)
Resistance	R _T	Ω	17.0
Inductance	L	mH	9.35
No-Load Current	I _{NL}	A	0.09
Peak Current (Stall) ²	I _P	A	1.41
Motor Constant	K _M	oz-in/√W (N-m/√W)	1.49 (1.05E-02)
Friction Torque	T _F	oz-in (N-m)	0.35 (2.5E-03)
Rotor Inertia	J _M	oz-in-s ² (kg-m ²)	2.3E-04 (1.6E-06)
Electrical Time Constant	τ _E	ms	0.54
Mechanical Time Constant	τ _M	ms	14.7
Viscous Damping	D	oz-in/krpm (N-m-s)	0.020 (1.4E-06)
Damping Constant	K _D	oz-in/krpm (N-m-s)	1.6 (1.1E-04)
Maximum Winding Temperature	θ _{MAX}	°F (°C)	311 (155)
Thermal Impedance	R _{TH}	°F/watt (°C/watt)	70.5 (21.4)
Thermal Time Constant	τ _{TH}	min	10.7
Gearbox Data			
Reduction Ratio			19.5
Efficiency ³			0.87
Maximum Allowable Torque		oz-in (N-m)	175 (1.24)
Encoder Data			
Channels			3
Resolution		CPR	500

- Included Features**
- 2-Pole Stator
 - Ceramic Magnets
 - Heavy-Gauge Steel Housing
 - 7-Slot Armature
 - Silicon Steel Laminations
 - Stainless Steel Shaft
 - Copper-Graphite Brushes
 - Diamond Turned Commutator
 - Motor Ball Bearings
 - Output Ball Bearing
 - Wide Face Gears
- Customization Options**
- Alternate Winding
 - Sleeve or Ball Bearings
 - Modified Output Shaft
 - Custom Cable Assembly
 - Special Brushes
 - EMI/RFI Suppression
 - Alternate Gear Material
 - Special Lubricant
 - Optional Encoder
 - Fail-Safe Brake

1 - Specified at max. winding temperature at 25°C ambient without heat sink. 2 - Theoretical values supplied for reference only.
 3 - Effective gearbox efficiency for this unit improved by use of ball bearings.



All values are nominal. Specifications subject to change without notice. Graphs are shown for reference only.

© 2001 Pittman.

PITTMAN, 343 Godshall Drive, Harleysville, PA 19438, Phone: 877-PITTMAN, Fax: 215-256-1338, E-mail: info@pittmanner.com, Web Site: www.pittmanner.com

FIG. B.2 – Caractéristiques des moteurs.

B.4 Simulateur à six degrés de liberté

La plate-forme à six degrés de liberté de type Gough-Stewart utilisée pour les comparaisons de sensations de mouvements est la plate-forme de recherche de l'UTIAS (University of Toronto Institute for Aerospace Studies). La figure B.3 montre l'installation de cette plate-forme, et le tableau B.4 résume ses limites de déplacements, de vitesses et d'accéléérations. Les limites de rotations sont données pour les angles ϕ , θ et ψ , qui correspondent respectivement aux angles de roulis, tangage et lacet de la convention d'angles d'Euler $Z(\psi)Y(\theta)X(\phi)$ couramment utilisée en aéronautique.



FIG. B.3 – Simulateur à 6 degrés de liberté de l'UTIAS.

Linéaire		Angulaire	
x	± 0.65 (m)	ϕ	$\pm 20.8(^{\circ})$
y	± 0.59 (m)	θ	$\pm 21.3(^{\circ})$
z	± 0.52 (m)	ψ	$\pm 20.0(^{\circ})$
v_{max}	0.80 (m/s)	ω_{max}	34.4($^{\circ}$ /s)
a_{max}	10.0 (m/s ²)	α_{max}	400($^{\circ}$ /s ²)

TAB. B.2 – Débattements et limites physiques de l'architecture de l'UTIAS.

Annexe C

Glossaire des termes informatiques utilisés

Ce glossaire donne une liste alphabétique des termes et sigles informatiques employés dans ce mémoire, ainsi que leur définition. Cette liste a été rédigée en se basant sur le Grand Dictionnaire Terminologique ainsi que le dictionnaire informatique [37]. La traduction française des termes utilisés a été faite dans le mémoire dans la mesure du possible en tenant compte de l'usage des mots. Pour les termes anglais employés couramment en français, la traduction proposée par l'Office de la langue française est fournie dans la définition même si elle n'est pas utilisée en pratique.

- **Bus PCI** (Peripheral Component Interconnect, c'est à dire interconnexion de composants périphériques) :

Bus ultra-rapide défini par Intel pour la carte mère des ordinateurs IBM et compatibles. L'interconnexion de composants périphériques permet l'intégration sur la carte mère de plusieurs organes d'entrée-sortie généralement déportés sur des cartes d'extension, dont les contrôleurs de disque.

- **DLL** (Dynamically Linked Library, c'est à dire bibliothèque de liens dynamiques) : Ensemble de fonctions formant une bibliothèque, liée et accessible à une application au moment de son exécution (contrairement à une bibliothèque statique dont les fonctions sont liées à l'application au moment de la compilation).

- **FIFO** (First In First Out, c'est à dire premier entré, premier sorti) : Canal de communication auquel on a attribué un nom précis, permettant d'échanger des données entre des applications tournant sur des ordinateurs distants, et ce, à travers le réseau. À la différence du message transmis par un canal de communication normal, le message expédié par FIFO peut être lu par tout programme connaissant le nom du canal. Le nom anglais vient du mode de traitement des données selon lequel les données entrées en premier seront les premières à être lues.

Termes proposés par l'office de la langue française : canal nommé, tube fifo.

- **Mémoire vive** : Mémoire primaire d'un ordinateur, rapidement accessible, dans laquelle les données peuvent être lues, écrites ou effacées. On dit de la mémoire vive qu'elle est une mémoire à accès aléatoire, parce qu'elle autorise l'écriture ou la lecture des données selon leurs adresses.

- **Module du noyau Linux** :

Les modules du noyau Linux sont des programmes qui peuvent être chargés ou déchargés dans le noyau à n'importe quel moment. Ils étendent les fonctionnalités de ce noyau sans nécessiter de redémarrer le système. Par exemple, un type de module correspond aux pilotes de périphériques, qui permettent au noyau d'accéder aux périphériques du système. Un autre exemple qui nous intéresse ici est celui du module d'ordonnancement, qui est un des modules constituant le noyau temps réel RTLinux, et qui permet de donner des priorités aux tâches temps réel.

- **Noyau :**

Principale composante d'un système d'exploitation, qui demeure en permanence en mémoire principale et exécute des tâches fondamentales et essentielles au bon fonctionnement d'un ordinateur, telles que la gestion des tâches et de la mémoire, la gestion des échanges avec les périphériques, l'exécution des applications et l'attribution des ressources.

- **Plugin :**

Logiciel d'application complémentaire à un logiciel existant, qui lui apporte des améliorations et ne peut fonctionner qu'en association avec lui.

Termes proposés par l'office de la langue française : compagnon, additif (mot-valise formé à partir des mots addition et logiciel).

- **Port série :**

C'est le port, ou interface, qui peut être utilisé pour une communication sérielle, pour laquelle seulement 1 bit est transmis à la fois. Les mots à transmettre sont donc auparavant codés puis ensuite décodés à leur arrivée pour pouvoir être interprétés, les bus internes des ordinateurs étant de type parallèle. La vitesse de transfert des données est très limitée à cause de cette structure sérielle, elle s'exprime en bits par seconde ou bauds. La plupart des ports série sur les ordinateurs se conforment aux standard RS-232 ou RS-422.

- **Port USB :**

C'est un port utilisant le standard USB (Universal Serial Bus) pour réaliser des communications entre un ordinateur et un périphérique extérieur à travers un câble utilisant une transmission bi-sérielle.

Un seul port permet de relier en série, à l'unité centrale de traitement de l'ordinateur, plus de 120 périphériques.

- **Protocole TCP/IP** (Transmission Control Protocol/Internet Protocol) :

Ensemble des protocoles de communication utilisés dans Internet et permettant de gérer la circulation des données dans le réseau, tout en assurant le bon échange de ces données entre un point et un autre du réseau. Bien qu'on utilise le terme singulier protocole TCP/IP pour désigner cette notion, TCP/IP n'est pas constitué d'un seul protocole, mais d'une famille de protocoles. Dans l'usage, aussi bien en français qu'en anglais, on considère donc le terme protocole TCP/IP comme un

collectif.

- **SDK** (Software Development Kit, ou en français trousse de développement logiciel) :

Ensemble d'outils logiciels conçus pour aider les programmeurs à créer des programmes destinés à tourner sur une plate-forme donnée.

- **Socket** :

Interface de connexion permettant à des applications de communiquer d'un ordinateur à l'autre, indépendamment du type de réseau utilisé. Les sockets fonctionnent toujours par paire. Une interface est utilisée pour le poste client et l'autre, pour le serveur. Techniquement, l'interface de connexion est une adresse Internet qui combine l'adresse IP et le numéro de port. Le terme anglais socket est couramment utilisé en français. De plus, comme l'équivalent français de socket n'est pas encore définitivement fixé, on trouve de nombreuses appellations concurrentes au terme interface de connexion (proposé par l'Office de la langue française) comme prise de communication, point de communication et mécanisme d'échange.

- **Système d'exploitation temps-réel** :

C'est un système d'exploitation pour lequel la validité des calculs ne dépend pas seulement des résultats logiques de ces calculs, mais aussi du temps auquel ces résultats sont produits. Si les contraintes temporelles du système ne sont pas respectées, une erreur est produite. Le temps-réel est donc une propriété des systèmes d'exploitation, et dans un système temps-réel la valeur d'un calcul dépend de la pertinence temporelle de sa réponse. Par exemple, un calcul complété trop tard a une valeur diminuée, voire plus de valeur du tout, alors qu'un calcul complété trop tôt n'aura pas de valeur additionnelle.

Des problèmes peuvent apparaître lorsqu'il y a une compétition pour partager les ressources disponibles entre différentes activités. Un des aspect important des systèmes temps-réel est donc la détermination des priorités des activités, pour permettre à toutes d'être complétées en temps voulu. On parle de temps-réel dur lorsque tout calcul n'a aucune valeur si son résultat arrive trop tard, entraînant des conséquences catastrophiques pour le système. Un système de temps-réel dur est donc un système pour lequel tous les calculs doivent impérativement être réalisé en temps voulu.

Annexe D

Listes des fichiers et fonctions programmés

D.1 Liste des plugins développés pour X-Plane

Pour l'ordinateur B, qui est l'ordinateur maître, trois différents plugins peuvent être utilisés pour modifier le logiciel :

1. un premier plugin envoie les données de vol aux ordinateurs C et D, et reconfigure la touche F5 pour qu'en appuyant sur celle-ci on démarre ou arrête à tout moment la simulation,

2. Un deuxième plugin empêche X-Plane de dessiner la vue extérieure, ce qui permet d'avoir uniquement le tableau de bord en sortie graphique, et une fréquence de calcul du modèle mathématique très élevée.
3. Un dernier plugin optionnel permet d'enregistrer les commandes lors d'une manœuvre, et de les rejouer à la place des commandes réelles. Ainsi, la même manœuvre peut être réalisée plusieurs fois, et des tests peuvent être conduits sur cette manœuvre. Aussi, cette fonctionnalité peut être utilisée pour réaliser des simulations passives, où le pilote dans le simulateur subit la manœuvre qu'une autre personne a réalisé préalablement (c'est un mode très utilisé si le simulateur est employé dans un système d'amusement où un film est joué par exemple).

Pour l'ordinateur C, qui est l'ordinateur esclave, deux plugins sont utilisés :

1. un premier plugin permet de dessiner la vue extérieure à partir des données reçues de B, et ainsi évite au logiciel de calculer de nouveau le modèle mathématique de l'avion. Ainsi, le logiciel X-Plane qui est exécuté sur l'ordinateur C est entièrement dédié aux calculs graphiques de la vue extérieure,
2. finalement, comme on l'a déjà mentionné, un plugin fourni par l'entreprise Elumens permet de projeter cette vue extérieure sur l'écran sphérique en traitant la vue de la caméra de X-Plane.

D.2 Génération des mouvements du simulateur

D.2.1 Application Linux

L'application Linux est l'exécutable nommé Principal, et compilé à partir du fichier Main.c, qui est la fonction principale, et qui utilise la bibliothèque de fonctions SimuLibrairieControle.a, elle même compilée à partir de différents fichiers. Voici la liste de ces fichiers, avec les fonctions publiques et privées qu'ils contiennent par ordre alphabétique dans chaque catégorie :

- **acquisition.c**

Fonctions publiques :

AcquisitionFichier

ArretAcquisitionFichier

InitAcquisitionFichier

Fonctions privées : aucune

- **connexion.c**

Fonctions publiques :

BoucleDeVeille

Deconnexion

InitConnexion

RecupereDonnees

TestPasDeTemps

VerifieConnexion

Fonctions privées : aucune

- **consigne.c**

Fonctions publiques :

CalculConsigneXPlane

InitConsigne

Fonctions privées :

ConvAngles

Pgi

VerifLimitesArticulaires

- **interface.c**

Fonctions publiques :

EcrisMemoirePartagee

FermeFifo

FermeMemoirePartagee

FifoDemandeTemps

FifoNvelleConsigne

FifoStopConsigne

FifoStopProgramme

OuvreFifo

OuvreMemoirePartagee

Fonctions privées : aucune

- **mathvect.c**

Fonctions publiques : aucune

Fonctions privées :

toute la bibliothèque mathématique vectorielle développée au laboratoire de robotique par Lionel Birglen, qui est utilisée par les autres fonctions de la bibliothèque SimuLibrairieControle, mais qui n'est pas rendue publique à l'application principale.

- **washout.c**

Fonctions publiques :

InitFiltre

PasseEnW

WashoutFilters

Fonctions privées : aucune

D.2.2 Module RTLinux

Le module RTLinux ModuleSimu.o est créée à partir d'un fichier principal nommé ModuleMain.c, qui contient les fonctions principales, les définitions de tâches temps-réel, et le chronomètre, et de deux fichiers secondaires, le premier étant ConsigneVirtuelle.c, qui contient les fonctions nécessaires à l'élaboration de consignes virtuelles par extrapolation si les consignes venant de X-Plane ne rencontrent pas les contraintes de temps, et le deuxième étant la bibliothèque mathvect.c. Voici la liste de ces fichiers, avec les fonctions publiques et privées qu'ils contiennent par ordre alphabétique dans chaque catégorie :

- **ModuleMain.c**

Fonctions publiques :

chrono

cleanup_module

init_module

msg_nvelle_csg

test_csg

transmission_csg

Fonctions privées :

ArretChronometre

DemarrageChronometre

Initialisation

InitialisationChronometre

TestResolutionEncodeurs

ValeurActuelleChronometre

- **ConsigneVirtuelle.c**

Fonctions publiques :

CalculConsigneVirtuelle

Fonctions privées :

Pgi

- **mathvect.c**

La même bibliothèque de mathématique vectorielle que celle utilisée par l'application Linux.