

École doctorale n° 432 : Sciences des Métiers de l'Ingénieur

Doctorat ParisTech

THÈSE

pour obtenir le grade de docteur délivré par

l'École Nationale Supérieure d'Arts et Métiers

Spécialité " Automatique "

présentée et soutenue publiquement par

Baris AYKENT

le 16 décembre 2013

Etude des lois de commande de la plateforme de simulation de conduite et influence sur le mal de simulateur

Directeur de thèse : **Frédéric MERIENNE**

Co-encadrement de la thèse : **Damien PAILLOT, Andras KEMENY**

Jury

M. Daniel MESTRE, Directeur de recherche, CRVM, Université de la Méditerranée
M. Jean-Christophe POPIEUL, Professeur, LAMIH, Université de Valenciennes
M. Jean-Michel BOUCHEIX, Professeur, LEAD, Université de Bourgogne
M. Frédéric MERIENNE, Professeur, Le2i, Arts et Métiers ParisTech
M. Damien PAILLOT, Maître de conférences, Le2i, Université de Bourgogne
M. Andras KEMENY, Professeur associé, Renault, Arts et Métiers ParisTech

Président
Rapporteur
Rapporteur
Examinateur
Examinateur
Examinateur

**T
H
È
S
E**

Remerciements

Je tiens à remercier très sincèrement Frédéric MERIENNE, professeur des universités et directeur de l'institut Carnot ARTS d'avoir accepté la direction de cette thèse et de m'avoir guidé tout au long de ce travail.

Je tiens à exprimer ma plus profonde sympathie à Frédéric MERIENNE, Damien PAILLOT et Andras KEMENY qui ont encadré mes travaux. Je les remercie non seulement pour la qualité scientifique de leurs encadrements, mais aussi pour leurs disponibilités.

Je suis extrêmement reconnaissant envers tous les membres de mon jury pour l'attention qu'ils ont bien voulu porter à ce travail.

Daniel MESTRE pour m'avoir fait l'honneur de présider ce jury.

Les professeurs des universités Jean-Michel BOUCHEIX et Jean-Christophe POPIEUL pour avoir accepté le charge de rapporteur.

Je tiens à remercier Yves BALLAY qui m'a montré les rudiments pour utiliser le système de mesure EMG du Laboratoire INSERM U1093, Université de Bourgogne, Faculté des Sciences du Sport (UFR Staps)

Je voudrais également remercier le professeur des universités Thierry POZZO le directeur du Laboratoire INSERM U1093, Université de Bourgogne, Faculté des Sciences du Sport (UFR Staps) qui nous a permis d'emprunter système mesure EMG.

Ces remerciements ne pourraient se terminer sans que j'exprime mes plus profonds remerciements à Géraldine ROUX, qui m'a supporté durant ces trois années pour mes problèmes administratifs et pour mes achats de matériels et d'équipements.

Enfin, je tiens à remercier sincèrement ma famille qui m'a soutenue pendant cette période.

Table des matières

Remerciements	3
Table des matières	5
Table des figures	9
Tables des tableaux	13
Introduction	15
Chapitre I Simulation de conduite	19
1.1. Objectif	19
1.2. Historique	19
1.3. Technologie	20
1.3.1. Principes des simulateurs statiques et dynamiques	20
1.3.2. Structure de Simulateur statique.....	20
1.3.3. Structure du simulateur dynamique	22
1.3.4. Exemples de simulateurs dynamiques.....	25
1.4. Evolution des simulateurs	29
Chapitre II Problématique scientifique	31
2.1. Contexte	31
2.2. Questions de recherche	32
2.3. Approche proposée	33
2.4. Analyse du mal du simulateur	33
2.4.1. Analyse subjective.....	34
2.4.2. Analyse objective	35
2.5. Contrôle de la plateforme	37
2.5.1. Algorithme classique.....	37
2.5.2. Algorithme adaptatif	38
2.5.3. Algorithme optimal.....	39
2.5.4. Algorithme prédictif (MPC Model Predictive Control).....	40
Chapitre III Mal du simulateur	43
3.1. Les différentes théories du mal du simulateur	44
3.2. Organes impliqués	46
3.2.1. Système Vestibulaire (oreille interne)	46
3.2.2. Système visuel.....	47
3.2.3. Système somato-sensoriel.....	48
3.3. Symptômes	49
3.4. Facteurs	49
3.5. Critères du mal du mouvement/simulateur	51
3.5.1. L'approche de 'MSDV (Motion Sickness Dose Value)-IR (Illness Rating)'	51
3.5.2. L'approche de 'VDV (Vibration Dose Value)'	53

3.5.3.	L'approche de la stabilité posturale	54
3.5.4.	L'approche du seuil de la perception du mouvement.....	55
3.6.	Mesures objectives	58
3.6.1.	Mesure dynamiques neuromusculaires (EMG)	58
Chapitre IV	Contrôle de la plateforme	63
4.1.	Algorithmes mis en œuvre.....	63
4.2.	Algorithme classique	64
4.3.	Algorithme adaptatif (Discrete Time Model Adaptive Control – MRAC).....	65
4.3.1.	Définition de l'algorithme adaptatif	65
4.3.2.	Modélisation de l'algorithme adaptatif.....	65
4.4.	Algorithme optimal (Linear Quadratic Regulator-LQR).....	70
4.4.1.	Définition de l'algorithme optimal.....	70
4.4.2.	Modélisation de l'algorithme optimal.....	71
4.5.	Algorithme hybride (MRAC+LQR).....	72
Chapitre V	Conditions Expérimentales.....	75
5.1.	Scénario 1 (Chicane).....	75
5.2.	Scénario 2 (Country Road)	76
5.3.	Scénario 3 (Rugosité sinusoïdale)	77
5.4.	Scénario 4 (Courbure de la route)	77
5.5.	Scénario 5 (Braquage sinusoïdal)	79
5.6.	Scénario 6 (Country Road pour 15 min)	80
Chapitre VI	Effet de la restitution inertielle	83
6.1.	Introduction	83
6.2.	L'approche proposée	84
6.3.	Protocole.....	85
6.4.	Sujets.....	87
6.5.	L'analyse des données.....	87
6.6.	Résultats et discussion.....	88
6.7.	Conclusion	91
Chapitre VII	Influence de l'engagement du sujet en conduite simulée.....	95
7.1.	Introduction	95
7.2.	Approche proposée.....	96
7.2.1.	Engagement du sujet en simulation de conduite	96
7.2.2.	Guide visuel pour le passager.....	97
7.2.3.	Evaluation du mal du simulateur.....	97
7.2.4.	Questions de recherche.....	99
7.3.	Méthode expérimentale	99
7.3.1.	Protocole	99
7.3.2.	Mesures.....	100
7.3.3.	Panel de sujets	101
7.4.	Analyse du questionnaire	102
7.5.	Etudes expérimentales	103
7.5.1.	Analyses de données	103
7.5.2.	Résultats et discussions.....	103
7.6.	Conclusion et perspectives.....	110
Chapitre VIII	Influence du contrôle de la plateforme.....	113
8.1.	Introduction	113
8.2.	Conditions expérimentales	113
8.3.	Comparaison entre l'algorithme classique et l'algorithme MRAC	114
8.3.1.	Sujets.....	114
8.3.2.	L'analyse des données.....	114

8.3.3. Résultats et discussion	115
8.3.4. Interaction vestibulaire et neuromusculaire.....	122
8.3.5. Conclusion	125
8.4. Comparaison algorithme classique et LQR	125
8.4.1. Data Denoising.....	125
8.4.2. Résultats et Discussion.....	126
8.4.3. Conclusion	132
Conclusion Générale et Perspective	135
Bibliographie.....	139
Liste des publications réalisées au cours de cette thèse	151
Articles publiés dans des revues avec comité de lecture.....	151
Communications publiées avec actes et comité de lecture	152
Annexe - Statistiques	155
Annexe - Code C++	159

Table des figures

Fig. I-1 : Différence entre simulateurs de conduite statique et dynamique.....	20
Fig. I-2. : Simulateur statique à Arts et Métiers ParisTech pendant un essai.....	21
Fig. I-3. : Simulateur d'ECO ₂ pendant un essai à Hannover Messe.....	21
Fig. I-4. : Simulateur dynamique de conduite SAAM (Institut Image).	22
Fig. I-5. : Structure du simulateur dynamique (SAAM) contrôle en boucle fermée.....	24
Fig. I-6. : Matériel informatique structure de SAAM [APM+11].....	24
Fig. I-7. : Connexion des ordinateurs du simulateur de conduite (SAAM).	25
Fig. I-8. : Simulateur de conduite ULTIMATE Renault.....	25
Fig.I-9. : Simulateur de conduite MARS.....	26
Fig. I-10. : Simulateur de conduite NADS-1 [NAD10, Lev12].....	27
Fig. I-11. : Simulateur de conduite Daimler.....	28
Fig. I-12 : Evolution des simulateurs en fonction de leur complexité.....	29
Fig. II-1 : Motion cueing algorithm (algorithme de pilotage).....	32
Fig. II-2 : Approche proposée.....	33
Fig. II-3 : Mesure au niveau vestibulaire.....	35
Fig. II-4 : Analyse d'EMG.....	36
Fig. II-5 : Matériels informatiques et outils logiciels pour la mesure de l'instabilité posturale	36
Fig. II-6 : Algorithme classique.....	37
Fig. II-7 : Algorithme adaptatif.....	38
Fig. II-8 : Algorithme optimal.....	39
Fig. III-1 : Organes vestibulaires a) oreille interne – b) système vestibulaire.....	46
Fig. III-2 : Tilt coordination [SRVB11a].....	47
Fig. III-3 : Champ visuel.....	48
Fig. III-4 : Sensibilité somatique.....	49
Fig. III-5 :GUI pour le logiciel PostureWin.....	55
Fig. III-6 : Schéma des accélérations d'un véhicule.....	56
Fig. III-7 : Mesure au niveau vestibulaire.....	57
Fig. III-8 : Analyse d'EMG.....	58
Fig. III-9 : Branchement des électrodes pour l'analyse des EMG.....	59
Fig. III-10 : Muscles des bras choisis pour l'acquisition.....	59
Fig. III-11 : Les muscles des sternocleidomastoïdiens [Gil07].....	59

Fig. IV-1 : Détails des filtres algorithmes classiques	64
Fig. IV-2 : Algorithmes Classique et le MRAC	65
Fig. IV-3 : Algorithmes classiques et optimal LQR.....	70
Fig. IV-4. : Conflit sensoriel d'inertiel-vestibulaire	70
Fig. IV-5 : Roulis de la plateforme en fonction du temps et selon les algorithmes	72
Fig. V-1 : Manœuvre de changements de couloir dans SCANeR studio.....	75
Fig. V-2 : Country scenario d'OKTAL SCANeR studio et trajet.....	76
Fig. V-3 : IR longitudinal mesuré au niveau vestibulaire - scénario «Country Road» pour un sujet	76
Fig. V-4 : Aspect du trajet (clothoïde) Rayon de courbure de 50 m et longueur 200 m	77
Fig. V-5 : Comparaison IR latérale vestibulaire avec ECO ₂ et SAAM (statique).....	78
Fig. V-6 : Vitesse du véhicule scénario de braquage sinusoïdal.....	79
Fig. V-7 : Déplacement du véhicule lors du scénario de braquage sinusoïdal	79
Fig. V-8 : La vitesse et le centre de gravité du véhicule (CG) XY trajectoire pendant l'expérience.	80
Fig. V-9 : Restitution (ligne verte) mouvements visuo-vestibulaire)	80
Fig. VI-1 : Procédure de corrélation entre les mesures objectives et subjectives	84
Fig. VI-2 : Chicane de l'OTAN [GBE+98, YG11].....	85
Fig. VI-3 : Cercle de corrélation et valeurs propres de l'analyse en composantes principales pour les conditions statiques et dynamiques	89
Fig. VII-1 : Rotation de la tête pour diminuer les conflits visuo-vestibulaire.....	96
Fig. VII-2 : Induction de mouvements visuo-vestibulaires par guide visuel (ligne verte).	97
Fig. VII-3 : Vitesse du véhicule et trajectoire pendant l'expérience.	100
Fig. VII-4 : Système d'acquisition de la dynamique de la tête pendant les phases de conduite	101
Fig. VII-5 : Angles de roulis pour un passager non-guidé	104
Fig. VII-6 : Angles de roulis pour un passager guidé	104
Fig. VII-7 : Angles de roulis pour un conducteur.....	105
Fig. VII-8 : Angles de roulis et écart type	106
Fig. VII-9 : Niveau des accélérations latérales vestibulaires pour 1 sujet	107
Fig. VII-10 : Moyenne cumulée des VDV au niveau vestibulaire	108
Fig. VII-11 : Moyenne de la VDV latérale (intervalle de confiance de 95%)	109
Fig. VIII-1 : Trajectoire exécutée pour le test.....	114
Fig. VIII-2 : Mouvement de la tête et l'analyse EMG	115
Fig. VIII-3 : Angles de volant (°).....	115
Fig. VIII-4 : EMG RMS total puissance (mV ² /Hz) et vestibulaire vitesse roulis (°/s)	116

Fig. VIII-5 : Interaction dynamique de véhicule/Système Vestibulaire/Système neuromusculaire	120
Fig. VIII-6 : Les relations de conflit multi sensoriel avec le mal des transports	121
Fig. VIII-7 : Zone de déplacement latéral	122
Fig. VIII-8 : Analyse relation de RVT et des EMG RMS puissances maximales totales pour tous les sujets	124
Fig. VIII-9 : Analyse relation de RVT et des EMG RMS puissances moyennes totales pour tous les sujets	124
Fig. VIII-10 : Comparaison RVT pour l'algorithme classique et l'algorithme MRAC pour tous les sujets	125
Fig. VIII-11 : Procédure débruitage	126
Fig. VIII-12 : Intégration d'algorithme de la plateforme dans le simulateur	127
Fig. VIII-13 : Trajectoire du scénario conduit dans les expériences	128
Fig. VIII-14 : Profil de vitesse du véhicule pour des expériences du simulateur	128
Fig. VIII-15 : Interaction multi sensorielle dans le simulateur de conduite dynamique.....	129
Fig. VIII-16 : Analyse fréquentielle d'accélération roulis de la cabine.....	130
Fig. VIII-17 : Analyse fréquentielle d'accélération roulis de la plate-forme.....	131

Tables des tableaux

Tableau I-1 : Technologies mises en œuvre et modalités sensori-motrices en simulation de conduite	22
Tableau I-2 : Caractéristiques techniques du simulateur SAAM.....	22
Tableau II-1 : Questionnaire mal du simulateur	34
Tableau III-1 : Facteurs potentiels associés au mal du simulateur dans l’immersion virtuelle[Kol95]..	50
Tableau III-2 : Organes sensoriels concernés par la perception du mouvement [MAT09a, MAT09b]	51
Tableau III-3 : Principales méthodes de mesure du mal du simulateur.....	57
Tableau IV-1 : Facteur des déplacements en tridimensionnel selon les algorithmes	63
Tableau IV-2 : Paramètres de l’algorithme classique	64
Tableau V-1 : Bosses (ondulation sinusoïdale)	77
Tableau V-2 : Courbes.....	78
Tableau VI-1 : Questionnaire proposé aux sujets.....	85
Tableau VI-2 : Paramètres d’algorithme classique	86
Tableau VI-3 : L’inertie des valeurs propres et le pourcentage de la variance	89
Tableau VI-4 : Coefficient de Pearson (r) des données sub. et obj. en statique.....	90
Tableau VI-5 : Coefficient de Pearson (r) des données sub. et obj. en dynamique	90
Tableau VII-1 : Résultats de l’auto-évaluation.....	102
Tableau VII-2 : <i>Statistiques pour les angles de roulis</i>	105
Tableau VII-3 : <i>Table des différences bilatérales pour des angles de roulis</i>	106
Tableau VII-4 : <i>p-valeurs des angles de roulis</i>	106
Tableau VII-5 : Comparaison des accélérations vestibulaires latérales.....	107
Tableau VII-6 : Statistiques pour VDV lateral.....	109
Tableau VII-7 : Différences par paire pour VDV latéral.....	110
Tableau VII-8 : p-valeurs pour VDV latéral	110
Tableau VIII-1 : Interactions dynamique vestibulaire/muscle du bras – algorithme classique	118
Tableau VIII-2 : Interactions dynamique vestibulaire/muscle du bras – algorithme MRAC	119
Tableau VIII-3 : Zone de déplacement latéral pour les deux algorithmes	121
Tableau VIII-4 : Résultats basés accélération roulis cabine dans le domaine fréquentiel	130
Tableau VIII-5 : Résultats basés accélération roulis de la plate-forme dans le domaine fréquentiel...	132

Introduction

La Réalité Virtuelle (RV) est l'une des technologies les plus innovantes des années 90 en matière d'interfaces homme-machine. Déjà populaire dans les jeux d'arcades, les technologies de la réalité virtuelle ont déjà été appliquées dans de nombreux domaines tels que la formation, la médecine, l'architecture, l'astronomie, la visualisation de données complexes et la télé-opération.

Bien que cette nouvelle technologie soit très prometteuse, il existe de freins à sa diffusion en raison du malaise qu'elle peut provoquer chez certains sujets au cours d'une expérience d'immersion virtuelle. Des réactions similaires ont été observées dans les simulateurs de conduite et de vols militaires. Ce phénomène est appelé 'mal du simulateur' et il est similaire au 'mal des transports'. Il y a un lien direct entre le mal des transports et le mal du simulateur dans les environnements virtuels : les deux sont souvent induits par un décalage des informations provenant de la vision et du système vestibulaire. La littérature est abondante sur le mal du simulateur ainsi que sur le mal des transports. Bien que la plupart de la recherche consiste en l'étude du mal du simulateur pour des pilotes militaires et des simulateurs de vol, de nombreuses conclusions peuvent être directement applicables aux systèmes de réalité virtuelle. Ces résultats peuvent aider à identifier les facteurs potentiels impliqués dans le mal de simulateur ainsi qu'à proposer des moyens pour le minimiser.

L'objectif est d'identifier les facteurs impliqués dans le mal du simulateur dans des environnements virtuels afin que ce mal puisse être évité ou du moins réduit dans son amplitude et/ou sa durée. Pour atteindre cet objectif, beaucoup de documents liés au mal du simulateur, ainsi que du mal des transports, ont été étudiés. Certains documents sont spécifiquement applicables aux environnements virtuels, mais ces recherches sont encore rares en raison de la jeunesse de ce domaine.

Le premier chapitre de ce manuscrit présente les généralités de la simulation de conduite, l'historique et les différentes technologies utilisées. Ce chapitre introduit aussi les bases de la perception en milieu de simulation de conduite et définit la notion de conflit sensoriel.

Le second chapitre détaille la problématique scientifique. Le contexte de notre étude, les questions de recherche ainsi que l'approche proposée sont explicitées dans ce chapitre.

Le troisième chapitre développe les caractéristiques du mal du simulateur. Les organes impliqués dans ce phénomène ainsi que les mesures et indicateurs qui seront utilisés pour les expérimentations sont développés dans ce chapitre.

Le quatrième chapitre traite des différents algorithmes de pilotage de la plateforme. Les algorithmes présents dans la littérature et les différents algorithmes proposés dans le cadre de ces travaux y sont détaillés.

Le cinquième chapitre fait état des expérimentations réalisées au cours de ces trois années de thèse. Toutes les expérimentations ne sont pas développées dans ce document néanmoins elles font l'objet de publications.

Les chapitres six, sept et huit présentent les résultats obtenus en terme d'amélioration du mal du simulateur. Trois scénarios ont été traités :

Restitution inertielle (comparaison entre le fonctionnement statique et dynamique)

Implication du sujet (comparaison entre un sujet passif et un sujet actif)

Influence du contrôle de la plateforme (type d'algorithme et paramétrage).

Le neuvième chapitre conclut le document en faisant la synthèse des travaux et résultats obtenus.

Chapitre I

Simulation de conduite

1.1. Objectif

L'objectif de l'utilisation des simulateurs de conduite est de pouvoir mettre l'humain dans la boucle d'analyse et d'étudier les facteurs humains pendant les phases de développement des automobiles. L'avantage des simulateurs de conduite est de placer les sujets dans une configuration de sécurité totale. Ainsi, les situations accidentogènes peuvent être recréées et les comportements des conducteurs analysés [QD05].

1.2. Historique

L'histoire des simulateurs commence avant la deuxième guerre mondiale. Les premiers simulateurs sont les simulateurs de vol et ont été utilisés à des fins de formation. L'idée principale est de réduire les coûts opérationnels de l'utilisation de l'équipement réel. Deux approches principales ont été utilisées dans la conception des simulateurs de vol: la conception d'un simulateur haute fidélité (c'est à dire un simulateur très proche de la réalité) ou la conception d'un simulateur moins réaliste mais sans compromettre l'efficacité de la formation [RC80]. Des premières années [Tho13] jusqu'à une période récente [Log88], beaucoup de théories ont été élaborées sur le transfert des compétences acquises avec un simulateur vers des situations réelles, mais aucune d'entre elles n'a été validée comme meilleure.

Les premières études sur les simulateurs de conduite de type automobiles ont débuté à la fin des années 1950 et le premier simulateur a été opérationnel au début des années 1960 [Rob80]. Puis il y a eu une baisse des avancées et des développements des simulateurs automobiles au milieu des années 60 en raison de l'insuffisance des moyens informatiques et techniques de l'époque, mais qui a été surmonté à la fin des années 60. Une grande partie de la technologie a été développée par la National Aeronautics and Space Administration (NASA) pour soutenir son programme spatial. Ces améliorations ont été bénéfiques pour les techniques de simulation routière, en 1975 plusieurs «simulateurs» opéraient à travers les États-Unis (au moins 16 en utilisant des techniques différentes pour la génération du champ

visuel). Au même moment, deux simulateurs de conduite étaient déployés en Europe (SAAB et VW) [Rob80, Bla96].

Durant la dernière décennie, il y a eu une forte augmentation de l'utilisation des simulateurs de conduite pour la formation et la recherche dans le domaine du comportement du conducteur dans une situation de trafic grâce aux avancées technologiques et informatiques. Il était aussi crucial d'améliorer notre compréhension du comportement du conducteur et donc d'améliorer la sécurité du trafic mais sous des conditions expérimentales contrôlées. Dans le passé, ces environnements contrôlés ont souvent manqué de réalisme et leur relation avec les conditions réelles de conduite restait plutôt ténue [Bla96].

1.3. Technologie

1.3.1. Principes des simulateurs statiques et dynamiques

Les simulateurs de conduite peuvent être classés en deux catégories selon qu'ils soient statiques (sans excitation du système vestibulaire) ou dynamiques (utilisant une plateforme dynamique pour exciter le système vestibulaire en relation avec l'image). La Fig. I-1 montre les caractéristiques essentielles de ces différents types de simulateurs [Bla96].

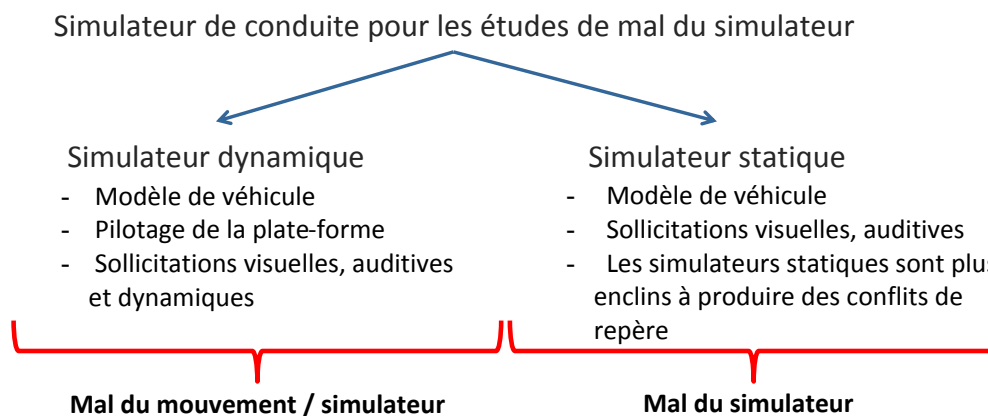


Fig. I-1 : Différence entre simulateurs de conduite statique et dynamique.

1.3.2. Structure de Simulateur statique

Le simulateur statique d'Arts et Métiers ParisTech est constitué d'un siège Playseat Alcantara et d'un Volant Logitech G27, composé des éléments de commande ci-dessous:

- Volant,
- Boîte de vitesse automatique,
- Pédales d'accélérateur et de frein.

La Fig. I-2 montre une vue d'ensemble du simulateur statique et vue en fonctionnement.

Les simulateurs statiques n'ont pas de système de restitution de mouvements ; ce qui signifie que le repérage de la conduite se fait à travers le visuel et l'information auditive uniquement. De ce fait, ils sont plus enclins à générer le mal du simulateur. Selon les théories de « conflits de repère » et du « réflexe vestibulo-oculaire (RVO) » [Kol95], le sujet qui conduit le simulateur statique a tendance à se sentir mal en l'absence de signal inertiel.



Fig. I-2. : Simulateur statique à Arts et Métiers ParisTech pendant un essai.

La Fig. I-3 montre le simulateur ECO₂ qui a été développé par la société OKTAL en partenariat avec Renault. L'application présentée est une application pour la formation à la conduite écologique.



Fig. I-3. : Simulateur d'ECO₂ pendant un essai à Hannover Messe.

1.3.3. Structure du simulateur dynamique

Le simulateur SAAM de l'institut image est un simulateur de conduite ayant une projection d'image permettant de couvrir un champ de vision important (environ 150°) et actionné par une plateforme dynamique de type plateforme Stewart. La Fig. I-4 montre une image de ce simulateur.



Fig. I-4. : Simulateur dynamique de conduite SAAM (Institut Image).

Le simulateur de conduite dynamique SAAM est composé de dispositifs permettant d'agir sur plusieurs modalités sensori-motrices de l'utilisateur. Le Tableau I-1 rappelle les technologies mises en œuvre et les modalités sensori-motrices sur lesquelles elles agissent. Le Tableau I-2 liste les caractéristiques techniques de la plateforme MOOG utilisée sur le simulateur SAAM.

Technologie	Modalité sensori-motrice
Vidéo-projecteurs et écran de visualisation	Vision
Plateforme dynamique de mouvement	Système vestibulaire
Haut-parleurs	Audition
Pédales actives et volant actif	Haptique

Tableau I-1 : Technologies mises en œuvre et modalités sensori-motrices en simulation de conduite

DOF	Déplacement	Vitesse	Accélération
<i>Tangage</i>	±22 deg	±30 deg/s	±500 deg/s ²
<i>Roulis</i>	±21 deg	±30 deg/s	±500 deg/s ²
<i>Lacet</i>	±22 deg	±40 deg/s	±400 deg/s ²
<i>Pilonement</i>	±0.18 m	±0.30 m/s	±0.5 g
<i>Cavalement</i>	±0.25 m	±0.5 m/s	±0.6 g
<i>Embardée</i>	±0.25 m	±0.5 m/s	±0.6 g

Tableau I-2 : Caractéristiques techniques du simulateur SAAM

L'ensemble est géré par le logiciel SCANeR Studio 1.1, développé par la société française OKTAL. Le logiciel permet de :

- Créer un terrain,
- Gérer la dynamique de la plateforme,
- Gérer les caractéristiques d'un véhicule (modèle dynamique et cinématique du véhicule),
- Gérer le trafic,
- Créer des modules, dans la thèse un module d'acquisition en temps réel a été implémenté,
- Gérer le son,
- Gérer les multiples canaux visuels (vue frontale, rétroviseurs intérieur et extérieurs),

Sur la base de la théorie Interface Homme Machine (IHM) [Eng01], le conducteur d'un simulateur est une partie de ce système en boucle fermée. Ainsi, les signaux dans l'environnement de simulation comprennent non seulement ceux qui viennent du simulateur lui-même, mais aussi ceux qui viennent du conducteur (système proprio-perceptifs, système vestibulaire, système neuromusculaire, etc.).

Les Fig. I-5, Fig. I-6 et Fig. I-7 décrivent le simulateur de conduite SAAM de l'Institut Image.

Le simulateur SAAM se compose des éléments suivants [APM+11] :

- Véhicule de série Renault Twingo 2 (Volant, pédales, boîte de vitesses automatique)
- Système de projection (3 projecteurs)
- Plateforme dynamique 6-DOF (hexapode MOOG)
- Système auditif pour la simulation du bruit du moteur et du trafic
- PCs pour contrôler le système.

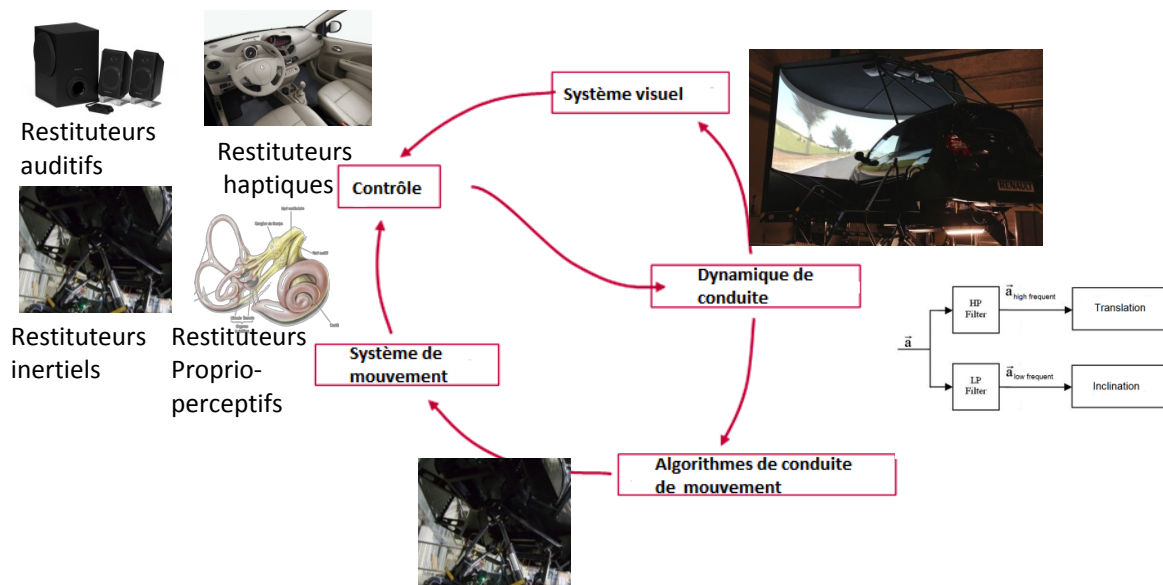


Fig. I-5. : Structure du simulateur dynamique (SAAM) contrôle en boucle fermée.

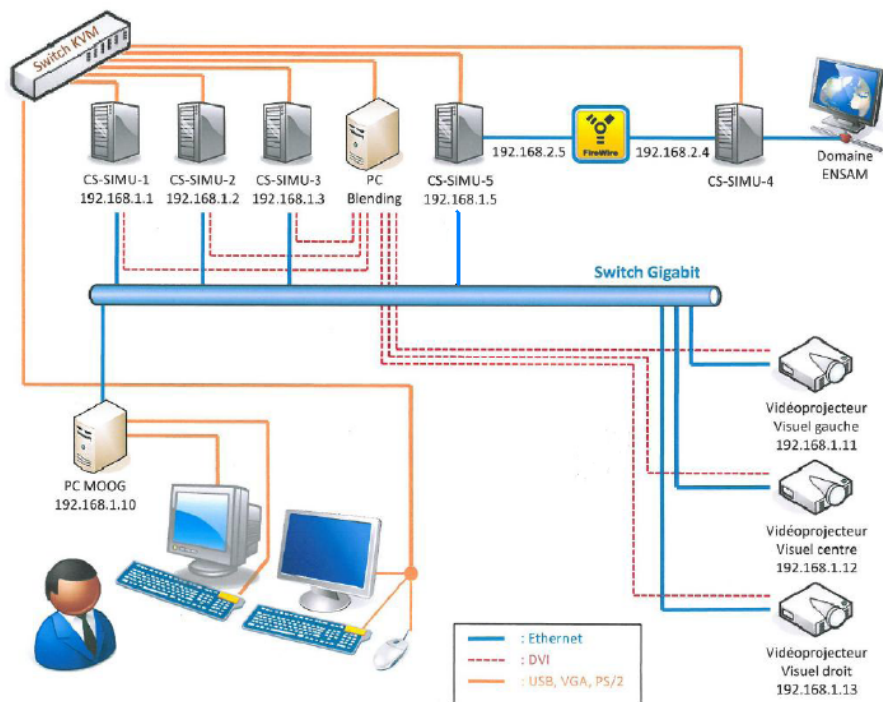


Fig. I-6. : Matériel informatique structure de SAAM [APM+11].

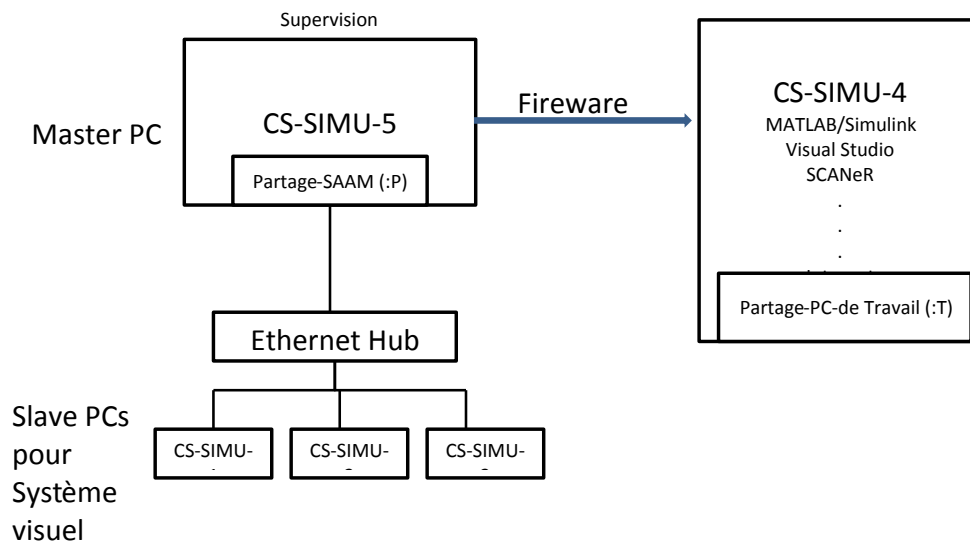


Fig. I-7. : Connexion des ordinateurs du simulateur de conduite (SAAM).

1.3.4. Exemples de simulateurs dynamiques

Renault Ultimate Simulateur

Le simulateur ULTIMATE (cf. Fig. I-8) utilise une plate-forme compacte à six axes montée sur un deuxième grand système bidirectionnel linéaire. Ce système permet des accélérations maximales de 7 m/s^2 dans les directions X et Y. Un écran cylindrique de 150° permet un affichage complet. Parallèlement, lorsque l'intégration des composants physiques est impossible ou difficile, le conducteur peut utiliser un casque de réalité virtuelle (HMD) qui permet de conduire dans un cockpit entièrement virtuel [Kem04].



Fig. I-8. : Simulateur de conduite ULTIMATE Renault

MARS à l'IFAS

Le simulateur de conduite MARS (Modular Simulateur recherche d'automobile, Fig.I-9) à l'IFAS est basé sur un système hexapode commandé électriquement, qui est supporté par trois vérins pneumatiques pour supporter la charge statique. Ce système est monté sur un rail qui permet un déplacement jusqu'à 1,5 mètres. Sa plage d'accélération va jusqu'à 7 m/s^2 [MAT09b, MAT09a].



Fig.I-9. : Simulateur de conduite MARS

Simulateur NADS

Le simulateur NADS-1 de l'université d'Iowa, Etats-Unis, (cf. Fig. I-10) est monté sur rails qui permettent un déplacement de 19.5 m en longitudinal et en latéral. La cabine embarque un écran cylindrique de 360° [NAD10, Lev12].

Ce sont les capacités uniques du système de mouvement qui le distinguent des autres simulateurs, permettant ainsi de reproduire avec précision des mouvements pour une accélération soutenue, des manœuvres de freinage, des mouvements à travers plusieurs voies de circulation, et l'interaction avec divers revêtements routiers [NAD10, Lev12].

À l'intérieur du dôme, la cabine est montée sur un plancher. Quatre vérins hydrauliques produisent des vibrations qui émulent les sensations de la route. Le dôme est monté sur un anneau qui peut le faire tourner autour de son axe vertical de 330 degrés dans chaque sens. L'ensemble est monté sur un hexapode hydraulique traditionnel, qui à son tour est monté sur

deux poutres à entraînement par courroie qui peuvent se déplacer indépendamment le long des axes X et Y. L'assemblage XY produit des accélérations latérales et longitudinales en se déplaçant de 19,5 mètres dans les deux directions [NAD10, Lev12].



Fig. I-10. : Simulateur de conduite NADS-1 [NAD10, Lev12].

Le système visuel est constitué de huit écrans à cristaux liquides (LCD) à l'intérieur du dôme. Le conducteur est immergé dans un environnement virtuel photo réaliste de 360 degrés. Une résolution plus élevée est utilisée dans le champ de vision principal pour une meilleure reconnaissance des caractères et pour réduire la fatigue oculaire. L'image est rafraîchie 60 fois par seconde [NAD10, Lev12].

Simulateur Toyota

Le simulateur de conduite de Toyota se compose de 12 degrés de liberté. Ce simulateur de conduite est essentiellement utilisé pour [MTS09, TOY07]:

L'analyse des caractéristiques de conduite et le développement de la technologie de sécurité active: Conduite d'une analyse des caractéristiques de conduite dans des conditions telles que la sensibilisation réduite de son environnement (l'endormissement au volant et la somnolence), l'inattention au danger (en regardant de gauche à droite et de ne pas avoir vérifié que la route est sûre), conduite avec facultés affaiblies (ivresse, fatigue ou maladie), ainsi que le développement de technologies de sécurité active pour réduire efficacement le nombre d'accidents.

Vérification de l'efficacité des technologies de sécurité active : Sur la base de ce qui précède, l'évaluation de l'efficacité et de la durabilité de l'intégration des systèmes d'alerte

conducteur et de véhicule de commande pour réduire le nombre d'accidents de la circulation, afin de vérifier les effets de la technologie de sécurité active.

Simulateur Daimler

Le simulateur (cf. Fig. I-11) possède sept degrés de liberté : les mouvements de plateforme haute en x, y et z, et tourne sur les trois axes (tangage, roulis et lacet). Le système de mouvement complet est monté sur le rail latéral, ce qui ajoute à la simulation des mouvements latéraux tels que les changements de voie qu'un hexapode classique ne peut pas simuler en raison des limites de course des vérins [Zee10, DAI10].

Le système de mouvement de la marque MOOG est entraîné le long du rail au moyen de moteurs linéaires [Zee10, DAI10].



Fig. I-11. : Simulateur de conduite Daimler

Les caractéristiques principales sont [Zee10]:

- Système de rail de guidage linéaire d'une longueur de 28 m,
- Coussins d'air pour le mouvement linéaire, moins de frottement entrefer inférieur à 40 μm
- Capacité dynamique de charge entre 200 kN et 400 kN
- Accélération 1g, vitesse max. 10 m/s, 212 kN de force, modulable en longueur
- Espace de déplacement: + 1.4/-1.3 m longitudinal, $\pm 1,3$ m latéral, vertical $\pm 1,0$ m $\pm 38^\circ$ de lacet, tangage $+24^\circ / -19^\circ$, $\pm 20^\circ$ roulis
- Dôme sphérique, diamètre env. 7.5 m, hauteur env. 4.5 m

1.4. Evolution des simulateurs

La complexité des simulateurs est variable (dynamique, statique, ...) ainsi que leur performance et leur coût. La Fig. I-12 illustre les technologies en fonction de leur complexité et le sens de présence induit. Des ajouts spécifiques peuvent enrichir la perception lors de l'utilisation de ces simulateurs. Ces ajouts seront conditionnés et définis par les modalités perceptives du sujet principalement perception visuelle, vestibulaire et proprioceptive.

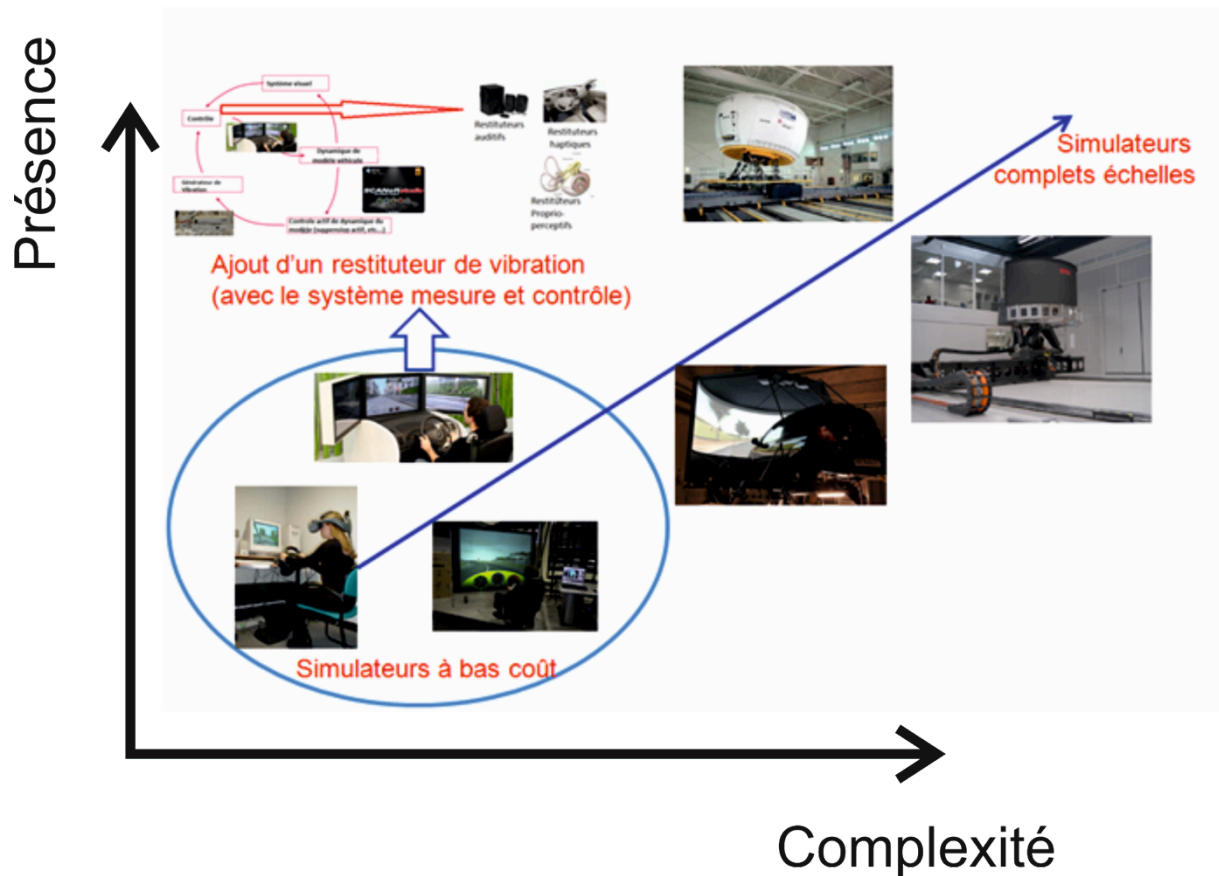


Fig. I-12 : Evolution des simulateurs en fonction de leur complexité

Chapitre II

Problématique scientifique

2.1. Contexte

Cette thèse a été développée en partenariat avec le centre technique de simulation (CTS) de Renault. Le CTS est déjà équipé de plusieurs simulateurs dynamiques et statiques avec des niveaux de finitions et de complexités différentes. Cependant, de nombreux verrous scientifiques et technologiques persistent. Cette thèse a pour objectif d'apporter des méthodes et des outils pour augmenter la pertinence et la fidélité des simulateurs dans le cadre de développement de véhicules. Les utilisateurs de ces simulateurs sont toujours sujets à des malaises lors des phases de conduite. Le mal du mouvement ou cinétose, également connu comme maladie du transport, est une situation dans laquelle un désaccord existe entre le mouvement perçu visuellement et la perception du système vestibulaire.

Selon l'origine du mal, il peut aussi être appelé mal de mer, mal du mouvement, mal de simulation, mal du simulateur ('simulator adaptation syndrome'), mal de l'air ou mal de l'espace [HB57, CRW86, Mon69].

Les limitations mécaniques des simulateurs ne permettent pas de produire une expérience de conduite comparable à une situation réelle, en particulier, lors des accélérations soutenues susceptibles de provoquer le mal du simulateur. Afin d'éviter/réduire ce phénomène, des algorithmes de pilotage de la plateforme peuvent être mis en œuvre sur le simulateur de conduite dynamique (cf. Fig. II-1). L'objectif principal de ces algorithmes est de maximiser la contribution du système de mouvement à la réalité du simulateur. C'est à dire que les algorithmes sont directement liés à la structure physique du simulateur. C'est cette même structure physique qui sera responsable de la limitation de la vitesse maximale, accélération maximale et de la course des actionneurs. La reproduction des sorties réelles (position, vitesse, accélération) du modèle dynamique une à une n'est donc pas possible. Un filtrage et une adaptation des signaux sont nécessaires. Le meilleur compromis doit être trouvé entre les

différents paramètres du modèle dynamique. La littérature propose plusieurs types d’algorithmes, ces algorithmes seront détaillés dans la suite de ce manuscrit et nous proposerons un nouveau type d’algorithme qui sera évalué au cours de ces travaux.

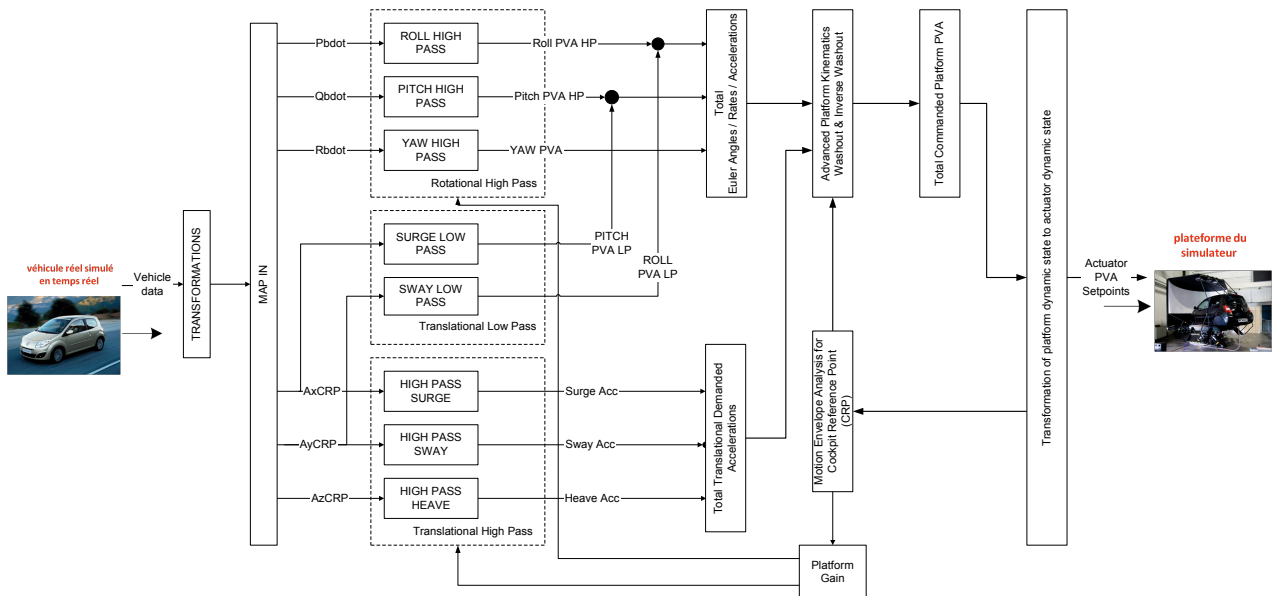


Fig. II-1 : Motion cueing algorithm (algorithme de pilotage)

2.2. Questions de recherche

Comme nous l’avons expliqué précédemment, le facteur limitant lors de l’utilisation des simulateurs est le mal engendré par le mouvement de la plateforme. Pour pouvoir mettre en place des méthodes et outils de réduction du mal du simulateur, il est important de bien comprendre ce dernier. Des outils d’analyse et de mesure du mal du simulateur sont requis pour cela. La littérature fait état de mesures subjectives (par questionnaires) et objectives (mesures physiologiques) qu’il convient de préciser et mettre en liaison avec la problématique posée.

Les paramètres sur lesquels il est possible d’agir pour réduire le mal du simulateur sont liés à la dynamique du simulateur qui s’exprime au niveau du modèle de véhicule, du mouvement de la plateforme ou du sujet (mouvement de la tête et des muscles).

Compte tenu de ce contexte scientifique, les questions de recherche qui sont adressées sont les suivantes :

- Quelles sont les méthodes et outils permettant d'évaluer le mal de simulateur en conduite simulée ?
- Est-il possible de mesurer le mal de simulateur de façon objective ?
- L'engagement du sujet dans la conduite influence-t-il le mal de simulateur ?
- Quelle est l'influence du contrôle de la plateforme sur le mal du simulateur ?

2.3. Approche proposée

L'approche proposée pour adresser les questions de recherche citées consiste à évaluer le mal de simulateur selon différentes configurations du conducteur et/ou du contrôle de la plateforme. Le schéma suivant résume l'approche proposée.

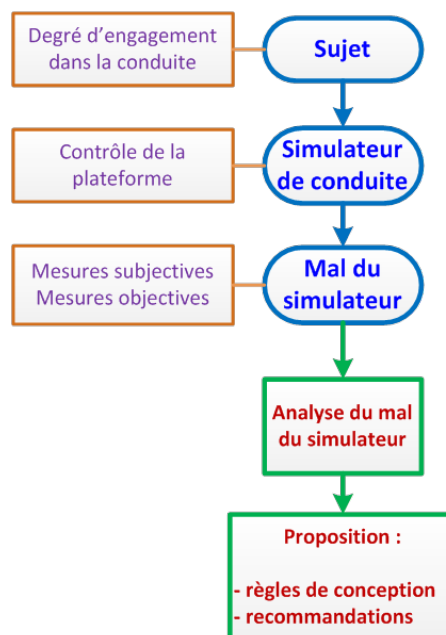


Fig. II-2 : Approche proposée

2.4. Analyse du mal du simulateur

Généralement, le mal du simulateur est évalué de manière subjective. Il est donc difficile d'utiliser des critères mathématiques pour valider ou non un algorithme. C'est pourquoi nous

avons décidé d'étudier le mal du simulateur selon deux approches. Dans un premier temps, de manière subjective puis dans un deuxième temps de manière objective, ces deux approches sont complémentaires et permettent de mieux comprendre le ressenti (méthode subjective)[KBA+88, KBD94, KBL+87, KFBL92, KHL90, KLBL93, KMKL10] et le vécu des différents sujets (méthode objective).

2.4.1. Analyse subjective

La mesure subjective consiste à réaliser des questionnaires auprès des sujets à l'issue des phases de conduite sur simulateur. Ces questions sont liées au ressenti du sujet. Les questions portent sur le degré de la nausée ressentie, des vomissements éventuels, des maux de tête, des étourdissements, de la peur, de l'inquiétude, ...etc. Le Tableau II-1 donne la liste des questions posées au sujet après chaque phase de conduite. Le questionnaire proposé dans cette thèse a été construit et modifié à partir des articles suivants [KLBL93, KMKL10], le questionnaire permet ainsi d'évaluer la désorientation (tendance à vomir, nausée, étourdissement). L'échelle des réponses se situe entre 1 et 10, la littérature propose généralement entre 1 et 4. Notre échelle permet d'avoir plus de choix de réponses possible.

Tableau II-1 : Questionnaire mal du simulateur

Questions	Expression des évaluations du questionnaire
Q1	Avez-vous eu tendance à vomir?
Q2	Avez-vous ressenti des nausées?
Q3	Avez-vous eu des sueurs froides?
Q4	Avez-vous ressenti des étourdissements?
Q5	Avez-vous ressenti de la fatigue oculaire?
Q6	Avez-vous eu des troubles oculaires?
Q7	Avez-vous eu des maux de tête?
Q8	Avez-vous ressenti une pression cérébrale?
Q9	Avez-vous eu peur?
Q10	Vous êtes-vous ennuyé?
Q11	L'expérience était-elle fatigante?
Q12	Etiez-vous anxieux/inquiet?

Le sujet doit répondre à chacune de ces questions par une valeur. Cette valeur doit refléter sa perception psychophysique de l'expérience (1: trop peu → 10: trop fort). Ensuite, ces valeurs seront analysées statistiquement (cf. Annexes).

2.4.2. Analyse objective

L'analyse des questionnaires étant une mesure subjective ; les résultats sont parfois peu significatifs. C'est pourquoi, nous proposons de mettre en œuvre des mesures objectives. Cinq mesures ont été choisies :

- accélération du véhicule (modèle dynamique),
- accélération et vitesse angulaire de la plate-forme,
- accélération et vitesse angulaire au niveau de la tête (dynamiques vestibulaires). La Fig. II-3 illustre la mesure de la dynamique de la tête des sujets lors des expériences en temps réel de simulation de conduite. Plus d'informations sur le capteur peut être obtenue à partir de [XSe10].



Fig. II-3 : Mesure au niveau vestibulaire

mesures électromyographiques. L'électromyographie (EMG) est une technique d'évaluation et d'enregistrement de l'activité électrique produite par les muscles squelettiques [emg13] EMG est réalisée à l'aide d'un instrument appelé électromyogramme, de produire un document appelé électromyogramme. Un électromyogramme détecte le potentiel électrique généré par les cellules musculaires [emg13] lorsque ces cellules sont électriquement ou neurologique activé. Les signaux peuvent être analysés pour détecter des anomalies médicales, le niveau d'activation, de l'ordre de recrutement ou d'analyser la biomécanique du mouvement humain ou animal.



Fig. II-4 : Analyse d'EMG

L'EMG (électromyogramme) a été mise en œuvre pour une partie particulière des essais afin d'expliquer la dynamique neuromusculaire des conducteurs lors des séances sur simulateur.

Il existe deux types d'EMG : invasive et non- invasive. Dans les expérimentations réalisées à l'Institut Image, la technique EMG non invasive a été utilisée. En outre, aucun stimulus électrique n'a été appliqué.

- La Fig. II-4 illustre la position des électrodes sur le bras du sujet pour une tentative de manœuvre. La stabilité posturale des sujets a été mesurée en utilisant un Stabilotest de Technoconcept (cf. Fig. II-5). Le système trace la position de la projection du centre de gravité des sujets. Les mesures ont été réalisées les yeux ouverts après les séances d'entraînement sur le simulateur. L'acquisition des données a été effectuée pendant 30 secondes à 40 Hz.

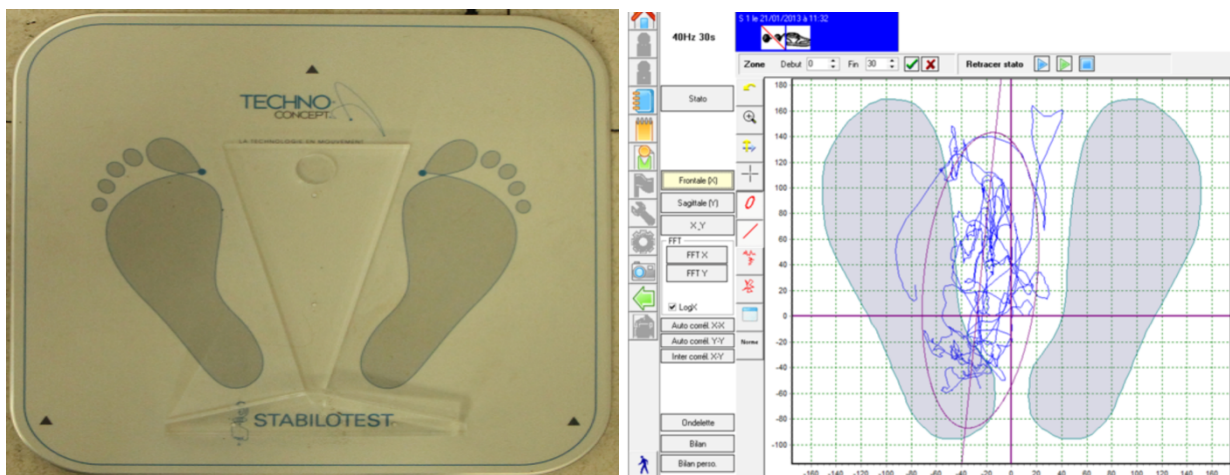


Fig. II-5 : Matériels informatiques et outils logiciels pour la mesure de l'instabilité posturale

De ces cinq mesures, les deux premières mesures ne sont pas liées au sujet. Elles dépendent soit du modèle dynamique soit de l'algorithme de calcul du mouvement de la

plateforme. Les trois dernières mesures sont intrinsèquement liées au corps du sujet ainsi que des déséquilibres induits par la phase de conduite. Ces trois derniers aspects sont cruciaux puisqu'ils vont permettre de quantifier la qualité d'un simulateur de conduite par rapport au modèle dynamique et algorithme de calcul du déplacement de la plateforme.

2.5. Contrôle de la plateforme

La littérature propose plusieurs modèles d'algorithmes pour piloter le mouvement de la plateforme :

- Classique,
- Adaptatif,
- Optimal,
- Prédictif.

2.5.1. Algorithme classique

Cet algorithme utilise un filtre linéaire passe-haut pour reproduire les accélérations transitoires de la plate-forme. L'accélération du véhicule simulé est passée à travers ce filtre pour éliminer les composants durables qui emmènent la plate-forme sur ses limites physiques. Le signal résultant est intégré deux fois pour calculer la position des actionneurs.

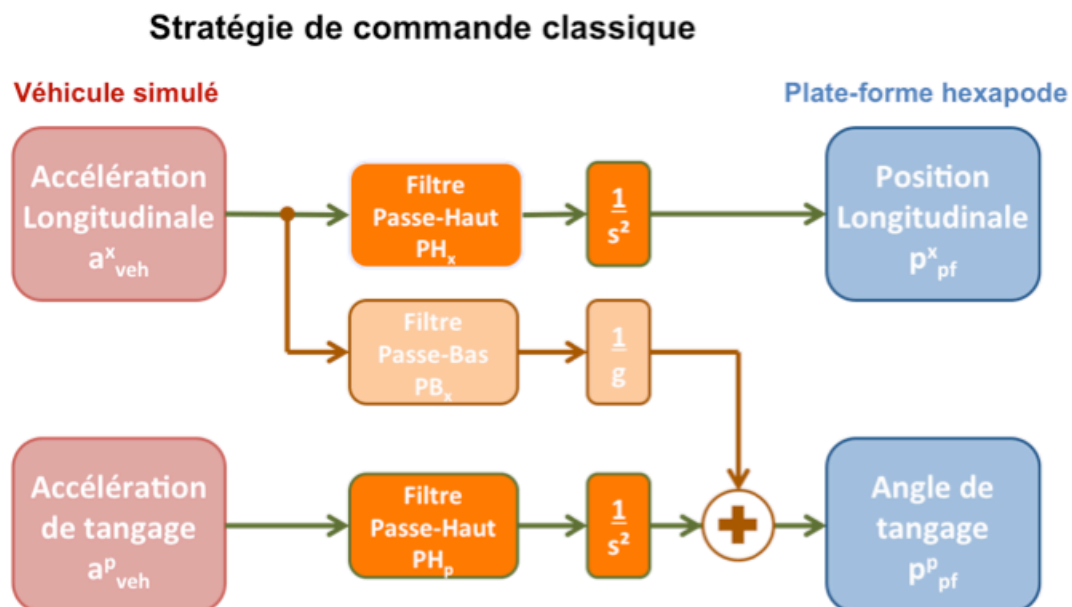


Fig. II-6 : Algorithme classique.

L'algorithme classique (cf. Fig. II-6) exprime le compromis dans le domaine fréquentiel. La commande plate-forme résulte d'un filtrage passe-haut de l'accélération du véhicule. En

supprimant la composante basse fréquence, les accélérations à haute fréquence sont restituées tandis que les déplacements de la plate-forme sont réduits [Ste65, RN85].

L'inconvénient principal est dû au contrôle en boucle ouverte, le conflit entre le véhicule et la dynamique de la plate-forme est plus élevé. Cet algorithme est approprié pour les simulateurs de vol.

2.5.2. Algorithme adaptatif

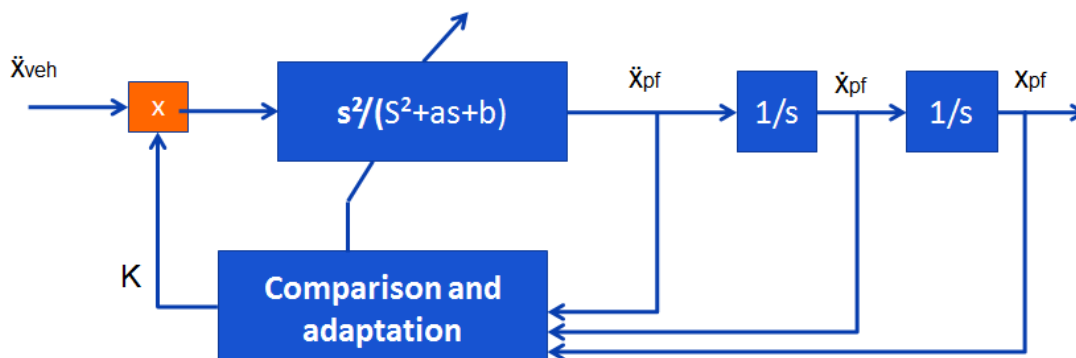
L'algorithme adaptatif (cf. Fig. II-7) est basé dans le domaine temporel et le domaine fréquentiel. Sa structure est dérivée de celle de l'algorithme classique, mais ici, les gains et les fréquences de coupure des filtres ne sont pas constants: à chaque étape de calcul, ils sont choisis pour réduire au minimum un coût qui recueille l'excursion de plate-forme et l'écart entre l'accélération du véhicule et l'accélération de la plate-forme [PMC+76].

Son avantage est d'être un contrôle en boucle fermée. En fonction de la commande adaptative, elle minimise la différence d'accélération au niveau de la plate-forme.

Il est basé sur la minimisation d'une fonction de coût contenant

- l'erreur d'accélération,
- les contraintes sur le déplacement de la plate-forme.

L'adaptation est réalisée en utilisant la méthode de la plus grande pente ('the steepest descent method').



(Nehaoua et al 2006)

$$\ddot{x}_{pf} = K \cdot \ddot{x}_{veh} - a \cdot \dot{x}_{pf} - b \cdot x_{pf}$$

K, a, b : paramètres d'adaptation

Fig. II-7 : Algorithme adaptatif.

Cet algorithme semble combler certaines pénuries du format classique. Dans cette approche, les filtres utilisés dans les algorithmes de pilotage classiques sont remplacés par des

filtres avec des gains variant dans le temps ainsi, la réponse est adaptée à toutes les situations. Par ailleurs, la possibilité de pouvoir régler les paramètres en fonctionnement permet de mieux exploiter le potentiel de la plateforme. [AL09].

2.5.3. Algorithme optimal

L'algorithme optimal résulte en une combinaison de filtres passe-haut et passe-bas linéaires. Il calcule les commandes de la plate-forme qui réduisent au minimum un coût global. Ce coût correspond au déplacement de la plate-forme et l'écart entre l'accélération du véhicule et l'accélération de la plate-forme au cours d'une session de simulation infinie [SISH82].

Son avantage est d'être un contrôle en boucle fermée. En fonction de la commande optimale, elle minimise la différence d'accélération au niveau de la plate-forme.

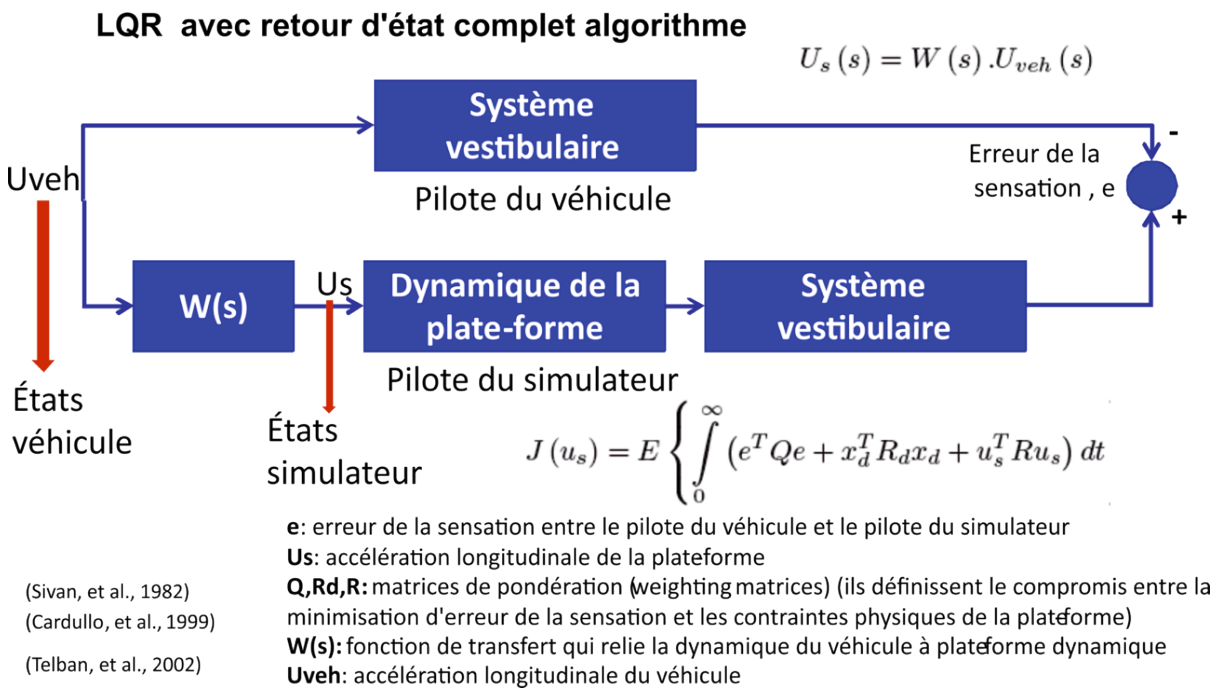


Fig. II-8 : Algorithme optimal.

Le défi est d'établir une fonction de transfert qui relie les accélérations et les rotations du véhicule avec ceux qui doivent être appliquées à la plate-forme, de sorte que l'écart entre les signaux ressentis soit minimisé. Cette technique nécessite la manipulation de plusieurs poids liés à des états qui souvent n'ont pas une représentation physique claire, rendant la tâche de réglage très complexe. En outre, la fonction de transfert n'est optimale que pour un ensemble de scénarios différents d'entrée recréés par un bruit blanc. En d'autres termes, le réglage final

est obtenu grâce à un compromis entre plusieurs réglages, ainsi le comportement de la plateforme ne sera pas systématiquement le meilleur dans toutes les situations [AL09].

2.5.4. Algorithme prédictif (MPC Model Predictive Control)

Les algorithmes prédictifs se rapprochent de la famille des algorithmes optimaux. Ils sont basés sur le contrôle d'un modèle prédictif. L'algorithme de prédiction est l'arbitrage dans le domaine temporel en calculant à chaque étape la commande de plate-forme qui minimise l'écart entre l'accélération du véhicule et l'accélération de simulateur soumise à des contraintes physiques et explicites [DRK+04].

La caractéristique principale du MPC, qui la distingue clairement des autres techniques de contrôle, est sa capacité à gérer les contraintes d'entrée et de sortie, par incorporation directe de ces contraintes à l'algorithme d'optimisation. Cet aspect du MPC est d'une grande importance parce que les opérations les plus efficaces sont généralement obtenues lorsque le processus fonctionne avec une prise en compte de ces contraintes [MH97, AL09]. C'est une méthode très polyvalente qui est en mesure de contrôler un grand nombre de scénarios différents [LAR+09, Tya04].

En outre, il a aussi des avantages quant à la manière dont l'objectif de contrôle est défini. Tous les objectifs de contrôle sont établis en fonction de coûts uniques où les différents paramètres sont pondérés en fonction des préférences de l'utilisateur pour le comportement du système qui en résulte. MPC peut aussi mettre en œuvre la commande anticipatrice afin de compenser les effets des perturbations [LAR+09].

D'autre part, MPC a aussi quelques inconvénients. Cet algorithme requiert une grande puissance de calcul en raison du nombre d'itérations. Les propriétés de l'algorithme sont difficiles à estimer et sont actuellement un domaine de recherche actif. La stabilité, par exemple, est compliquée à déterminer en raison de l'introduction de contraintes au processus d'optimisation, ce qui rend le système non linéaire en boucle fermée [AL09, Tya04].

Chapitre III

Mal du simulateur

Bien que le mal du simulateur et le mal des transports aient des symptômes similaires, ils ne sont pas identiques. Tyler et Bard, 1949 ont indiqué que «le mal des transports est un trouble spécifique qui est évoqué chez les personnes sensibles et les animaux quand ils sont soumis à des mouvements qui ont certaines caractéristiques »[TB49]. Le mouvement propre des simulateurs peut être la cause du mal du simulateur. Cependant, il est possible de ressentir aussi ce mal même dans les simulateurs statiques car il peut être provoqué par une stimulation visuelle du mouvement propre du corps. Aussi, bien que les symptômes de la maladie du mouvement et du mal du simulateur peuvent être différents, leurs causes peuvent être similaires. En règle générale, la seule stimulation vestibulaire est suffisante pour provoquer le mal des transports [Mon69]. Il n'y a donc pas qu'une seule cause exacte de la maladie du simulateur. La maladie du simulateur est plus probablement due à la composition du repérage visuel et du mouvement et pas uniquement en raison du mouvement seul.

Les premières études par Wendt et ses associés au cours des années 1940 cité dans [KBL+87] ont lié exposition et très faible fréquence de vibration (VLF) au mal des transports. Dans la même veine [KBL+87] présentent des travaux concernant les symptômes post-simulateur à des vibrations VLF dans les simulateurs de vol. Les symptômes de la maladie ont été évalués grâce à un questionnaire. Une des conclusions de cette recherche est que le mal du simulateur peut être dû aux vibrations VLF [KFBL92]. Néanmoins, la vision joue un rôle important dans le mal du simulateur [KBA+88] - notamment en raison de son influence sur l'orientation et la perception de soi en mouvement. En fait, il a été soutenu que le mal du simulateur est d'abord induit visuellement [KFBL92].

3.1. Les différentes théories du mal du simulateur

Les théories existantes sur le mal du simulateur peuvent être divisées en quatre catégories :

- Théorie du conflit sensoriel (cue conflict theory) :

Comme il a été souligné, un modèle global du mal du simulateur n'existe pas actuellement. La théorie comprend essentiellement une divergence entre le sens concernant l'information sur l'orientation du corps et du mouvement [KBA+88]. Cette théorie est généralement désignée sous le nom de la théorie du conflit de repère. Les premières études à la fois sur bases fixes et mobiles montrent les conflits de repère comme la source des problèmes [Cas85]. La théorie des conflits de repère est la théorie la plus largement acceptée pour le mal du simulateur. Le conflit se produit lorsqu'il y a une disparité entre les sens. Des sources potentielles de conflit de repère qui pourraient se produire dans un environnement virtuel ont été présentées [MS92]. Ces indices sont nécessaires pour appuyer le contrôle postural et la locomotion, avec des indices vestibulaires et la vision périphérique utilisés en particulier pour l'orientation spatiale et la détection du mouvement propre du corps. Ainsi, avec des informations ambiguës, un conflit de repère peut se développer [MS92]. Les systèmes de perception et de motricité sont modifiables, les gens peuvent apprendre à fonctionner de manière adéquate en dépit de mauvaises conditions de telle sorte que des réarrangements visuel et auditif sont possibles [Wel78]. Il a été souligné que l'adaptation à un monde transformé ne se produit pas immédiatement [MS92, Tre77]. En outre, le temps d'adaptation dépend du type de transformation [WDW86].

- Théorie des restructurations (Rearrangement theory) :

Selon la théorie standard du réarrangement sensoriel, “toutes les situations qui provoquent le mal des transports sont caractérisées par un état de transposition sensorielle dans laquelle le mouvement des signaux transmis par les yeux, le système vestibulaire et les propriocepteurs non-vestibulaires sont en contradiction soit avec un autre ou avec ce qui est attendu de l'expérience précédente” [Rea70, RB75, Rea78, Gri96, Cra90].

- Théorie de l'instabilité posturale (Postural Instability Theory) :

Bien que le conflit de repère soit la théorie la plus largement acceptée pour le mal de simulateur, d'autres théories existent. Des chercheurs ont soulevé plusieurs

autres problèmes [Fra86, SR91] sur le caractère prédictif du mal du simulateur. Il a été présenté une critique de la théorie des conflits de repère à partir d'un point de vue de la psychologie écologique, conduisant à une théorie écologique de la cinétose [SR91]. La théorie écologique qu'ils proposent repose sur la stabilité posturale ou l'absence de celle-ci. Ainsi, une instabilité posturale, ou ataxie, est un effet possible de l'exposition à un simulateur. Riccio et Stoffregen, 1991 formulent l'hypothèse que les symptômes apparaissent lorsque la personne n'a pas ou n'a pas encore appris les stratégies pour le maintien de la stabilité posturale. Pour appuyer leur théorie, ils ont décrit comment plusieurs environnements inducteurs impliquent une instabilité posturale et ils discutent également des influences sur la stabilité. [SR91, RMS92, RS88, RS91].

- Théorie du réflex (Reflex theory) :

Le réflexe opto-cinématique (ROC) est l'ensemble des mouvements oculaires pour identifier une cible dans la scène visuelle, afin de maintenir la cible visée dans l'axe de la fovéa. Le ROC permet d'évaluer les informations de la rétine entière afin de déterminer si un glissement se produit sur l'image. S'il y a un glissement d'image, alors un mouvement correspondant est effectué pour stabiliser l'image. A titre d'exemple de ce processus, quand on regarde par la fenêtre d'un véhicule, l'œil compense le déplacement du véhicule. Il y a une autre réponse similaire causée par les changements d'orientation de la tête détectés par le système vestibulaire appelé réflexe vestibulo-oculaire (RVO). Le ROC et le RVO sont des réponses involontaires qui agissent en synergie pour produire une image rétinienne stable sous une variété de visualisations dynamiques et les conditions de mouvements [ZY81]. Le RVO est un réflexe très rapide qui sert à compenser les mouvements de la tête dans la gamme 1-7 Hz. Toutefois, le RVO est beaucoup moins précis à des fréquences inférieures. Le ROC a les caractéristiques de performances complémentaires. Il a une longue latence due à l'évaluation de l'information visuelle nécessaire pour déterminer une réponse à basses fréquences ($< 0,1$ Hz). Entre les fréquences 0,1 et 1 Hz, le ROC commence à perdre du gain (un gain égal à 1 correspond à une simulation 100% réaliste, un gain inférieur à 1 indique qu'un conflit sensoriel apparaît) et se produit également une phase de latence. Les deux réflexes travaillant à l'unisson sont en mesure de fournir des images rétiniennes stables grâce à un large éventail de fréquences. Un autre lien potentiel entre les

RVO et ROC est liée à l'adaptation du RVO. Il a été démontré que la réponse RVO est adaptable par un ajustement des valeurs de gain pour tenir compte des différentes modalités sensorielles [VDG+98, WF81, VDS96, SRVB11b, SRVB11a, SRKB01, SS07, PDF+97, PF98]. Dans le cas de la recherche à travers une optique agrandie telle que des lunettes de plongée, le RVO adaptera son gain pour correspondre à la quantité de mouvement oculaire requis afin de stabiliser l'image même dans les conditions modifiées. Dans une étude visant à évaluer les effets du facteur d'échelle visuelle sur le RVO, Draper, 1998 a constaté que des grossissements visuels de 2x et 0,5 x ont résulté en des adaptations RVO correspondants et que l'adaptation visuelle est en corrélation avec le mal du simulateur.

3.2. Organes impliqués

3.2.1. Système Vestibulaire (oreille interne)

Pour la perception de son mouvement propre (proprioception) et le contrôle de la marche, l'homme utilise ses systèmes musculaire, neuromusculaire et surtout vestibulaire.

Le système vestibulaire (cf. Fig. III-1), à proximité de l'appareil auditif, est composé des otolithes (utricule, saccule) qui sont responsables de la perception des mouvements linéaires et des canaux semi-circulaires qui sont responsables de la perception des mouvements de rotation.

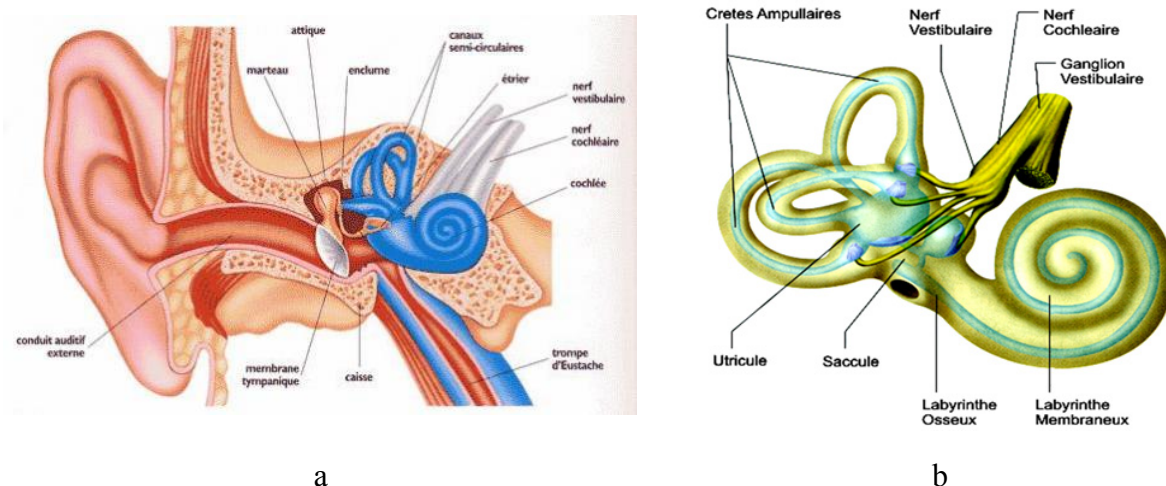


Fig. III-1 : Organes vestibulaires
a) oreille interne – b) système vestibulaire

Les cellules réceptrices dans les otolithes permettent, par inclinaison, de détecter les accélérations linéaires. Ils peuvent aussi ressentir l'inclinaison de la tête. Ce principe est utilisé par les plates-formes dynamiques pour restituer les accélérations latérales et

vestibulaires, ressenties par le conducteur dans les virages, lors des phases d'accélération ou de freinage.

La composante basse fréquence du mouvement est produite par l'inclinaison de la plateforme (inclinaison de coordination). L'inclinaison de coordination est destinée à incliner le conducteur, avec une rotation au-dessous du seuil de détection du canal semi-circulaire vestibulaire (afin d'empêcher le conducteur de détecter la rotation), de telle sorte que la direction de la gravité par rapport à la tête soit orientée de la même façon que l'accélération gravito-inertielle (GIA) (cf. Fig. III-2). Ce mouvement est synchronisé avec la simulation visuelle de l'accélération linéaire [SRVB11a].

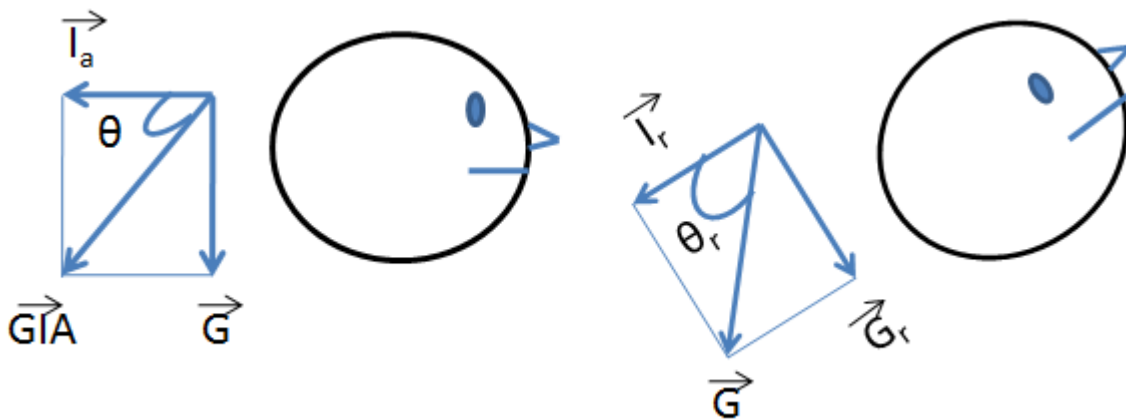


Fig. III-2 : Tilt coordination [SRVB11a]

Les situations sont, d'un point de vue perceptif, équivalentes. Holly et McCollum décrivent la base de la technique du tilt coordination [HM98]. L'accélération, I_a , ainsi que la gravité confère au vecteur résultant GIA (partie gauche de la figure) la même perception que si le sujet avait la tête inclinée (partie droite de la figure). Le système vestibulaire ne peut pas distinguer ces deux situations. Par conséquent, une accélération linéaire peut être simulée par l'inclinaison de la plateforme mais en respectant une vitesse limite [SRVB11a].

Pendant, pour rendre le mouvement transitoire horizontal, des actionneurs linéaires sur rails doivent être utilisés. De plus, avec ces plates-formes opérant dans le domaine des basses fréquences, des plates-formes supplémentaires ou des sièges vibrants peuvent être utilisés, pour rendre le contact du véhicule avec la route (fréquences 0-30 Hz) [Kem99].

3.2.2. Système visuel

L'œil humain est un organe de perception, en fournissant des informations portées par la lumière sur l'environnement. Le système de repérage visuel d'un simulateur stimule

partiellement la perception visuelle humaine par l'affichage d'images sur un écran. Ces images sont calculées dans une pyramide de visualisation définie par l'ouverture de champ de vision et la distance d'observation, puis sont affichées par un système de projection et vues par l'observateur [Kem99].

La génération d'images est effectuée à des fréquences et des résolutions compatibles avec la vision humaine. La fréquence d'image choisie doit garantir une continuité dans la perception à la fois dans les domaines frontaux et périphériques, compte tenu de la persistance de la vision.

Il doit être réglé à 60 Hz pour un confort visuel acceptable, 120 Hz pour la vision stéréoscopique [Kem99]. La Fig. III-3 illustre le champ visuel humain.

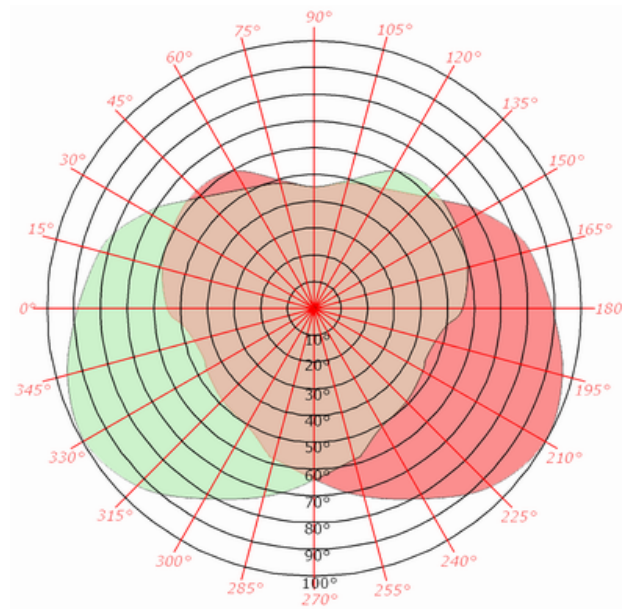


Fig. III-3 : Champ visuel

3.2.3. Système somato-sensoriel

Le système somato-sensoriel permet de percevoir différentes informations, sensations (pression, chaleur, douleur, ...) provenant du corps (peau, articulations, viscère, ...).

Le système somato-sensoriel permet d'être conscient de ce que nous faisons et de l'impact de notre environnement sur nous. Ce système a une place privilégiée dans la compréhension du mal du simulateur. Ces sensations sont élaborées à partir des informations fournies par de nombreux récepteurs sensitifs du système somatosensoriel, situés dans les tissus de l'organisme (mécanorécepteurs du derme et des viscères, fuseaux neuromusculaires des muscles, fuseaux neurotendineux des tendons, plexus de la racine des poils...).

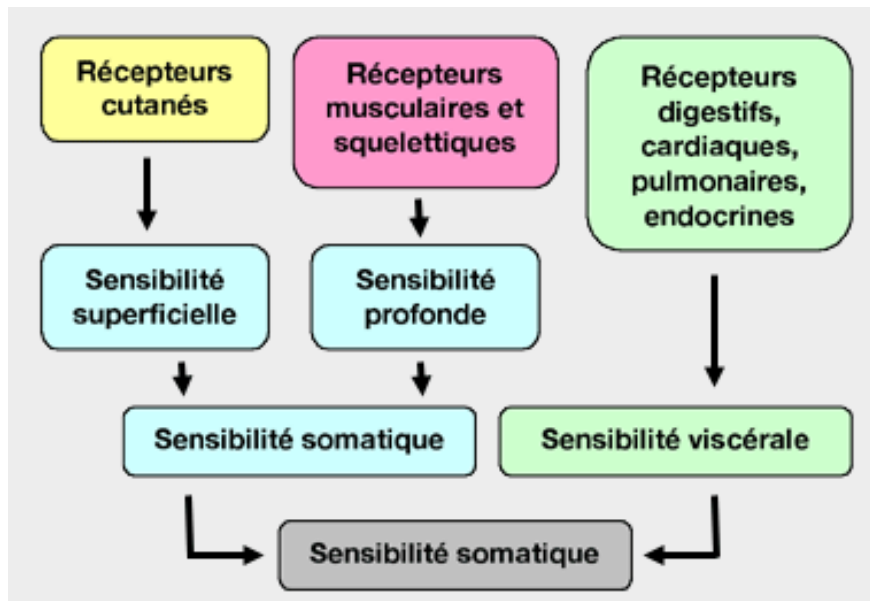


Fig. III-4 : Sensibilité somatique

La figure Fig. III-4 illustre les différents récepteurs de la sensibilité somatique.

3.3. Symptômes

Les principaux symptômes observés lorsqu'une personne est victime du mal du simulateur sont les suivants [Kol95]:

- fatigue oculaire
- peur
- vomissement
- fatigue
- nausées
- anxiété
- étourdissements
- désorientation
- vision floue
- manque de coordination
- maux de tête
- manque d'appétit
- pression mentale
- sécheresse de la bouche
- sueurs froides
- pâleur

3.4. Facteurs

Les facteurs qui provoquent le mal du simulateur peuvent être divisés en trois groupes, les facteurs associés aux individus, les facteurs associés aux simulateurs et les facteurs associés à la tâche.

Tableau III-1 : Facteurs potentiels associés au mal du simulateur dans l'immersion virtuelle[Kol95]

Individual	Simulator	Task
age (Reason & Brand 1975)	binocular viewing (Levine & Shefner 1991, Arditi 1986, Singer et al 1995)	altitude above terrain (Kennedy et al 1993, McCauley & Sharkey 1992)
concentration level (Regan 1993)	calibration (McCauley & Sharkey 1992)	degree of control (Casali 1986, Casali & Wierwille 1986, Pausch et al 1992)
ethnicity (Stern et al 1993)	color (Levine & Shefner 1991, Pausch et al 1992)	duration (Fowlkes et al 1987, McCauley & Sharkey 1992, Kennedy et al 1993)
experience with real-world task (Havron & Butler 1957, Kennedy et al 1987, Kennedy et al 1988, Pausch et al 1992)	contrast (Pausch et al 1992)	global vision flow (Owen 1990, McCauley & Sharkey 1992, Kennedy et al 1993)
experience with simulator (adaptation) (Uliano et al 1986, Kennedy & Frank 1983, Regan 1993)	field of vision (Pausch et al 1992, Kennedy et al 1988,1989; Anderson & Braunstein 1985, Maxwell 1992, Boff & Lincoln 1988)	head movements (Reason & Brand 1975, Dichgans & Brandt 1973, Guedry & Montague 1961, Regan 1993,)
flicker fusion frequency threshold (Grandjean 1988, Botwinick & Brinley 1963, Maxwell 1992, Wilson 1963)	flicker (Harwood & Foley 1987, Boff & Lincoln 1988, Farrell et al 1988, Rinalducci & MacArthur 1990, Pausch et al 1992)	luminance level (Farrell et al 1988, Pausch et al 1992)
gender (Biocca 1992, Kennedy&Frank 1983)	inter-pupillary distance (Regan & Price 1993)	unusual maneuvers (Frank & Casali 1986, McCauley & Sharkey 1992)
illness and personal characteristics (Kennedy et al 1987, Kennedy et al 1993, Biocca 1992)	motion platform (Casali 1986, Kennedy et al 1987, Kennedy et al 1993, McCauley & Sharkey 1992)	methode of movement (Regan 1993)
mental rotation ability (Parker & Harm 1992)	phosphor lag (Pausch et al 1992)	rate of linear or rotational acceleration (McCauley & Sharkey 1992)
perceptual style (Witkin et al 1977, Barrett & Thornton 1968, Barrett et al 1969,1970 ; Long et al 1975, Frank&Casali 1986, Ebenholtz 1977,	position tracking error (Biocca 1992, Hettinger et al 1990)	self-movement speed (McCauley & Sharkey 1992)
postural stability (Kolasinski et al 1994, Jones et al 1993)	refresh rate (Thorell & Smith 1990, Pausch et al 1992, Farrell et al 1988, Maxwell 1992)	sitting vs. standing (Reason & Brand 1975, Riccio and Stoffregen 1991, Regan 1993)
	scene content (Pausch et al 1992)	vection (Kennedy et al 1988,1989; Hettinger et al 1990, Kennedy et al 1993, Anderson & Braunstein 1985)
	viewing region (Casali & Wierwille 1986, Pausch et al 1992)	type of application (McCauley & Sharkey 1992)
	update rate (frame rate) (Pausch et al 1992)	

Les organes sensoriels principalement concernés sont sollicités diversement en fonction de la situation de conduite. Le Tableau III-2 dresse la liste des organes sensoriels concernés par la perception du mouvement pour différentes situation de conduite [MAT09].

Tableau III-2 : Organes sensoriels concernés par la perception du mouvement [MAT09a, MAT09b]

	Organes sensoriels				
	Système visuel	Vestibulaire	Epicritique	Haptique	Oreille
Déviations de la voie	X				
Vitesse latérale / Vitesse de lacet	X				
Vitesse de conduite	X				X
Latérale / Longitudinale accélération		X	X	X	
Accélération de lacet		X			
Inclinaison (tangage, roulis)	X	X			
Volant	X				
Couple au volant			X	X	
Déviations par rapport à la désirée route	X				
Forces			X	X	
Son de moteur/conduite					X

3.5. Critères du mal du mouvement/simulateur

Certaines approches peuvent être utilisées pour définir les types de mal du simulateur/mouvement :

- MSDV (motion sickness dose value)-IR (Illness Rating),
- VDV (vibration dose value),
- stabilité posturale,
- seuil de la perception du mouvement,
- ratio vitesses angulaires entre niveaux vestibulaire et véhicule,
- ratio accélérations entre niveaux vestibulaire et véhicule.

3.5.1. L'approche de 'MSDV (Motion Sickness Dose Value)-IR (Illness Rating)'

Förstberg [1996,2000] estime que les barèmes en vigueur pour le bien-être sont trop approximatifs. Aussi, il a proposé d'étudier les symptômes de la « Motion Sickness Incidence » (SMSI). Le SMSI est le rapport entre les sujets présentant des symptômes et le nombre de sujets. Förstberg a utilisé l'étourdissement, les nausées et la négation de : « je me sens bien » à partir de l'échelle de bien-être.

La définition d'un sujet présentant un symptôme du mal des transports (SMS : symptome motion sickness) commence par le fait de se sentir bien au début de l'essai puis de ressentir l'un des symptômes suivants: étourdissements, nausées ou « je ne me sens pas bien ».

La notation utilisée, définie comme le taux de mal du simulateur (Illness Rating - IR) est classée en plusieurs catégories : je me sens bien, je ne me sens pas très bien, je me sens un peu malade, je me sens mal, je me sens très mal.

En d'autres termes, SMS = 0 si des vertiges + nausées + IR = 0 sinon SMS = 1, Si IR (au début) = 0. Si IR (au début) >= 1, alors le sujet ressent un malaise significatif. Le SMSI est un pourcentage. Il est évalué sur tous les essais.

$$SMSI_x = \frac{\sum_{parts,sujets} SMS \cdot \delta(s)}{\sum_{parts,sujets} \delta(s)} \quad (1)$$

Où $\delta(s) = 1$; si le sujet a pris part à la condition d'inclinaison X, sinon $\delta(s) = 0$

Dans l'évaluation objective de l'environnement en mouvement, le concept de dose de mouvement (de l'ISO 2631-1997) [Gri90, LG87] est généralement utilisé. Il est également possible d'adopter cette notation pour l'évaluation du mal de simulateur [Per08]. La fonction de pondération (w_F) (ISO 2631-1,1997) pour calculer la dose de mouvement a été légèrement modifiée et remplacée par un filtre passe-bande (BP) de 0.08 à 0.35 Hz filtre. La raison de cette modification est que le filtrage numérique dans le domaine temporel a été plus facile avec ce filtre que l'original w_F -filtre. La différence est faible et ils sont dans l'erreur de la tolérance marginale du filtre (ISO 2631-1,1997) entre 0.5 et 0.7 Hz. Ce filtre a été utilisé pour tous les calculs de la dose de mouvement vertical, accélérations latérales et déploiement à partir de signaux temporels.

La valeur de dose du mal du simulateur (MSDV) est un outil pour l'objectivation d'évaluation subjective qui a été définie dans le cadre de la norme ISO 2631-1 1997, [Gri96, Gri04, GN04a, GN04b, ISO97].

$$MSDV_Z = \left[\int_0^T a_{w_f}^2(t) dt \right]^{0.5} \left[m/s^{1.5} \right] \quad (2)$$

Pourcentage de personnes qui peuvent vomir = $K_m \cdot MSDV_Z$ [%] avec $K_m = 1 / 3$ de la population mixte d'adultes.

Notation de Maladie (Illness Rating) (IR)

0 = Je me sentais bien,

1 = J'ai ressenti un léger malaise,

2 = Je me sentais très mal.

≥ 3 = Je me sentais absolument mal.

$$IR = 1/50 * MSDV_z \quad (3)$$

3.5.2. L'approche de 'VDV (Vibration Dose Value)'

L'utilité de RMQ (root mean quadrat) est la moyenne des accélérations contenant des facteurs de crête élevés combinés à la forme soutenable de la dépendance temporelle $a^4(t)$ sur de longues durées. Cela signifie qu'une procédure de calcul de la moyenne simple et unique pourrait être applicable sur toute la gamme de conditions de circulation (défauts de la route, impacts répétés à vibrations intermittentes et des vibrations continues de longue durée). Avec la même relation entre l'amplitude et la durée de toutes les périodes, il devient très simple de définir un moyen d'accumuler l'exposition à des périodes de différentes grandeurs [Gri96]. Ce résultat est obtenu en faisant la somme mathématique de la quatrième puissance de l'accélération pondérée en fréquence, $a(t)$, au cours de la période au cours de laquelle le mouvement peut se produire :

$$dose\ value = \int_{t=0}^{t=T} a^4(t) dt \ m^4\ s^{-7} \quad (4)$$

Tel que défini ci-dessus, les valeurs de dose peuvent devenir très grandes et il n'y a pas de relation linéaire entre la grandeur physique du stimulus et la réponse subjective. Les deux problèmes sont résolus par la détermination de la racine quatrième de la valeur de la dose. Cela a été nommé la valeur de dose de vibrations (VDV).

$$VDV = \left[\int_{t=0}^{t=T} a^4(t) dt \right]^{1/4} \quad (5)$$

Tel que défini ici, les unités de la VDV sont $ms^{-1.75}$ (ou $ms^{7/4}$). Une unité sans dimension ayant les mêmes caractéristiques pourrait être définie: cela peut consoler l'utilisateur non-mathématicien, mais serait une perte pour le physicien et l'ingénieur. La dépendance du temps de VDV correspond à une réduction de 1,5 dB en amplitude par un doublement de la durée d'exposition [Gri96]. Le chiffre correspondant pour la moyenne quadratique (et $a^2(t) =$ constante) est de 3 dB. Dans les unités linéaires, une augmentation de 16 fois la durée requiert une réduction de moitié de l'amplitude des vibrations pour maintenir la même dose de vibrations [Gri96].

La relation entre la quatrième puissance d'accélération et le temps peut être approximée par la réduction de la charge admissible RMS (root mean square) magnitude de 1,5 dB par doublement de la durée d'exposition et d'éviter ainsi l'utilisation de l'équation ci-dessus. Les valeurs ne sont utiles que si la vibration est « bien élevée ». En général, les valeurs efficaces ne peuvent pas être assimilées à des chocs, à des vibrations intermittentes ou à d'autres conditions variables dans le temps. Il a été montré que lorsque le facteur de crête est inférieur à environ six, la valeur estimée de vibration dose (eVDV) qui est calculée à partir de la valeur efficace de l'accélération pondérée en fréquence, fournit une bonne approximation de la valeur de dose de vibrations vrai [Gri96].

$$eVDV = [(1.4 \times \text{valeur r.m.s, } m s^{-2})^4 \times (\text{duration, } s)]^{1/4} m s^{-1.75} \quad (6)$$

ou

$$eVDV = 1.4 a_t t^{1/4} \quad (7)$$

où a_t est l'accélération r.m.s. qui est la fréquence pondérée une période déterminée de t secondes. Il s'agit d'une relation déterminée empiriquement [Gri96]. Le eVDV tend à être plus élevé que le VDV pour un facteur de crête très faibles (≤ 3) et plus bas que le VDV pour les facteurs de crête élevés (≥ 6). Le facteur de crête ne donne pas une très bonne indication des différences entre les mesures eVDV et VDV: avec quelques mouvements, il est possible d'obtenir une estimation utile de la valeur de dose de vibrations de RMS mesure où le facteur de crête est aussi élevé que 12 selon la littérature [Gri96].

3.5.3. L'approche de la stabilité posturale

La méthode de la stabilité posturale peut être utilisée pour évaluer le niveau de mal de simulateur des participants qui ont participé aux expériences menées sur simulateur de conduite.

L'analyse posturale est l'un des moyens puissants et efficaces pour identifier les déséquilibres musculo-squelettal qui conduisent à des problèmes de blessures et de performance. Cette analyse impose la prescription d'exercice. Chaque entraîneur personnel, un physiothérapeute, un ergothérapeute, et l'entraîneur l'utilise pour évaluer avec précision les déséquilibres posturaux et prescrire un programme d'exercices.

Dans nos expérimentations, nous avons utilisé l'analyse posturale comme un indicateur du mal de simulateur. Les mesures ont été réalisées pendant 30 secondes et ils ont été échantillonnés à 40 Hz.

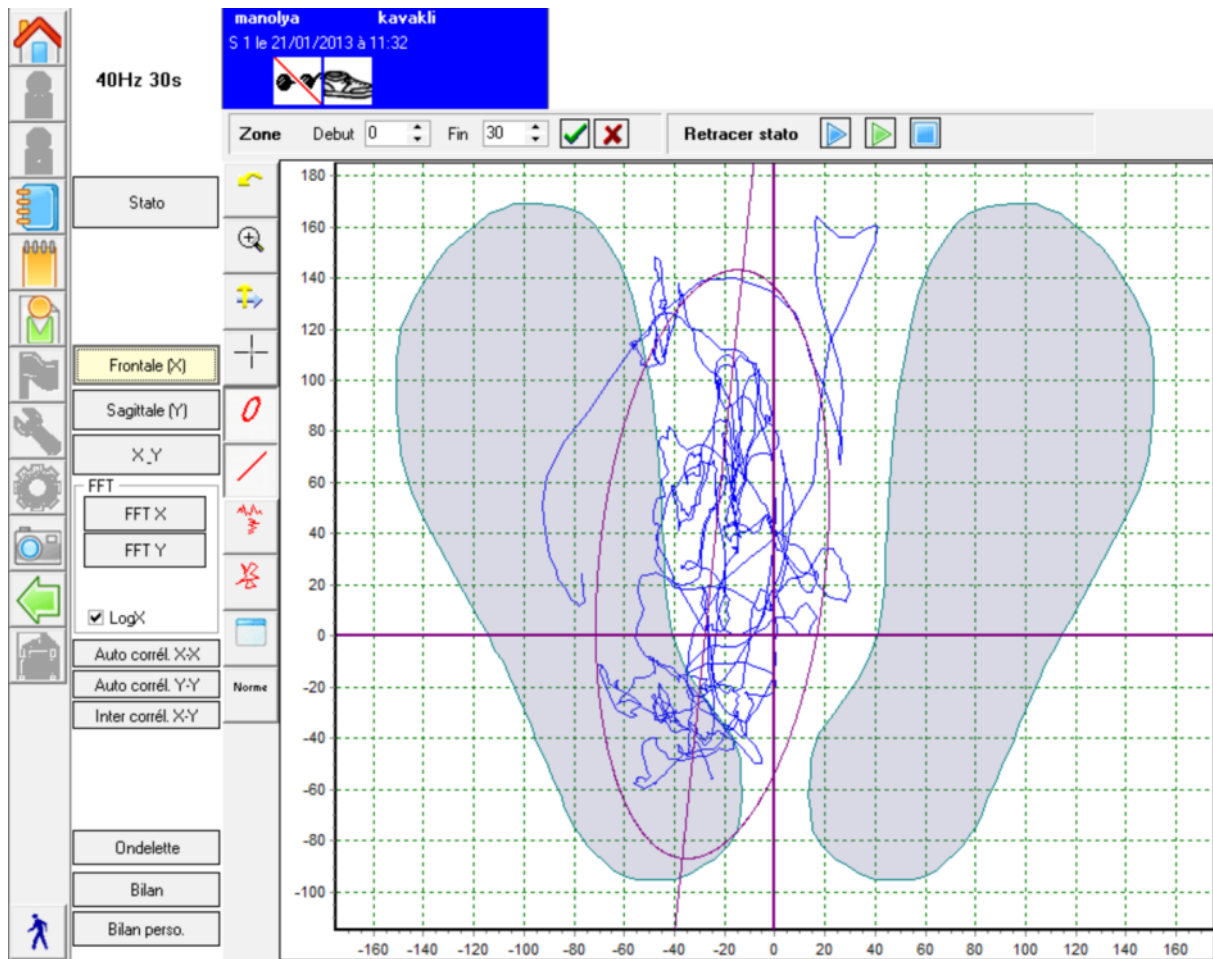


Fig. III-5 :GUI pour le logiciel PostureWin

3.5.4. L'approche du seuil de la perception du mouvement

Les vitesses et accélérations ($\dot{\varphi}, \dot{\theta}, \dot{\psi}, \ddot{\varphi}, \ddot{\theta}, \ddot{\psi}$) de roulis, tangage et lacet peuvent être utilisés pour identifier le seuil de perception du mouvement des sujets.

Le mouvement de roulis (φ) est la rotation autour de l'axe longitudinal (X_V), le tangage (θ) est la rotation autour de l'axe latéral (Y_V) et le lacet (ψ) est la rotation autour de l'axe vertical (Z_V).

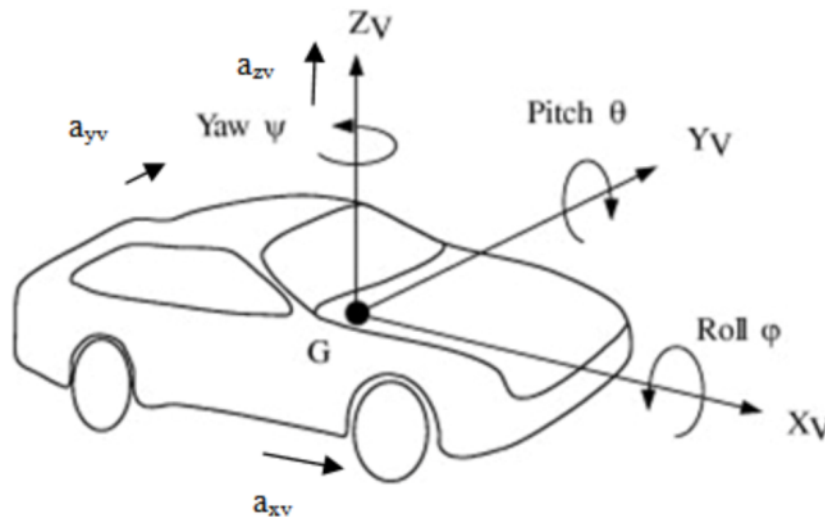


Fig. III-6 : Schéma des accélérations d'un véhicule

L'organe vestibulaire peut détecter seulement les mouvements qui se situent entre les seuils de perception. Beaucoup d'études expérimentales indiquent que le seuil de détection humaine des mouvements de rotation est compris entre $0.1 \text{ }^\circ / \text{s}$ et $3 \text{ }^\circ / \text{s}$. Pour les mouvements linéaires, le seuil de perception est entre 0.014 et 0.25 m/s^2 [AQT, GM61, Gue64, Gue74, Gun78, BSS86, Ben88, BHB89, Ben90, Kin05]. Le seuil de rotation autour de l'axe de lacet a été déterminé pour les stimuli d'accélération constante $0.22 \text{ }^\circ/\text{s}^2$ [RSM81]. Les seuils d'accélération médiane pour la perception de la direction du mouvement linéaire pour le mouvement antéropostérieur est de 8.5 cm/s^2 et le mouvement latéral 6.5 cm/s^2 . Selon la littérature, les seuils d'accélération sont dépendants du type de stimuli alors que les seuils de vitesse ne le sont pas. Les seuils de vitesse médiane pour la perception de la direction du mouvement linéaire pour le mouvement antéropostérieur est de 13.5 cm/s et le mouvement latéral est de 10.4 cm/s [Kin05]. Ces valeurs seuils de perception dépendent de l'accumulation des stimuli de mouvement ainsi que les taux des accélérations. Les seuils de perception de la rotation et les mouvements linéaires peuvent varier d'un sujet à l'autre [AQT, GM61, Gue64, Gue74, Gun78, BSS86, Ben88, BHB89, Ben90, Kin05].

La figure ci-dessous illustre la mesure de la dynamique de la tête des sujets lors des expériences en temps réel de simulation de conduite.



Fig. III-7 : Mesure au niveau vestibulaire

Afin de sauvegarder les données d'accélération par rapport au niveau vestibulaire, un capteur de suivi de mouvement a été utilisé (cf. Fig. III-7). Ce suivi de mouvement peut mesurer les données telles que les angles de roulis, tangage, lacet, et les vitesses ainsi que les accélérations dans les directions X, Y et Z. Le taux d'échantillonnage pour l'enregistrement des données pendant les mesures des capteurs est de 20 Hz. Une calibration est effectuée avant chaque campagne de mesures [XSens, AMP+13] ce qui permet d'aligner l'ensemble des axes de mesure.

Tableau III-3 : Principales méthodes de mesure du mal du simulateur.

Méthode de mesure du mal du simulateur	Références
Questionnaire mal du simulateur (Simulator Sickness Questionnaire (SSQ))	(Kennedy et al.,1993)
Questionnaire mal du mouvement (Motion Sickness Questionnaire (MSQ))	(Kennedy et al., 1992)
Tests posturaux (Postural tests)	(Baltzley et al, 1989 ; Hamilton et al, 1989)
Historique sur les questionnaires du mal du mouvement	(Motion Sickness History Questionnaire (MHQ))
Méthodes biofeedback (Physiologique mesure)	- potentiel de la peau (Skin potential (SP)), la température de la peau (Skin temperature (ST)), résistance de la peau (Skin resistance (SR)) [Bertin et al., 2004a ; Bertin et al.,2004b ;Bertin et al.,2005] - La fréquence cardiaque (Heart rate), fréquence du pouls (pulse rate) - Electrogastragram (EGG) - Electrocardiogram (ECG) [Wu et al.,2009]
Valeur Dose du mal du mouvement (Motion Sickness Dose Value)	cf. Chapitre III

3.6. Mesures objectives

Pour les mesures objectives, les méthodes de biofeedback (physiologique tels que l'EMG, ECG, etc) et de mesures de mouvements de la tête (dynamique vestibulaires) sont utilisées [KMKL10, Kem90, Kem99, Kem04, Kem09, ATM09, APM+11, APMK12a, APMK12b, BSPB+07, Ber09, BSPB10].

3.6.1. Mesure dynamiques neuromusculaires (EMG)

L'électromyographie (EMG) est une technique d'évaluation et d'enregistrement de l'activité électrique produite par les muscles du squelette [emg13]. Un électromyogramme détecte le potentiel électrique généré par les cellules musculaires [emg13] lorsque ces cellules sont activées. Les signaux peuvent être analysés pour détecter des anomalies médicales, le niveau d'activation, ou pour analyser la biomécanique du mouvement humain.



Fig. III-8 : Analyse d'EMG

L'EMG (électromyogramme) a été mise en œuvre pour une partie particulière des essais afin d'expliquer la dynamique neuromusculaire des conducteurs lors des séances sur simulateur.

Il existe deux types d'EMG : **1** – invasives **2**– non- invasive.

Dans les expérimentations réalisées à l'Institut Image, la technique EMG non invasive a été utilisée. En outre, aucun stimulus électrique n'a été appliqué.

La Fig. III-9 illustre la position des électrodes sur le bras du sujet.



Fig. III-9 : Branchement des électrodes pour l'analyse des EMG

La Fig. III-10 explique en détail les muscles sollicités pendant les essais. Pour l'intérieur du bras : flexor carpi radialis , pour l'extérieur : brachioradialis ; '+'[Bri13]

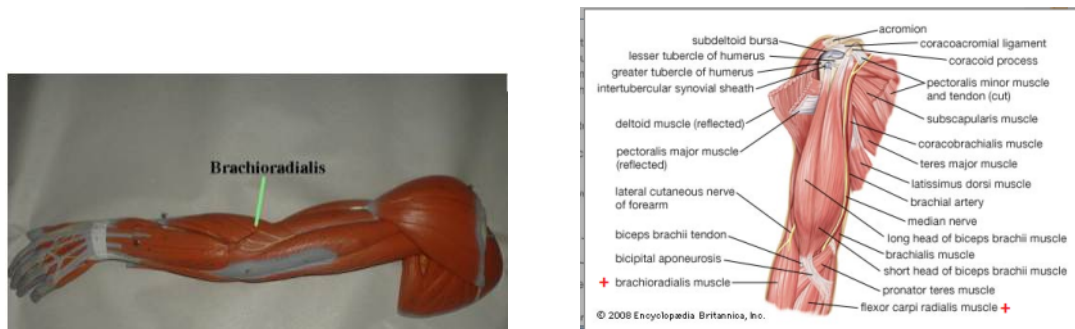


Fig. III-10 : Muscles des bras choisis pour l'acquisition

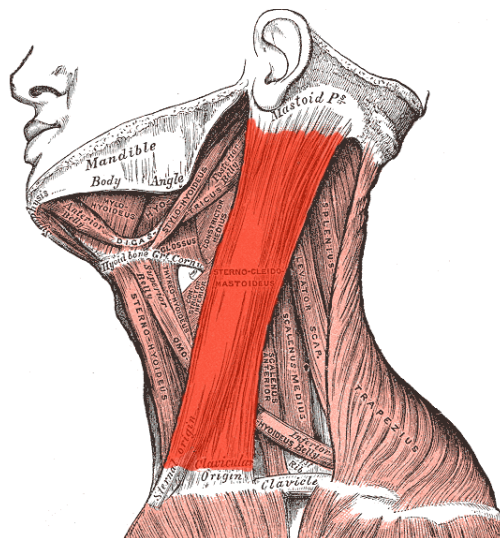


Fig. III-11 : Les muscles des sternocleidomastoidis [Gil07]

La mesure de l'activation des muscles permet d'observer la différence lors de mêmes conditions mais en changeant :

- L'algorithme de contrôle de la plateforme (classique, optimale, adaptatif, hybride)
- La rugosité de la route (effet de l'amplitude, la fréquence et la vitesse du véhicule)
- L'effet de la plate-forme de mouvement (statique, dynamique)
- Différents scénarios (Country road, l'OTAN Chicane manœuvre, sinus steer, bosses, courbes, etc.)
- Différents temps d'exposition des vibrations (2 minutes, 15 minutes)

Chapitre IV

Contrôle de la plateforme

4.1. Algorithmes mis en œuvre

Au cours de cette thèse, nous avons développé huit algorithmes: un algorithme pour le simulateur en fonctionnement statique et 7 en mode dynamique :

- Statique,
- Classique,
- Adaptatif 1 et 2,
- Optimale 1 et 2,
- Hybride 1 et 2.

Ces différents algorithmes sont détaillés dans la suite de ce chapitre.

L'algorithme en mode statique n'a pas d'entrée et de sortie au niveau de la plateforme dynamique car seul le modèle dynamique du véhicule est utilisé.

Tableau IV-1 : Facteur des déplacements en tridimensionnel selon les algorithmes

	Facteur de déplacement longitudinal (X)	Facteur de déplacement latéral (Y)	Facteur de déplacement vertical (Z)
<i>Statique</i>			
<i>Classique (dynamique)</i>	0.2	0.2	0.22
<i>Adaptatif1 (dynamique)</i>	0.2	0.2	0.22
<i>Adaptatif2 (dynamique)</i>	1	1	1
<i>Optimale1 (dynamique)</i>	0.2	0.2	0.22
<i>Optimale2 (dynamique)</i>	1	1	1
<i>Hybride1 (dynamique)</i>	0.2	0.2	0.22
<i>Hybride2 (dynamique)</i>	1	1	1

Le Tableau IV-1 illustre les différences que nous avons imputées à chacun des algorithmes.

La variation du facteur de déplacement permet d'analyser les réponses des algorithmes.

4.2. Algorithme classique

L'algorithme classique utilisé dans nos expérimentations est celui présenté dans le Chapitre II. La Fig. IV-1 donne les détails des filtres utilisés pour cet algorithme. L'algorithme fonctionne en boucle ouverte. Il servira de référence pour les comparaisons que nous ferons avec les autres types d'algorithmes mis en œuvre.

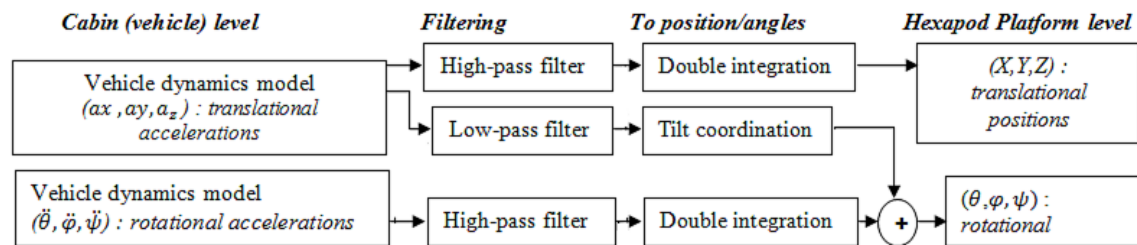


Fig. IV-1 : Détails des filtres algorithme classique

Le Tableau IV-2 donne le détail des valeurs utilisées dans les filtres de l'algorithme classique.

Tableau IV-2 : Paramètres de l'algorithme classique

Symbol	Longitudinal	Lateral	Roll	Pitch	Yaw
2 nd order LP cut-off frequency (Hz)			0.3	0.7	
2 nd order LP damping factor			0.3	0.7	
1 st order LP time constant (s)	0.1	0.1			0.1
2 nd order HP cut-off frequency (Hz)	0.5	0.5			2
2 nd order HP damping factor	1	1			1
1 st order HP time constant (s)	2	2			2

4.3. Algorithme adaptatif (Discrete Time Model Adaptive Control – MRAC)

4.3.1. Définition de l'algorithme adaptatif

L'algorithme classique est représenté sur la Fig. IV-2 par les lignes continues noires. L'algorithme MRAC quant à lui inclut les lignes continues noires et les lignes discontinues rouges.

L'algorithme MRAC de repérage proposé utilise ici les mêmes filtres et gains (Fig. IV-2 et Tableau IV-2) que l'algorithme classique. L'idée principale pour le MRAC, qui est également connu comme MRAS (Model Reference Adaptive System), est de construire un contrôleur en boucle fermée avec des paramètres qui peuvent être mis à jour pour changer la réponse du système. La sortie du système est comparée à une réponse désirée à partir d'un modèle de référence. Les paramètres de commande sont mis à jour sur la base de cette erreur. L'objectif est de faire converger les paramètres vers des valeurs idéales qui provoquent la réaction de la plateforme en fonction de la réponse du modèle de référence [IS96, DP97, TI93, GLT09, MGT11].

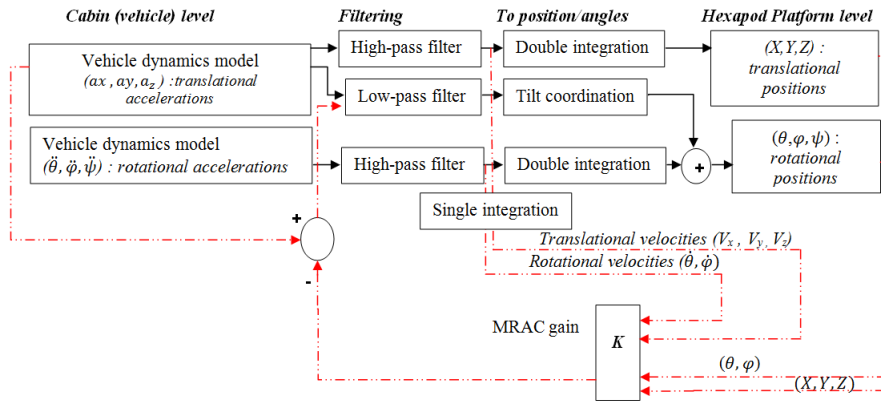


Fig. IV-2 : Algorithmes Classique et le MRAC

4.3.2. Modélisation de l'algorithme adaptatif

L'équation (1) décrit l'état du système [IS96, TI93, GLT09, MGT11], il s'agit d'un système MIMO (multiple inputs – multiple outputs).

$$x(k+1) = Ax(k) + Bu(k), \quad y(k) = Cx(k) \quad (1)$$

avec $A \in R^{n \times n}$, $B \in R^{n \times M}$ et $C \in R^{M \times n}$ étant des matrices de paramètres inconnus et constants et $x(k) \in R^n$, $u(k) \in R^M$ et $y(k) \in R^M$ étant respectivement l'état du système, les signaux d'entrées et de sorties de vecteur.

L'objectif de cet algorithme est de concevoir un état d'asservissement du signal $u(k)$ dans l'équation (1) de telle sorte que tous les signaux en boucle fermée restent limités et que le

système de signal de sortie $y(k)$ suite le modèle de sortie de référence $y_m(k) \in \mathbb{R}^M$ qui est généré par le modèle de référence (cf. équation 2)

$$y_m(k) = W_m(z)[r](k) \quad (2)$$

$W_m(z)$ est une matrice de dimension $M \times M$ de transfert réelle, et $r(k)$ un vecteur de dimension M , $r(k)$ est le signal d'entrée borné.

Hypothèses.

(H1) Dans la fonction de transfert $G_i(z) = C_i(zI - A_i)^{-1}B_i$, $i = 1, 2, \dots, N$ tous les zéros se situent dans le cercle unité dans le plan z .

(H2) $G_i(z)$, $i = 1, 2, \dots, N$, sont de rang plein, il existe une matrice $\xi_m(z)$ pour tout $G_i(z)$, $i = 1, 2, \dots, N$ et la matrice de référence du système $W_m(z) = \xi_m^{-1}(z)$,

(H3) Tous A_i , $i = 1, 2, \dots, M$, de la matrice de gain à haute fréquence K_p sont non nuls et leurs signes sont connus.

La structure du régulateur pour la boucle fermée est la suivante :

$$u(k) = K_1^T(k)x(k) + K_2(k)r(k) \quad (3)$$

où $K_1(k) \in \mathbb{R}^{n \times M}$ et $K_2(k) \in \mathbb{R}^{M \times M}$ sont des matrices de paramètres mises à jour à partir des lois d'adaptation, de sorte que le vecteur d'état du signal $x(k)$ puisse suivre asymptotiquement une référence $x_m(k)$ définie selon l'équation suivante :

$$x_m(k+1) = A_m x_m(k) + B_m r(k) \quad (4)$$

où $A_m \in \mathbb{R}^{n \times n}$ est stable et $B_m \in \mathbb{R}^{n \times M}$. Pour une telle conception de la commande adaptative, les conditions de raccordement $K_1(k)$ et $K_2(k)$ sont les estimations de la valeur nominale et qui satisfont aux conditions d'appariement

$$C_i(zI - A_i - B_i K_1^{*T})^{-1} B_i K_2^* = W_m(z), \quad K_2^{*-1} = K_p \quad (5)$$

où $K_p = \lim_{z \rightarrow \infty} \xi_m(z) G_i(z)$ est le gain haute fréquence de la matrice de $G_i(z)$

L'existence est garantie par l'état du système nominal

(H4) (A, B) est stabilisable et (A, C) est observable.

Suivi équation de l'erreur. En substituant la loi de commande (3) dans (1), nous obtenons

$$\begin{aligned}
 x(k+1) &= (A + BK_1^{*T})x(k) + BK_2^*r(k) \\
 &+ B((K_1^T(k) - K_1^{*T})x(k) + (K_2(k) - K_2^*)r(k)), \\
 y(k) &= Cx(k)
 \end{aligned} \tag{6}$$

Selon le modèle de référence (2) et les équations (5) et (6), la sortie d'erreur de suivi $e(k) = y(k) - y_m(k)$ est

$$e(k) = W_m(k)K_p[\tilde{\theta}^T \omega](k) + Ce^{(A+BK_1^{*T})k}x(0) \tag{7}$$

où $Ce^{(A+BK_1^{*T})k}x(0)$ converge vers zéro de façon exponentielle et

$$\tilde{\theta}(k) = \theta(k) - \theta^* \tag{8}$$

$$\theta(k) = [K_1^T(k), K_2(k)]^T \tag{9}$$

$$\theta^* = [K_1^{*T}, K_2^*]^T \tag{10}$$

$$\omega(k) = [x^T(k), r^T(k)]^T \tag{11}$$

Ensuite, il est nécessaire d'utiliser une décomposition de type LDS pour déterminer K_p [Tao03].

Il est crucial de développer un modèle d'erreur en fonction de certains paramètres connexes

$$e(k) = y(k) - y_m(k)$$

Modèle d'erreur. En négligeant le terme $Ce^{(A+BK_1^{*T})k}x(0)$, on obtient à partir de (7) et (2)

$$\xi_m(z)[e](k) = K_p \tilde{\theta}^T(k) \omega(k) \tag{12}$$

Pour faire face à l'incertitude du gain à haute fréquence matrice K_p , nous utilisons la décomposition LDS

$$K_p = L_s D_s S \tag{13}$$

où $S \in \mathbb{R}^{M \times M}$ avec $S = S^T > 0$, L_s est une matrice $M \times M$ unité triangulaire, et

$$\begin{aligned}
 D_s &= \text{diag}\{s_1^*, s_2^*, \dots, s_M^*\} \\
 &= \text{diag}\left\{ \text{sign}[\Delta_1] \gamma_1, \dots, \text{sign}\left[\frac{\Delta_M}{\Delta_{M-1}}\right] \gamma_M \right\}
 \end{aligned} \tag{14}$$

de telle sorte que $\gamma_i > 0$, $i = 1, \dots, M$, en remplaçant (13) dans (12) on obtient

$$L_s^{-1} \xi_m(z)[e](k) = D_s S \tilde{\theta}^T(k) \omega(k) \quad (15)$$

Pour paramétrer l'inconnu L_s , θ_0^* , est introduit dans (16)

$$\theta_0^* = L_s^{-1} - I = \begin{bmatrix} 0 & 0 & \dots & 0 \\ \theta_{21}^* & 0 & \dots & 0 \\ \theta_{31}^* & \theta_{32}^* & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \theta_{M-1,1}^* & & 0 & 0 \\ \theta_{M,1}^* & & \theta_{M,M-1}^* & 0 \end{bmatrix} \quad (16)$$

Ainsi, après calcul,

$$\xi_m(z)[e](k) + \theta_0^* \xi_m(z)e(k) = D_s S \tilde{\theta}^T(k) \omega(k) \quad (17)$$

Un filtre a été conçu $h(z) = 1/f(z)$, où $f(z)$ est un polynôme de degré stable unitaire égal au degré de $\xi_m(z)$, en multipliant des deux côtés de (17) par $h(z) \cdot I_M$

$$\begin{aligned} \bar{e}(k) + [0, \theta_2^{*T} \eta_2(k), \theta_3^{*T} \eta_3(k), \dots, \theta_M^{*T} \eta_M(k)] \\ = D_s S h(z) [\tilde{\theta}^T \omega](k) \end{aligned} \quad (18)$$

Où

$$\bar{e}(k) = \xi_m(z) h(z) [e](k) = [\bar{e}_1(k), \dots, \bar{e}_M(k)]^T \quad (19)$$

$$\eta_i(k) = [\bar{e}_1(k), \dots, \bar{e}_{i-1}(k)]^T \in R^{i-1}, i = 2, \dots, M \quad (20)$$

$$\theta_i^* = [\theta_{i1}^*, \dots, \theta_{ii-1}^*(k)]^T \in R^{i-1}, i = 2, \dots, M \quad (21)$$

Sur la base de cette équation d'erreur paramétrée, nous atteignons le signal d'estimation de l'erreur

$$\begin{aligned} \hat{e}(k) = [0, \theta_2^T \eta_2(k), \theta_3^T \eta_3(k), \dots, \theta_M^T \eta_M(k)] \\ + \psi(k) \xi(k) + \bar{e}(k) \end{aligned} \quad (22)$$

où $\Theta_i(k)$, $i = 2, 3, \dots, M$ sont les estimations de Θ_i^* , et $\psi(k) - \psi^*$ sont liés aux erreurs de paramètres.

Lois adaptatives. Dans le modèle de l'erreur d'estimation (22), les lois choisies

$$\dot{\theta}_i(k) = \frac{\Gamma_{\theta_i \varepsilon_i(k)} \eta_i(k)}{m^2(k)}, i = 2, 3, \dots, M \quad (23)$$

$$\dot{\theta}^T(k) = -\frac{D_s \varepsilon(k) \xi^T(k)}{m^2(k)} \quad (24)$$

$$\dot{\Psi}(k) = \frac{\Gamma_{\varepsilon(k)} \xi^T(k)}{m^2(k)} \quad (25)$$

$$\omega(k) = [x^T(k), r^T(k)]^T \quad (26)$$

où le signal de $\varepsilon(k) = [\varepsilon_1(k), \varepsilon_2(k), \dots, \varepsilon_M(k)]$ est calculé à partir de (22), $\Gamma_{\theta_i} = \Gamma_{\theta_i}^T > 0$, $i = 2, 3, \dots, M$ et $\Gamma = \Gamma_t > 0$ sont des matrices de gain d'adaptation et

$$m(k) = 1 + \xi^T(k) \xi(k) + \xi^T(k) \xi(k) + \sum_{i=2}^M \eta_i^T(k) \eta_i(k))^{1/2} \quad (27)$$

est un signal de normalisation standard.

Nous considérons le modèle dynamique du simulateur hexapode décrit par (28) qui est une variable à trois états (x). Les entrées de commande sont (u), l'angle de tangage (θ_v), l'angle de roulis (ϕ_v) et l'angle de lacet (ψ_v) du modèle de véhicule. Notre modèle MRAC est défini par l'équation (29) avec un intervalle de temps de 1/60 s pour obtenir l'algorithme temps réel de pilotage de la plateforme.

$$\begin{aligned} x &= [x(k+1), y(k+1), z(k+1)]^T, \\ u &= [\theta_v(k), \phi_v(k), \psi_v(k)]^T \end{aligned} \quad (28)$$

$$\begin{aligned} \begin{bmatrix} x(k+2) \\ y(k+2) \\ z(k+2) \end{bmatrix} &= \begin{bmatrix} \frac{-5.0625 + 2 \cdot x(k+1)_{HP}}{x(k+2)_{HP} + 1} & 0 & 0 \\ 0 & \frac{-5.0625 + 2 \cdot y(k+1)_{HP}}{y(k+2)_{HP} + 1} & 0 \\ 0 & 0 & \frac{-5.0625 + 2 \cdot z(k+1)_{HP}}{z(k+2)_{HP} + 1} \end{bmatrix} \cdot \begin{bmatrix} x(k+1) \\ y(k+1) \\ z(k+1) \end{bmatrix} \\ &+ \begin{bmatrix} -\sin^{-1}(x(k+2)_{LP} / 9.81) & 0 & 0 \\ 0 & -\sin^{-1}(y(k+2)_{LP} / 9.81) & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \theta_v(k) \\ \phi_v(k) \\ \psi_v(k) \end{bmatrix} \\ &- \begin{bmatrix} \frac{10 \cdot x(k)_{HP} + 6.3167}{x(k+2)_{HP} + 1} & 0 & 0 \\ 0 & \frac{10 \cdot y(k)_{HP} + 6.3167}{y(k+2)_{HP} + 1} & 0 \\ 0 & 0 & \frac{10 \cdot z(k)_{HP} + 6.3167}{z(k+2)_{HP} + 1} \end{bmatrix} \cdot \begin{bmatrix} x(k) \\ y(k) \\ z(k) \end{bmatrix} \end{aligned} \quad (29)$$

4.4. Algorithme optimal (Linear Quadratic Regulator-LQR)

4.4.1. Définition de l'algorithme optimal

La Fig. IV-3 illustre la différence entre l'algorithme classique et l'algorithme optimal. Les lignes rouges illustrent la boucle fermée du LQR. L'idée principale de l'algorithme optimal LQR est de trouver une matrice de transfert linéaire $W(s)$ qui minimise un certain coût quadratique impliquant l'erreur e [GM61, Gue64, IS96, KNG+11, KP03, KBL+87, KMKL10, Kol95, DP97], sans permettre à la plate-forme de dépasser les contraintes du simulateur.

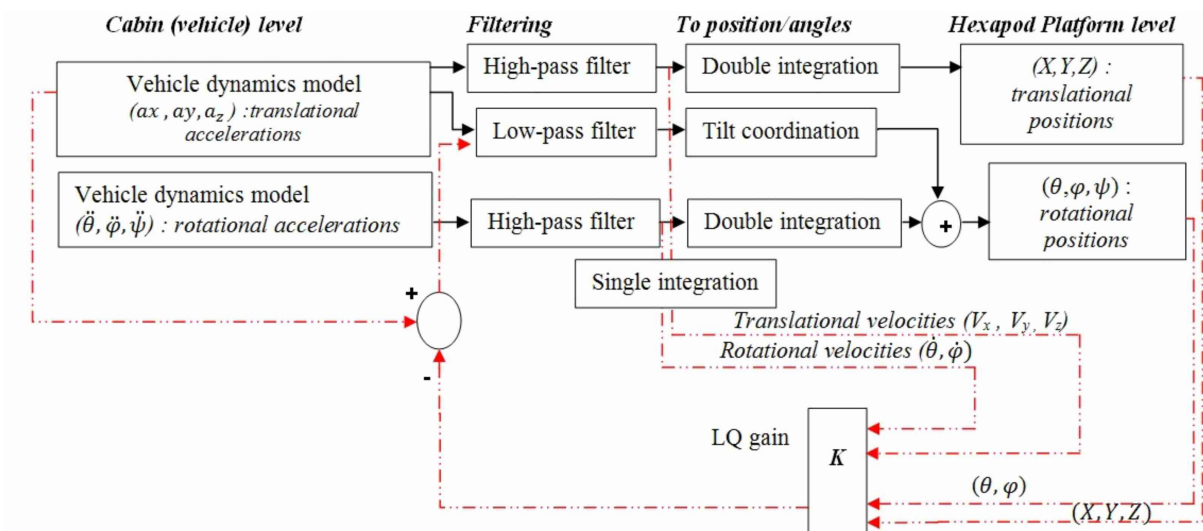


Fig. IV-3 : Algorithmes classique et optimal LQR

La Fig. IV-4 donne le détail du fonctionnement de l'algorithme. Ainsi, le conducteur est pris en compte à travers un modèle du système vestibulaire, et les contraintes physiques de la plateforme aussi.

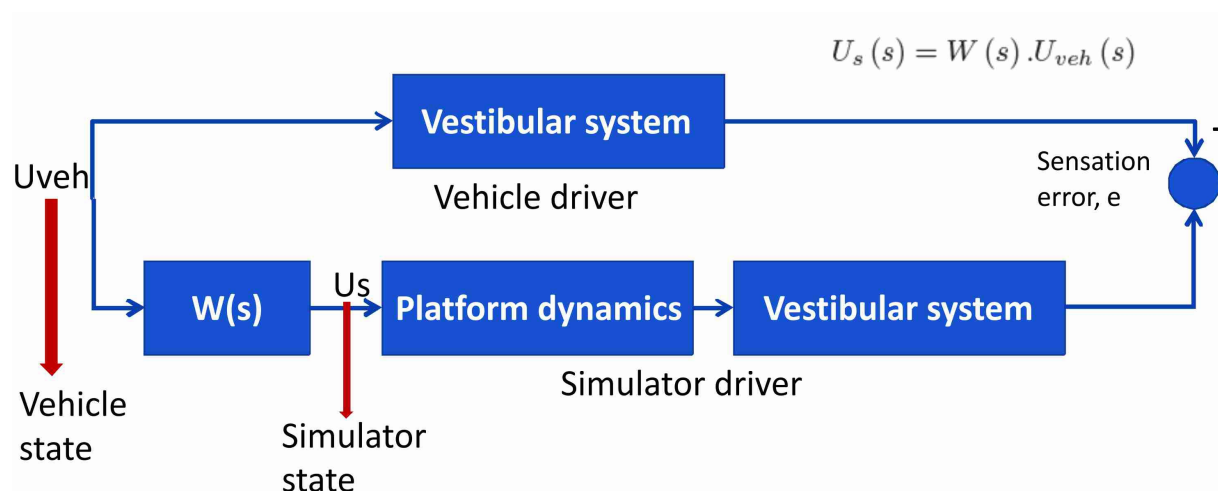


Fig. IV-4. : Conflit sensoriel d'inertiel-vestibulaire

4.4.2. Modélisation de l'algorithme optimal

L'algorithme de LQR repérage doit être construit avec un modèle mathématique du système vestibulaire humain. Pour cette étude, l'entrée u proposée pour le système vestibulaire est l'équation suivante :

$$u = \left[\int a_x dt \int a_y dt \int a_z dt \dot{\theta} \dot{\phi} \dot{\psi} \right]^T$$

Ensuite, le mouvement de rotation ressenti \hat{q} (vitesses de tangage, de roulis et de lacet) peut être donnée par le modèle mathématique des canaux semi-circulaires [CKK+03, TCG00]. Le mouvement de translation ressenti \hat{v} est associé à la force d'entrée spécifique du modèle de l'otolithe [CKK+03, TCG00]. Comme les variables X_{veh} et X_s représentant les états du système vestibulaire et du véhicule, le vecteur d'état correspondant peut être défini comme X_e [TCG00] :

$$X_e \triangleq X_s - X_{veh} \quad (1)$$

De plus, le filtre *washout* doit prendre l'état X_d du simulateur en compte, ce qui est nécessaire pour garantir que le simulateur obéit aux contraintes. La définition de l'état X_d du simulateur est la suivante :

$$X_d = \left[\iint a_x^s dt^2 \quad \iint a_y^s dt^2 \quad \iint a_z^s dt^2 \quad \int \dot{\theta}^s dt \quad \int \dot{\phi}^s dt \quad \int \dot{\psi}^s dt \right]^T \quad (2)$$

où $a_x^s, a_y^s, a_z^s, \dot{\theta}^s, \dot{\phi}^s, \dot{\psi}^s$ sont les accélérations de translation (longitudinale, latérale et verticale) et angulaires (tangage, roulis et lacet) dans la plate-forme de simulation. Ensuite, avec les modèles dynamiques de la perception et de l'erreur, le problème de la conception d'un filtre de *washout* optimal se formalise par l'équation suivante :

$$J = \int_{t_0}^{t_1} (e^T Q e + u_s^T R u_s + x_d^T R_d x_d) dt \quad (3)$$

où $[t_0, t_1]$ est la durée d'entraînement du simulateur, Q , R et R_d sont les matrices de pondération. La solution de ce problème d'optimisation peut être résolue par une équation de Riccati avec MATLAB [APMK12b, CKK+03, HL05], ce qui donne la matrice de transfert $W(s)$. Les matrices de pondération Q , R et R_d sont déterminées, après filtrage. La matrice $W(s)$ correspondante ainsi obtenue nous permet de calculer u_{veh} . La procédure est répétée jusqu'à ce que toutes les contraintes de simulation soient satisfaites :

$$\begin{aligned} |\dot{\theta}^s(t)| \leq \dot{\theta}_{max}^s, \quad |a_x^s(t)| \leq a_{x,max}^s, \quad |\theta^s(t)| \leq \theta_{max}^s, \quad |V_x^s(t)| \leq |V_{x,max}^s| \quad \text{et} \quad |X^s(t)| \leq \\ |X_{max}^s|, \quad |\dot{\phi}^s(t)| \leq \dot{\phi}_{max}^s, \quad |a_y^s(t)| \leq a_{y,max}^s, \quad |\phi^s(t)| \leq \phi_{max}^s, \end{aligned}$$

$$|V_y^s(t)| \leq |V_{y,max}^s| \text{ et } |Y^s(t)| \leq |Y_{max}^s|, |\dot{\psi}^s(t)| \leq \dot{\psi}_{max}^s, |a_z^s(t)| \leq a_{z,max}^s, |\psi^s(t)| \leq \psi_{max}^s, |V_z^s(t)| \leq |V_{z,max}^s| \text{ et } |Z^s(t)| \leq |Z_{max}^s| \text{ pour } t \in [t_0, t_1]$$

où

$$\begin{aligned} & \dot{\theta}_{max}^s, a_{x,max}^s, \theta_{max}^s, V_{x,max}^s, X_{max}^s \\ & \dot{\phi}_{max}^s, a_{y,max}^s, \phi_{max}^s, V_{y,max}^s, Y_{max}^s \\ & \dot{\psi}_{max}^s, a_{z,max}^s, \psi_{max}^s, V_{z,max}^s, Z_{max}^s \\ & \text{pour } t \in [t_0, t_1] \end{aligned}$$

sont les valeurs maximales (vitesses angulaires, accélérations, vitesses et déplacements) admissibles par la plateforme de simulation.

L'algorithme LQR est utilisé afin de produire une rétroaction qui stabilise le système [Lew92, Xu11].

4.5. Algorithme hybride (MRAC+LQR)

C'est le couplage de l'algorithme adaptatif MRAC et de l'algorithme optimal LQR. Les algorithmes sont activés simultanément pendant le fonctionnement de la plateforme dynamique. A titre d'exemple, la Fig. IV-5 montre la réduction du roulis grâce à l'utilisation de l'algorithme hybride. L'apport de la solution hybride se justifie pour un large panel de scénarios. En effet, les avantages des algorithmes optimal ou adaptatif sont utilisés dans la version hybride ce qui permet de réduire davantage le mal du simulateur.

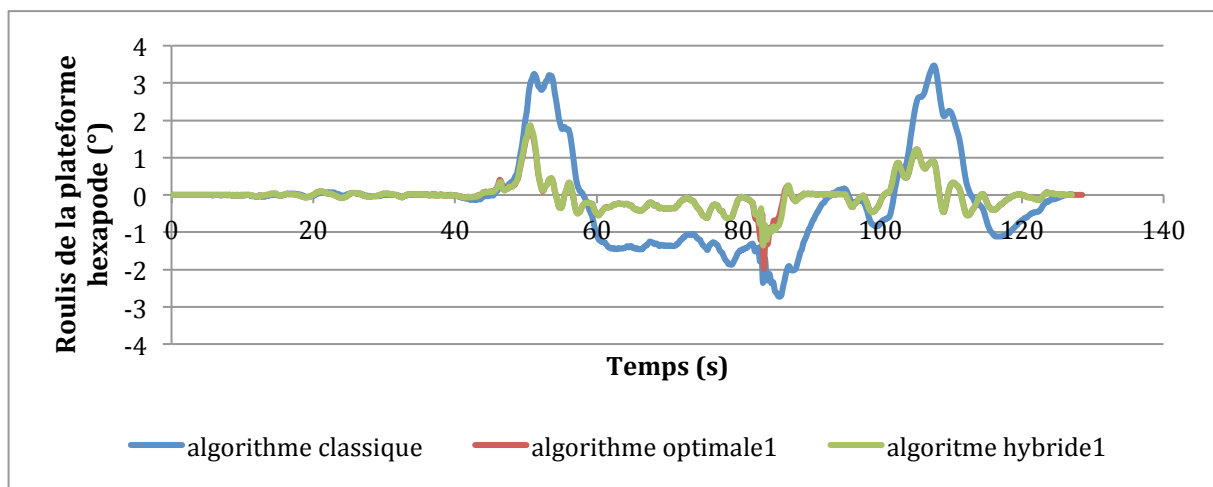


Fig. IV-5 : Roulis de la plateforme en fonction du temps et selon les algorithmes

Chapitre V

Conditions Expérimentales

Au cours de ces travaux de recherche plusieurs scénarios ont été développés. L'ensemble de ces scénarios a été mis en place sur le simulateur SAAM de l'Institut Image. Le simulateur a été utilisé aussi bien en mode statique qu'en mode dynamique. Le logiciel SCANeR Studio® (Oktal) a permis de modifier les algorithmes de pilotage de la plateforme et a permis aussi d'enregistrer les valeurs liées au comportement de la plateforme en vue de les analyser.

Les scénarios 1, 2 et 6 sont développés dans ce manuscrit, en revanche les scénarios 3, 4 et 5 ne sont pas présents dans ce manuscrit mais sont en cours de publication et sont soumis à des revues.

5.1. Scénario 1 (Chicane)

Le scénario 1 consiste à effectuer un évitement à droite puis à gauche. Ce scénario a pour objectif de mettre les algorithmes à contribution dans le pilotage de la plateforme et en particulier mettre en évidence leur capacité à restituer les accélérations latérales. La Fig. V-1 illustre la manœuvre de chicane [APMK12a].



Fig. V-1 : Manœuvre de changements de couloir dans SCANeR studio.

5.2. Scénario 2 (Country Road)

La Fig. V-2 illustre le scénario «Country Road» et le trajet associé [APM+11, APMK12b]. La partie gauche de la figure montre l'aspect du terrain et la partie droite montre en rouge le trajet suivi par les conducteurs. La Fig. V-3 compare l'effet de la commande sur l'IR (Illness Rating). On voit que la commande hybride a fortement diminué l'IR longitudinale (l'effet de contrôle en boucle fermée), non seulement après la manœuvre de freinage, mais aussi plus tôt dans le parcours (les valeurs IR ont atteint 3.7 pour l'algorithme classique où il a atteint 0.7 pour l'algorithme hybride).

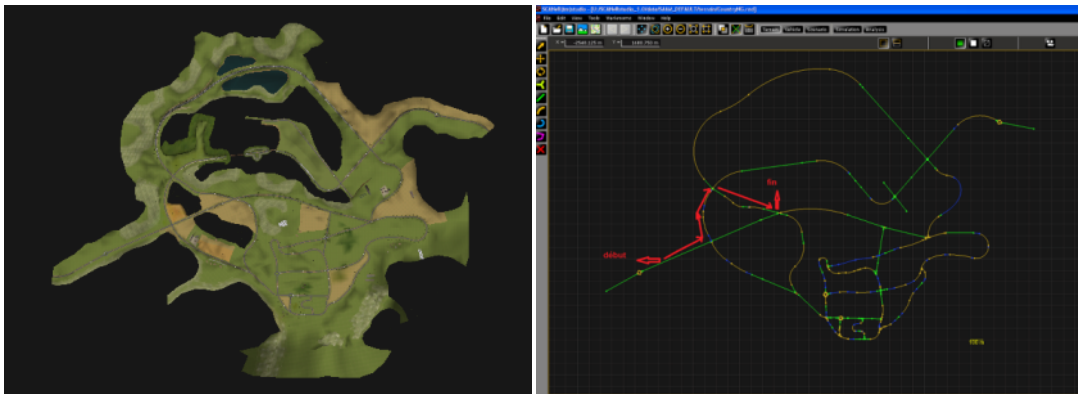


Fig. V-2 : Country scenario d'OKTAL SCANeR studio et trajet.

Avant chaque essai, les participants ont eu un entraînement afin de se familiariser avec le simulateur pour réduire la probabilité d'une mauvaise évaluation.

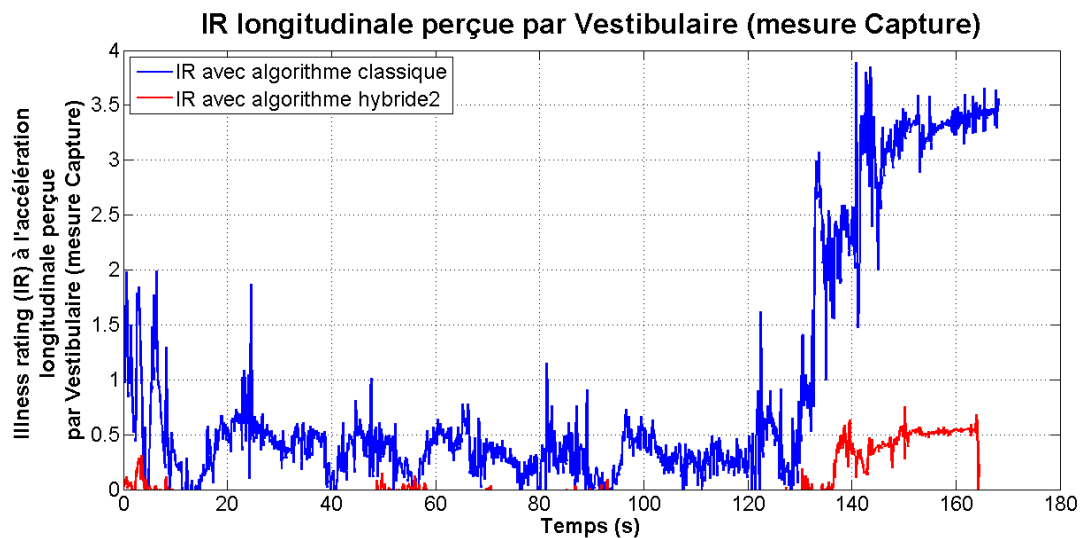


Fig. V-3 : IR longitudinal mesuré au niveau vestibulaire - scénario «Country Road» pour un sujet

5.3. Scénario 3 (Rugosité sinusoïdale)

Ce scénario a pour objectif d'évaluer les accélérations longitudinales (tangage) et verticales. Pour ce faire, un profil de route sinusoïdal a été réalisé. Le Tableau V-1 donne les conditions d'ondulation du profil de la route qui ont été testées sur le simulateur de conduite dynamique.

Tableau V-1 : Bosses (ondulation sinusoïdale)

Amplitude (m)	Frequence (Hz)	Vitesse longitudinale du véhicule (km/h)	Type du simulateur
0.02	1	50	SAAM (dynamique)
0.02	0.5	40	SAAM (dynamique)
0.02	1	40	SAAM (dynamique)
0.05	0.5	50	SAAM (dynamique)
0.05	0.5	40	SAAM (dynamique)
0.05	0.5	40	SAAM (statique)

5.4. Scénario 4 (Courbure de la route)

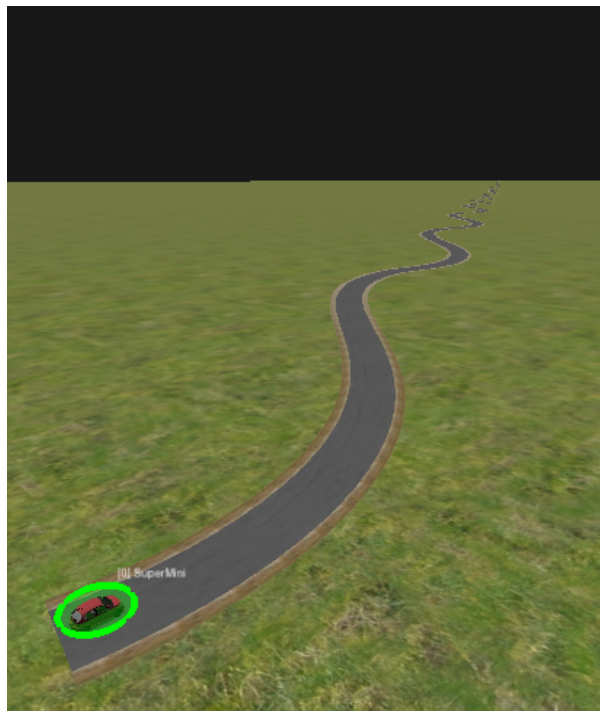


Fig. V-4 : Aspect du trajet (clothoïde)
Rayon de courbure de 50 m et longueur 200 m

La Fig. V-4 montre la route utilisée pour le scénario testé. Le rayon de courbure des virages est 50 m et la longueur totale est de 200 m. Le Tableau V-2 résume les paramètres qui ont été utilisés pour l'évaluation de la dynamique latérale.

Tableau V-2 : Courbes

Rayon (m)	Vitesse longitudinale du véhicule (km/h)	Accélération latérale max (m/s^2)	Type du simulateur
50	50	3.85	SAAM (dynamique)
50	50	3.85	SAAM (statique)
50	40	2.46	SAAM (dynamique)
50	50	3.85	ECO ₂ (statique)

L'idée ici est d'évaluer l'accélération latérale maximale sur le mal du simulateur (avec deux accélérations différentes latérale maximale, avec et sans activation de la plate-forme). De plus, l'impact du champ visuel a été étudié en utilisant l'ECO₂ et le simulateur SAAM en mode statique.

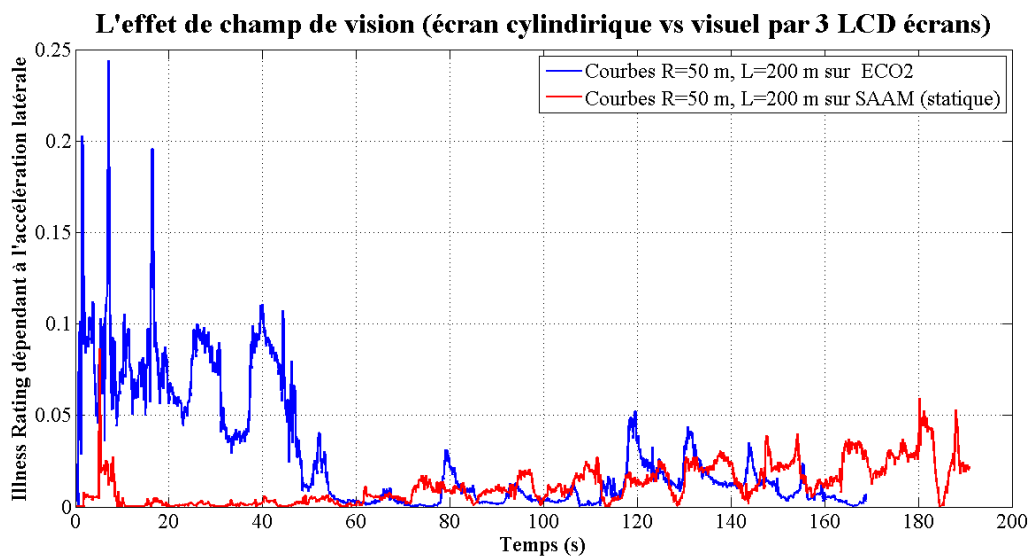


Fig. V-5 : Comparaison IR latérale vestibulaire sue ECO₂ et SAAM (statique)

La Fig. V-5 montre que le mal du simulateur (IR) est plus faible dans le cas de conduite avec le simulateur SAAM. L'immersion visuelle étant plus importante dans le simulateur SAAM (écran circulaire de 150°) que dans le simulateur ECO₂ (3 écrans LCD, champs visuel plus limité).

5.5. Scénario 5 (Braquage sinusoïdal)

Les conducteurs sont amenés à se déplacer sur une route dont les virages représentent une sinusoïde. Cette configuration est importante pour les constructeurs automobiles car ils permettent d'évaluer la stabilité latérale du véhicule ainsi que la réactivité des conducteurs. La Fig. V-6 représente la vitesse du véhicule pendant la phase de conduite du scénario de braquage sinusoïdal.

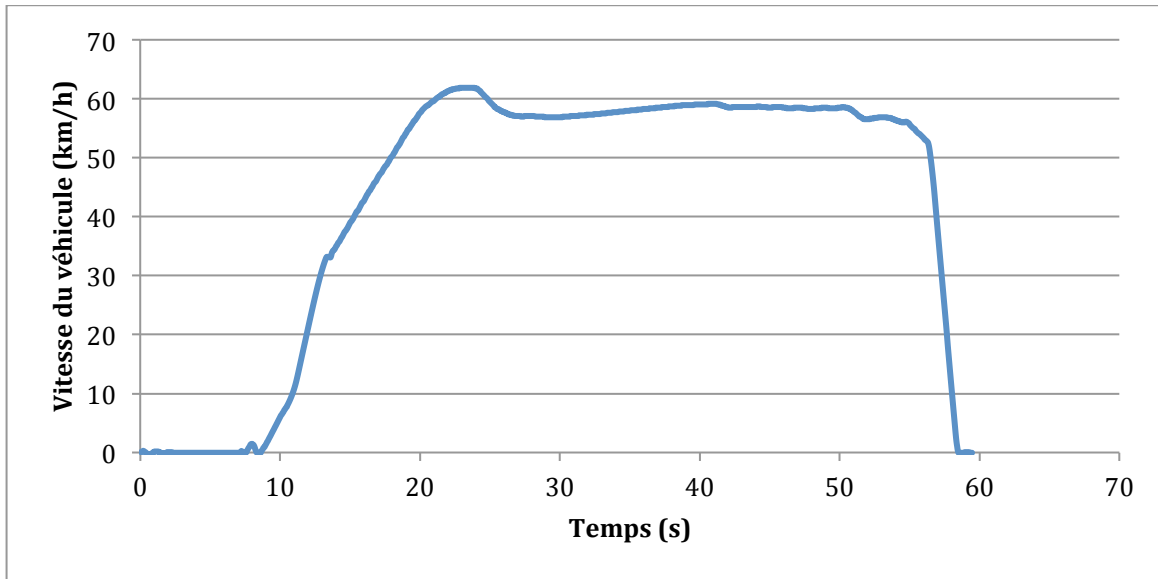


Fig. V-6 : Vitesse du véhicule scénario de braquage sinusoïdal

La Fig. V-7 donne le trajet du véhicule en scénario du braquage sinusoïdal. On remarque l'aspect sinusoïdale du déplacement.

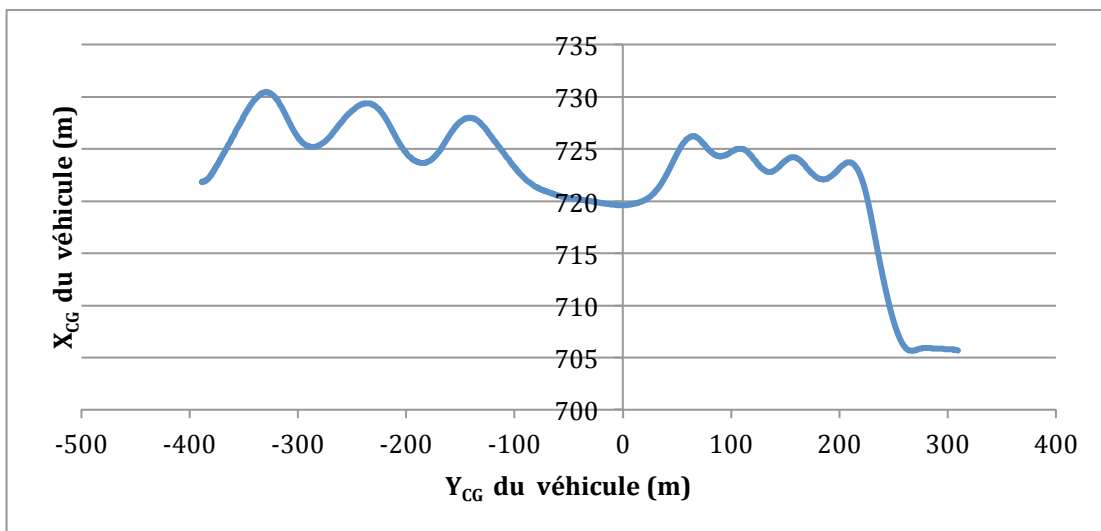


Fig. V-7 : Déplacement du véhicule lors du scénario de braquage sinusoïdal

5.6. Scénario 6 (Country Road pour 15 min)

La Fig. V-8 donne la vitesse et le centre de gravité du véhicule (CG) XY trajectoire pendant l'expérience pour l'expérimentations d'effet engagemment du sujet (voir chapitre 7).

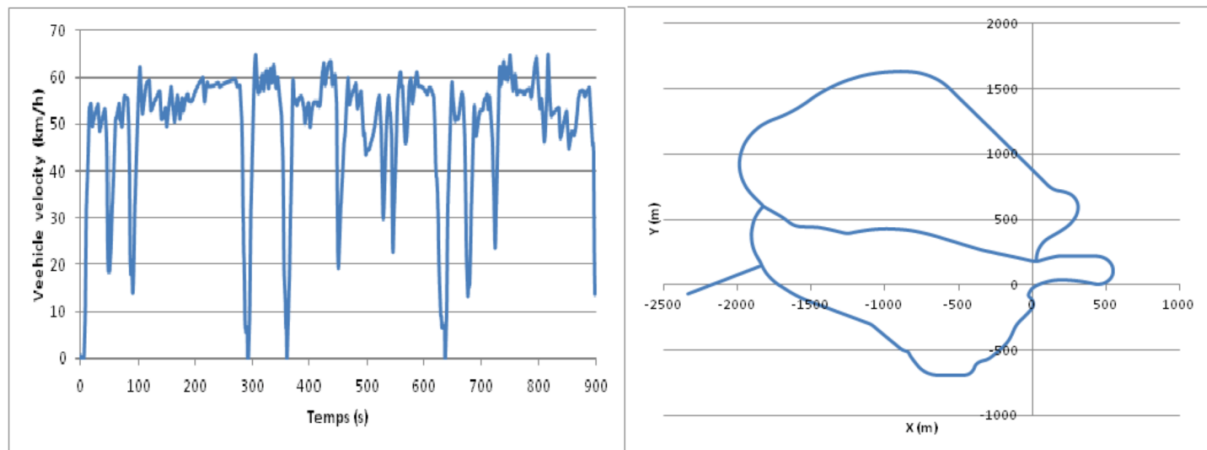


Fig. V-8 : La vitesse et le centre de gravité du véhicule (CG) XY trajectoire pendant l'expérience.

Pour améliorer l'engagement du sujet , nous proposons un guide visuel. Le guide visuel est conçu pour diminuer le mal des transports du passager. Le but est de provoquer une rotation de la tête, conformément à la route. Le principe consiste à afficher une ligne sur la scène qui tourne en conformité avec l'inclinaison de la route. Le passager est invité à incliner la tête en fonction de la rotation de la ligne.

La Fig. V-9 montre les mouvements de la ligne verte dans les séances d'entraînement. La barre verte peut tourner uniquement sur l'horizon avec un angle maximum de $\pm 0,2$ rad ($\sim 11,5$ degrés). Le côté droit de la Fig. V-9 montre la rotation de la ligne lorsque le véhicule entre dans un virage à droite alors qu'il reste horizontal lorsque le véhicule roule en ligne droite, illustré sur le gauche de la figure.



Fig. V-9 : Restitution (ligne verte) mouvements visuo-vestibulaire)

Avec ce guide visuel, nous créons un état où le passager est plus engagé. Notre hypothèse est que les passagers guidés ressentent un conflit inertio-vestibulaire plus faible que les passagers non guidés.

Chapitre VI

Effet de la restitution inertielle

6.1. Introduction

Les indices sensoriels (visuel, auditif, tactile, inertielle, vestibulaire, neuromusculaire) jouent un rôle important dans la qualité de la perception en simulation de conduite [KEM03]. Les simulateurs de conduite visent à donner une sensation de conduite aussi réaliste que possible. Pour atteindre cet objectif, le simulateur de conduite doit renforcer l'immersion virtuelle du sujet. Pour cette raison, restituer les signaux inertiels sur simulateurs de conduite est essentiel pour acquérir un fonctionnement plus propre [Kol95]. En raison de l'espace de travail restreint, il n'est pas possible de représenter la dynamique du véhicule en permanence à l'échelle 1 sur la plate-forme mobile. Néanmoins, l'objectif le plus recherché est de minimiser l'écart des accélérations entre le visuel (modèle du véhicule) et la dynamique inertielle (plate-forme mobile) [BSPB10].

L'objectif de cette étude est de proposer une méthode de mesure bimodale (objective et subjective) et en examinant leur corrélation nous pourrions évaluer le niveau de mal de simulateur des sujets lors d'une phase de conduite.

Cette expérimentation vise à valider les hypothèses suivantes :

Il existe une corrélation significative entre le niveau de mal du simulateur (mesure objective) et la perception de ce mal (mesure subjective) dans le cas des simulateurs statiques et dynamiques

Il existe une différence de perception entre le simulateur statique et dynamique en terme de mal des transports

L'approche proposée consiste à évaluer le mal des transports au niveau du conducteur. Afin de prendre en compte la dynamique vestibulaire (accélérations longitudinale, latérale et verticale), une mesure au niveau de l'oreille droite des sujets via un capteur Xsens est réalisée

(cf. Chapitre III). Puis ces mesures d'accélération de la tête ont été converties en «Note maladie». Le même scénario de manœuvre de changement de voie à vitesse constante (60 km/h) dans les mêmes conditions ont été conduits en statique puis en dynamique. Le scénario a été développé avec le logiciel SCANeRstudio 1.1 de la société OKTAL.

6.2. L'approche proposée

La mesure des accélérations permet de calculer le coefficient MSDV ainsi que l'indice IR (cf. Chapitre III). La Fig. VI-1 montre l'approche que nous avons utilisée pour permettre la corrélation (coefficient de corrélation de Pearson avec un intervalle de confiance de 95%) entre les données objectives et subjectives. Selon cette figure, nous avons évalué le degré de mal de simulateur au niveau vestibulaire de manière objective (mesure capteur) et subjective (questionnaire).

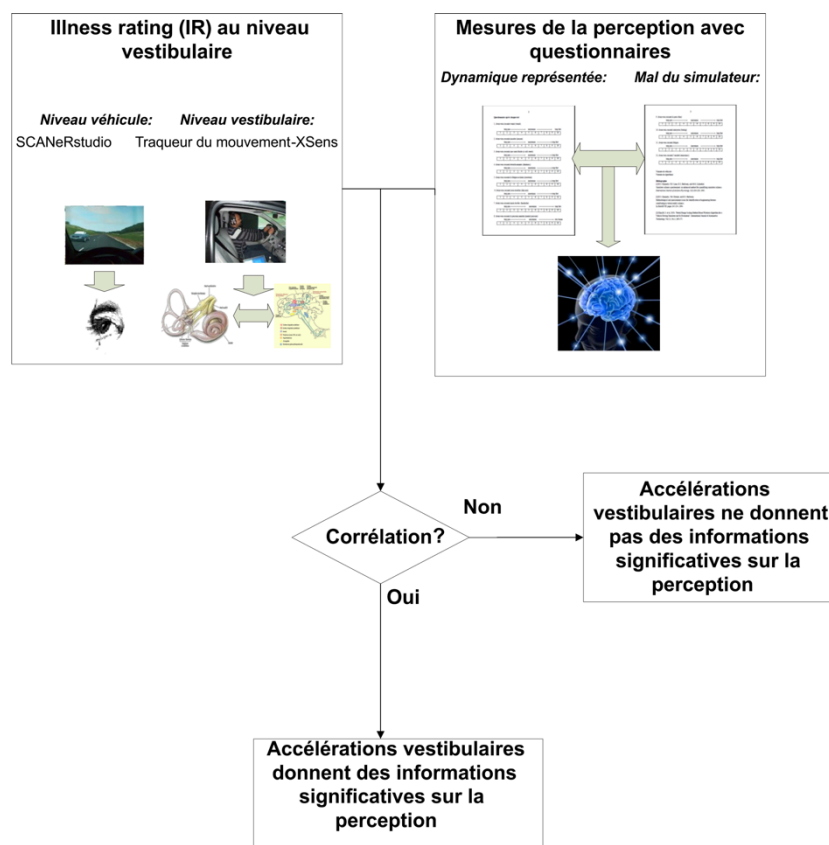


Fig. VI-1 : Procédure de corrélation entre les mesures objectives et subjectives

Le Tableau VI-1 ci-dessous regroupe les questions qui ont été posées à chacun des sujets à l'issue de chaque phase de conduite [APM+11].

Questions	
Q1	Etiez-vous enclin à vomir ?
Q2	Avez-vous ressenti des nausées ?
Q3	Avez-vous eu des sueurs froides ?
Q4	Avez-vous ressenti des étourdissements ?
Q5	Avez-vous ressenti de la fatigue visuelle ?
Q6	Avez-vous eu mal aux yeux ?
Q7	Avez-vous eu des maux de tête ?
Q8	Avez-vous ressenti une pression mentale ?
Q9	Avez-vous eu peur ?
Q10	Vous êtes-vous ennuyé ?
Q11	Avez-vous trouvé l'expérience fatigante ?
Q12	Etiez-vous anxieux ?

Tableau VI-1 : Questionnaire proposé aux sujets

Pour chacune des questions, le sujet devait répondre avec une valeur comprise entre 1 et 10, 1 représentant le niveau bas et 10 le niveau haut.

6.3. Protocole

La Fig. VI-2 illustre la vue de dessus de la trajectoire qui a été proposée lors de l'expérimentation. La position du circuit ainsi que l'espacement entre les cônes sont définis à partir de la largeur et la longueur du véhicule (W et L).

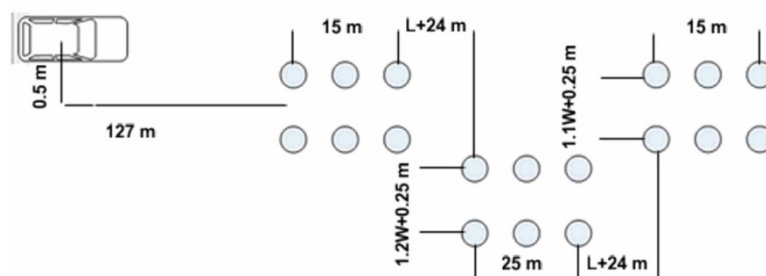


Fig. VI-2 : Chicane de l'OTAN [GBE+98, YG11]

Les sujets ont été invités à conduire sur cette chicane en mode statique et dynamique. Après chaque tentative, les sujets ont rempli le questionnaire d'évaluation subjective de la «perception due à la psychophysique». En même temps, au cours de chaque essai, les accélérations ont été enregistrées avec une période d'échantillonnage de 0.05 s. Chaque sujet a eu une phase d'apprentissage. Les tests ont été réalisés à une vitesse de conduite constante de $V_x = 60 \text{ km / h}$ et une durée qui varie de 37-50 secondes. Les questionnaires du mal du simulateur (SSQ) [KLBL93], de cinétose (MSQ) [KFBL92] et les méthodes de biofeedback (BFM) [KMKL10] ont été proposés aux sujets et les coefficients MSDV [Gri96, ISO97] et IR calculés. Le Tableau VI-2 montre les paramètres qui ont été utilisés dans ces expériences pour les algorithmes de pilotage.

Symbol	Longitudinal	Lateral	Roll	Pitch	Yaw
2 nd order LP cut-off frequency (Hz)			0.3	0.7	
2 nd order LP damping factor			0.3	0.7	
1 st order LP time constant (s)	0.1	0.1			0.1
2 nd order HP cut-off frequency (Hz)	0.5	0.5			2
2 nd order HP damping factor	1	1			1
1 st order HP time constant (s)	2	2			2

Tableau VI-2 : Paramètres d'algorithme classique

La fréquence de coupure du filtre passe bas du second ordre, ainsi que le facteur d'amortissement servent à la coordination d'inclinaison (roulis et tangage). La constante de temps du filtre passe-bas du premier ordre illustre les retards pour les accélérations. La fréquence de coupure du filtre passe haut du second ordre, ainsi que le facteur d'amortissement interviennent pour l'apparition du mouvement de lacet et les accélérations longitudinale et latérale. La constante de temps du filtre du premier ordre passe haut pondère les délais d'apparition des signaux.

6.4. Sujets

Vingt sujets, dix-huit hommes et deux femmes, avec une répartition par âge de 28.6 ± 5.97 années et une distribution d'expérience de conduite de 9.4 ± 6.11 années ont participé aux expériences. Ils ont eu un entraînement de familiarisation sur le simulateur de conduite avant chaque session afin de vérifier s'il y avait une prédisposition en ce qui concerne le mal du simulateur, pour éviter les mauvaises réponses et les aider à évaluer le plus objectivement possible.

6.5. L'analyse des données

L'analyse en composante principale (PCA) est une technique statistique utilisée pour la réduction de la dimension et la représentation des ensembles de données multidimensionnelles représentées par une matrice X de dimension $N \times p$ de N observations et p variables [HLP09]. Cette méthode transforme un jeu de données multi-variables corrélées entre elles dans un ensemble de données non corrélées. Avant l'analyse, les colonnes de X sont centrées si les variables ont des variances très hétérogènes. En plus de ce centrage, lorsque les variables sont mesurées avec des unités différentes, il est d'usage de normaliser chaque variable. Ceci est obtenu en divisant chaque variable par sa norme. Dans ce cas, l'analyse est appelée PCA corrélation.

Pour un ensemble de données de dimension p , le procédé PCA applique une transformation orthogonale sur un sous-espace de dimension m ($1 < m < p$) défini par des axes orthogonaux appelés axes principaux le long desquels la variance des données est maximisée. Les axes principaux peuvent être donnés par les vecteurs propres principaux associés aux valeurs propres de la matrice de corrélation. Les valeurs des données le long de ces axes sont appelées composantes principales. En fait, un petit nombre de composantes principales permet de décrire plus que la variance de l'ensemble des données d'origine et sont donc utilisées comme des dispositifs pour remplacer la représentation des données d'origine.

Comme les variables sont normalisées, elles peuvent être représentés par des vecteurs situés sur une sphère de dimension m , ce qui nous permet de projeter sur les deux premiers axes principaux. Ces projections sont des vecteurs dont les coordonnées sont les corrélations entre les variables avec les deux premières composantes principales.

Ainsi, les variables seront positionnées à l'intérieur d'un cercle appelé cercle de corrélation qui est utile pour évaluer les affinités et les antagonismes entre les variables. Plus une variable est dans le cercle de corrélation, mieux nous pouvons reconstruire cette variable à partir des

deux premiers paramètres. Dans ce cas, si les variables sont proches les unes des autres, elles sont sensiblement en corrélation positive (r proche de 1), si elles sont orthogonales, elles ne sont pas corrélées (r proche de 0), si elles sont diamétralement opposées, elles sont en corrélation négative (r proche de -1). Quand une variable est proche du centre du cercle, elle est moins importante pour les deux premiers paramètres.

Dans cette étude, la méthode PCA a été utilisée pour comparer la perception du mal des transports entre un fonctionnement du simulateur en statique et en dynamique. Par la suite, la perception de fidélité a été vérifiée à l'aide de la corrélation de Pearson entre les données objectives et les données subjectives.

6.6. Résultats et discussion

Chaque sujet a répondu à deux questionnaires, un pour l'expérimentation en statique et un pour l'expérimentation en dynamique. Grâce aux mesures des accélérations, nous avons calculé les valeurs de IR pour chaque expérimentation pour chaque sujet. De ces questionnaires et de ces valeurs de IR (cf. Tableau VI-4 et Tableau VI-5), nous avons construit une matrice qui a été utilisée pour l'analyse statistique PCA. Elle a été réalisée en utilisant le langage R [GIB+97]. Sur le côté droit de la Fig. VI-3, on voit que les deux premières valeurs propres de la matrice de corrélation correspondent respectivement à 63.42% et 13.38% de la variance. Sur le côté gauche de la Fig. VI-3 les données d'évaluation subjective ont été projetées sur le cercle des corrélations sur les deux premiers axes principaux associés respectivement aux composantes principales Comp1 (axe horizontal) et Comp2 (axe vertical). La plupart des variables originales projetées ont une bonne représentation sur le plan principal engendré par les deux premiers axes principaux. Il apparaît que les données subjectives sont corrélées positivement avec Comp1 pour le fonctionnement en mode dynamique et que ces données sont corrélées négativement avec Comp1 pour le fonctionnement en mode statique. Une corrélation négative montre l'existence d'un conflit sensoriel.

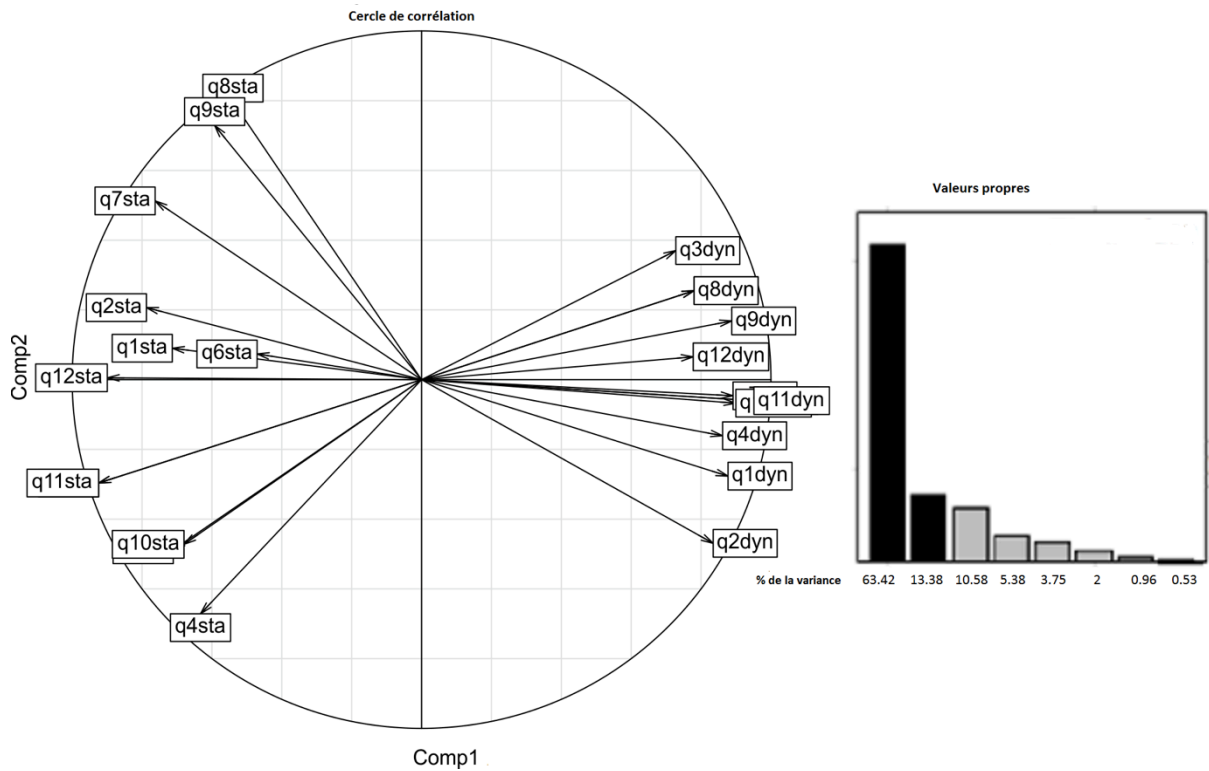


Fig. VI-3 : Cercle de corrélation et valeurs propres de l'analyse en composantes principales pour les conditions statiques et dynamiques

	Inertie des valeurs propres	Pourcentage de la variance(%)
Comp1	15,22	63,42
Comp2	3,21	13,38
Comp3	2,54	10,58
Comp4	1,29	5,38
Comp5	0,9	3,75
Comp6	0,48	2
Comp7	0,23	0,96
Comp8	0,096	0,53

Tableau VI-3 : L'inertie des valeurs propres et le pourcentage de la variance

Variables	IR _x	IR _y	IR _z	IR _{tot}
q1 _{STA}	0.625	0.108	-0.302	0.538
q2 _{STA}	0.453	-0.145	-0.211	0.385
q3 _{STA}	0.836	0.213	0.367	0.855
q4 _{STA}	0.922	0.006	0.602	0.970
q5 _{STA}	0.948	0.175	0.363	0.958
q6 _{STA}	0.661	0.054	-0.275	0.575
q7 _{STA}	0.135	-0.012	-0.561	0.029
q8 _{STA}	-0.308	0.174	-0.617	-0.391
q9 _{STA}	-0.335	0.300	-0.681	-0.424
q10 _{STA}	0.984	0.187	0.134	0.953
q11 _{STA}	0.836	0.213	0.367	0.855
q12 _{STA}	0.613	0.149	0.134	0.603

Les valeurs en gras correspondent à p-value < 0.05

Tableau VI-4 : Coefficient de Pearson (r) des données sub. et obj. en statique

Variables	IR _x	IR _y	IR _z	IR _{tot}
q1 _{DYN}	-0.291	-0.262	0.112	-0.292
q2 _{DYN}	0.496	-0.250	0.694	0.498
q3 _{DYN}	-0.283	0.902	-0.114	-0.221
q4 _{DYN}	0.409	-0.259	0.765	0.416
q5 _{DYN}	0.730	0.118	0.906	0.758
q6 _{DYN}	0.415	0.597	0.599	0.468
q7 _{DYN}	0.730	0.118	0.906	0.758
q8 _{DYN}	0.415	0.597	0.599	0.468
q9 _{DYN}	0.415	0.597	0.599	0.468
q10 _{DYN}	0.730	0.118	0.906	0.758
q11 _{DYN}	0.730	0.118	0.906	0.758
q12 _{DYN}	0.415	0.597	0.599	0.468

Les valeurs en gras correspondent à une p-value < 0.05

Tableau VI-5 : Coefficient de Pearson (r) des données sub. et obj. en dynamique

Selon les critères de corrélation, nous pouvons dire que :

- La question Q3 (sueur froide) est significativement corrélée avec la valeur de IR_y en dynamique, alors qu'elle est significativement corrélée avec IR_x et IR_{tot} en statique.
- La question Q4 (étourdissement) est significativement corrélée avec IR_z en dynamique, alors qu'elle est significativement corrélée avec IR_x et IR_{tot} en statique.

- La question Q5 (fatigue visuelle) est significativement corrélée avec IR_z et IR_{tot} en, alors qu'elle est en corrélation significative avec IR_x et IR_{tot} en statique.
- La question Q7 (maux de tête) est significativement corrélée avec IR_z et IR_{tot} en dynamique alors qu'elle n'existe pas de corrélation en mode statique.
- La question Q10 (ennui) est significativement corrélée avec la IR_z et IR_{tot} en dynamique, alors qu'elle est en corrélation significative avec IR_x et IR_{tot} en statique.
- La question Q11 (fatigue) est significativement corrélée avec la IR_x et IR_{tot} en statique alors qu'elle est significativement corrélée avec IR_z et IR_{tot} en dynamique.

6.7. Conclusion

La méthode PCA montre qu'il y a une corrélation négative sur la composante principale (Comp1) entre le mode statique et le mode dynamique. Cela indique une contribution positive de la plate-forme en mode dynamique. En effet, la Fig. VI-3 montre que la corrélation est positive pour l'aspect dynamique (valeurs proche de 1, cercle unité) ce qui induit que le modèle physique et l'évaluation subjective (cognition) sont convergents. En revanche, la corrélation est négative pour l'aspect statique (valeurs proche de -1) ce qui induit que le modèle physique et l'évaluation subjective (cognition) sont divergents.

Afin d'évaluer la fidélité de la perception, la corrélation de Pearson entre les données subjectives et objectives a été vérifiée. Avoir des sueurs froides (indicateur de la peur), étourdissements (désorientation), sensation de fatigue visuelle (maladie visuelle), ennui (situation psychophysique), fatigue (situation psychophysique) ont une corrélation positive avec IR_x et IR_{tot} en statique. Ces résultats montrent donc pour le cas statique que le principal facteur qui induit le malaise est dû à l'accélération longitudinale (composante x) de la tête (niveau vestibulaire). Alors que les accélérations latérale (composante y) et verticale (composante z) de la tête ne sont pas corrélées significativement dans la perception du mal du mouvement. La dynamique totale de la tête a été également significativement corrélée avec l'indicateur de la peur, le sentiment de désorientation, de la maladie visuelle et la situation psychophysique.

Les sueurs froides sont significativement corrélées avec IR_y pour le cas dynamique. Cela montre que la dynamique latérale de la tête est le principal élément qui a induit un sentiment de peur. Par ailleurs sentir des vertiges, sensation de fatigue oculaire, sensation de l'ennui et fatigue sont significativement en corrélation positive avec IR_z et IR_{tot} . Ces résultats pour le

mode dynamique montrent que le principal facteur qui a provoqué la désorientation, les problèmes visuels et l'ennui sont liés à la dynamique verticale des sujets car la dynamique longitudinale et latérale de la tête n'ont pas eu un rôle important sur la perception de la maladie de mouvement.

Cette étude montre de manière statistique l'apport de la dynamique dans la réduction du mal du simulateur. Elle permet aussi de classer les différents maux selon les axes longitudinal, vertical et latéral du véhicule.

Chapitre VII

Influence de l'engagement du sujet en conduite simulée

7.1. Introduction

La simulation de conduite a pour but de restituer la sensation de conduite à la façon d'une situation réelle. Pour une situation similaire, le conducteur doit réagir de la même façon que dans la réalité. Pour permettre ce comportement similaire, le simulateur de conduite doit assurer l'immersion virtuelle du sujet dans la situation de conduite. Le sujet doit ainsi percevoir le mouvement de son propre corps dans une scène virtuelle avec une voiture virtuelle comme s'il était dans une voiture réelle. Pour cette raison, la restitution de retours inertiels est importante. Le mal de simulateur est l'un des sujets importants de recherche dans le cadre de la simulation de conduite. D'une part, la restitution inertielle joue un rôle prépondérant pour garantir une fidélité du comportement du conducteur dans son simulateur. Les problématiques scientifiques et technologiques sont alors liées au mouvement de la plateforme dynamique en lien avec le mouvement simulé du véhicule et le retour d'informations visuelles données au sujet. D'autre part, la façon dont le sujet est engagé dans son expérience de conduite simulée peut jouer un rôle significatif dans le mal de simulateur perçu.

Ce chapitre traite du mal du simulateur du point de vue du degré d'engagement du sujet en simulation de conduite avec une plateforme dynamique (utilisant les mêmes algorithmes de contrôle pour toute l'expérience). Des analyses sont faites en étudiant les effets d'accélération latérales induites par les virages car ce sont ces configurations qui apportent le plus de mal de transport en conduite automobile.

7.2. Approche proposée

7.2.1. Engagement du sujet en simulation de conduite

Le mal de simulateur dépend de beaucoup de facteurs au nombre desquels on trouve généralement les capacités et le contrôle de la plateforme de simulation. Mais le mal de simulateur dépend également de l'engagement du sujet dans la conduite.

Dans un simulateur de conduite, comme dans une conduite réelle, le conducteur est plus engagé que le passager dans son véhicule. Autrement dit, le conducteur est comme dans une relation fusionnelle avec son véhicule pendant la conduite (comme si le véhicule était un prolongement de son propre corps). Le conducteur contrôlant son véhicule, il anticipe les mouvements de ce dernier et le perçoit comme une partie de son propre corps. Aussi, sa tête oscille avec les mouvements du véhicule dans les virages afin que le sujet perçoive mieux les accélérations en fonction des retours proprioceptifs de l'ensemble du système (corps du conducteur et du véhicule). Comparativement, le passager anticipe bien moins que le conducteur les mouvements du véhicule. Il est plus dépendant du mouvement du véhicule et son corps subit les accélérations du véhicule. Pour cette raison, le passager est supposé être plus malade que le conducteur durant la conduite.

Si le sujet est engagé dans la conduite (en tant que conducteur), son mouvement de tête accompagne les mouvements de la voiture dans les virages. Le sujet penche la tête en fonction des virages. Ainsi, le sujet, de façon inconsciente, oriente sa tête en fonction de retours proprioceptifs de façon à ressentir le mal de simulateur le moins possible comme le montre la Fig. VII-1.

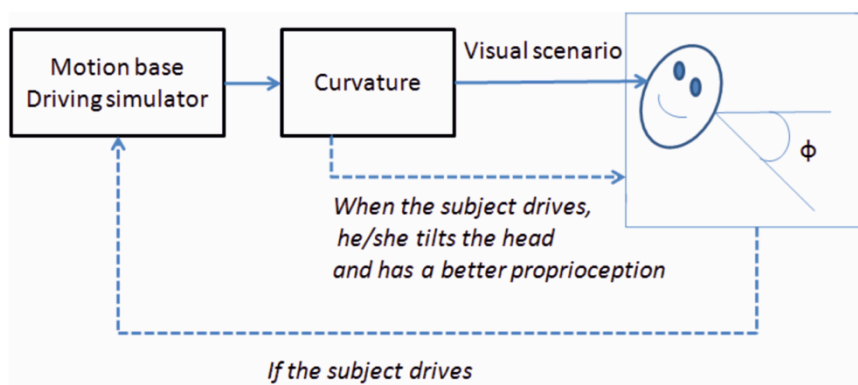


Fig. VII-1 : Rotation de la tête pour diminuer les conflits visuo-vestibulaire.

L'objectif de ce chapitre est d'étudier l'influence de l'engagement du sujet dans la conduite sur son mal de simulateur. Plusieurs verrous scientifiques doivent être adressés dans ce cadre :

- Proposition d'un protocole expérimental
- Proposition de différentes conditions en terme d'engagement du sujet dans la conduite
- Mesures du mal de simulateur

7.2.2. Guide visuel pour le passager

Afin d'améliorer l'engagement du sujet passager dans la conduite, un guide visuel est proposé pour le passager. Le guide visuel est conçu pour diminuer le mal de simulateur du passager. L'objectif est d'induire une rotation de la tête en fonction des virages de la route. Le sujet devrait être plus en accord avec le véhicule en terme de mouvements. Le principe est d'afficher une ligne dans la scène qui effectue une rotation en accord avec les virages de la route (mouvement latéral du véhicule induit par le virage). Lors de l'expérience, on demande au passager de tourner la tête en fonction de la rotation du guide visuel. La Fig. VII-2 montre les mouvements de la ligne verte au cours d'une session de conduite. Le guide visuel effectue des rotations sur l'horizon avec un angle de roulis de ± 0.2 rad (~ 11.5 degrés). La partie gauche de la figure montre la ligne en position horizontale dans une ligne droite. La partie droite de la figure montre la rotation de la ligne lorsque le véhicule entre dans un virage à droite.

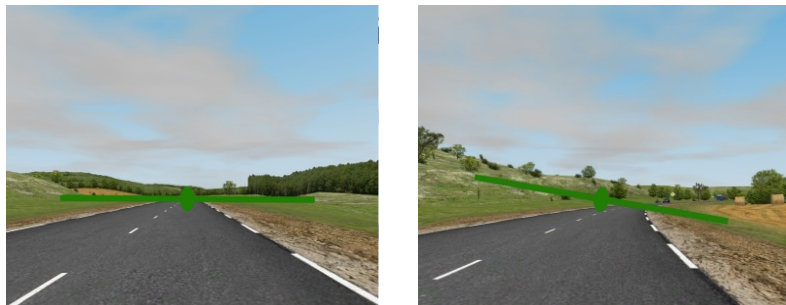


Fig. VII-2 : Induction de mouvements visuo-vestibulaires par guide visuel (ligne verte).

Avec le guide visuel, nous créons une situation où le passager est plus engagé dans la conduite. Notre hypothèse est que le passager guidé subit un conflit visuo-vestibulaire plus bas que le passager non guidé. Ainsi, le passager non guidé est supposé être plus malade que le passager guidé.

7.2.3. Evaluation du mal du simulateur

L'évaluation objective du mal de simulateur n'est pas aisée. Selon la littérature, cette évaluation peut être effectuée sur la base de l'accélération latérale au niveau vestibulaire. Les

situations de conduite faisant souvent l'objet de vibrations induites, la méthode de dose de valeur de vibrations (VDV) a été choisie. Cette méthode est également indiquée pour la restitution cumulative dans le temps de mesures de vibrations ; le mal de simulateur étant très sensible à l'accumulation temporelle. La mesure cumulative des vibrations permet de comparer les niveaux de vibrations cumulés au niveau de la plateforme (inertie) par rapport à ceux au niveau de la tête (vestibulaire) [Gri96]. Le calcul de VDV est basé sur l'accélération latérale. L'accélération latérale mesurée au niveau vestibulaire est calculée par l' Eq. (1) où :

$a_{y,sensed}$: accélération latérale perçue (m/s²)

a_{yv} : accélération latérale longitudinale au niveau vestibulaire (m/s²)

φ : angle de roulis au niveau vestibulaire (°)

g : accélération gravitationnelle (m/s²)

$$a_{y,sensed} = a_{yv} \cdot \cos \varphi + g \cdot \sin \varphi \quad (1)$$

L'accélération $a_{y,sensed}$ est mesurée au niveau de l'oreille droite du sujet grâce à un capteur accéléromètre pour le même scénario de conduite avec le même algorithme de contrôle de la plateforme dynamique. En accord avec la norme ISO 2631-1, les valeurs d'accélération RMS pour chaque axe sont définies pour refléter l'exposition du corps humain aux vibrations selon des critères de santé. Les coefficients sont décrits dans la norme sur la base des fréquences et directions de vibrations exposées au corps. Les coefficients $\omega_k = 0.426$ (axe céphalocaudal) et $\omega_d = 0.067$ (axes antéroposterieur et médiolatéral) sont utilisés pour obtenir des pondérations sur chacun des axes. Dans les équations (3) et (4), T est le temps d'exposition aux vibrations (durée d'une session de conduite dans notre expérience). Pour l'évaluation des effets sur la santé, le coefficient $k_y=1.4$ a été choisi. La valeur VDV_y a été calculée au niveau vestibulaire en utilisant l'accélération latérale " a_y " au niveau vestibulaire [AAL+07, Gri04, Ben88, ISO97, APMK12b]. L'équation du calcul de VDV est à la puissance 4 des termes RMS de l'accélération latérale en fonction du temps. Cette méthode est connue comme la méthode de la racine quatrième moyennée. Cette technique d'évaluation est ainsi sensible aux pics d'accélération [Gri96] .

$$a_{yRMS} = k_y \cdot \omega_d \cdot a_y \quad (2)$$

$$VDV = \left[\int_{t=0}^{t=T} a_{RMS}^4(t) dt \right]^{\frac{1}{4}} \quad (3)$$

$$VDV_y = \left[\int_{t=0}^{t=T} (k_y \cdot \omega_d \cdot a_y)^4(t) dt \right]^{\frac{1}{4}} \quad (4)$$

7.2.4. Questions de recherche

L'approche proposée est de comparer les conflits visuo-vestibulaires pour trois conditions expérimentales d'engagement du sujet comme conducteur, passager et passager guidé. Une mesure d'accélération au niveau de la plateforme dynamique permet de déterminer une valeur de dose de vibrations au niveau de la plateforme qui servira de référentiel pour nos comparaisons. Une mesure de l'accélération latérale au niveau de l'oreille interne décrira la valeur de dose de vibrations pour chacune des trois conditions expérimentales. Cette approche permet de considérer que des valeurs de dose de vibrations au niveau de l'oreille interne proches de celles au niveau de la plateforme seront le signe de conflit visuo-vestibulaire faible [Oma82, Kol95]. De façon inverse, si les valeurs sont divergentes, le mal de simulateur sera important.

Selon la problématique scientifique exposée, les questions de recherche relatives à l'engagement du sujet dans la conduite sont les suivantes :

- Est il possible de restituer une évaluation de mal de simulateur avec une mesure objective ?
- Quelle est l'influence de l'engagement du sujet dans sa conduite sur le mal de simulateur ?

7.3. Méthode expérimentale

7.3.1. Protocole

Trois conditions expérimentales ont été menées pour les mêmes sujets (N=16) sur un scénario spécifique sur simulateur temps réel. On a demandé aux sujets de conduire selon un scénario spécifique sur simulateur et de prendre la place en tant que conducteur, passager et passager guidé. Le scénario « country road » a été utilisé dans le cadre de cette expérience. La Fig. VII-3 décrit la trajectoire selon le plan de masse (XY) et les profils de vitesse du véhicule durant les phases expérimentales. L'ensemble de l'expérience a été réalisée avec une vitesse constante de 60 km/h en 15 min.

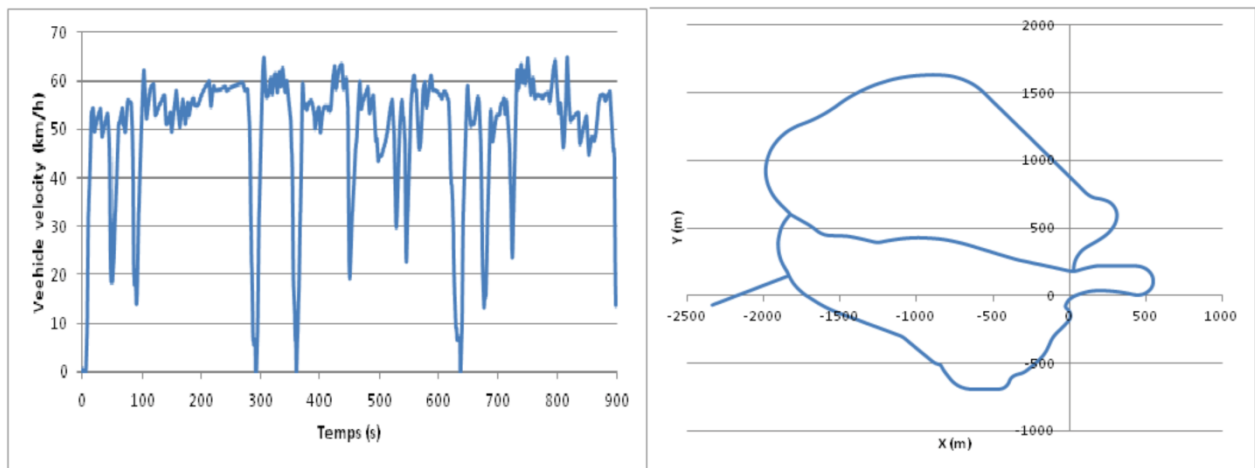


Fig. VII-3 : Vitesse du véhicule et trajectoire pendant l'expérience.

L'expérience consiste à comparer les valeurs de VDV extraites à partir des accélérations latérales sous trois conditions : conducteur, passager et passager guidé. Ces valeurs sont comparées à une valeur VDV de référence calculée à partir de l'accélération de la plateforme. Nos hypothèses sont les suivantes :

- Le passager est plus malade que le conducteur
- Le passager est plus malade que le passager guidé.

7.3.2. Mesures

Ici, les niveaux d'accélérations inertiels se réfèrent aux valeurs d'accélérations mesurées sur la plateforme dynamique qui bouge dans l'espace de travail du simulateur de conduite. Les niveaux d'accélération vestibulaires représentent les accélérations de la tête du sujet (oreille interne plus précisément). Ces accélérations vestibulaires sont mesurées grâce à un accéléromètre attaché à un casque (Fig. VII-4) [AMP+13]. Le capteur XSens a été mis en œuvre dans ce but (cf. Chapitre III). Ce capteur est capable de mesurer les données telles que les angles et accélérations de rotation selon les 3 axes de rotation d'un repère cartésien ainsi que les accélérations longitudinales selon les 3 axes du repère. Les données sont calibrées à l'aide de quaternion d'orientation. La fréquence d'échantillonnage de l'enregistrement des données est de 20 Hz. Pour le calibrage des données d'acquisition, une remise à zéro d'alignement a été choisie qui combine simplement l'objet et le « *heading reset* » au même instant. Cela permet d'aligner tout le système de coordonnées par une simple action [AMP+13, XSe10]. Les détails du capteur de mouvement XSens sont donnés dans [AMP+13, XSe10].

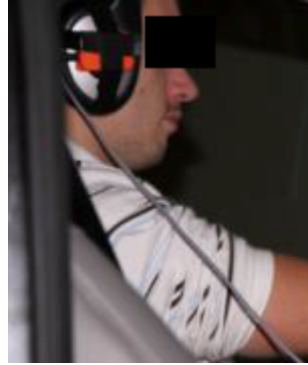


Fig. VII-4 : Système d'acquisition de la dynamique de la tête pendant les phases de conduite

L'accélération $a_{y,sensed}$ représente les mesures au niveau de l'oreille du sujet selon le principe de superposition. Les données au niveau du véhicule (enregistrées par le logiciel SCANeRstudio) peuvent être séparées en données de contrôle (angle volant, pédale d'accélération, pédale de frein...), mouvement de la plateforme (accélérations longitudinales et de rotation), données du véhicule (dynamique, moteur...), analyse fréquentielle du mouvement de la plateforme et du véhicule (en utilisant une décomposition en transformation rapide de Fourier [AMP+13]).

7.3.3. Panel de sujets

L'expérience a été réalisée pour analyser l'effet de l'engagement du sujet (sous les 3 conditions conducteur, passager et passager guidé) avec le même algorithme de mouvement de plateforme. 16 sujets (N=16, 13 hommes et 3 femmes) âgés (moyenne=31.2 ans, SD=7.9 ans) et possédant une certaine expérience de conduite (moyenne=12.6 ans, SD=7.3 ans) (SD : déviation standard) ont participé aux expériences de ces trois conditions.

7.4. Analyse du questionnaire

Le Tableau VII-1 met en évidence les résultats d'auto-évaluation (questionnaire SSQ cf. Chapitre II) pour les nausées, les étourdissements, la fatigue et l'anxiété à l'aide d'un test *t* bilatéral avec un intervalle de confiance de 95%. Il n'y a pas de différences significatives entre les passagers non guidés et les conducteurs, entre les passagers guidés et les conducteurs et entre les passagers non guidés et guidés car toutes les *p-value* sont supérieures à zéro (16 sujets, N = 16).

		Nausée		Etourdissement		Fatigue		Anxiété	
Auto-évaluation pour les passagers non guidés et les conducteurs	Type	Pass. non-guidés	Cond.	Pass. non-guidés	Cond.	Pass. non-guidés	Cond.	Pass. non-guidés	Cond.
	Moy	1.714	2.267	1.893	2.267	4.071	2.533	1.643	2.000
	Ecart type	±1.326	±2.017	±1.212	±1.335	±2.645	±1.807	±1.151	±1.464
Valeurs p de test t bilatéral		0.380>0.05		0.429>0.05		0.069>0.05		0.464>0.05	
Auto-évaluation pour les passagers guidés et les conducteurs (mean±sd)	Type	Pass. guidés	Cond.	Pass. guidés	Cond.	Pass. guidés	Cond.	Pass. guidés	Cond.
	Moy	2.000	2.267	1.833	2.267	2.917	2.533	1.667	2.000
	Ecart type	±2.558	±2.017	±1.030	±1.335	±1.881	±1.807	±1.073	±1.464
Valeurs p de test t bilatéra		0.768>0.05		0.341>0.05		0.592>0.05		0.495>0.05	
Auto-évaluation pour les passagers guidés et les passagers non guidés (mean±sd)	Type	Pass. non-guidés	Pass. guidés	Pass. non-guidés	Pass. guidés	Pass. non-guidés	Pass. guidés	Pass. non-guidés	Pass. guidés
	Moy	1.714	2.000	1.893	1.833	4.071	2.917	1.643	1.667
	Ecart type	±1.326	±2.558	±1.212	±1.030	±2.645	±1.881	±1.151	±1.073
Valeurs p de test t bilatéral		p=0.727>0.05		p=0.892>0.05		p=0.195>0.05		p=0.957>0.05	

Tableau VII-1 : Résultats de l'auto-évaluation

7.5. Etudes expérimentales

7.5.1. Analyses de données

L'influence de l'engagement du sujet dans la conduite sur son exposition aux vibrations (indicateur de mal du simulateur) a été étudiée. Les mesures d'accélération latérales ont été collectées au niveau du système vestibulaire du sujet (en utilisant un capteur de mouvement) et de la plateforme dynamique (en utilisant les données du logiciel de contrôle de la plateforme).

Les tests de Friedman avec un intervalle de confiance de 95% ont été mis en œuvre pour comparer les différences entre les valeurs VDV des accélérations vestibulaires (VDV_{y_vest}) avec les valeurs VDV des accélérations de la plateforme (VDV_{y_plat}). Les tests de Mann-Whitney U avec un intervalle de confiance de 95% ont été utilisés pour comparer les différences entre les valeurs VDV des accélérations vestibulaires des sujets conducteur avec les valeurs VDV des accélérations vestibulaires des sujets passager non guidé ainsi que pour comparer les différences entre les valeurs VDV des accélérations vestibulaires des sujets passager guidé avec les valeurs VDV des accélérations vestibulaires des sujets passager non guidé.

Le logiciel d'analyse statistique XLSTAT a été utilisé pour les analyses.

7.5.2. Résultats et discussions

La Fig. VII-5 représente la moyenne des angles de roulis mesurés au cours des phases de conduite pour le passager non guidé (courbe en violet) et sa comparaison avec la plate-forme hexapode (courbe en bleu), le véhicule (courbe en rouge) et l'orientation de la courbe de guidage (courbe verte). Selon la Fig. VII-5, nous pouvons voir que le niveau du roulis de la tête du passager non guidé (mesures de niveau vestibulaire) passe à 14 degrés (virage à gauche à 100. secondes) à -32 degrés (virage à droite à 300 secondes). Ce qui signifie un changement brusque de la position de la tête due aux virages. Il en est de même à de 280 et 690 secondes voir Fig. VII-3.

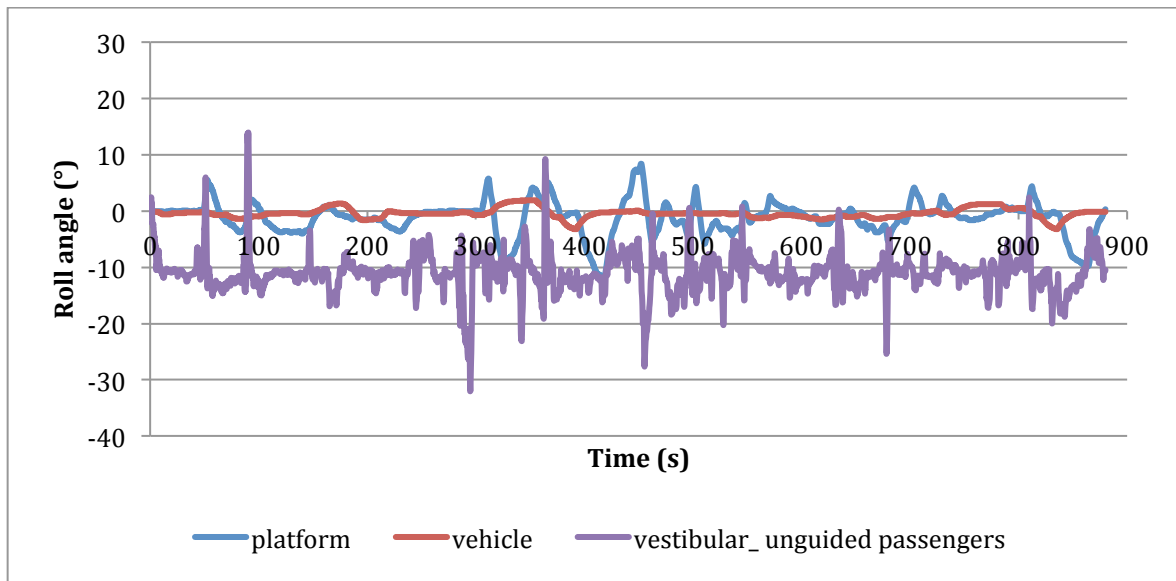


Fig. VII-5 : Angles de roulis pour un passager non-guidé

La Fig. VII-6 représente la moyenne des angles de roulis mesurés au cours des phases de conduite pour le passager guidé (courbe en violet) et sa comparaison avec la plate-forme hexapode (courbe en bleu), le véhicule (courbe en rouge) et l'orientation de la courbe de guidage (courbe verte). Selon la Fig. VII-6, on voit que le niveau des angles de roulis de la tête du passager guidé changent jusqu'à 20 degrés (virage à gauche à 100. secondes) et jusqu'à -32 degrés (virage à droite à 480 secondes). En dehors de ces pics, la courbe représentant le roulis de la tête du passager guidé est plus réaliste par rapport à la configuration de la route.

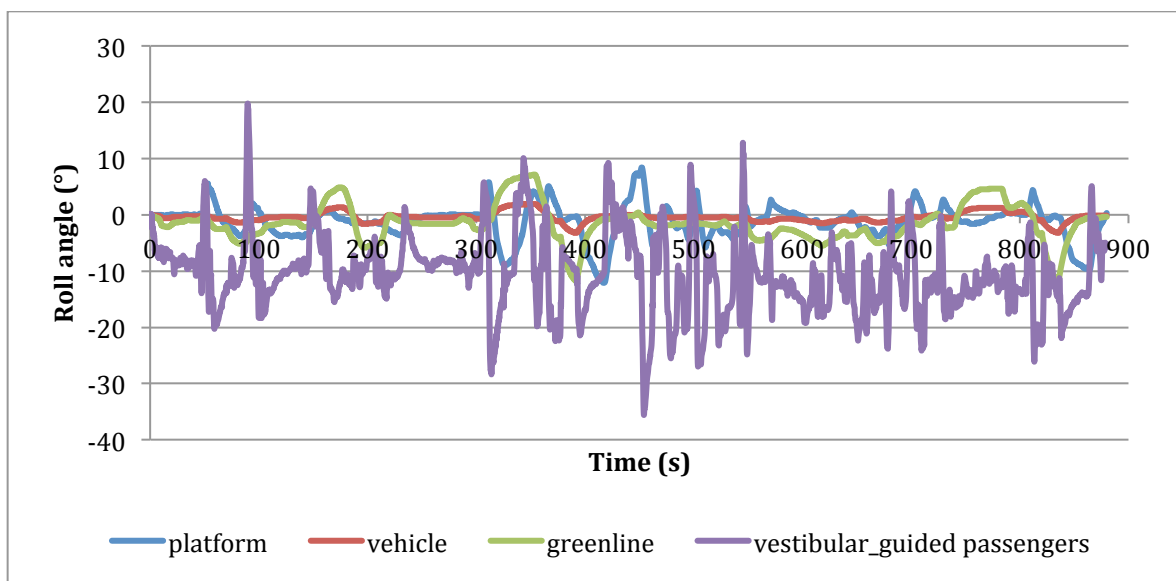


Fig. VII-6 : Angles de roulis pour un passager guidé

La Fig. VII-7 montre la moyenne des angles de roulis mesurés au cours des phases de conduite pour le conducteur (en courbe mauve) et sa comparaison avec la plate-forme hexapode (en courbe bleue), véhicule (courbe rouge) l'orientation de la courbe de guidage (courbe verte). Selon la Fig. VII-7, on voit que le niveau de roulis de la tête du conducteur (mesures de niveau vestibulaires) passe de -31 degrés (virage à droite à 300 secondes) à 18 degrés (effet de contrecoup 305 secondes). Même si il y a un changement à 305 secondes, en dehors de cela; le conducteur réagit avec une plus grande coordination. Sa tête oscille dans la gamme de -10 et -15 degrés. (voir le trajet Fig. VII-3).

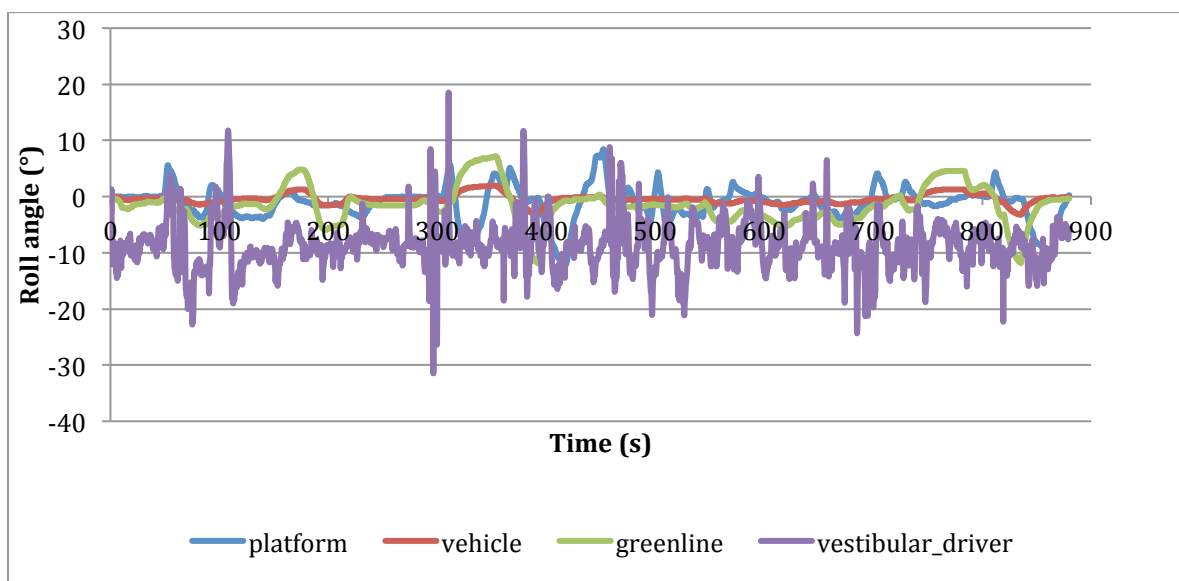


Fig. VII-7 : Angles de roulis pour un conducteur

Le Tableau VII-2 montre les statistiques générales telles que les valeurs minimales, maximales, moyenne et écart-type (SD) d'angles de roulis (en degrés) du conducteur, du passager guidé et du passager non-guidé. Il illustre également les valeurs minimum, maximum, moyenne et écart-type (SD) de l'angle de roulis (plate-forme de mouvement angle de roulis en degrés) de la plate-forme de mouvement hexapode. Grâce au Tableau VII-2, les angles de roulis en degré peuvent être ordonnés $-10,270 \pm 6.828$, $-10,823 \pm 3.895$ et $-8,701 \pm 4.385$ pour les passagers guidés, des passagers non guidés et des conducteurs. L'angle de roulis pour la plate-forme hexapode est de -1.119 ± 2.938 °.

Variable	Minimum	Maximum	Moyenne±SD
Angle de roulis passagers guidés (°)	-35.579	19.834	-10.270±6.828
Angles de roulis pour les passagers non-guidés (°)	-32.031	13.960	-10.823±3.895
Angles de roulis pour les conducteurs (°)	-31.453	18.538	-8.701±4.385
Angles de roulis pour la plateforme (°)	-12.136	8.414	-1.119±2.938

Tableau VII-2 : Statistiques pour les angles de roulis

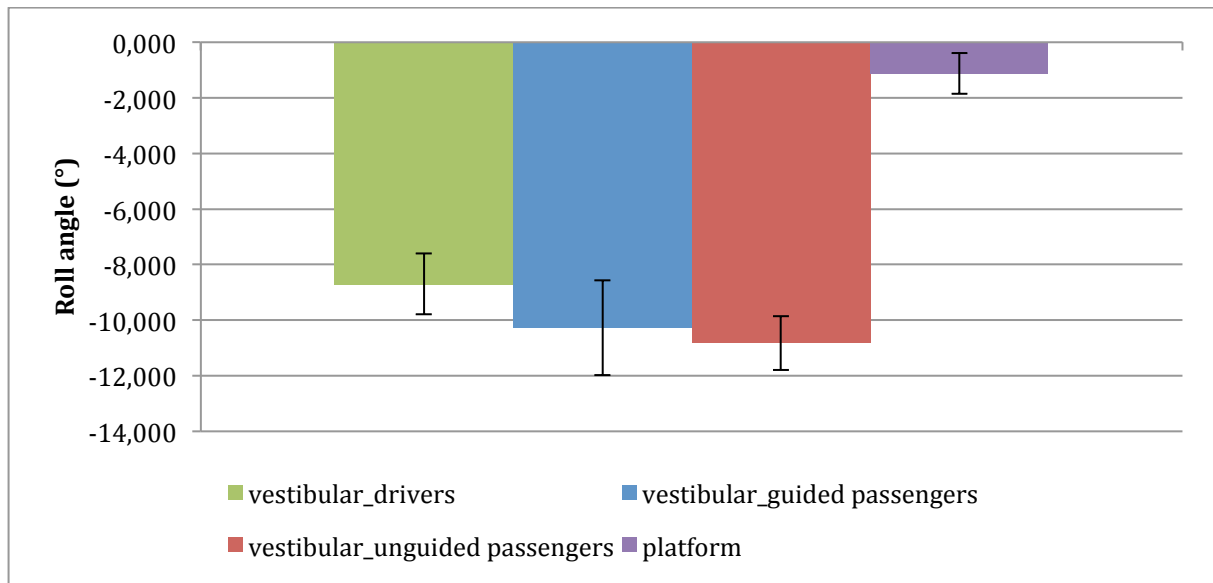


Fig. VII-8 : Angles de roulis et écart type

La Fig. VII-8 illustre la mesure d'angle de roulis au niveau vestibulaire et au niveau de la plate-forme lors des phases de conduite. En raison de ce chiffre et le test bilatéral de Friedman, nous pouvons voir que les angles de roulis pour les passagers guidés, les passagers non guidés et les conducteurs sont significativement plus élevés que l'angle de roulis de la plate-forme de mouvement respectivement $Q(16), p < 0,0001, \text{diff} = 1.797 > 0.0854$, $Q(16), p < 0,0001, \text{diff} = 1.986 > 0.0854$ et $Q(16), p < 0,0001, \text{diff} = 1.493 > 0.0854$.

	Angles de roulis passagers guidés (°)	Angles de roulis passagers non guidés (°)	Angles de roulis conducteurs (°)
Angles de roulis de la plateforme (°)	1.797	1.986	1.493
Angles de roulis pour les passagers non guidés	-0,189	0	-0,493

Critical difference: 0.0854

Tableau VII-3 : Table des différences bilatérales pour des angles de roulis

	Angles de roulis passagers guidés (°)	Angles de roulis passagers non guidés (°)	Angles de roulis conducteurs (°)
Angles de roulis de la plateforme (°)	< 0.0001	< 0.0001	< 0.0001
Angles de roulis pour les passagers non guidés	< 0.0001	1	< 0.0001

Bonferroni corrected significance level: 0.0083

Tableau VII-4 : p-valeurs des angles de roulis

D'après les Tableau VII-3 et Tableau VII-4, on peut résumer que l'angle de roulis pour les passagers non guidés est significativement plus élevé que pour les passagers guidés et les conducteurs respectivement $Q(16), p < 0,0001$, diff (en valeur absolue) = $|-0,189| > 0,0854$ et $Q(16), p < 0,0001$, diff (en valeur absolue) = $|-0,493| > 0,0854$.

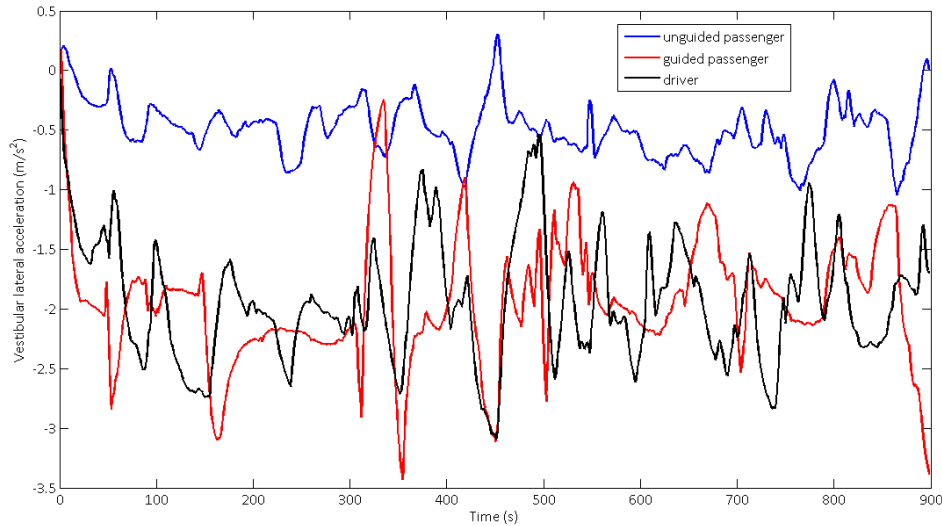


Fig. VII-9 : Niveau des accélérations latérales vestibulaires pour 1 sujet

Auto-évaluation pour les passagers non guidés et les conducteurs (moy/écart type)		Valeurs p de test t bilatéral	Auto-évaluation pour les passagers guidés et les conducteurs (moy/écart type)		Valeurs p de test t bilatéral
a_{y_vest} <i>passagers non-guidés</i>	-0.4198±0.2750	p=0.0000<0.05	a_{y_vest} <i>passagers guidés</i>	-1.8190±0.6707	p=0.2833>0.05
a_{y_vest} <i>conducteurs</i>	-1.7898±0.6340		a_{y_vest} <i>conducteurs</i>	-1.7898±0.6340	

Tableau VII-5 : Comparaison des accélérations vestibulaires latérales

La Tableau VII-5 et Fig. VII-9 résument les différences d'engagement en simulation de conduite. Selon les tests-t bilatéraux, les passagers non guidés sont plus exposés au niveau

accélération latérale de la tête que les conducteurs ($t(16), p = 0,0000 < 0,05$). En revanche, les passagers guidés et les conducteurs qui sont soumis au même scénario de conduite n'a révélé aucune différence significative en termes d'accélération vestibulaire latérale ($t(16), p=0,2833 > 0,05$). Cela nous montre que la présence des passagers guidés est augmentée et s'approche de celle des conducteurs.

La Fig. VII-10 montre les moyennes cumulées pendant le temps d'exposition des valeurs VDV des accélérations vestibulaires (pour les sujets) et de la plateforme en fonction des conditions d'engagement du sujet. On voit que les réponses dynamiques pour les conducteurs et les passagers guidés convergent vers la réponse de la plateforme alors que la réponse dynamique pour les passagers non guidés a un écart important avec la réponse dynamique de la plateforme.

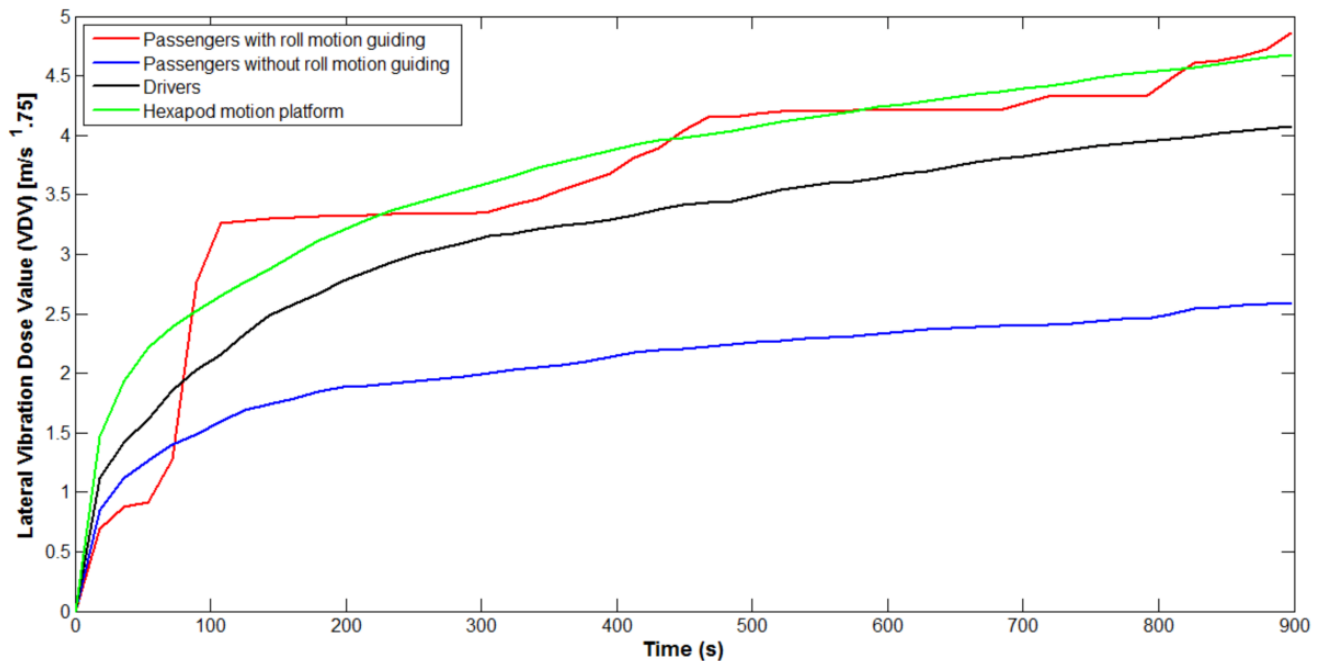


Fig. VII-10 : Moyenne cumulée des VDV au niveau vestibulaire

Le Tableau VII-6 donne les statistiques générales sur les valeurs VDV comme les valeurs minimales, maximales, déviations standards pour les sujets conducteurs, passagers guidés et passagers non-guidés ainsi que pour la plateforme. Selon ce tableau, les valeurs VDV en $m/s^{1.75}$ peuvent être ordonnées par les valeurs : 3.612 ± 1.077 , 2.065 ± 0.489 et 3.175 ± 0.852 pour les passagers guidés, passagers non guidés et conducteurs. Les valeurs VDV pour la plateforme sont dans les valeurs suivantes : $3.697 \pm 0.931 m/s^{1.75}$. Les valeurs montrent bien que les valeurs VDV des conducteurs et passagers guidés convergent vers les valeurs de la

plateforme alors que le conflit visuo-vestibulaire est plus important pour les passagers non-guidés.

Variable	Minimum	Maximum	Mean±SD
Passengers with guiding VDV_{y_vest} ($m/s^{1.75}$)	0.000	4.855	3.612±1.077
Passengers without guiding VDV_{y_vest} ($m/s^{1.75}$)	0.000	2.587	2.065±0.489
Drivers VDV_{y_vest} ($m/s^{1.75}$)	0.000	4.074	3.175±0.852
Platform VDV_{y_plat} ($m/s^{1.75}$)	0.000	4.675	3.697±0.931

Tableau VII-6 : Statistiques pour VDV latéral

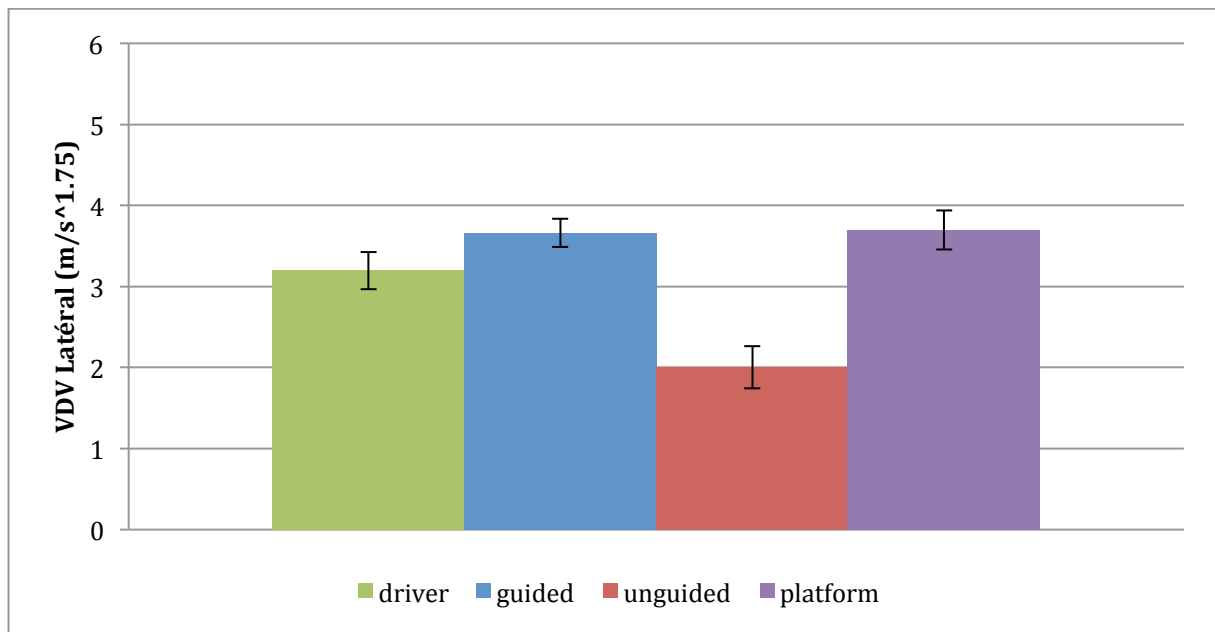


Fig. VII-11 : Moyenne de la VDV latérale (intervalle de confiance de 95%)

La Fig. VII-11 illustre les mesures moyennes VDV des accélérations latérales pour les sujets et la plateforme durant l'expérience. Les tests U de Mann-Whitney confirment nos deux hypothèses. En effet, les tests statistiques montrent que les valeurs VDV sont proches pour les sujets conducteurs et passagers guidés alors que les différences sont significatives pour les

valeurs VDV concernant les sujets conducteurs et les sujets passagers d'une part et les sujets passagers guidés et les sujets passagers non guidés d'autre part.

	<i>Passagers guidés</i> $VDV_{y_vest} (m/s^{1.75})$	<i>Passagers non-guidés</i> $VDV_{y_vest} (m/s^{1.75})$	<i>Conducteurs</i> $VDV_{y_vest} (m/s^{1.75})$
<i>Plateforme</i> $VDV_y (m/s^{1.75})$	0.314	2.451	1.471

Différence critique : 0.6745

Tableau VII-7 : Différences par paire pour VDV latéral

	<i>Passagers guidés</i> $VDV_{y_vest} (m/s^{1.75})$	<i>Passagers non-guidés</i> $VDV_{y_vest} (m/s^{1.75})$	<i>Conducteurs</i> $VDV_{y_vest} (m/s^{1.75})$
<i>Platform</i> $VDV_y (m/s^{1.75})$	0.220	< 0.0001	< 0.0001

Niveau de correction Bonferroni : 0.0083

Tableau VII-8 : p-valeurs pour VDV latéral

Le test de Friedman pour les accélérations latérales VDV montre qu'il n'y a pas de différence significative entre les passagers guidés et la plateforme (Q(16), $p=0.220 > 0.0083$, $\text{diff}=0.314 < 0.6745$). Il existe par contre une différence significative entre les passagers non guidés et la plateforme (Q (16), $p < 0.0001$, $\text{diff}=2.451 > 0.6745$) voir les Tableau VII-7 et Tableau VII-8.

7.6. Conclusion et perspectives

L'influence du niveau d'engagement du sujet dans la conduite (conducteur, passager, passager guidé) sur l'exposition aux vibrations au niveau de la tête (système vestibulaire) a été analysée au regard du mouvement inertiel de la plateforme dynamique.

Le mouvement de la plateforme du simulateur dynamique et les valeurs VDV au niveau de la tête convergent pour le conducteur comparé au passager. Il existe une différence significative entre le conducteur et le passager (U (16) =2286.500 > 1300.500, $p < 0.0001$). Les niveaux de vibrations sont plus proches de la plateforme pour le passager guidé que le passager non guidé également (U (16) =2355.500 > 1300.500, $p < 0.0001$).

Après avoir complété le test de Friedman pour les valeurs de VDV, aucune différence significative n'a été trouvée entre le passager guidé et la plateforme (Q(16), $p=0.220 > 0.0083$,

diff=0.314<0.6745). C'est un indicateur important de la corrélation entre les dynamiques au niveau de la tête et de la plateforme. Cela induit également un relativement faible conflit visuo-vestibulaire. Cependant, il est constaté une différence significative entre le passager non guidé et la plateforme en terme de VDV_y (Q (16), $p<0.0001$, diff=2.451>0.6745). La différence entre le conducteur et la plateforme reste faible mais significative (Q (16), $p<0.0001$, diff=1.471>0.6745).

Ces données montrent que l'engagement du sujet dans la conduite a une influence significative sur son mal de simulateur.

Les travaux futurs envisagés concernent l'intégration de guide multi-sensoriel (en particulier haptique) pour améliorer l'engagement du passager dans la conduite. Il serait également intéressant d'évaluer le système de guidage en situation de conduite réelle (en utilisant un système de retours proprioceptif).

Chapitre VIII

Influence du contrôle de la plateforme

8.1. Introduction

Cette étude a pour objectif de comparer le fonctionnement des algorithmes de pilotage de la plateforme en mode boucle ouverte ou boucle fermée. Ainsi, l'algorithme MRAC et l'algorithme LQR seront comparés à l'algorithme de pilotage classique.

8.2. Conditions expérimentales

Les conditions expérimentales mises en œuvre dans ces deux tests consistent à analyser le comportement du conducteur lors du franchissement d'une chicane. La Fig. VIII-1 montre la trajectoire de type chicane que nous avons utilisée dans ce protocole expérimental. La vitesse du véhicule au cours des expériences a été choisie constante et égale à 60 km/h. Le même scénario a été utilisé avec l'algorithme classique, l'algorithme MRAC et l'algorithme LQR (définis dans les chapitres précédents) afin de comparer les interactions biomécaniques des participants avec la plateforme.

La mesure de la dynamique au niveau vestibulaire des participants sera réalisée grâce au capteur Xsens (implémentation identique aux expérimentations précédentes). La mesure de la dynamique au niveau du véhicule provient du logiciel de pilotage de la plateforme (logiciel SCANeR Studio de la société Oktal). La mesure par électromyographie sera aussi mise en œuvre pour cette étude ; cette mesure permet d'avoir des informations objectives sur le stress et le tension du sujet.

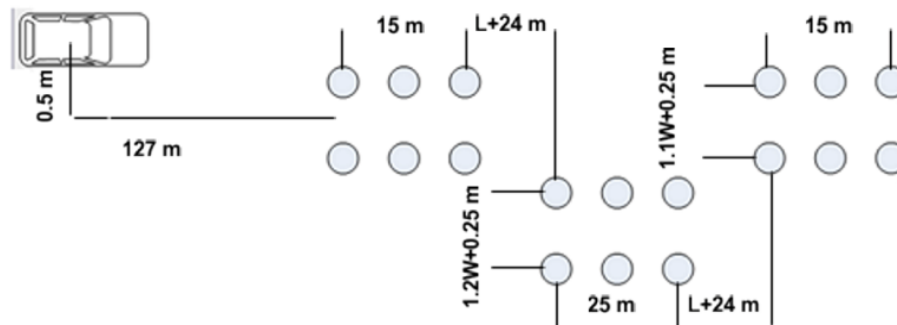


Fig. VIII-1 : Trajectoire exécutée pour le test.

8.3. Comparaison entre l'algorithme classique et l'algorithme MRAC

8.3.1. Sujets

Vingt-six participants en bonne santé ont participé aux expériences (4 femmes, 22 hommes) avec un âge moyen de 28.85 ± 5.8 ans et un permis de conduire de maintien avec une expérience moyenne de 9.73 ± 6.58 années.

8.3.2. L'analyse des données

Le taux d'échantillonnage pour l'enregistrement des données du capteur est fixé à 20 Hz. Pour la calibration des données d'acquisition, l'alignement des axes a été choisi de telle sorte qu'il soit simple à mettre en œuvre. Ceci a pour avantage de synchroniser l'ensemble des systèmes de mesure en un seul clic.

Pour cette étude, nous avons choisi d'utiliser le paramètre EMG RMS (root mean square: qui est le produit de la dynamique longitudinale, latérale et verticale liée à la puissance dissipée). Ce paramètre est exprimé en V^2/Hz dans le domaine temporel, afin d'étudier leurs associations avec RVT (roll velocity perception threshold – en $\%$, indicateur de conflit sensoriel en dynamique).

Pour le calibrage de l'électromyographie, un gain de 1000 a été utilisé. La Fig. VIII-2 illustre l'acquisition de données au cours des expériences, les activités électriques des muscles ont été enregistrées par un procédé non invasif EMG de surface à travers deux canaux analogiques. Les signaux ont été recueillis à travers un filtre passe bande de 10 - 500 Hz.

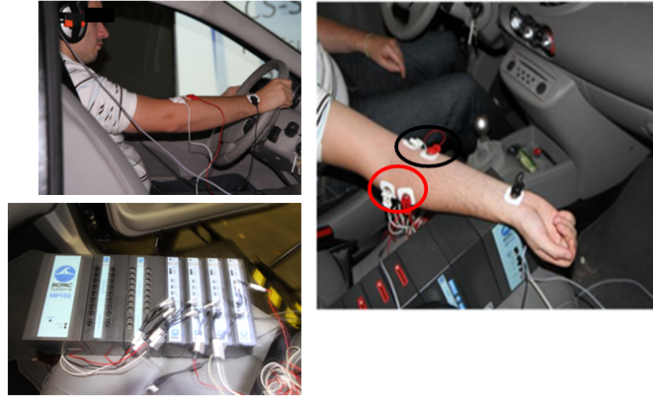


Fig. VIII-2 : Mouvement de la tête et l'analyse EMG

Les électrodes cerclées en noir dans la Fig. VIII-2 sont connectées au muscle fléchisseur radial du carpe et les électrodes cerclées en rouge sont reliées au muscle long supinateur. Nous avons mesuré et enregistré les variations d'activités électriques de ces deux muscles.

8.3.3. Résultats et discussion

La Fig. VIII-3 correspond aux angles des roues appliqués par le conducteur lors du franchissement de la chicane en vue de l'analyse EMG-RMS. La Fig. VIII-4 représente une mesure en temps réel de l'EMG RMS pour un des vingt-six sujets qui a participé aux expériences et pour chacun des algorithmes (classique et MRAC).

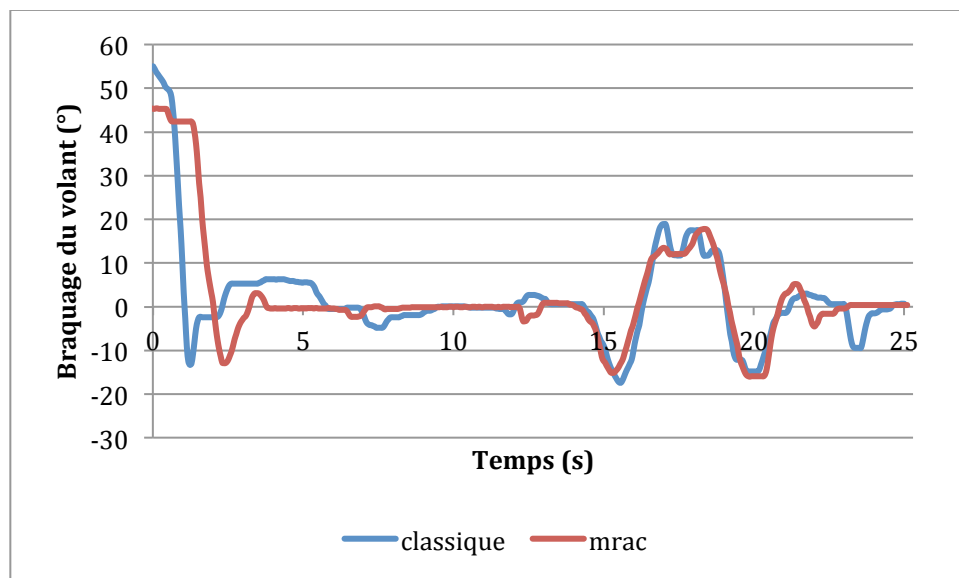


Fig. VIII-3 : Angles de volant (°)

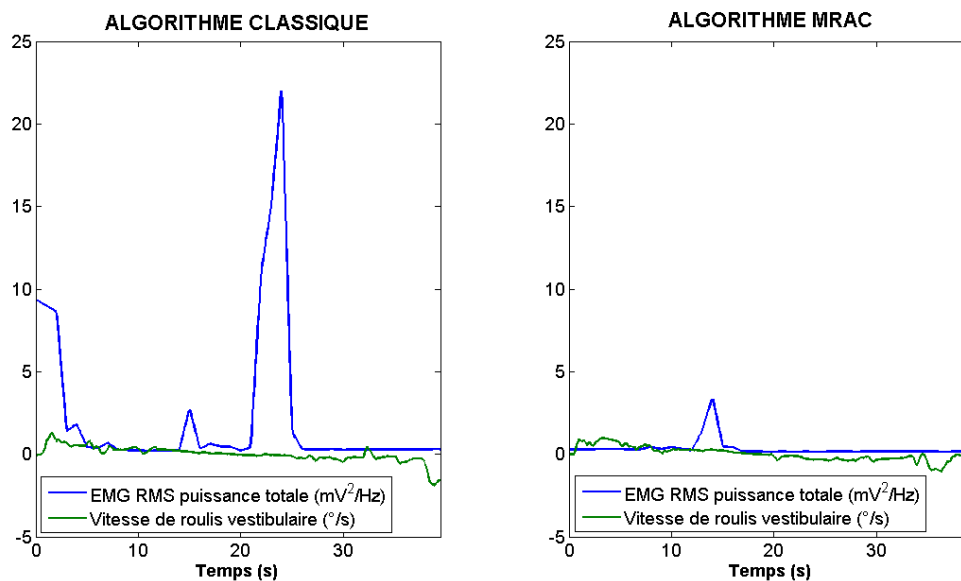


Fig. VIII-4 : EMG RMS total puissance (mV²/Hz) et vestibulaire vitesse roulis (°/s)

L'analyse des figures montre qu'à l'approche de 5 secondes le sujet arrive au premier virage à droite de la chicane. La Fig. VIII-4 quant à elle montre que l'énergie dissipée est plus faible avec l'algorithme MRAC qu'avec l'algorithme classique.

Entre 8 et 11 secondes, l'algorithme classique montre des caractéristiques de survirage lors de l'alignement du véhicule après le premier virage. De ce fait, la puissance EMG RMS du muscle du bras augmente à la suite de ce survirage. La région de 17 à 24 s. correspond à la ligne droite avant le dernier virage de sortie de la chicane. A environ 17 s., la vitesse vestibulaire montre un déplacement en contradiction avec la dynamique du véhicule (cf. Fig. VIII-3). Il augmente dans la direction opposée (virage à gauche) pendant que le véhicule tourne à droite (survirage au virage à droite). De même à environ 22 s., la vitesse vestibulaire montre un mouvement contradictoire avec la dynamique du véhicule. Il augmente dans la direction opposée (à gauche), tandis que le véhicule tourne à gauche (survirage au virage à gauche). Cependant, cette incidence de survirage n'a pas d'influence sur le signal vestibulaire (cf. Fig. VIII-5). Cela prouve également que le conflit neuromusculaire/dynamique du véhicule influe sur le conflit visuo-vestibulaire, mais la logique inverse n'est pas toujours valable.

On peut constater que l'écart entre la puissance EMG RMS et la vitesse de déploiement au niveau vestibulaire a été réduit en utilisant l'algorithme MRAC. Cela prouve aussi moins de signaux contradictoires sur le muscle du bras et le système vestibulaire, en d'autres termes moins de perte d'adhérence du véhicule ou de plus d'agilité du conducteur dans les dynamiques latérales.

Le Tableau VIII-1 résume la corrélation entre la puissance musculaire dissipée par le bras (muscle fléchisseur radial du carpe) avec et le déplacement latéral du centre de gravité (CG) du véhicule avec l'algorithme classique d'une part et la vitesse de roulis au niveau de la tête avec le déplacement latéral du CG du véhicule d'autre part avec l'algorithme classique. D'après le tableau, on voit qu'il y a une corrélation significative entre la vitesse de roulis du véhicule vestibulaire et le déplacement latéral du CG du véhicule sauf pour le sujet 1 et 13. Par ailleurs, seul le sujet 6 a donné une corrélation positive entre la vitesse de roulis niveau vestibulaire et la puissance du muscle du bras. En dehors de ce sujet, tous les sujets ont montré une corrélation négative ce qui représente en fait une augmentation du niveau de conflit visuo-vestibulaire. On peut conclure qu'il y a des corrélations significatives entre la puissance musculaire des bras et les déplacements latéraux du CG pour les treize sujets ayant subi les tests avec l'algorithme classique. Par ailleurs, une corrélation négative entre la puissance des muscles du bras et les déplacements latéraux du CG est présente pour les sujets 1,3,7,8 et 13 pour l'algorithme classique, tandis qu'une corrélation positive a été obtenue pour les autres sujets. Ces corrélations nous montrent que 8 sujets ont subi une perte de contrôle du véhicule.

	Algorithme classique <i>Corrélation de la vitesse du roulis au niveau vestibulaire et du déplacement latéral du centre de gravité</i>		Algorithme classique <i>Corrélation de la puissance dissipée par les muscles et du déplacement latéral du centre de gravité</i>	
	<i>r</i>	<i>p</i>	<i>r</i>	<i>p</i>
<i>Subject 1</i>	-0.0252	0.6025	-0.7730	0.0000***
<i>Subject 2</i>	-0.5375	0.0000***	0.4004	0.0000***
<i>Subject 3</i>	-0.5815	0.0000***	-0.1068	0.0244*
<i>Subject 4</i>	-0.2872	0.0000***	0.1714	0.0003***
<i>Subject 5</i>	-0.6761	0.0000***	0.5870	0.0000***
<i>Subject 6</i>	0.3348	0.0000***	0.4380	0.0000***
<i>Subject 7</i>	-0.4486	0.0000***	-0.2023	0.0000***
<i>Subject 8</i>	-0.3638	0.0000***	-0.1652	0.0007***
<i>Subject 9</i>	-0.1770	0.0003***	0.1994	0.0000***
<i>Subject 10</i>	-0.6624	0.0000***	0.1413	0.0051**
<i>Subject 11</i>	-0.6265	0.0000***	0.1931	0.0001***
<i>Subject 12</i>	-0.7544	0.0000***	0.2197	0.0000***
<i>Subject 13</i>	-0.0691	0.1448	-0.1083	0.0220*

Tableau VIII-1 : Interactions dynamique vestibulaire/muscle du bras – algorithme classique

	Algorithme MRAC <i>Corrélation de la vitesse du roulis au niveau vestibulaire et du déplacement latéral du centre de gravité</i>		Algorithme MRAC <i>Corrélation de la puissance dissipée par les muscles et du déplacement latéral du centre de gravité</i>	
	<i>r</i>	<i>p</i>	<i>r</i>	<i>p</i>
<i>Sujet 1</i>	-0.2241	0.0000***	0.3925	0.0000***
<i>Sujet 2</i>	-0.6283	0.0000***	0.3586	0.0000***
<i>Sujet 3</i>	0.4826	0.0000***	0.0896	0.0618
<i>Sujet 4</i>	-0.5073	0.0000***	-0.3667	0.0000***
<i>Sujet 5</i>	-0.4519	0.0000***	0.0969	0.0182*
<i>Sujet 6</i>	0.1222	0.0155*	0.4443	0.0000***
<i>Sujet 7</i>	-0.1934	0.0001***	0.1651	0.0008***
<i>Sujet 8</i>	0.0368	0.4506	0.1650	0.0007***
<i>Sujet 9</i>	-0.1310	0.0063**	0.0667	0.1658
<i>Sujet 10</i>	-0.5408	0.0000***	0.1386	0.0064**
<i>Sujet 11</i>	-0.1222	0.0198*	0.0036	0.9456
<i>Sujet 12</i>	-0.6599	0.0000***	0.3367	0.0000***
<i>Sujet 13</i>	-0.4904	0.0000***	0.3724	0.0000***

Tableau VIII-2 : Interactions dynamique vestibulaire/muscle du bras – algorithme MRAC

Le Tableau VIII-2 donne la corrélation entre la puissance musculaire dissipée par le bras (muscle fléchisseur radial du carpe) avec le déplacement latéral du CG et la vitesse du roulis au niveau de la tête avec le déplacement latéral du CG pour l'algorithme MRAC. D'après ce tableau, on peut voir qu'il existe une corrélation significative entre la vitesse de roulis au niveau vestibulaire et le déplacement latéral du CG, sauf pour le sujet 8. Par ailleurs, les sujets 3, 6 et 8 ont obtenu une corrélation positive entre la vitesse de roulis au niveau vestibulaire et la puissance dissipée par les muscles du bras. Les autres sujets ont obtenu une corrélation négative.

Toujours selon le Tableau VIII-2, on peut conclure qu'il existe des corrélations significatives entre les puissances musculaires des bras et les déplacements latéraux du CG en dehors des sujets 3, 9 et 11 avec l'algorithme MRAC. Une corrélation négative entre la puissance dissipée par les muscles du bras et les déplacements latéraux du CG a été obtenue seulement pour le sujet 4, tandis qu'une corrélation positive a été obtenue pour les autres sujets. Ces corrélations nous montrent qu'un sujet sur les 13 (sujet 4) a conduit le véhicule

avec une tendance à perdre de contrôle, en ne prenant en compte que la corrélation entre la puissance dissipée par les muscles du bras et de la vitesse niveau vestibulaire.

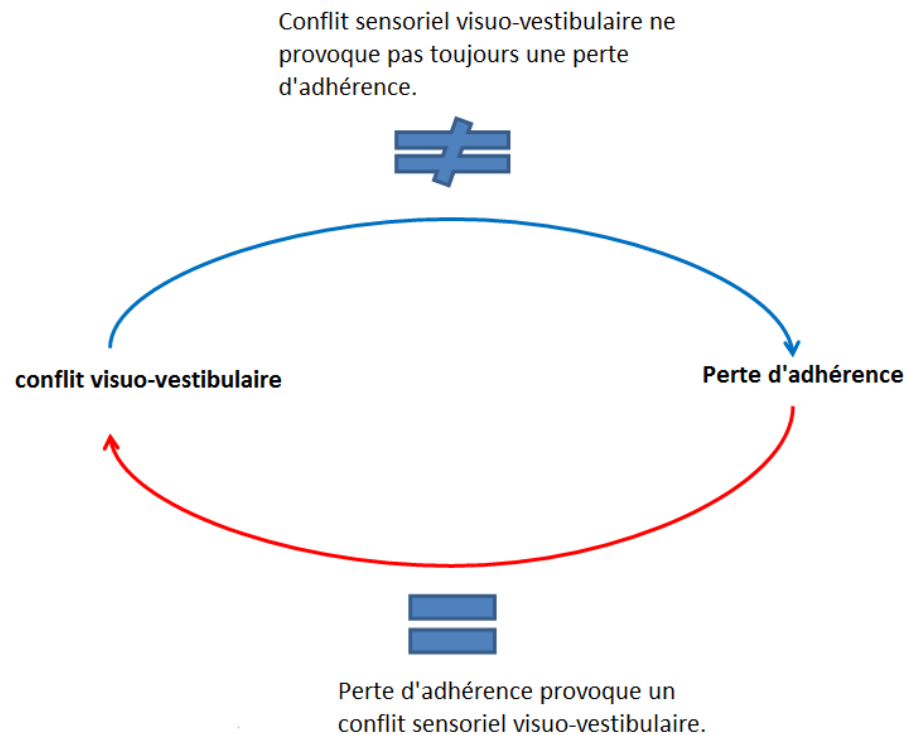


Fig. VIII-5 : Interaction dynamique de véhicule/Système Vestibulaire/Système neuromusculaire

D'après le Tableau VIII-1, le Tableau VIII-2 et la Fig. VIII-5, on voit que la perte d'adhérence du véhicule provoque un conflit visuo-vestibulaire (si la corrélation de la vitesse de roulis du centre CG avec le déplacement latéral est négative) mais un conflit visuo-vestibulaire n'est pas toujours suivie d'une perte d'adhérence (si la corrélation de l'EMG RMS totale avec le déplacement du CG est négative).

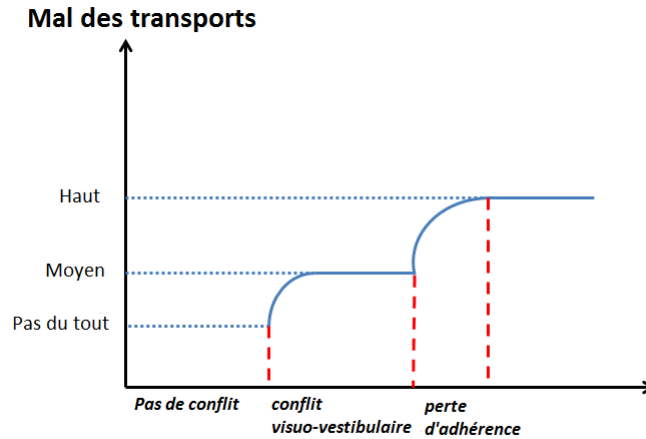


Fig. VIII-6 : Les relations de conflit multi sensoriel avec le mal des transports

La Fig. VIII-6 résume nos conclusions sur les relations multi-sensorielles (vestibulaires, neuromusculaires, des véhicules (visuelle)) pour le mal des transports.

	Algorithme classique	Algorithme MRAC	changement en perte d'adhérence (%) du algorithme classique au algorithme MRAC
	<i>Zone de déplacement latéral (m.s)</i>	<i>Zone de déplacement latéral (m.s)</i>	
Sujet 1	3.2166x10⁴	1.6569x10⁴	48.4 % de réduction
Sujet 2	1.5717x10⁴	1.4588x10⁴	7.1 % de réduction
Sujet 3	2.1987x10⁴	1.7697x10⁴	19.5 % de réduction
Sujet 4	1.6291x10⁴	1.5022x10⁴	7.8 % de réduction
Sujet 5	1.3508x10⁴	1.2317x10⁴	8.8 % de réduction
Sujet 6	1.0828x10 ⁴	1.3160x10 ⁴	21.5 % d'augmentation
Sujet 7	1.3432x10 ⁴	1.3630x10 ⁴	1.4 % d'augmentation
Sujet 8	1.4279x10 ⁴	1.4479x10 ⁴	1.4 % d'augmentation
Sujet 9	1.3804x10 ⁴	1.5422x10 ⁴	11.7 % d'augmentation
Sujet 10	1.2864x10⁴	1.2560x10⁴	2.36 % de réduction
Sujet 11	1.4529x10⁴	1.1474x10⁴	21 % de réduction
Sujet 12	1.2642x10⁴	1.1966x10⁴	5.35 % de réduction
Sujet 13	1.7150x10⁴	1.5708x10⁴	8.4 % de réduction

Tableau VIII-3 : Zone de déplacement latéral pour les deux algorithmes

Le Tableau VIII-3 illustre la zone de déplacement latéral (équation (70) en ms) [eCDicaoN, MAT09a, MAT09b] (voir Fig. VIII-7) dans le cadre du déplacement latéral du CG (Y_{CG}) à partir des séries chronologiques où t est le temps.



Fig. VIII-7 : Zone de déplacement latéral

Selon le Tableau VIII-3, on peut résumer qu'en dehors des sujets 6, 7, 8 et 9 la perte d'adhérence a diminué. En revanche pour les autres sujets, le niveau de contrôle du véhicule des conducteurs a augmenté avec l'utilisation de l'algorithme MRAC.

8.3.4. Interaction vestibulaire et neuromusculaire

Après avoir terminé l'évaluation des données individu par individu, nous allons procéder à l'analyse globale des données pour chaque algorithme.

L'analyse globale des données a été effectuée en utilisant une analyse ANOVA à deux voies (analyse de la variance) et d'une corrélation de Pearson avec un alpha de 0.05. Afin de réaliser l'analyse statistique, nous avons pris :

- La vitesse de roulis au niveau vestibulaire,
- L'EMG RMS

Double sens ANOVA (Two-way ANOVA)

L'étude ANOVA pour l'algorithme classique montre qu'il n'y a aucune preuve d'un effet d'interaction significatif ($F = 3.74$, $p = 0.085$). Nous pouvons donc conclure qu'il n'existe pas d'interaction entre la puissance moyenne EMG RMS et le seuil de vitesse de roulis. Le test pour le seuil de perception de la vitesse vestibulaire de roulis ($F = 45.17$, $p < 0.0001$) montre un effet de seuil de perception de vitesse de roulis significatif au niveau de la puissance maximale de l'EMG RMS. Enfin, le test de l'effet principal pour la puissance moyenne l'EMG RMS ($F = 1360.89$, $p < 10^{-9}$) nous dit qu'il existe une preuve pour conclure que la puissance moyenne de l'EMG RMS a un effet significatif sur le niveau de puissance maximale de l'EMG RMS..

Par l'analyse de l'ANOVA à deux voies pour l'algorithme MRAC, nous voyons qu'il n'y a aucune preuve d'un effet d'interaction significatif ($F = 0.43$, $p = 0.528$). Nous ne pouvons donc pas conclure qu'il existe une interaction entre la puissance moyenne EMG RMS et le seuil de vitesse de roulis. Le test pour le seuil de perception de la vitesse vestibulaire de roulis ($F = 9.22$, $p = 0.01412 < 0.005$) indique un effet significatif du seuil de perception de vitesse de roulis sur le niveau EMG RMS de puissance maximale. Enfin, le test de l'effet principal de l'EMG RMS puissance moyenne ($F = 232.44$, $p < 10^{-6}$) nous dit qu'il y a une preuve pour conclure qu'il y a un effet significatif d'EMG RMS sur le niveau de puissance EMG RMS de puissance maximale.

- Corrélation de Pearson

Après avoir analysé les rapports de seuils de vitesse de roulis avec EMG RMS et la puissance maximale totale, nous avons constaté que si nous utilisons l'algorithme MRAC, il montre une agilité plus grande qu'avec l'algorithme classique.

Nous avons décrit la vitesse de roulis au niveau vestibulaire (RVT) [BHB89, Ben90, GM61, Gue64, Gue74]. Le logiciel MATLAB / Simulink a été utilisé pour traiter les données. Les données ont été conditionnées avec un filtre passe bas du 1er ordre Butterworth à 5 Hz.

Les valeurs EMG RMS (mV) sont calculées sur la base d'une autonomie totale de mouvement pendant la phase de conduite des expériences de simulation [emg13, FLZ+12, BIO11], en utilisant l'équation suivante (71):

$$EMG_{RMS} = \sqrt{\frac{1}{T} \int_t^{t+T} EMG^2(t).dt} \quad (71)$$

Pour l'analyse des données dans le domaine fréquentiel, une transformée de Fourier rapide (FFT) a été utilisée pour calculer le spectre des puissances des signaux EMG (mV) [emg13, FLZ+12, BIO11] en utilisant l'équation suivante (72) :

$$PSD(f) = |FFT(x_n)|^2 \quad (72)$$

où X_n est un ensemble de signaux EMG consécutifs, la (PSD) densité spectrale de puissance est en mV^2/Hz [emg13, FLZ+12, BIO11].

Les Fig. VIII-8 et Fig. VIII-9 donnent les résultats obtenus pour les puissances totales maximales et moyennes dissipées par les muscles des bras en fonction du seuil de perception de la vitesse de roulis RVT pour les algorithmes classiques (en bleu) et MRAC (en rouge).

Afin d'évaluer les relations, nous avons utilisé MATLAB.

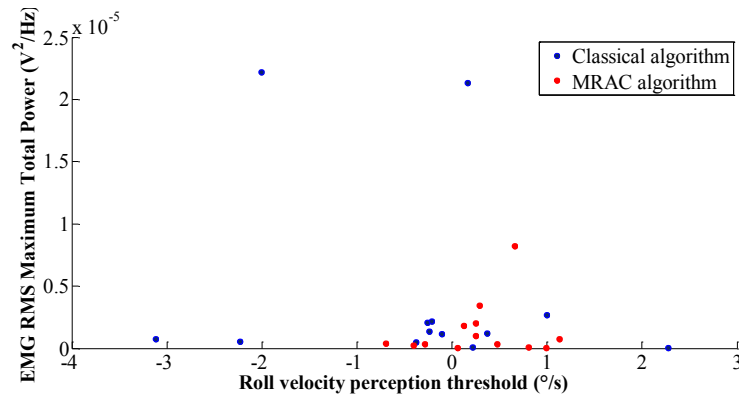


Fig. VIII-8 : Analyse relation de RVT et des EMG RMS puissances maximales totales pour tous les sujets

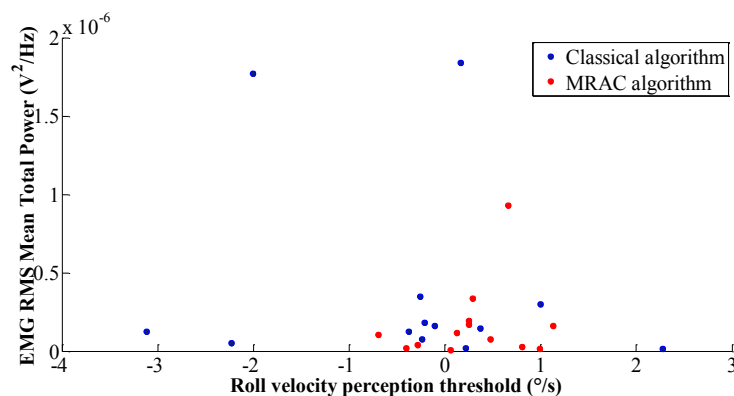


Fig. VIII-9 : Analyse relation de RVT et des EMG RMS puissances moyennes totales pour tous les sujets

La Fig. VIII-10 montre que certains des participants sont parvenus à des valeurs extrêmes (étoiles rouges) avec l'algorithme classique tandis que le RVT pour l'algorithme MRAC a montré l'absence de valeurs extrêmes. En outre, la divergence de RVT pour l'algorithme classique (de $-3.1^\circ / s$ à $2.4^\circ / s$, soit un écart de $5.5^\circ / s$) est supérieur à celui pour le MRAC (de $-0.8^\circ / s$ à $1.1^\circ / s$, soit un écart de $1.9^\circ / s$).

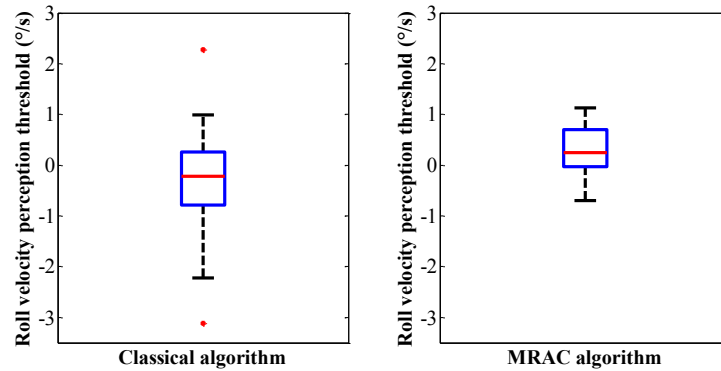


Fig. VIII-10 : Comparaison RVT pour l'algorithme classique et l'algorithme MRAC pour tous les sujets

Selon l'analyse de corrélation de Pearson, il est démontré que le RVT et le EMG RMS sont corrélés positivement ($r = 0.228$, $p = 0.453$) pour l'algorithme MRAC et sont corrélés négativement ($r = -0.167$, $p = 0.586$) pour l'algorithme classique.

Selon l'analyse de corrélation de Pearson, il est démontré que le RVT et la puissance EMG RMS maximum sont corrélés positivement ($r = 0.192$, $p = 0.531$) pour l'algorithme MRAC et ils sont corrélés négativement ($r = -0.178$, $p = 0.560$) pour l'algorithme classique. (Fig. VIII-9 et Fig. VIII-10).

8.3.5. Conclusion

En conclusion, l'algorithme MRAC optimise la condition du simulateur dynamique par rapport à l'algorithme classique, ainsi il en résulte une amélioration de la situation pour les conducteurs en termes d'évitement et de réduction du mal des transports selon la théorie du conflit sensoriel.

Il est également avéré que le conflit dynamique neuromusculaire et dynamique du véhicule a une influence sur le conflit visuo-vestibulaire, mais le conflit visuo-vestibulaire n'influence pas les interactions dynamiques neuromusculaires et dynamique du véhicule.

8.4. Comparaison algorithme classique et LQR

8.4.1. Data Denoising

La transformée en ondelettes est utilisée comme un outil mathématique pour analyser toutes séries chronologiques non-stationnaires, montrant la variabilité temporelle de la densité spectrale de puissance [BGV09].

Les données recueillies à partir du logiciel SCANeRstudio ont été débruitées en utilisant les ondelettes de Daubechies (DBN) ondelettes 1-D de MATLAB. L'entropie de Shannon [CW92, DJ+94, MMOP08] a été choisie pour la suite des calculs.

La méthode de seuillage doux comme seuil de forme fixe (échelle w_n) a été appliquée pour traiter les bruits des signaux d'accélération de roulis par rapport aux niveaux de cabine et plate-forme.

Le seuil global a été ajusté à 80% du seuil maximum. Enfin, le signal débruité est présenté sur la Fig. VIII-11.

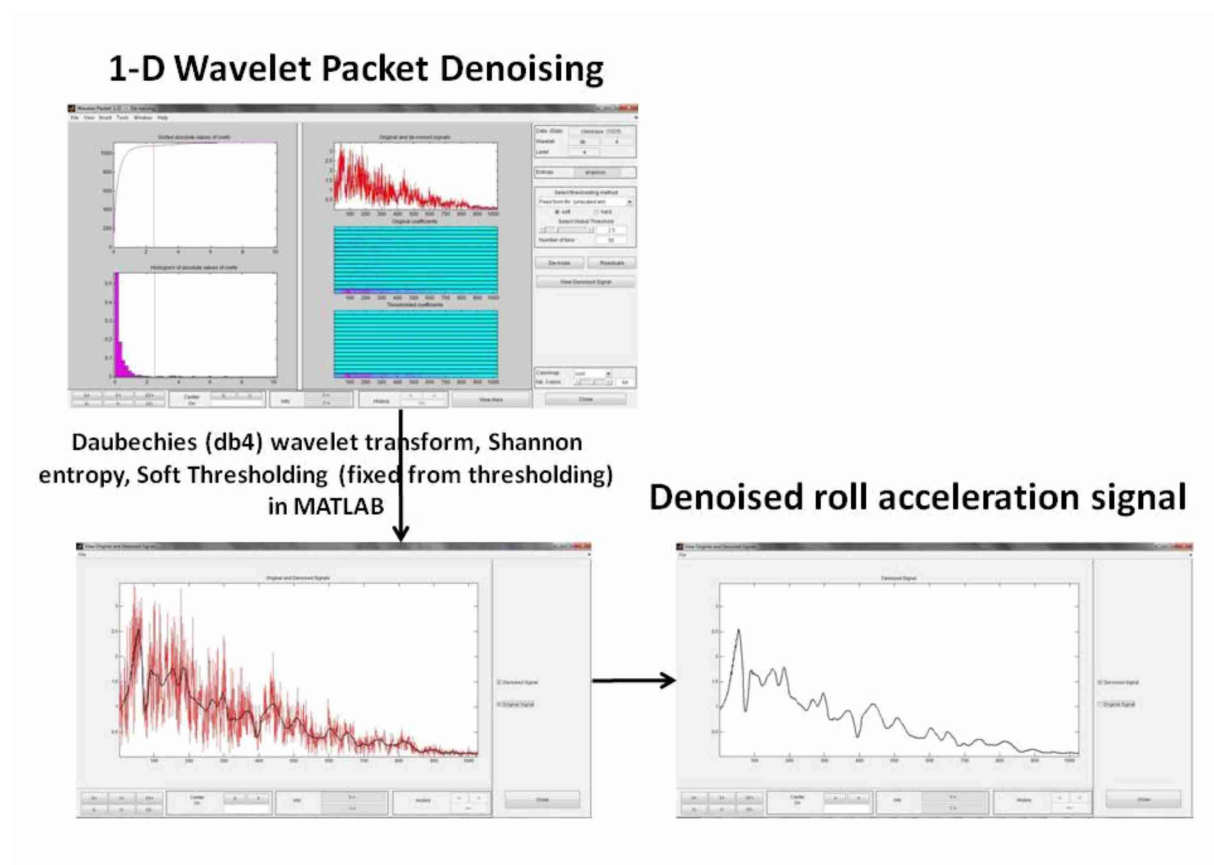


Fig. VIII-11 : Procédure débruitage

8.4.2. Résultats et Discussion

Dans cette étude, deux différents algorithmes de pointage sur de mouvement ont été proposés comme classiques et optimale (LQR DT) des algorithmes. Leurs erreurs rouleur secousse sensation base (Tableau VIII-4 et Tableau VIII-5) ont été comparés afin de déterminer le conflit sensorielle [OMA89, Oma90] pour des conditions le même scénario choisi (Fig. VIII-13).

De plus, les fréquences naturelles et leurs amplitudes correspondantes ont été discutées.

Les expériences ont été conduits sur deux conditions de mouvement entrées en lecture comme classiques et optimale (DT LQR) algorithmes (Fig. VIII-12). Selon l'approche proposée, l'erreur a été la sensation visant à diminuer afin de réduire le mal du simulateur

avec la mise en œuvre de l'utilisation optimale (DT LQR) algorithmes sur le simulateur par rapport à l'algorithme classique déjà mis en œuvre.

La Fig. IV-3 montre les diagrammes de la LQR DT utilisés et des algorithmes classiques pointage sur mouvement. Les lignes en pointillés représentent la stratégie pour la LQR DT, tandis que les lignes continues montrent la stratégie classique. Il peut facilement être identifié que la stratégie classique est en boucle ouverte, à l'inverse, la LQR DT est un système en boucle fermée avec un retour variable d'état qui permet de contrôler la dynamique de la plateforme.

La Fig. VIII-13 illustre un scénario spécifique au pays route qui a été utilisé dans le stade expérimental. En raison de cette figure, la courbe rouge montre la trajectoire suivie par le véhicule.

Selon le scénario le véhicule va droit, il prend la première rue à gauche, il continue jusqu'à ce que le virage à droite soit atteint, il prend le virage à droite et la pousse jusqu'à la section carrefour du scénario. Puis il s'arrête avec un frein complet (Fig. VIII-14). Au cours de la ligne droite, le véhicule roule à une vitesse constante de 60 km / h pour 126 secondes (Fig. VIII-14).

Le tableau I montre les limites maximales des déplacements en translation et angulaires, vitesses et accélérations de la plate-forme hexapode pour le simulateur de conduite SAAM [APM+11, APMK12a, APMK12b] où l'accélération gravitationnelle est donnée par $g \approx 9.81 \text{ m/s}^2$.

Les algorithmes de contrôle mis en œuvre ont pris les contraintes de la plate-forme hexapode du simulateur en compte (Fig. IV-3 et Tableau I-2).

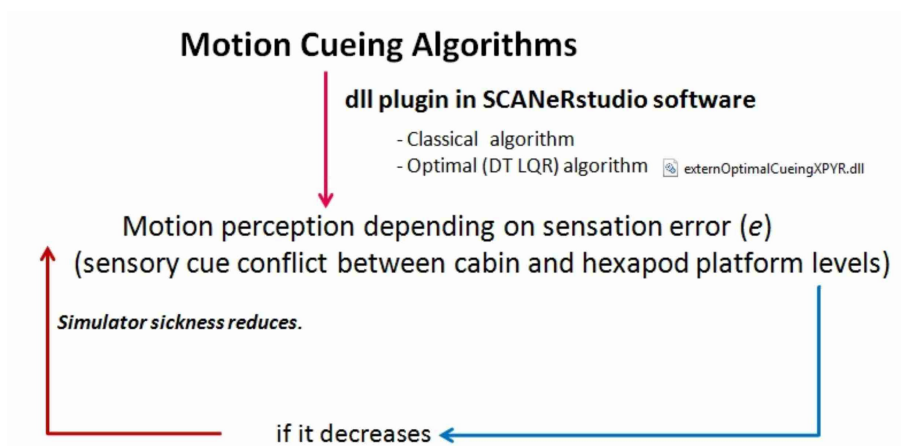


Fig. VIII-12 : Intégration d'algorithme de la plateforme dans le simulateur

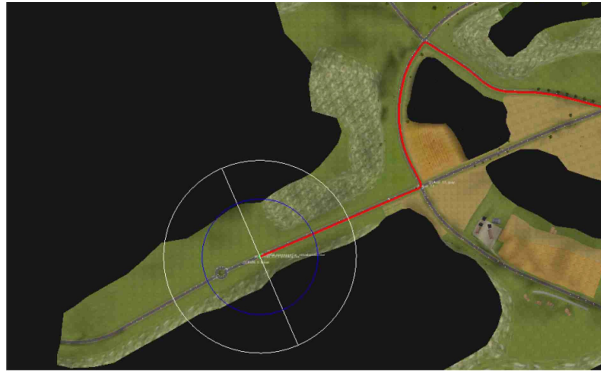


Fig. VIII-13 : Trajectoire du scénario conduit dans les expériences

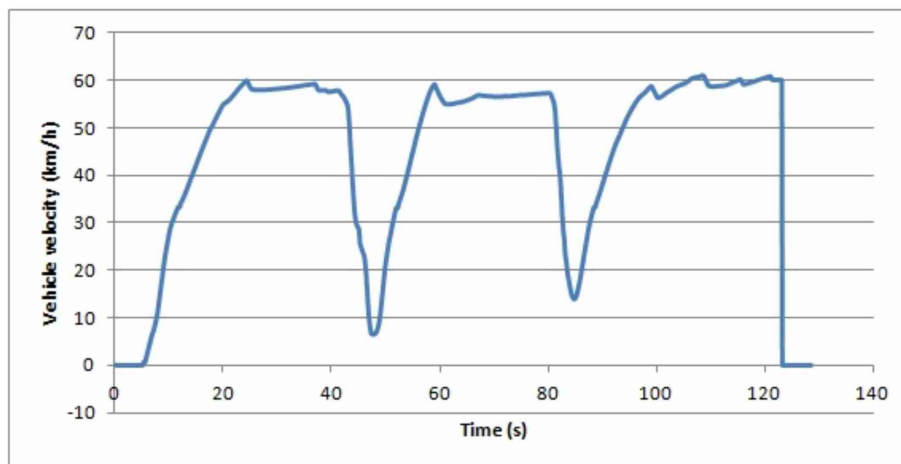


Fig. VIII-14 : Profil de vitesse du véhicule pour des expériences du simulateur

Le Tableau IV-2 résume le mouvement de repérage des paramètres utilisés pour le fonctionnement dynamique du SAAM [APM+11, APMK12a, APMK12b].

La Fig. VIII-15 reprend la logique de base de simulateurs automobiles dynamiques. Ce document examine les algorithmes de mouvement d'entraînement et les effets sur la maladie du simulateur en intégrant deux types de stratégie de repérage de mouvement.

L'algorithme DT LQR a été conçu pour avoir une erreur de sensation plus faible (plateforme de changement de niveau de la cabine, voir les Tableau VIII-4 et Tableau VIII-5), ainsi le mal du simulateur pourrait être diminué.

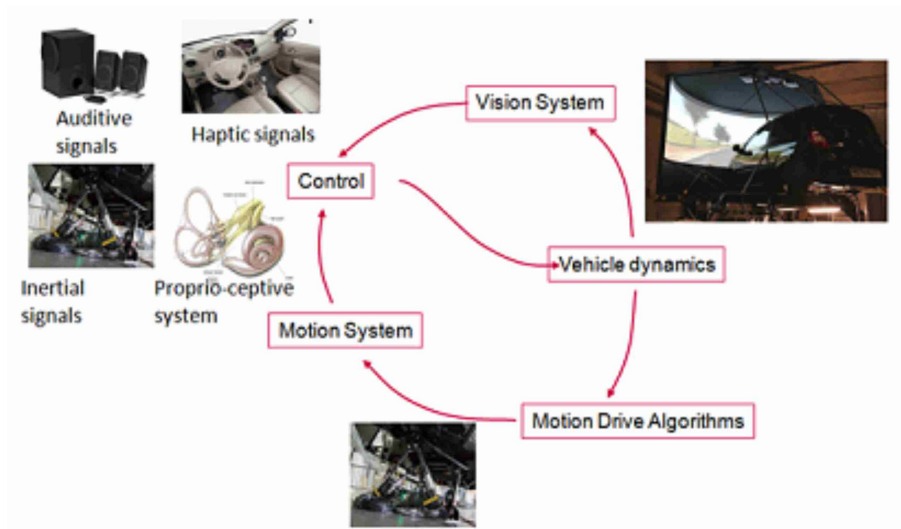


Fig. VIII-15 : Interaction multi sensorielle dans le simulateur de conduite dynamique

La maladie conflit sensoriel base a été évaluée au niveau du véhicule plate-forme dans le domaine fréquentiel (Fig. VIII-16 et Fig. VIII-17). Les courbes bleues montrent l'algorithme de repérage mouvement classique tandis que les rouges en pointillés illustrent l'utilisation optimale (DT LQR) algorithme des Fig. VIII-16 et Fig. VIII-17.

L'Eq. (100) permet de calculer la somme des secousses de roulis ressenties au niveau de la cabine (véhicule). Le modèle dynamique du véhicule qui se déplace dans l'environnement visuel pendant les expériences d'entraînement sur simulateur représente le mouvement au niveau de la cabine.

Le modèle du véhicule doit être créé en fonction des données (données Cinématique: angle de carrossage, pincement et toe-out, etc. Dynamics données: suspension, amortisseurs, pneus, etc.) d'un véhicule réel dans le logiciel de simulation de conduite [WBE99]. Il devrait être noté que même modèle de voiture passager a été utilisé pour les deux algorithmes pendant la phase expérimentale.

$$\text{Somme des secousses du roulis au niveau cabine (véhicule)} = \int_0^f \ddot{\varphi}_{veh} \cdot df \quad (100)$$

où $\ddot{\varphi}_{veh}$ est l'accélération roulis du véhicule (cabine) en $^{\circ}/s^2$ et f est la fréquence en Hz.

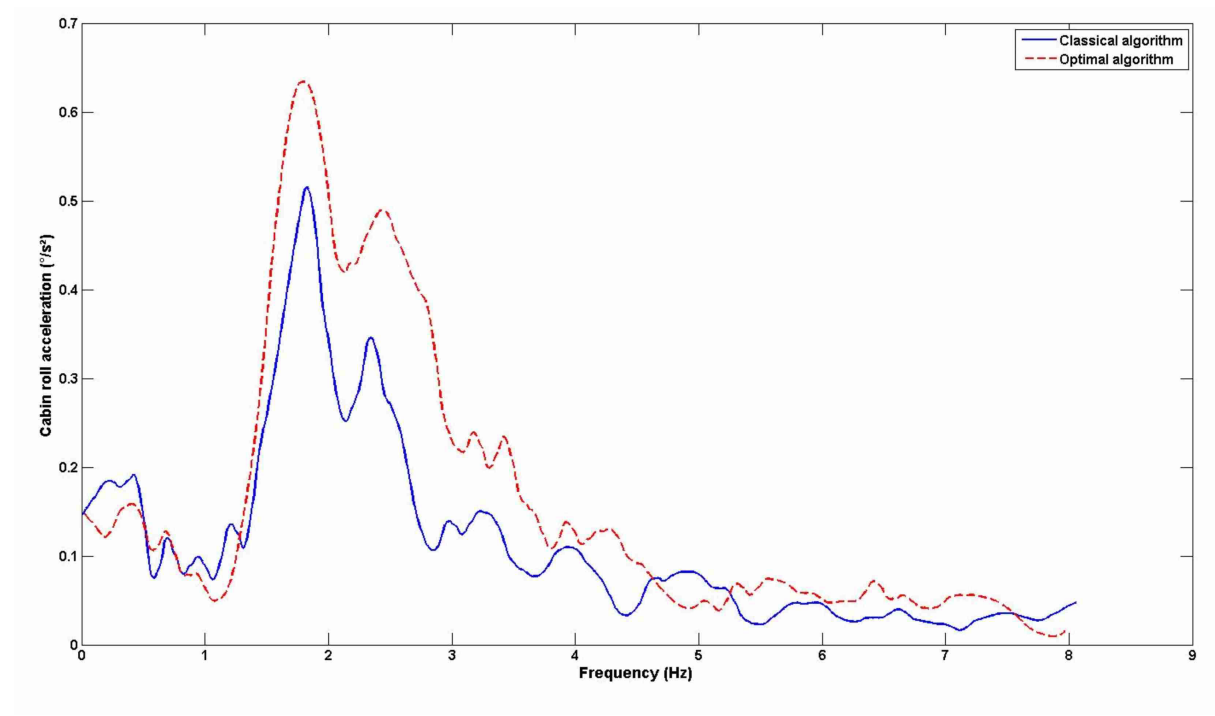


Fig. VIII-16 : Analyse fréquentielle d'accélération roulis de la cabine

Tableau VIII-4 : Résultats basés accélération roulis cabine dans le domaine fréquentiel

	Algorithme classique	Algorithme optimale (DT LQR)
Fréquences naturelles-(Hz)	1.8-2.2	1.8-2.5
Amplitudes correspondant à des fréquences naturelles (°/s²)	0.5-0.34	0.62-0.48
Somme des secousses de roulis détectées-(°/s³)	0.92	1.30

Eq. (101) calcule la somme des secousses de roulis ressenties au niveau de la plateforme de simulation. Les secousses sont définies comme la dérivée première de l'accélération. Ces secousses sont engendrées par la dynamique de plate-forme hexapode qui se déplace pendant les expériences d'entraînement sur le simulateur (Fig. VIII-15).

Selon le type de mouvement de repérage, s'il s'agit d'une commande en boucle ouverte ou d'un contrôle en boucle fermée (Fig. IV-3 et Fig. VIII-12), les réponses fréquentielles sont différentes (cf. Fig. VIII-17).

Somme des secousses de roulis ressenties-au niveau plateforme du simulateur = $\int_0^f \ddot{\varphi}_s \cdot df$ (101)

où $\ddot{\varphi}_s$ est l'accélération roulis de la plateforme du simulateur en $^{\circ}/s^2$ et f est la fréquence en Hz.

Dans la phase de post-traitement des données acquises en temps réel pendant les entraînements sur simulateur, la commande 'trapz' a été utilisée dans MATLAB pour calculer la somme des secousses de roulis pour les deux algorithmes aux niveaux cabine et plateforme.

La fréquence de résonance du roulis est fixée à environ 2 Hz dans la littérature ou plus dans le véhicule (masse suspendue) [Dix08]. D'autres auteurs fixent cette valeur à 1,5 Hz pour une voiture particulière [CN06].

[Don95] ont montré que le mouvement latéral (roulis, lacet, etc.) dans la gamme de fréquences de 0.1-0.5 Hz induit le mal des transport.

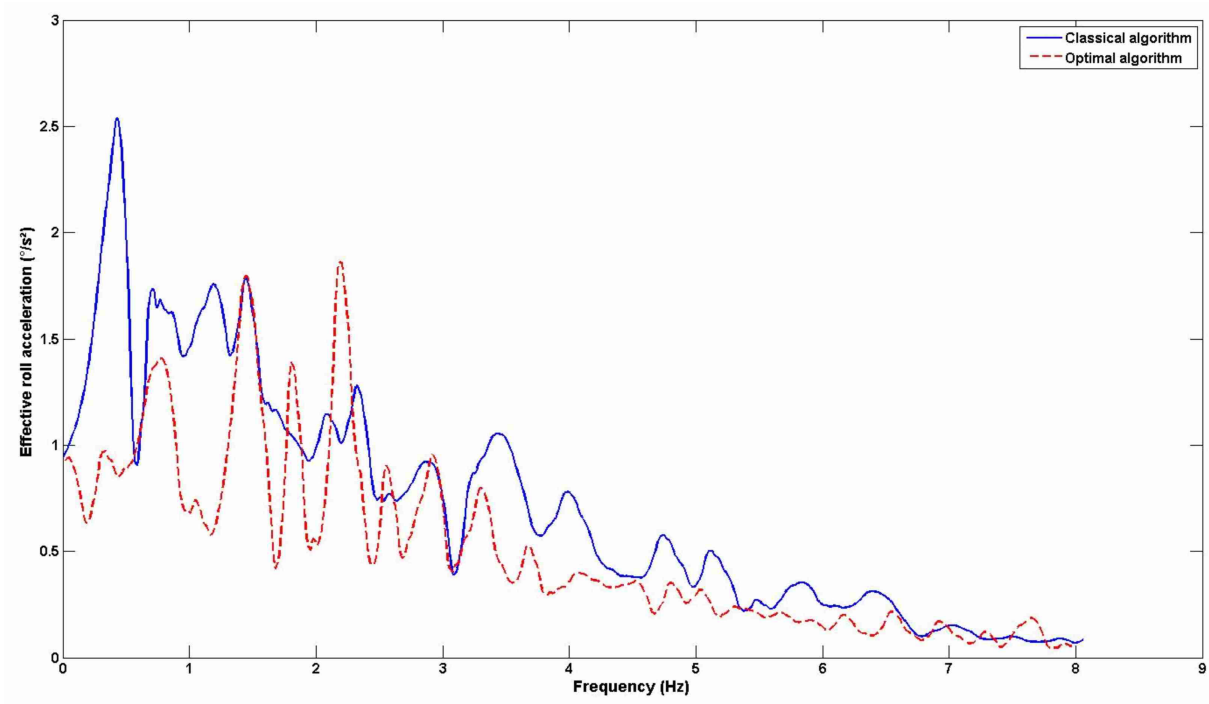


Fig. VIII-17 : Analyse fréquentielle d'accélération roulis de la plateforme

Tableau VIII-5 : Résultats basés accélération roulis de la plate-forme dans le domaine fréquentiel

	Algorithme classique	Algorithme optimal (DT LQR)
Fréquences naturelles -(Hz)	0.4-1.5	1.5-2.2
Amplitudes correspondant à des fréquences naturelles ($^{\circ}/s^2$)	2.6-1.7	1.7-1.8
Somme des secousses de roulis détectées-($^{\circ}/s^3$)	5.76	4.05

Les Tableau VIII-4 et Tableau VIII-5 montrent les résultats de l'analyse fréquentielle des accélérations de roulis mesurées pour le véhicule et la plate-forme dans des expériences de simulation en temps réel.

On voit d'après ces tableaux que les sensibilités épidermiques au roulis ($5.76/0.92 \approx 6.26$ pour l'algorithme classique et $4.05/1.30 \approx 3.12$ pour l'algorithme DT LQR), on voit que ce ratio est divisé par deux avec l'utilisation de l'algorithme DT LQR.

8.4.3. Conclusion

Cette étude a examiné l'effet d'un contrôle de suivi en boucle fermée (algorithme DT LQR, stratégie basée sur un contrôle optimal) de la plate-forme hexapode (entre le véhicule plates-formes niveaux).

L'algorithme optimal (DT LQR) engendre deux fois moins de signaux contradictoires entre les signaux visuels (véhicule) - inertiels (plate-forme) pour le même scénario de conduite avec les mêmes conditions.

En ce qui concerne le mal des transports, l'algorithme classique présente une fréquence de roulis naturel de 0.4 Hz au niveau de la plate-forme ce qui induit le mal des transport selon [Don95] (Tableau VIII-5). D'autre part, en utilisant l'algorithme DT LQR la fréquence de roulis naturel a augmenté à 1.5 Hz pour la masse suspendue (carrosserie du véhicule) ce qui est positif pour les voitures de tourisme [CN06] et permet de réduire le mal des transports.

Conclusion Générale et Perspective

Ces travaux de recherche avaient pour objectif de répondre aux questions suivantes :

- Quels sont les méthodes et outils permettant d'évaluer le mal de simulateur en conduite simulée ?
- Est-il possible de mesurer le mal de simulateur de façon objective ?
- L'engagement du sujet dans la conduite influence-t-il le mal de simulateur ?
- Quelle est l'influence du contrôle de la plateforme sur le mal du simulateur ?

La littérature et la majorité des études actuelles montrent que le mal du simulateur est souvent évalué de manière subjective notamment grâce aux questionnaires. L'état de l'art mené dans ces travaux fait état de ces questionnaires et des limitations qu'ils induisent. Ainsi, il n'est pas toujours facile d'évaluer le degré de nausée ou de vomissement atteint lors d'une simulation de plus si ce degré doit être compris entre 1 et 10.

C'est pourquoi, nous avons cherché et proposé des méthodes de mesures objectives. Dans un premier temps, nous avons équipé l'utilisateur du accéléromètre afin de mesurer l'impact des algorithmes de pilotage sur la tête du conducteur. Cette donnée supplémentaire permet de quantifier la restitution inertielle. Le mouvement de la plateforme n'étant forcément pas réaliste, nous pouvons grâce à cette mesure évaluer si les mouvements de la tête sont en phase avec la simulation. Le mal du simulateur est aussi lié au stress, à la fatigue ou la tension des conducteurs. Nous nous sommes équipés d'une centrale d'acquisition permettant des mesures électromyographiques. Grâce à ces capteurs, nous avons pu mesurer la tension superficielle au niveau des bras des conducteurs reflétant ainsi la crispation sur le volant. Cette donnée est très importante car elle renseigne directement l'état physiologique du conducteur. Cette donnée est aussi acquise en temps réel, nous pouvons par conséquent l'associer et la comparer aux autres enregistrements en particulier les accélérations qui sont la conséquence directe du

choix de l'algorithme de pilotage de la plateforme. Ainsi, nous pouvons dire que nous avons proposé des méthodes de mesures objectives du mal du simulateur.

L'implication du sujet dans une simulation de conduite est aussi un facteur permettant de réduire le mal du simulateur. L'expérimentation avec les conducteurs, passagers guidés et passagers non-guidés montre que le guidage permet de réduire le sentiment de malaise pour les passagers mais n'atteint pas le niveau des conducteurs. Il en résulte que l'implication, l'action de conduite prédispose le corps et ses récepteurs dans le mouvement du véhicule ce qui permet de mieux le supporter.

Ces travaux de recherche ont aussi permis de travailler sur les algorithmes de pilotage. A l'origine, la plateforme était pilotée par un algorithme fonctionnant en boucle ouverte. Différentes solutions ont été proposées et étudiées afin de réduire le mal du simulateur et de proposer une réponse stable et rapide aux sollicitations du modèle dynamique du véhicule. Ainsi, l'algorithme optimal LQR a permis de réduire les conflits entre les indices visuels et signaux inertiels. Par ailleurs, les algorithmes hybrides (adaptatif et optimal) offrent une alternative intéressante pour combiner les avantages des deux familles d'algorithmes.

Ces travaux ont permis d'améliorer considérablement le fonctionnement et le réalisme du simulateur. Désormais, nous pouvons envisager de l'utiliser pour des expérimentations orientées vers des usages du simulateur. Par exemple, pour des situations de conduite accidentogènes, il serait envisageable et possible d'enrichir les mesures objectives, en ajoutant la mesure et l'analyse de la fréquence cardiaque ou le niveau de sudation des conducteurs.

Par ailleurs, l'amélioration du simulateur est toujours possible, en terme de champs de vision (agrandissement) ou d'enrichissement sensoriel au sein de la cabine. Un projet de master recherche a permis cette année d'installer des systèmes vibrants dans le cockpit permettant ainsi d'avoir un retour physique du contact pneu/route. Cet axe pourrait être encore investigué.

Bibliographie

[AAL⁺07] A.F.J. Abercromby, W.E. Amonette, C.S. Layne, B.K. MCFARLIN, M.R. HINMAN, and W.H. PALOSKI. Vibration exposure and biodynamic responses during whole-body vibration training. *Medicine & Science in Sports & Exercise*, 39(10):1794, 2007.

[AC08] D.E. Angelaki and K.E. Cullen. Vestibular system: the many facets of a multimodal sense. *Annu. Rev. Neurosci.*, 31:125–150, 2008.

[Adv01] S. Advani. Integrated motion cueing algorithm and motion-base design for effective road vehicle simulation. 2001.

[AGD09] D.E. Angelaki, Y. Gu, and G.C. DeAngelis. Multisensory integration: psychophysics, neurophysiology and computation. *Current opinion in neurobiology*, 19(4):452, 2009.

[AL09] Bruno Augusto and Rui Loureiro. Motion cueing in the chalmers driving simulator-a model predictive control approach. 2009.

[AMP⁺13] Baris Aykent, Frédéric Merienne, Damien Paillot, Andras Kemeny, et al. Influence of inertial stimulus on visuo-vestibular cues conflict for lateral dynamics at driving simulators. *J Ergonomics*, 3(1):1–8, 2013.

[APM⁺11] B. Aykent, D. Paillot, F. Mérienne, Z. Fang, and A. Kemeny. Study of the influence of different washout algorithms on simulator sickness for a driving simulation task. ASME, 2011.

[ANW⁺03] Hironori Akiduki, Suetaka Nishiike, Hiroshi Watanabe, Katsunori Matsuoka, Takeshi Kubo, and Noriaki Takeda. Visual-vestibular conflict induced by virtual reality in humans. *Neuroscience letters*, 340(3):197–200, 2003.

[APMK12] Baris Aykent, Damien Paillot, Frédéric Merienne, and Andras Kemeny. A lqr washout algorithm for a driving simulator equipped with a hexapod platform: The relationship of neuromuscular dynamics with the sensed illness rating. 2012.

[APMK12a] Baris Aykent, Damien Paillot, Frédéric Mérienne, and Andras Kemeny. The influence of the feedback control of the hexapod platform of the saam dynamic driving simulator on neuromuscular dynamics of the drivers. In *Driving Simulation Conference, Paris, France*, 2012.

[APMK12b] Baris AYKENT, Damien Paillot, Frédéric Merienne, and Andras Kemeny. A lqr washout algorithm for a driving simulator equipped with a hexapod platform: The relationship of neuromuscular dynamics with the sensed illness rating. 2012.

[AQT] Imad Al Qaisi and Ansgar Traechtler. Constrained linear quadratic optimal controller for motion control of atmos driving simulator, DSC 2012.

- [AS84] D. Ariel and R. Sivan. False cue reduction in moving flight simulators. *Systems, Man and Cybernetics, IEEE Transactions on*, (4):665–671, 1984.
- [ATM09] B Aykent, W Tomaske, and M Meywerk. Study of the influence of an active roll controller on a load-dependent vehicle stability. *SAE International*, pages 01–1669, 2009.
- [AWM01] D.E. Angelaki, M. Wei, and D.M. Merfeld. Vestibular discrimination of gravity and translational acceleration. *Annals of the New York Academy of Sciences*, 942(1):114–127, 2001.
- [Ayk10] B Aykent. Effects of sway acceleration control on rollover propensity and assessment of lateral specific forces. *SAE Technical Paper*, pages 01–1325, 2010.
- [BBdG⁺98] W. Bles, J.E. Bos, B. de Graaf, E. Groen, and A.H. Wertheim. Motion sickness: only one provocative conflict? *Brain research bulletin*, 47(5):481–487, 1998.
- [Ben88] A.J. Benson. Motion sickness. *Aviation Medicine*,, pages 318–338, 1988.
- [Ben90] AJ Benson. Sensory functions and limitations of the vestibular system. Perception and Control of Self-motion Eds R Warren, AH Wertheim (Hillsdale, NJ: Lawrence Erlbaum Associates) pp, pages 145–170, 1990.
- [Ber09] D.R. Berger. Scene scaling during simulated forward accelerations: Are explicit size cues used? *Max Planck Institute for Biological Cybernetics*, 2009.
- [BGV09] MJA Bolzan, FL Guarnieri, and P.C. Vieira. Comparisons between two wavelet functions in extracting coherent structures from solar wind time series. *Brazilian Journal of Physics*, 39(1):12–17, 2009.
- [BHB89] AJ Benson, EC Hutt, and SF Brown. Thresholds for the perception of whole body angular movement about a vertical axis. *Aviation, space, and environmental medicine*, 1989.
- [BIO11] BIOPAC. Software guide for life science research applications, data acquisition and analysis with biopac mp systems reference manual,for acqknowledge® 4.2 software & mp150 or mp36r hardware/firmware on windows® 7 or vista or mac os® x 10.4-10.6, 2011. pp.357-358.
- [Bla96] Eva Blana. A survey of driving research simulators around the world. 1996.
- [BPB02] A.J. Benson, B.K. Pandoff, and E.R. Burr. Motion sickness. *Medical aspects of harsh environments*, 2:1048–1083, 2002.
- [Bri13] Britannica. arm: muscles of the upper arm, 2013. <http://www.britannica.com/EBchecked/media/119224/Muscles-of-the-upper-arm>.
- [BSPB⁺07] D.R. Berger, J. Schulte-Pelkum, H.H. Bühlhoff, P. Breuer, KI Kim, W. Kienzle, V. Blanz, B. Schölkopf, T. Meilinger, BE Riecke, et al. Simulating believable forward accelerations on a stewart motion platform. *month*, (159), 2007.
- [BSPB10] Daniel R. Berger, Jörg Schulte-Pelkum, and Heinrich H. Bühlhoff. Simulating believable forward accelerations on a stewart motion platform. *ACM Trans. Appl. Percept.*, 7(1):5:1–5:27, January 2010.
- [BSS86] AJ Benson, MB Spencer, and JR Stott. Thresholds for the detection of the direction of whole-body, linear movement in the horizontal plane. *Aviation, space, and environmental medicine*, 1986.

- [BVMB12] M. Becerra-Vargas and E. Morgado Belo. Application of h-infinity theory to a 6 dof flight simulator motion base. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 34(2):193–204, 2012.
- [CAC⁺02] R. Curry, B. Artz, L. Cathey, P. Grant, and J. Greenberg. Kennedy ssq results: fixed-vs motion-based ford simulators. In *Proceedings of DSC*, pages 289–300, 2002.
- [Cas85] J.G. Casali. Vehicular simulation-induced sickness. volume 1. an overview. Technical report, DTIC Document, 1985.
- [CKK⁺03] Y.M. Cho, I.K. Kim, M.S. Kam, H.S. Kim, S.M. Hur, and K.D. Min. Fast design of the qp-based optimal trajectory for the eclipse-11 motion simulator. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 2, pages 1232–1237. IEEE, 2003.
- [CLC09] Y.H. Chang, C.S. Liao, and W.H. Chieng. Optimal motion cueing for 5-dof motion simulations via a 3-dof motion simulator. *Control Engineering Practice*, 17(1):170–184, 2009.
- [CN06] B. Cheung and A. Nakashima. A review on the effects of frequency of oscillation on motion sickness. Technical report, DTIC Document, 2006.
- [Cra90] G.H. Crampton. Neurophysiology of motion sickness. *Motion and Space Sickness*. CRC Press, Boca Raton, pages 29–42, 1990.
- [CRW86] John G Casali, J Richard Roesch, and Walter W Wierwille. *Vehicular simulator-induced sickness*. Human Factors Laboratory, Department of Industrial Engineering and Operations Research, Virginia Polytechnic Institute and State University, 1986.
- [CTH99] F. Cardullo, R. Telban, and J. Houck. Motion cueing algorithms: A human centered approach. In *5th International Symposium on Aeronautical Sciences*, 1999.
- [CW92] R.R. Coifman and M.V. Wickerhauser. Entropy-based algorithms for best basis selection. *Information Theory, IEEE Transactions on*, 38(2):713–718, 1992.
- [DAI10] DAIMLER. New driving simulator taken into operation in sindelfingen: Investment in cutting-edge technologies, october 2010. <http://www.daimler.com/dccom/0-5-1418703-1-1338848-1-0-0-0-0-7751-7165-0-0-0-0-0-0-0.html> .
- [DB73] J. Dichgans and T. Brandt. Optokinetic motion sickness and pseudo-coriolis effects induced by moving visual stimuli. *Acta Oto-Laryngologica*, 76(1-6):339–348, 1973.
- [DBP⁺06] M. Dagdelen, J.C. Berlioux, F. Panerai, G. Reymond, and A. Kemeny. Validation process of the ultimate high-performance driving simulator. *DSC 2006 Europe*, 2006.
- [Dix08] J. Dixon. *The shock absorber handbook*. Wiley, 2008.
- [DJ⁺94] D.L. Donoho, I.M. Johnstone, et al. Ideal denoising in an orthonormal basis chosen from a library of bases. *Comptes rendus de l'Académie des sciences. Série I, Mathématique*, 319(12):1317–1322, 1994.
- [DL88] P. DiZio and JR Lackner. The effects of gravitoinertial force level and head movements on post-rotational nystagmus and illusory after-rotation. *Experimental Brain Research*, 70(3):485–495, 1988.
- [DL89] P. DiZio and J.R. Lackner. Perceived self-motion elicited by postrotary head tilts in a varying gravitoinertial force background. *Attention, Perception, & Psychophysics*, 46(2):114–118, 1989.

- [DL92] P. DiZio and J.R. Lackner. Spatial orientation, adaptation, and motion sickness in real and virtual environments. *Presence*, 1(3):319–328, 1992.
- [DM95] N.I. Durlach and A.S. Mavor. *Virtual reality: Scientific and technological challenges*. National Academies Press, 1995.
- [Don95] D.L. Donoho. De-noising by soft-thresholding. *Information Theory, IEEE Transactions on*, 41(3):613–627, 1995.
- [DP97] Manuel A Duarte and Rodrigo F Ponce. Discrete-time combined model reference adaptive control. *International journal of adaptive control and signal processing*, 11(6):501–517, 1997.
- [Dra96] M.H. Draper. Can your eyes make you sick? investigating the relationship between the vestibulo-ocular reflex and virtual reality. *Retrieved February*, 12:2000, 1996.
- [Dra98] M.H. Draper. The adaptive effects of virtual interfaces: vestibulo-ocular reflex and simulator sickness. PhD thesis, University of Washington, 1998.
- [DRK⁺04] M. Dagdelen, G. Reymond, A. Kemeny, M. Bordier, and N. Maki. Mpc based motion cueing algorithm: Development and application to the ultimate driving simulator. In *Conférence simulation de conduite*, pages 221–233, 2004.
- [DRK⁺09] M. Dagdelen, G. Reymond, A. Kemeny, M. Bordier, and N. Mazi. Model-based predictive motion cueing strategy for vehicle driving simulators. *Control Engineering Practice*, 17(9):995–1003, 2009.
- [eCDicaoN] ELECTRONIC CODE OF FEDERAL REGULATIONS. e-CFR Data is current as of November 2 2012, U.S. GOVERNMENT PRINTING OFFICE. Title 49: Transportation part 571—federal motor vehicle safety standards subpart b—federal motor vehicle safety standards § 571.126 standard no. 126; electronic stability control systems.
- [emg13] emg. Electromyography, 2013. <http://en.wikipedia.org/wiki/Electromyography>.
- [Eng01] Douglas C Engelbart. Augmenting human intellect: a conceptual framework (1962). *PACKER, Randall and JORDAN, Ken. Multimedia. From Wagner to Virtual Reality*. New York: *WW Norton & Company*, pages 64–90, 2001.
- [FLZ⁺12] Weijie Fu, Yu Liu, Songning Zhang, Xiaojie Xiong, and Shutao Wei. Effects of local elastic compression on muscle strength, electromyographic, and mechanomyographic responses in the lower extremity. *Journal of Electromyography and Kinesiology*, 22(1):44–50, 2012.
- [FM94] R. Fitzpatrick and DI McCloskey. Proprioceptive, visual and vestibular thresholds for the perception of sway during standing in humans. *The Journal of physiology*, 478(Pt 1):173–186, 1994.
- [Fra86] LH Frank. Simulator sickness: A review of its costs, countermeasures, and prediction (technical paper 861782). warrendale, pa: Society of automotive engineers. frank, lh, casali, jg, & wierwille, ww (1988). effects of visual display and motion system delays on operator performance and uneasiness in a driving simulator. *Human Factors*, 30(2):201–217, 1986.
- [GB12] N.J.I. Garrett and M.C. Best. Evaluation of a new body-sideslip-based driving simulator motion cueing algorithm. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 2012.

- [GBE⁺98] D.D. Gunter, W.W. Bylsma, K. Edgar, M.D. Letherwood, and D.J. Gorsich. Using modeling and simulation to evaluate stability and traction performance of a track laying robotic vehicle. In *Proc. of SPIE Vol*, volume 5805, page 67, 1998.
- [GIB⁺97] Robert Gentleman, Ross Ihaka, D Bates, et al. The r project for statistical computing. *R home web site: <http://www.r-project.org>*, 1997.
- [Gil07] Uwe Gille. File:sternocleidomastoideus.png, 2007. <http://commons.wikimedia.org/wiki/File:Sternocleidomastoideus.png>.
- [GLT09] J. Guo, Y. Liu, and G. Tao. Multivariable mrac with state feedback for output tracking. In *American Control Conference, 2009. ACC'09.*, pages 592–597. IEEE, 2009.
- [GM61] F.E. Guedry and EK Montague. Quantitative evaluation of the vestibular coriolis reaction. *Aerospace Med*, 32(6):487, 1961.
- [GN04a] MJ Griffin and MM Newman. An experimental study of low-frequency motion in cars. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 218(11):1231–1238, 2004.
- [GN04b] M.J. Griffin and M.M. Newman. Visual field effects on motion sickness in cars. *Aviation, space, and environmental medicine*, 75(9):739–748, 2004.
- [Gri90] MJ Griffin. *Handbook of human vibration*. Academic press, 1990.
- [Gri96] MJ Griffin. *Handbook of human vibration*. Academic Pr, 1996.
- [Gri04] MJ Griffin. Minimum health and safety requirements for workers exposed to hand-transmitted vibration and whole-body vibration in the european union; a review. *Occupational and environmental medicine*, 61(5):387–397, 2004.
- [Gue64] F.E. Guedry. Visual control of habituation to complex vestibular stimulation in man. *Acta oto-laryngologica*, 58(1-6):377–389, 1964.
- [Gue74] FE Guedry. Psychophysics of vestibular sensation. Vestibular system. Part 2- Psychophysics, applied aspects, and general interpretations.(A 75-41751 20-52) Berlin, Springer-Verlag, 1974., pages 3–154, 1974.
- [Gun78] AJ Gundry. Thresholds of perception for periodic linear motion. *Aviation, space, and environmental medicine*, 1978.
- [HB57] M Dean Havron and Leslie F Butler. Evaluation of training effectiveness of the 2fh2 helicopter flight trainer research tool. *Naval Training Device Center, Port Washington, NY*, 1957.
- [HBK⁺90] L.J. Hettinger, K.S. Berbaum, R.S. Kennedy, W.P. Dunlap, and M.D. Nolan. Vection and simulator sickness. *Military Psychology*, 2(3):171–181, 1990.
- [Het02] LJ Hettinger. Illusory self-motion in virtual environments. *Handbook of virtual environments Design implementation and applications*, pages 471–492, 2002.
- [HL05] D.C. Hanselman and B. Littlefield. *Mastering Matlab 7*. Pearson Prentice Hall, 2005.
- [HLP09] François Husson, Sébastien Lê, and Jérôme Pagès. *Analyse de données avec R*. Presses Universitaires de Rennes, 2009.
- [HM98] Jan E Holly and Gin McCollum. Timing of secondary vestibular neuron responses to a range of rotational head movements. *Biological cybernetics*, 79(1):39–48, 1998

- [HR92] L. Hettinger and G. Ricco. Virtually induced motion sickness in virtual environments. *Presence: Teleoperators and Virtual Environments*, 1(3):306–310, 1992.
- [Hua10] C.I. Huang. Adaptive washout filter design with human visual-vestibular based (hvvb) for vr-based motion simulator. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–6. IEEE, 2010.
- [HYLS09] T.S. Hwang, S.K. Yeh, J.R. Lin, and W.P. Su. Adaptive motion washout filter design by using self-tuning fuzzy control. In *Advanced Intelligent Mechatronics, 2009. AIM 2009. IEEE/ASME International Conference on*, pages 811–815. IEEE, 2009.
- [IS96] P.A. Ioannou and J. Sun. Robust adaptive control. 1996.
- [ISO97] ISO. Mechanical Vibration and Shock: Evaluation of Human Exposure to Whole-body Vibration. Part 1, General Requirements: International Standard ISO 2631-1: 1997 (E). ISO, 1997.
- [JHA07] H. Jamson, A.J. Horrobin, and R.A. Auckland. Whatever happened to the lads? design and development of the new university of leeds driving simulator. In *Driving Simulation Conference, North America 2007 (DSC-NA 2007)*, 2007.
- [KBA⁺88] RS Kennedy, KS Berbaum, GO Allgood, NE Lane, MG Lilienthal, and DR Baltzley. Etiological significance of equipment features and pilot history in simulator sickness. *AGARD, Motion Cues in Flight Simulation and Simulator Induced Sickness 22 p(SEE N 89-12171 03-52)*, 1988.
- [KBD94] RS Kennedy, KS Berbaum, and J. Drexler. Methodological and measurement issues for identification of engineering features contributing to virtual reality sickness. *Image*, 7, 1994.
- [KBL⁺87] R.S. Kennedy, KS Berbaum, MG Lilienthal, WP Dunlap, and BE Mulligan. Guidelines for alleviation of simulator sickness symptomatology. Technical report, DTIC Document, 1987.
- [Kem90] A. Kemeny. Prometheus-design technics. *Transportation Electronics, Vehicle Electronic in the, 90*, 1990.
- [Kem99] A. Kemeny. Simulation and perception. In *Proceedings of Driving Simulation Conference: DSC*, volume 99, 1999.
- [Kem01] A. Kemeny. Recent developments in visuo-vestibular restitution of self-motion in driving simulation. In *Proceedings of the DSC*, volume 1, pages 15–18, 2001.
- [KEM03] A. KEMENY. Motion cueing in driving simulation experiments. pages 9–10, 2003.
- [Kem04] A. Kemeny. Ultimate, 2004. <http://www.experts.renault.com/kemeny/projects/-ultimate/documents/Ultimate-FR.pdf>.
- [Kem04] A. Kemeny. Multi sensory integration and perceptive immersion in driving simulator experiments. In *Proceedings of the Driving Simulation Conference, DSC*, pages 17–20, 2004.
- [Kem09] A. Kemeny. Driving simulation for virtual testing and perception studies. In *Proceedings of the Driving Simulation Conference Europe*, pages 16–23, 2009.
- [KFBL92] R.S. Kennedy, J.E. Fowlkes, K.S. Berbaum, and M.G. Lilienthal. Use of a motion sickness history questionnaire for prediction of simulator sickness. *Aviation, Space, and Environmental Medicine*, 63(7):588–593, 1992.

- [KHL90] R.S. Kennedy, L.J. Hettinger, and M.G. Lilienthal. Simulator sickness. *Motion and space sickness*, pages 317–341, 1990.
- [Kin05] H. Kingma. Thresholds for perception of direction of linear acceleration as a possible evaluation of the otolith function. *BMC Ear, Nose and Throat Disorders*, 5(1):5, 2005.
- [KLBL93] R.S. Kennedy, N.E. Lane, K.S. Berbaum, and M.G. Lilienthal. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The international journal of aviation psychology*, 3(3):203–220, 1993.
- [KMKL10] MS Kim, YG Moon, GD Kim, and MC Lee. Partial range scaling method based washout algorithm for a vehicle driving simulator and its evaluation. *International Journal of Automotive Technology*, 11(2):269–275, 2010.
- [KNG⁺11] T. Kaminski, M. Niezgoda, J. Gacparska, R. Grzeszczyk, and P. Filipek. Advanced car driving simulator—as 1200-6. *Journal of KONES Powertrain and Transport*, 18(4), 2011.
- [Kol95] E.M. Kolasinski. Simulator sickness in virtual environments. Technical report, DTIC Document, 1995.
- [KP03] A. Kemeny and F. Panerai. Evaluating perception in driving simulation experiments. *Trends in Cognitive Sciences*, 7(1):31–37, 01 2003.
- [LAR⁺09] D Limon, T Alamo, DM Raimondo, D Munoz de la Pena, JM Bravo, A Ferramosca, and EF Camacho. Input-to-state stability: a unifying framework for robust model predictive control. In *Nonlinear model predictive control*, pages 1–26. Springer, 2009.
- [Lev12] Richard M Levy. *The virtual reality revolution: The vision and the reality*. 2012.
- [Lew92] F.L. Lewis. *Applied optimal control & estimation*, volume 10. Prentice Hall Englewood Cliffs, NJ, 1992.
- [LG87] Anthony Lawther and Michael J Griffin. Prediction of the incidence of motion sickness from the magnitude, frequency, and duration of vertical oscillation. *The Journal of the Acoustical Society of America*, 82:957, 1987.
- [LGMM02] BD Lawson, DA Graeber, AM Mead, and ER Muth. Signs and symptoms of human syndromes associated with synthetic experiences. *Handbook of virtual environments: Design, implementation, and applications*, pages 589–618, 2002.
- [LJ78] WH Levison and AM Junker. A model for the pilot’s use of motion cues in steady-state roll-axis tracking tasks. In *Flight Simulation Technologies Conference, Arlington, Tex*, pages 149–159, 1978.
- [Log88] Gordon D Logan. Toward an instance theory of automatization. *Psychological review*, 95(4):492–527, 1988.
- [MAT09a] M. Meywerk, B. Aykent, and W. Tomaske. Einfluss der fahrdynamikregelung auf die sicherheit von n1-fahrzeugen bei unterschiedlichen beladungszuständen. teil 1: Grundlagen, unfallstatistik, abstütz- und beladungseinrichtung, fahrzeugdatenermittlung. fe 82.329/2007. 2009.
- [MAT09b] M. Meywerk, B. Aykent, and W. Tomaske. Einfluss der fahrdynamikregelung auf die sicherheit von n1-fahrzeugen bei unterschiedlichen beladungszuständen. teil 2: Fahrversuche und fahrimulatorversuche. fe 82.329/2007. 2009.

- [MB02] J.G. May and D.R. Badcock. Vision and virtual environments. *Handbook of Virtual Environments: Design, Implementation, and Applications*. Mahwah, NJ: Lawrence Erlbaum Associates, Inc, pages 29–64, 2002.
- [McC84] M.E. McCauley. *Research issues in simulator sickness: Proceedings of a workshop*. National Academies, 1984.
- [MGT11] D. Maiti, J. Guo, and G. Tao. A discrete-time multivariable state feedback mrac design with application to linearized aircraft models with damage. In *American Control Conference (ACC), 2011*, pages 606–611. IEEE, 2011.
- [MH97] Jan M Maciejowski and Mihai Huzmezan. *Predictive control*. Springer, 1997.
- [MMOP08] M. Misiti, Y. Misiti, G. Oppenheim, and JM Poggi. *Matlab user’s guide: wavelet toolbox™ 4*, 2008.
- [Mon69] K.E. Money. Motion sickness. Technical report, DTIC Document, 1969.
- [MS92] M. McCauley and T. Sharkey. Cybersickness- perception of self-motion in virtual environments. *Presence: Teleoperators and Virtual Environments*, 1(3):311–318, 1992.
- [MTS09] MTS. Better understanding driver behavior, 2009. http://www.mts.com/ucm/-groups/public/documents/library/dev_004365.pdf.
- [NAA05] L. Nehaoua, A. Amouri, and H. Arioui. Classic and adaptive washout comparison for a low cost driving simulator. In *Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation*, pages 586–591. IEEE, 2005.
- [NAD10] NADS-1. Nads-1, 2010. http://www.nads-sc.uiowa.edu/sim_nads1.php.
- [NAEM06] L. Nehaoua, H. Arioui, S. Espie, and H. Mohellebi. Motion cueing algorithms for small driving simulator. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 3189–3194. IEEE, 2006.
- [NR89] M.A. Nahon and L.D. Reid. ”simulator motion-drive algorithms: A designer’s perspective”. *J. Guidance*, 13(2), 1989.
- [NRK92] MA Nahon, LD Reid, and J. Kirdeikis. Adaptive simulator motion software with supervisory control. *Journal of guidance, control, and dynamics*, 15(2):376–383, 1992.
- [Oma82] Charles M Oman. A heuristic mathematical model for the dynamics of sensory conflict and motion sickness. *Acta Oto-Laryngologica*, 1982.
- [OMA89] C. OMAN. Sensory conflict in motion sickness: an observer theory approach. *NASA, Ames Research Center, Spatial Displays and Spatial Instruments 15 p(SEE N 90-22918 16-54)*, 1989.
- [Oma90] C.M. Oman. Motion sickness: a synthesis and evaluation of the sensory conflict theory. *Canadian Journal of Physiology and Pharmacology*, 68(2):294–303, 1990.
- [PDF⁺97] J.D. Prothero, M.H. Draper, T.A. Furness, D.E. Parker, and M.J. Wells. Do visual background manipulations reduce simulator sickness. In *Proceedings of the International Workshop on Motion Sickness: Medical and Human Factors*, pages 18–21, 1997.
- [PDMJ12] R.V. Parrish, J.E. Dieudonne, and D.J. Martin Jr. Coordinated adaptive washout for motion simulators. *Journal of Aircraft*, 12(1), 2012.

- [Per08] R. Persson. Motion sickness in tilting trains: description and analysis of the present knowledge: literature study. *VTI Rapport*, (614A), 2008.
- [PF98] Jerrold D Prothero and Thomas A Furness. *The role of rest frames in vection, presence and motion sickness*. University of Washington, 1998.
- [PLY⁺01] M.K. Park, M.C. Lee, K.S. Yoo, K. Son, W.S. Yoo, and M.C. Han. Development of the pneu vehicle driving simulator and its performance evaluation. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 3, pages 2325–2330. IEEE, 2001.
- [PMC⁺76] R.V. Parrish, D.J. Martin, Langley Research Center, United States. National Aeronautics, and Space Administration. Comparison of a linear and a nonlinear washout for motion simulators utilizing objective and subjective data form CTOL transport landing approaches. NASA technical note. National Aeronautics and Space Administration, 1976.
- [QD05] G Quarck and P Denise. Le mal des transports. *La Lettre d'oto-rhino-laryngologie et de chirurgie cervico-faciale*, (298):11–15, 2005.
- [RB75] J.T. Reason and J.J. Brand. Motion sickness. 1975.
- [RC80] SN Roscoe and JM Childs. Reliable, objective flight checks(real time pilot performance scoring). *Aviation psychology.(A 81-15676 04-53) Ames, Iowa State University Press, 1980,*, pages 145–158, 1980.
- [RDK02] G. Reymond, J. Droulez, and A. Kemeny. Visuo-vestibular perception of self-motion modeled as a dynamic optimization process. *Biological Cybernetics*, 87(4):301–314, 2002.
- [Rea70] JT Reason. Motion sickness: a special case of sensory rearrangement. *Advancement of science*, 26(130):386, 1970.
- [Rea78] J. Reason. Motion sickness: Some theoretical and practical considerations. *Applied Ergonomics*, 9(3):163–167, 1978.
- [Rea78a] J. Reason. Motion sickness: Some theoretical and practical considerations. *Applied Ergonomics*, 9(3):163–167, 1978.
- [Rea78b] JT Reason. Motion sickness adaptation: a neural mismatch model. *Journal of the Royal Society of Medicine*, 71(11):819, 1978.
- [RK00] G. Reymond and A. Kemeny. Motion cueing in the renault driving simulator. *Vehicle System Dynamics*, 34(4):249–259, 2000.
- [RMS92] Gary E Riccio, Eric J Martin, and Thomas A Stoffregen. The role of balance dynamics in the active perception of orientation. *Journal of experimental psychology. Human perception and performance*, 18(3):624, 1992.
- [RN85] LD Reid and Meyer A Nahon. Flight simulation motion-base drive algorithms. part 1: Developing and testing the equations. 1985.
- [Rob80] King M Roberts. Fhwa highway driving simulator. *Public Roads*, 44(HS-031 069), 1980.
- [RS88] Gary E Riccio and Thomas A Stoffregen. Affordances as constraints on the control of stance. *Human Movement Science*, 7(2):265–300, 1988.
- [RSM81] M Rodenburg, HPW Stassen, and AJJ Maas. The threshold of perception of angular acceleration as a function of duration. *Biological Cybernetics*, 39(3):223–226, 1981.

- [RS91] G.E. Riccio and T.A. Stoffregen. An ecological theory of motion sickness and postural instability. *Ecological psychology*, 3(3):195–240, 1991.
- [SISH82] R. Sivan, J. Ish-Shalom, and J.K. Huang. An optimal control approach to the design of moving flight simulators. *Systems, Man and Cybernetics, IEEE Transactions on*, 12(6):818–827, 1982.
- [SJK03] J.B. Song, U.J. Jung, and H.D. Ko. Washout algorithm with fuzzy-based tuning for a motion simulator. *Journal of Mechanical Science and Technology*, 17(2):221–229, 2003.
- [SR91] T.A. Stoffregen and G.E. Riccio. An ecological critique of the sensory conflict theory of motion sickness. *Ecological Psychology*, 3(3):159–194, 1991.
- [SRKB01] I. Siegler, G. Reymond, A. Kemeny, and A. Berthoz. Sensorimotor integration in a driving simulator: contributions of motion cueing in elementary driving tasks. In *Proceedings of Driving Simulation Conference*, pages 21–32, 2001.
- [SRVB11a] A. Stratulat, V. Roussarie, J. Vercher, and C. Bourdin. Improving the realism in motion-based driving simulators by adapting tilt-translation technique to human perception. In *Virtual Reality Conference (VR), 2011 IEEE*, pages 47–50. IEEE, 2011.
- [SRVB11b] A.M. Stratulat, V. Roussarie, J.L. Vercher, and C. Bourdin. Does tilt/translation ratio affect perception of deceleration in driving simulators? *Journal of Vestibular Research*, 21(3):127–139, 2011.
- [Ste65] D. Stewart. A platform with six degrees of freedom. *Proceedings of the Institution of Mechanical Engineers*, 180(1):371–386, 1965.
- [Tao03] Gang Tao. *Adaptive control design and analysis*, volume 37. Wiley. com, 2003.
- [TB49] David B Tyler and Philip Bard. Motion sickness. *Physiological reviews*, 29(4):311–369, 1949.
- [TCG00] R.J. Telban, F.M. Cardullo, and L. Guo. Investigation of mathematical models of otolith organs for human centered motion cueing algorithms. In *Proceedings of the AIAA Modelling and Simulation Technologies Conference, Denver (CO), August 14*, volume 17, 2000.
- [TCH02] R.J. Telban, F.M. Cardullo, and J.A. Houck. A nonlinear, human-centered approach to motion cueing with a neurocomputing solver. *NASA, Tech. Rep*, pages 2002–4692, 2002.
- [Tho13] Edward L Thorndike. *Educational psychology*, vol 1: The original nature of man. 1913.
- [TI93] G. Tao and P.A. Ioannou. Model reference adaptive control for plants with unknown relative degree. *Automatic Control, IEEE Transactions on*, 38(6):976–982, 1993.
- [Too07] Matlab Statistical Toolbox. 6 user’s guide. *Math Works Inc*, 2007.
- [TOY07] TOYOTA. Toyota develops world-class driving simulator, november 2007. http://www.toyota.co.jp/en/news/07/1126_1.html .
- [Tre77] Anne Treisman. Focused attention in the perception and retrieval of multidimensional stimuli. *Attention, Perception, & Psychophysics*, 22(1):1–11, 1977.
- [Tya04] Andrey Alexandrovich Tyagunov. *High-performance model predictive control for process industry*. PhD thesis, Ph. D. thesis, Eindhoven university of technology, 2004.

- [XSe10] XSens. Mti and mtx user manual and technical documentation. document mt0100p, revision o., 2010.
- [XSens] XSens Technologies BV 15. Mti and mtx user manual and technical documentation. document mt0100p, revision o., 2010.
- [Xu11] X. Xu. Discrete time mixed lqr/h-infinity control problems. 2011.
- [VDG⁺98] E. Viirre, M. Draper, C. Gailey, D. Miller, T. Furness, et al. Adaptation of the vor in patients with low vor gains. *Journal of Vestibular Research*, 8(4):331–334, 1998.
- [VDS96] FAM Van Der Steen. Simulating self-motion. *Brain research bulletin*, 40(5):473–475, 1996.
- [Wat00] G. Watson. A synthesis of simulator sickness studies conducted in a high-fidelity driving simulator. In *Proceedings of Driving Simulation Conference*, pages 69–78, 2000.
- [WBE99] C.B. Winkler, D. Blower, and R.D. Ervin. Rollover of heavy commercial vehicles. Technical report, 1999.
- [WDW86] Robert B Welch, Lance D DuttonHurt, and David H Warren. Contributions of audition and vision to temporal rate perception. *Attention, Perception, & Psychophysics*, 39(4):294–300, 1986.
- [Wel78] Robert B Welch. Perceptual modification: Adapting to altered sensory environments. Academic Press New York, 1978.
- [WF81] S.C.P. Wong and BJ Frost. The effect of visual-vestibular conflict on the latency of steady-state visually induced subjective rotation. *Attention, Perception, & Psychophysics*, 30(3):228–236, 1981.
- [wik13] wikipedia. Electromyography, 2013. <http://en.wikipedia.org/wiki/Electromyography>.
- [Yar92] L. Yardley. Motion sickness and perception: A reappraisal of the sensory conflict approach. *British journal of Psychology*, 83(4):449–471, 1992.
- [YG11] X. Yang and J. Gander. Driver’s preview strategy and its impact on nato double lane change maneuver. *SAE International Journal of Mate*
- [Zee10] Eberhard Zeeb. Daimler’s new full-scale, high-dynamic driving simulator—a technical overview. In *Conference Proc. Driving Simulator Conference Europe, Paris*, 2010.
- [ZMD02] L.H. Zupan, D.M. Merfeld, and C. Darlot. Using sensory weighting to model the influence of canal, otolith and visual cues on spatial orientation and eye movements. *Biological cybernetics*, 86(3):209–230, 2002.
- [ZY81] GL Zacharias and LR Young. Influence of combined visual and vestibular cues on human perception and control of horizontal rotation

Liste des publications réalisées au cours de cette thèse

Articles publiés dans des revues avec comité de lecture

Titre : Influence of a new discrete-time LQR-based motion cueing on driving simulator

Auteurs : B Aykent, F Merienne, D Paillot, A Kemeny

Revue : Optimal Control Applications and Methods, Wiley

Parution : 15/07/2013

Titre : Influence of Inertial Stimulus on Visuo-Vestibular Cues Conflict for Lateral Dynamics at Driving Simulators

Auteurs : B Aykent, F Merienne, D Paillot, A Kemeny

Revue : Journal of Ergonomics 3 (1), 1-8

Parution : 16/01/2013

Titre : The role of motion platform on postural instability and head vibration exposure at driving simulators

Auteurs : B Aykent, F Merienne, D Paillot, A Kemeny

Revue : Human Movement Science, Elsevier

Parution : 07/12/2013

Titre : Motion sickness evaluation and comparison for a static driving simulator and a dynamic driving simulator

Auteurs : B Aykent, F Merienne, D Paillot, C Guillet, A Kemeny

Revue : Proc. IMechE, Part D: Journal of Automobile Engineering, SAGE

Parution : accepté

Communications publiées avec actes et comité de lecture

Titre : The contribution of closed loop tracking control of motion platform on laterally induced postural instability of the drivers at SAAM dynamic simulator

Auteurs : B Aykent, F Merienne, D Paillot, A Kemeny

Conférence : Confere 2013, Biarritz, France

Titre : Vibrations in dynamic driving simulator: study and implementation

Auteurs : J Plouzeau, D Paillot, B Aykent, F Merienne

Conférence : Confere 2013, Biarritz, France

Titre : A LQR Washout Algorithm for a Driving Simulator Equipped with a Hexapod Platform: The Relationship of Neuromuscular Dynamics with the Sensed Illness Rating

Auteurs : B Aykent, D Paillot, F Merienne, A Kemeny

Conférence : Confere 2012, Biarritz, France

Titre : The Influence of the feedback control of the hexapod platform of the SAAM dynamic driving simulator on neuromuscular dynamics of the drivers

Auteurs : B Aykent, D Paillot, F Merienne, A Kemeny

Conférence : Driving Simulation Conference (DSC) 6-7 September 2012, Paris, France

Titre : Study of the Influence of Different Washout Algorithms on Simulator Sickness for a Driving Simulation Task

Auteurs : B AYKENT, D Paillot, F Merienne, Z Fang, A Kemeny

Conférence : ASME 2011 World Conference on Innovative Virtual Reality (2011) pp. 331-341

Annexe - Statistiques

Les tests d'hypothèse ont comme objectif de vérifier la validité de certaines hypothèses relatives à une population à partir des données d'un ou plusieurs échantillons. On peut distinguer différents types de catégories de tests d'hypothèse :

- Les tests d'ajustement qui permettent de comparer la distribution d'un échantillon observé et d'une population théorique, comme par exemple sa normalité.
- Les tests de conformité qui ont un objectif plus précis que les tests d'ajustement en vérifiant qu'un échantillon donné peut être extrait d'une population possédant telle valeur d'un paramètre comme la moyenne ou la variance.
- Les tests d'égalité ou d'homogénéité qui ont pour but de comparer plusieurs populations à partir du même nombre d'échantillons

Dans tous les cas de figure, l'hypothèse à tester ou hypothèse nulle est en général notée H_0 . On calcule ensuite la probabilité d'observer un écart, soit entre certaines caractéristiques d'une population et d'un échantillon extrait dans le cas des tests d'ajustement ou de conformité, soit entre caractéristiques de plusieurs échantillons dans le cas de tests d'égalité. Le **risque de première espèce α** de rejeter à tort l'hypothèse nulle H_0 , alors qu'elle est vraie, s'appelle le niveau de signification du test et est défini à priori. Si Z désigne la variable étudiée, il est tel qu'il existe une valeur Z_α vérifiant $p(Z > Z_\alpha) = \alpha$.

Lors d'un test d'hypothèse, la **p-value** représente le plus petit niveau de probabilité pour lequel l'hypothèse nulle H_0 est rejetée. Elle est déterminée à posteriori car elle nécessite le calcul de Z_0 , valeur observée de la variable Z sur l'échantillon dont on dispose, telle que $p(Z > Z_0) = p$. Elle est alors comparée à α :

- Si $p < \alpha$ alors on rejette H_0
- Si $p > \alpha$ alors on accepte H_0

Les valeurs de α les plus courantes sont $\alpha=0,05$ ou $\alpha=0,01$.

Les analyses statistiques s'appuient sur des tests qui, pour la plupart, se basent sur l'estimation de paramètres de la distribution des données issues d'une population étudiée et sur des conditions d'application concernant la forme de cette distribution. Ces tests sont appelés **tests paramétriques**.

Il existe une autre famille de tests qui ne se basent pas sur l'estimation de paramètres ou des conditions d'application liées à la distribution. Ce sont les **tests non paramétriques**. Un de leurs avantages est le fait qu'ils ne sont pas soumis à ces conditions d'application très restrictives comme le sont les tests paramétriques. Ils sont ainsi plus puissants que les tests paramétriques lorsque la population n'est pas normale. Un autre intérêt de leur utilisation est qu'ils sont plus sensibles aux médianes qu'aux moyennes.

Les tests qui suivent permettent la comparaison de **$k=2$** échantillons issus d'une même population :

Le **test t** est utilisé lorsqu'on ne connaît pas la variance de la population. On considère alors la variance d'échantillon comme estimation de la variance de la population. Le **test z**, quant à lui, est utilisé lorsqu'on connaît la variance de la population. Dans les deux cas, on impose comme condition d'application, la normalité de la population dont l'échantillon a été prélevé.

Le **test U de Mann-Withney** consiste à comparer deux échantillons indépendants de faible effectif d'une même population en s'affranchissant de la condition de normalité de cette population. Pour évaluer l'hypothèse nulle H_0 selon laquelle il n'y a pas de différence entre les rangs des deux échantillons, on range par ordre croissant les données puis on calcule la somme des rangs par groupe. On calcule ensuite les quantités suivantes à partir des données :

$$U_1 = n_1 n_2 + \left(\frac{n_1(n_1 + 1)}{2} \right) - \sum_{\text{Groupe 1}} \text{rangs}$$
$$U_2 = n_1 n_2 - U_1$$

puis:

$$z = \frac{U - \frac{n_1 n_2}{2}}{\sqrt{\frac{n_1 n_2 (n_1 + n_2 + 1)}{12}}}$$

Si $z < 1,96$ alors $p > 0,05$ donc l'hypothèse nulle H_0 est validée donc il n'y a pas de différence significative entre les échantillons

Si $z > 1,96$ alors $p < 0,05$ donc l'hypothèse alternative H_1 est validée donc il y a une différence significative entre les échantillons

Les tests qui suivent permettent la comparaison de **k > 2** échantillons issus d'une même population :

Le **test de Friedman** est un test non paramétrique permettant de comparer k paramètres sur k échantillons liés. Ce test est l'alternative de **l'analyse de variance** (ANOVA) à un facteur avec mesures répétées qui est un test paramétrique.

Références bibliographiques :

Dagnelie Pierre, Statistique théorique et appliquée 1, Statistique descriptive et bases de l'inférence statistique, Bruxelles : De Boeck , DL 2013, cop. 2013

Dagnelie Pierre, Statistique théorique et appliquée 2, Inférence statistique à une et à deux dimensions, Bruxelles : De Boeck , DL 2013, cop. 2013

Howell David C ., Méthodes statistiques en sciences humaines, 2e édition, Bruxelles : De Boeck , DL 2008

Annexe - Code C++

A1. Fichiers C++

externCueing.cpp

```
#include "externCueing.h"
#include "signal.h"
#include <iostream>
#include "windows.h"
#include "math.h"
#include <fstream>

//*****
//*****
//  VARIABLES STATIQUES
//*****
//*****

static double count;
static double countIsStop;
static double stopGain;

// -----
// Fichier d'écriture des résultats
// -----
static std::ofstream myFile;

// -----
// Période d'échantillonnage des signaux
// correspondant à la fréquence de
// communication avec la plate-forme
// -----
static double PeriodeEchantillonnage = 1/60.0;

// -----
// Déclaration des signaux
// axe X : longitudinal
// -----
static signal* XAccInput;
static signal* XVelInput;
static signal* XAccFiltered;
static signal* XError;
static signal* XErrorFiltered;
static signal* XErrorFiltered2;
```

```
static signal* XAccFilteredAB;
static signal* XErrorAB;
static signal* XCriterion;
static signal* XAccCons;
static signal* XAccConsLP;
static signal* XAccConsWashout;
static signal* XAccOutput;
static signal* XVelCons;
static signal* XVelOutput;
static signal* XPosCons;
static signal* XPosOutput;

// -----
// Déclaration des signaux
// axe P : Pitch (tangage)
// -----

static signal* X2PAccFiltered;
static signal* PTiltAngle;
static signal* PTiltAngleSat;
static signal* PAngleInput;
static signal* PAngleFiltered;
static signal* PAngleFilteredSat;
static signal* PAngleCons;
static signal* PAngleOutput;
static signal* PVelCons;
static signal* PVelOutput;
static signal* PAccCons;
static signal* PAccOutput;

// -----
// Déclaration des signaux
// axe Y : latéral
// -----

static signal* YAccInput;
static signal* YVelInput;
static signal* YAccInputSat;
static signal* YAccFiltered;
static signal* YError;
static signal* YErrorFiltered;
static signal* YErrorFiltered2;
static signal* YAccFilteredAB;
static signal* YErrorAB;
static signal* YCriterion;
static signal* YAccCons;
static signal* YAccConsLP;
static signal* YAccConsWashout;
static signal* YAccOutput;
static signal* YVelCons;
static signal* YVelOutput;
static signal* YPosCons;
static signal* YPosOutput;

// -----
// Déclaration des signaux
// axe R : Roll (tangage)
// -----

static signal* Y2RAccFiltered;
static signal* RTiltAngle;
```

```

static signal* RTiltAngleSat;
static signal* RAngleInput;
static signal* RAngleFiltered;
static signal* RAngleFilteredSat;
static signal* RAngleCons;
static signal* RAngleOutput;
static signal* RVelCons;
static signal* RVelOutput;
static signal* RAccCons;
static signal* RAccOutput;

// -----
// Déclaration des signaux
// axe H : Heading (lacet)
// -----

static signal* HAccInput;
static signal* HAccInputSat;
static signal* HAccFiltered;
static signal* HError;
static signal* HErrorFiltered;
static signal* HErrorFiltered2;
static signal* HAccFilteredAB;
static signal* HErrorAB;
static signal* HCriterion;
static signal* HAccCons;
static signal* HAccConsLP;
static signal* HAccConsWashout;
static signal* HAccOutput;
static signal* HVelCons;
static signal* HVelOutput;
static signal* HAngleCons;
static signal* HAngleOutput;

// -----
// Déclaration des signaux
// axe Z : vertical
// -----

static signal* ZAccInput;
static signal* ZAccInputInt;
static signal* ZVit;
static signal* ZVitInt;
static signal* ZVitIntFiltered;
static signal* ZVitIntFiltered2;
static signal* ZPosCons;
static signal* ZPosOutput;
static signal* ZVelCons;
static signal* ZVelOutput;
static signal* ZAccCons;
static signal* ZAccOutput;

//*****
//*****
// externInit
//*****
//*****
void externInit ( void )
{
    count = 0.0;
    countIsStop = 0.0;

```

```

stopGain = 1.0;

// -----
// Ouverture du fichier d'écriture
// -----
myFile.open("Results.txt");
if (myFile.is_open())
{
    std::cout<<"File open"<<std::endl;
}
else
{
    std::cout<<"Problem while opening file"<<std::endl;
}

// -----
// Initialisation des signaux
// -----
XAccInput          = new signal(PeriodeEchantillonnage);
XVelInput          = new signal(PeriodeEchantillonnage);
XAccFiltered       = new signal(PeriodeEchantillonnage);
XError             = new signal(PeriodeEchantillonnage);
XErrorFiltered     = new signal(PeriodeEchantillonnage);
XErrorFiltered2    = new signal(PeriodeEchantillonnage);
XAccFilteredAB     = new signal(PeriodeEchantillonnage);
XErrorAB           = new signal(PeriodeEchantillonnage);
XCriterion         = new signal(PeriodeEchantillonnage);
XAccCons           = new signal(PeriodeEchantillonnage);
XAccConsLP         = new signal(PeriodeEchantillonnage);
XAccConsWashout    = new signal(PeriodeEchantillonnage);
XAccOutput         = new signal(PeriodeEchantillonnage);
XVelCons           = new signal(PeriodeEchantillonnage);
XVelOutput         = new signal(PeriodeEchantillonnage);
XPosCons           = new signal(PeriodeEchantillonnage);
XPosOutput         = new signal(PeriodeEchantillonnage);

X2PAccFiltered    = new signal(PeriodeEchantillonnage);
PTiltAngle        = new signal(PeriodeEchantillonnage);
PTiltAngleSat     = new signal(PeriodeEchantillonnage);
PAngleInput       = new signal(PeriodeEchantillonnage);
PAngleFiltered    = new signal(PeriodeEchantillonnage);
PAngleFilteredSat = new signal(PeriodeEchantillonnage);
PAngleCons        = new signal(PeriodeEchantillonnage);
PAngleOutput      = new signal(PeriodeEchantillonnage);
PVelCons          = new signal(PeriodeEchantillonnage);
PVelOutput        = new signal(PeriodeEchantillonnage);
PAccCons          = new signal(PeriodeEchantillonnage);
PAccOutput        = new signal(PeriodeEchantillonnage);

YAccInput         = new signal(PeriodeEchantillonnage);
YVelInput         = new signal(PeriodeEchantillonnage);
YAccInputSat      = new signal(PeriodeEchantillonnage);
YAccFiltered      = new signal(PeriodeEchantillonnage);
YError            = new signal(PeriodeEchantillonnage);
YErrorFiltered    = new signal(PeriodeEchantillonnage);
YErrorFiltered2   = new signal(PeriodeEchantillonnage);
YAccFilteredAB    = new signal(PeriodeEchantillonnage);
YErrorAB          = new signal(PeriodeEchantillonnage);
YCriterion        = new signal(PeriodeEchantillonnage);
YAccCons          = new signal(PeriodeEchantillonnage);
YAccConsLP        = new signal(PeriodeEchantillonnage);

```

```

YAccConsWashout      = new signal(PeriodeEchantillonnage);
YAccOutput           = new signal(PeriodeEchantillonnage);
YVelCons             = new signal(PeriodeEchantillonnage);
YVelOutput           = new signal(PeriodeEchantillonnage);
YPosCons             = new signal(PeriodeEchantillonnage);
YPosOutput           = new signal(PeriodeEchantillonnage);

Y2RAccFiltered       = new signal(PeriodeEchantillonnage);
RTiltAngle           = new signal(PeriodeEchantillonnage);
RTiltAngleSat        = new signal(PeriodeEchantillonnage);
RAngleInput          = new signal(PeriodeEchantillonnage);
RAngleFiltered       = new signal(PeriodeEchantillonnage);
RAngleFilteredSat    = new signal(PeriodeEchantillonnage);
RAngleCons           = new signal(PeriodeEchantillonnage);
RAngleOutput         = new signal(PeriodeEchantillonnage);
RVelCons             = new signal(PeriodeEchantillonnage);
RVelOutput           = new signal(PeriodeEchantillonnage);
RAccCons             = new signal(PeriodeEchantillonnage);
RAccOutput           = new signal(PeriodeEchantillonnage);

HAccInput            = new signal(PeriodeEchantillonnage);
HAccInputSat         = new signal(PeriodeEchantillonnage);
HAccFiltered         = new signal(PeriodeEchantillonnage);
HError               = new signal(PeriodeEchantillonnage);
HErrorFiltered       = new signal(PeriodeEchantillonnage);
HErrorFiltered2      = new signal(PeriodeEchantillonnage);
HAccFilteredAB       = new signal(PeriodeEchantillonnage);
HErrorAB             = new signal(PeriodeEchantillonnage);
HCriterion           = new signal(PeriodeEchantillonnage);
HAccCons             = new signal(PeriodeEchantillonnage);
HAccConsLP           = new signal(PeriodeEchantillonnage);
HAccConsWashout      = new signal(PeriodeEchantillonnage);
HAccOutput           = new signal(PeriodeEchantillonnage);
HVelCons             = new signal(PeriodeEchantillonnage);
HVelOutput           = new signal(PeriodeEchantillonnage);
HAngleCons           = new signal(PeriodeEchantillonnage);
HAngleOutput         = new signal(PeriodeEchantillonnage);

ZAccInput            = new signal(PeriodeEchantillonnage);
ZAccInputInt         = new signal(PeriodeEchantillonnage);
ZVit                 = new signal(PeriodeEchantillonnage);
ZVitInt              = new signal(PeriodeEchantillonnage);
ZVitIntFiltered      = new signal(PeriodeEchantillonnage);
ZVitIntFiltered2     = new signal(PeriodeEchantillonnage);
ZPosCons             = new signal(PeriodeEchantillonnage);
ZPosOutput           = new signal(PeriodeEchantillonnage);
ZVelCons             = new signal(PeriodeEchantillonnage);
ZVelOutput           = new signal(PeriodeEchantillonnage);
ZAccCons             = new signal(PeriodeEchantillonnage);
ZAccOutput           = new signal(PeriodeEchantillonnage);
}

//*****
//*****
// computeX
//*****
//*****

void computeX (void)
{
    XVelInput->integer(XAccInput);
}

```

```

XAccFiltered->HP1(XAccInput,1.0,2.0);
XError->subtract(XAccInput,XAccFiltered);
XErrorFiltered->LP1(XError,2.0,2.0);
double lTemp = XErrorFiltered->value(0);
XErrorFiltered2->update(exp(-10.0*lTemp*lTemp));
XAccFilteredAB->update(XAccFiltered->value(0) * XErrorFiltered2-
>value(0));
XErrorAB->subtract(XAccInput,XAccFilteredAB);
double lTemp2 = XErrorAB->value(0);
double lTemp3 = XErrorFiltered->value(0);
double lCriterion = tanh( 2 * ( abs(lTemp2) - abs(lTemp3) ) );
lCriterion = (lCriterion + 1) / 2;
XCriterion->update( lCriterion );
// acc_cons = (1-criterion) x acc_filtered_ab + criterion x
acc_filtered
XAccCons->update( (1-lCriterion)*XAccFilteredAB->value(0) +
lCriterion*XAccFiltered->value(0) );
XAccConsLP->LP1(XAccCons, 1.0, 0.1);
XAccConsWashout->HP2(XAccConsLP,1.0,1.0,0.5);
// Saturation pour respecter les limites en accélération de la PF
XAccOutput->saturate(XAccConsWashout,-6.0,6.0);
XVelCons->integer(XAccOutput);
// Saturation pour respecter les limites en vitesse de la PF
XVelOutput->saturate(XVelCons,-0.5,0.5);
XPosCons->integer(XVelOutput);
// Saturation pour respecter les limites en position de la PF
XPosOutput->saturate(XPosCons, -0.23, 0.25);
}

//*****
//*****
// computeP
//*****
//*****

void computeP (void)
{
X2PAccFiltered->LP2(XAccInput,0.25,0.7,0.3);
PTiltAngle->update(-asin(X2PAccFiltered->value(0) / 9.81));
PTiltAngleSat->saturate(PTiltAngle, -0.24, 0.24);
//PAngleFiltered->LP1(PAngleInput, 1.0, 1.0);
//PAngleFilteredSat->saturate(PAngleFiltered, -0.10, 0.10);
PAngleFilteredSat->saturate(PAngleInput, -0.10, 0.10);
PAngleCons->update(- PTiltAngleSat->value(0) - PAngleFilteredSat-
>value(0));
PAngleOutput->saturate(PAngleCons, -0.34, 0.34);
PVelCons->derivate(PAngleOutput);
PVelOutput->saturate(PVelCons, -0.52, 0.52);
PAccCons->derivate(PVelOutput);
PAccOutput->saturate(PAccCons, -8.5, 8.5);
}

//*****
//*****
// computeY
//*****
//*****

void computeY (void)
{
YAccInputSat->saturate(YAccInput,-8.0,8.0);

```



```

    YVelInput->integer(YAccInput);
    YAccFiltered->HP1(YAccInputSat,1.0,2.0);
    YError->subtract(YAccInputSat,YAccFiltered);
    YErrorFiltered->LP1(YError,2.0,2.0);
    double lTemp = YErrorFiltered->value(0);
    YErrorFiltered2->update(exp(-10.0*lTemp*lTemp));
    YAccFilteredAB->update(YAccFiltered->value(0) * YErrorFiltered2-
>value(0));
    YErrorAB->subtract(YAccInput,YAccFilteredAB);
    double lTemp2 = YErrorAB->value(0);
    double lTemp3 = YErrorFiltered->value(0);
    double lCriterion = tanh( 2 * ( abs(lTemp2) - abs(lTemp3) ) );
    lCriterion = (lCriterion + 1) / 2;
    YCriterion->update( lCriterion );
    // acc_cons = (1-criterion) x acc_filtered_ab + criterion x
acc_filtered
    YAccCons->update( (1-lCriterion)*YAccFilteredAB->value(0) +
lCriterion*YAccFiltered->value(0) );
    YAccConsLP->LP1(YAccCons, 1.0, 0.1);
    YAccConsWashout->HP2(YAccConsLP,1.0,1.0,0.5);
    // Saturation pour respecter les limites en accélération de la PF
    YAccOutput->saturate(YAccConsWashout,-6.0,6.0);
    YVelCons->integer(YAccOutput);
    // Saturation pour respecter les limites en vitesse de la PF
    YVelOutput->saturate(YVelCons,-0.5,0.5);
    YPosCons->integer(YVelOutput);
    // Saturation pour respecter les limites en position de la PF
    YPosOutput->saturate(YPosCons, -0.23, 0.25);
}

//*****
//*****
//   computer
//*****
//*****

void computer (void)
{
    Y2RAccFiltered->LP2(YAccInputSat,0.25,0.7,0.3);
    RTiltAngle->update(-asin(Y2RAccFiltered->value(0) / 9.81));
    RTiltAngleSat->saturate(RTiltAngle, -0.24, 0.24);
    //RAngleFiltered->LP1(RAngleInput, 1.0, 1.0);
    //RAngleFilteredSat->saturate(RAngleFiltered, -0.10, 0.10);
    RAngleFilteredSat->saturate(RAngleInput, -0.10, 0.10);
    RAngleCons->update( - RTiltAngleSat->value(0) + RAngleFilteredSat-
>value(0) );
    RAngleOutput->saturate(RAngleCons, -0.34, 0.34);
    RVelCons->derivate(RAngleOutput);
    RVelOutput->saturate(RVelCons, -0.52, 0.52);
    RAccCons->derivate(RVelOutput);
    RAccOutput->saturate(RAccCons, -8.5, 8.5);
}

//*****
//*****
//   computeH
//*****
//*****

void computeH (void)
{

```

```

HAccInputSat->saturate (HAccInput, -8.0, 8.0);
HAccFiltered->HP1 (HAccInputSat, 1.0, 2.0);
HError->subtract (HAccInputSat, HAccFiltered);
HErrorFiltered->LP1 (HError, 2.0, 2.0);
double lTemp = HErrorFiltered->value (0);
HErrorFiltered2->update (exp (-10.0*lTemp*lTemp));
HAccFilteredAB->update (HAccFiltered->value (0) * HErrorFiltered2-
>value (0));
HErrorAB->subtract (HAccInput, HAccFilteredAB);
double lTemp2 = HErrorAB->value (0);
double lTemp3 = HErrorFiltered->value (0);
double lCriterion = tanh ( 2 * ( abs (lTemp2) - abs (lTemp3) ) );
lCriterion = (lCriterion + 1) / 2;
HCriterion->update ( lCriterion );
// acc_cons = (1-criterion) x acc_filtered_ab + criterion x
acc_filtered
HAccCons->update ( (1-lCriterion)*HAccFilteredAB->value (0) +
lCriterion*HAccFiltered->value (0) );
HAccConsLP->LP1 (HAccCons, 1.0, 0.1);
HAccConsWashout->HP2 (HAccConsLP, -1.0, 1.0, 2.0);
// Saturation pour respecter les limites en accélération de la PF
HAccOutput->saturate (HAccConsWashout, -6.981, 6.981);
HVelCons->integer (HAccOutput);
// Saturation pour respecter les limites en vitesse de la PF
HVelOutput->saturate (HVelCons, -0.698, 0.698);
HAngleCons->integer (HVelOutput);
// Saturation pour respecter les limites en orientation de la PF
HAngleOutput->saturate (HAngleCons, -0.34, 0.34);
}

//*****
//*****
// computeZ
//*****
//*****

void computeZ (void)
{
    ZAccInputInt->integer (ZAccInput);
    ZVit->HP1 (ZAccInputInt, 1.0, 0.1);
    ZVitInt->integer (ZVit);
    ZVitIntFiltered->HP1 (ZVitInt, 1.0, 0.1);
    ZVitIntFiltered2->HP1 (ZVitIntFiltered, 0.4, 3.0);
    // On rajoute une constante (0.24) pour que la position de base
    // soit au milieu verticalement
    ZPosCons->update ( ZVitIntFiltered2->value (0) + 0.24 );
    // Saturation pour respecter les limites en position de la PF
    ZPosOutput->saturate ( ZPosCons, 0.09, 0.39 );
    ZVelCons->derivate (ZPosOutput);
    // Saturation pour respecter les limites en vitesse de la PF
    ZVelOutput->saturate ( ZVelCons, -0.3, 0.3 );
    ZAccCons->derivate ( ZVelOutput );
    // Saturation pour respecter les limites en accélération de la PF
    ZAccOutput->saturate ( ZAccCons, -5.0, 5.0 );
}

//*****
//*****
// endSim
//*****
//*****

```

```

void endSim (double acc)
{
    acc = abs(acc);

    // On incrémente la variable countIsStop si la vitesse du véhicule
    est nulle
    // Si la vitesse est non nulle, on remet la variable countIsStop à 0
    if (acc < 0.1)
    {
        countIsStop += PeriodeEchantillonnage ;
    }
    else
    {
        countIsStop = 0.0;
    }

    // Si le véhicule est arrêté depuis plus de 20 secondes, on
    décrémente la
    // variable stopGain pour faire descendre la plate-forme
    if (countIsStop > 25.0)
    {
        stopGain -= (0.1*PeriodeEchantillonnage);
    }
    // Si le véhicule n'est plus arrêté, on incrémente stopGain pour
    faire remonter
    // la plate-forme si jamais elle était descendue, sauf si elle est
    arrivée en
    // position basse
    else
    {
        if (stopGain > 0.0)
        {
            stopGain += (0.1*PeriodeEchantillonnage);
        }
    }

    // stopGain doit être compris entre 0.0 et 1.0 !
    if (stopGain > 1.0)
    {
        stopGain = 1.0;
    }
    if (stopGain < 0.0)
    {
        stopGain = 0.0;
    }
}

//*****
//*****
// externStep
//*****
//*****
void externStep (double posvitacc[18], double retourposvitacc[18])
{
    //posvitacc[0] --> X position
    //posvitacc[1] --> Y position
    //posvitacc[2] --> Z position
    //posvitacc[3] --> H angle
    //posvitacc[4] --> P angle
    //posvitacc[5] --> R angle

```

```

//posvitacc[6]  --> X velocity
//posvitacc[7]  --> Y velocity
//posvitacc[8]  --> Z velocity
//posvitacc[9]  --> H velocity
//posvitacc[10] --> P velocity
//posvitacc[11] --> R velocity
//posvitacc[12] --> X acceleration
//posvitacc[13] --> Y acceleration
//posvitacc[14] --> Z acceleration
//posvitacc[15] --> H acceleration
//posvitacc[16] --> P acceleration
//posvitacc[17] --> R acceleration

// -----
// Mise à jour des signaux d'entrée
// -----

double RampTime = 10.0 ;
double coeff = count / (RampTime * 60);
if (coeff > 1)
{
    coeff = 1;
}
XAccInput->update (coeff * posvitacc[12]);
PAngleInput->update (coeff * posvitacc[4]);
YAccInput->update (coeff * posvitacc[13]);
RAngleInput->update (coeff * posvitacc[5]);
HAccInput->update (coeff * posvitacc[15]);
ZAccInput->update (coeff * posvitacc[14]);

// -----
// Calcul des signaux de sortie
// -----

computeX();
computeP();
computeY();
computeR();
computeH();
computeZ();
//endSim(XAccInput->value(0));

// -----
// Mise à jour de la variable de sortie
// -----

retourposvitacc[0] = XPosOutput->value(0) * 0.2; // X position
retourposvitacc[1] = YPosOutput->value(0) * 0.2; // Y position
retourposvitacc[2] = 0.22 ; // Z position
retourposvitacc[3] = 0 ; // HAngleOutput->value(0) ; // H angle
retourposvitacc[4] = PAngleOutput->value(0) ; // P angle
retourposvitacc[5] = RAngleOutput->value(0) ; // R angle
retourposvitacc[6] = 0 ; // X velocity
retourposvitacc[7] = 0 ; // Y velocity
retourposvitacc[8] = 0 ; // Z velocity
retourposvitacc[9] = 0 ; // H velocity
retourposvitacc[10] = 0 ; // P velocity
retourposvitacc[11] = 0 ; // R velocity
retourposvitacc[12] = 0 ; // X acceleration
retourposvitacc[13] = 0 ; // Y acceleration
retourposvitacc[14] = 0 ; // Z acceleration

```

```

retourposvitacc[15]    = 0    ; // H acceleration
retourposvitacc[16]    = 0    ; // P acceleration
retourposvitacc[17]    = 0    ; // R acceleration

for (int i=0; i<18; i++)
{
    retourposvitacc[i] = retourposvitacc[i] * stopGain;
}

// -----
// Ecriture des résultats
// -----

// myFile << count * PeriodeEchantillonnage << ";";

/*myFile << XAccInput->value(0) << ";";
myFile << XAccFiltered->value(0) << ";";
myFile << XError->value(0) << ";";
myFile << XErrorFiltered->value(0) << ";";
myFile << XErrorFiltered2->value(0) << ";";
myFile << XAccFilteredAB->value(0) << ";";
myFile << XErrorAB->value(0) << ";";
myFile << XCriterion->value(0) << ";";
myFile << XAccCons->value(0) << ";";
myFile << XAccConsWashout->value(0) << ";";
myFile << XAccOutput->value(0) << ";";
myFile << XVelCons->value(0) << ";";
myFile << XVelOutput->value(0) << ";";
myFile << XPosCons->value(0) << ";";
myFile << XPosOutput->value(0) << ";";

myFile << PAngleInput->value(0) << ";";
myFile << X2PAccFiltered->value(0) << ";";
myFile << PTiltAngle->value(0) << ";";
myFile << PTiltAngleSat->value(0) << ";";
myFile << PAngleFiltered->value(0) << ";";
myFile << PAngleFilteredSat->value(0) << ";";
myFile << PAngleCons->value(0) << ";";
myFile << PAngleOutput->value(0) << ";";
myFile << PVelCons->value(0) << ";";
myFile << PVelOutput->value(0) << ";";
myFile << PAccCons->value(0) << ";";
myFile << PAccOutput->value(0) << ";";

myFile << YAccInputSat->value(0) << ";";
myFile << YAccFiltered->value(0) << ";";
myFile << YError->value(0) << ";";
myFile << YErrorFiltered->value(0) << ";";
myFile << YErrorFiltered2->value(0) << ";";
myFile << YAccFilteredAB->value(0) << ";";
myFile << YErrorAB->value(0) << ";";
myFile << YCriterion->value(0) << ";";
myFile << YAccCons->value(0) << ";";
myFile << YAccConsWashout->value(0) << ";";
myFile << YAccOutput->value(0) << ";";
myFile << YVelCons->value(0) << ";";
myFile << YVelOutput->value(0) << ";";
myFile << YPosCons->value(0) << ";";
myFile << YPosOutput->value(0) << ";";

myFile << RAngleInput->value(0) << ";";

```

```

myFile << Y2RAccFiltered->value(0) << ";";
myFile << RTiltAngle->value(0) << ";";
myFile << RTiltAngleSat->value(0) << ";";
myFile << RAngleFiltered->value(0) << ";";
myFile << RAngleFilteredSat->value(0) << ";";
myFile << RAngleCons->value(0) << ";";
myFile << RAngleOutput->value(0) << ";";
myFile << RVelCons->value(0) << ";";
myFile << RVelOutput->value(0) << ";";
myFile << RAccCons->value(0) << ";";
myFile << RAccOutput->value(0) << ";";

myFile << HAccInput->value(0) << ";";
myFile << HAccFiltered->value(0) << ";";
myFile << HError->value(0) << ";";
myFile << HErrorFiltered->value(0) << ";";
myFile << HErrorFiltered2->value(0) << ";";
myFile << HAccFilteredAB->value(0) << ";";
myFile << HErrorAB->value(0) << ";";
myFile << HCriterion->value(0) << ";";
myFile << HAccCons->value(0) << ";";
myFile << HAccConsWashout->value(0) << ";";
myFile << HAccOutput->value(0) << ";";
myFile << HVelCons->value(0) << ";";
myFile << HVelOutput->value(0) << ";";
myFile << HAngleCons->value(0) << ";";
myFile << HAngleOutput->value(0) << ";";

myFile << ZAccInput->value(0) << ";";
myFile << ZAccInputInt->value(0) << ";";
myFile << ZVit->value(0) << ";";
myFile << ZVitInt->value(0) << ";";
myFile << ZVitIntFiltered->value(0) << ";";
myFile << ZVitIntFiltered2->value(0) << ";";
myFile << ZPosCons->value(0) << ";";
myFile << ZPosOutput->value(0) << ";";
myFile << ZVelCons->value(0) << ";";
myFile << ZVelOutput->value(0) << ";";
myFile << ZAccCons->value(0) << ";";
myFile << ZAccOutput->value(0) << "\n";*/

//myFile << XAccInput->value(0) << ";";
//myFile << countIsStop << ";";
//myFile << stopGain << "\n";

// -----
// Comptage
// -----
count++;

// -----
// Si on veut être en statique mais calculer
// tout de même les commandes
// -----

/*
for (int i=0; i<18; i++)
{
    retourposvitacc[i] = 0.0;
}
*/

```

 }

B. C++ code pour algorithme hybride

B1. Fichiers C++

externHybridCueing.cpp

```

#include "externHybridCueing.h"
#include "signalHybrid.h"
#include <iostream>
#include "windows.h"
#include "math.h"
#include <fstream>

//*****
//*****
//  VARIABLES STATIQUES
//*****
//*****

static double count;
static double countIsStop;
static double stopGain;

// -----
// Fichier d'écriture des résultats
// -----
static std::ofstream myFile;

// -----
// Période d'échantillonnage des signaux
// correspondant à la fréquence de
// communication avec la plate-forme
// -----
static double PeriodeEchantillonnage = 1/60.0;

```

```
// -----  
// Déclaration des signaux  
// axe X : longitudinal  
// -----  
static signal* XAccInput;  
static signal* XAccInput1;  
static signal* XAccInput2;  
static signal* XVelInput;  
static signal* XAccFiltered;  
static signal* XError;  
static signal* XErrorFiltered;  
static signal* XErrorFiltered2;  
static signal* XAccFilteredAB;  
static signal* XErrorAB;  
static signal* XCriterion;  
static signal* XAccCons;  
static signal* XAccCons1;  
static signal* XAccConsLP;  
static signal* XAccConsWashout;  
static signal* XAccOutput;  
static signal* XVelCons;  
static signal* XVelOutput;  
static signal* XVelFiltered;  
static signal* XPosCons;  
static signal* XPosOutput;  
static signal* XPosFiltered;  
  
// -----  
// Déclaration des signaux  
// axe P : Pitch (tangage)  
// -----  
  
static signal* X2PAccFiltered;  
static signal* PTiltAngle;  
static signal* PTiltAngleSat;  
static signal* PAngleInput;  
static signal* PAngleFiltered;  
static signal* PAngleFilteredSat;  
static signal* PAngleCons;  
static signal* PAngleCons1;  
static signal* PAngleCons2;  
static signal* PAngleOutput;  
static signal* PVelCons;  
static signal* PVelCons1;  
static signal* PVelCons2;  
static signal* PVelOutput;  
static signal* PAccCons;  
static signal* PAccOutput;
```



```
// -----  
// Déclaration des signaux  
// axe Y : latéral  
// -----  
  
static signal* YAccInput;  
static signal* YAccInput1;  
static signal* YAccInput2;  
static signal* YVelInput;  
static signal* YAccInputSat;  
static signal* YAccFiltered;  
static signal* YError;  
static signal* YErrorFiltered;  
static signal* YErrorFiltered2;  
static signal* YAccFilteredAB;  
static signal* YErrorAB;  
static signal* YCriterion;  
static signal* YAccCons;  
static signal* YAccCons1;  
static signal* YAccConsLP;  
static signal* YAccConsWashout;  
static signal* YAccOutput;  
static signal* YVelCons;  
static signal* YVelOutput;  
static signal* YVelFiltered;  
static signal* YPosCons;  
static signal* YPosOutput;  
static signal* YPosFiltered;  
  
// -----  
// Déclaration des signaux  
// axe R : Roll (roulis)  
// -----  
  
static signal* Y2RAccFiltered;  
static signal* RTiltAngle;  
static signal* RTiltAngleSat;  
static signal* RAngleInput;  
static signal* RAngleFiltered;  
static signal* RAngleFilteredSat;  
static signal* RAngleCons;  
static signal* RAngleCons1;  
static signal* RAngleCons2;  
static signal* RAngleOutput;  
static signal* RVelCons;  
static signal* RVelCons1;  
static signal* RVelCons2;  
static signal* RVelOutput;  
static signal* RAccCons;
```

```
static signal* RAccOutput;

// -----
// Déclaration des signaux
// axe H : Heading (lacet)
// -----

static signal* HAccInput;
static signal* HAccInputSat;
static signal* HAccFiltered;
static signal* HError;
static signal* HErrorFiltered;
static signal* HErrorFiltered2;
static signal* HAccFilteredAB;
static signal* HErrorAB;
static signal* HCriterion;
static signal* HAccCons;
static signal* HAccConsLP;
static signal* HAccConsWashout;
static signal* HAccOutput;
static signal* HVelCons;
static signal* HVelOutput;
static signal* HAngleCons;
static signal* HAngleOutput;

// -----
// Déclaration des signaux
// axe Z : vertical
// -----

static signal* ZAccInput;
static signal* ZAccInputInt;
static signal* ZVit;
static signal* ZVitInt;
static signal* ZVitIntFiltered;
static signal* ZVitIntFiltered2;
static signal* ZPosCons;
static signal* ZPosOutput;
static signal* ZPosFiltered;
static signal* ZVelCons;
static signal* ZVelOutput;
static signal* ZVelInput;
static signal* ZVelFiltered;
static signal* ZAccCons;
static signal* ZAccOutput;
static signal* ZAccFiltered;

//*****
//*****
```

```

//  externInit
//*****
//*****
void externInit ( void )
{
    count = 0.0;
    countIsStop = 0.0;
    stopGain = 1.0;

    // -----
    // Ouverture du fichier d'écriture
    // -----
    myFile.open("Results.txt");
    if (myFile.is_open())
    {
        std::cout<<"File open"<<std::endl;
    }
    else
    {
        std::cout<<"Problem while opening file"<<std::endl;
    }

    // -----
    // Initialisation des signaux
    // -----
    XAccInput          = new signal(PeriodeEchantillonnage);
    XAccInput1         = new signal(PeriodeEchantillonnage);
    XAccInput2         = new signal(PeriodeEchantillonnage);
    XVelInput          = new signal(PeriodeEchantillonnage);
    XAccFiltered       = new signal(PeriodeEchantillonnage);
    XError             = new signal(PeriodeEchantillonnage);
    XErrorFiltered     = new signal(PeriodeEchantillonnage);
    XErrorFiltered2    = new signal(PeriodeEchantillonnage);
    XAccFilteredAB     = new signal(PeriodeEchantillonnage);
    XErrorAB           = new signal(PeriodeEchantillonnage);
    XCriterion         = new signal(PeriodeEchantillonnage);
    XAccCons           = new signal(PeriodeEchantillonnage);
    XAccCons1          = new signal(PeriodeEchantillonnage);
    XAccConsLP         = new signal(PeriodeEchantillonnage);
    XAccConsWashout    = new signal(PeriodeEchantillonnage);
    XAccOutput         = new signal(PeriodeEchantillonnage);
    XVelCons           = new signal(PeriodeEchantillonnage);
    XVelOutput         = new signal(PeriodeEchantillonnage);
    XVelFiltered       = new signal(PeriodeEchantillonnage);
    XPosCons           = new signal(PeriodeEchantillonnage);
    XPosOutput         = new signal(PeriodeEchantillonnage);
    XPosFiltered       = new signal(PeriodeEchantillonnage);

    X2PAccFiltered     = new signal(PeriodeEchantillonnage);

```

```
PTiltAngle           = new signal(PeriodeEchantillonnage);
PTiltAngleSat        = new signal(PeriodeEchantillonnage);
PAngleInput          = new signal(PeriodeEchantillonnage);
PAngleFiltered       = new signal(PeriodeEchantillonnage);
PAngleFilteredSat    = new signal(PeriodeEchantillonnage);
PAngleCons           = new signal(PeriodeEchantillonnage);
PAngleCons1          = new signal(PeriodeEchantillonnage);
PAngleCons2          = new signal(PeriodeEchantillonnage);
PAngleOutput         = new signal(PeriodeEchantillonnage);
PVelCons             = new signal(PeriodeEchantillonnage);
PVelOutput           = new signal(PeriodeEchantillonnage);
PAccCons             = new signal(PeriodeEchantillonnage);
PAccOutput           = new signal(PeriodeEchantillonnage);

YAccInput            = new signal(PeriodeEchantillonnage);
YVelInput            = new signal(PeriodeEchantillonnage);
YAccInputSat         = new signal(PeriodeEchantillonnage);
YAccFiltered         = new signal(PeriodeEchantillonnage);
YError               = new signal(PeriodeEchantillonnage);
YErrorFiltered       = new signal(PeriodeEchantillonnage);
YErrorFiltered2      = new signal(PeriodeEchantillonnage);
YAccFilteredAB       = new signal(PeriodeEchantillonnage);
YErrorAB             = new signal(PeriodeEchantillonnage);
YCriterion           = new signal(PeriodeEchantillonnage);
YAccCons             = new signal(PeriodeEchantillonnage);
YAccConsLP           = new signal(PeriodeEchantillonnage);
YAccConsWashout      = new signal(PeriodeEchantillonnage);
YAccOutput           = new signal(PeriodeEchantillonnage);
YVelCons             = new signal(PeriodeEchantillonnage);
YVelOutput           = new signal(PeriodeEchantillonnage);
YVelFiltered         = new signal(PeriodeEchantillonnage);
YPosCons             = new signal(PeriodeEchantillonnage);
YPosOutput           = new signal(PeriodeEchantillonnage);
YPosFiltered         = new signal(PeriodeEchantillonnage);

Y2RAccFiltered       = new signal(PeriodeEchantillonnage);
RTiltAngle           = new signal(PeriodeEchantillonnage);
RTiltAngleSat        = new signal(PeriodeEchantillonnage);
RAngleInput          = new signal(PeriodeEchantillonnage);
RAngleFiltered       = new signal(PeriodeEchantillonnage);
RAngleFilteredSat    = new signal(PeriodeEchantillonnage);
RAngleCons           = new signal(PeriodeEchantillonnage);
RAngleOutput         = new signal(PeriodeEchantillonnage);
RVelCons             = new signal(PeriodeEchantillonnage);
RVelOutput           = new signal(PeriodeEchantillonnage);
RAccCons             = new signal(PeriodeEchantillonnage);
RAccOutput           = new signal(PeriodeEchantillonnage);

HAccInput            = new signal(PeriodeEchantillonnage);
```

```

HAccInputSat           = new signal(PeriodeEchantillonnage);
HAccFiltered           = new signal(PeriodeEchantillonnage);
HError                 = new signal(PeriodeEchantillonnage);
HErrorFiltered        = new signal(PeriodeEchantillonnage);
HErrorFiltered2       = new signal(PeriodeEchantillonnage);
HAccFilteredAB        = new signal(PeriodeEchantillonnage);
HErrorAB              = new signal(PeriodeEchantillonnage);
HCriterion            = new signal(PeriodeEchantillonnage);
HAccCons              = new signal(PeriodeEchantillonnage);
HAccConsLP           = new signal(PeriodeEchantillonnage);
HAccConsWashout       = new signal(PeriodeEchantillonnage);
HAccOutput            = new signal(PeriodeEchantillonnage);
HVelCons              = new signal(PeriodeEchantillonnage);
HVelOutput            = new signal(PeriodeEchantillonnage);
HAngleCons           = new signal(PeriodeEchantillonnage);
HAngleOutput         = new signal(PeriodeEchantillonnage);

ZAccInput             = new signal(PeriodeEchantillonnage);
ZAccInputInt          = new signal(PeriodeEchantillonnage);
ZVit                 = new signal(PeriodeEchantillonnage);
ZVitInt              = new signal(PeriodeEchantillonnage);
ZVitIntFiltered      = new signal(PeriodeEchantillonnage);
ZVitIntFiltered2    = new signal(PeriodeEchantillonnage);
ZPosCons             = new signal(PeriodeEchantillonnage);
ZPosOutput           = new signal(PeriodeEchantillonnage);
ZPosFiltered         = new signal(PeriodeEchantillonnage);
ZVelCons             = new signal(PeriodeEchantillonnage);
ZVelOutput           = new signal(PeriodeEchantillonnage);
ZVelFiltered         = new signal(PeriodeEchantillonnage);
ZVelInput            = new signal(PeriodeEchantillonnage);
ZAccCons             = new signal(PeriodeEchantillonnage);
ZAccOutput           = new signal(PeriodeEchantillonnage);
ZAccFiltered         = new signal(PeriodeEchantillonnage);
}

//*****
//*****
//  computeX
//*****
//*****

void computeX (void)
{
    XVelInput->integer(XAccInput);
    XAccFiltered->HP2(XAccInput,1.0,1.0,0.4);
    XVelFiltered->LP2vitesse(XAccInput,1.0,1.0,0.4);
    XPosFiltered->LP2position(XAccInput,1.0,1.0,0.4);
    //XError->subtract(XAccInput,XAccFiltered);
    //XErrorFiltered->LP1(XError,2.0,2.0);
}

```

```

        //double lTemp = XErrorFiltered->value(0);
        //XErrorFiltered2->update(exp(-10.0*lTemp*lTemp));
        //XAccFilteredAB->update(XAccFiltered->value(0) * XErrorFiltered2-
>value(0));
        //XErrorAB->subtract(XAccInput,XAccFilteredAB);
        //double lTemp2 = XErrorAB->value(0);
        //double lTemp3 = XErrorFiltered->value(0);
        //double lCriterion = tanh( 2 * ( abs(lTemp2) - abs(lTemp3) ) );
        //lCriterion = (lCriterion + 1) / 2;
        //XCriterion->update( lCriterion );
        XVelCons->integer(XAccCons);
        XPosCons->integer(XVelCons);
        double lCriterion2 = (-2 * (XVelFiltered->value(0)) * (XVelCons-
>value(0)) + (XAccFiltered->value(0)) * (XAccInput->value(0)-XAccCons-
>value(0)-(20 * XPosFiltered->value(0) * (XPosCons->value(0)) * 0.5)));
        double lCriterion3 = (lCriterion2)- 5.0265*XVelCons->value(0)-
6.3167*XPosCons->value(0);

        // acc_cons = (1-criterion) x acc_filtered_ab + criterion x
acc_filtered
        XAccCons->update( lCriterion3);
        //XAccConsLP->LP1(XAccCons, 1.0, 0.1);
        //XAccConsWashout->HP2(XAccConsLP,1.0,1.0,0.5);
        // Saturation pour respecter les limites en accélération de la PF
        XAccOutput->saturate(XAccCons,-6.0*9.81,6.0*9.81);
        XVelCons->integer(XAccOutput);
        // Saturation pour respecter les limites en vitesse de la PF
        XVelOutput->saturate(XVelCons,-0.5,0.5);
        XPosCons->integer(XVelOutput);
        // Saturation pour respecter les limites en position de la PF
        XPosOutput->saturate(XPosCons, -0.241, 0.259);
}

//*****
//*****
// computeP
//*****
//*****

void computeP (void)
{
    X2PAccFiltered->LP2vitesse(XAccInput,0.25,0.7,0.3);
    PTiltAngle->update(-asin(X2PAccFiltered->value(0) / 9.81));
    PTiltAngleSat->saturate(PTiltAngle, -0.24, 0.24);
    //PAngleFiltered->LP1(PAngleInput, 1.0, 1.0);
    //PAngleFilteredSat->saturate(PAngleFiltered, -0.10, 0.10);
    PAngleFilteredSat->saturate(PAngleInput, -0.10, 0.10);
    PAngleCons->update(- PTiltAngleSat->value(0) - PAngleFilteredSat-
>value(0));
    PAngleOutput->saturate(PAngleCons, -0.34, 0.34);
}

```

```

    //PVelCons->derivate(PAngleOutput);
    PVelCons1 ->derivate(PAngleOutput);
    double lCriterion8 = -(-10 * ((-0.9771* (XPosCons->value(0)))-10*(-
1.0061*(XVelCons->value(0)))-10*(5.0919*(PAngleCons->value(0)))-
10*(0.42*(PVelCons1->value(0))));
    //double lCriterion9 = (lCriterion8)
    PVelCons->update(lCriterion8);
    PVelCons2->LP1(PVelCons,1.0,0.1);
    PAngleCons1->integer(PVelCons2);
    PAngleCons2->update(PAngleCons1->value(0)*9.81);
    double lCriterion9 = XAccInput1-PAngleCons2;
    XAccCons1 ->update (lCriterion9);
    PVelOutput->saturate(PVelCons, -0.52, 0.52);
    PAccCons->derivate(PVelOutput);
    PAccOutput->saturate(PAccCons, -8.5, 8.5);
}

//*****
//*****
// computeY
//*****
//*****

void computeY (void)
{
    YVelInput->integer(YAccInput);
    YAccFiltered->HP2(YAccInput,1.0,1.0,0.4);
    YVelFiltered->LP2vitesse(YAccInput,1.0,1.0,0.4);
    YPosFiltered->LP2position(YAccInput,1.0,1.0,0.4);
    YVelCons->integer(YAccCons);
    YPosCons->integer(YVelCons);
    double lCriterion4 = (-2 * (YVelFiltered->value(0)) * (YVelCons-
>value(0)) + (YAccFiltered->value(0)) * (YAccInput->value(0)-YAccCons-
>value(0)-(20 * YPosFiltered->value(0) * (YPosCons->value(0)) * 0.5));
    double lCriterion5 = (lCriterion4)- 5.0265*YVelCons->value(0)-
6.3167*YPosCons->value(0);
    YAccCons->update( lCriterion5);
    // Saturation pour respecter les limites en accélération de la PF
    YAccOutput->saturate(YAccCons,-6.0*9.81,6.0*9.81);
    YVelCons->integer(YAccOutput);
    // Saturation pour respecter les limites en vitesse de la PF
    YVelOutput->saturate(YVelCons,-0.5,0.5);
    YPosCons->integer(YVelOutput);
    // Saturation pour respecter les limites en position de la PF
    YPosOutput->saturate(YPosCons, -0.259, 0.259);
}

//*****
//*****
// computeR

```

```

//*****
//*****

void computeR (void)
{
    Y2RAccFiltered->LP2vitesse (YAccInputSat,0.25,0.7,0.3);
    RTiltAngle->update (-asin (Y2RAccFiltered->value (0) / 9.81));
    RTiltAngleSat->saturate (RTiltAngle, -0.24, 0.24);
    //RAngleFiltered->LP1 (RAngleInput, 1.0, 1.0);
    //RAngleFilteredSat->saturate (RAngleFiltered, -0.10, 0.10);
    RAngleFilteredSat->saturate (RAngleInput, -0.10, 0.10);
    RAngleCons->update ( - RTiltAngleSat->value (0) + RAngleFilteredSat-
>value (0));
    RAngleOutput->saturate (RAngleCons, -0.34, 0.34);
    //RVelCons->derivate (RAngleOutput);
    RVelCons1 ->derivate (RAngleOutput);
    double lCriterion10 = -(-10 * ((-0.9771* (YPosCons->value (0)))-10* (-
1.0061*(YVelCons->value (0)))-10*(5.0919*(RAngleCons->value (0)))-
10*(0.42*(RVelCons1->value (0)))));
    //double lCriterion11 = (lCriterion10)
    RVelCons->update (lCriterion10);
    RVelCons2->LP1 (RVelCons,1.0,0.1);
    RAngleCons1->integer (RVelCons2);
    RAngleCons2->update (RAngleCons1->value (0)*9.81);
    double lCriterion11 = YAccInput1-RAngleCons2;
    YAccCons1 ->update (lCriterion11);
    RVelOutput->saturate (RVelCons, -0.52, 0.52);
    RAccCons->derivate (RVelOutput);
    RAccOutput->saturate (RAccCons, -8.5, 8.5);
}

//*****
//*****
//    computeH
//*****
//*****

void computeH (void)
{
    HAccInputSat->saturate (HAccInput,-8.0,8.0);
    HAccFiltered->HP1 (HAccInputSat,1.0,2.0);
    HError->subtract (HAccInputSat,HAccFiltered);
    HErrorFiltered->LP1 (HError,2.0,2.0);
    double lTemp = HErrorFiltered->value (0);
    HErrorFiltered2->update (exp (-10.0*lTemp*lTemp));
    HAccFilteredAB->update (HAccFiltered->value (0) * HErrorFiltered2-
>value (0));
    HErrorAB->subtract (HAccInput,HAccFilteredAB);
    double lTemp2 = HErrorAB->value (0);

```



```

double lTemp3 = HErrorFiltered->value(0);
double lCriterion = tanh( 2 * ( abs(lTemp2) - abs(lTemp3) ) );
lCriterion = (lCriterion + 1) / 2;
HCriterion->update( lCriterion );
// acc_cons = (1-criterion) x acc_filtered_ab + criterion x
acc_filtered
HAccCons->update( (1-lCriterion)*HAccFilteredAB->value(0) +
lCriterion*HAccFiltered->value(0) );
HAccConsLP->LP1(HAccCons, 1.0, 0.1);
HAccConsWashout->HP2(HAccConsLP,-1.0,1.0,2.0);
// Saturation pour respecter les limites en accélération de la PF
HAccOutput->saturate(HAccConsWashout,-6.981,6.981);
HVelCons->integer(HAccOutput);
// Saturation pour respecter les limites en vitesse de la PF
HVelOutput->saturate(HVelCons,-0.698,0.698);
HAngleCons->integer(HVelOutput);
// Saturation pour respecter les limites en orientation de la PF
HAngleOutput->saturate(HAngleCons, -0.34, 0.34);
}

//*****
//*****
// computeZ
//*****
//*****

void computeZ (void)
{
ZVelInput->integer(ZAccInput);
ZAccFiltered->HP2(ZAccInput,1.0,1.0,0.4);
ZVelFiltered->LP2vitesse(ZAccInput,1.0,1.0,0.4);
YPosFiltered->LP2position(ZAccInput,1.0,1.0,0.4);
ZVelCons->integer(ZAccCons);
ZPosCons->integer(ZVelCons);
double lCriterion6 = (-2 * (ZVelFiltered->value(0)) * (ZVelCons-
>value(0)) + (ZAccFiltered->value(0)) * (ZAccInput->value(0) - ZAccCons-
>value(0)-(20 * ZPosFiltered->value(0) * (ZPosCons->value(0)) * 0.5)));
double lCriterion7 = (lCriterion6)- 5.0265*ZVelCons->value(0)-
6.3167*ZPosCons->value(0);
ZAccCons->update( lCriterion7);
// Saturation pour respecter les limites en accélération de la PF
ZAccOutput->saturate(ZAccCons,-5.0*9.81,5.0*9.81);
ZVelCons->integer(ZAccOutput);
// Saturation pour respecter les limites en vitesse de la PF
YVelOutput->saturate(YVelCons,-0.3,0.3);
YPosCons->integer(YVelOutput);
// Saturation pour respecter les limites en position de la PF
YPosOutput->saturate(YPosCons, -0.178, 0.178);
}

```

```
//*****  
//*****  
//    endSim  
//*****  
//*****  
  
void endSim (double acc)  
{  
    acc = abs(acc);  
  
    // On incrémente la variable countIsStop si la vitesse du véhicule  
est nulle  
    // Si la vitesse est non nulle, on remet la variable countIsStop à 0  
    if (acc < 0.1)  
    {  
        countIsStop += PeriodeEchantillonnage ;  
    }  
    else  
    {  
        countIsStop = 0.0;  
    }  
  
    // Si le véhicule est arrêté depuis plus de 20 secondes, on  
décrémente la  
    // variable stopGain pour faire descendre la plate-forme  
    if (countIsStop > 25.0)  
    {  
        stopGain -= (0.1*PeriodeEchantillonnage);  
    }  
    // Si le véhicule n'est plus arrêté, on incrémente stopGain pour  
faire remonter  
    // la plate-forme si jamais elle était descendue, sauf si elle est  
arrivée en  
    // position basse  
    else  
    {  
        if (stopGain > 0.0)  
        {  
            stopGain += (0.1*PeriodeEchantillonnage);  
        }  
    }  
  
    // stopGain doit être compris entre 0.0 et 1.0 !  
    if (stopGain > 1.0)  
    {  
        stopGain = 1.0;  
    }  
    if (stopGain < 0.0)  
    {
```

```

        stopGain = 0.0;
    }
}

//*****
//*****
//    externStep
//*****
//*****
void externStep (double posvitacc[18], double retourposvitacc[18])
{
    //posvitacc[0]  --> X position
    //posvitacc[1]  --> Y position
    //posvitacc[2]  --> Z position
    //posvitacc[3]  --> H angle
    //posvitacc[4]  --> P angle
    //posvitacc[5]  --> R angle
    //posvitacc[6]  --> X velocity
    //posvitacc[7]  --> Y velocity
    //posvitacc[8]  --> Z velocity
    //posvitacc[9]  --> H velocity
    //posvitacc[10] --> P velocity
    //posvitacc[11] --> R velocity
    //posvitacc[12] --> X acceleration
    //posvitacc[13] --> Y acceleration
    //posvitacc[14] --> Z acceleration
    //posvitacc[15] --> H acceleration
    //posvitacc[16] --> P acceleration
    //posvitacc[17] --> R acceleration

    // -----
    // Mise à jour des signaux d'entrée
    // -----

    double RampTime = 10.0 ;
    double coeff = count / (RampTime * 60);
    if (coeff > 1)
    {
        coeff = 1;
    }
    XAccInput->update (coeff * posvitacc[12]);
    PAngleInput->update (coeff * posvitacc[4]);
    YAccInput->update (coeff * posvitacc[13]);
    RAngleInput->update (coeff * posvitacc[5]);
    HAccInput->update (coeff * posvitacc[15]);
    ZAccInput->update (coeff * posvitacc[14]);

    // -----
    // Calcul des signaux de sortie

```

```
// -----  
  
computeX();  
computeP();  
computeY();  
computeR();  
computeH();  
computeZ();  
//endSim(XAccInput->value(0));  
  
// -----  
// Mise à jour de la variable de sortie  
// -----  
  
retourposvitacc[0] = XPosOutput->value(0) * 0.2; // X position  
retourposvitacc[1] = YPosOutput->value(0) * 0.2; // Y position  
retourposvitacc[2] = 0.22 ; // Z position  
retourposvitacc[3] = 0 ; // HAngleOutput->value(0) ; // H angle  
retourposvitacc[4] = PAngleOutput->value(0) ; // P angle  
retourposvitacc[5] = RAngleOutput->value(0) ; // R angle  
retourposvitacc[6] = 0 ; // X velocity  
retourposvitacc[7] = 0 ; // Y velocity  
retourposvitacc[8] = 0 ; // Z velocity  
retourposvitacc[9] = 0 ; // H velocity  
retourposvitacc[10] = 0 ; // P velocity  
retourposvitacc[11] = 0 ; // R velocity  
retourposvitacc[12] = 0 ; // X acceleration  
retourposvitacc[13] = 0 ; // Y acceleration  
retourposvitacc[14] = 0 ; // Z acceleration  
retourposvitacc[15] = 0 ; // H acceleration  
retourposvitacc[16] = 0 ; // P acceleration  
retourposvitacc[17] = 0 ; // R acceleration  
  
for (int i=0; i<18; i++)  
{  
    retourposvitacc[i] = retourposvitacc[i] * stopGain;  
}  
  
// -----  
// Ecriture des résultats  
// -----  
  
myFile << count * PeriodeEchantillonnage << " ";  
  
myFile << XAccInput->value(0) << " ";  
myFile << XAccFiltered->value(0) << " ";  
myFile << XError->value(0) << " ";  
myFile << XErrorFiltered->value(0) << " ";  
myFile << XErrorFiltered2->value(0) << " ";
```

```
myFile << XAccFilteredAB->value(0) << ";";
myFile << XErrorAB->value(0) << ";";
myFile << XCriterion->value(0) << ";";
myFile << XAccCons->value(0) << ";";
myFile << XAccConsWashout->value(0) << ";";
myFile << XAccOutput->value(0) << ";";
myFile << XVelCons->value(0) << ";";
myFile << XVelOutput->value(0) << ";";
myFile << XPosCons->value(0) << ";";
myFile << XPosOutput->value(0) << ";";

myFile << PAngleInput->value(0) << ";";
myFile << X2PAccFiltered->value(0) << ";";
myFile << PTiltAngle->value(0) << ";";
myFile << PTiltAngleSat->value(0) << ";";
myFile << PAngleFiltered->value(0) << ";";
myFile << PAngleFilteredSat->value(0) << ";";
myFile << PAngleCons->value(0) << ";";
myFile << PAngleOutput->value(0) << ";";
myFile << PVelCons->value(0) << ";";
myFile << PVelOutput->value(0) << ";";
myFile << PAccCons->value(0) << ";";
myFile << PAccOutput->value(0) << ";";

myFile << YAccInputSat->value(0) << ";";
myFile << YAccFiltered->value(0) << ";";
myFile << YError->value(0) << ";";
myFile << YErrorFiltered->value(0) << ";";
myFile << YErrorFiltered2->value(0) << ";";
myFile << YAccFilteredAB->value(0) << ";";
myFile << YErrorAB->value(0) << ";";
myFile << YCriterion->value(0) << ";";
myFile << YAccCons->value(0) << ";";
myFile << YAccConsWashout->value(0) << ";";
myFile << YAccOutput->value(0) << ";";
myFile << YVelCons->value(0) << ";";
myFile << YVelOutput->value(0) << ";";
myFile << YPosCons->value(0) << ";";
myFile << YPosOutput->value(0) << ";";

myFile << RAngleInput->value(0) << ";";
myFile << Y2RAccFiltered->value(0) << ";";
myFile << RTiltAngle->value(0) << ";";
myFile << RTiltAngleSat->value(0) << ";";
myFile << RAngleFiltered->value(0) << ";";
myFile << RAngleFilteredSat->value(0) << ";";
myFile << RAngleCons->value(0) << ";";
myFile << RAngleOutput->value(0) << ";";
myFile << RVelCons->value(0) << ";";
```

```
myFile << RVelOutput->value(0) << ";";
myFile << RAccCons->value(0) << ";";
myFile << RAccOutput->value(0) << ";";

myFile << HAccInput->value(0) << ";";
myFile << HAccFiltered->value(0) << ";";
myFile << HError->value(0) << ";";
myFile << HErrorFiltered->value(0) << ";";
myFile << HErrorFiltered2->value(0) << ";";
myFile << HAccFilteredAB->value(0) << ";";
myFile << HErrorAB->value(0) << ";";
myFile << HCriterion->value(0) << ";";
myFile << HAccCons->value(0) << ";";
myFile << HAccConsWashout->value(0) << ";";
myFile << HAccOutput->value(0) << ";";
myFile << HVelCons->value(0) << ";";
myFile << HVelOutput->value(0) << ";";
myFile << HAngleCons->value(0) << ";";
myFile << HAngleOutput->value(0) << ";";

myFile << ZAccInput->value(0) << ";";
myFile << ZAccInputInt->value(0) << ";";
myFile << ZVit->value(0) << ";";
myFile << ZVitInt->value(0) << ";";
myFile << ZVitIntFiltered->value(0) << ";";
myFile << ZVitIntFiltered2->value(0) << ";";
myFile << ZPosCons->value(0) << ";";
myFile << ZPosOutput->value(0) << ";";
myFile << ZVelCons->value(0) << ";";
myFile << ZVelOutput->value(0) << ";";
myFile << ZAccCons->value(0) << ";";
myFile << ZAccOutput->value(0) << "\n";

// myFile << XAccInput->value(0) << ";";
// myFile << countIsStop << ";";
// myFile << stopGain << "\n";

// -----
// Comptage
// -----
count++;

// -----
// Si on veut être en statique mais calculer
// tout de même les commandes
// -----

/*
for (int i=0; i<18; i++)
```

```

    {
        retourposvitacc[i] = 0.0;
    }
    */
}

```

C. Code source pour intégrer XSens Motion Tracker dans le logiciel SCANeRstudio

C1. Fichiers C++

ExportChannelManager.cpp

```

#include "ExportChannelManager.h"

ExportChannelManager::ExportChannelManager(const std::string& p_file)
: m_counterForEC(0)
{
    cfg.getConfig(p_file);
    createDataInterfaces();
}

ExportChannelManager::~ExportChannelManager(void)
{
}

void ExportChannelManager::createDataInterfaces()
{
    std::string tmp;

    //get channel for accX
    if( cfg.get(DTR_ACCELERATION_X,tmp) )
    {
        m_exportChannels[DTR_ACCELERATION_X] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
    }

    //get channel for accY
    if( cfg.get(DTR_ACCELERATION_Y,tmp) )
    {
        m_exportChannels[DTR_ACCELERATION_Y] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
    }
}

```

```
//get channel for accZ
if( cfg.get(DTR_ACCELERATION_Z,tmp) )
{
    m_exportChannels[DTR_ACCELERATION_Z] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
}

//get channel for heading speed
if( cfg.get(DTR_SPEED_H,tmp) )
{
    m_exportChannels[DTR_SPEED_H] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
}

//get channel for pitch speed
if( cfg.get(DTR_SPEED_P,tmp) )
{
    m_exportChannels[DTR_SPEED_P] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
}

//get channel for roll speed
if( cfg.get(DTR_SPEED_R,tmp) )
{
    m_exportChannels[DTR_SPEED_R] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
}

/////heading acc
//if(cfg.get(DTR_ACC_H,tmp))
//{
//    m_exportChannels[DTR_ACC_H] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
//}

/////pitch acc
//if(cfg.get(DTR_ACC_P,tmp))
//{
//    m_exportChannels[DTR_ACC_P] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
//}

/////roll acc
//if(cfg.get(DTR_ACC_R,tmp))
//{
//    m_exportChannels[DTR_ACC_R] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
//}

if( cfg.get(DTR_QUAT1,tmp) )
{
    m_exportChannels[DTR_QUAT1] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
}

if( cfg.get(DTR_QUAT2,tmp) )
{
    m_exportChannels[DTR_QUAT2] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
}
```



```

        if( cfg.get(DTR_QUAT3,tmp) )
        {
            m_exportChannels[DTR_QUAT3] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
        }

        if( cfg.get(DTR_QUAT4,tmp) )
        {
            m_exportChannels[DTR_QUAT4] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
        }
    }

void ExportChannelManager::set(const std::string& p_dataName, float p_val)
{
    std::string tmp;
    if( cfg.get(p_dataName,tmp) )
    {
        Com_setShortData(m_exportChannels[p_dataName] , "no" ,
        atoi(tmp.c_str()) );
        Com_setLongData( m_exportChannels[p_dataName] , "frame_nb",
m_counterForEC);
        Com_setFloatData(m_exportChannels[p_dataName] , "val" ,
p_val);
    }
    else
    {
        std::cout<<"no DataInterface to send "<<p_dataName<<std::endl;
    }

    //Com_setShortData(m_exportChannels[p_dataName] , "no" ,          26
);
    //Com_setLongData( m_exportChannels[p_dataName] , "frame_nb",
m_counterForEC);
    //Com_setFloatData(m_exportChannels[p_dataName] , "val" ,          2);
}

void ExportChannelManager::send()
{
    Com_updateOutputs(UT_AllData);
    m_counterForEC ++;
}

```

XSensManager.cpp

```

#include "ExportChannelManager.h"

ExportChannelManager::ExportChannelManager(const std::string& p_file)
: m_counterForEC(0)
{
    cfg.getConfig(p_file);
    createDataInterfaces();
}

ExportChannelManager::~ExportChannelManager(void)
{

```

```
}

void ExportChannelManager::createDataInterfaces ()
{
    std::string tmp;

    //get channel for accX
    if( cfg.get(DTR_ACCELERATION_X,tmp) )
    {
        m_exportChannels[DTR_ACCELERATION_X] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
    }

    //get channel for accY
    if( cfg.get(DTR_ACCELERATION_Y,tmp) )
    {
        m_exportChannels[DTR_ACCELERATION_Y] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
    }

    //get channel for accZ
    if( cfg.get(DTR_ACCELERATION_Z,tmp) )
    {
        m_exportChannels[DTR_ACCELERATION_Z] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
    }

    //get channel for heading speed
    if( cfg.get(DTR_SPEED_H,tmp) )
    {
        m_exportChannels[DTR_SPEED_H] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
    }

    //get channel for pitch speed
    if( cfg.get(DTR_SPEED_P,tmp) )
    {
        m_exportChannels[DTR_SPEED_P] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
    }

    //get channel for roll speed
    if( cfg.get(DTR_SPEED_R,tmp) )
    {
        m_exportChannels[DTR_SPEED_R] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
    }

    ///heading acc
    //if(cfg.get(DTR_ACC_H,tmp))
    //{
    //    m_exportChannels[DTR_ACC_H] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
    //}

    ///pitch acc
    //if(cfg.get(DTR_ACC_P,tmp))
    //{
```

```

        //      m_exportChannels[DTR_ACC_P] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
        //}

        ////roll acc
        //if(cfg.get(DTR_ACC_R,tmp))
        //{
        //      m_exportChannels[DTR_ACC_R] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
        //}

        if( cfg.get(DTR_QUAT1,tmp) )
        {
                m_exportChannels[DTR_QUAT1] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
        }

        if( cfg.get(DTR_QUAT2,tmp) )
        {
                m_exportChannels[DTR_QUAT2] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
        }

        if( cfg.get(DTR_QUAT3,tmp) )
        {
                m_exportChannels[DTR_QUAT3] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
        }

        if( cfg.get(DTR_QUAT4,tmp) )
        {
                m_exportChannels[DTR_QUAT4] =
Com_declareOutputData(NETWORK_IUSER_EXPORTCHANNEL , atoi(tmp.c_str()) );
        }
}
void ExportChannelManager::set(const std::string& p_dataName, float p_val)
{
        std::string tmp;
        if(  cfg.get(p_dataName,tmp) )
        {
                Com_setShortData(m_exportChannels[p_dataName] , "no" ,
                atoi(tmp.c_str()) );
                Com_setLongData( m_exportChannels[p_dataName] , "frame_nb",
m_counterForEC);
                Com_setFloatData(m_exportChannels[p_dataName] , "val" ,
                p_val);
        }
        else
        {
                std::cout<<"no DataInterface to send "<<p_dataName<<std::endl;
        }

        //Com_setShortData(m_exportChannels[p_dataName] , "no" ,          26
);
        //Com_setLongData( m_exportChannels[p_dataName] , "frame_nb",
m_counterForEC);
        //Com_setFloatData(m_exportChannels[p_dataName] , "val" ,          2);
}

void ExportChannelManager::send()
{

```

```
Com_updateOutputs(UT_AllData);  
m_counterForEC ++;  
}
```


ETUDE DES LOIS DE COMMANDE DE LA PLATEFORME DE SIMULATION DE CONDUITE ET INFLUENCE SUR LE MAL DE SIMULATEUR

RESUME : La simulation de conduite est fortement utilisée dans la recherche et le développement pour l'industrie automobile. Les simulateurs de conduite sont utilisés pour évaluer les prototypes véhicules pour la dynamique du véhicule et les systèmes d'aide à la conduite. Cependant, l'utilisation des simulateurs de conduite induit une problématique scientifique qui peut limiter son développement. En raison de son principe même, le simulateur de conduite ne restitue pas des mouvements du véhicule à l'échelle 1. Ce verrou cause des phénomènes de mal du simulateur qu'il est important d'étudier. Cette thèse propose d'étudier des méthodes et outils à mettre en œuvre dans les simulateurs de conduite statique ou dynamique. De cette mise en œuvre, des études sur le mal du simulateur sont menées grâce à des mesures objectives (via un capteur de suivi de mouvement, plate-forme de stabilité du corps, électromyographie) et subjectives (par l'intermédiaire de questionnaires). Des solutions algorithmiques et matérielles sont proposées et évaluées dans le contexte de la simulation de conduite. Les approches proposées dans cette thèse pour réduire le mal du simulateur sont:

- Elaborer et évaluer les algorithmes de contrôle de la plate-forme mobile hexapode: sept algorithmes différents sont mis en œuvre.
- Mesurer les effets liés au mal de simulateur sur les sujets aux niveaux vestibulaire, neuromusculaire et posturale.
- Evaluer l'influence de l'implication des sujets sur le mal de simulateur (conducteurs et passagers).

Mots clés : Mal du simulateur, simulateur de conduite, algorithme de contrôle, évaluation objective, évaluation subjective, IR (Illness Rating), analyse d'EMG, posturale stabilité.

STUDY OF CONTROL COMMAND OF DYNAMIC PLATFORM FOR DRIVING SIMULATION AND INFLUENCE ON SIMULATOR SICKNESS

ABSTRACT : Simulation has been intensively involved nowadays in research and development for automotive industry. Driving simulators are one of those simulation techniques which are used to evaluate the prototypes for the vehicle dynamics and driving assistance systems. However with the driving simulator, there is a lock associated with its use. Because representing a permanent scenario as scale 1 is quite difficult. Because of that difficulty, motion/simulator sickness is an inevitably important topic to study.

This thesis proposes to explore methods and tools to implement in static or dynamic simulators. In this implementation, studies of simulator sickness are conducted with objective measures (via a motion tracking sensor, platform for body stability, electromyography) and subjective (through questionnaires). These algorithmic or hardware solutions studies should be defined and applied at simulators. The proposed approaches to reduce or avoid simulator sickness in this thesis are:

- Building control algorithms of motion hexapod platform: seven different algorithms are implemented.
- Measuring the effects of inertia on subjects at vestibular, neuromuscular and postural levels.
- Assessing the involvement of subjects (drivers and passengers).

Keywords : Simulator sickness, driving simulator, motion cueing, objective evaluation, subjective evaluation, IR (Illness Rating), EMG analysis, postural stability